

ADM100

**System Administration I of SAP
S/4HANA and SAP Business Suite**

**PARTICIPANT HANDBOOK
INSTRUCTOR-LED TRAINING**

Course Version: 24

Course Duration: 5 Day(s)

Material Number: 50166725

SAP Copyrights, Trademarks and Disclaimers

© 2024 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <https://www.sap.com/corporate/en/legal/copyright.html> for additional trademark information and notices.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials may have been machine translated and may contain grammatical errors or inaccuracies.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

This information is displayed in the instructor's presentation



Demonstration



Procedure



Warning or Caution



Hint



Related or Additional Information



Facilitated Discussion



User interface control

Example text

Window title

Example text

Contents

vii Course Overview

1 Unit 1: Fundamentals of AS ABAP

3	Lesson: Explaining the Basic Architecture of AS ABAP
11	Lesson: Configuring SAP Logon
17	Lesson: Describing the Process of a System Logon (AS ABAP)
21	Lesson: Setting Up Logon Groups in AS ABAP
25	Lesson: Appendix - Executing Basic Administration Transactions

37 Unit 2: Starting and Stopping AS ABAP

39	Lesson: Understanding the Concept of Start and Stop Procedures
53	Lesson: Introducing Different Tools for Start and Stop Procedures
63	Lesson: Starting and Stopping AS ABAP
77	Lesson: Log and Trace Information for System Start and Stop

99 Unit 3: System Configuration

101	Lesson: Configuring SAP Systems via Profile Parameters
111	Lesson: Setting Up Operation Modes for Resource Configuration
125	Lesson: Executing ABAP Task Lists for Automated Configuration

137 Unit 4: Introduction to Database Administration

139	Lesson: Describing the Architecture of Database Systems
149	Lesson: Scheduling and Monitoring Database Actions
159	Lesson: Performing Regular Database Checks
165	Lesson: Appendix: Describing the Concept of SAP Data Archiving

175 Unit 5: Users and Authorizations

177	Lesson: Creating, Copying, and Maintaining User Master Records
187	Lesson: Maintaining User Authorizations with Roles and Profiles
199	Lesson: Define Profile Parameters to Secure Passwords and Logins
209	Lesson: Evaluate Users and Authorizations

217 Unit 6: RFC Communication

219	Lesson: Describing the Communication Between SAP Systems Using RFC
225	Lesson: Managing RFC Resources
231	Lesson: Creating RFC Destinations

239 Unit 7: Maintaining SAP Software

- | | |
|-----|---|
| 241 | Lesson: Describing Maintenance Levels |
| 245 | Lesson: Implementing SAP Notes |
| 253 | Lesson: Preparing Software Maintenance using Maintenance Planner |
| 271 | Lesson: Describing Support Package Manager (SPAM) and Software Update Manager (SUM) |
| 277 | Lesson: Importing Support Packages with the Support Package Manager (SPAM) |
| 285 | Lesson: Importing Support Packages with the Software Update Manager (SUM) |
| 289 | Lesson: Explaining SAP System Upgrade and SAP S/4HANA Conversion |

301 Unit 8: Printing

- | | |
|-----|---|
| 303 | Lesson: Exploring SAP Spool System |
| 307 | Lesson: Creating Output Devices |
| 317 | Lesson: Working with Device Types |
| 325 | Lesson: Working with Spool Servers |
| 331 | Lesson: Monitoring and Troubleshooting Printing Issues |
| 343 | Lesson: Planning the SAP Print Architecture |
| 349 | Lesson: Appendix: Managing Output Devices with Printing Assistance for Landscapes (PAL) |

361 Unit 9: Background Processing

- | | |
|-----|--|
| 363 | Lesson: Understanding the Concept of Background Processing |
| 371 | Lesson: Monitoring Background Processing |
| 377 | Lesson: Setting Up the Background Processing Environment |
| 387 | Lesson: Scheduling Background Jobs |
| 399 | Lesson: Troubleshooting Background Issues |

411 Unit 10: System Monitoring and Troubleshooting

- | | |
|-----|--|
| 413 | Lesson: Using Local Monitoring Tools |
| 427 | Lesson: Explaining the Concept of Monitoring Tools in SAP Solution Manager |
| 437 | Lesson: Tracing and Logging in the SAP System |
| 445 | Lesson: Troubleshooting Procedures |

456 Glossary

Course Overview

TARGET AUDIENCE

This course is intended for the following audiences:

- Support Consultant
- Technology Consultant
- Project Manager
- Systems Architect
- System Administrator

UNIT 1

Fundamentals of AS ABAP

Lesson 1

Explaining the Basic Architecture of AS ABAP

3

Lesson 2

Configuring SAP Logon

11

Lesson 3

Describing the Process of a System Logon (AS ABAP)

17

Lesson 4

Setting Up Logon Groups in AS ABAP

21

Lesson 5

Appendix - Executing Basic Administration Transactions

25

UNIT OBJECTIVES

- Explain the basic architecture of the Application Server ABAP
- Configure SAP Logon
- Explain the process of an AS ABAP system logon using a custom application server
- Describe the concept of dialog work process multiplexing
- Explain the benefits of logon groups in an AS ABAP system
- Create a logon group in an AS ABAP based SAP system
- List basic administration transactions and explain their use

Explaining the Basic Architecture of AS ABAP

LESSON OVERVIEW

This lesson introduces the basic architecture of SAP systems. It explains the terms “system” and “(application server) instance”. It also introduces the individual processes of Application Server ABAP based on simple client/server configuration.

Business Example

You are a new member of the SAP system administration team in your company that runs SAP S/4HANA Server systems either on premise or in SAP S/4HANA Cloud, private edition. Your task is to prepare the installation of a new AS ABAP-based SAP system. For this, you need to plan the system architecture and to decide how you are going to distribute the required SAP system processes among the available hardware.

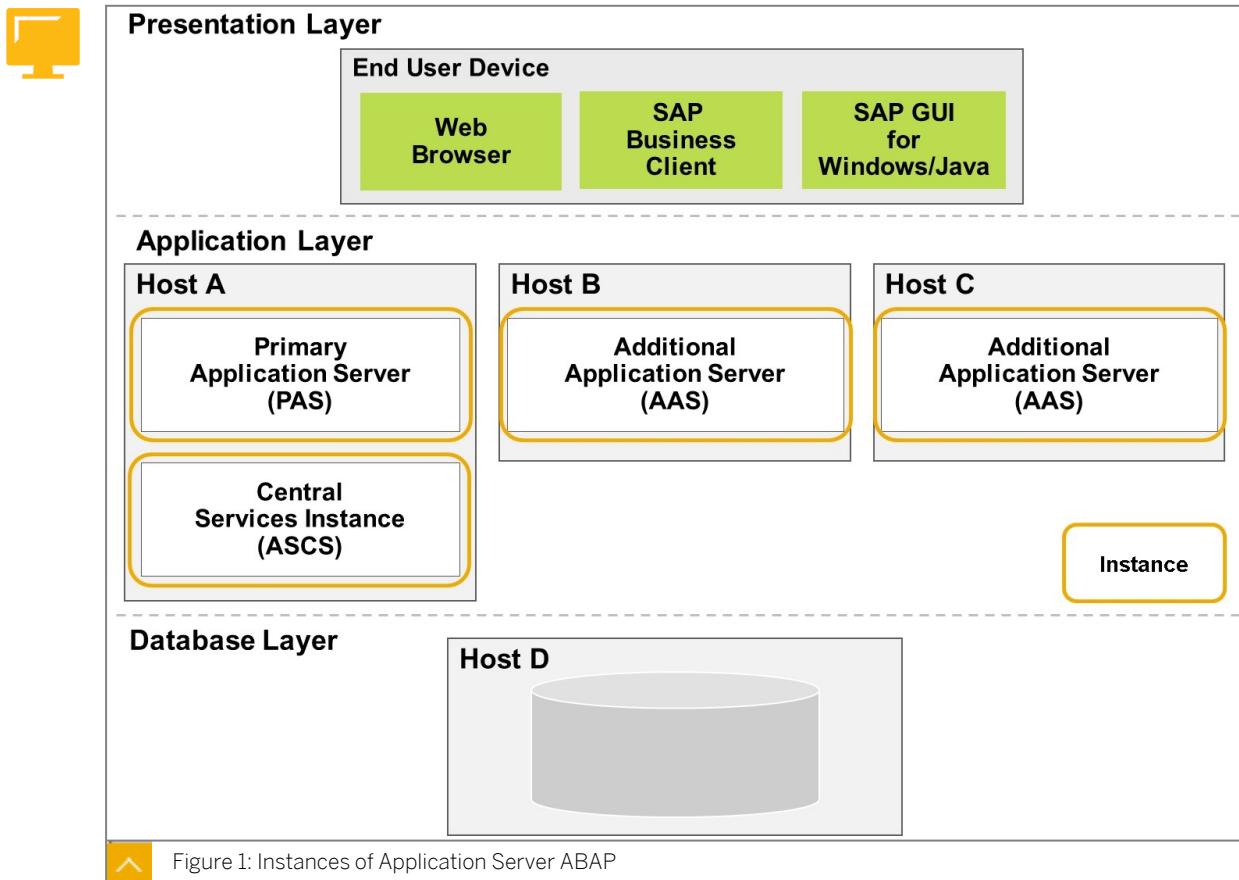


LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the basic architecture of the Application Server ABAP

Application Server Instances of an AS ABAP Based SAP System



An **SAP system** is an installed software that provides a defined set of functionalities that are part of an SAP solution. These functionalities are implemented in a set of software components. An SAP system is installed and configured as a unit. It always consists of exactly one logical database (except in special circumstances), several (application server) instances, and optional components. An SAP system is always identified by a unique system ID (SID), which consists of 3 letters or digits (for example, PRD). The database usually has the same system ID (DB SID) as the SAP system (SAP SID).

Many SAP systems are based on ABAP technology. The remaining part of this lesson focusses on SAP systems of this kind.



Note:

Examples of SAP systems that are based on ABAP include SAP S/4HANA Server, SAP ECC and SAP BW.

When you install an SAP system, you have the option of separating the processes at application level from those at database level. You can install and operate the database for an SAP system on a separate host, away from the instances of the application server ABAP (see figure *Instances of Application Server ABAP*).

An (SAP) **instance** (also known as *application server instance* to distinguish this term from the term “instance” that is used object-oriented programming) is an administrative unit that combines one or more system services. The services provided by an instance are started or

stopped together. Every SAP system consists of one or more application server instances (AS instance) and one central services instance. Application server instances provide the actual data processing functions of a system and offer the corresponding services. A central services instance provides services for communication between instances and lock management. These services are only available once per system.



Hint:

An application server instance is often referred to as simply "**application server**". The terms are often used synonymously.

Instances of one SAP system can run on a single physical host, or they can be distributed among several hosts. An instance can be uniquely identified by the host name, an ID of the system it belongs to, and a two-digit instance number. You set parameters for all the services of an instance using a common instance profile. Each instance has its own, separate memory areas.

The following factors have an influence on the design of an SAP system:

- Release of the SAP system (due to architecture changes throughout the releases)
- Decision of the SAP customer (scalability, high availability)

The first application server instance that you install when you install an AS ABAP based SAP system is called *Primary Application Server (PAS)*. In case that you decide to install additional application server instances, these instances are then called *Additional Application Server (AAS)*.

In addition to the application server instances, the SAP system provides central services on application server level which also form a separate instance. This instance is also known as "**ABAP Central Services Instance**" and is often abbreviated as "**ASCS instance**".



Note:

In previous releases of AS ABAP which did not have any ASCS instance (AS ABAP ≤ 7.03 that is up to SAP Enhancement package 6 for SAP ERP 6.0)), the first application server instance that you have installed also contained the processes of the central services instance. This application server instance then was called *central instance*, and the other application server instances were called *dialog instance*.

There are different installation options for an SAP system. When the PAS and database (and the ASCS, if any) are installed on the same host, it is called **central system**. If all instances and the database run on separate hosts, it is called **distributed system**.

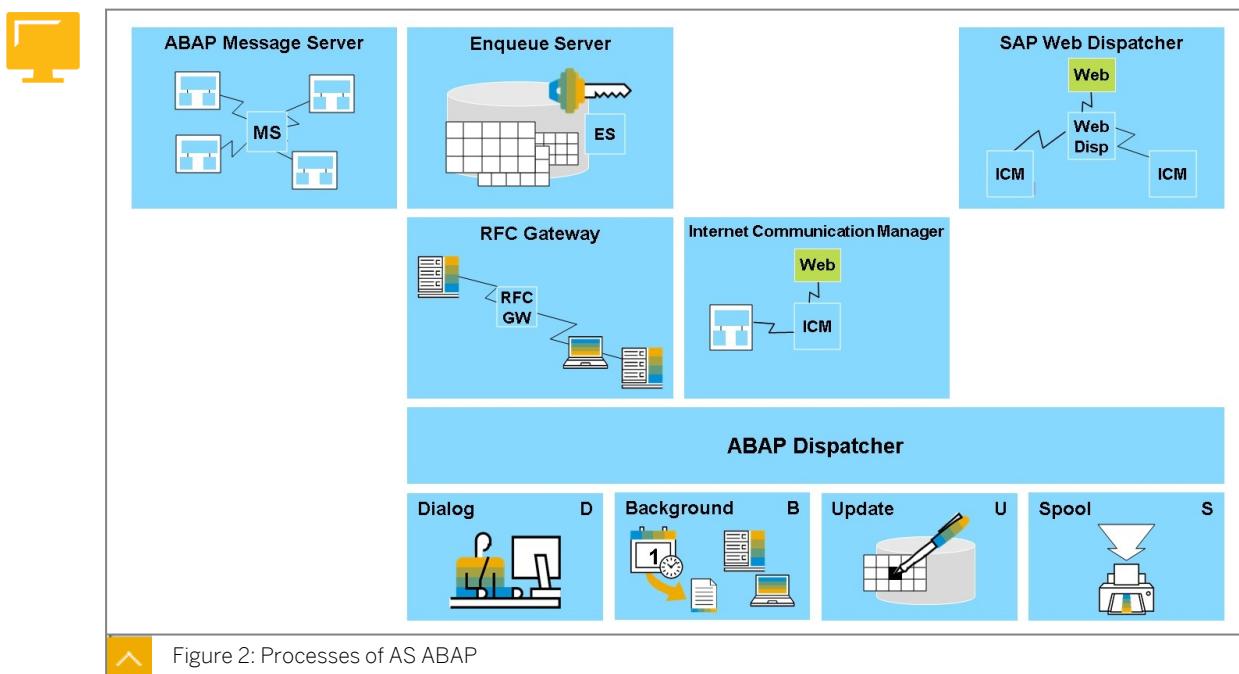
Processes of the Application Server ABAP

The SAP runtime system consists of a large number of parallel processes on operating system level that work together.

In AS ABAP, the processes on every application server include the ABAP dispatcher and a number of work processes (see the following figure):

- Every application server instance contains a **dispatcher** that manages different types of work processes, which take on different tasks when executing the business processes of the SAP system. The dispatcher distributes the requests to the work processes.

- **Dialog work processes** fulfill all requests for the execution of dialog steps triggered by an active user. Additionally, dialog work processes work on requests that are not covered by other work process types (see below). Every dispatcher requires at least two dialog work processes.
- **Background work processes** execute programs that run without user interaction. You need at least two background work processes for each SAP system. You can configure more than one background work process for each dispatcher.
- **Spool work processes** pass sequential data flows on to output devices (printers). At least one spool work process is required for each SAP system. You can configure more than one spool work process per dispatcher.
- **Update work processes** execute update requests. Similar to spool work processes, you need at least one update work process per SAP system and you can configure more than one per dispatcher.



As the figure *Processes of AS ABAP* shows, in addition to the dispatcher and its work processes, the AS ABAP runtime system also provides additional, separate services for internal and external communication:

- The **ABAP Message Server** (MS) handles the communication between the distributed dispatchers within the AS ABAP. It enables scalability of several parallel application servers. The Message Server provides the AS ABAP with a central message service for internal communication, such as starting updates, requesting and removing locks, or triggering background requests. It also provides information about which application servers of the system are currently available. The Message Server is configured only once per SAP system.
- The **Enqueue Server** (ES) manages the lock table that contains the logical database locks of the SAP system. There is one enqueue server per SAP system and it is part of the ASCS instance.

- The **RFC Gateway** reader (RFC GW) enables communication between SAP systems, or between SAP systems and external application systems. One GW exists per dispatcher. Communication from other SAP systems via Remote Function Call (RFC) is received by the RFC GW.



Caution:

Do not confuse the RFC Gateway (part of SAP runtime environment (kernel)) with the software component SAP_GWFND (*SAP Gateway Foundation*) which enables the communication using OData.

- The **Internet Communication Manager** (ICM) enables the communication with the SAP system using web protocols such as HTTP, HTTPS and SMTP. The ICM receives requests from the client and either performs them directly or forwards them to the dispatcher of its application server instance for processing. It can also direct web requests from its application server instance to a Web server and send the response back to the application server instance. In an SAP S/4HANA server system you have always one ICM process per dispatcher configured. If you run an SAP Business Suite system on AS ABAP, you can configure no or a maximum of one ICM process per dispatcher.
- SAP Web Dispatcher** is used to distribute HTTP(S) requests across all instances of an SAP system. It is the central point of entry to the SAP system from the Internet. For each incoming request, it must decide on the application server instance to which it forwards the request. When it accepts a connection, it distributes the requests to ensure an even distribution across the application servers (load balancing).

The SAP Web Dispatcher is usually installed in the demilitarized zone (DMZ) before the SAP system, from where it distributes incoming HTTP(S) requests to the individual application servers of the SAP system according to specified rules.



Note:

In the latest versions of AS ABAP (as of SAP kernel version 7.53), it is also possible to install an SAP Web Dispatcher as part of the ABAP Central Service instance (ASCS). This might be useful for development systems because you then do not need an additional installation. For more information, see SAP Note [3115889](#) – SAP Web Dispatcher embedded deployment in an ASCS/SCS instance.

The services that an application server can provide are determined by the type of work processes it has. An application server can take on several roles, such as a dialog server being simultaneously an update server if it provides several dialog work processes and at least one update work process.

Architecture of AS ABAP

As the administrator of an SAP system, you use profile parameters to configure the number and types of work processes for each instance with a dispatcher.

**Note:**

To display an overview of the application servers of an AS ABAP installation, call transaction SM51 (*Overview of AS Instances*).

To display an overview of the work processes of the application server where you are logged on, call transaction SM50 (*Work Processes*).

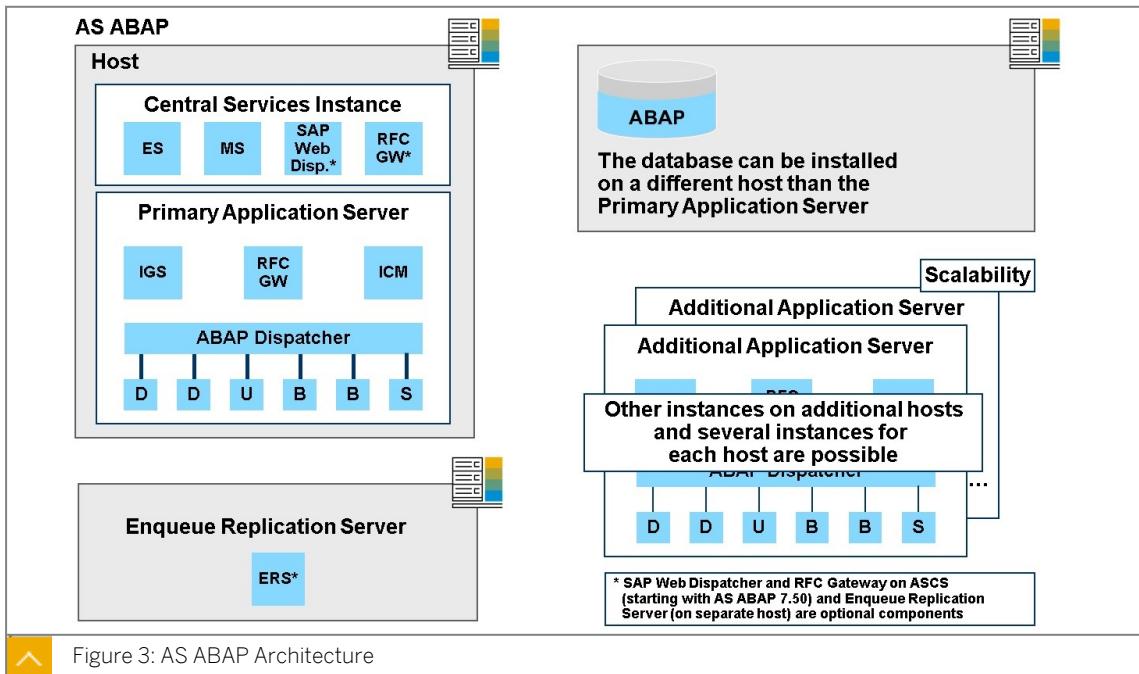


Figure 3: AS ABAP Architecture

An SAP system based on AS ABAP provides a complete development and runtime environment for ABAP-based applications. It consists of at least one application server instance and – if applicable – exactly one central services instance.

- The **ABAP Central Services Instance (ASCS)** provides central services of AS ABAP, namely the message service for the internal communication and the enqueue service for the lock management.

It is not possible to log on to the ASCS as a dialog user. This is why the ASCS is not called an application server.

The ASCS is located in the file system under `/usr/sap<SID>/ASCS<instance_number>`.

**Note:**

ASCS is optional as of AS ABAP 7.00 and is the default installation option in AS ABAP 7.31 and later.

In case your system has an ASCS instance, the lock management is being offered by the enqueue server process that is part of the ASCS.

If not, there will be at least one enqueue work process on an application server instance. The enqueue work process (independent of where it runs) maintains the lock table with its logical database locks of the SAP system. There should be no more than one instance offering enqueue services within one SAP system.

- The **Primary Application Server (PAS)** is the first AS instance installed on the SAP system. If the system does not have an ASCS, the PAS also includes the processes for lock management and internal communication.

The PAS is located in the file system under `/usr/sap/<SID>/D<instance_number>`.

**Note:**

Before AS ABAP 7.50, the PAS file system was located at `/usr/sap/<SID>/DVEBMGS<instance_number>`.

- All other application server instances of an SAP system are called **Additional Application Servers (AAS)**.

The AAS is located in the file system under `/usr/sap/<SID>/D<instance_number>`.

When you log on to the AS ABAP via SAP GUI for Windows or SAP GUI for Java using logon groups, the Message Server performs a load distribution of users to the available application servers. This load distribution which takes place during the logon procedure is also known as *logon load balancing*. After the load balancing by the Message Server, the SAP GUI communicates directly with the dispatcher of the selected application server instance. The user remains logged on to this application server until the logoff.

**Note:**

To display an overview of users who are logged on to the same application server as you, or in another view, are logged on to the complete SAP system, call transaction SM04 (*User Sessions*).

To see which application server you are logged on to, choose *System → Status*.

**LESSON SUMMARY**

You should now be able to:

- Explain the basic architecture of the Application Server ABAP

Unit 1

Lesson 2

Configuring SAP Logon

LESSON OVERVIEW

In this lesson, you will learn how to configure the SAP Logon program.

Business Example

You want to enable your end users to log on to the SAP system using the SAP logon.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Configure SAP Logon

The SAP Logon Program

The *SAP Logon* program provides users with a convenient logon to an SAP system with the SAP GUI for Windows (from Windows-based front ends) or with the SAP GUI for Java (for other operating systems). *SAP Logon* evaluates various configuration files. Some of these files can also be edited using *SAP Logon*.

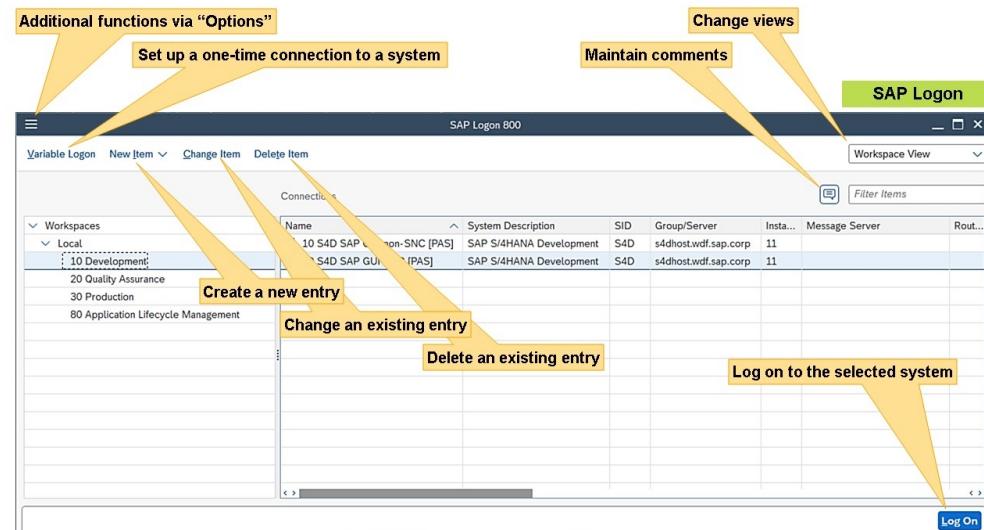
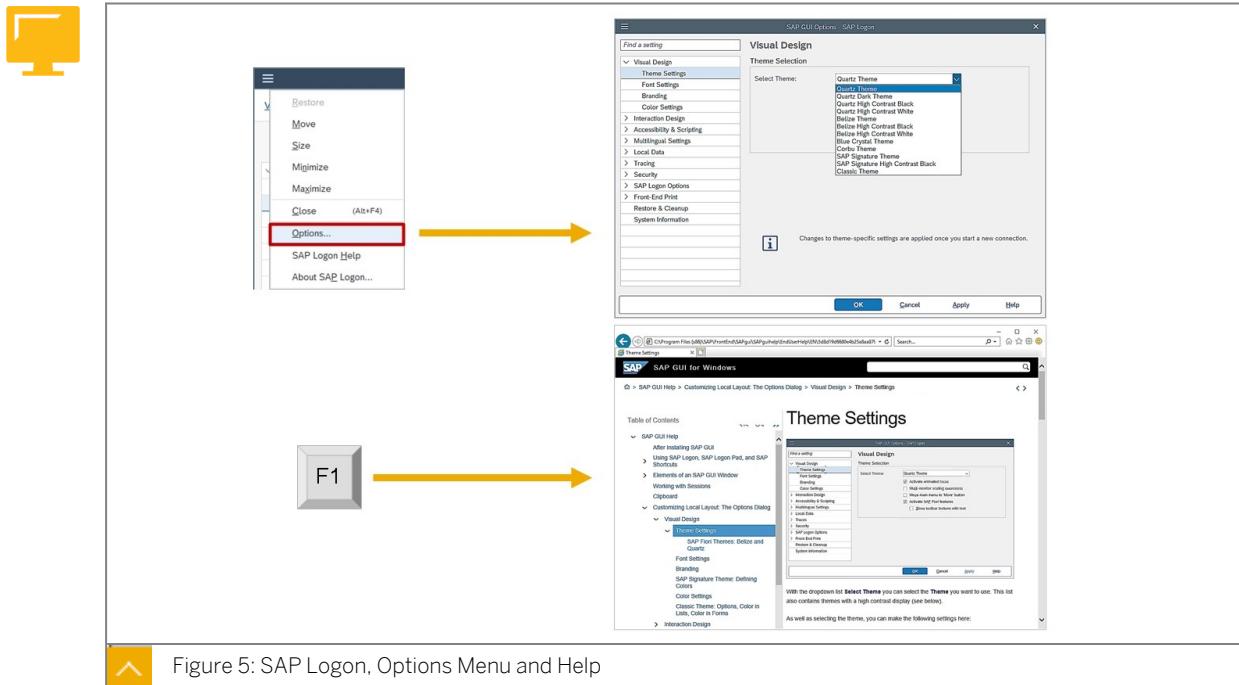


Figure 4: SAP Logon, Main Screen

The figure *SAP Logon, Main Screen* shows the options available to you when you start this program.

Simply put, the main purpose of *SAP Logon* is to start the SAP GUI (*sapgui* program on operating system level of the user's frontend) for a selected SAP system with specific parameters (also see section *Addendum: SAP GUI Connection String*).



As shown in the figure *SAP Logon, Options Menu and Help*, you can configure a variety of general settings using the *Options* in SAP Logon. You can display the documentation (help) for SAP Logon by choosing *F1*.



Note:

SAP Note [147519](#) – *Maintenance strategy / deadlines for SAP GUI for Windows / SAP GUI for Java* explains the maintenance strategy for SAP GUI for Windows and SAP GUI for Java. Concerning expected release dates for patch levels or new versions of SAP GUI for Windows, see SAP Note [1053737](#) – *Expected release dates for SAP GUI for Windows*.

Configuration Files for SAP Logon

The SAP GUI for Windows and Java stores all of its local configuration information and user preferences in files. The figure *SAP Logon Configuration Files* shows the relevant configuration files for SAP Logon.

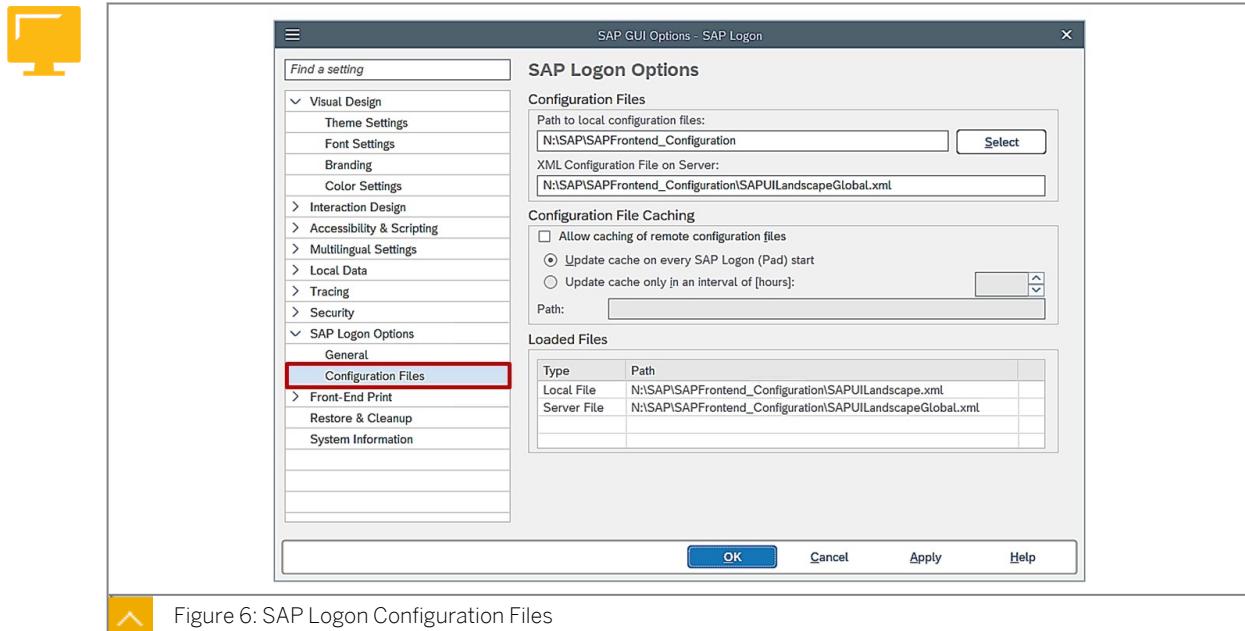


Figure 6: SAP Logon Configuration Files

File Name	Contains	Access
SAPUILandscape.xml	System information and fast access	read/write
SAPUILandscapeGlobal.xml	SID, texts, and ports of message servers and SAP-router connections	read

**Note:**

If SAP Business Client is installed on the front-end computer with SAP Logon, it also uses these files.

In the **SAPUILandscape.xml** file, the following information is stored:

- Connections to SAP systems defined by the user
- Hierarchical structure on the SAP Logon items when using hierarchical view
- User specific notes attached to a connection string
- System descriptions
- Message server entries
- Router entries
- URLs for getting a system status of an SAP system

When SAP GUI is started for the first time, these local configuration files are created empty or – if applicable – they import the values stored in old configuration files from SAP Logon versions before 7.40. This import only happens once.

**Note:**

You can create, display and edit SAP UI Landscape XML data centrally with the SAP UI Landscape Maintenance Tool which can be either accessed via transaction **SLMT** or by starting report **RSLSMT**. You find more information on the functionality in the system documentation of the corresponding transaction or report (i-button) and in SAP Note [231166](#) – SAP UI Landscape maintenance tool.

As of SAP GUI 8.00, backup files are created whenever SAP Logon writes a local SAP UI Landscape file. For more information, see SAP Note [3263097](#) – SAP UI Landscape: Backup of local configuration files.

The file **SAPUILandscapeGlobal.xml** has the same structure as **SAPUILandscape.xml** and can be used to provide a certain set of connection data to many end users.

**Hint:**

If you want to provide the server configuration file(s) in SAP UI Landscape format for your SAP Logon (Pad) users, SAP Note [2075073](#) – SAP Logon (Pad): Create/distribute server configuration file in the SAP UI landscape format provides additional information.

The *SAP Frontend Installation Guide* and the *SAP GUI Administration Guide* contain additional information. They are available from the *SAP GUI for Windows* section on SAP Help Portal (https://help.sap.com/docs/sap_gui_for_windows).

From section *SAP Logon Configuration* of the *SAP GUI Administration Guide*, you can jump to the *SAP UI Landscape Configuration Guide* which contains additional information on configuration files.

Changing Entries in the Configuration File

If you would like to log on to an SAP system that is not yet known in the configuration files, in SAP Logon you can choose *New Item* → *Connection* → *User Specified System*. You can then enter all of the system information that is relevant for a logon, such as the name of the host where you want to contact an instance, its instance number, the System ID, and a description.

**Note:**

You can choose between a connection to a custom application server or to select a logon group. When you log on using a logon group, the ABAP message server is always contacted first, to identify the application server with the best performance within the selected logon group.

You can change the settings for a connection, such as its *Description* name, using the *Change Item* button. On the *Network* tab page, you can activate the use of *Secure Network Connection* for this entry or you can restrict the amount of data transferred (system to front end), for example, to allow for slow network connections (also see SAP Note [161053](#) – Use of SAP GUI in WAN).

You can also specify an SAProuter string for SAP GUI connections. An SAProuter is then assigned the task of transferring data for this connection. SAProuter strings can also be specified using aliases. These aliases are processed using the **SAPUILandscapeGlobal.xml** file.

**Note:**

You can also set up shortcuts on the *Create New SAP Shortcut* window, which take you directly to a preassigned transaction in the chosen system after you answer a password prompt.

**Hint:**

If you do not want to allow users to edit the system entries, you can use the SAP Logon Pad (program *saplgpad*). SAP Logon Pad looks very similar to SAP Logon. In SAP Logon Pad, however, you cannot make any entries such as, add, change, or delete connections. The corresponding buttons in the toolbar are not active.

Appendix: SAP GUI Connection String

A connection string describes a connection address for a destination, for example, an SAP system application server, rather like an Internet URL describes a location for a web page. The *SAP GUI connection string* describes a number of call parameters for the SAP GUI program.

In its simplest form, an SAP GUI call looks like this:

```
sapgui <host> <instance number>
```

When a logon group is used, the connection string has a somewhat more complex structure. It then may consist of the parts

- */M/<message server host>* (specifying the computer where the message server of the selected system is located)
- */S/<message server port>* (specifying the port of the message server)
- */G/<logon group>* (specifies the name of the logon group to select)

An SAP GUI call then looks like

```
sapgui /M/<message server host>/S/<message server port>/G/<logon group>
```

(for example, **sapgui /M/s4dhost/S/3610/G/SPACE**), which constitutes a complete, generic connect string.

**Note:**

You can use the command line switch **/WAN** to declare the connection to be a low-speed connection, with the corresponding reduction in network traffic (for example, **sapgui /M/s4dhost/S/3610/G/SPACE /WAN**).



Note:

When you call an SAP GUI using an SAProuter connection, you have to append the SAP GUI connection string list to the SAProuter connect string. The SAProuter connection string has a similar syntax, but is different in some details:

- `/H/` indicates the host name of the SAProuter. For technical reasons, the host name has to be at least 2 characters long.
- `/S/` specifies the service (port). This specification is optional, the default value is 3299.
- `/W/` indicates the password for the connection of the predecessor and the successor on the route and is also optional (the default value is "", or no password).

An example for an SAP router connection string is `/H/<saprouter>/W/<password>/H/<target_server>`.



LESSON SUMMARY

You should now be able to:

- Configure SAP Logon

Describing the Process of a System Logon (AS ABAP)

LESSON OVERVIEW

This lesson revises the process of a system logon using a custom application server, as well as dialog work process multiplexing.

Business Example

As an SAP system administrator in your organization, you want to understand the process of a system logon and how dialog requests of a user are processed in the system.



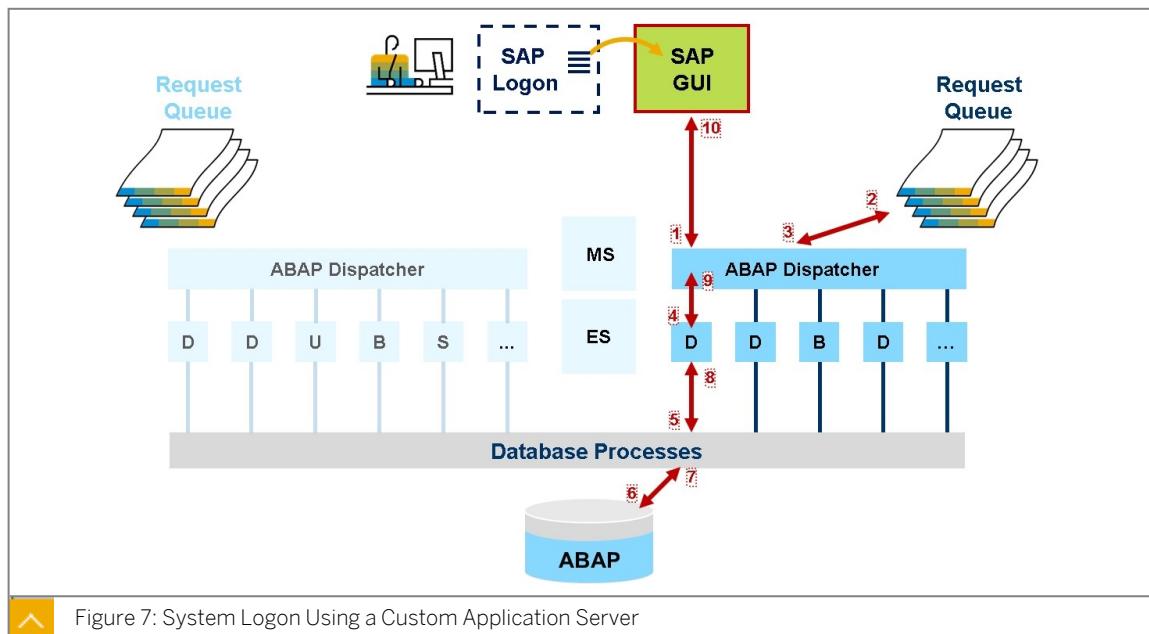
LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the process of an AS ABAP system logon using a custom application server
- Describe the concept of dialog work process multiplexing

Process of a System Logon Using a Custom Application Server

Connection between the front end of an end user and an application server of an SAP system is created by the SAP GUI (sapgui.exe program). To establish the connection between, SAP GUI requires information in the form of start parameters. This parameter string is normally created by the SAP Logon (saplogon.exe program) using information about the SAP system selected for logon.



As shown in figure *System Logon Using a Custom Application Server*, when selecting a direct logon to a customer application server, SAP Logon starts the SAP GUI with the required start parameters.

After the transfer of the logon screen from the dispatcher to the front end (not shown in the figure), the SAP GUI sends the user's logon data to the application server (step 1) where it will be stored in the request queue (step 2). The dispatcher determines a free work process to process the logon request, then it transfers the logon data to this work process (steps 3 and 4). The work process, in turn, checks whether the received combination of user ID and password is known to the system by making a request to the database (steps 5 - 8). A positive response from the database prompts the work process to return the initial screen of the SAP system to the front end (steps 9 and 10).



Note:

During a logon session (starting with the log on, ending with the log off), the assignment of the user to the application server is fixed.

Dialog Work Process Multiplexing

The processing of a transaction that consists of multiple screens can be executed using multiple, different dialog work processes. This distribution is called (dialog) **work process multiplexing**.

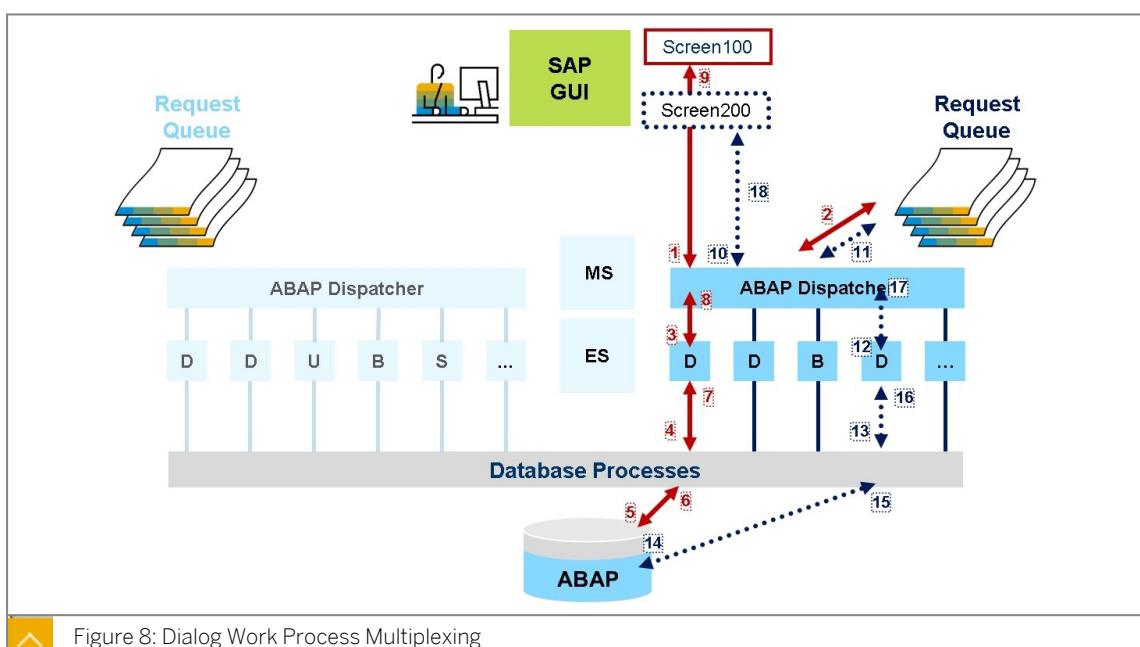


Figure 8: Dialog Work Process Multiplexing

Work process multiplexing means that an SAP system function, with content that is logically connected but consisting of multiple substeps, can be processed by different dialog work processes. The sum of these substeps are described as **transactions**. A transaction that consists of multiple screens, such as screens 100 and 200 (and therefore multiple steps), can be processed by multiple dialog work processes (see figure "Dialog Work Process Multiplexing").

**Note:**

The multiplexing procedure is used exclusively for dialog work processes. Other work process types (such as background work processes) process entire functions without using work process multiplexing.

In terms of an SAP transaction, dialog work processes may process only parts of transactions that are connected from a business point of view, for example when there is a change to database table content included, which in turn may be performed by an update work process.

**Note:**

For more information on the update process, see (for example) SAP training class *SAPTEC: Technology Fundamentals for SAP S/4HANA and SAP Business Suite*.

**LESSON SUMMARY**

You should now be able to:

- Explain the process of an AS ABAP system logon using a custom application server
- Describe the concept of dialog work process multiplexing

Setting Up Logon Groups in AS ABAP

LESSON OVERVIEW

This lesson introduces the concept of logon groups in AS ABAP.

Business Example

You want to use logon groups to restrict user logons to certain application servers of your system.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the benefits of logon groups in an AS ABAP system
- Create a logon group in an AS ABAP based SAP system

The Idea of Logon Groups

SAP systems sometimes have far more than just one or two instances. Each of these instances offers a specific number of work processes of various types and can access the resources of your hardware.

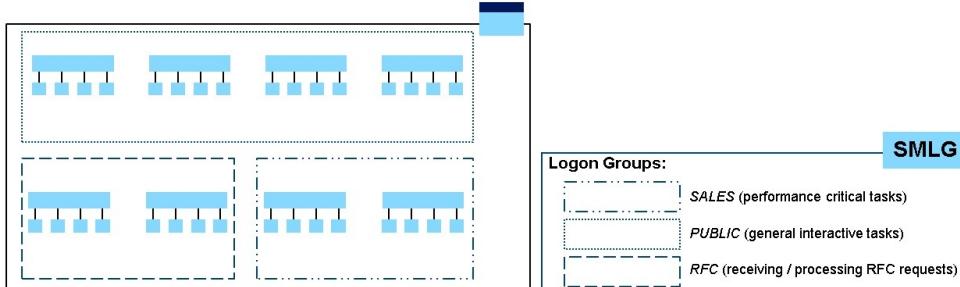
Situations may occur where the tasks to be performed on an instance place massive demands on the underlying hardware, thus slowing down all work on this instance. Long dialog response times are particularly annoying since the end users affected by this incur real costs due to poor system availability. The following figure list some examples:

- Asynchronous RFCs are used to parallelize applications. If the degree of parallel processing is not limited, there may be a spawning increase in the number of requests that may paralyze the application server instances on the receiver side.
- The developer can decide to parallelize complex background tasks (then they may use multiple dialog work processes in parallel).
- Member of the controlling team may run expensive controlling reports in dialog mode.

All these items can cause the members of the telephone sales team who run time-critical applications in dialog experience bad response times.



- Issue: Heavy load situations that can affect the end user, e.g. due to
 - high number of incoming RFC requests
 - complex tasks in background processing
 - expensive reports in dialog
- Solution: Separate performance critical applications from other work in the SAP system by creating logon groups
 - separate logon group for receiving and processing RFC requests
 - separate logon group for performance critical interactive tasks
- Example:



 From principal point of view, an application server (instance) can also be part of multiple logon groups

 Figure 9: Logon Groups – The Basic Idea

To separate dialog load from other work in the SAP system and improve dialog response times for end users, you can set up logon groups and adjust the number of work processes on the individual application server instances. A *logon group* is a group of application server instances that belongs to one SAP system. Logon groups are used to automatically distribute user logons to individual instances or to groups of SAP instances (Applications Server).

In the figure above, the following logon groups are used:

- a logon group *SALES* for the telephone sales team (used for time critical applications), the application server instances in this group may for example contain neither background work processes nor update work processes and – perhaps – particular fast processors.
- a logon group *RFC* to separate incoming RFC requests from interactive requests.



Note:

To use the concept of logon groups successfully, RFC destinations from external systems must reference the specially created logon group. Consider setting up the *RFC* logon group on all your AS ABAP based SAP systems and referencing this logon group for every RFC destination to an SAP system. For more information, see SAP Note [593058: New RFC load balancing procedure](#).

- a logon group *PUBLIC* for generic (that is: not time-critical) interactive tasks.

**Note:**

In addition, you could also consider a separate logon group for use with SAP Web Dispatcher (that is: for incoming HTTP(S) requests) that can be assigned to certain services in the *Internet Communication Framework* (transaction SICF) – see also SAP Note [751873](#) – *Problem analysis during the HTTP load distribution*.

For background tasks, special job server groups can be created in transaction SM61. Job server groups can be used similarly to logon groups for load distribution purposes in background processing.

The following figure shows the logon process using logon groups in more detail.



- Example: Logon using logon group *PUBLIC*, Message Server returns IP address X.Y.Z.20*

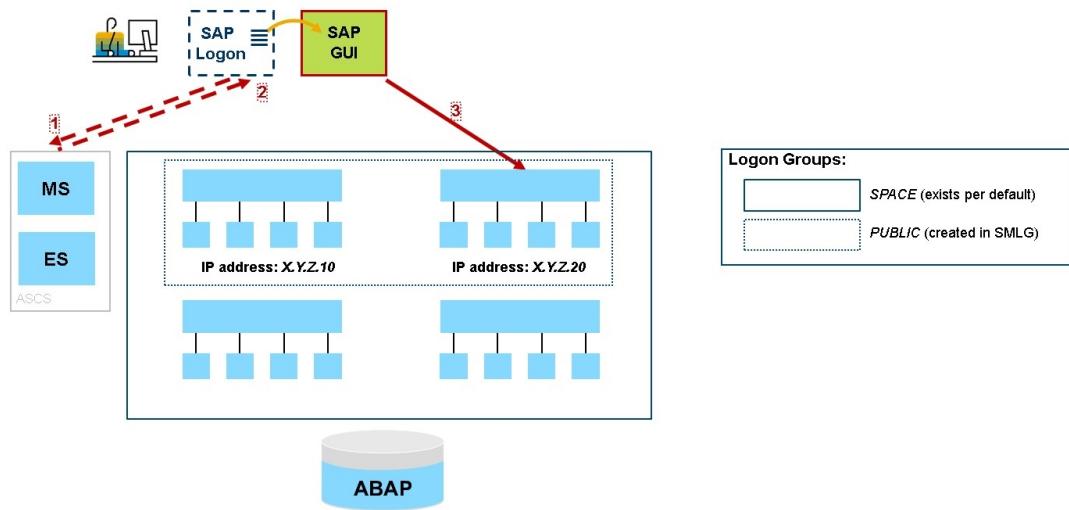


Figure 10: System Logon Using Logon Groups

As shown in the figure *System Logon Using Logon Groups*, by default, every application server of an SAP system is assigned to logon group *SPACE*. This name exists per default and contains all active servers (see SAP Note [113440](#): *Default logon group 'SPACE' for load balancing*).

Then there is a logon group *PUBLIC* for dialog requests which is intended to be used by end users so that they have comparable response times. When a user chooses the respective entry in SAP Logon that points to logon group *PUBLIC*, the Message Server is contacted first to determine the application server with the best performance in the logon group (step 1). The Message Server sends the information back to the SAP Logon (step 2) that then starts the SAP GUI with the required start parameters. As a result, the SAP GUI sends the user request directly to the dispatcher of this application server in logon group *PUBLIC* (step 3).

**Note:**

An application server can be included in several logon groups.

Setup of Logon Groups

You set up logon groups in transaction **SMLG** (*Maintain Logon Groups*), where you can create a new logon group and assign individual application server instances to it.



Note:

You can display the current status of the application server under *Goto → Load distribution* and you can also see the instance to which the next user will be distributed, depending on which logon group is being used.

When logging on using a logon group, the Message Server is always contacted first to identify the instance with the best performance within the selected logon group. The load information on each instance is determined by a report (**SAPMSSY6**) that runs automatically every five minutes. This report saves the information in a special memory area in the Message Server. This information can be then used by SAP GUI (or SAP Web Dispatcher in case of HTTP(S) request) to request the logon favorite, meaning an application server in the logon group with the best performance at the moment of the logon. To avoid the logon favorite becoming overloaded with new logons, whenever a large number of users attempt to log on, the load information is updated for each instance every five minutes and/or after every fifth logon.



Note:

For RFC destinations, load balancing is done for every new connection. RFC calls on an existing connection do not cause load balancing, you stay connected with the same server.

Additional Information

In addition to the information provided in the text, the following SAP Notes provide further information on specific topics:

- SAP Note [51789](#): *Poor user distribution in logon distribution*
- SAP Note [118093](#): *Concepts of defining 'limits' in logon load balancing*
- SAP Note [888279](#): *Regulating / distributing the workflow load*



LESSON SUMMARY

You should now be able to:

- Explain the benefits of logon groups in an AS ABAP system
- Create a logon group in an AS ABAP based SAP system

Appendix - Executing Basic Administration Transactions

LESSON OVERVIEW

In this lesson, you will obtain an overview of a number of frequently used analysis transactions for system administration.

Business Example

As a new member of the SAP Basis team in your company that runs SAP S/4HANA Server systems on premise or in SAP S/4HANA Cloud, private edition, you want to get an overview of some transactions that help you to perform your daily work as a system administrator.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- List basic administration transactions and explain their use

Basic Administration Transactions for AS ABAP

Introduction

The transactions shown in figure “Common Administration Transactions” help you to deal with your daily SAP system administration work. SAP system administrators should be familiar with the use and interpretation of these transactions.



SM51	Started Application Server Instances
SM21	System Log
SM50 / SM66	Work Processes
SM04 / AL08	Users Logged On
SU01 / SU10	User Maintenance
SM12	Enqueue Administration
SM13	Update Requests
SM36 / SM37	Background Processing
SMMS	Message Server Monitor
SM02	System Messages



Figure 11: Common Administration Transactions

Overview of Started Application Servers and Related Transactions

Transaction SM51 lets you display the application servers that are logged on to the Message Server. They represent the application servers (without the ASCS instance) of the SAP system. Their status indicates which application servers are active in the SAP system.

In the application server display, the following functions are available as menu items; important menu items also appear as buttons:

- Display and manage work processes (transaction SM50): Transaction SM50 shows a snapshot of the work process status of the application server you are logged on to. You need to refresh the display to receive updated information. The process overview is intended primarily for information gathering. For example, you can monitor processes to determine whether the number of work processes in your system is adequate, to assess whether the application server is working at full capacity, to gather information for troubleshooting, or for tuning.



Hint:

As of AS ABAP 7.40, you can display the **system-wide** list of all work processes in SM50. To do so, navigate to *Goto → System-Wide List*. This corresponds to transaction SM66.



Note:

A new feature as of AS ABAP 7.40 is the priority of queries:

- *High*: Priority for online sessions and internal system processes
- *Normal*: Priority for RFC calls from within online sessions
- *Low*: Priority for background processing (batch) and RFC calls of background programs (batch jobs)

- Display and manage user sessions (transaction SM04): Transaction SM04 enables you to display all the users who are logged on to the application server you are currently logged on to.



Hint:

As of AS ABAP 7.40, you can display **all users currently logged on to the SAP system** also from transaction SM04. To do so, navigate to *Goto → System-Wide List*. This corresponds to transaction AL08.

- *System log* (transaction SM21): An SAP system logs system messages, system warnings, and system error messages in system logs. Each instance has a local log in which system messages, warnings, and error messages in this instance are logged.

In transaction SM21 you can read logs created locally in all the instances of the SAP system.

You can also access a variety of other transactions in transaction SM51 (which are not listed in the figure above). For example, by choosing the menu path *Goto → Monitors*, you can access:

- The *Gateway Monitor* (transaction **SMGW**) for RFC connections in the RFC Gateway.
- The *ICM Monitor* (transaction **SMICM**) to monitor and administrate the Internet Communication Manager, which sends and receives HTTP(S) requests to and from the Internet.
- The OS monitor (**ST06**) which – among others – displays CPU utilization, memory utilization and disk response time.

**Note:**

For a list containing all patch information for the most important kernel components, run report **RSMONREL_ALV**.

User Maintenance

The *User Maintenance* transaction **SU01** enables you to maintain individual master records. The user administrator can use it to create new user master records or make changes to existing user master records, such as assigning new roles and authorization profiles.

**Note:**

For the mass maintenance of users, you can use transaction **SU10** (*User Maintenance: Mass Changes*).

Transactions for the Monitoring of Specific Work Process Types

You can manage lock entries in the lock table of the enqueue server using transaction **SM12**. Lock management is used to monitor the lock logic in the SAP system. It lets you determine which locks are currently set.

With transaction **SM13**, you manage the following update tasks:

- Display update requests.
- Analyze problems relating to the update.
- Test and clean up canceled update requests.
- Display and reset the status of update requests.
- Delete update requests.
- Display statistics for the update task.

This function provides you with an overview of the update requests and a tool to investigate problems.

**Note:**

Transaction **SM13** does not display requests that have been updated successfully.

To create background jobs in the SAP system, you can use transaction **SM36** (*Define Background Job*). To display an overview of the background jobs in the SAP system, use transaction **SM37** (*Job Selection*).

Monitoring the ABAP Message Server

A single message server runs in each SAP system. This message server performs the following tasks in the SAP system:

- It is the central communication channel between the individual application servers of the SAP system. This enables certain work process types to be used across all application servers.
- It performs load balancing of logons using SAP GUI and RFC with logon groups.
- It is the information point for the SAP Web Dispatcher and the application servers. Each application server of the SAP system logs on to the message server first.

When an application server is started, the dispatcher process contacts the message server so that it can announce the services it provides (DIA, BTC, SPO, UPD, and so on). If the connection setup to the message server fails, an entry is made in the system log (syslog), which you can analyze with transaction SM21 (see above).

If the message server stops working, it must be restarted as quickly as possible to ensure that the SAP system continues to operate smoothly.

Sometimes it is important to check the (ABAP) message server of an SAP system more precisely. There are several options for doing so. One is transaction SMMS (Message Server Monitor) to monitor the message server. The initial screen of this transaction displays the status of all active application servers, similar to transaction SM51. In the Message Server Monitor, you can check and change all the important settings, generate and view trace files, read statistics, and so on.

The following functions are available in transaction SMMS as menu items; important menu items also display as buttons (to access these functions, you may need to select a line with an application server instance first):

- Goto → Logon data → Display displays information about the available communication logs and ports, including dialog, RFC, HTTP, SMTP, and more.
- Goto → Hardware Keys shows the hardware key (also called the customer key) of the message server hardware.



Note:

For more information on the hardware key, see SAP Note [174911 – Determining the hardware key \(customer key\)](#).

- Goto → Parameters → Display Provides comprehensive information about the message server.
- Goto → Trace → File → Display displays the trace file (dev_ms)

In addition, a number of programs are provided and available at the operating system level. You can normally find the test programs in the executable (kernel) directory of your SAP instance.

- The msmon monitoring program (command to call, for example, `msmon name=<SID>`) provides the same functions as transaction SMMS in the SAP system.

- You can use the *lgtst* test program (command to call, for example, **lgtst -H <host> -S sapms<SID>**) to check the connection to the message server and to display the active application servers and logon groups that the message server can currently see.
- You can use the *msprot* program to monitor the message server. The program continuously issues the status of the application servers logged on the message server, and is stopped if the message server is stopped. You are notified of the termination of the message server and can respond to it. For a detailed description, see SAP Note [636938 – Description of the test program msprot](#).

For security settings see SAP Note [821875](#) – Security settings in the message server. For trace options, see SAP Note [396750](#) – Enhanced trace options for Message Server.

Sending System Messages

You can send a system message to all users in your SAP system with transaction **SM02** (System Messages). You can also restrict the recipients to users of a certain client or users that are logged on to a specific application server. A message is only displayed to the recipient once a day. The SAP system displays messages in the following cases:

- When a user logs on to the SAP system.
- If the user is already logged on: as soon as a logged on user performs the next dialog step.

In transaction **SM02**, non-expired and expired system messages are displayed in separate views. In the menu, you can toggle between these views by choosing *Goto → Current Messages* and / or *Goto → Archived messages*.



Hint:

To display a system message on the *Fiori Launchpad* page, you can implement a plug-in or an extend shell header. For more information, see SAP Knowledge Base Article [2475155](#) - Flash message into the SAP Fiori Launchpad.



LESSON SUMMARY

You should now be able to:

- List basic administration transactions and explain their use

Learning Assessment

1. Which AS ABAP process receives user requests from an SAP GUI for Windows session?

Choose the correct answer.

- A Work process
- B ABAP dispatcher
- C Server process
- D SAP GUI

2. Which work process types are there in an AS ABAP system?

Choose the correct answers.

- A Dialog work process
- B Message server
- C Update work process
- D Internet Communication Manager (ICM)
- E Background work process
- F Spool work process

3. Which services are offered by an ABAP Central Services Instance?

Choose the correct answers.

- A Enqueue Server
- B Internet Communication Manager
- C Update Service
- D Message Server

4. Work process multiplexing means that a specific SAP GUI session can use different work processes of one instance while the user goes through several steps of the same transaction in the SAP system.

Determine whether this statement is true or false.

True

False

5. Every AS ABAP-based SAP system offers at least one logon group, named *SPACE*.

Determine whether this statement is true or false.

True

False

6. In which transaction can you get an overview of the application server instances of your SAP system?

Choose the correct answer.

A SM50

B SM51

C SM21

D SM04

Learning Assessment - Answers

1. Which AS ABAP process receives user requests from an SAP GUI for Windows session?

Choose the correct answer.

- A Work process
- B ABAP dispatcher
- C Server process
- D SAP GUI

You are correct! The ABAP dispatcher receives the user request (from a user working in SAP GUI for Windows or SAP GUI for Java) on the AS ABAP and passes it on to an available work process. The SAP presentation program, SAP GUI, is not part of the application server (software-oriented view), and the server process is a process on AS Java.

2. Which work process types are there in an AS ABAP system?

Choose the correct answers.

- A Dialog work process
- B Message server
- C Update work process
- D Internet Communication Manager (ICM)
- E Background work process
- F Spool work process

You are correct! All of the above processes can in theory be configured on an AS ABAP. However, not all of the above processes are work processes. The message server and ICM process are not work processes.

3. Which services are offered by an ABAP Central Services Instance?

Choose the correct answers.

- A Enqueue Server
- B Internet Communication Manager
- C Update Service
- D Message Server

You are correct! The Enqueue Server and the Message Server are always offered by the ABAP Central Services Instance of an AS ABAP-based SAP system. The Internet Communication Manager and Update services are offered by AS ABAP instances that have an ABAP dispatcher process – unlike the ABAP Central Services Instance.

4. Work process multiplexing means that a specific SAP GUI session can use different work processes of one instance while the user goes through several steps of the same transaction in the SAP system.

Determine whether this statement is true or false.

- True
- False

You are correct! Work process multiplexing means that a specific SAP GUI session can use different work processes of one instance while the user goes through several steps of the same transaction in the SAP system.

5. Every AS ABAP-based SAP system offers at least one logon group, named *SPACE*.

Determine whether this statement is true or false.

- True
- False

You are correct! Every AS ABAP-based SAP system offers at least the logon group *SPACE*.

6. In which transaction can you get an overview of the application server instances of your SAP system?

Choose the correct answer.

- A SM50
- B SM51
- C SM21
- D SM04

You are correct! Transaction SM51 shows an overview of the application server instances of your SAP system. Transaction SM50 lists the work processes of an instance/your system. Transaction SM21 lets you display the system log and Transaction SM04 displays a list of users of an instance/your system.

Lesson 1

Understanding the Concept of Start and Stop Procedures

39

Lesson 2

Introducing Different Tools for Start and Stop Procedures

53

Lesson 3

Starting and Stopping AS ABAP

63

Lesson 4

Log and Trace Information for System Start and Stop

77

UNIT OBJECTIVES

- Explain the file system structure of AS ABAP
- Explain the general mechanism of starting and stopping AS ABAP
- List organizational considerations before stopping AS ABAP
- Start the SAP Management Console (SAP MC) using a configuration file
- Use SAPControl on operating system level
- Start the SAP Microsoft Management Console (SAP MMC)
- Start and stop using SAP MC
- Start and stop using SAPControl
- Start and stop using SAP MMC
- List log and trace information for troubleshooting
- Use SAP MC and SAPControl to access log and trace information
- Appendix: further analysis options

Understanding the Concept of Start and Stop Procedures

LESSON OVERVIEW

This lesson provides a general overview of how to start and stop SAP systems. It will discuss the areas of the file system that contain the important data for starting instances.

Business Example

As the administrator of SAP systems, you need to stop the systems for maintenance purposes or after changing system parameters, and then restart them.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the file system structure of AS ABAP
- Explain the general mechanism of starting and stopping AS ABAP
- List organizational considerations before stopping AS ABAP

File System Structure

Each instance of an SAP system requires data in the file system. This includes data that all instances can access (globally) and data that can only be accessed by an instance individually. In the file system, the data for the instances is contained in the *usr/sap* directory, where it is further organized in subdirectories. This structure is shown in the following figure *File System Structure of an AS ABAP based SAP System (generic view)*.



Hint:

The business data and other data is stored in the database, which has its own structure in the file system, depending on the database manufacturer, but is not covered by the *usr/sap* directory.

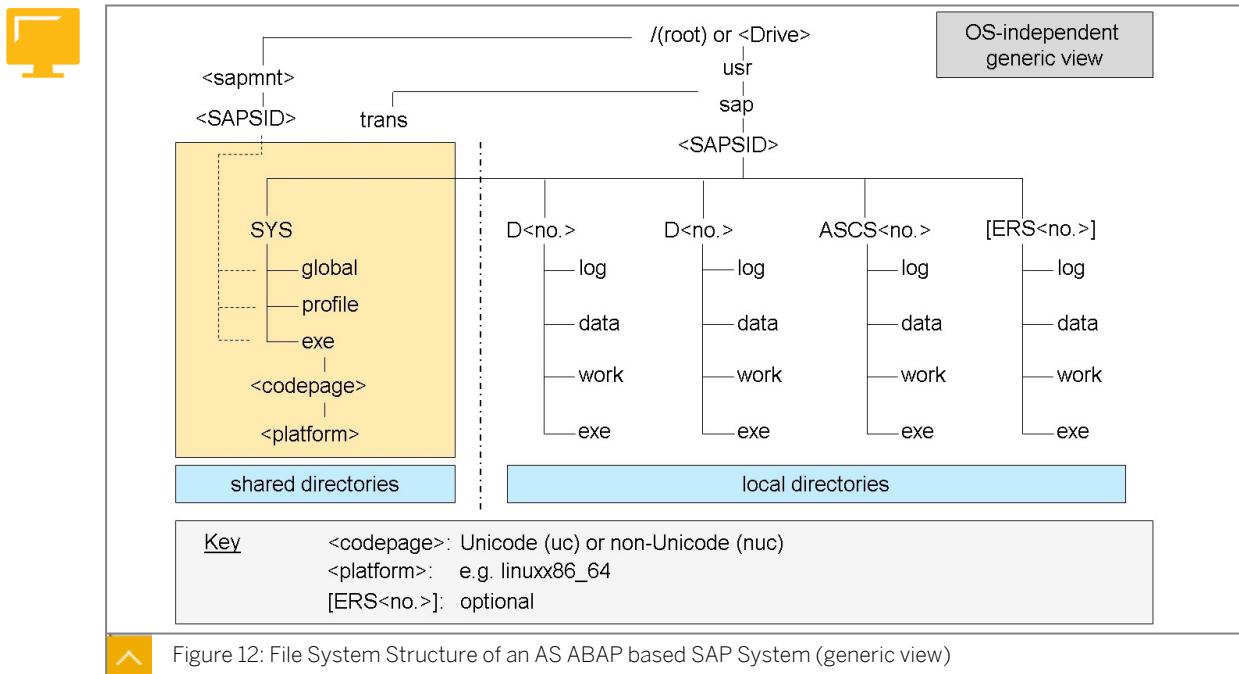


Figure 12: File System Structure of an AS ABAP based SAP System (generic view)

**Caution:**

The OS-specific characteristics are not covered in the figure above. For example there is no separation of physically and logically shared directories. However, an abstraction of the underlying operating system makes only sense to a specific extent. That is the reason, why some OS-specifics are covered below.

The <SAPSID> directory contains all data that must be accessed only by this SAP system.

As shown in the figure above, the SYS directory contains information that can be accessed by all instances:

UNIX/Linux operating system

SYS directory is located in its own file system with the name *sapmnt*. The SYS directory is then created in the file system of each instance and contains links to the subdirectories of the *sapmnt* file system.

Windows operating system

SYS is usually located in the file system in which the PAS instance is located. A share with the name *sapmnt* is created based on the sap directory above the <SAPSID>/SYS directory. This share allows all instances to access the SYS directory.

The SYS directory contains the following subdirectories:

- **global** contains information, which is relevant for the entire SAP system, but stored centrally (for example, security data like the SSFS-key or snapshots created by the SAP system).
- **profile** contains profiles used to start the SAP system and their instances (for example, the *default profile* and the *instance profiles*).
- **exe** contains the global kernel directory (including programs and libraries), which serves as a source for the installed instances.

**Note:**

The global exe directory contains the subdirectory <codepage>, which itself includes a folder called <platform>. The only supported <codepage> is in the meantime uc (*Unicode*), but in exceptional cases still nuc (*non-Unicode*) for systems based on lower releases.

**Hint:**

SAP HANA based systems only support *Unicode* as codepage.

Each instance has its own directory tree and the name contains the 2-digit instance number. If several instances are installed on the same host, each individual instance gets a unique instance number. The instance-specific content (PAS, AAS, ASCS) is located in the respective D<no> directories (where <no> is the instance number). The instance directories include individual work sub-folders, which themselves serve as storage location for - among several other files - log and trace files written during start and stop operations. A further important instance-specific sub-folder is the exe directory where the (replicated) kernel is stored locally. These are essential programs that are started instance-specific when the SAP system is started.

Start Process: SAP System

As an SAP system administrator, you should be familiar with the process of starting the ABAP based SAP system. It is a fundamental administrative task, which requires knowledge of the SAP system architecture including some specific release-dependent changes. Furthermore the start procedure is the basic prerequisite for any ABAP user to be able to log-on and work within the SAP system. In addition, restarts of SAP systems are required for a wide range of different maintenance tasks, whether hardware enhancements, software changes or configuration changes to the SAP system's profiles.

This lesson describes the process flow when starting an SAP system. Every SAP system uses a database and consists of instances. An SAP system based on AS ABAP usually includes an ABAP central services instance (ASCS) that offers the ABAP Enqueue Server and the ABAP Message Server, as well as a Primary Application Server (PAS) instance with an ABAP Dispatcher and ABAP Work Processes. In this case, the PAS instance does not have an ABAP Enqueue Work Process or an ABAP Message Server.

Start Process: Database

The database is the fundamental element of the entire SAP system. Working within an ABAP based SAP system always requires an up and running database.

For an SAP S/4HANA Server system, or generally speaking SAP systems running on an SAP HANA database, the database typically is started separately (manually or scripted) before the SAP system starts, regardless if it is installed on the same host or not.



Note:

For most non SAP HANA based databases, the following mechanism is used:

- If the database has not yet been started, this will be done automatically, shortly before starting the PAS and AAS instances (but only if the database is installed on the same host as the PAS or AAS instances).
- If the database is installed on separate host/s, it has to be started separately before starting the SAP system.

PAS (Primary Application Server) and AAS (Additional Application Server) instances require an already started database, to which their *ABAP work processes* are able to connect when the instance is started. Thus, the database must be in an operational state before starting this type of instance (AS-based). This also applies to the formerly known “central instance” and “dialog instances” in earlier technical releases.

Within this class there is no further discussion of the dedicated database tools to start (or stop) the corresponding database. Nevertheless, databases supported by SAP can also be controlled by SAP tools like the *SAP Management Console* (SAP MC) via the *SAP Host Agent*, *SAPControl* or by utilities provided by the database vendor.



Hint:

The start sequence is explained in more details later on in this lesson.

The requirement to ensure that the database is in operational state when starting the PAS and/or AAS instances will be covered in a following lesson within this Unit. This is of particular importance, if the database is installed on an explicit host.

Start Process: Priorities

To make the overall start procedure of an SAP system easier and provide a better transparency, SAP implemented the use of *Priorities*. These are assigned numbers using a parameter during installation to different types of instances.

Core principle of *priorities*:

The lower the number, the higher the *priority*. As a consequence, instances with low *priority* numbers are started earlier than instances with higher *priority* numbers.

The use of *priorities* allows you to start (and stop) complete SAP systems as a whole unit without the need to manually take care of a special start (or stop) sequence. The system automatically ensures the correct starting (or stopping) order of the instances.

Priority Definitions



- Priority 1: Instances with an *Enqueue* but without *ABAP Dispatcher* (ASCS)
- Priority 2: Instances with an *Enqueue* and with *ABAP Dispatcher*
- Priority 3: Instances without an *Enqueue* (PAS/AAS)

**Note:**

There is no co-existence of an instance with *priority 1* and another instance with *priority 2* in ABAP environments. It is an “or” relationship based on the technical release.

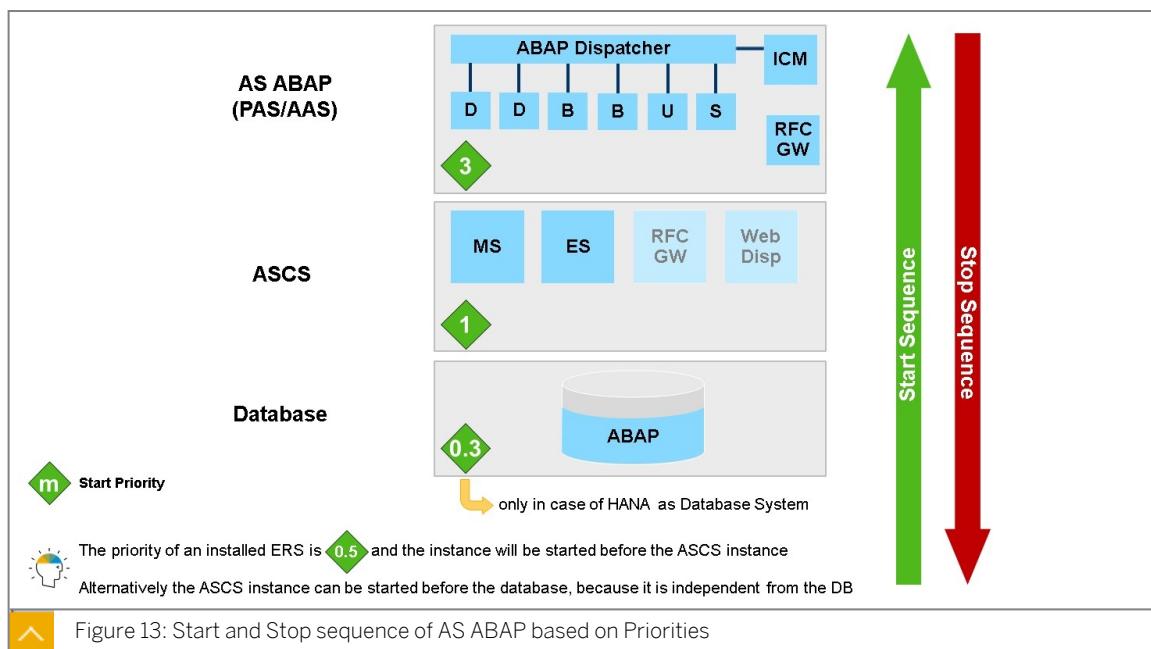
**Hint:**

SAP provides a further special instance called *Enqueue Replication Server* (ERS) to achieve high availability on software level for the enqueue server, which is a single point of failure in any SAP system. This instance gets assigned ***priority 0.5***.

**Hint:**

Normally only SAP instances get assigned a specific priority. Nevertheless, the SAP HANA database system also gets a start-up priority value during installation, which is ***0.3***. In this way all SAP HANA based SAP systems, which only run on SAP HANA, can be started using the *priorities*' mechanism.

The list of *priority* definitions indicates that the *enqueue server* (or process) plays a special role for the start (and stop) sequence. Typically the *enqueue (server)* is part of the ASCS instance, which runs centrally once per ABAP stack. The instance hosting the *enqueue* also includes the *Message Server* and must be started before all other instances (in earlier releases, the instance covering these services was called *central instance*). In comparison to the former *central instance*, the ASCS does not require a running database, because it does **not** provide work processes, which require a connection to the database.



In the figure above the mandatory instances and their priority is shown, which is important for the identification of the start sequence of an ABAP based SAP system.

The start process is performed in multiple steps using the operating system user **<sid>adm** (or **SAPService<SID>** on Windows):

- If the SAP system includes an ASCS instance, it will be started first (*priority 1*). An exception would be the usage of an *Enqueue Replication Server* (ERS) instance. If installed and configured, the ERS instance gets started before the ASCS instance, simply caused by the higher priority (0.5) and then followed by the ASCS instance.
- Next, the PAS instance is started together with all *Additional Application Server* (AAS) instances. The PAS instance and the AAS instances require a running database.

**Note:**

The figure above does not provide information on the distribution of the different core elements of an SAP system to hosts. Nevertheless, for a successful start procedure it is not enough to rely on the assigned *priorities*, but guarantee that even for distributed SAP environments all elements can be accessed by the tool in use. This will be covered with some further details in another lesson.

Start Process: sapstartsrv as Instance-specific Service

Processes of an SAP instance are started by a program called *sapstartsrv* that runs once per instance. It is part of the *SAP kernel* delivered with the installation and provides a wide range of functions to handle different types of SAP instances. In addition, the *sapstartsrv daemon* (UNIX/Linux) or service (Windows) gather important information needed by the administrative personnel.



Feature	Tool	Functions
Starting and stopping instances	<ul style="list-style-type: none"> • SAP MC/MMC • command line 	<ul style="list-style-type: none"> • Provide basic management services for instances + processes • Provide runtime information by SOAP Web service SAPControl with a broad spectrum of functions • Starting and Stopping complete <SID>, as well as individual instances and/or processes
Monitoring runtime state	<ul style="list-style-type: none"> • SAP MC/MMC • command line 	<ul style="list-style-type: none"> • Provide shared memory segments for runtime information for a wide range of KPIs, like CPU load, Memory consumption, Disk usage, various response times etc.
Reading log, trace, and configuration files	<ul style="list-style-type: none"> • SAP MC/MMC • command line 	<ul style="list-style-type: none"> • Analyze log and trace files • Automatic registration in LDAP and/or SAP SLD
Providing technical information	<ul style="list-style-type: none"> • SAP MC/MMC • command line 	<ul style="list-style-type: none"> • Provide ports for HTTP/HTTPS (5<xx>13/5<xx>14) • Provide active sessions, thread lists etc. • Retrieve network access points
Further features	<ul style="list-style-type: none"> • SAP MC/MMC • command line 	<ul style="list-style-type: none"> • Client support based on SOAP Web service (e.g. ABAP, Java, .NET, gSOAP etc.)

 SAP MC = SAP Management Console (see next lesson)
 SAP MMC = SAP Microsoft Management Console (see next lesson)

Figure 14: SAP Start Service – Overview of features and functions

The implementation of the program *sapstartsrv* is OS-dependent. On *UNIX/Linux* operating system platforms a *daemon* is used, whereas on *Windows* a *service* is implemented. By default a setting made during installation on all supported platforms ensures that *sapstartsrv* is automatically started when the host is started up or rebooted.

On *UNIX/Linux* OS-platforms the start script *sapinit* is installed and in turn starts *sapstartsrv*. This can be seen on the following figure:

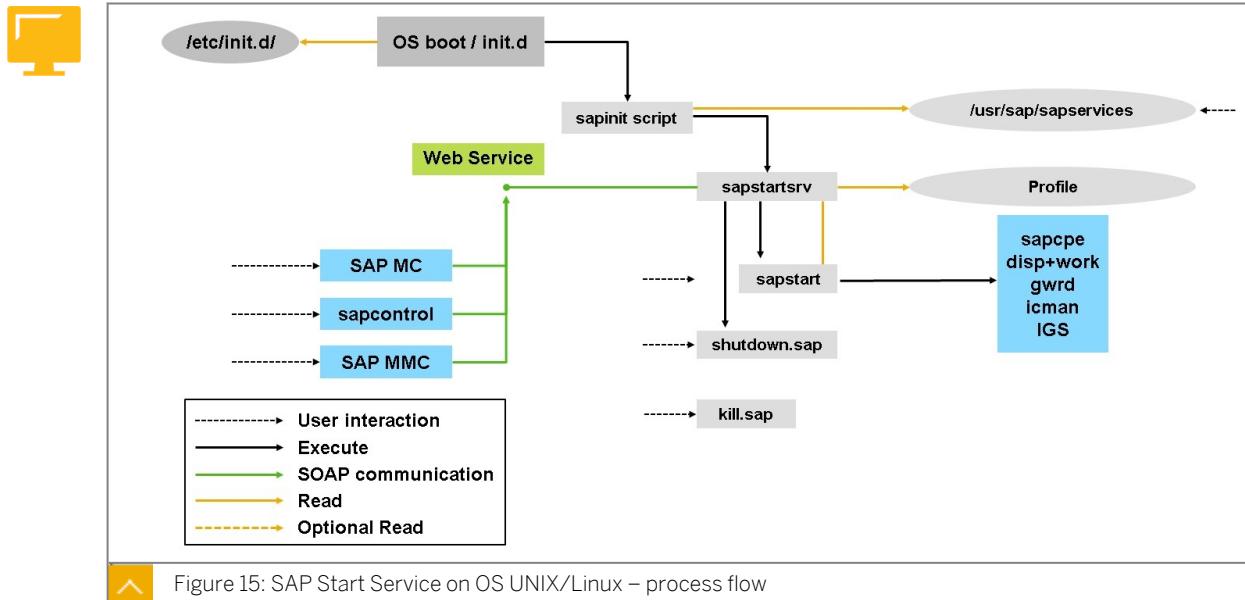


Figure 15: SAP Start Service on OS UNIX/Linux – process flow

On Windows platforms a service called SAPService<SAPSID>_<instance_number> is created under a special user SAPService<SAPSID> and starts the executable sapstartsrv. Get details from the following figure:

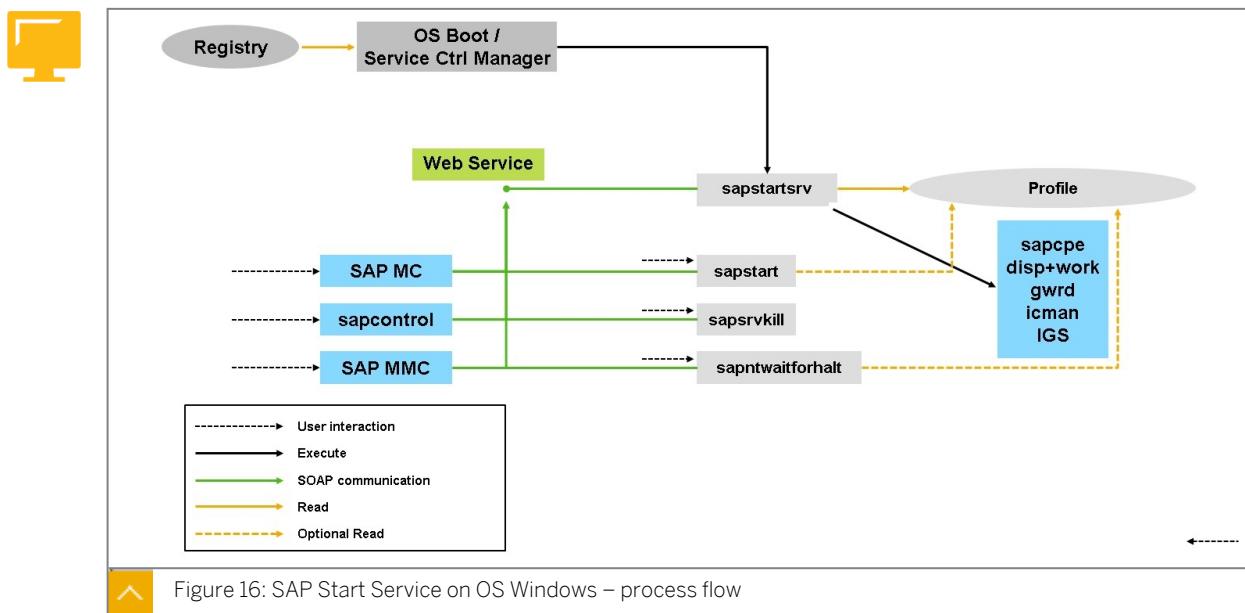


Figure 16: SAP Start Service on OS Windows – process flow

**Hint:**

Remember that the goal of the OS-specific views is not completeness, but rather to make clear there are differences.

In spite of the different OS implementation, the program sapstartsrv provides the same functions when using (for example) the SOAP interface. Details are discussed in a further lesson.

Start Process: Evaluation of Profiles

SAP systems use profiles for configuration purposes. These profiles are provided by the installation tool and, if required, can be adjusted according to the needs of the customer.

According to the architecture of an ABAP-based SAP system, every installation (<SAPSID>) gets exactly one *Default profile* for system-wide parametrization. In addition, every SAP system consists of instances and uses *Instance profiles*, which are used for individual configuration of the instance-specific processes and services. In this way the settings provided in the *Default profile* can be replaced per instance with individual values or even simply enhanced with new parameters and values.

Thereby every *Instance profile* includes different types of parameters:

- Parameters to start programs in order to initialize the instance (for example *sapcpe*, database, *ABAP Dispatcher*, *IGS*)
- Parameters to configure the instance according to requirements and resources (for example type and amount of *work processes*, security, main memory, and so on.)



Note:

The above mentioned parameters are not visible in blocks 1:1 within the *Instance profile*. This is only to remind the administrators that such a profile consists of different types of parameters. Parameters to start programs use prefixes like "Start_Program_<number>" or "Execute_<number>".



Hint:

Here, for the topic of starting SAP systems, only some parameters to start programs are relevant and described without the corresponding syntax.



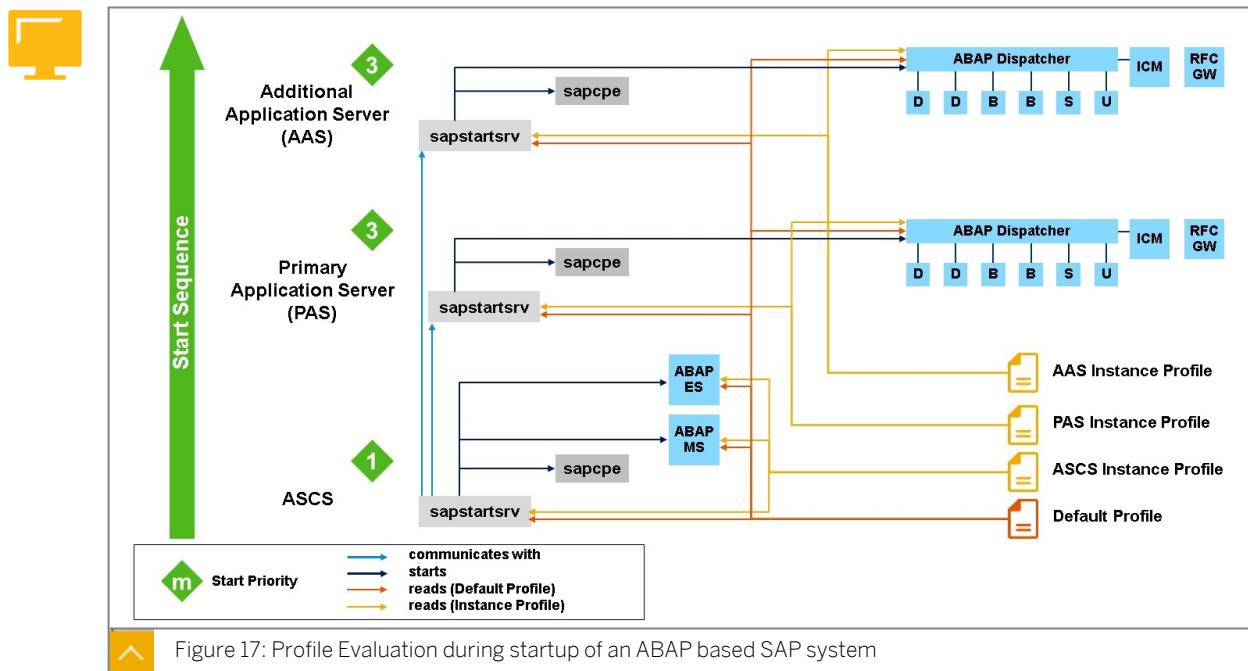
Hint:

Nevertheless, be aware of the following:

An incorrect parameter or parameter value can also be the root cause for an unsuccessful start of an SAP system or an individual SAP instance failure.

For any further instance, an additional *Instance profile* will be used and stored within the central *Profile* directory: under /sapmnt/<SAPSID>/SYS/profile.

- *UNIX/Linux*: /usr/sap/<SAPSID>/SYS/profile
- *Windows*: \\<SAPGLOBALHOST>\saploc\<SAPSID>\SYS\profile



How does an SAP start procedure for a complete SAP system work?

Assuming that all involved `sapstartsrv` daemons on *UNIX/Linux* or services on *Windows* are up-and-running, the `sapstartsrv` with the highest *priority* takes over control (this is typically the `sapstartsrv` of the *priority 1* ASCS instance - see figure above).

Next, the content of the profiles is read by `sapstartsrv`. First the global parameters from the *Default profile* are taken into account, followed by the initialization of the instance. In this phase the start section from the respective *Instance profile* is evaluated, that is, the *instance profile* defines the steps to be performed during the start process and the processes of the instance to be started by `sapstartsrv`.

The start section of the ASCS *Instance profile* is read:

- Typically the start section begins with the execution of `sapcpe`, which replicates, copies, or links directories and programs (details are presented below in this lesson).
- Because the ASCS instance does not need an up-and-running database, it continues the scan of the start section and will start the *Message Server* and the *Enqueue Server* (optionally an *RFC Gateway Service* and/or an integrated *SAP Web Dispatcher* is/are started).

The `sapstartsrv` of the instance with the highest *priority* has done its work and gives over control to the next *priority LEVEL*. In the figure above these are the PAS and AAS instances. Both have the same assigned *priority LEVEL 3*.

Every individual `sapstartsrv` scans the common global *Default profile* and the specific *Instance profile* in order to initialize the instance according to the respective start section in the *Instance profile*. This is done simultaneously, because both instances have the same assigned *priority LEVEL 3*:

- `sapcpe` will be executed (see above)
- The database state will be checked and if it is not yet up-and-running, a script or command is executed to start the database (this is only valid, if the database is installed on the same host as the PAS or AAS instances - see discussion at the beginning of this lesson).



Hint:

In our training environment the available SAP S/4HANA Server installations use explicit hosts for the SAP HANA databases. Therefore we should not expect a starting action for the database triggered from the *Instance profile*.



Note:

Every Application Server instance (PAS/AAS) requires a database in operational state when it is started. This is caused by the follow-up procedure of the *ABAP Dispatcher*. When started itself, it initializes in turn the configured types and amounts of *work processes*, which immediately try to connect to the database.

- The *ABAP Dispatcher* will be started and performs follow-up actions like starting the *work processes*.
- The *Internet Graphics Service* (IGS) watchdog will be started. It can be used to generate graphical or non-graphical output for data from an SAP system or another source.



Caution:

Normally, you should not change the entries for the start process in the instance profile. However, if these start entries are changed, the `sapstartsrv` has to be restarted so that the changes made will take effect for the next system restart.

Start Process: `sapcpe` for Kernel Replication

The program `sapcpe` (central patch environment) is used to ensure the consistency of kernel binaries and libraries across the entire SAP system by implementing a verification and copy procedure during the start process of the ABAP-based SAP system.

The mechanism is pre-configured by the installation routine and initialized before startup of an SAP system in order to update or exchange content of the instance-specific SAP *kernel* directories. This is done by replicating the content of the global kernel directory `/exe/<codepage>/<platform>` to the local `exe` directory underneath the respective instance-specific directory `<instance_name><no>` with the help of program `sapcpe`.

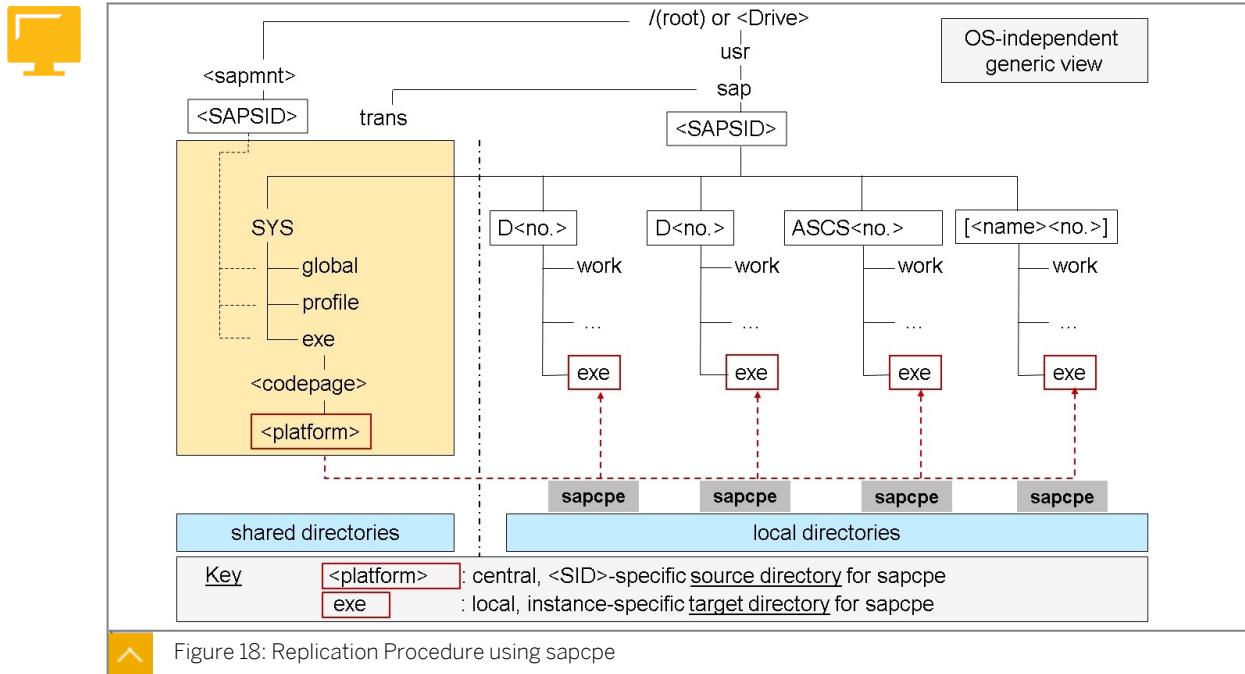


Figure 18: Replication Procedure using sapcpe

In this way every instance will make use of the same program versions after a successful start of an SAP system.

The program sapcpe can be used in the context of different strategic approaches, depending on the source and target content to be verified. It is also able to work with compressed programs or executables and make use of a configuration file (sapcpeft), which includes direct instructions on how to proceed and what should be done.

Finally there are also parameters for a *command line call* to determine source and target directories, as well as files and directories to be copied. For more information about the program sapcpe use the following link: https://help.sap.com/docs/ABAP_PLATFORM/85f9aa682547440b80d72492fea86a42/48f5f300af4c3a63e10000000a421138.html?q=sapcpe

Stop Process of the SAP System

The stopping of an SAP system is performed in the reverse order of the starting process according to the instance *priority*. Because there is a wide range of different technical ABAP-based releases used in the market, two main scenarios have to be differentiated:

SAP ABAP system with an ASCS instance

This setup reflects a typical up-to-date ABAP-based SAP system including an ASCS instance. All Application Server (AS) instances (PAS and AAS) are stopped first. Next the ASCS instance will be shut down (and finally the ERS instance, if installed and configured).

SAP ABAP system without an ASCS instance

ABAP-based SAP system installations of earlier technical releases were installed without an ASCS instance and might still not include it. As a consequence the central services (message service and enqueue process) are **not** part of an explicit ABAP Central Services instance, but provided by the first installed Application Server (AS) instance, the “central instance”, which has *priority LEVEL 2*. In case there is a scaling of AS instances, any further “dialog instance” gets *priority LEVEL 3*. Accordingly the stop procedure first takes into account the “dialog instance/s” (*priority LEVEL 3*) and then the central instance (*priority LEVEL 2*).



Note:
Conclusion:

- The stop procedure is dependent on the architecture and the technical release
- SAP instances with an *enqueue server* or *enqueue process* are always stopped last
- If the database is stopped, it will be typically shut down after the SAP instances

Tasks Before Stopping an ABAP-based SAP System

You may need to stop an SAP system for a variety of reasons, such as to restart after permanently changing profile parameters, before installing kernel patches, or to allow hardware upgrades to be carried out.



Check the status ...

- of active user sessions: **SM04 → System-Wide List (or AL08)**
- of background processing: **SM37 / SM37DISP**
 - are jobs active or scheduled?
 - are jobs triggered by external systems?
- of the update processing: **SM13**
- of external interfaces



Send a system message via SM02



Ensure that no database backup is running



Figure 19: Before stopping an AS ABAP

Before you stop the system, you should check the system status. This includes checks such as:

- Active users:

Call the *User Sessions for AS Instance <hostname>_<SAPSID>_<instance_number>* overview via transaction **SM04** and go to *System-Wide List* to determine, which users sessions are active in the system.

- Background processing:

Check which jobs are active with the job overview (transaction **SM37** or **SM37DISP**). If jobs are terminated by restarting the system, you must re-schedule them. Jobs that are scheduled to run during the time when the system is stopped, run automatically when the system is restarted.

- Batch Input:

The *Batch Input: Session Overview* (transaction SM35) displays running batch input jobs (covered in figure above as background processing).

- Update:

Use the *Update Requests: initial Screen* (transaction SM13), ensure the filters are fitting and choose *Execute* to check, whether update processes are terminated by the system stop. These update records are rolled back during the system stop process and are set to status *init* and re-processed again during the restart.

Before you stop your system, you should inform users early enough (if possible) about the upcoming downtime with a *system message* (transaction SM02).



Note:

Normally a system or an instance cannot simply be stopped, there are still organizational framework conditions to consider. For example, the system downtime must be planned with the respective departments and the downtime must be logged. You can use *SAP Solution Manager* to plan the system downtime. The *SAP Solution Manager Launchpad* contains an *SAP Fiori Group* called *Technical Administration* that includes *SAP Fiori App* "IT Calendar and Work Modes". There you can plan IT and business events for technical systems, instances, databases, hosts and external services, and schedule work modes, such as maintenance or planned downtime.



Hint:

To display a *system message* on the *SAP Fiori Launchpad* page, you can implement a plug-in or extend the shell header. For more information, see SAP Knowledge Base Article [2475155](#) - *Flash message into the SAP Fiori Launchpad*.



LESSON SUMMARY

You should now be able to:

- Explain the file system structure of AS ABAP
- Explain the general mechanism of starting and stopping AS ABAP
- List organizational considerations before stopping AS ABAP

Introducing Different Tools for Start and Stop Procedures



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Start the SAP Management Console (SAP MC) using a configuration file
- Use SAPControl on operating system level
- Start the SAP Microsoft Management Console (SAP MMC)

SAP MC Layout

The *SAP Management Console* (SAP MC) provides a common framework for centralized system management. It allows you to monitor and perform basic administration tasks on the SAP system centrally, thus simplifying system administration.

The structure of the graphical tool is quite simple. On the left there is a navigation tree, which typically provides one or more SAP system landscapes. Every SAP system is visualized in a hierarchical structure and can be expanded or collapsed.

Whenever a node is selected in the *Navigation Pane*, the *Result Pane* on the right hand side shows details about the selected entry. Functions are mainly provided by the context menu of an individual node.

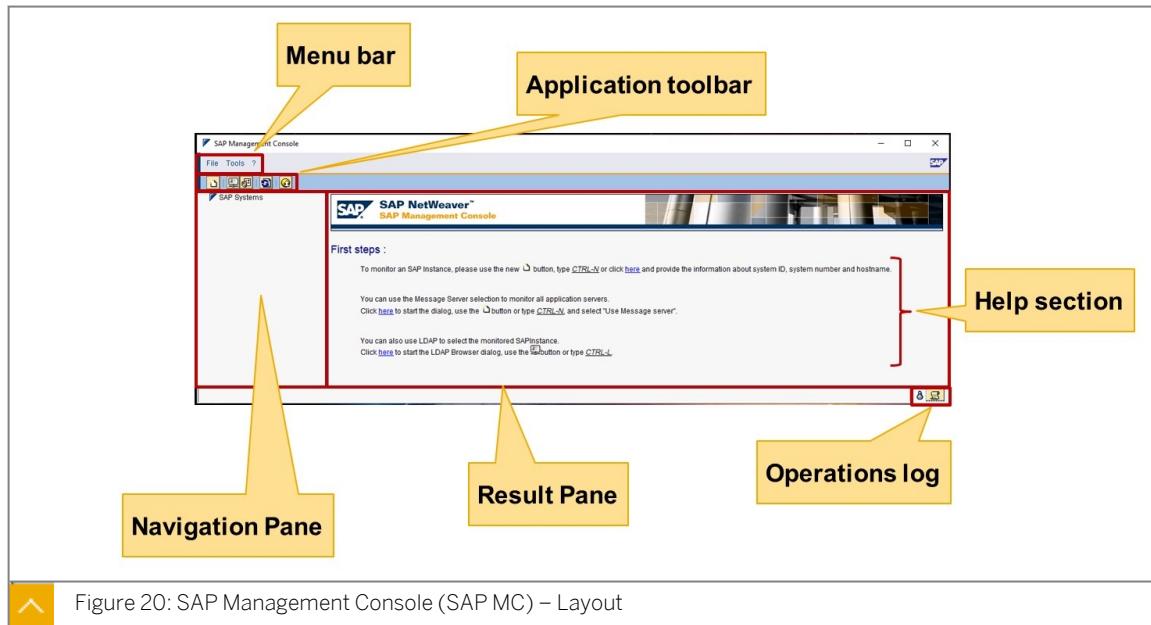


Figure 20: SAP Management Console (SAP MC) – Layout

The other areas of the SAP MC are self-explanatory. There is a small *menu bar*, an *Application toolbar*, and in the lower right corner a link to the *operations log*. The minimalistic help section *First steps* only appears in the *Result Pane* if initially nothing is selected in the *Navigation Pane* on the left.

Dynamic behavior is a core characteristic of the Java-based graphical tool. The system icons and the nodes for instances, processes, and alerts are displayed in different colors depending on their state:

- Gray: Unknown state, only outdated values, system/instance/process/service is offline
- Green: Running
- Yellow: Starting or stopping or warning (not all elements are running)
- Red: Error

When selecting a node in the tree structure of the *Navigation Pane*, the status of the individual elements (for example, SAP systems or instances) is also shown in the section on the right with detailed status information, which is very useful in case there is a one-to-many relationship between the relevant elements on the left and on the right.

Further functions like *filters*, *sorting* (descending/ascending), *progress bar*, and other features can be used in combination with selected content.

The list of tree nodes in the *Navigation Pane* is exhaustive. The SAP system hierarchy covers the included instances, process lists and process tables, current status and open alerts for monitoring, as well as sessions, caches, log and trace files, and many more useful information.

You can view the SAP MC version in the menu by selecting the question mark icon ? → *About* in the *Application toolbar*.

SAP MC: Background Information

SAP Management Console (SAP MC) is a platform-independent application that allows you to display monitoring information and perform administrative tasks, such as starting and stopping, in the same user interface on different operating systems and databases. SAP MC is a standalone Java application that is supplied as standard with the *SAP kernel* and is ready for use without any additional installation.

According to the SAP recommendation customers should always use the latest version of the SAP MC, otherwise certain features might not be available.

The SAP MC is provided with the SAP kernel and can be found in the explicit kernel sub-directories /servicehttp/sapmc on *UNIX/Linux* and \servicehttp\sapmc on *Windows* at the well known kernel locations. As already discussed in a previous lesson, the global kernel is replicated during startup of the SAP system to the local (instance-specific) directories. Global sources of the SAP MC are the following locations:

- *UNIX/Linux*: /sapmnt/<SAPSID>/exe/<codepage>/<platform>/servicehttp/sapmc
- *Windows*: \\<SAPGLOBALHOST>\sapmnt\<SAPSID>\SYS\exe\<codepage>\<platform>\servicehttp\sapmc

The SAP MC is a Java application and - in former times - it was typically started as a *Java applet* by making use of a *Java Runtime Environment* (JRE 5.0 or higher) and a URL in combination with some mandatory browser settings (for example, Java Plugin installed and scripting of *Java applets* enabled). But the Java Plugin is a third-party component and was only supported until March 2019. As a consequence of the end-of-life date for the browser

support, a different method should be used to start the Java application SAP MC, which is a local configuration on a frontend computer making use of Java Web Start (see next section in this lesson for more information).

More information is provided by the following SAP Notes:

- [1014480](#) - SAP Management Console (SAP MC)
- [2283330](#) - SAP browser information about the Java Plugin

Because the start of the SAP MC via an URL is still defined as default start option within the default configuration file `sapmc.jnlp`, and for the sake of completeness, the URLs are still mentioned here as follows:

- `http://<host name>:5<instance_number>13`
- `https://<host name>:5<instance_number>14`

With the above URLs the `sapstartsrv` daemon or service uses its web service function and sends the Java applet to the browser that is starting the SAP Management Console.

The communication with and the use of `sapstartsrv` always raises security relevant questions, which have to be answered in the context of the individual customer scenario. Depending on the requirements and guidelines further configuration has to be done by security experts.



Note:

Security relevant features and configurations are not covered in this class. Please refer to the relevant *Security Guide* of the product in use and the appropriate section in the online documentation.

The following examples might be security relevant topics in combination with `sapstartsrv`:

- The use of *Transport Layer Security* (TLS) requires `https` configuration steps for `sapstartsrv`. The steps are described in the following SAP Note: [1036107](#) - Using SAP MC via HTTPS
- The `sapstartsrv` process manages an internal list of protected operations. These can be changed, if necessary, with the instance profile parameter `service/protectedwebmethods`.
- Determine the IP address and host name to which the web service port should be connected (default: all / 0.0.0.0) to limit accessibility in the network with the instance profile parameter `service/hostname`. Refer to SAP Note [927637](#) - Web service authentication in `sapstartsrv` as of release 7.00.

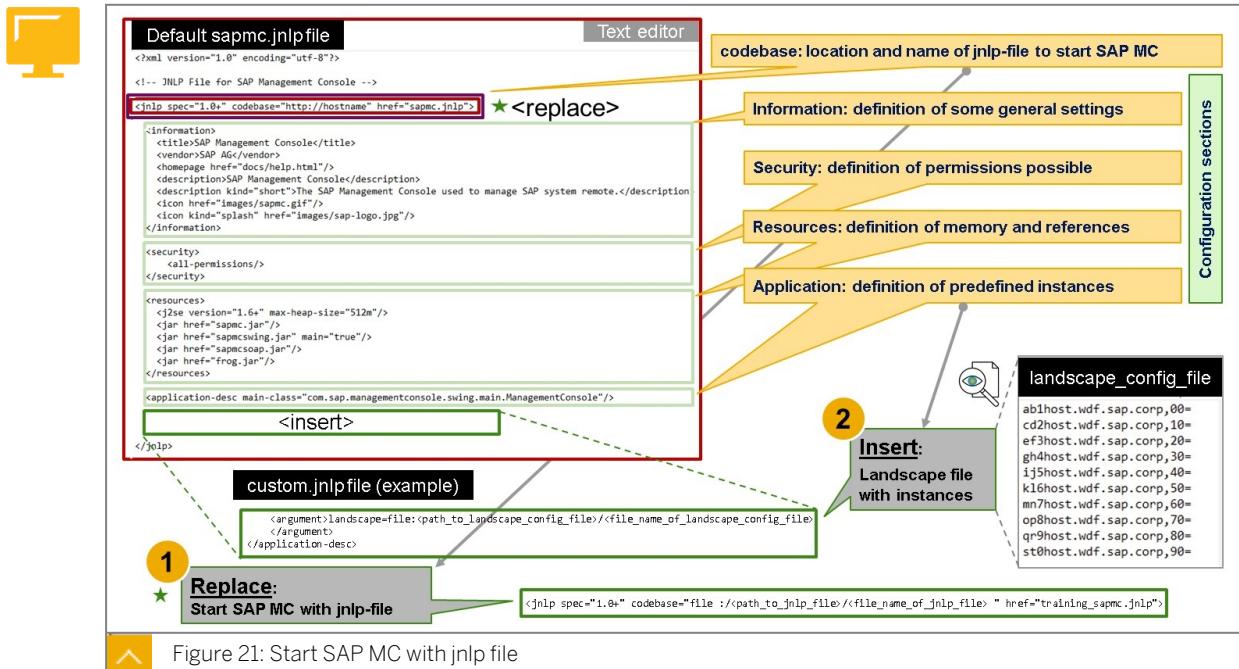
SAP MC: Start Using a Locally Configured jnlp-file

To start the SAP MC with a locally configured jnlp file based on Java Web Start, the complete content of folder `sapmc` has to be provided in combination with a locally installed Java Runtime Environment (JRE 5.0 or higher).

The configuration of the start file for the local call of the SAP MC is a minimalistic approach. It can be enhanced by a simple landscape file, specifying the SAP systems and SAP instances to be handled within the tool. So, the focus here is on the adjustment of the by default delivered jnlp-file and in addition on providing a customer-specific landscape file, which links the SAP MC to the respective resources.

The default `sapmc.jnlp` file within the above mentioned kernel sub-directory provides a specification of which `codebase` should be used. The standard `codebase` is still a URL and

points to the host (or PC or front end) where the referencing jnlp-file is provided. Even if you want to make use of this method, the **hostname** has to be adjusted according to your local environment.



Steps to Configure SAP MC locally

1. Configure in the **<jnlp_file>** the following entries:

- **codebase**: *file* instead of *http*
- **<hostname>**: adjust according to your front end
- **<file_path><jnlp_file>**: adjust according to your front-end

2. Create a **<landscape_file>** and refer to it in the **<jnlp_file>**

- Create file **<landscape_file>**
- Insert into the **<landscape_file>:<instance_hostname>** and **<instance_number>** of SAP instances (in csv format)
- Insert into the **<jnlp_file>: argument **landscape=file**:** with **<path><landscape_file>**

Step 1 - Details

Assuming Java Web Start should be used as method to start the SAP MC, you are forced to change the **codebase** and (maybe) the **hostname**. Instead of using a URL via a web browser, the keyword **file** indicates a fully qualified path to the respective jnlp-file, which should be used (see step 1 in the figure above for an example).

By purely changing the **codebase** the SAP MC would not provide any content by means of SAP systems and SAP instances. In contrast, the start method using a URL in a web browser already provides a dedicated host and port of the instance-specific *sapstartsrv* and with the default setting *Always show all SAP instances* (on the same host) of an SAP system are initially already visible.

Provide the full path to the **<jnlp_file>** and save the file on OS level of your front end (PC).

Step 2 - Details

Create a file in the same directory as the **<jnlp_file>** using a text editor and provide a list of SAP instances, which should be displayed when starting the SAP MC locally on your front end (PC). Take care of the expected csv format and put into one explicit row only one SAP instance by specifying **<instance_hostname>,<instance_number>=** (see figure above for an example).

Finally insert the following argument as parameter in the **<jnlp_file>** underneath section **<application-desc main class="...">**:

```
<argument>landscape=file:<path>/<landscape_file>
<argument>
</application-desc>
```

Further configuration can be made within the **<jnlp_file>**, for example the configuration of permissions and adjustments of the memory consumption.

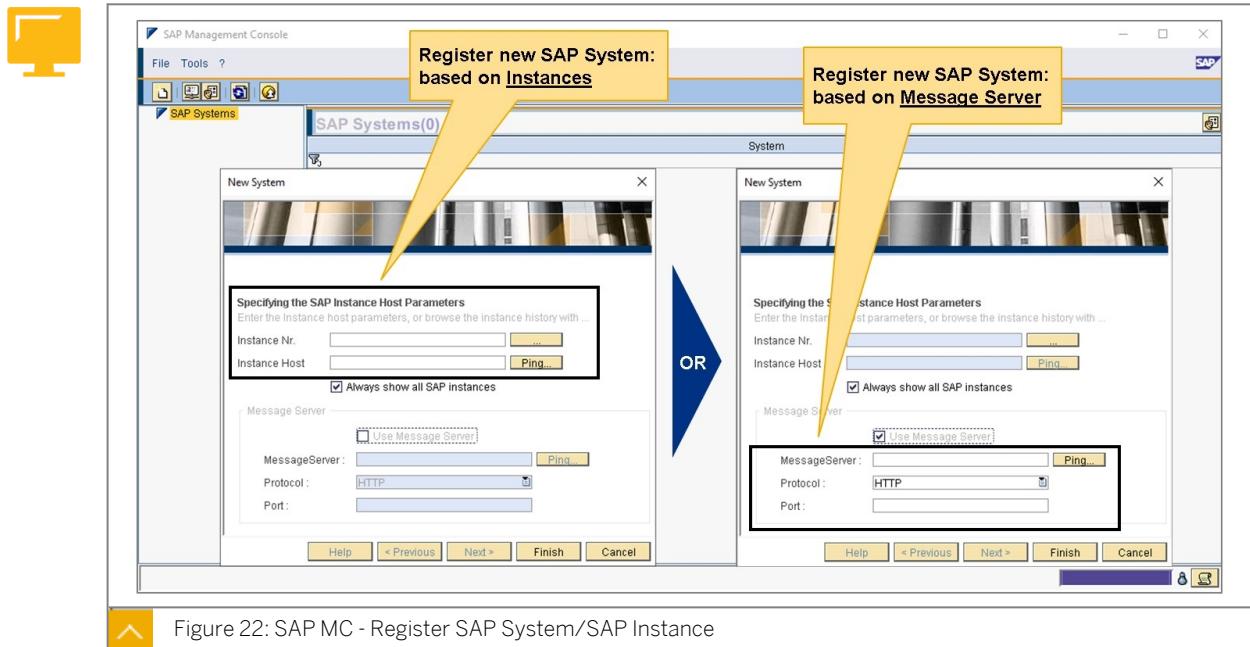
SAP MC: Loading SAP Systems, SAP Instances, and System Landscapes

The SAP MC can be used to register single SAP systems or SAP instances manually or even predefine complete system landscapes in file format and load them as needed into the tool. For both approaches the SAP MC offers a comfortable reuse of loaded content based on the existing history.

Register complete SAP systems or individual SAP instances

There are two different options to register content in the SAP MC:

- Registering Instances of SAP systems by *Specifying the SAP instance Host Parameters*
- Registering complete SAP systems by *Use Message Server* parameters

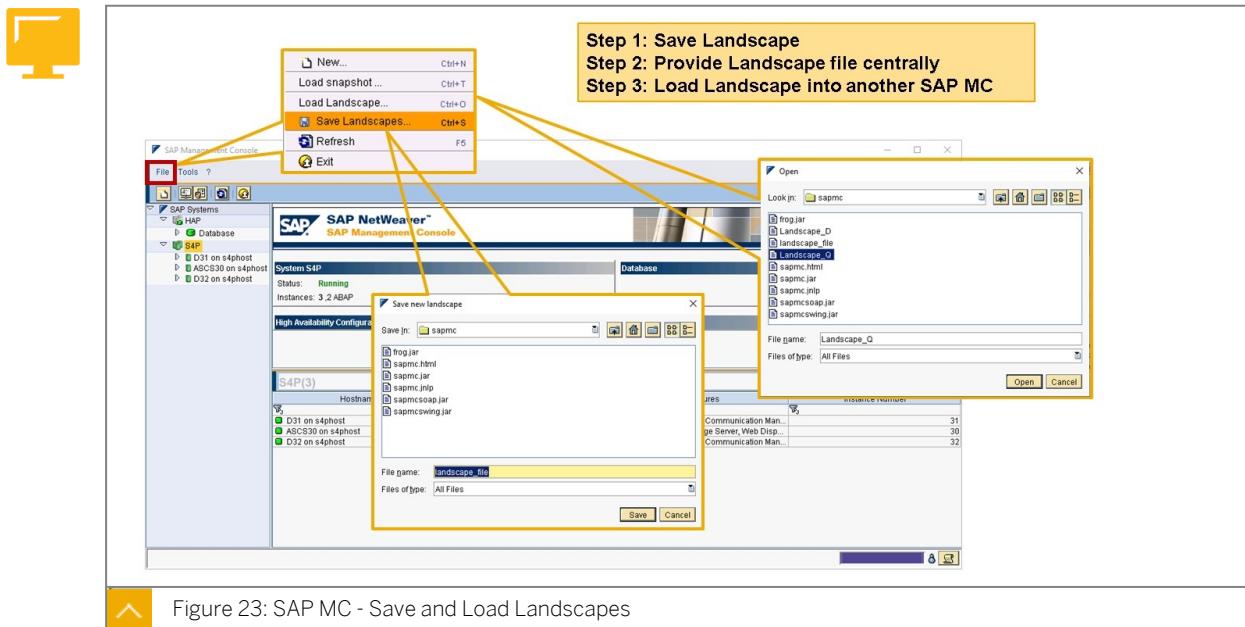


Both options offer a complete scan of the respective resource by using the flag *Always show all SAP instances*. In this way the user is not forced to repeatedly use the same wizard, if there are multiple instances detected on the specified host.

Saving and Loading Landscapes

There are also two different ways to load *Landscapes* in the SAP MC:

- Load a *Landscape* via menu path *File* → *Load Landscape...*
- Load a *Landscape* initially with the start of the SAP MC using the argument *landscape=file:<path><landscape_file>* as configuration in the *<jnlp_file>* (see figure Start SAP Management Console (SAP MC) with *jnlp* file above in this lesson).



To force a consolidated customer environment based on administrative personnel sharing the same or comparable responsibilities, it makes sense to agree on a common structure, which can be defined as a *Landscape* definition file and provided for all team members. In case individual adjustments have to be done, you are still flexible in configuring your own SAP MC environment.

Explain the Basic Idea of the SAPControl Web Service

The SAP Start Service (*sapstartsrv*) provides basic management services for systems and instances and single server processes. These core functions and many further options are exposed by the SOAP Web Service Interface named *SAPControl*.

SAPControl is part of the SAP kernel and is available on all operating system platforms supported by SAP.



Note:

The **startsap** and **stopsap** commands are deprecated since April 2015. SAP recommends that you do not use them any longer. For more information, see SAP Note [1763593](#) - Starting and stopping SAP system instances - *startsap/stopsap are deprecated*.

The *Web Service Interface* builds an abstraction layer on top of the *SAP Start Service* and therefore goes well with the operating system independent strategy of SAP. Specifically in heterogeneous environments it is feasible to use *SAPControl*, for example in the context of

scripting, without being forced to change the content when migrating to another platform (there are only some OS-specific exceptions).

In addition *SAPControl* offers significantly improved functionality with respect to platform independent monitoring.

Most methods use similar in and out parameters and return information in a table like data structure. The interface is using SOAP exception and HTTP error code for error handling.



Hint:

For more information on *SAPControl*, read the following document: **How to use the SAPControl Web Service Interface** - <https://www.sap.com/documents/2016/09/0a40e60d-8b7c-0010-82c7-eda71af511fa.html>

Check the Bandwidth of SAPControl Commands

You can use *SAPControl* to start or stop an SAP system from the command line and use a variety of options to handle SAP systems and their elements, like instances, processes, and services.



Note:

Be aware of the following:

SAPControl is the official name, but the core element is the command line tool, which is treated to be case-sensitive on OS-level. It has to be used according to the correct syntax, by means of *sapcontrol* itself always in lower case letters, whilst the functions and/or options require a specific mixture of upper and lower case characters.



The screenshot shows a terminal window with the following content:

```

<hostname>:~$ apadm 108> sapcontrol -h
NAME
    sapcontrol (Version: 785, patch 200, changelist 2150416)
SYNOPSIS
    sapcontrol [-prot <protocol>]
                [-trace <filename>]
                [-debug]
                [-user <user> <password>]
                [-queryuser]
                [-repeat <n> <d>]
                [-format <format>]
                [-host <hostname>]
                [-systempki <profile>]
                [-tio <timeout>]
                [-tmax <timeout>]
                [-nr <instance number>]
                -function <webmethod> [parameter list]

DESCRIPTION
    Control and monitor SAP instances via WebService interface of SAP Start Service.

OPTIONS
    -prot <protocol>
        Specify the protocol for the communication with the SAP instance.
        Available protocols are:
        NI_HTTP    HTTP using SAP NI sockets (default, prefer Unix domain sockets)
        NI_HTTPS   HTTPS using SAP NI sockets (prefer Unix domain sockets)
        GSOAP_HTTP HTTP using gsoap built-in sockets
        WINHTTP   HTTP using Windows winhttp
        WINHTTPS  HTTPS using Windows winhttp
        PIPE      Windows named pipes (on Unix same as NI_HTTP)

    -trace <filename>
        Trace SOAP request/response

    -debug
        Write local trace to stderr

    -user <user> <password>
        OS user and password for Webservice authentication

    -queryuser
        Query interactively for user and password

    -repeat <n> <d>
        Repeat webmethod call <n> times (-1=forever) with <d> sec delay
    -format <format>

```

To the right of the terminal window, there are two boxes:

- Syntax:**

```

sapcontrol [-prot <protocol>] [-user <user> <password>]
           [-host <host name>] -nr <instance no.>
           -function <webmethod>

```
- WEBMETHODS:**

```

Start [runlevel]
Start [hostname] <instance number> [<runlevel>]
Bootstrap [<hostname>] <instance number>
Stop [<softtimeout sec>]
InstantRestart [<hostname>] <instance number> [<softtimeout sec>]
Shutdown
RestartInstance [<softtimeout sec>] [<runlevel>]
StopService
StartService <ID>
RestartService
ParameterValue [<parameter>]
GetConfigFile
GetTraceFile
ListConfigFiles
ReadConfigFile <filename>
GetConfigTree
GetAlerts
GetEnvironment
GetVersionInfo
GetQueueStatistic
GetProcessList
...

```

A red box highlights the `WEBMETHODS` section of the syntax.

Figure 24: SAPControl – Syntax Start/Stop

Some core functions and the synopsis can be seen in the figure above.



Hint:

Calling **sapcontrol** in the command line without any other options displays the syntax description. Alternatively use **sapcontrol -h**.

For more information and examples on how to use the command line tool *sapcontrol*, use the following link:

[Starting and Stopping SAP Systems Using SAPControl](#)

Check the Status of the SAP System and Instances

An important preparatory task for start and stop actions of SAP systems is the verification of the status. Besides the usage of a graphical tool like the SAP MC, the command line interface of SAPControl offers a variety of so called *Webmethods* to check the status of the elements building an SAP system.

The following checks are most often used:

- Overall Status of the SAP instances for an SAP system using *Webmethod GetSystemInstanceList*

This method returns a list of all instances of the SAP system with its assigned *priority LEVEL*. The column *features* identifies the instance type specifying the core processes and/or services (for example, ABAP, MESSAGESERVER, and so on).

- Overall Status of one SAP instance using *Webmethod GetProcessList*

This method returns a list of all processes directly started by the respective *sapstartsrv* Web service according to the *Instance profile*.

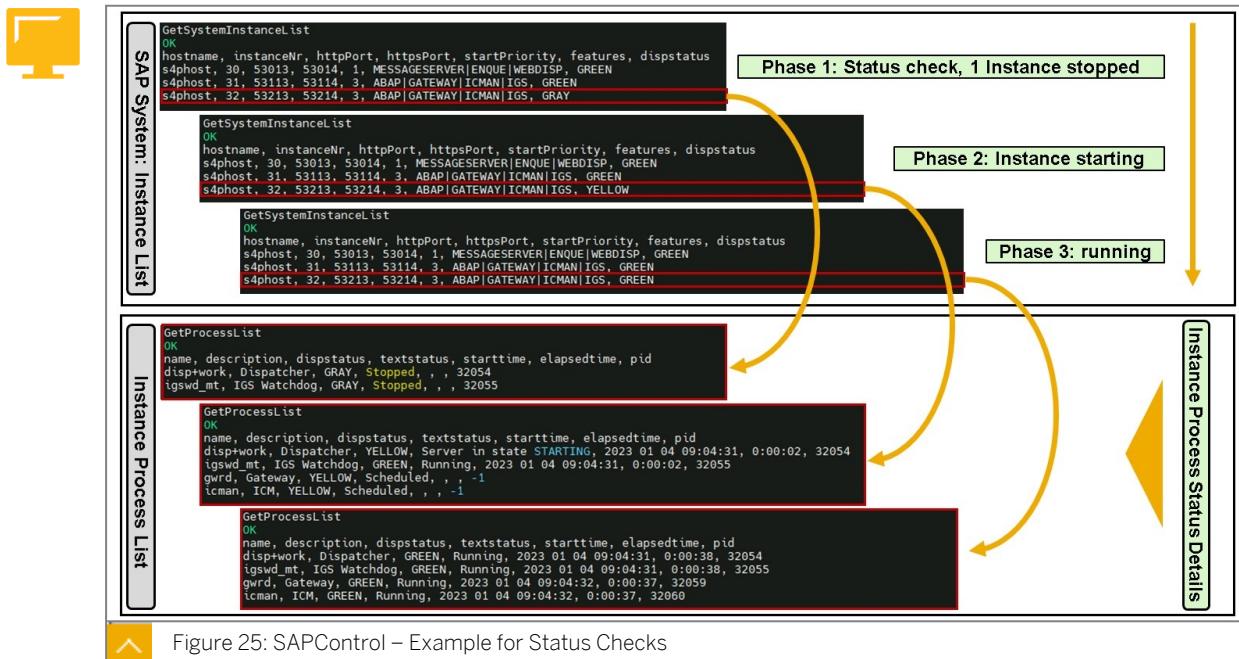


Figure 25: SAPControl – Example for Status Checks

Additional useful checks:

- Version information using *Webmethod GetVersionInfo*

This method returns a list of version information for the most important files of the instance.

- List and status of Work Processes for one SAP instance using Webmethod **ABAPGetWPTable**

This method returns a list of the ABAP work processes, similar to transaction SM50.

- List the content of the ABAP Syslog using Webmethod **ABAPReadSyslog**

This method reads the ABAP Syslog and returns it as an array of entries, similar to transaction SM21.

Status checks are often repeatedly executed. For this, SAPControl offers a feature with the option **-repeat <N> <D>** (<N> is a number for the amount of repetitions and <D> stands for a delay in seconds between the repetitions).

Use the SAP MMC

With the *Microsoft Windows* operating system, you can use both graphical tools, the SAP MC and the SAP Microsoft Management Console with an SAP snap-in (SAP MMC) to start, stop, and monitor an SAP system.

The SAP MMC allows you to start and stop the instances of SAP systems. For some databases, you can also display administrative information and the status of the database. Further database types can also be administered using the SAP MMC.

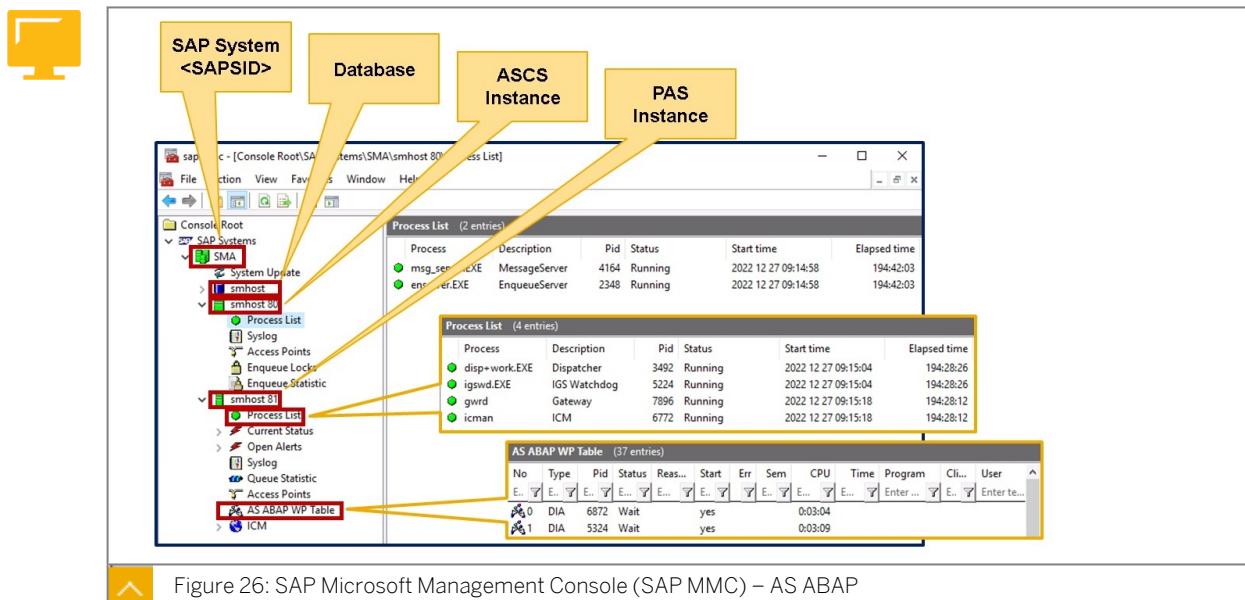


Figure 26: SAP Microsoft Management Console (SAP MMC) – AS ABAP

The look and feel of the SAP MMC is very similar to the SAP MC. The same applies to the information displayed, as well as the functions provided.

In contrast to the SAP MC, SAP instances are not named with their physical directory name in the file system (for example, *D<instance_number>*), but with the host name and instance number. The identification of the instance type has to be done using the <instance_number> or drilling down to the *Process List* and verifying the allocated processes and services.



Hint:

A leading zero as part of the 2-digit <instance_number> will be hidden in the navigation tree of the SAP MMC.

Using functions in the SAP MMC is not different in comparison to the recommended SAP MC. The SAP MMC also asks for authentication with an OS user like the <sid>adm (or the SAPService<SID> on Windows platforms) when the first function based on an SAP system or instance will be addressed. It also takes care of the assigned *priorities* in case complete systems are started or stopped. Manual handling of individual instances is also possible.



LESSON SUMMARY

You should now be able to:

- Start the SAP Management Console (SAP MC) using a configuration file
- Use SAPControl on operating system level
- Start the SAP Microsoft Management Console (SAP MMC)

Unit 2

Lesson 3

Starting and Stopping AS ABAP

LESSON OVERVIEW

This lesson presents the tools for starting and stopping SAP systems. You can use the SAP Management Console (SAP MC) for starting and stopping an SAP system and individual instances. Alternatively, you can also use the command-line tool SAPControl. These tools can be used independently of the operating system. For SAP systems that are installed on Microsoft Windows hosts, you can use the SAP Microsoft Management Console (SAP MMC).

Business Example

You are using an Application Server ABAP (AS ABAP) and different operating system platforms such as Microsoft Windows or UNIX. To start and stop the SAP systems based on AS ABAP you require information about the use of the available tools.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Start and stop using SAP MC
- Start and stop using SAPControl
- Start and stop using SAP MMC

Database Start and Stop Using SAP MC and SAP Host Agent

The SAP MC also enables you to start and stop the database. This function is not implemented in other tools, such as the SAP Microsoft Management Console (SAP MMC) or SAPControl. With tools, such as SAPControl, the database is only started when a PAS or AAS instance is started on the same host on which the database is installed (on condition that the database belongs to the same SAP system).

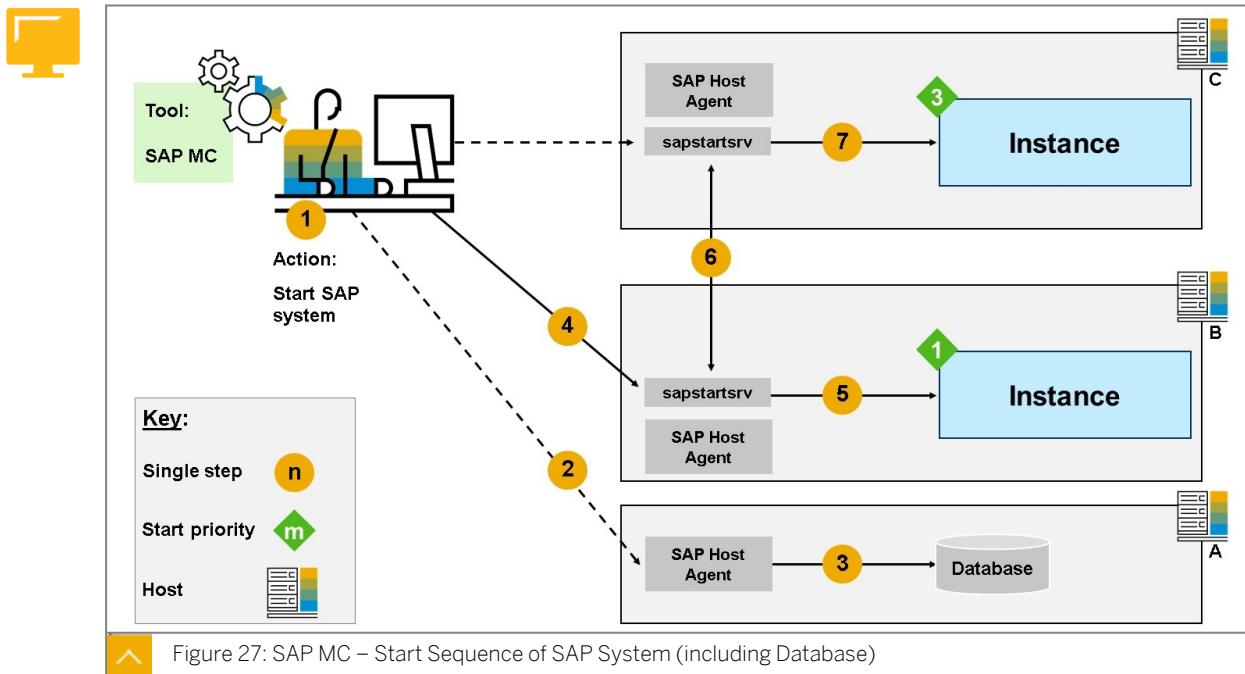
On the host where the database of the SAP system is installed, an SAP Host Agent must be installed, too. The SAP Host Agent starts and stops the database and provides monitoring data from the operating system to the SAP MC. The SAP MC receives the core information from sapstartsrv of the SAP instances that communicates on which host the database is installed. The SAP MC connects to the SAP Host Agent of the communicated host via port 1128 (or 1129 for secure network communications - SNC).



Note:

On each host on which an SAP instance is installed, an SAP Host Agent should be installed, too. The SAP MC also connects to the SAP Host Agent of the individual instances to gather information about memory consumption, file system allocation, and processes of the corresponding operating system.

The SAP MC holds a connection to each sapstartsrv process and the SAP Host Agent (see figure below).



When it comes to a start action, the *SAP Host Agent* is addressed first to start the database, if the status is *Stopped*. For all SAP instances the *sapstartsrv* negotiate, based on the *priority LEVEL*, the further start sequence. Accordingly the instances are started. The communication takes place directly between the *sapstartsrv daemon* or services. Instances with *priority LEVEL 1* (ASCS instance in ABAP environments) are in most scenarios started directly after the database, although they could also be started before the database. *Central Services* are not dependent on a successful connection to the database at this stage within the start procedure.

The procedure to start an ABAP-based SAP system with instance *priorities LEVEL 1 and 3* is a typical scenario and explained here with the steps shown in the previous figure *SAP MC: Sequence when Starting an SAP System (including Database)* as follows:

1. The administrator uses the *SAP MC* to start the SAP system.
2. Before starting the SAP system, the *SAP Host Agent* on the database host receives the command to start the underlying database.
3. If the database has the status *Stopped*, the *SAP Host Agent* uses a database start command. In case the database status returns *Running*, the next step is executed.
4. The *sapstartsrv* of the instance with *priority LEVEL 1* receives the command to start the instances according to their *priority*. The connection point is the result of identifying the instance with the highest *priority LEVEL*, which corresponds to the lowest number!
5. The instance with *priority LEVEL 1* is started by the corresponding *sapstartsrv*.
6. The *sapstartsrv* of the instance with *priority LEVEL 1* triggers the *sapstartsrv* of the instance with *priority LEVEL 3*, signaling it to start its instance.

**Note:**

In case there is a scaled SAP system given, which is a system with at least one further AAS instance, the signal to start will be sent in parallel to all instances with the same *priority LEVEL*.

7. The instance with *priority LEVEL 3* is started by the respective *sapstartsrv*.

The status of a database in the SAP MC is self-explanatory, according to the colors (already discussed in a former lesson): a stopped database is displayed in gray, a started database in green color. In phases starting and stopping it shows up in yellow color.

**Note:**

The SAP MC provides for some databases an option to stop the database when you stop the SAP system. If the flag is set, the SAP Host Agent stops the database when all the instances have been stopped. This does **not** apply to SAP HANA-based SAP systems, like the SAP systems used in this class!

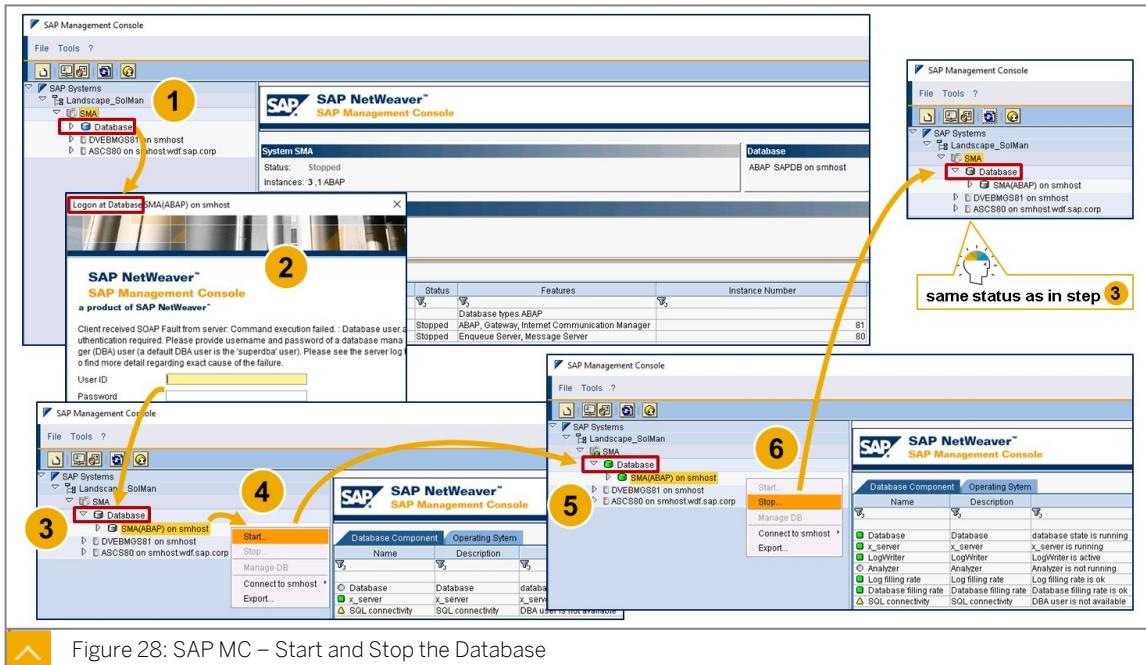


Figure 28: SAP MC – Start and Stop the Database

You can start and stop the database independently from the SAP system in the SAP MC by expanding the *Database* node and selecting the relevant option in the context menu based on the <DBSID>.

**Hint:**

Remember the dependency of the *Application Server* (AS) instances to a database in operational state. Technically spoken it is possible to stop the database even if AS-based instances were started before. But practically users cannot continue their work in the ABAP-based SAP system.

**Hint:**

If the database icon is colored *blue* in the SAP MC, it could have different reasons, for example:

- The SAP Host Agent may not be installed
- The SAP Host Agent version is outdated
- The database authentication is missing

Start of an SAP System Using the SAP MC

In the *Navigation Pane* on the left of the SAP MC the SAP system displays directly or included in a *Landscape* (folder). The individual SAP systems can be expanded to view details, for example different types of SAP *instances*, and further down other elements like *Process Lists*, technical information, and monitoring data.

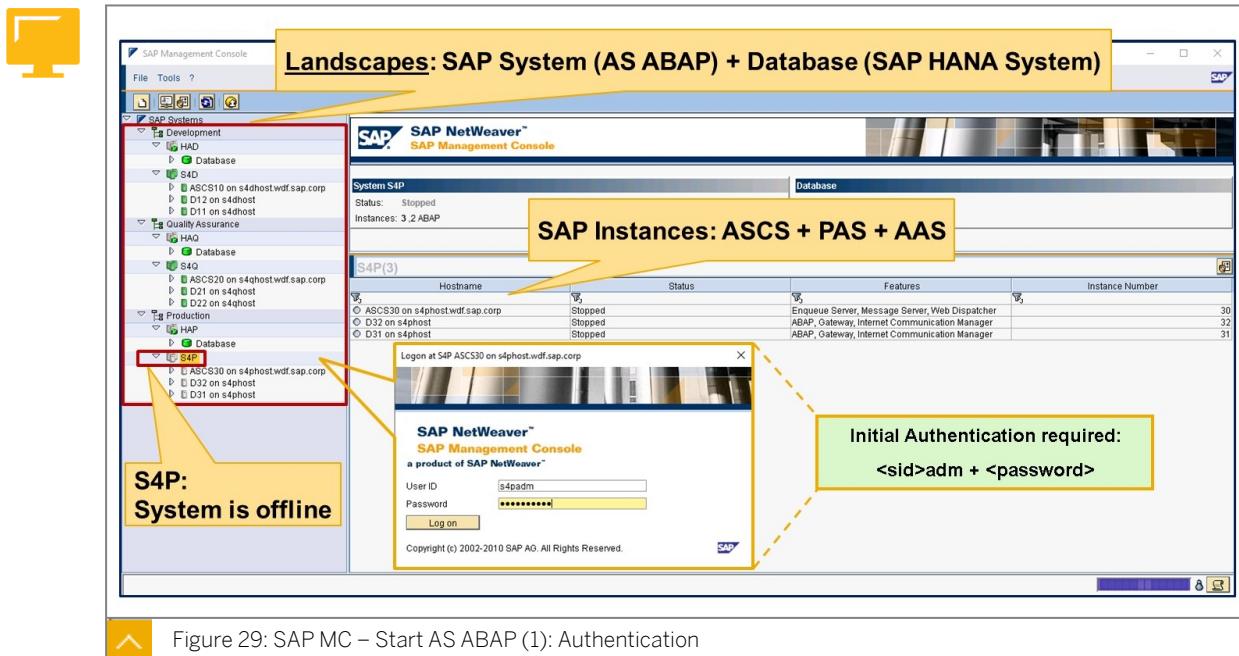


Figure 29: SAP MC – Start AS ABAP (1): Authentication

Different colors are used to identify the status of systems, instances, processes, and services within the SAP MC (please refer to the list in the predecessor lesson).

When an access to an SAP system in the *Navigation Pane* takes place for the first time after starting the SAP MC, an authentication box appears, requesting an operating system user and password. Typically the `<sid>adm` will be used. This user is authorized to start and stop an entire SAP system or individual instances. The SAP MC remembers the credentials until closing the tool.

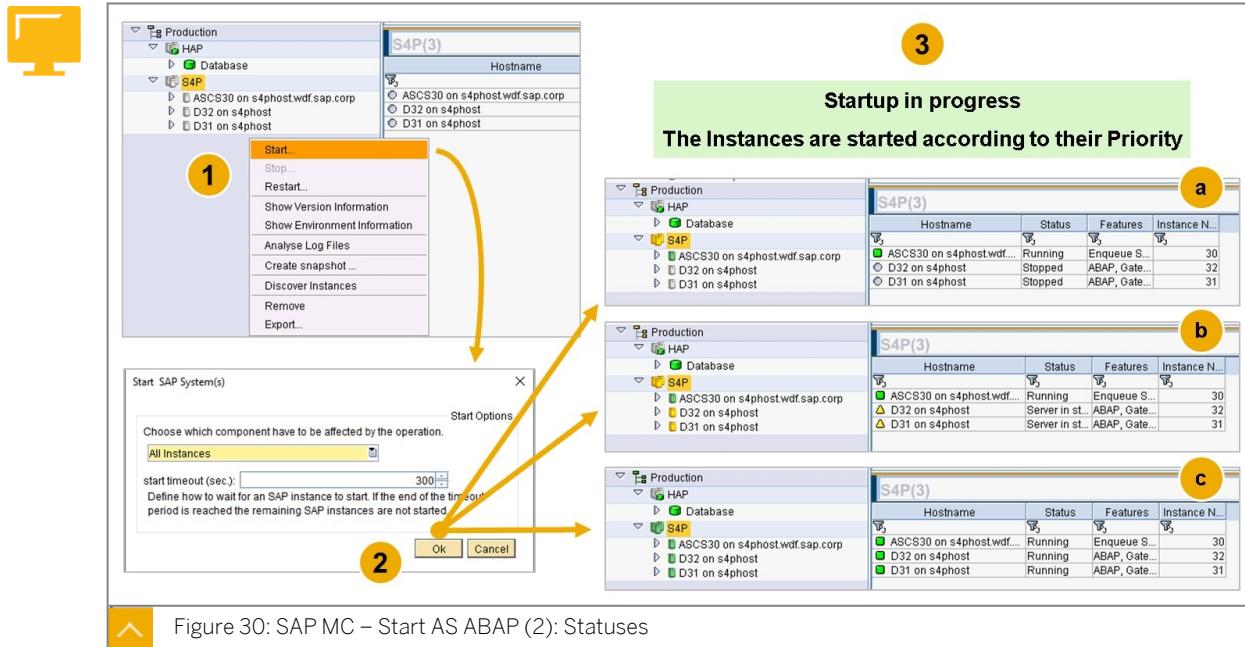


Figure 30: SAP MC – Start AS ABAP (2): Statuses

In the context of starting a complete SAP system, there is an option to select which components should be started (see step no. 2 in figure above). The default settings is *All Instances*. The SAP MC starts the individual instances of the system according to the start priority *LEVEL* and in the correct sequence. Nevertheless, alternatively the administrative personnel are able to choose another variant, like type of instance or priority *LEVEL*.



Hint:

Have a look at the possible selection for the components, which can be started when doing the exercises.

A further option when starting an SAP system is the definition of so called *start timeout*. This can be seen in the figure above at step 2, looking at the dialog box. This *start timeout* is defined as a maximum period for the instance to start. If all instances with the same priority *LEVEL* do not start within this timespan, instances with the next priority *LEVEL* cannot be started. The default value is 300 seconds and can be overridden in the field of the dialog box. The timeout countdown begins again when the start process of another priority *LEVEL* is completed. If the *start timeout* limit was reached, the remaining SAP instances are not started and the procedure will be stopped.

From a handling perspective the administrative personnel are supported by visualization features in the SAP MC. By selecting a specific node (for example, an instance) in the *Navigation Pane* tree structure at the left of the SAP MC, the details about the current status of the single elements can be identified. The information is provided in the *Result Pane* on the right.

Here are some example statuses during startup of an SAP system:

- *Running*

All elements of the selected component have been started successful and are running.

- *Running but status info unavailable*

All elements of the selected component have been started successful, but no further information is available.

- Server startup procedure running

The start was initialized and the procedure is still ongoing, but did not yet finish.

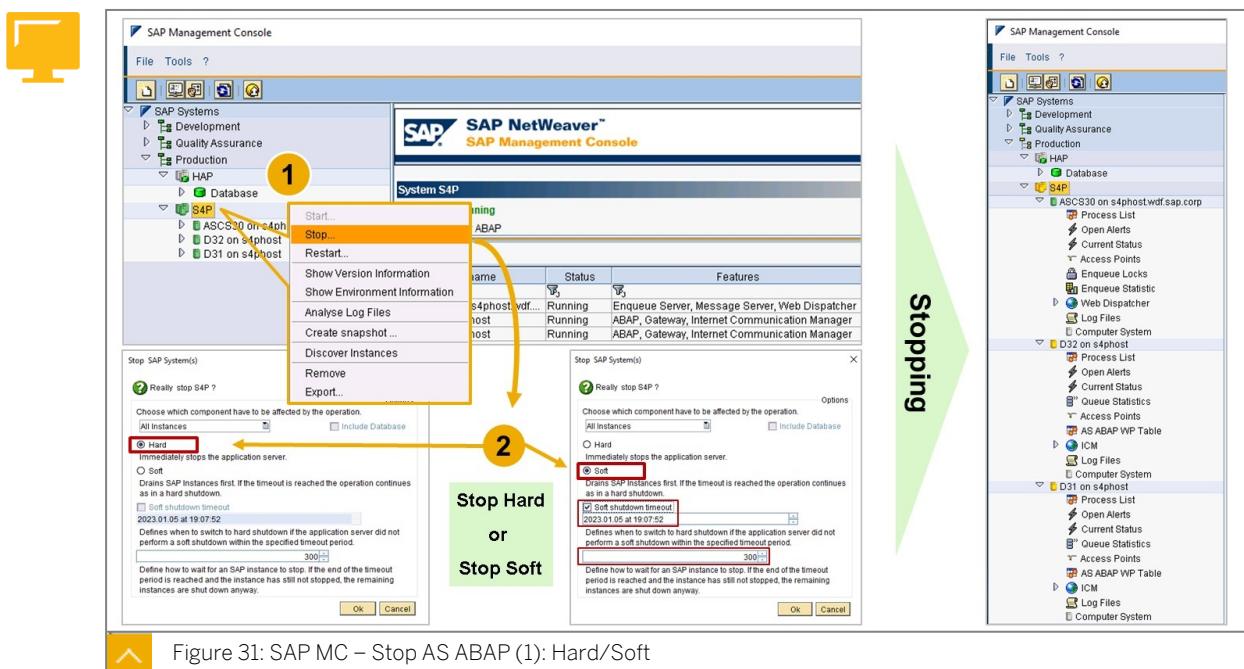
- Dialog Queue standstill

This status indicates that the start process is not fully finished and that *ABAP work processes* may not yet be fully initialized.

In any case use the recommended best practice procedure for verification of a start action, which is logging on to the system or instance to check the system availability.

Stop of an SAP System Using the SAP MC

When an SAP system is stopped using the SAP MC, the stop process is executed following the reverse sequence of the start process taking the *priority LEVEL* into account. After calling the stop process, a dialog window appears where you can choose between *hard* and *soft* shutdown options. *Soft shutdown* means that no further user requests are received by the instance and processes are terminated if no process is handling a request. *Hard shutdown* means that all request processing is terminated immediately and open transactions are rolled back in the underlying database.



The *Soft shutdown timeout* specifies how much time instances with the same *priority LEVEL* are given to complete the stop process before instances with the next lowest *priority LEVEL* are stopped. If the PAS and AAS instances (with priority 3) cannot be stopped within 300 seconds, for example, the ASCS instance (with *priority LEVEL* 1) is stopped once this timeout has expired, and so on.

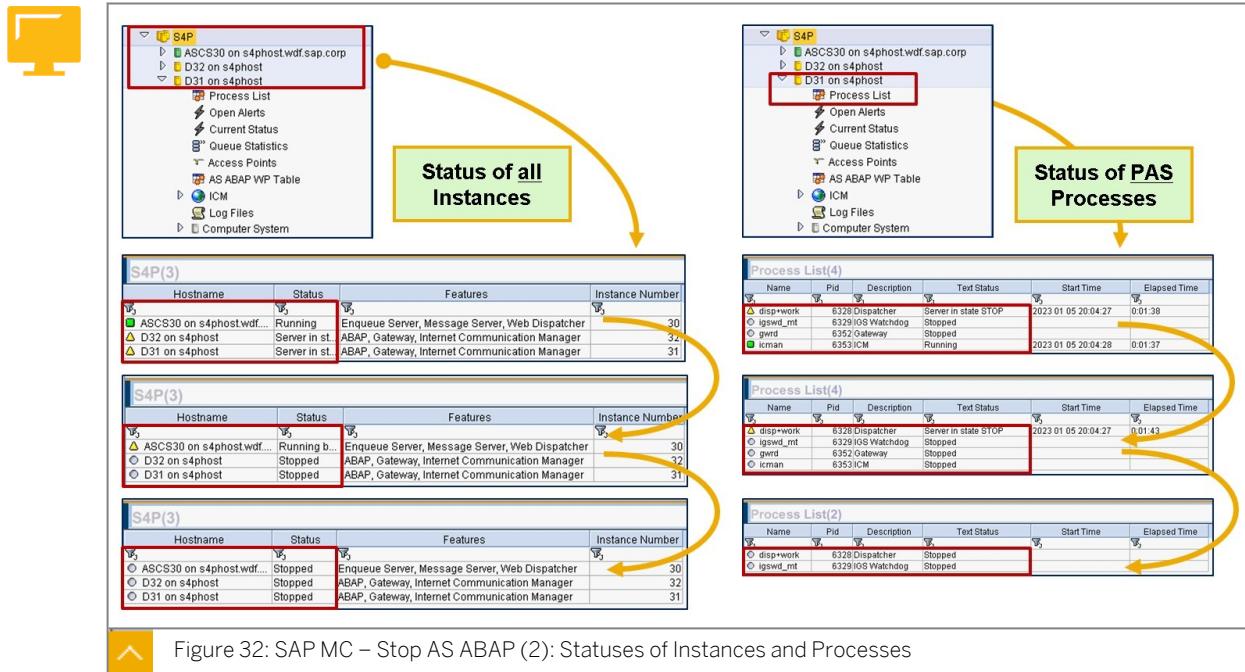


Figure 32: SAP MC – Stop AS ABAP (2): Statuses of Instances and Processes

The figure above clarifies the different statuses during an SAP system stop procedure. The statuses on the left show in which sequence the instances of the SAP system are stopped, whilst on the right the statuses of the processes of one AS instance can be seen. In this case it is the PAS instance of the SAP system.

As already discussed in this and a predecessor lesson, only some databases can profit from an integrated database shutdown within the SAP MC and if specific conditions are fulfilled. If it is supported in the specific scenario, the flag *Include Database* is active and can be set by the administrator when stopping the SAP system.

Instance Start Using the SAP MC

In some situations it is meaningful to start an SAP system stepwise or even start or restart only individual instances. The SAP MC provides the complete range of functionality and supports the administrative department with those features. The required options are provided in the *Navigation Pane* by making use of the context menu of the instance nodes.

A typical use case is an instance *Restart*. There are different reasons why an individual instance *Restart* might be necessary, for example:

- Parameter changes to activate new parameter values
- Solving hardware allocation problems (for example, main memory or CPU)
- Operating system software maintenance or parametrization with the need of a reboot
- Hardware changes

In highly scaled ABAP environments with multiple AAS instances, the administration might not start the SAP systems completely. Initially specific instances are started in the correct sequence and others remain offline, for example because of the need for a further downtime window.



Note:

The step-by-step start of an SAP system is treated to be an advanced feature by means of mandatory knowledge concerning interdependencies. As you already know, the *priority LEVEL* of individual instances plays an important role within the start procedure. The system administration is in charge of providing reliable start and stop procedures. As the individual start of an SAP system instance-by-instance requires mandatory further background knowledge, be careful when using these start options in productive environments.

The relevant start and stop handling options for individual instances can be found in the context menu of the respective instance in the *Navigation Pane* underneath the <SAPSID>. Depending on the current status of an instance, the functions *Start...* or *Stop...* are active or inactive (grayed out). The function *Restart...* always appears as an active option, as even if an instance is stopped, it simply gets started with this function (without a preceding stop action).



Caution:

The SAP MC does not avoid a stop or restart of a “Single Point of Failure”, if you decide to execute this action. There is only the following standard question presented in the dialog box: “Really stop <SAPSID> instance <instance_number> on <host>?”. Confirm with *Ok* or abort with *Cancel*.

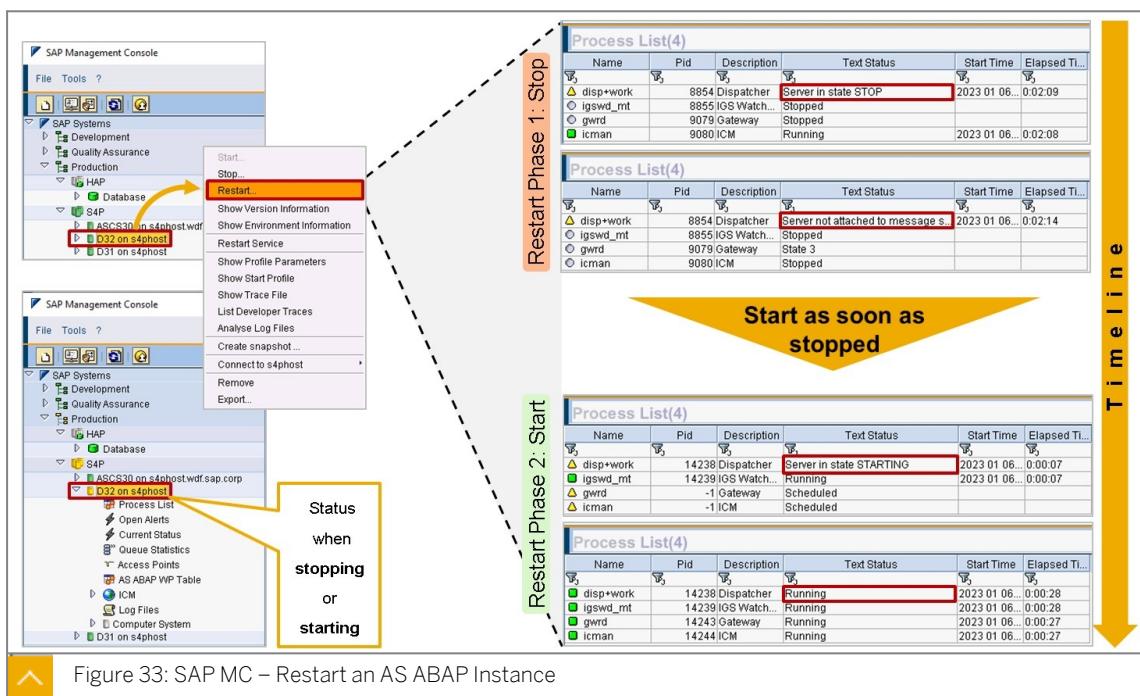


Figure 33: SAP MC – Restart an AS ABAP Instance

When you expand the instance, other elements are displayed in the SAP MC, such as the *Process List* and the *AS ABAP WP Table*. The *Process List* provides status information about the start process and indicates which processes and/or services have been started by the responsible *sapstartsrv*. At the beginning of the start process, it is possible to see the *sapcpe* process in the *Process List*, verifying, whether a replication of kernel binaries is needed.

In the figure, you can see that the ABAP Dispatcher (*disp+work*) of the AAS instance is still starting. The *IGS Watchdog* (*igswd_mt*), the *RFC Gateway* (*gwrd*) and the Internet Communication Manager (*icman*) are already running.

By checking the content of the AS ABAP WP Table, the status of a PAS or AAS instance can be verified. The status of the individual ABAP work processes, allows a conclusion of the overall status of an instance and in whether the initialization was completed (column *Start* displays yes).

Related Information

For further information about starting and stopping SAP systems, refer to the following SAP Notes:

- SAP Note [936273](#): *sapstartsrv for all platforms*
- SAP Note [927637](#): *Web service authentication in sapstartsrv as of release 7.00*
- SAP Note [823941](#): *SAP Start Service on Unix*
- SAP Note [995116](#): *Backward porting of sapstartsrv for earlier releases*
- SAP Note [1014480](#): *SAP Management Console (SAP MC)*
- SAP Note [927637](#): *Web service authentication in sapstartsrv as of release 7.00*
- SAP Note [877795](#): *Problems with sapstartsrv from Release 7.00 and 6.40 patch 169*
- SAP Note [2200230](#): *Problems with use of system PKI*

How to use the SAPControl Web Service Interface: <https://www.sap.com/documents/2016/09/0a40e60d-8b7c-0010-82c7-eda71af511fa.html>

Interface

Start of an SAP System Using SAPControl

Only as a reminder:

As already discussed in a former lesson, SAPControl is a command line tool for starting and stopping SAP systems. It uses the *Web Service interface* of the *sapstartsrv*, is part of the SAP kernel, and is available on all supported operating system platforms.



Note:

The **startsap** and **stopsap** commands are deprecated. SAP recommends that you do not use them any longer. For more information, see SAP Notes [1763593](#) - *Starting and stopping SAP system instances - startsap/stopsap are deprecated*.

The following figure shows the basic structure of the syntax with some selected examples. The execution is not yet shown, but will follow in the flow of this lesson.



Hint:

Calling **sapcontrol -h** or **sapcontrol --help** or even without any other option in the command line displays the syntax description of the tool.

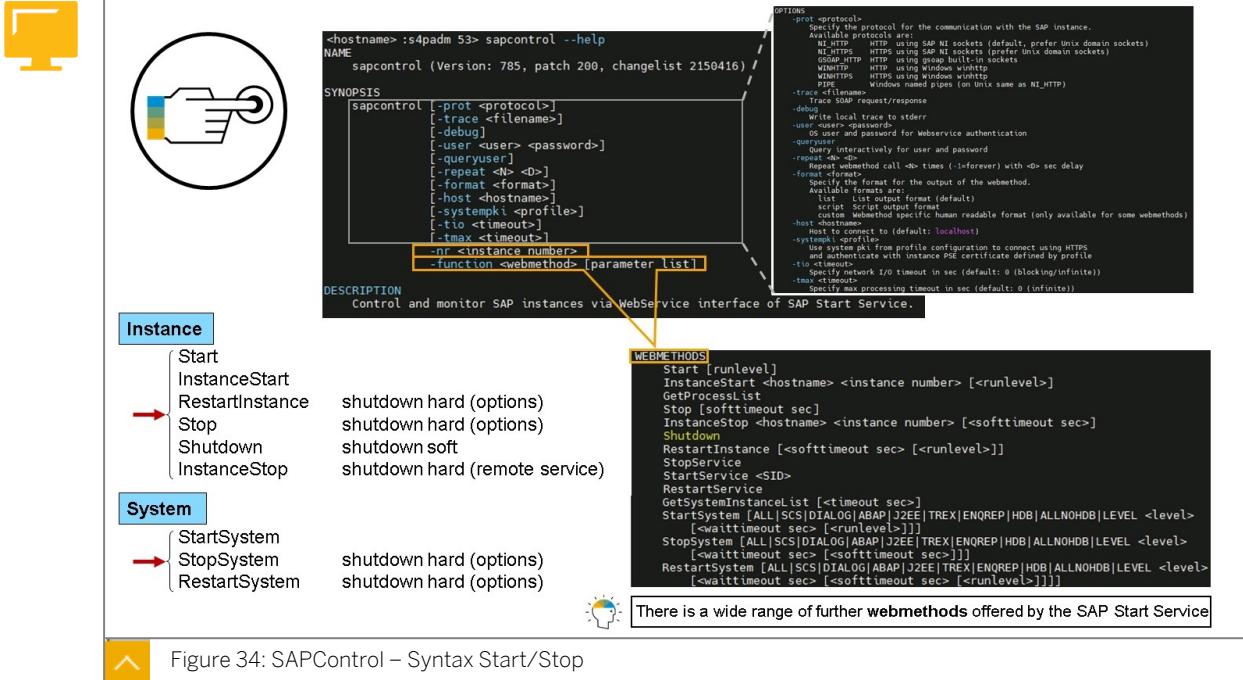


Figure 34: SAPControl – Syntax Start/Stop

As you can see in the figure above, there are multiple alternatives to start or stop an SAP system or elements of it with *SAPControl*. Most methods use by default hard shutdown options. With further parameters it is possible to trigger a soft shutdown, which can be limited, so that after a time limit is reached, the stop continues with a hard shutdown.

In addition there is a *Webmethod* to handle the *sapstartsrv*. For example to restart the *daemon* or service after important changes, which affect the processes to be started by an instance.

Use SAPControl to Start a complete SAP system

As with other *command line tools* the syntax of *SAPControl* is highly relevant. This is an important reason, why the access to the tool should be strictly limited to some administrators only. Operating system access has to be treated to be a highly critical task from a security perspective. The execution of *SAPControl* commands is bound to a shell and requires an OS user with administrative permissions.

In SAP environments there are different predefined OS users for the use case of start and stop actions. The main approach uses the *<sid>adm* created during installation. In addition an OS user with access to the *SAP Host Agent* is able to make use of *SAPControl* functions.



Note:

There are differences regarding the supported operating system platforms. Please refer to the previous lesson and the OS-specific information about *sapstartsrv*.

Which option should be used, highly depends on the environment (OS, user, security configuration, and so on) and is not discussed in details in this class. Nevertheless, there are some hints and use cases within this lesson.

Before starting or stopping an SAP system using *SAPControl*, check the status of the SAP instances building the SAP system (please refer to the previous lesson). Alternatively use a command on OS-level to check the status of the relevant processes (for example, *ps -ef | grep* or *pstree*).

The screenshot shows a terminal window titled '2.s4phost' displaying SAPControl commands and their execution. A legend on the right defines the colors: yellow for 'Start command', red for 'Check command', and black for 'Status information'. The terminal output includes:

- `<hostname>:s4padm 108> sapcontrol -user <user> <password> -nr 30 -function GetSystemInstanceList` (Status: OK)
- `<hostname>:s4padm 109> sapcontrol -user <user> <password> -nr 30 -function StartSystem ALL` (Status: OK)
- `<hostname>:s4padm 110> sapcontrol -user <user> <password> -nr 30 -function GetSystemInstanceList` (Status: OK)
- `<hostname>:s4padm 111> sapcontrol -user <user> <password> -nr 30 -function GetSystemInstanceList` (Status: OK)
- `<hostname>:s4padm 112> sapcontrol -user <user> <password> -nr 30 -function GetSystemInstanceList` (Status: OK)

On the right, a process tree diagram shows the start of SAP processes (sapstart) leading to various SAP components like SAP_S4P_30_ENQ, SAP_S4P_30_MS, SAP_S4P_32_DP, SAP_S4P_31_DP, etc., with their respective sub-processes.

Figure 35: SAPControl – Start an SAP System

In the first example `sapcontrol -user <user> <password> -nr <instance_number> -function StartSystem ALL`, the web method `StartSystem` is called with the option `ALL`. As with the SAP MC, a `sapstartsrv` process is addressed to start the instances. `ALL` means that all instances are started according to the given *priority LEVEL*. It is assumed that the instances exist and are not already operational. Typically the ASCS with an assigned *priority LEVEL 1* will take over control.

Note:

There are two exceptions:

- An *EBS* instance exists with *priority LEVEL 0.5*
- There is no *ASCS* instance installed, but a *Central* instance exists with *priority LEVEL 2*

The PAS and any AAS instances have *priority LEVEL 3* and are started after the *ASCS* instance is running. Instances can run on the same host or on different hosts. With *StartSystem LEVEL 3*, which corresponds to the above used and mentioned *StartSystem ALL*, the complete SAP system is started according to the *priority* of the individual instances, unless the instances are already running. This is the same process as with a system start using SAP MC.

Special Functions of SAPControl

Making use of special options

The use of username and password directly in the command line interface (SSH session) carries the risk that sensitive data can be viewed in the history log or process list. Under security aspects this is not acceptable and should be avoided.

If the SSH session is used directly with the user `<sid>adm` (see example above with *StartSystem ALL*), the user/password input is not needed for execution of a SAPControl command.

Another alternative is the use of a *protected web method* to use a trusted connection. No user name or password is provided because the operating system authentication as <sid>adm is sufficient. However, it is only possible to execute a *protected web method* for the local host.

- *UNIX/Linux:* **sapcontrol -prot NI_HTTP -nr <instance_number> -function Start**
- *Windows:* **sapcontrol -prot PIPE -nr <instance_number> -function Start**

If the command is used based on a specific instance like with *Webmethod Start* or *Stop*, it should work properly. But in case *StartSystem ALL* or option *LEVEL 3* is used, it will fail in a distributed environment with instances installed on different hosts. The communication with instances on other hosts is not permitted under trust connections. Only local functions can be executed with this method.



Caution:

Read SAP Note [927637](#) - Web service authentication in sapstartsrv as of release 7.00.

The limitation of executing *protected web methods* only on a local basis can be eliminated with the use of option **-systempki**. In this way it is possible to use *Webmethods* like *StartSystem priority LEVEL* or *StartSystem ALL* instead of explicitly specifying the OS user and password with the option **-<user><password>** in the command line. This method also works in distributed environments, where the instances are installed on different hosts. Nevertheless, a dedicated configuration is needed to make use of this option (not covered in this class).

The following commands are examples using the option **-systempki**:

- **sapcontrol -systempki <path_to_instance_profile><name_of_instance_profile> -nr <instance_number> -function StartSystem ALL**
- **sapcontrol -systempki <path_to_instance_profile><name_of_instance_profile> -nr <instance_number> -function StopSystem LEVEL 3**

Stop of an SAP Instance Using SAPControl

Use SAPControl to Stop specific Instances of an SAP system

If the administration is planning to stop only a specific instance type, this action can also be done using a *StopSystem* command. In case all AS instances should be stopped, the option *LEVEL* with value 3 can be executed. All instances with lower *priority LEVEL* values remain up-and-running, for example the ASCS instance (if existent). With *StopSystem LEVEL 1*, also the ASCS instance would be stopped (but not an installed ERS instance!).

```

<hostname>:s4padm 108> sapcontrol -systempki /usr/sap/S4P/SYS/profile/S4P_ASCS30_s4phost -nr 30 -function GetSystemInstanceList
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
s4phost, 30, 53013, 53014, 1, MESSAGESERVER|ENQUE|WEBDISP, GREEN
s4phost, 31, 53113, 53114, 3, ABAP|GATEWAY|ICMAN|IGS, GREEN
s4phost, 32, 53213, 53214, 3, ABAP|GATEWAY|ICMAN|IGS, GREEN
<hostname>:s4padm 108> sapcontrol -systempki /usr/sap/S4P/SYS/profile/S4P_D31_s4phost -nr 31 -function Stop
Stop
OK
<hostname>:s4padm 109> sapcontrol -systempki /usr/sap/S4P/SYS/profile/S4P_D31_s4phost -nr 31 -function GetSystemInstanceList
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
s4phost, 30, 53013, 53014, 1, MESSAGESERVER|ENQUE|WEBDISP, GREEN
s4phost, 31, 53113, 53114, 3, ABAP|GATEWAY|ICMAN|IGS, GREEN
s4phost, 32, 53213, 53214, 3, ABAP|GATEWAY|ICMAN|IGS, GREEN
<hostname>:s4padm 110> sapcontrol -systempki /usr/sap/S4P/SYS/profile/S4P_D31_s4phost -nr 31 -function GetSystemInstanceList
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
s4phost, 30, 53013, 53014, 1, MESSAGESERVER|ENQUE|WEBDISP, GREEN
s4phost, 32, 53213, 53214, 3, ABAP|GATEWAY|ICMAN|IGS, GREEN
s4phost, 31, 53113, 53114, 3, ABAP|GATEWAY|ICMAN|IGS, GRAY
<hostname>:s4padm 111> sapcontrol -systempki /usr/sap/S4P/SYS/profile/S4P_D31_s4phost -nr 31 -function GetProcessList
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
disp-work, Dispatcher, GRAY, Stopped, , , 5780
qswd mt, IGS Watchdog, GRAY, Stopped, , , 5781

```

Figure 36: SAPControl – Stop an Instance

Start and Stop Using SAP MMC

The *Microsoft Management Console* (SAP MMC) is used on *Windows* operating system platforms. The installation of an SAP system on *Windows* adds a *Snap-in* to make available SAP systems in the console, in order to provide start and stop functions. This tool, with the *Snap-in*, is called SAP MMC and can be used on any *Windows* operating system platform, so even on front-end computers. SAP systems on platform *UNIX/Linux* can be included using the *Snap-in*.

The SAP MMC is OS-dependent and provides functions comparable to the SAP MC.



Note:

In this class, the SAP Microsoft Management Console (SAP MMC) is only covered for the sake of completeness. SAP recommends to use the OS-independent SAP MC as graphical tool to start and stop SAP systems.

Start and Stop an SAP System using the SAP MMC

The following figure describes the start of an ABAP based SAP system.

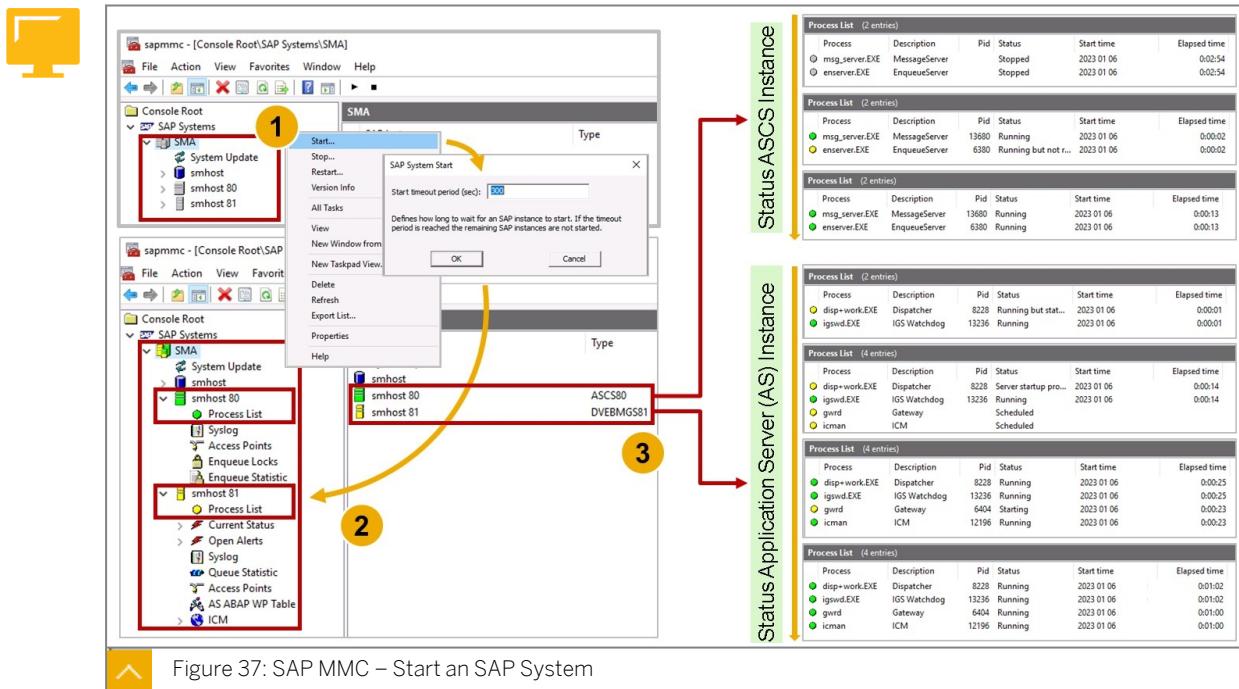


Figure 37: SAP MMC – Start an SAP System

When stopping or restarting an SAP system in the SAP MMC, options will be provided on how to shutdown. There is no difference in comparison to the already discussed tools SAP MC and SAPControl. The following figure shows the corresponding options.

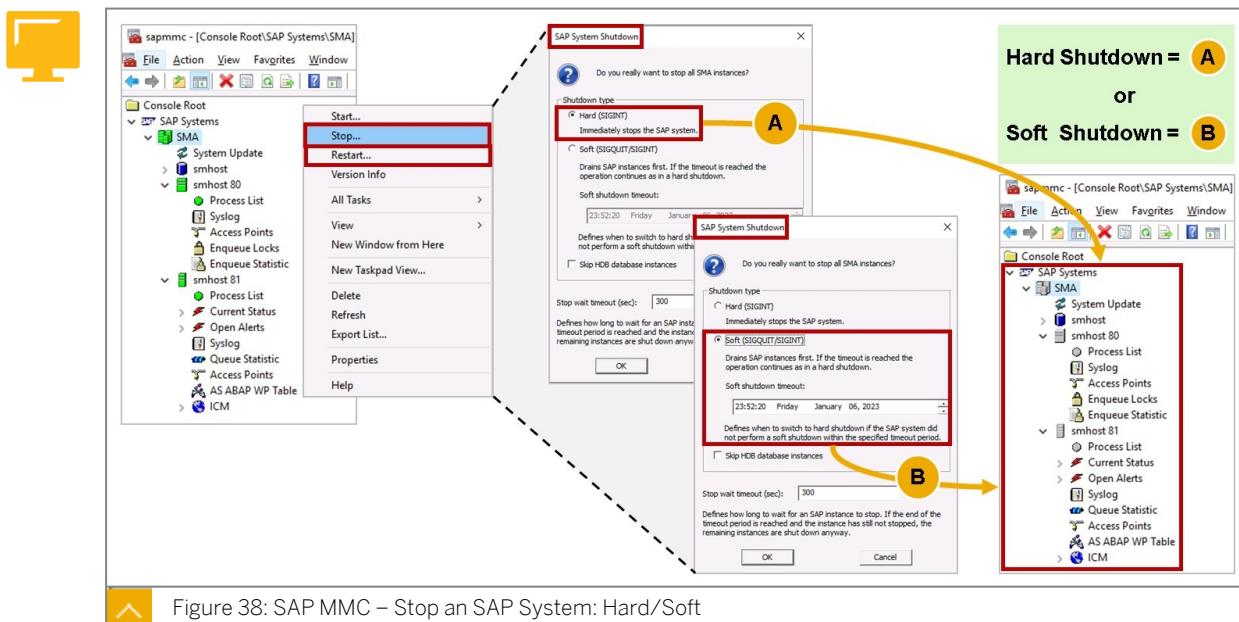


Figure 38: SAP MMC – Stop an SAP System: Hard/Soft



LESSON SUMMARY

You should now be able to:

- Start and stop using SAP MC
- Start and stop using SAPControl
- Start and stop using SAP MMC

Log and Trace Information for System Start and Stop

LESSON OVERVIEW

In this lesson, you will become familiar with the most important log and trace files that log the start of an ABAP-based SAP system.

Business Example

Problems occurred when you started an SAP system. To correct them, the administrator analyzes the logs and trace files that were generated during the start.



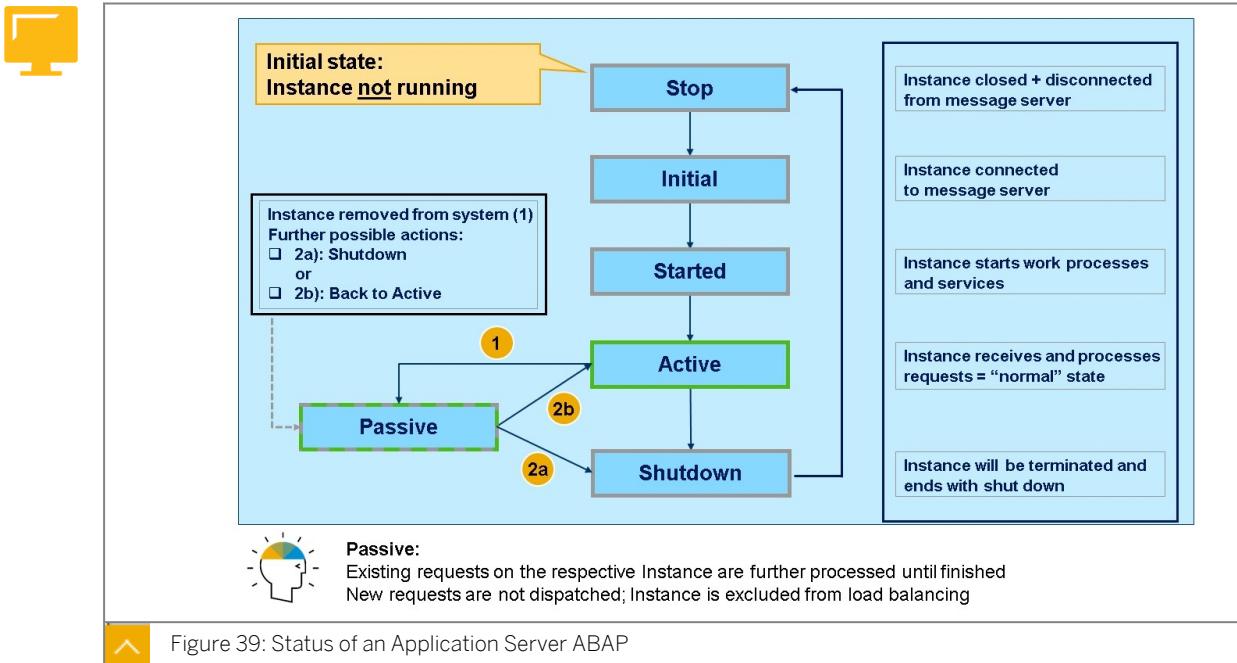
LESSON OBJECTIVES

After completing this lesson, you will be able to:

- List log and trace information for troubleshooting
- Use SAP MC and SAPControl to access log and trace information
- Appendix: further analysis options

Statuses of Application Server ABAP

Before drilling down to log and trace files, it makes sense to have a closer look at the different statuses of an ABAP *Application Server* (AS) instance. With this information in mind, it should be easier to identify a potential problem and isolate the root cause. This knowledge is essential for administrative personnel and should support a target-oriented troubleshooting procedure. Treat it as background information for your practical work. Within this class there is no further detailed discussion about this topic, but the different statuses might be seen in multiple log and trace files when reviewing a start or stop procedure.



In case some instances of the SAP system are in status *Active* and others not, and you are able to log on to the SAP system, use transaction **SM51** or **SMMS** to check the status of the *Application Server* instances.

Without being able to log on to the ABAP stack, log and trace files can be used to search for the specific status of an SAP *Application Server* instance (see section below for details). The typical statuses in the context of a start action of an AS ABAP are shutdown, stop, initial, started, and active and can be seen in the log files. A special status called “*Passive*” can be established to take out an instance temporarily from load balancing for maintenance reasons. When this status is pushed, existing requests are further processed until finished, but newly incoming requests are only distributed to the remaining AS ABAP instance(s). In this way the instance will once be isolated and can be maintained when all existing requests were finished. At the end of the maintenance, the instance is brought again to status active, for example using transaction **SM51** or **SMMS**.

Log and Trace Files for the Start Process

Preliminary Note

In this lesson you should get an idea about:

- Where to find log and trace information relevant for start and stop procedures of SAP systems
- Which files are most relevant for the start procedure
- Which tools are used to access this information

This is our main goal, and not troubleshooting. As a team member of the administrative department, responsible for the start and stop procedures of SAP systems, you should get a mental map on where to have a look at which information, and using which tools, so that in case of a problem you are able to start the analysis of what happened and (hopefully) how to solve the issue.

Troubleshooting is much more. It requires knowledge from different affected areas. Troubleshooting is a systematic approach to solve a problem with the goal to determine why something does not work as expected and explain how to resolve the problem. But the

starting point of troubleshooting is always describing the problem. It helps to find where to start your search for the cause with questions like:

- What are the symptoms?
- When and where does it occur?
- What are the conditions under which it occurs?
- Can it be reproduced?

Where to find Start-relevant Information

The start process is an especially important phase that is logged by the operating system, the SAP system, and the underlying database. If the SAP system does not start-up successfully, you can find the relevant error messages in log and trace files written on operating system level.



Note:

Not all mentioned log file types can be found in every work directory of instances (ASCS is different from PAS/AAS).



`/usr/sap/<SAPSID>/<instance_name><instance_number>/work/...`



Operating System

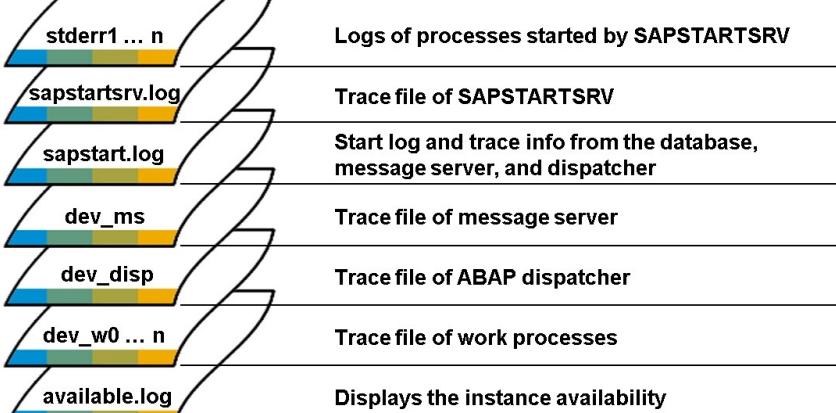


Figure 40: Log and Trace files in the Start Process

In the figure above, the most important location to find start-relevant information is propagated, the home directory `DIR_HOME`, in slang words also called “work directory”. Every SAP instance holds an own “work directory” and stores there the appropriate log and trace files. If problems occur during the starting procedure, these logs and traces provide useful information, such as warnings, error messages, error codes, or even problem descriptions in text form.

In addition, a kind of “Best Practice” can already be seen in the figure above (looking at the clock on the left):

When it comes to a problem in the context of a starting activity, access the “work directory” of the affected Instance and sort the files according to date and time stamp in descending order to get the most up-to-date files on top. This can already be an indicator of the problem origin.

STDERR<number> Files

The log files *STDERR<number>* are created by the instance-specific *sapstartsrv* daemon (on *UNIX/Linux* platforms) or service (on *Windows* platform) during the start of the instance. The instance processes started by the *sapstartsrv* write to the individual *STDERR<number>* files, depending on the sequence in which they are listed in the *Instance* profile. The content of these log files, therefore depends on the individual system setup.



Note:

The sequential number as part of the *STDERR<number>* files refers to the parameter *Start_Program_<2-digit_number>* in the *Instance* profile. When the file is created, the leading zero is hidden.

Here are some typical examples for *STDERR<number>* files:

- *STDERR<number>* - Information about the start process of the database system
- *STDERR<number>* -Information about the start process of the message server
- *STDERR<number>* - Information about the start process of the dispatcher



Hint:

A *STDERR<number>* file does not necessarily include errors. From a severity perspective It might include also warnings or even “only” information.

Log Files of *sapstart* and *sapstartsrv*

The *sapstartsrv.log* is the log file of the *sapstartsrv* process itself. It contains information about the *sapstartsrv* activities.

There are some differences on the operating system platforms (as already discussed in another lesson).

Only as reminder: On *UNIX/Linux* platforms, the *sapstartsrv* triggers the *sapstart* process to start the instance. The *sapstart* process writes consequently to a own *sapstart.log* file.

Trace Files of further important Processes and Services

All instance processes (and services) write to their own trace files, such as the *Message Server* to the file *dev_ms* or the *Internet Communication Manager* to the file *dev_icm*.

In case there are general problems in the initialization phase of the work processes, trace files with the naming convention *dev_w<number>* will include details. These files are also highly relevant for troubleshooting live, so not only for the start procedure. The work processes are used by the logged-on users in ABAP. If a work process fails or creates errors, you can identify the work process by its number and evaluate the trace file to figure-out the reason for the issue.

Configuration of the Log and Trace Level

You can set the granularity of the logged and traced information using the profile parameter *rdisp/TRACE*. The possible values for this parameter are:

- 0: No trace
- 1: Error messages

- 2: Full trace, whereas the trace entries depend on the SAP program that is being traced
- 3: As with 2, but data blocks are also traced

The trace level can be set separately for individual work processes in the process overview (transaction SM50). Alternatively the trace level can be increased or decreased using for example the SAP MC. This should be typically done for troubleshooting purposes and only on a temporary basis (see note below).



Note:

The higher the trace level, the larger the amount of logged information, and therefore the larger the size of the files. Specifically the increase of the trace level can cause further problems in the areas capacity and performance. So be careful and change the default trace level only explicitly and on a temporary basis for troubleshooting.

As a consequence: Use trace levels >1 only for short periods of time, for analyzing error situations.

Instance Availability

The *available.log* registers times during which the instance was not operational, for example when starting or stopping.

Problem Analysis Areas

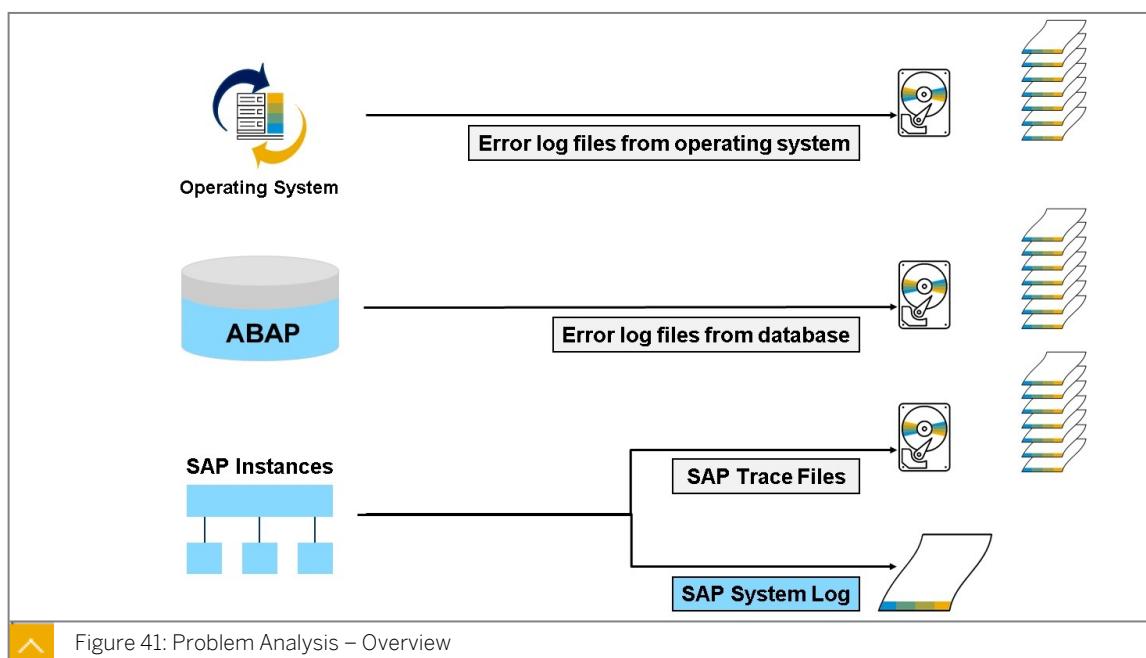


Figure 41: Problem Analysis – Overview

The SAP system can fail to start correctly for a number of reasons. If, for example, there may have been problems starting the underlying database, the SAP system - more precisely the PAS and/or AAS instances - could not subsequently be started. As shown in the figure above, the problem analysis can be done as follows:

- Check the error messages and warnings of the relevant operating system with the corresponding operating system tools.

- Check the status of the relevant database system using the error log files.
- Check the start logs in the SAP MC. To do so, select the affected instance, open the context menu, and choose *List Developer Traces*.
- Check the log files, *STDERR<number>* created by the *sapstartsrv*.
- Check the trace files of the individual instance processes and work processes:
 - *dev_ms*: Developer trace for the *Message Server*
 - *dev_rd*: Developer trace for the *RFC Gateway Service*
 - *dev_disp*: Developer trace for the *ABAP Dispatcher*
 - *dev_w<number>* (*<number>* denotes the number of the work process): Developer trace for the work processes

If you can still log on to the SAP system, check the system log of the SAP system using transaction **SM21**. The system log can be also displayed and analyzed with the *SAP MC* or *SAPControl*.

SAP MC: Access Log and Trace Files

The *SAP MC* provides access to important log and trace files via the context menu. This is a core functionality besides the visualization of the status information in the *Navigation Pane*.

Functions provided in the Navigation Pane at the left - based on <SAPSID> and SAP Instances

The following functions are provided for an SAP system and for any SAP instance in the context menu of the *Navigation Pane*:

Show Version Information

Lists important binaries, their version and time stamp.

Show Environment Information

Lists variables with the corresponding values.

Analyze Log Files

Merge of important log and trace files and provides an output with filter options like period to scan, severity of messages, instance name, file name, details and the text of the messages.

Additional Functions based on an explicit SAP instance

There are additional functions, which are only provided on the level of an SAP instance.

Show Profile Parameters

Show Profile Parameters: Displays a list of all profile parameters with their values like *RSPFPAR* in the ABAP SAP system

Show Start Profile

Displays the instance profile.

Show Trace File

Displays the *sapstartsrv.log* file.

List Developer Traces

Lists all log and trace files of the work directory with file name, size, and modification time. You can select one file at a time to display the content.

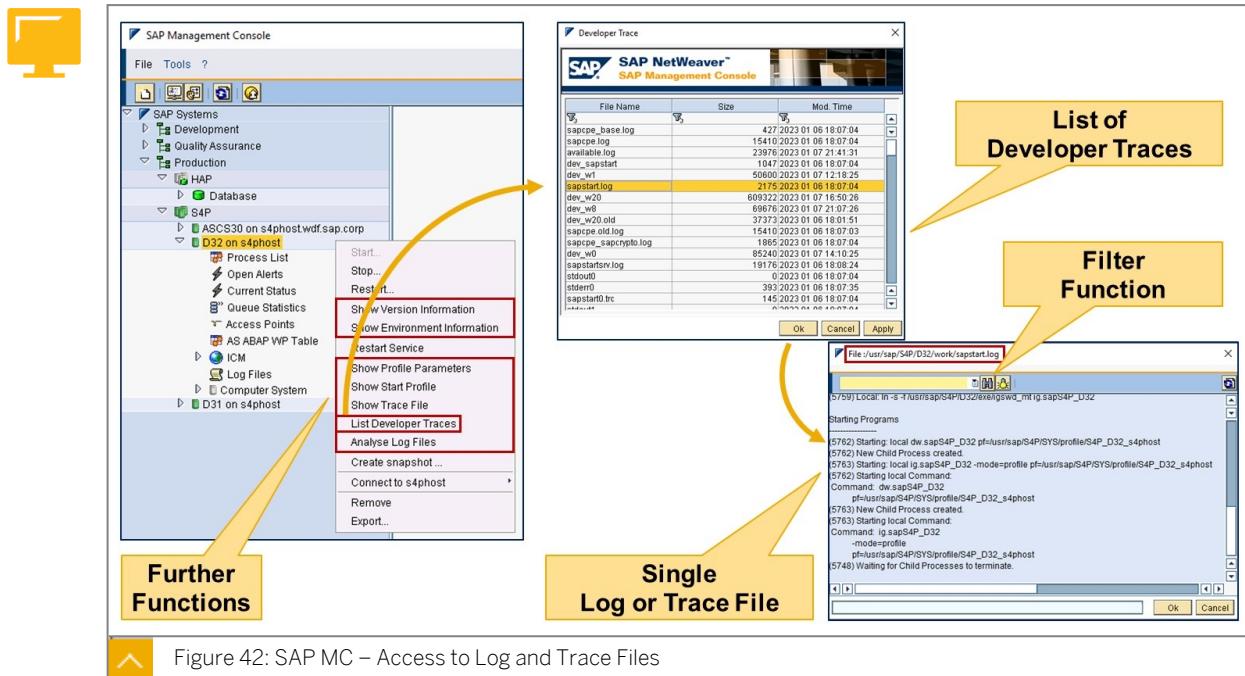


Figure 42: SAP MC – Access to Log and Trace Files

Functions provided in the Result Pane on the right - based on processes

The following functions are provided for processes of an SAP instance in the *Process List*. In the list of processes in the *Result Pane* use the context menu.

Process List: Show Developer Trace

Based on the process you flagged, the *Show Developer Trace* function opens the respective trace file (for example, of the ABAP dispatcher or the ICM process).

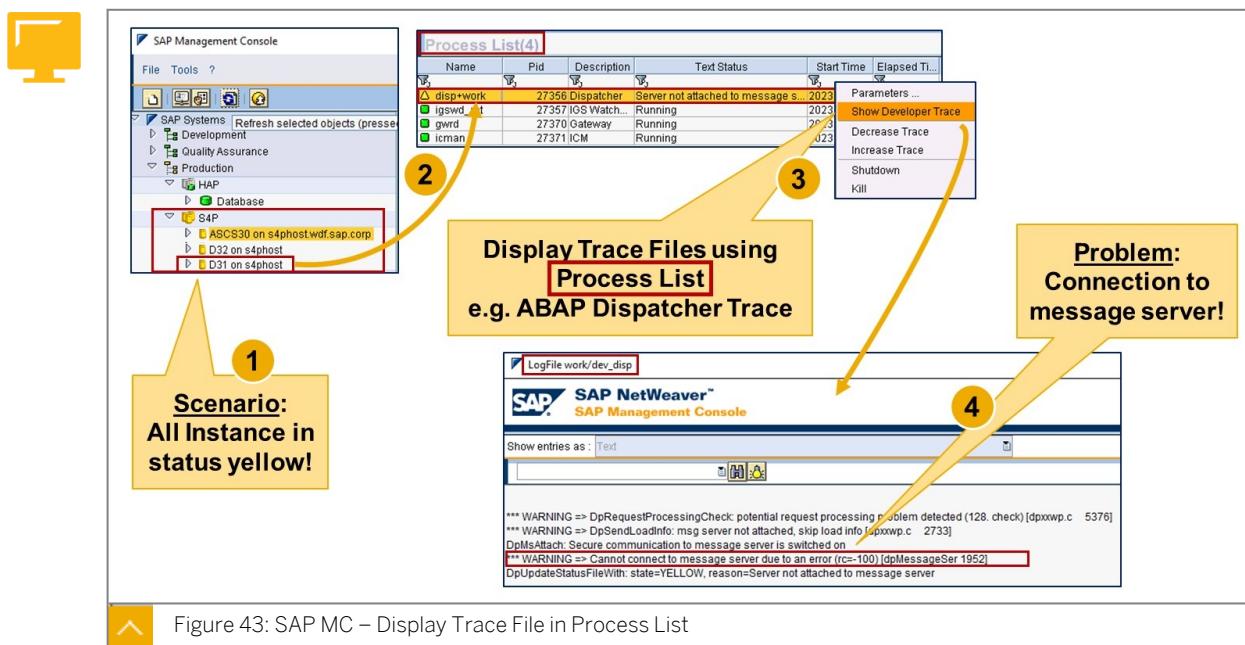


Figure 43: SAP MC – Display Trace File in Process List

AS ABAP WP Table: Show Developer Trace

Based on the work process, which you flagged, the *Show Developer Trace* function opens the trace file of the work process.

Log Files

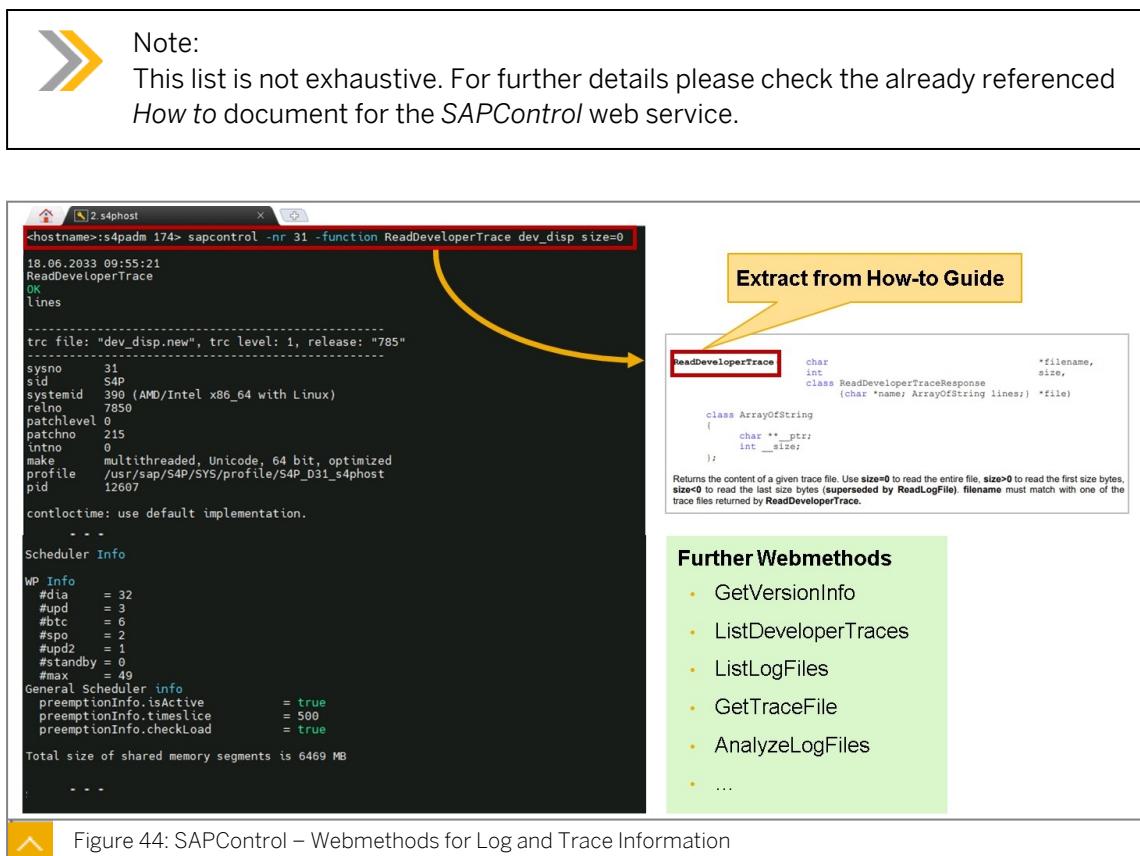
Select an Instance in the *Navigation Pane*, expand the node and select *Log Files*. The complete content of the work directory is shown in the *Result Pane*. Select a log or trace file and choose in the context menu the function *Show Log File* or double-click on the corresponding row to open the *Developer Trace*.

SAPControl: List Webmethods to Access Log and Trace Information

With SAPControl it is also possible to access log and trace information using specific Webmethods.

Every function provided by the SAP MC can also be executed with **SAPControl** on command line level. That is the reason, why the call of **sapcontrol --help** leads to such a comprehensive output.

Here are some examples for functions used to analyze start and stop issues:



GetVersionInfo

Returns a list of version information for the most important files of the instance.

ABAPGetWPTable

List output of the ABAP work processes like in SM50 or in the SAP MC

GetTraceFile

The content of the instance-specific Web service trace file `sapstartsrv.log` will be the output of this command.

ListDeveloperTraces

Output is a list of all instance trace files in DIR_HOME. Read a specific trace file with the next function *ReadDeveloperTrace*.

ReadDeveloperTrace

Returns the content of a given trace file with <filename> <filesize> (for example, with size=0 to read the entire file).

ListLogFile

This command returns a list of all instance log files.

AnalyseLogFile

This option scans all log files for a given time period and returns a merged list of all matching log file entries (see *How to use the SAPControl Web Service Interface* for details).

Using Snapshots

The *Snapshot* technology discussed here is based on the *Web Service of sapstartsrv* using *SAPControl*.



Hint:

Snapshots are always instance-specific, because every instance uses an own *sapstartsrv*. If you want to create *Snapshots* as a historical reference or a kind of “auditing” for parts of or complete SAP systems, scripts can be used to coordinate the time or event-based creation of the *Snapshots*. This has to done by the customer administrative department.

Snapshots can be used in SAP environments to collect important information about the current situation of a server as an automatism or manually. Automatically initialized snapshots dump all essential information into the developer trace (for example, *dev_disp* of the ABAP dispatcher) and create an archive including further related trace files, in order to have a kind of “package” to analyze the situation after it occurred.



Note:

Do not confuse this topic with *Storage Snapshot Technology*, which is often used in the context of backup technology.

Snapshots: Automatic Creation

The trigger to create *snapshots* automatically is given in the following scenarios (list is not complete):

- A hard shutdown occurs
- A deadlock situation occurs
- A resource bottleneck was detected
- A stop of request processing for more than 30 seconds occurs
- A crash occurs in a work process, ICM, or the RFC Gateway Service

In case a *snapshots* is triggered in an ABAP based SAP system, the processes write a special *snapshot section* inside the *developer traces*, which is needed to identify important meta data

within the traces, like “begin”, “end”, “<id>”, and “<reason>”. In between of the “begin” and “end” the respective process dumps what happened. in terms of

- The *ABAP dispatcher* dumps the request, the session, and the process which executed the request.
- The *ABAP work process* dumps the C-STACK, the ABAP stack, and details about critical sections.



Note:

Further details and background information for snapshots in the SAP system can be retrieved from the following space in the *Support Wiki*: <https://wiki.scn.sap.com/wiki/display/SI/Snapshots+in+the+SAP+System>

But it is also possible to create snapshots manually in different ways:

- In the SAP MC via the context menu of individual instances
- Using *SAPControl Webmethod CreateSnapshot*
- In the SAP system using transaction SNAPSHOTS

The screenshot shows the SAP System List of Snapshots interface. On the left, there's a sidebar with a monitor icon. The main area has a title bar "List of Snapshots" and "Snapshot of SAP System S4D". Below the title bar is a toolbar with "Refresh", "Download", "Display", "Delete", "Set Filter", "Sort in Ascending Order", and "More". A search bar is also present. The main content area is a table with columns: File Name, Description, File Size, Date, Time, and Server Name. The table lists several snapshots, with the first one checked. A yellow callout points to the "Reason for snapshot" column under the heading "Reason for snapshot". To the right of the table, a red callout points to a "Snapshot" tab in a modal window titled "Display Snapshot s4dhost_11_2_2022_11_28_05_57.zip". This window contains a tree view of snapshot details, with a yellow callout pointing to the "Status" node. Another yellow callout points to a "Detailed information" section at the bottom of the modal, which displays server status and last error information. A red callout points to the "Last Error Information" section.

Figure 45: Snapshots in the SAP System



Hint:

The feature to use *snapshots* is available since kernel release 7.40. The transaction is provided since SAP_BASIS release 7.40 SP7. In addition there are parameters to limit the creation of *snapshots*, both, from a perspective of content triggering the creation, and from a point of view of the amount of *snapshots* which can be created.

For details refer to the documentation of the following parameters:

- *service/max_snapshots*
- *rdisp/snapshot*



Note:

For sure, the scan of *snapshots* via a transaction code, like SNAPSHOTS, does not help to troubleshoot in a non-starting scenario of an SAP system, if any PAS and AAS instance is not starting successful and the administrator is not able to log-on to the ABAP stack.

Nevertheless, the *snapshots* are stored at a default location on operating system level, for example on *UNIX/Linux* the file path is: /usr/sap/<SAPSID>/SYS/global/sapcontrol/snapshots

The SAP MC can also be used to handle *snapshots*.

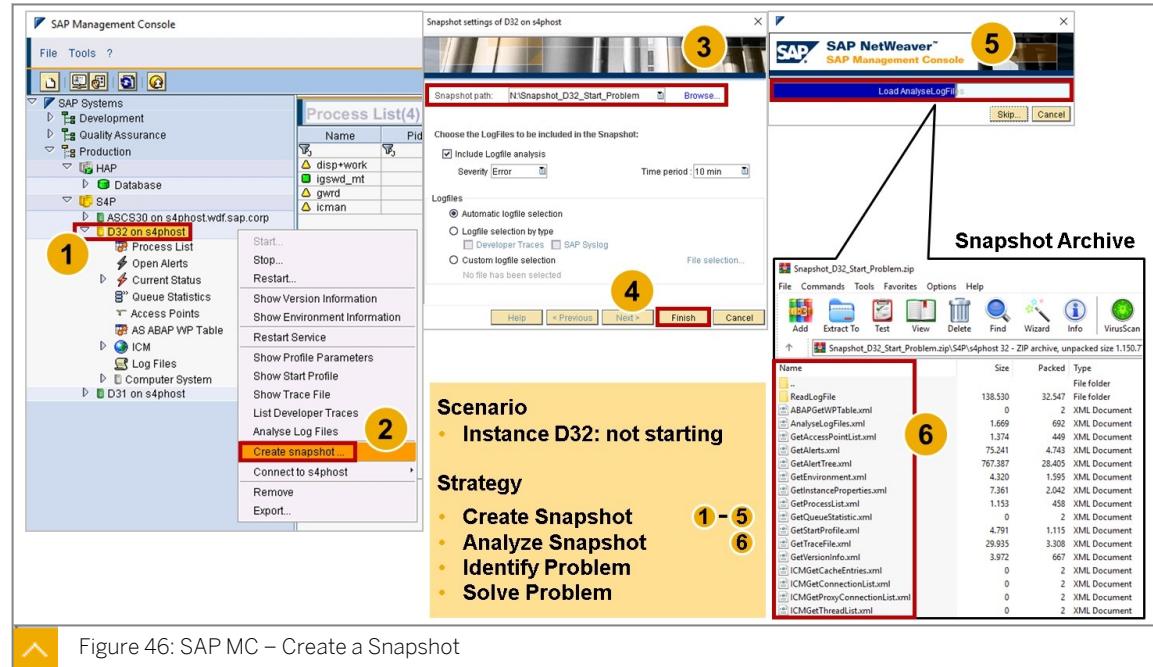


Figure 46: SAP MC – Create a Snapshot

**Note:**

Within the SAP MC it is possible to load already created *snapshots* and make them graphically visible, including the associated log and trace files. Use *File → Load Snapshot...* and browse to the location where the archive file is available. The *snapshot* will be displayed in the *Navigation Pane* under an explicit node *Snapshot*.

Finally *snapshots* can also be handled using *SAPControl* on command line level. There are some *Webmethods* for this purpose.



```

<hostname>:s4padm 198> sapcontrol -nr 32 -function ListSnapshots
18.06.2023 09:55:21
ListSnapshots
OK
filename, size, modtime, description
s4phost_32_12_2022_12_19_14_26_55.zip, 1255196, 2022 12 19 14:27:00, SNAPSHOT 1 (Reason: Hard Shutdown)
s4phost_31_12_2023_01_05_20_26_48.zip, 1612927, 2023 01 05 20:26:53, SNAPSHOT 1 (Reason: Hard Shutdown)
s4phost_32_0_2022_07_25_10_01_03.zip, 1265459, 2022 07 25 10:03:13, SNAPSHOT 182 (Reason: Workprocess 0 died)
s4phost_32_6_2022_07_25_10_31_03.zip, 1285197, 2022 07 25 10:33:14, SNAPSHOT 188 (Reason: Workprocess 0 died)
s4phost_31_18_2023_01_08_12_04_32.zip, 1237502, 2023 01 08 12:04:36,
s4phost_32_10_2022_10_01_12_03_56.zip, 1884335, 2022 10 01 12:05:02, SNAPSHOT 1 (Reason: Hard Shutdown)
s4phost_31_7_2022_11_09_19_31_14.zip, 1956196, 2022 11 09 19:31:20, SNAPSHOT 1 (Reason: Hard Shutdown)

<hostname>:s4padm 202> sapcontrol -nr 32 -function CreateSnapshot SnapITest_D32 2 2023 01 08 14:00:00 2023 01 08 14:30:00 10000
08.01.2023 15:24:28
CreateSnapshot
OK
filename: s4phost_32_15_2023_01_08_15_24_25.zip
<hostname>:s4padm 203

<hostname>:s4padm 204 sapcontrol -nr 32 -function ListSnapshots
08.01.2023 15:25:26
ListSnapshots
OK
filename, size, modtime, description
s4phost_32_15_2023_01_08_15_24_25.zip, 933276, 2023 01 08 15:24:28, SnapITest_D32
s4phost_31_12_2023_01_05_20_26_48.zip, 1612927, 2023 01 08 15:18:23, SNAPSHOT 1 (Reason: Hard Shutdown)

[<description> [<analyse_severity> [<analyse_starttime> YYYY MM DD HH:MM:SS] [<analyse_endtime> YYYY MM DD HH:MM:SS] [<maxentries>]]]

```

Figure 47: SAPControl – Display and Create Snapshots

CreateSnapshot

This command creates an instance *snapshot* and stores it in the system DIR_GLOBAL directory as a ZIP archive.

ReadSnapshot

To make use of this *Webmethod*, the local file name has to be specified. It then returns the binary ZIP archive content.

ListSnapshots

Checks for the available *snapshots*.

DeleteSnapshots

Deletes a given list of snapshots in the DIR_GLOBAL file system.

Using the Support Log Assistant

The *Support Log Assistant* is a *Self Service Tool* provided as application by the SAP Business Technology Platform that allows users to scan their own support related files for known issues and recommendations.

In the background there is a kind of Q&A database against which the customer-specific log and trace files are checked.

Since autumn 2019, the tool has been generally released by SAP for customers and supports since June 2021 six different languages with machine translation technology.

Currently there is already a wide range of supported file formats. From a security perspective it is important to mention that the files are not transferred to SAP.

In case you are interested in this topic, feel free to use the corresponding wiki page under the following link. Here you will find also an FAQ area:

To get an idea about the look and feel of the tool, check the following figures.

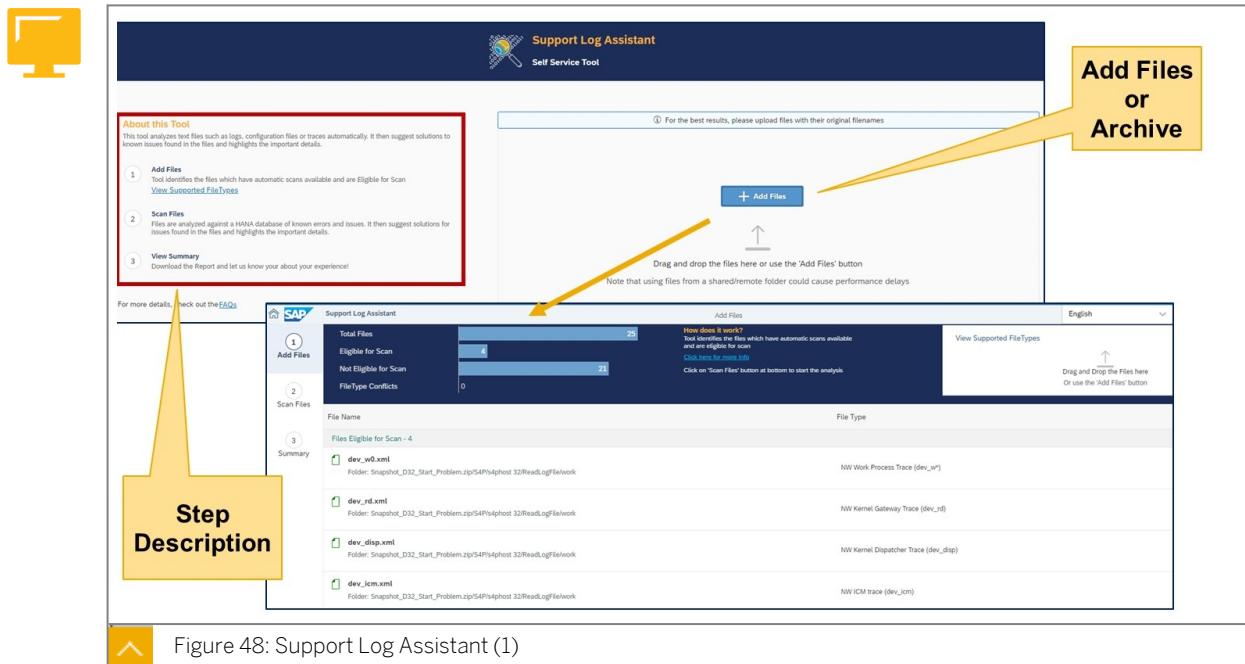


Figure 48: Support Log Assistant (1)

Support Log Assistant Wiki

<https://wiki.scn.sap.com/wiki/display/ATopics/Support+Log+Assistant>

Support Log Assistant - Link

<https://supportportal-pslogassistant-app.cfapps.eu10.hana.ondemand.com/>

Using the Support Log Assistant to automate support-related file analysis

<https://me.sap.com/notes/2838708>

Release Notes

https://support.sap.com/content/dam/support/en_us/library/ssp/my-support/releasenotes/2019/release-notes-wave6-2019.pdf

File Types supported

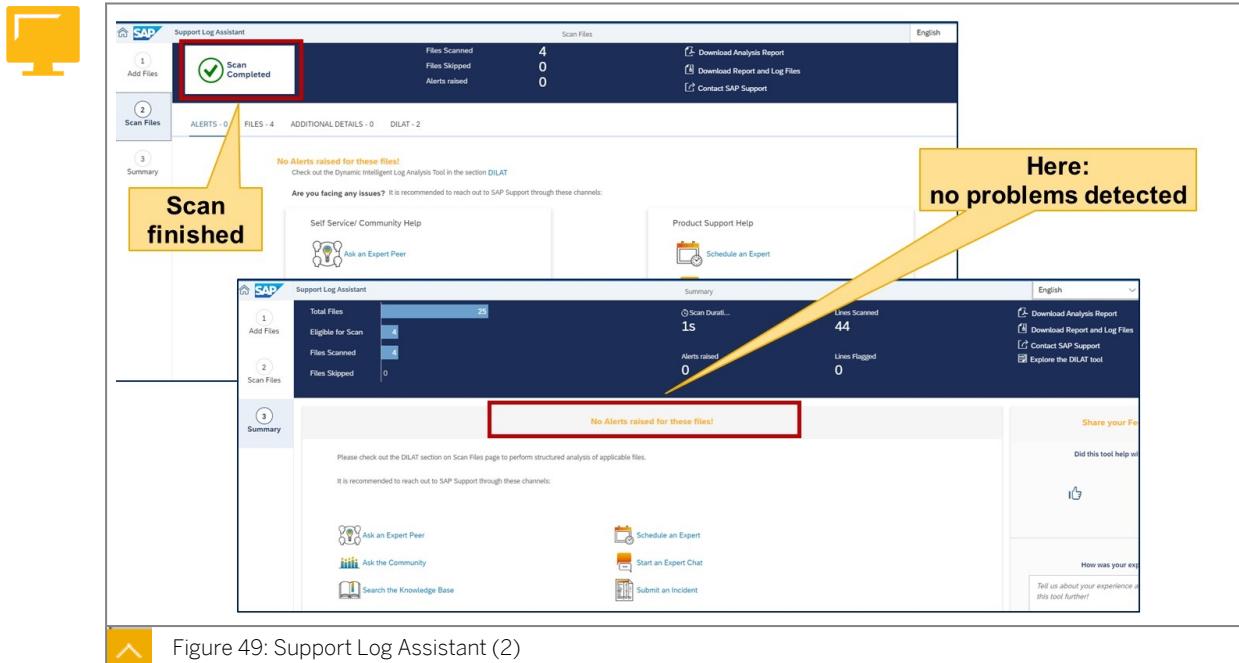
<https://supportportal-pslogassistant-app.cfapps.eu10.hana.ondemand.com/utilities/FileTypeList/index.html>

Expert Level File Analysis using the Support Log Assistant

<https://blogs.sap.com/2020/11/16/expert-level-file-analysis-using-the-support-log-assistant/>

Support Log Assistant 2.0 – Self Service Tool for Analyzing issues

<https://blogs.sap.com/2020/11/09/support-log-assistant-2.0-self-service-tool-for-analyzing-issues/>



LESSON SUMMARY

You should now be able to:

- List log and trace information for troubleshooting
- Use SAP MC and SAPControl to access log and trace information
- Appendix: further analysis options

Learning Assessment

1. The profile directory of an ABAP-based SAP system on operating system level can be accessed by any instance with an assigned 2-digit number, which is part of this SAP system.

Determine whether this statement is true or false.

- True
- False

2. The ABAP Central Services (ASCS) instance can be started successfully, even if the database of the SAP system is not yet up-and-running.

Determine whether this statement is true or false.

- True
- False

3. Which of the following instances does **not** have an assigned enqueue server or enqueue process?

Choose the correct answer.

- A Instance with priority LEVEL 1
- B Instance with priority LEVEL 2
- C Instance with priority LEVEL 3

4. Every sapstartsrv of one ABAP-based SAP system reads every existing instance profile of the respective SAP system

Determine whether this statement is true or false.

- True
- False

5. The stop action of an ABAP-based SAP system is performed in the following way regarding *priorities*:

Choose the correct answer.

- A The system stop action is done in the same order as the system start action.
- B The system stop action is done in the reverse order in comparison to the system start action.
- C The system stop action is done in any order. The order to stop the instances does not matter.

6. Which of the following tasks is important before stopping an ABAP-based SAP system?

Choose the correct answer.

- A Perform a database backup
- B Check finished jobs
- C Inform users beforehand and check user sessions right before stopping

7. Which of the following statements are true regarding the usage and the structure of the *SAP Management Console* (SAP MC)?

Choose the correct answers.

- A The SAP MC has to be installed explicitly by the customer
- B The structure of the SAP MC can be adjusted
- C The content of the SAP MC can be defined by the customer individually
- D The SAP MC is operating system independent

8. Which of the following functions is provided in the context menu of an <SAPSID> in the *Navigation Pane* of the SAP MC?

Choose the correct answer.

- A Reinstall...
- B Restart...
- C Reinitialize...

9. SAPControl provides features and functions for starting and stopping. Flag the correct answers regarding the usage of this tool in combination with ABAP-based SAP systems.

Choose the correct answers.

- A With SAPControl complete ABAP-based SAP systems can be started, without pushing every instance explicitly.
- B SAPControl uses webmethods to execute functions.
- C SAPControl always needs the specification of user and password to execute a function.
- D SAPControl can be used interactively on command line level.

10. Identify the directory in which you can find many relevant logs related to the startup of an SAP system (identify the directory structure that recently installed SAP systems will use)?

Choose the correct answer.

- A usr – sap – <SAPSID> – D<no.> – log
- B usr – sap – <SAPSID> – D<no.> – data
- C usr – sap – <SAPSID> – SYS – global
- D usr – sap – <SAPSID> – D<no.> – work

11. Which of the following areas - beside the sap log and trace files - might most probably provide further useful detailed log and trace information, in case the ASCS instance of your ABAP-based SAP system started successful, but the PAS and AAS instances did not come up?

Choose the correct answer.

- A The syslog of the operating system where the ASCS instance is installed.
- B Error log files from the underlying database.
- C The log file of the SAP GUI.

12. Snapshots provide useful log and trace information in the format of an archive. Which tools can be used to access these archives?

Choose the correct answers.

- A SAPControl on command line level.
- B Transaction SNAPSHOTS, for example, using SAP GUI.
- C Operating system syslog.
- D SAP Management Console (SAP MC)

Learning Assessment - Answers

1. The profile directory of an ABAP-based SAP system on operating system level can be accessed by any instance with an assigned 2-digit number, which is part of this SAP system.

Determine whether this statement is true or false.

True

False

You are correct! The profile directory can be accessed by any instance with an assigned 2-digit number of one and the same SAP system on operating system level and is located underneath the <SAPSID> and then the SYS directory.

2. The ABAP Central Services (ASCS) instance can be started successfully, even if the database of the SAP system is not yet up-and-running.

Determine whether this statement is true or false.

True

False

You are correct! The ASCS instance can be started successfully even if the database of the SAP system was not yet started.

3. Which of the following instances does **not** have an assigned enqueue server or enqueue process?

Choose the correct answer.

A Instance with *priority LEVEL 1*

B Instance with *priority LEVEL 2*

C Instance with *priority LEVEL 3*

Instances with *priority LEVEL 1* and *priority LEVEL 2* have an assigned enqueue server or enqueue process. Only an instance with *priority LEVEL 3* does not have an assigned enqueue server or process.

4. Every *sapstartsrv* of one ABAP-based SAP system reads every existing instance profile of the respective SAP system

Determine whether this statement is true or false.

True

False

The *sapstartsrv* is instance-specific and reads only the instance profile of the own instance. In addition it reads also the default profile.

5. The stop action of an ABAP-based SAP system is performed in the following way regarding *priorities*:

Choose the correct answer.

A The system stop action is done in the same order as the system start action.

B The system stop action is done in the reverse order in comparison to the system start action.

C The system stop action is done in any order. The order to stop the instances does not matter.

When stopping an ABAP-based SAP system, the reverse start order is used. Instances with a high number (*low priority*) are stopped first. Then, successively the other instances are stopped.

6. Which of the following tasks is important before stopping an ABAP-based SAP system?

Choose the correct answer.

A Perform a database backup

B Check finished jobs

C Inform users beforehand and check user sessions right before stopping

Users have to be informed about the shutdown beforehand, for example by creating a *system message* via transaction SM02. In addition it should be verified directly before the shutdown, whether users are still logged on. It is not necessary to check for finished jobs, but for jobs in status active or scheduled. And even the execution of a database backup is not needed.

7. Which of the following statements are true regarding the usage and the structure of the SAP Management Console (SAP MC)?

Choose the correct answers.

- A The SAP MC has to be installed explicitly by the customer
- B The structure of the SAP MC can be adjusted
- C The content of the SAP MC can be defined by the customer individually
- D The SAP MC is operating system independent

There is no need to install the SAP MC. It comes with the SAP kernel. Customers are able to define the content (SAP systems and instances), but the structure cannot be maintained. The software of the SAP MC is a Java application and operating system independent.

8. Which of the following functions is provided in the context menu of an <SAPSID> in the Navigation Pane of the SAP MC?

Choose the correct answer.

- A Reinstall...
- B Restart...
- C Reinitialize...

Based on an <SAPSID>, the function provided in the context menu is *Restart....* Depending on the current status of the SAP system, also other functions are provided. For example *Start...* or *Stop....* There are no functions like reinstall or reinitialize. Only a *Remove* of the <SAPSID> from the SAP MC is provided.

9. SAPControl provides features and functions for starting and stopping. Flag the correct answers regarding the usage of this tool in combination with ABAP-based SAP systems.

Choose the correct answers.

- A With SAPControl complete ABAP-based SAP systems can be started, without pushing every instance explicitly.
- B SAPControl uses webmethods to execute functions.
- C SAPControl always needs the specification of user and password to execute a function.
- D SAPControl can be used interactively on command line level.

SAPControl is able to start complete SAP systems with only one command as a *webmethod*. These are functions used by the tool. Depending on the operating system user used to execute *webmethods* and depending on the configuration, the authentication is done without providing user and password.

10. Identify the directory in which you can find many relevant logs related to the startup of an SAP system (identify the directory structure that recently installed SAP systems will use)?

Choose the correct answer.

- A usr – sap – <SAPSID> – D<no.> – log
- B usr – sap – <SAPSID> – D<no.> – data
- C usr – sap – <SAPSID> – SYS – global
- D usr – sap – <SAPSID> – D<no.> – work

You are correct! The *work* directory holds the most important log files related to the startup of an SAP system. The other listed directories store other information.

11. Which of the following areas - beside the sap log and trace files - might most probably provide further useful detailed log and trace information, in case the ASCS instance of your ABAP-based SAP system started successful, but the PAS and AAS instances did not come up?

Choose the correct answer.

- A The syslog of the operating system where the ASCS instance is installed.
- B Error log files from the underlying database.
- C The log file of the SAP GUI.

If the database-independent ASCS instance was started without problems, but the PAS/AAS instances not, there is a high probability that the underlying database is not operational. But there can be also lots of others reasons for this problematic.

12. Snapshots provide useful log and trace information in the format of an archive. Which tools can be used to access these archives?

Choose the correct answers.

- A SAPControl on command line level.
- B Transaction SNAPSHOTS, for example, using SAP GUI.
- C Operating system syslog.
- D SAP Management Console (SAP MC)

To access or even create or delete snapshots, multiple tools can be used. The SAP MC, SAPControl on command line level, and transaction SNAPSHOTS using for example SAP GUI. Nevertheless, the operating system syslog cannot handle snapshots.

Lesson 1

Configuring SAP Systems via Profile Parameters

101

Lesson 2

Setting Up Operation Modes for Resource Configuration

111

Lesson 3

Executing ABAP Task Lists for Automated Configuration

125

UNIT OBJECTIVES

- Describe profiles used for system configuration
- Describe the evaluation sequence of profile parameters
- Display profile parameters and their values
- Maintain profile parameters
- Monitor configuration changes
- Describe the concept of operation modes
- Describe dynamic and reserved work processes
- Set up operation modes
- Switch operation modes
- Outline the meaning of reserved and dynamic work processes
- Explain the options to change the work processes distribution using operation mode switches
- Describe the usage of consistency checks
- Describe ABAP task manager for lifecycle management automation
- Explain ABAP task lists
- Apply ABAP task lists

Configuring SAP Systems via Profile Parameters



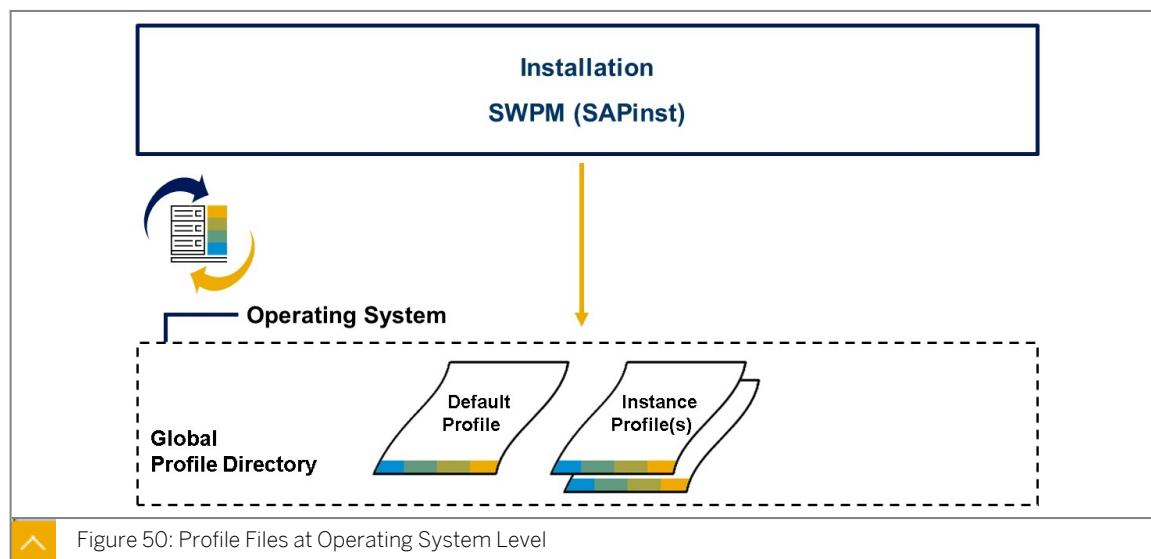
LESSON OBJECTIVES

After completing this lesson, you will be able to:

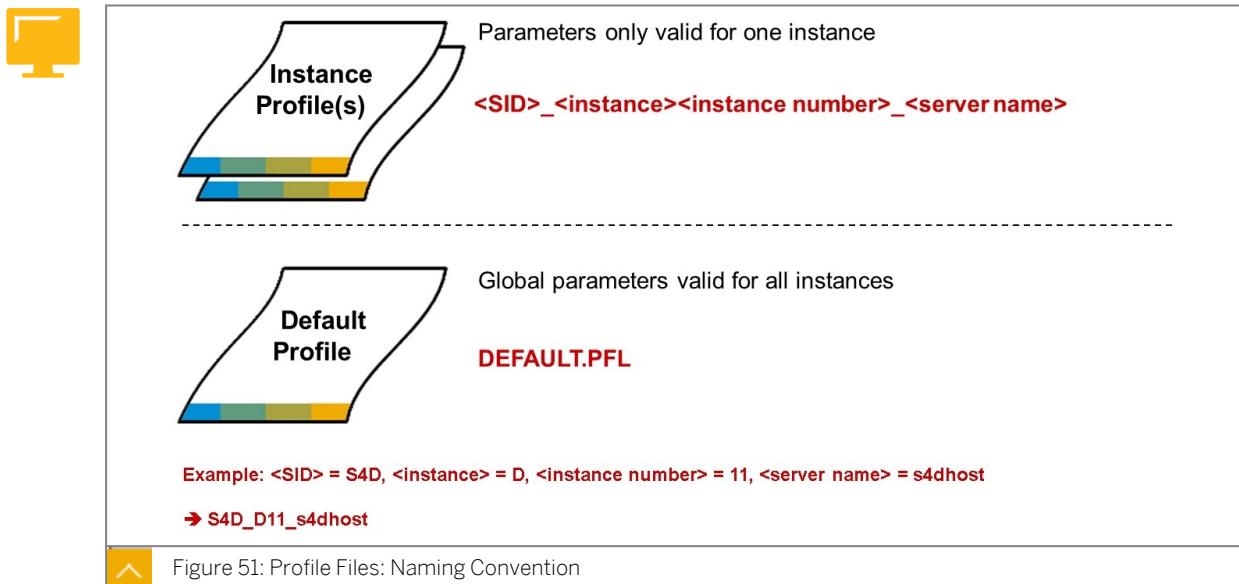
- Describe profiles used for system configuration
- Describe the evaluation sequence of profile parameters
- Display profile parameters and their values
- Maintain profile parameters
- Monitor configuration changes

Profile Files Used for System Configuration

When an SAP system is being installed, at least two profile files are created automatically. One is the default profile with profile parameters that are valid system-wide. The other one is an instance profile created for each instance with which profile parameter values can be configured for a specific instance.



As the figure *Profile Files at Operating System Level* shows, after the installation is complete, the profile files are stored at the operating system level in the directory: `/usr/sap/<SID>/SYS/profile`. This directory can be read by all instances of an SAP system using the share or mount technique.



The figure *Profile Files: Naming Convention* displays the naming conventions for the available profile files.

There is only one default profile (*DEFAULT.PFL*) for each SAP system and it is read by all instances during the startup procedure. It contains system-wide settings, such as the system name, the name of the database server, the name of the enqueue server, or the default logon client, and so on.

The instance profile (<SID>_<instance><instance number>_<server name>) defines parameters that apply for one instance, such as the number and type of work processes, or the definition of the size and allocation of memory areas used by the SAP instance. The content of the instance profile is instance-specific. The instance profile also contains the entries for the start up procedure.

The profile files are, in fact, created during the installation of the system and have some parameter already set. However, they can be edited afterward.

Evaluation of Profile Parameters

SAP systems consist of one central services instance (ASCS instance) and one or more application server instances. The SAP system and its instances are configured by profile parameters that are stored in profile files.

Initially, the default values of the profile parameters are defined and delivered in the program code of the SAP Kernel. That is why you need to differentiate between values coming from the SAP Kernel, referred to as kernel default, values set in the default profile, and in instance profiles. During the start up procedure of the SAP system, there is certain sequence of how the parameter values are evaluated and which value is valid at the end.

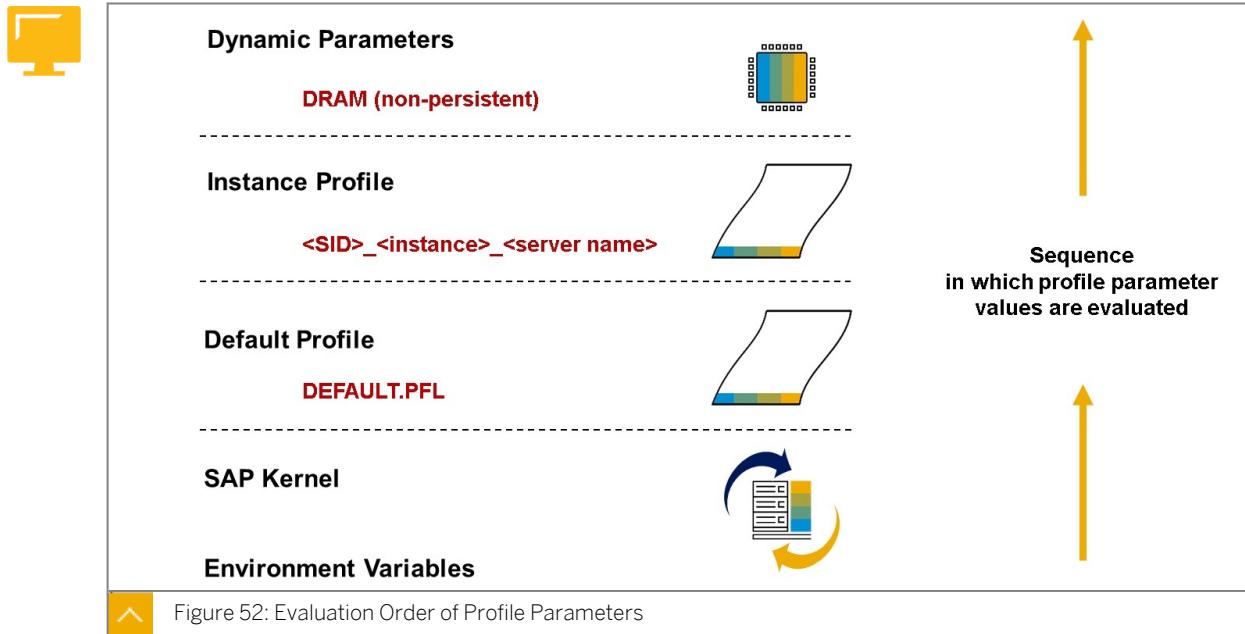


Figure 52: Evaluation Order of Profile Parameters

You can change the kernel default values using the default profile file and the instance profile files. These are read when an instance is started. The figure *Evaluation Order of Profile Parameters* shows the sequence in which the values for the profile parameters are evaluated by each instance of your AS ABAP.

First, the environment variables and the kernel default values are read. After that, the system reads the profile parameters that have been set in the default profile. So, if the value of a parameter in the default profile differs from the kernel default value, the value in the default profile is valid for all instances of the system. Only after that the instance profiles are read. So, if the same profile parameter has been explicitly set in the instance profile, this value is valid for this specific instance.



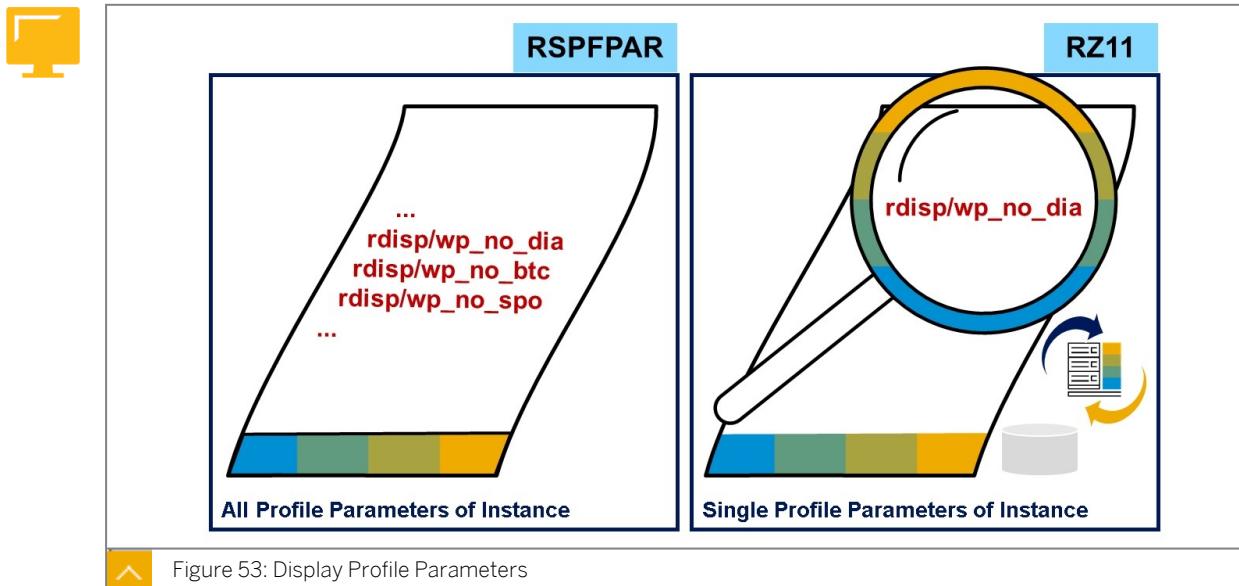
Note:

Changes to profile parameters require a restart of the affected instance or of the complete SAP system depending on in which profile the parameter has been set.

Certain profile parameters can be switched dynamically without a restart required.

Tools for Displaying Profile Parameters

Displaying Profile Parameters Inside the SAP System



The current values of the profile parameters can be displayed within the SAP system. There are two ways of doing this:

- You can use transaction `RSPFPAR`, which by the way, can also be run as a report with the same name.
- You can use transaction `RZ11`.

Both functions display the profile parameters for the instance to which the user is currently logged on.

Transaction `RSPFPAR` displays a list of **all** instance-specific profile parameters in a table view, including system-wide parameters. You can restrict this list to specific parameters. The table displays the system default value for the individual profile parameters as it is defined in the program code of the kernel or in the default profile. If the value of a profile parameter is set in the instance profile or if it has been overridden dynamically, the table also displays the user-defined (temporary) value. A short description and, if required, documentation of the parameters can also be displayed.

Transaction `RZ11` displays information and documentation for **individual** profile parameters. It also shows whether the parameter is a dynamic parameter and can be changed while the system is running so a restart of the system is not required.



Note:

If you want to display an overview of all dynamically switchable parameters, you can use the following method: call transaction `RZ11`, then choose *Goto → All Dynamic Parameters*.

Values of profile parameters can contain set values or variables, such as parameter references or formulas.

Parameter references are indicated by a single \$ symbol and the referenced parameter in round brackets. Double \$\$ symbol indicates the instance number.

As of SAP Basis 7.40 SP02 and higher, parameter values can be specified as formulas. A formula is always surrounded by round brackets (). The formula can reference other parameters and use standard operators such as +,-,*,/,ⁿ,(,). A formula can also include several standard functions such as min, max, sin, cos, tan, sqrt, exp, log2, log10, ceil, and floor. Use a period “.” for decimal places. The individual values for the formula are displayed in transaction RSPFPAR. When long formulas are used, however, the formulas themselves cannot be displayed completely in this transaction. To display long formulas in their entirety, use transaction RZ11.

For more information, check the official documentation: https://help.sap.com/docs/ABAP_PLATFORM_NEW/e067931e0b0a4b2089f4db327879cd55/e05f0c5000efc06fe10000000a423f68.html

Example

The profile parameter *rdisp/myname* is set to \$(SAPLOCALHOST)_\${(SAPSYSTEMNAME)}_\$\$ in the kernel. It references two other profile parameters, which is indicated by a single \$ symbol. Double \$\$ symbol references the instance number.

If the two parameters are set, for example, as follows: SAPLOCALHOST=s4ghost, SAPSYSTEMNAME=S4Q, and the instance number is 12, the parameter *rdisp/myname* is derived from those values and is set to s4ghost_S4Q_12.

Error messages from formulas are logged in the *stderr* files during the start process. We recommend examining the *stderr0* file first.



Hint:

When evaluating profile parameters using the tools described, some profile parameter values are used system-wide, while others can be set differently for each instance. Transaction RSPFPAR displays the configuration of the instance on which you are running the report.

Displaying Profile Parameters Outside the SAP System

You can also display the values of the profile parameters outside the SAP system. This is done at the operating system level with the user <sid>adm by executing the program **sappfpar**. The **sappfpar** program determines the instance where the call is to be performed by evaluating the **pf=<instance profile>** option. Here, **<instance profile>** is the full path and file name.

If you want to display parameters interactively, call **sappfpar** with the profile option: **sappfpar pf=<instance profile>**. It will return a list of all parameters. When you enter a specific parameter, its current value is displayed.

You can check the parameters that are set with: **sappfpar check pf=<instance profile>**.

Calling **sappfpar** without any options displays the help function of this program.

Profile Administration and Maintenance

To not only display, but also set the profile parameter values, you need to edit profiles. You can do this by using transaction RZ10. Also, you can edit the profile files with operating system tools or editors.



Caution:

In this case, however, you must ensure that the changes are performed and documented correctly. Parameters that were set incorrectly can prevent an instance from starting. Before you make changes to the profile files, create backups. If an error occurs, you can start the SAP system with the restored files and perform troubleshooting.

It is, however, recommended to use RZ10 for profile administration and parameter maintenance. It is safer and more convenient.

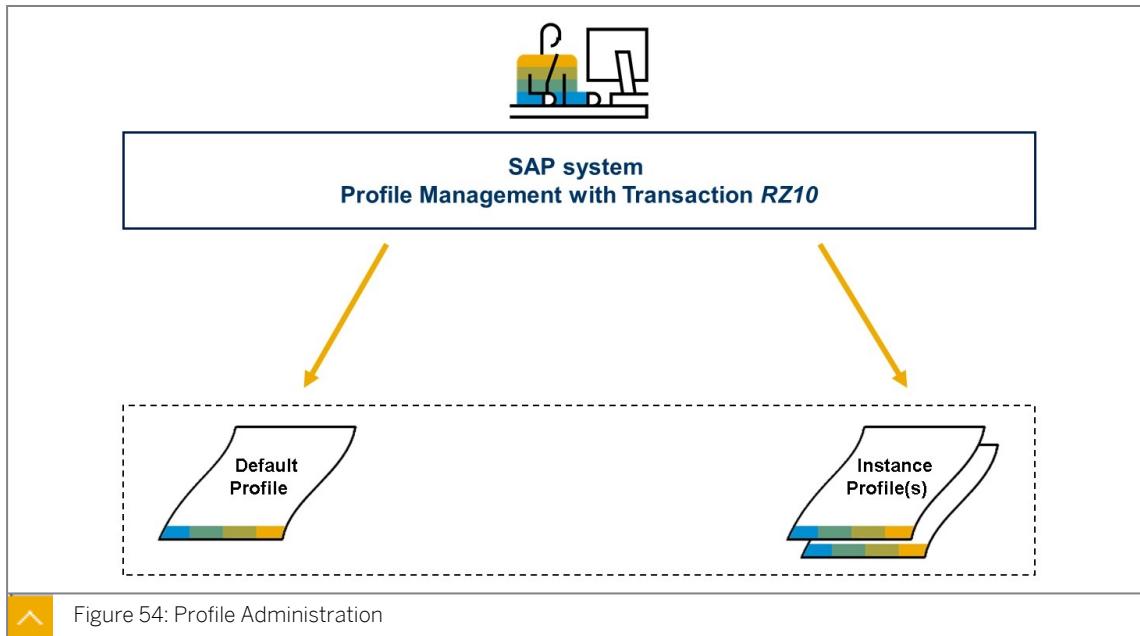


Figure 54: Profile Administration

As shown in figure *Profile Administration*, in RZ10 you can edit profiles that are created during the installation of the SAP system.

RZ10 offers a number of advantages to the user:

- Central administration and maintenance of the profile files
- Consistency checks when changing profile parameters
- Administration of multiple versions of a profile file
- Comparison between the actively used profile and the profile stored in the database

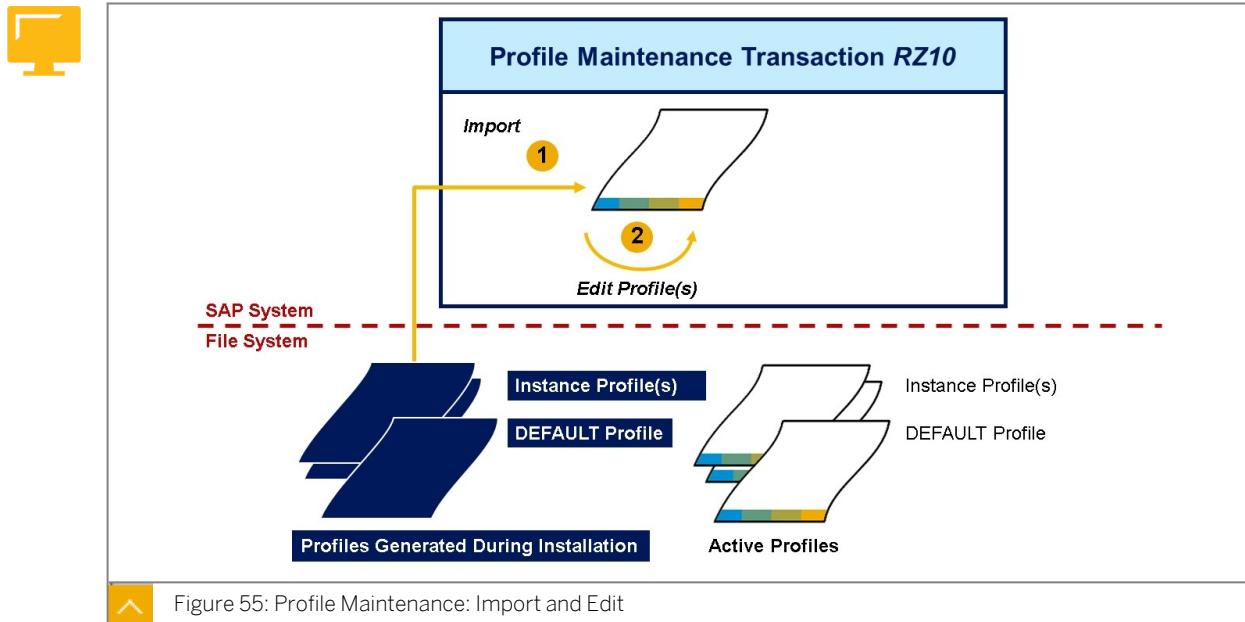


Figure 55: Profile Maintenance: Import and Edit

Immediately after installation, the profile files are created in the *profile* directory in the file system. They exist only at the operating system level. The installation program does not save profile files directly in the database. However, to administer the profile files in RZ10, they must be imported into the database, as shown in the figure *Profile Maintenance: Import and Edit*.

1. Importing Profile File Import After Installation

After installing the SAP system, profile files exist only at the operation system level. In transaction RZ10, you have to manually import the profile files from the file directory of the operating system into the database.

In the first step, you import the default profile and the instance profiles of the ASCS, PAS, and AAS into the database. To do so, choose *Utilities* → *Import Profiles* → *Of active servers* in RZ10.

During the import, the system performs an import check and compares the versions stored at the operating system level and in the database. After the check, the check log with the import results is displayed.

2. Editing Profile Files

After the profile file have been imported to the database, they can be edited. Changes to profile parameters can then be performed in the SAP system with the advantages described previously.

You can edit the imported profiles by changing the parameter values, adding new profile parameters, or deleting profile parameters. Changes are made in the database.

To edit a profile file, you have to select it, so you can change individual profile parameters. There are three different levels for maintaining the profiles. The *Administrative Data* contains the type of profile (default, or instance profile), a short description, the path of the file, the name of the instance and the time of the last activation. In *Basic maintenance* and *Extended maintenance*, you can maintain the parameters.

Basic maintenance allows you to adjust the most important parameters and supports you through the use of logical descriptions.

Extended maintenance displays the unformatted content of the profile; that is, the technical names of the profile parameters. In *Extended maintenance*, you can not only change the

values of individual parameters but also add new parameters and delete existing parameters. Usually, the administrator uses *Extended maintenance* rather than *Basic maintenance*.



Hint:

If you have installed a new application server instance or you have changed a profile file at operating system level, you can import profile files for individual instances. To do so, choose *Profile → Import* or the *Import* button.

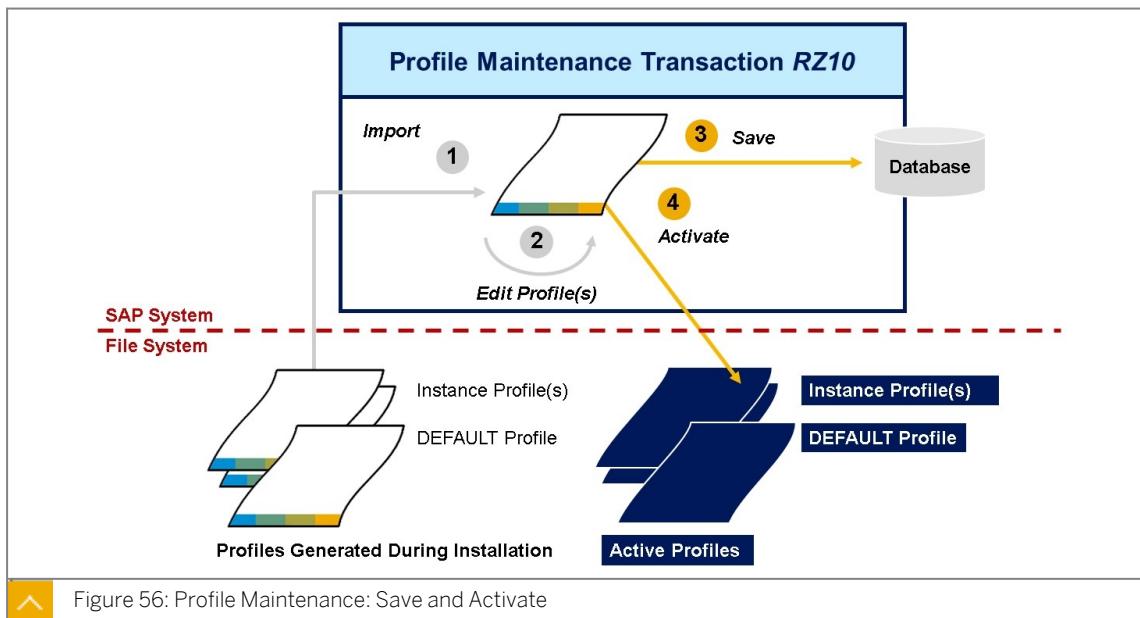


Figure 56: Profile Maintenance: Save and Activate

3. Saving Changes to the Database

As the figure *Profile Maintenance: Save and Activate* shows, in the third step, you have to save the changed values permanently. To do so, choose *Profile → Save* in RZ10. The changes are therefore successfully saved to the database.

4. Activating the Changed Profile Files

However, they need to be written back to the operating system level, meaning to the profile files in the file system. The changes only take effect when they are read by the system out of the profile files stored in the file directory; that is, at a restart of the system.

Thus, in the fourth step, the changed files in the database must be activated. To do so, choose *Profile → Activate* in RZ10.



Hint:

Activation can be done either automatically during saving by confirming the relevant query, or manually from the menu.

Changes to instance-specific profiles take effect after a restart of the corresponding instance.

Changes to the default profile take effect after a restart of all the instances (entire system).



Note:

Changes in the default profile are not propagated throughout the system until all instances of the system are restarted.



Hint:

The *profile* directory of the operating system also stores all old versions of the profile files, starting with *.1*. In earlier releases, there was only one older version of each profile, with the file extension *.BAK*.

Profile Files Checks

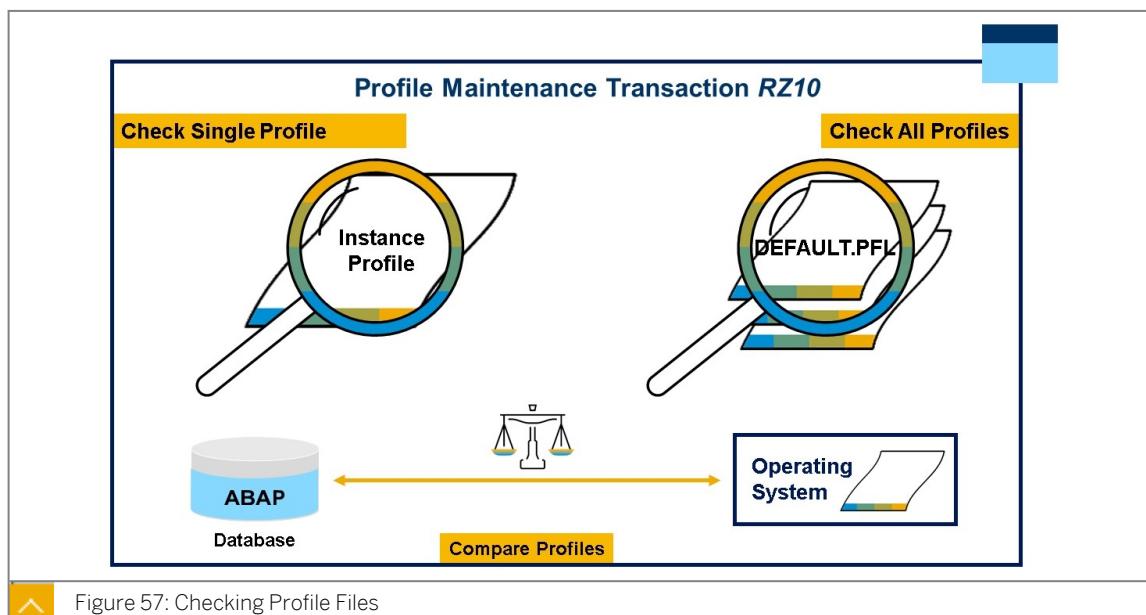


Figure 57: Checking Profile Files

In transaction RZ10, you can run several checks. For example, if a profile has been changed, you can perform a check for it. You can also perform a consistency check, to check the syntax, the semantics, and the parameter names of individual profiles or for all profiles. The results are displayed in a check log.

Check single profiles

By selecting a profile and choosing *Profile → Check*, you can check a single profile. This check detects errors in parameter values. You can execute it before activating a new version of the profile to avoid any configuration errors.

Check all profiles of active servers

The *Utilities → Check all profiles → Of active servers* option provides a consistency check of all profiles of running instances and the default profile. The system checks, for example, the unknown profile parameters or if existing formulas have been overwritten.

For more details on the checks and the check rules, please read the official documentation: https://help.sap.com/docs/ABAP_PLATFORM_NEW/e067931e0b0a4b2089f4db327879cd55/48c6090d895a154ee10000000a421937.html

Compare profiles

During the comparison of profiles, the system compares the active version of the profile at the operating system level with the most recent version of the profile stored in the database. To execute the comparison, select the profile file and choose *Profile* → *Comparisons* → *Profile in database* → *With active profile*. The system displays a check log that shows if the profile in the database is different than in the file system.

Monitor Parameter Changes

In transaction TU02, you can display an overview of parameter changes.

In the transaction, you get an overview of when profile parameters have been changed for the last time and for which application server instances. Also, you can select a time period to see which individual parameters have been changed on specific dates. You can also display a list of all active parameters and their values valid for a specific instance.

The report `RSPFL_HISTORY_DISPLAY` provides you the changelog of profile parameter values as well as the parameter snapshots.

In the changelog you can display which parameters have been changed for an instance, and you can search for a specific parameter to see the changed value and when the parameter has been changed last. You are also able to see the change history and all the timestamps. This data is stored in the database.



Note:

If you do not see the most recent date, you have to choose *Update Changelog* in the application toolbar to refresh the data.

The parameter snapshot files are created by each application server instance during the startup procedure and stored in the global directory. The data that is displayed in the changelog is based on the snapshot files.



LESSON SUMMARY

You should now be able to:

- Describe profiles used for system configuration
- Describe the evaluation sequence of profile parameters
- Display profile parameters and their values
- Maintain profile parameters
- Monitor configuration changes

Setting Up Operation Modes for Resource Configuration

LESSON OVERVIEW

In this lesson, you will learn about the concept of operation modes. You will learn how to set up a dynamic switch of work process types using operation modes.

Business Example

The configuration of the work processes using the profiles usually meets the requirements for day operation. The administrator can use operation modes to optimize the performance of the system for the different requirements during the day and during the night.

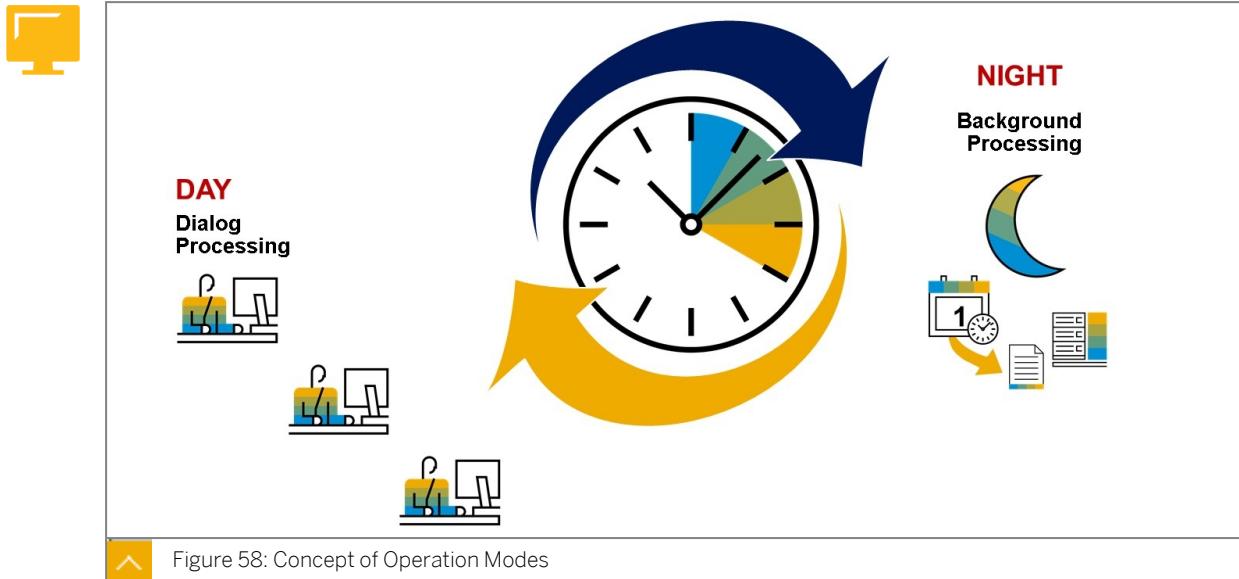


LESSON OBJECTIVES

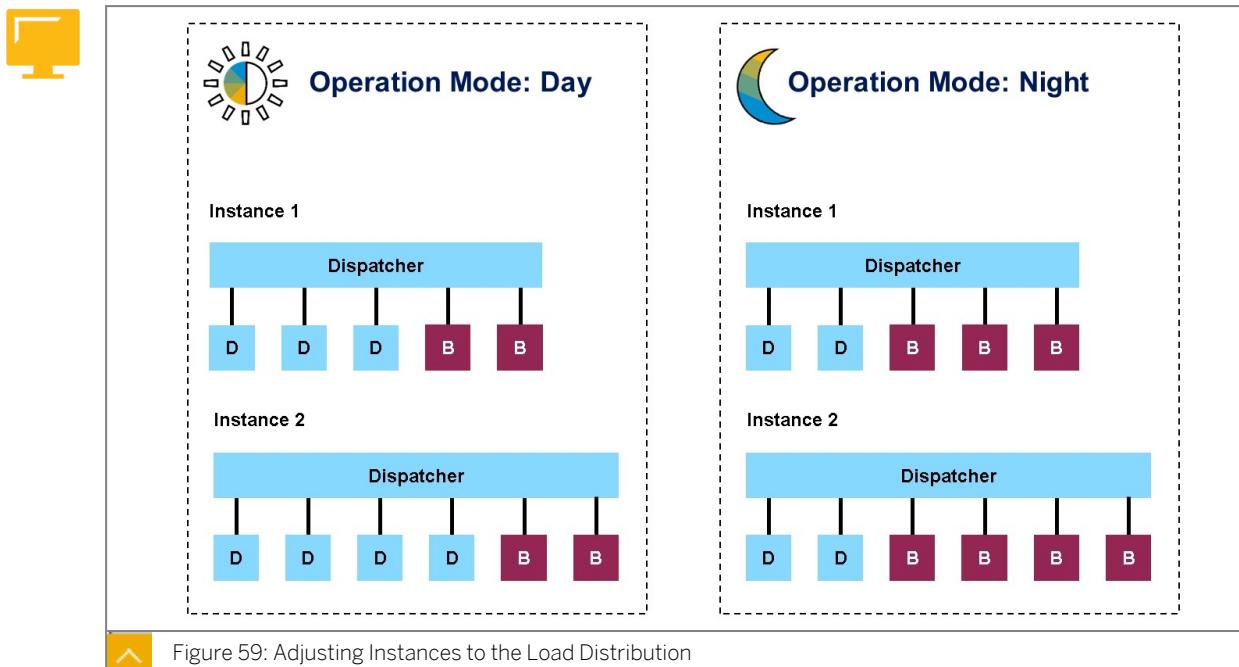
After completing this lesson, you will be able to:

- Describe the concept of operation modes
- Describe dynamic and reserved work processes
- Set up operation modes
- Switch operation modes
- Outline the meaning of reserved and dynamic work processes
- Explain the options to change the work processes distribution using operation mode switches
- Describe the usage of consistency checks

Operation Modes



As the figure *Concept of Operations Modes* illustrates, the demands of users on the SAP system may vary over the course of the day. During the day, a large number of dialog users are working in the system. They want to be served with high performance by the system. Therefore, a large number of dialog work processes should be available for users during the day. During the night, however, only a small number of these dialog work processes are used, because less users are working. This frees the system resources to process background jobs.



The type and number of work processes for each instance is defined by profile parameters in the profiles. The distribution of work processes in the profiles is optimized for fast dialog response times, meaning, there are usually many dialog work processes and a small number of background work processes. This means that during the night, system resources, such as

the main memory or the CPU, are still tied to the dialog work processes and cannot be fully utilized by the background processes. It would therefore be very practical, if the system could adjust different types and numbers of work processes to the different demands in the system. This can be achieved through the concept of operation modes.

As the figure *Adjusting Instances to the Load Distribution* shows, by using the operation modes, you can adjust the type and distribution of the work processes to the varying load distribution during the day. You can also adjust the distribution of the work processes to business requirements that only occur once. By defining operation modes, you can change not only the configurable number of work processes defined in the profiles, but also the type and distribution of the individual work process types within this configurable number. The switch between the work process types is performed dynamically during the runtime of the SAP system. The switch is triggered using a defined schedule. A reserved work process is not immediately terminated, but instead marked for switching. This means that certain delays may occur. This type change is recorded in the system log.

During the switch of the operation modes, neither the instance nor the affected work processes need to be restarted. As a result, the quality of the buffer of the SAP system is retained during an operation mode switch, and the work process finishes processing the current request. The individual work processes retain their process ID after the switch. You can observe this in the process overview (SM50).

Reserved, Standby, or Restricted Work Processes

In an SAP system, deadlocks can occur due to various reasons. Therefore, work processes can be reserved so that the system can free itself from deadlocks. That is why they are called *reserved processes*, also known as *standby* or *restricted* work processes.

The reserved work processes are always of type dialog. They can be defined via the *rdisp/wp_no_restricted* parameter. The default value of the parameter is 0.

Reserved work processes are started during the system start and run in the system occupying system resources. However, they are not used in normal operation, even if all dialog work processes are occupied and further requests are waiting in the dialog queue to be processed. To avoid possible deadlocks, these reserved dialog work processes are used only when all normal dialog work processes are in *Hold* status. You can monitor it in transaction SM50.

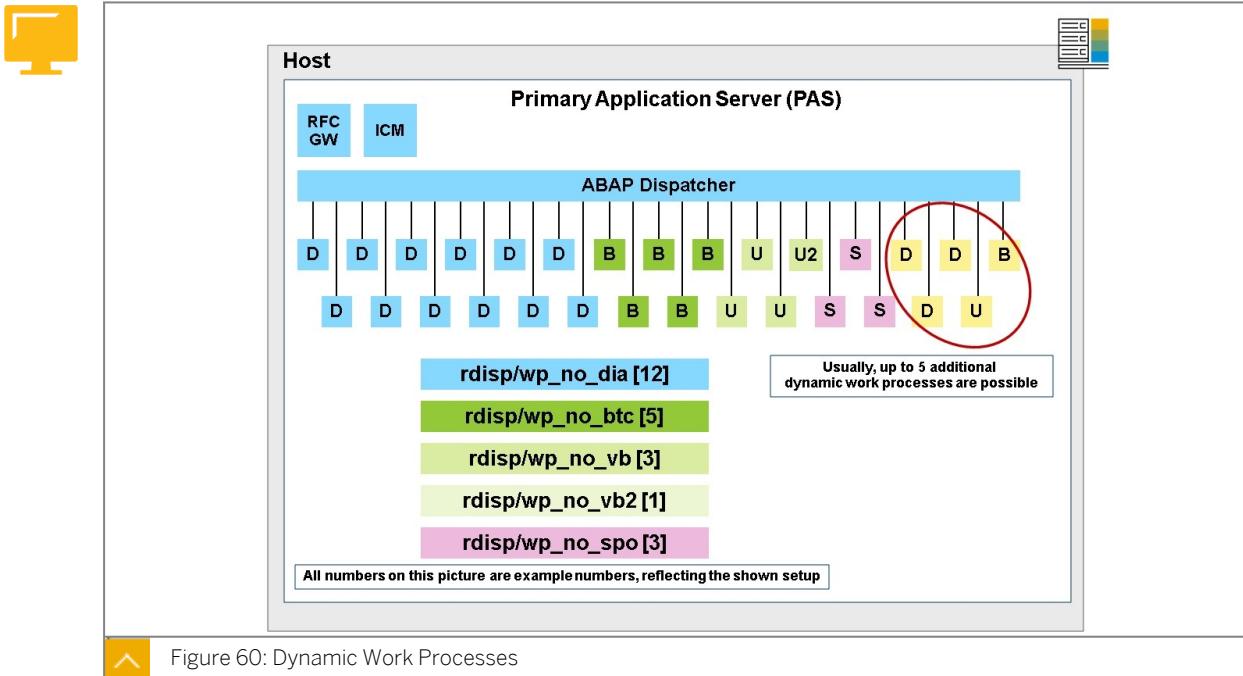
When all dialog work processes are hold, the reserved work processes start working and try to resolve the deadlock. After at least one of the dialog work processes is not on *Hold* any longer, the reserved work processes become *reserved* again and are no longer working.



Hint:

To prevent problems in the configuration of the operation mode switch, we recommend that you do not change the *rdisp/wp_no_restricted=0* parameter. Instead, we recommend configuring dynamic work processes when setting up operation modes.

Dynamic Work Processes



Dynamic work processes enable a kind of adaptive computing on the one hand, and on the other hand, enable the SAP system to free itself from deadlocks – at least within certain boundaries. Dynamic work processes can be seen as a supplement to CCMS operation mode switching (changing the work process type).

Dynamic work processes are not used in normal operation. The ABAP dispatcher starts them when they are needed and they can have any type. When the dynamic work processes are no longer needed, they are stopped again and no longer occupy any system resources. Unlike the restricted work processes, the dynamic work processes are started when all work processes of all types (not only dialog) have the status *Hold*. You can see it in transaction SM50. This is done to avoid deadlocks. However, it takes some time for dynamic work processes to start and become available. This is what you have to consider.

The number of dynamic work processes can only be defined indirectly. The profile parameter *rdisp/wp_max_no* specifies the total number of all work processes, including the dynamic work processes.

As of kernel 7.42, the profile parameter *rdisp/wp_max_no* is set to the following formula:

$$\$ (RDISP/WP_NO_DIA) + \$ (RDISP/WP_NO_VB) + \$ (RDISP/WP_NO_VB2) + \$ (RDISP/WP_NO_ENQ) + \$ (RDISP/WP_NO_BTC) + \$ (RDISP/WP_NO_SPO) + MAX(\$ (RDISP/WP_NO_RESTRICTED), 5)$$

The kernel default formula is a sum of all configured work processes and it defines the standard number of 5 work processes to be started dynamically.

This means each instance has five dynamic work processes, as long as no restricted, or standby, work processes are configured. If restricted work processes are configured, then there are fewer (or no) dynamic work processes, accordingly.

However, you can override the formula.



Hint:

We recommend that you leave this parameter at the default setting. You can change the number of dynamic work processes conveniently using the operation mode switch.

However, if you set this parameter to a fixed value, make sure that there are at least 5 more work processes than the total number of all configured work processes.

The number of configurable work processes is specified with the *rdisp/configurable_wp_no* parameter. As of kernel 7.42, the parameter is set to the following formula:

```
MAX( $(RDISP/WP_MAX_NO) - 5, $(RDISP/WP_NO_DIA) + $(RDISP/WP_NO_VB) + $(RDISP/WP_NO_VB2) + $(RDISP/WP_NO_ENQ) + $(RDISP/WP_NO_BTC) + $(RDISP/WP_NO_SPO) )
```

You should not change the *rdisp/configurable_wp_no* parameter.



Note:

With kernel 7.42, the default values of the profile parameters *rdisp/wp_max_no* and *rdisp/configurable_wp_no* have been changed from the string "DEFAULT" to formulas. The formulas lead to the same calculations as the string "DEFAULT" in kernel 7.41. However, the profile check in transaction RZ10 reports errors when the string "DEFAULT" is used with kernel 7.42 (or later) because these parameters expect a numeric type. For this reason the parameters should be removed from all profiles instead of setting them to the value "DEFAULT".



Hint:

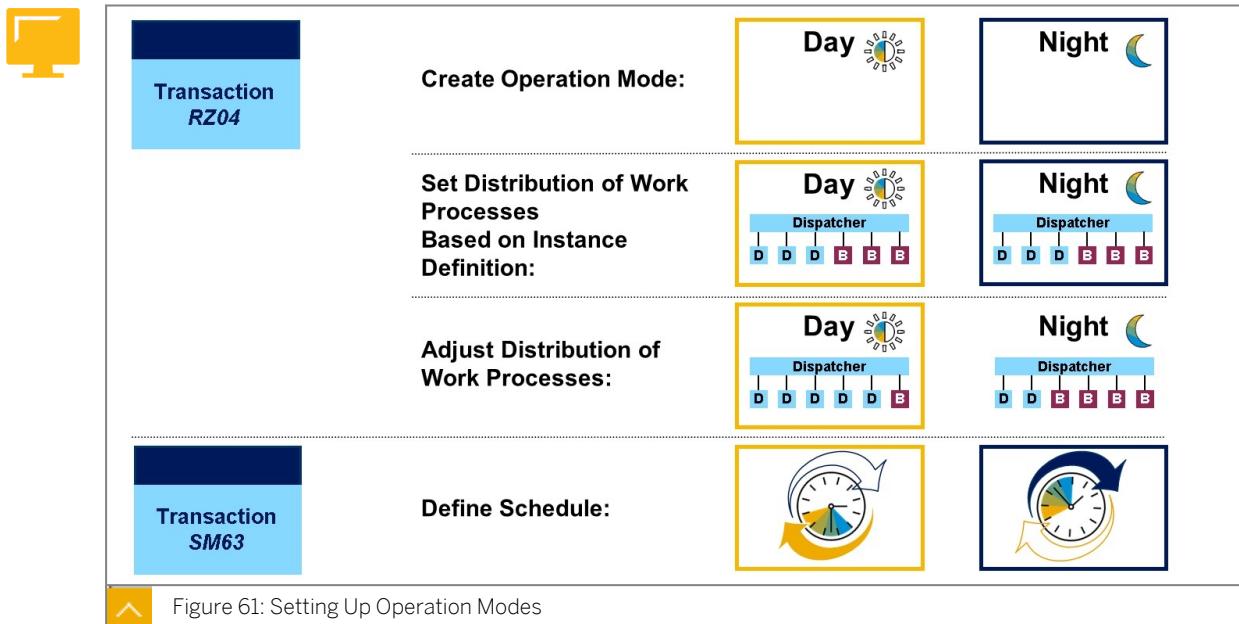
We recommend that you do not set the following parameters into any profile unless you have a compelling reason: *rdisp/configurable_wp_no*, *rdisp/wp_no_enq*, *rdisp/wp_no_restricted*, *rdisp/wp_max_no*.

We also do not recommend that you change profiles via the basic maintenance in transaction RZ10. The basic maintenance writes a lot of parameters into the profiles that are not required. This unnecessarily increases the complexity of profiles and the risk of inconsistent configuration. In the case that you change profiles via transaction RZ10, we recommend that you use **extended maintenance**.

For more information, please read the following SAP Note <https://me.sap.com/notes/2190597>.

Operation Modes Setup

As shown in the figure *Setting Up Operation Modes*, the operation modes are set up in a number of steps.



Steps to Configure Operation Modes

1. First, you create the operation modes in transaction RZ04. They are created as empty containers.
2. Next, all active instances of the system are detected and the work process configuration defined in the instance profiles is assigned to the operation modes as default value.
3. You can now make allocations for the individual operation modes and individual instances within the range of the total number of work processes taken from the instance profile.



Hint:

You should make the allocation primarily between the dialog and background work processes, as – for example – the number of spool work processes cannot be changed using operation modes.

4. You then specify the periods within a day (24 hours) for which the operation modes are valid and when the switch between the different operation modes should happen. You do this specification in the timetable (transaction SM63).

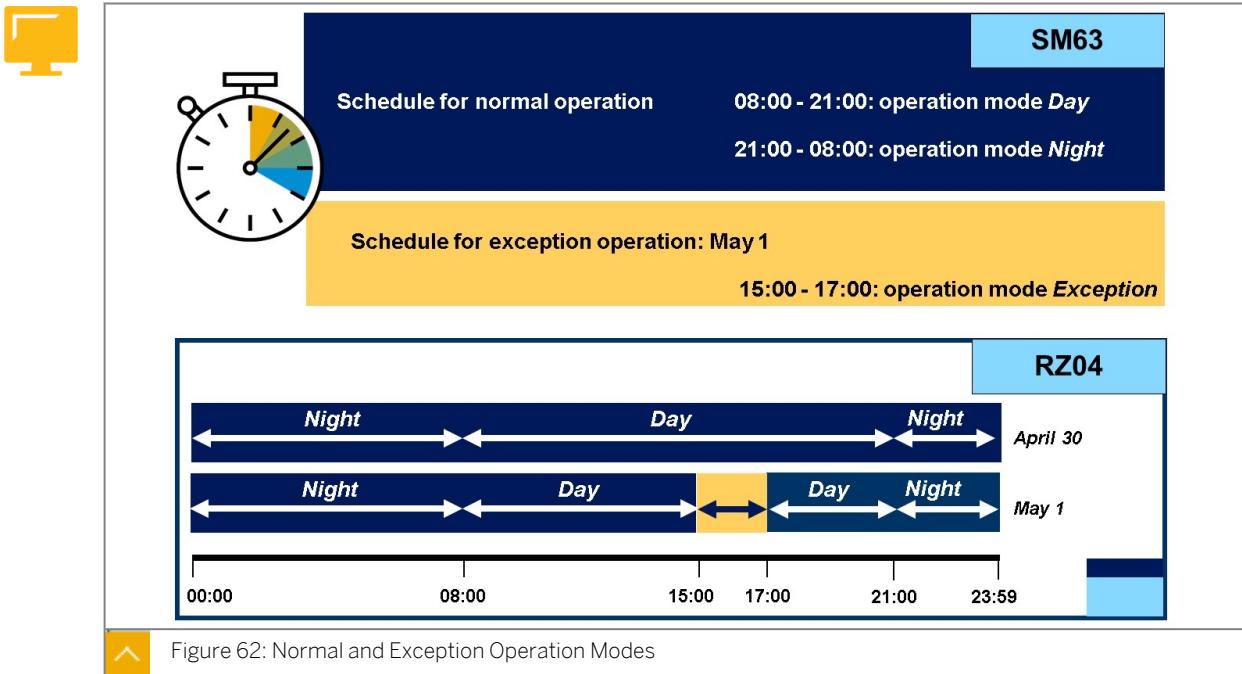


Figure 62: Normal and Exception Operation Modes

In transaction RZ04, you can define a normal and an exception operation mode.

A normal operation mode means that work process configuration for an instance is defined for time periods within a 24-hours time frame. No gaps are accepted here. This configuration is valid for each single day.

You must define a timetable for normal operation before a switch of operation modes can take place. The configuration in the instance profile remains active.

An exceptional operation means that for a very specific time period a different work process configuration is required, for example, when there is activity planned like import of transports or a client copy. You can define an exceptional operation mode for a specific day when the planned activity is going to happen. The exception operation can only be defined as a unique event. After the time period is over, the system switches back to the normal operation mode.

Operation Mode Switch

When operation modes are switched, the work processes are redistributed automatically without stopping and restarting the instances. Only the work process types are changed. For example, a work process used as a dialog process can be switched for use as a background process.

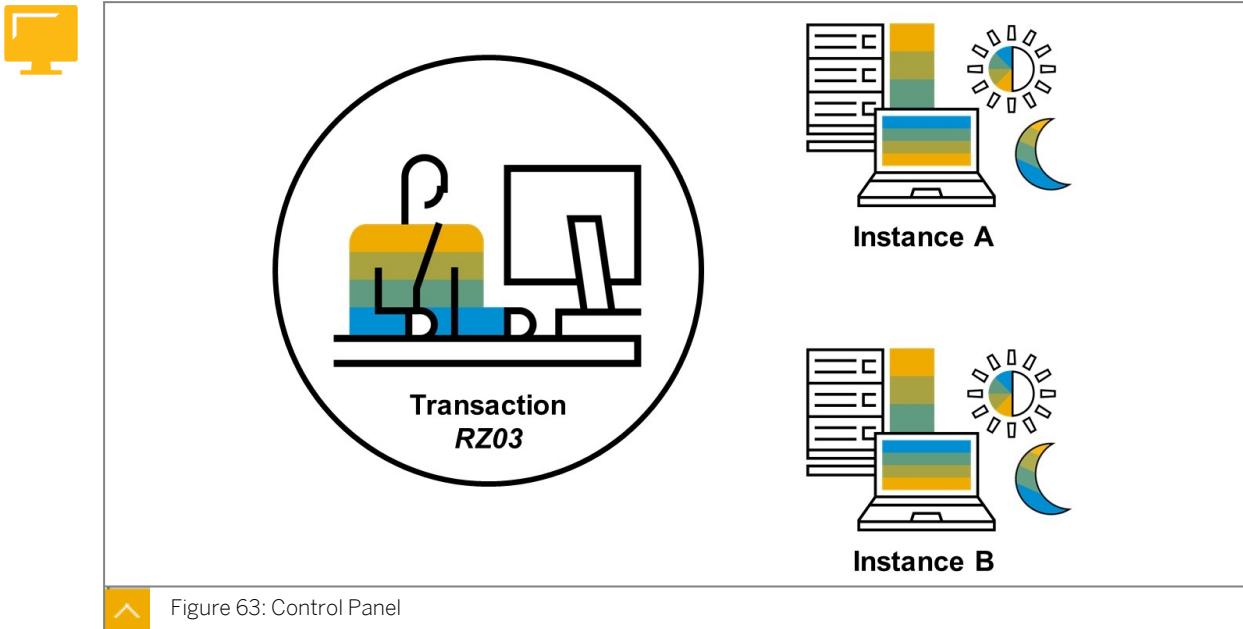
The new process type is not activated until the process is free. This means that a process might not be switched immediately. Instead, a process is set to be switched when next possible. For example, if all background processes to be switched to dialog processes still have jobs, the processes are switched one by one when the jobs are completed. Processing is not interrupted. Normal system operation continues uninterrupted during the operation mode switch.



Note:

Operation mode switches are recorded in the system log. The old process type and the new process type are recorded for each work process that is switched.

The operation mode switch is performed automatically according to the configuration in the time table (transaction SM63). If you have set up a normal operation, the switch is repeated within a 24-hour cycle every day. If you have configured the exceptional operation mode, the switch is performed automatically for the configured period of time on the date that you have defined.



You can also switch the operation modes manually in the *Control Panel* (transaction RZ03). You can do this at any time regardless the time table. However, you must always ensure that a manual operation mode switch does not interrupt system operation, for example, because there are too few dialog processes available.

You can switch the operation mode either for all instances by choosing *Control → Switch operation mode → All servers*, or for a selected instance by choosing *Control → Switch operation mode → Selected servers*.

You can also simulate the operation mode-switching first by choosing *Control → Switch operation mode → Simulation*. The system then checks which instances can be switched.

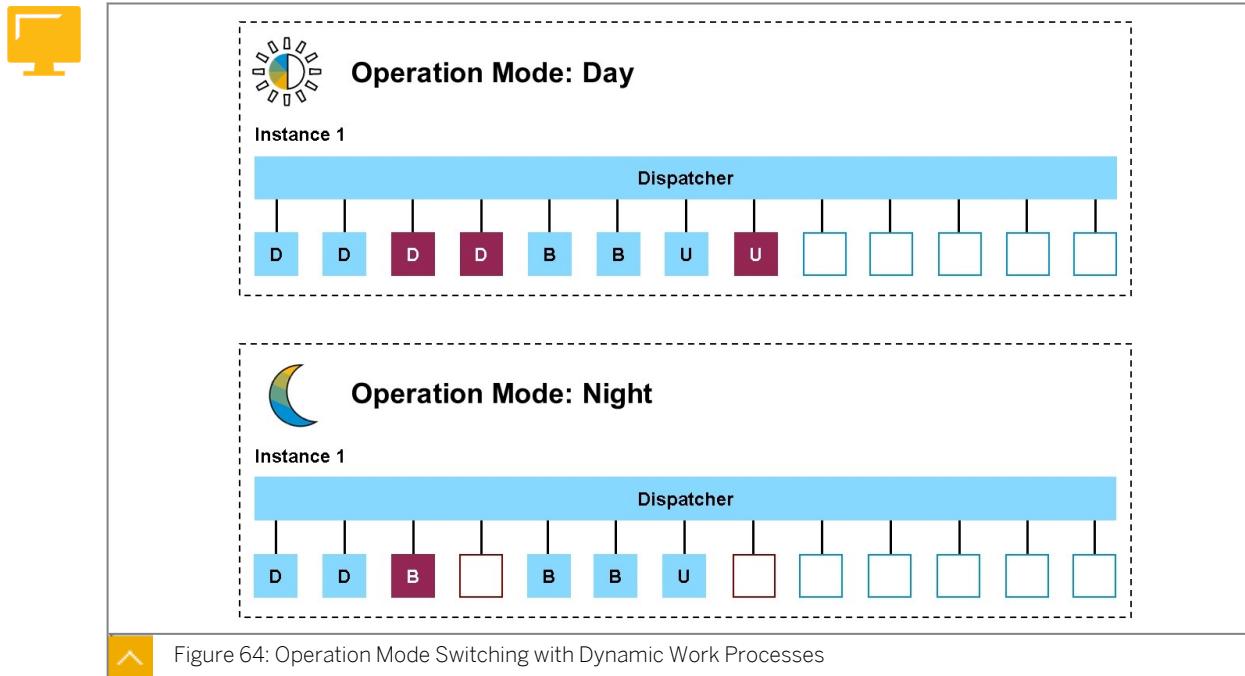
You can also trigger an operation mode switch via an ABAP program using a function module (*RZL_PERFORM_BA_SWITCH*).

For more information, please check the official documentation on the SAP Help Portal:
https://help.sap.com/docs/ABAP_PLATFORM_NEW/b17bd20044a44edb9aa7830ee52ffac3/04520e506567031de10000000a44538d.html

Dynamic and Reserved Work Processes

As already described in this lesson, there are two special types of work processes: reserved and dynamic work processes. The number of those special work processes in the system is defined by profile parameters.

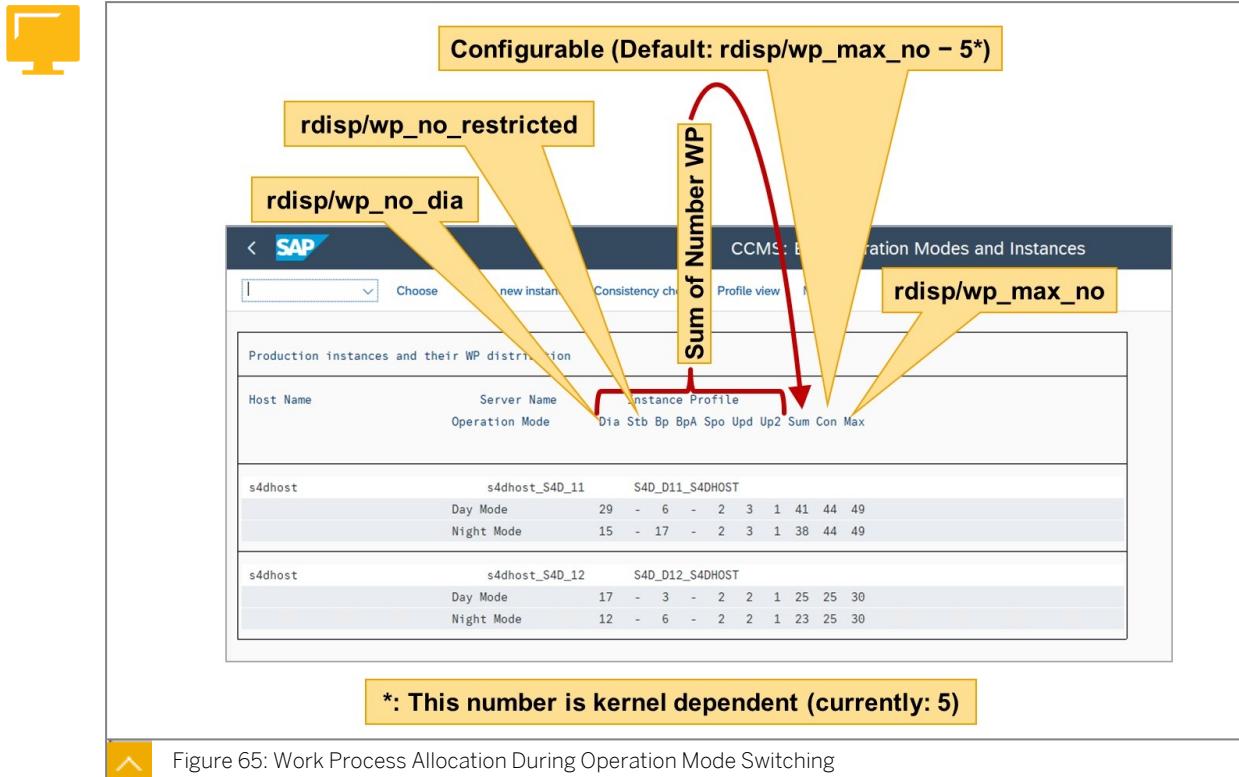
Operation mode switching enables you to adjust the number configurable work processes without changing the profile parameters. When setting up operation modes and performing the distribution of work processes, if you do not allocate all configurable work processes to the different work process types, the remaining work processes become dynamic work processes.



The figure *Operation Mode Switching with Dynamic Work Processes* shows that if you reduce 2 dialog work processes and 1 update work process, and increase the batch (background) work processes by 1 for the *Operation Mode: Night*, you get the following scenario:

In *Operation Mode: Night* the one reduced dialog work process switches a batch work process. The other reduced dialog work process and the update work process are stopped, because they have become dynamic work processes, since not all the configurable work processes were allocated to the different types. When *Operation Mode: Day* is switched again, the increased batch work process becomes a dialog work process again, and the two dynamic work processes become an dialog and a update work process.

Work Processes in the Operation Mode Switch



Operation mode switching and allocation of the work processes are defined in transaction RZ04. The figure *Work Process Allocation During Operation Mode Switching* shows the *Night* and *Day* scenario for the instances of SAP system S4Q. In *Night* operation mode, the number of dialog work processes is reduced and the number of background work processes is increased. The total number of all active work processes is indicated by the *Sum* column. The number of all configurable work process is shown in the *Con* column. The *Max* column indicates how many work process can be started dynamically. The default value is 5 more than all configurable work processes. The configurable and maximum number of work processes cannot be changed.

If the number in the *Sum* column is less than in the *Con* column, it means that not all configurable work processes are allocated to the different work process types. All the work processes that have not been allocated become dynamic work processes in addition to the default 5.

You cannot arbitrarily change the work process distribution.

Table 1: Work Process Types and Type of Changeability

Work Process Type	Type of changeability (+ = increase number, - = decrease number)
Dialog	+: Possible (at least 2D WPs remain) -: Possible
Reserved	Cannot be changed
Background	+: Possible -: Possible

Work Process Type	Type of changeability (+ = increase number, - = decrease number)
Class A	Defines the subset of the B WPs that are only to process class A background jobs.
Update	+: Possible, if there is at least 1 V-WP -: Possible, if at least 1 V-WP remains
U2 update	+: possible, if there is at least 1 U2-WP -: Possible, if at least 1 U2-WP remains
Spool	Cannot be changed
Sum	Only indirectly changeable
Configurable	Cannot be changed
Maximum	Cannot be changed

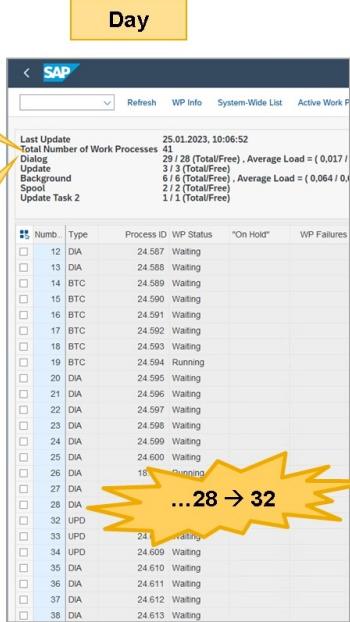


Note:

If you change the allocation, you should primarily change between the dialog and background work processes.



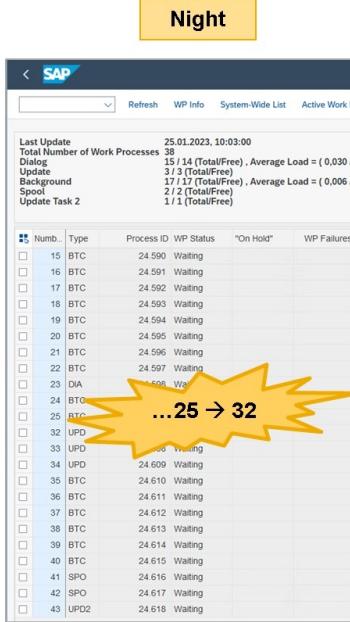
Day



Displays the sum of currently (!) configured work processes

This is not equal to rdisp/wp_no_dia (32) but determined by the # of dialog WPs configured by the active operation mode (29)

Night



...28 → 32

...25 → 32

The screens above shows an instance with a work process configuration based on two different operation modes: both operation modes use less work processes than the following profile parameters would allow: rdisp/wp_no_dia, rdisp/wp_no_vb, rdisp/wp_no_vb2, rdisp/wp_no_btc, rdisp/wp_no_spo. This allows to add work processes of any type by using operation modes, without the need to restart the instance.

◀
Figure 66: Process Overview During Mode Switching with Dynamic Work Processes

The figure *Process Overview During Mode Switching with Dynamic Work Processes* shows the impact of set profile parameter value and the operation mode switching between day and night. The work processes that have been switched to dynamic work processes, and therefore

were stopped, are not visible in transaction SM50. There, you will find corresponding gaps in the work process numbering.

Monitoring and Consistency Checks

In transaction RZ03, you can also monitor the instances and the operation modes.

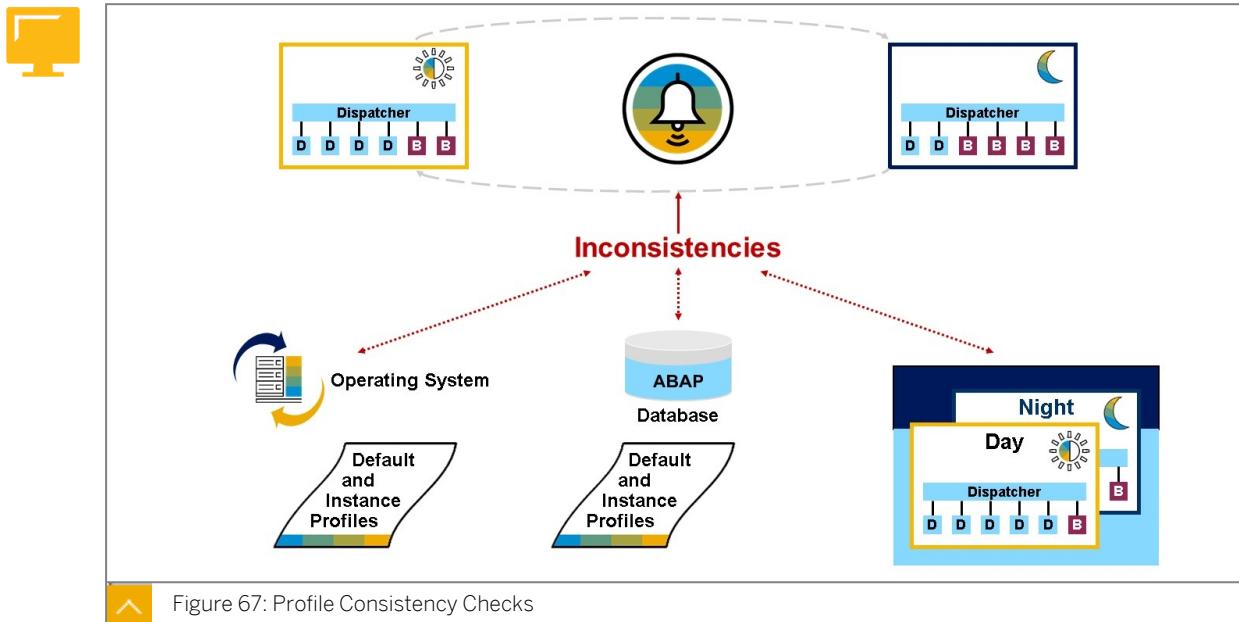
There, you can use the following functions:

- Check the status of all instances and of the operation modes.
- Manually switch an operation mode.
- Display an overview of the distributed work processes.
- Switch to the alert monitor.

You can display a detailed analysis of the status of the individual instances by selecting the instance and choosing *Monitoring → Status details*.

There you can see detailed information about the instance and the current work process configuration from the active operation mode.

By choosing *Monitoring → Alert Monitor*, you can switch to the transaction RZ20 and monitor the alerts.



As the figure *Profile Consistency Checks* shows, inconsistencies in the SAP system may prevent you from switching between operation modes. Such inconsistencies can occur if the number of work processes is defined differently in different places in the system, namely the instance profiles at operating system level, the instance profile in the database, and the definition of the operation modes themselves.

If, for example, the number of work processes in the profiles is changed, the system can no longer switch operation modes until after a restart of the instance. You must therefore adjust the configuration of the operation modes after any change to the work process distribution in the profiles.



LESSON SUMMARY

You should now be able to:

- Describe the concept of operation modes
- Describe dynamic and reserved work processes
- Set up operation modes
- Switch operation modes
- Outline the meaning of reserved and dynamic work processes
- Explain the options to change the work processes distribution using operation mode switches
- Describe the usage of consistency checks

Executing ABAP Task Lists for Automated Configuration

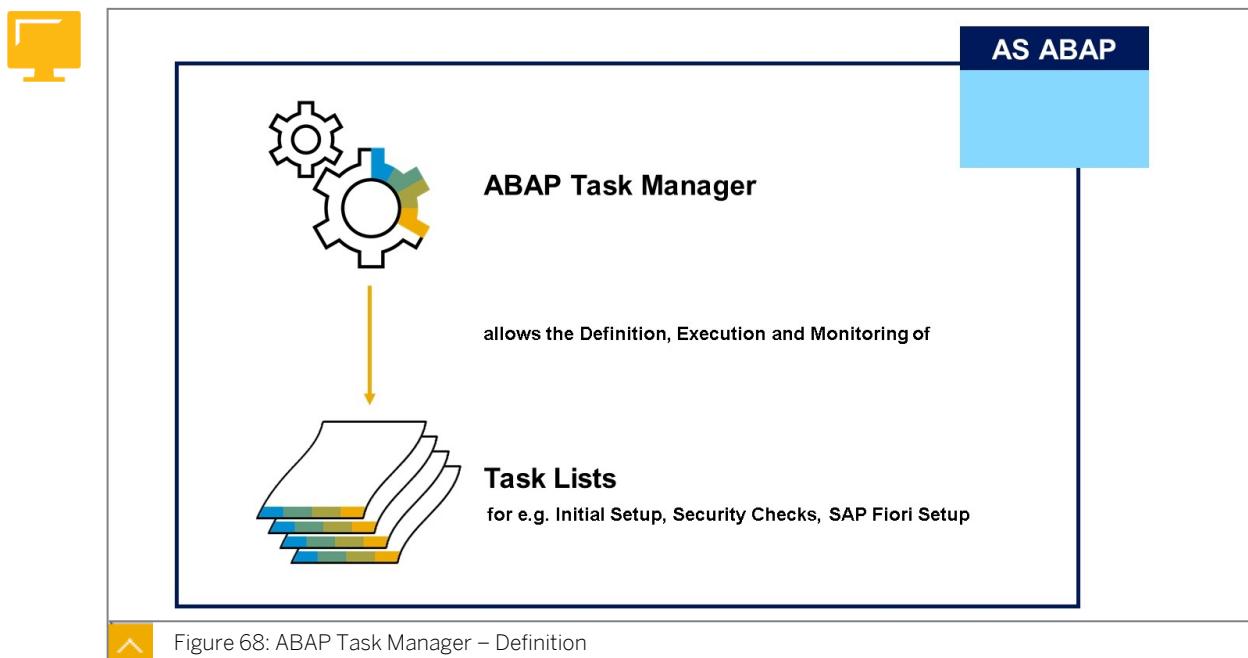


LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe ABAP task manager for lifecycle management automation
- Explain ABAP task lists
- Apply ABAP task lists

ABAP Task Manager for Lifecycle Management Automation



In SAP systems, you can automate many configuration tasks using predefined task lists. You can automate many configuration tasks using the ABAP task manager for lifecycle management automation (ABAP task manager). The ABAP task manager guides you through extensive configuration processes with predefined task lists, which you can customize as needed. You can also create your own task lists.

SAP offers predefined task lists for the following use cases:

ABAP Initial Setup

After installing a new SAP system, you can use task list to set up and configure the system to be able to use it. For example, you can create logon groups, configure the transport management system, configure security settings, or install an SAP license.

Security Checks

With task list for security checks, you can perform security system smoke tests, such as checking the Secure Sockets Layer (SSL) configuration or the basis system configuration.

SAP Fiori Setup

In order to enable SAP Fiori, you can perform system configuration tasks for SAP Gateway or SAP Fiori launchpad. For example, you perform basic configuration steps for SAP Gateway or activate launchpad OData and HTTP services on a Gateway system.

Embedded Search

You can run task lists to automate enterprise search configuration and administration.

ABAP Post-Copy Automation

There are also predefined task lists for ABAP post-copy automation. However, they are only available if you own an SAP Landscape Management (SAP LaMa) respectively SAP Landscape Virtualization Management (SAP LVM) enterprise edition license. Contact your SAP representative to obtain a license to make this function available.

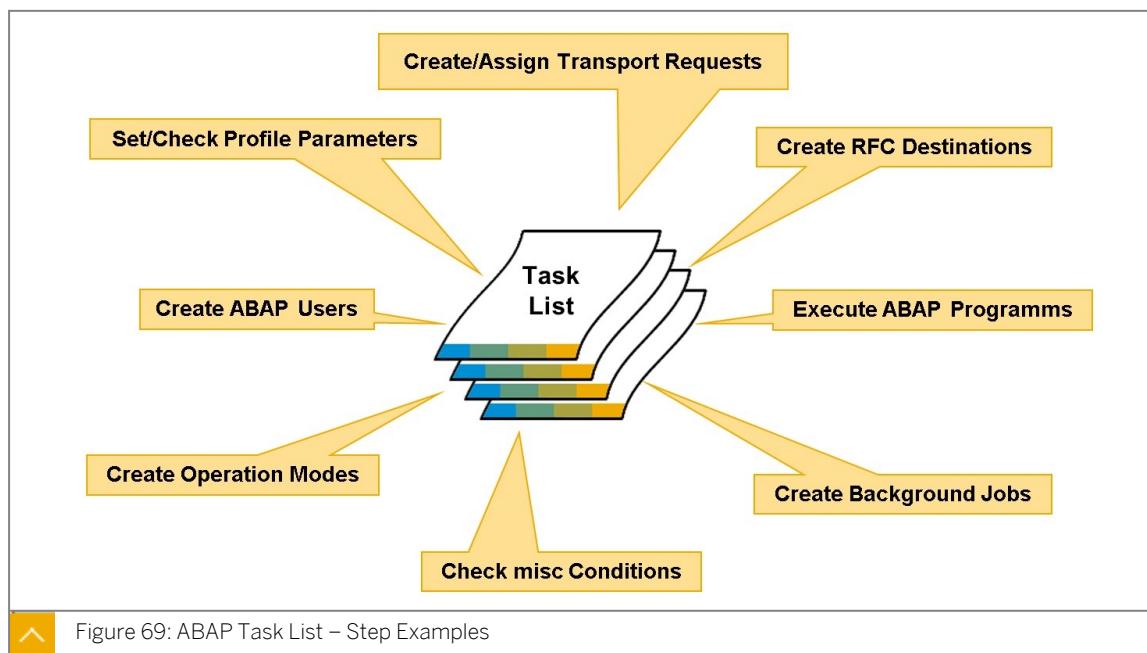
For more information, check the SAP Help documentation: https://help.sap.com/docs/ABAP_PLATFORM_NEW/b5670aaaa2364a29935f40b16499972d/1c8a2c5084505f03e10000000a44176f.html



Note:

The ABAP task manager is available as of SAP_BASIS 7.00 SP 14. For detailed system requirements, see [SAP Note 1589145](#).

Configuration Using ABAP Task Lists



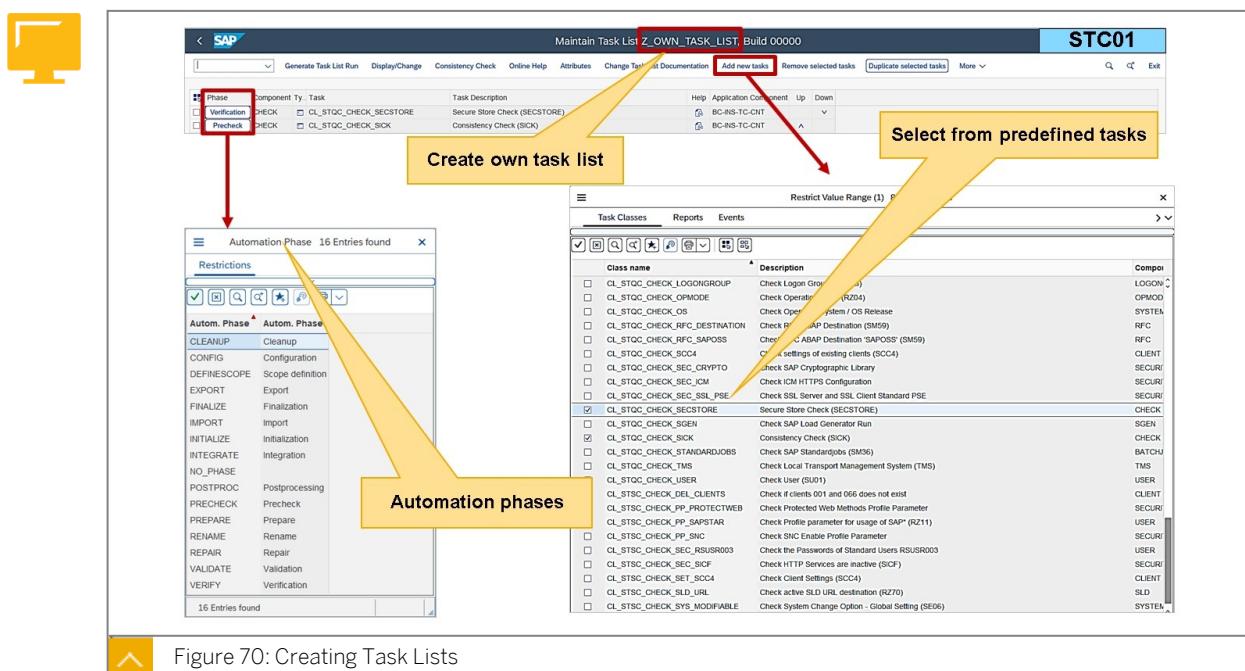
SAP is delivering a lot of automation content for ABAP systems in the form of task lists that you can adapt and execute in the ABAP task manager. The figure *ABAP Task List - Step Examples* only lists some of the examples for possible task list steps, such as create transport requests or check profile parameters. In up-to-date AS ABAP-based SAP systems, there are more than 900 task list steps available.

The ABAP Task manager comprise two transaction in an SAP system:

- STC01 - ABAP task manager for lifecycle management automation where you can create and execute task lists
- STC02 - Task list run monitor to display the log of executed task lists

In transaction STC01 you can do the following:

- Display task lists and included tasks.
- Add, remove, or change tasks in an existing task list.
- Delete task lists which are no longer needed.
- Copy task lists.
- Run a task list (in dialog or as background job).
- Create a variant of a task list (in case of tasks with parameters).
- Run a task list variant.



The figure *Creating Task Lists* illustrates the process of creating a task list. To create a task list, you need to perform the following steps in a development system:

1. On the initial screen of STC01, enter a name for the new task list.

**Note:**

When you create your own task lists, you create new development objects in the repository of the SAP system. Thus, the object must be created in the customer namespace.

2. Choose *Create New Task List*.
3. Provide a task list documentation by adding a related document class and documentation object if available.
4. Add tasks from the predefined tasks in the system.
The tasks belong to different automation phases. Not every task is a configuration task. Also other phases are possible, such as preparation, cleanup, post-processing or others.
5. Reorder tasks if needed and add the parameter settings according to your needs.
6. Save the new task list.
7. Since a task list is considered as a development object, you have to provide a package from the object repository or save it as a local object.
8. Provide a transport request to be able to transport the task list in your system landscape.

**Hint:**

You can provide variants for the task lists and create a list with a variant. In transaction STC01 you can create new variants.



The screenshot shows two SAP transaction screens. The top screen is titled "Maintain Task List Z_OWN_TASK_LIST, Build 00000" and shows a table of tasks. A red arrow points from the "Generate Task List Run" button to the second screenshot. The bottom screen is titled "Maintain Task List Run Z_OWN_TASK_LIST_20221206142652709" and shows a table of tasks with a "Run" button. A yellow arrow points from the "Run task list in dialog or background mode" button to the bottom screen. A yellow box labeled "Provide input if required" is overlaid on the bottom screen's input fields.



Figure 71: Executing Task Lists

If you want to execute an ABAP task list, whether it is your own created one or an SAP predefined task list, you have to generate a task list run. The ABAP task manager assigns a name to the task list run automatically. The list displays the tasks in the order of their execution that is defined for a successful configuration.

Some tasks may require an input. You can provide it as parameter in the *Parameter* column of each task. For some tasks input is mandatory. If you don't provide it, either a default setting applies or the task list will stop and prompt you for input.

After all information has been maintained, you can run or resume the task list in dialog mode. You can also schedule a background job that will execute the task list at a certain point in time.

If the task list run has any errors, it will stop. Choose the *Status* column of the failed task to see the log in the lower screen area. It provides detailed information and further instructions. You should resolve the problem and resume the task list run.

If you can not resolve the problem, deselect the task in the task list run to skip it. The configuration step is then not included in the process.

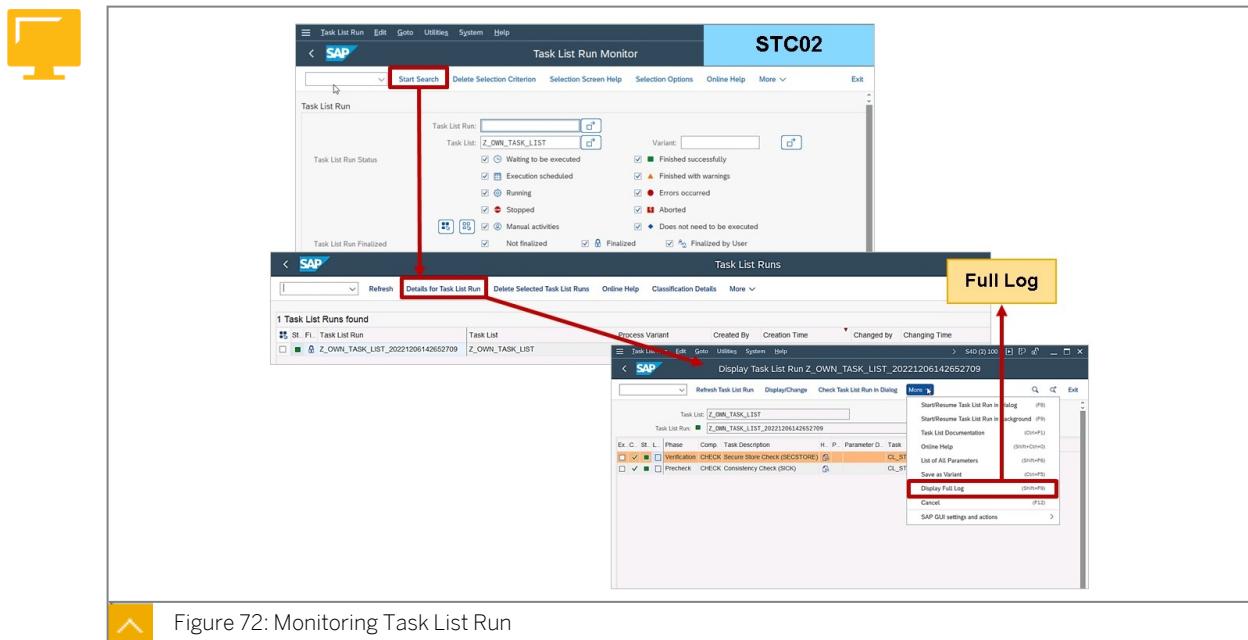


Figure 72: Monitoring Task List Run

You can monitor executed task list runs in transaction STC02. There, you can display log information for task list runs.

1. Call transaction STC02.
2. Search for the task list run you want to view. You can also search for the task list to display all related task list runs.
3. Choose *Task List Run → Display Log*.

For more information about working with task lists in general and the required authorizations, see the official documentation: https://help.sap.com/docs/ABAP_PLATFORM_NEW/b5670aaaa2364a29935f40b16499972d/1c8a2c5084505f03e10000000a44176f.html.

For more information about task lists in the area of SAP Fiori, check the online documentation: https://help.sap.com/docs/SAP_FIORI_OVERVIEW/9b4ee51ce91e4a9cadd5e6d43c2992ba/bfd1b053a647e842e10000000a4450e5.html.



LESSON SUMMARY

You should now be able to:

- Describe ABAP task manager for lifecycle management automation
- Explain ABAP task lists
- Apply ABAP task lists

Learning Assessment

1. In which sequence does the system read the system parameters from the specified locations?

Choose the correct answers.

- A 1. Instance profile 2. Default profile 3. Kernel
- B 1. start.ini 2. Kernel 3. Default profile 4. Instance profile
- C 1. Kernel 2. Default profile 3. Instance profile
- D 1. Kernel 2. start.ini 3. Default profile 4. Instance profile

2. Bring the different sources for configuration information used during the startup of an AS ABAP-based instance into the correct chronological sequence.

Arrange these steps into the correct sequence.

- Kernel
- Default profile
- Instance profile
- Environment Variables

3. In what profile should you change the number of background work processes?

Choose the correct answers.

- A Default profile
- B Background profile
- C Instance profile

4. Before using transaction RZ10 for maintaining profile parameters and their values, it is initially required to import the profiles into the SAP system.

Determine whether this statement is true or false.

- True
- False

5. In what sequence should you perform the listed steps when setting up operation modes?

Choose the correct answers.

- A 1. Create operation modes 2. Assign instance 3. Distribute work processes 4. Maintain time table
- B 1. Create operation modes 2. Maintain time table 3. Distribute work processes 4. Assign instances
- C 1. Create operation modes 2. Distribute work processes 3. Maintain time table 4. Assign instances

6. Using operation modes, which options are available to you?

Choose the correct answers.

- A Changing the number of Dialog work processes, provided at least 2 of those remain.
- B Switch operation modes at 4:15pm in the afternoon.
- C Increase the number of Spool work processes.
- D Decrease the number of update work processes from a number higher than 0 to Zero.

7. How can you execute a set of selected activities using the ABAP Task Manager?

Choose the correct answer.

- A You need to use a Task List that contains the wanted activities.
- B You are required to execute each wanted task separately, manually.
- C You upload a spreadsheet file to the ABAP Task Manager containing all wanted tasks.
- D You provide the location of a task file that you created earlier during a recording of the wanted activities.

Learning Assessment - Answers

1. In which sequence does the system read the system parameters from the specified locations?

Choose the correct answers.

- A 1. Instance profile 2. Default profile 3. Kernel
- B 1. start.ini 2. Kernel 3. Default profile 4. Instance profile
- C 1. Kernel 2. Default profile 3. Instance profile
- D 1. Kernel 2. start.ini 3. Default profile 4. Instance profile

You are correct! There is no file called *start.ini* involved in this context; it was made up.

2. Bring the different sources for configuration information used during the startup of an AS ABAP-based instance into the correct chronological sequence.

Arrange these steps into the correct sequence.

- 2** Kernel
- 3** Default profile
- 4** Instance profile
- 1** Environment Variables

You are correct! An AS ABAP-based instance first reads variables from the environment, then it evaluates the kernel default values followed by the default profile and the instance profile. Read more on this in the lesson “Understanding the Parameter Evaluation Sequence” of the course ADM100.

3. In what profile should you change the number of background work processes?

Choose the correct answers.

- A Default profile
- B Background profile
- C Instance profile

You are correct! The instance profile is the correct profile in this case, as settings in this profile only apply to the affected instance, and the number of work processes for an instance is defined here. In the default profile, you maintain parameters that apply to all instances or for the entire SAP system. The background profile does not exist. Read more about this in the course ADM100, Unit 1, lesson 2.

4. Before using transaction RZ10 for maintaining profile parameters and their values, it is initially required to import the profiles into the SAP system.

Determine whether this statement is true or false.

- True
- False

You are correct! Before using transaction RZ10 for maintaining profile parameters and their values, it is initially required to import the profiles into the SAP system. Read more on this in the lesson “Explaining How Profile Parameters are Set” of the course ADM100.

5. In what sequence should you perform the listed steps when setting up operation modes?

Choose the correct answers.

- A 1. Create operation modes 2. Assign instance 3. Distribute work processes 4. Maintain time table
- B 1. Create operation modes 2. Maintain time table 3. Distribute work processes 4. Assign instances
- C 1. Create operation modes 2. Distribute work processes 3. Maintain time table 4. Assign instances

You are correct! You must follow the correct sequence when first setting up the operation modes. Read more on this in the lessons 3 and 4 of the course ADM100.

6. Using operation modes, which options are available to you?

Choose the correct answers.

- A Changing the number of Dialog work processes, provided at least 2 of those remain.
- B Switch operation modes at 4:15pm in the afternoon.
- C Increase the number of Spool work processes.
- D Decrease the number of update work processes from a number higher than 0 to Zero.

You are correct! Operation modes can be used to adapt the number of dialog work processes, provided at least 2 dialog work processes remain configured. Also you can plan operation mode switches in quarter hour intervals. It is not possible to change the number of spool work processes or to decrease the number of update work processes, using operation modes. Read more on this in the lesson "Setting Up Operation Modes" of the course ADM100.

7. How can you execute a set of selected activities using the ABAP Task Manager?

Choose the correct answer.

- A You need to use a Task List that contains the wanted activities.
- B You are required to execute each wanted task separately, manually.
- C You upload a spreadsheet file to the ABAP Task Manager containing all wanted tasks.
- D You provide the location of a task file that you created earlier during a recording of the wanted activities.

You are correct! ABAP Task Manager executes tasks that you compiled into a Task List – or you can use a predefined Task List. It is not required to execute each task manually, nor is it an option to upload or provide the location of some file containing the wanted activities. Read more on this in the lesson "Working with ABAP Task Manager" of the course ADM100.

UNIT 4

Introduction to Database Administration

Lesson 1

Describing the Architecture of Database Systems

139

Lesson 2

Scheduling and Monitoring Database Actions

149

Lesson 3

Performing Regular Database Checks

159

Lesson 4

Appendix: Describing the Concept of SAP Data Archiving

165

UNIT OBJECTIVES

- Explain the purpose and architecture of databases
- Describe SAP HANA backup and recovery principles
- Explain the use of the DBA Cockpit to centralize database administration tasks
- Schedule and monitor a database backup with DBA Cockpit
- Perform a batabase backup with SAP HANA Cockpit
- Describe examples of periodic database checks
- Explain the idea and benefits of data archiving
- Outline the technical process of data archiving and list options for accessing archived data

Describing the Architecture of Database Systems

LESSON OVERVIEW

In this lesson, you will learn basic theory about the architecture of a relational database and how it works, as well as fundamental backup concepts. These include regular backups of data and log information.

Business Example

SAP systems store the application data in a relational database. The administrator regularly backs up this data.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the purpose and architecture of databases
- Describe SAP HANA backup and recovery principles

Fundamentals of Database Administration

In an SAP system, the task of the database is to store the application data persistently. In a system of the SAP Business Suite, databases from different vendors are used (called “AnyDB” in this lesson), as well as SAP HANA, SAP’s own in-memory database. The SAP S/4HANA system runs on the SAP HANA database, exclusively.



Hint:

This introductory course to SAP system administration emphasizes the importance of using the database correctly, and teaches basic theory about the functions of a database, such as, how to work with the DBA Cockpit for database management in the SAP system. Anything beyond this must be obtained in other courses from the database vendor and/or by attending the SAP Learning course HA200 for SAP HANA or other SAP Learning courses on AnyDBs, such as SAP ASE, MaxDB, and so on.

All databases use similar principles to store data persistently. These principles are described briefly here.

As the figure *Database: Concepts* illustrates, a database management system (DBMS) includes database processes, an area in the main memory, data files that contain the data, and log files, in which changes to the data are logged.

A DBMS performs the following functions:

- Manages large amounts of data in a multiuser environment so that many users can concurrently access the same data
- Maintains relationships between data
- Provides secure access to data using the user authorization concept
- Recovers data automatically to the most recent consistent status after a system failure
- Delivers high performance for processing data requests

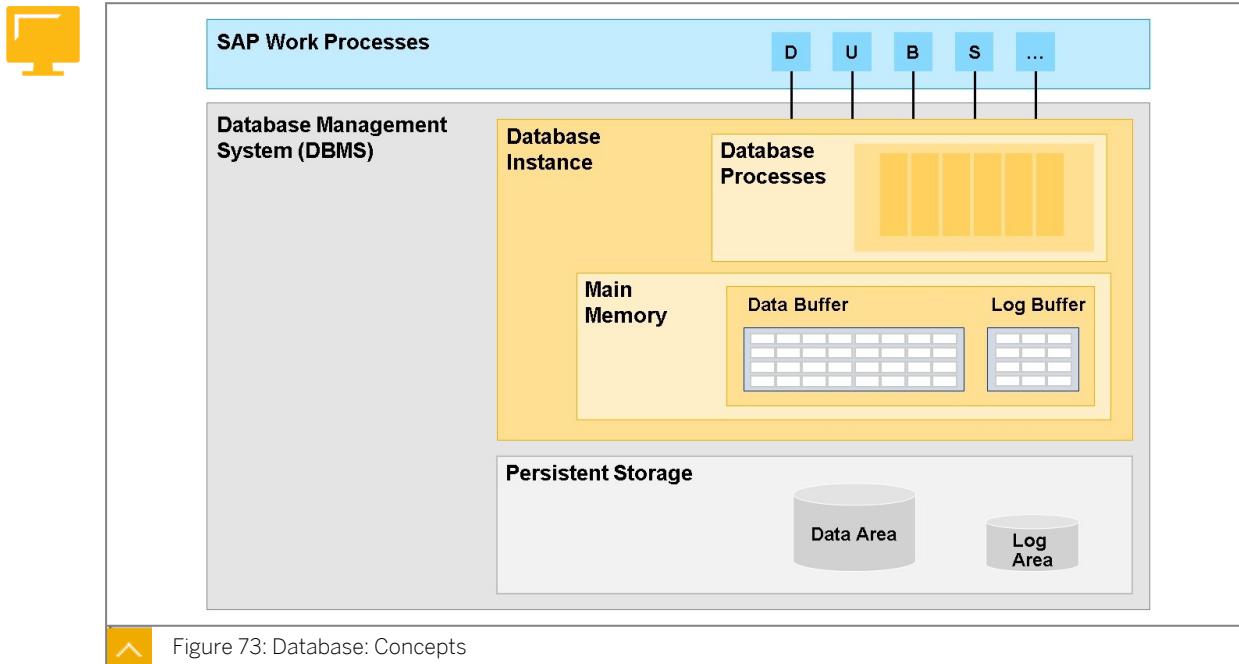


Figure 73: Database: Concepts

During the starting of an SAP system, all work processes connect to the database. Database queries are passed by SAP work processes to the database processes assigned to them, which execute the queries on the database.

The data is stored in the data files. The data is always accessed using the buffer in main memory.

Specific database processes are responsible for writing the changed data from the buffer to the data files. This always involves reading and writing entire blocks.

When data is changed, the changes are recorded in the log buffer and written synchronously to the log files during the commit, which means the log file contains the status changes of the database. Only the changes, and not the complete blocks, are logged in the log buffer. The entries are written from the log buffer to the log file, of which there can be one or more, depending on the database. Each database has a mechanism that saves the log information in the log file in continuous log backup files. This ensures that the log file does not become too large.

We recommend that you mirror the log file. Some databases provide special software mirrors to allow you to do this.



Caution:

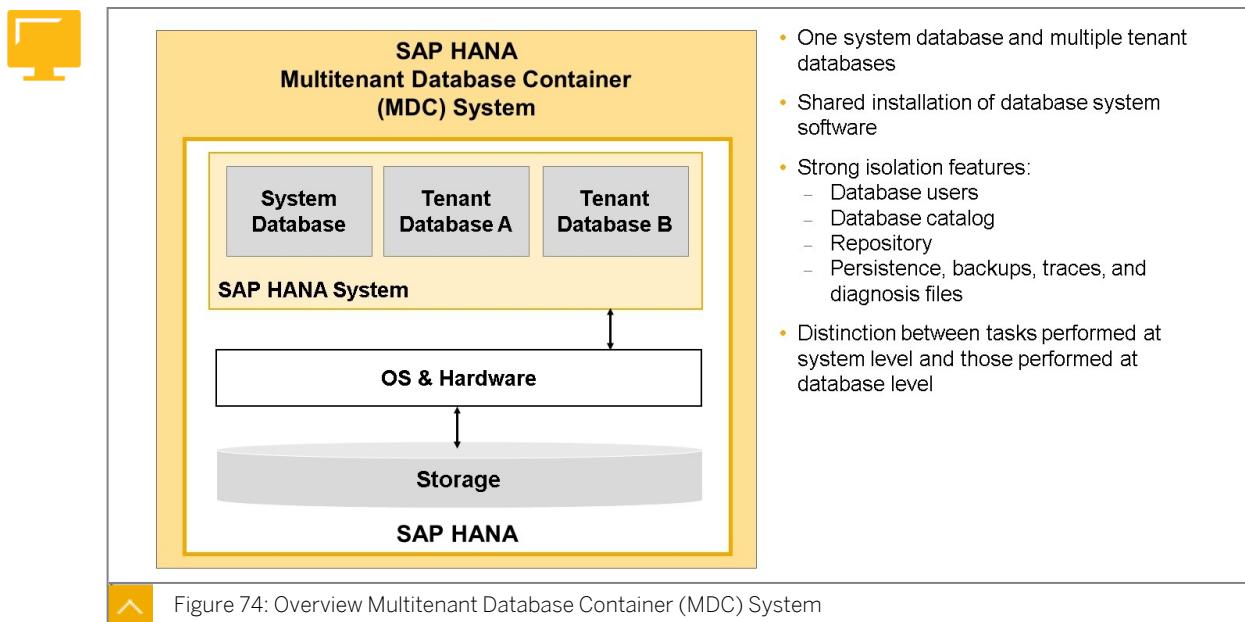
Deactivating the mechanism to mirror the log file could result in the loss of status changes. This would mean the threat of data loss if a hard disk crash occurs.

A database always includes structural data that contains essential information for the database, such as the number of data files, and so on.

Architecture of SAP HANA

Concept: Multitenant Database Container

An SAP HANA database system is installed as a multitenant database container system and can contain more than one tenant database.



A multiple-container system always has exactly one system database, used for central system administration, and any number of multitenant database containers (including zero), also called tenant databases. An SAP HANA system installed in multiple-container mode is identified by a single system ID (SID). Databases are identified by a database name and a SID. From the administration perspective, there is a distinction between tasks performed at system level and those performed at database level. Database clients, such as the *SAP HANA cockpit*, connect to specific databases.

The concept of a multiple tenant database container system is based on a single SAP HANA system, or database management system, with a single system ID. Installing an SAP system on an SAP HANA database system with its own single database system ID contains at least one tenant database, in addition to a system database. The system database keeps the system-wide landscape information, and provides system-wide configuration and monitoring. Users of one tenant database cannot connect to other tenant databases. They also cannot access application data there, unless the system is enabled for cross database access. The tenant databases are, by default, isolated from each other in terms of application data and user management. Each tenant database can be backed up and recovered independently from one another. Because all tenant databases are part of the same SAP HANA database management system, they all run with the same SAP HANA version (revision number). In addition, a defined high availability disaster recovery scenario applies to all tenant databases.

An SAP HANA database system consists of multiple services, for example, name server, index server, preprocessor server, and so on. The databases in an SAP HANA system run different combinations of these servers.

Only the system database runs the **name server** service. The name server contains landscape information about the system as a whole, including which tenant databases exist. The name

server service of the system database does not own the topology information. That is, it does not own information about the location of tables and table partitions in databases. Database-related topology information is stored in the relevant tenant database catalog. The system database can show monitoring data from tenant databases (for example, views in the schema SYS_DATABASES), but it can never show actual content from tenant databases.

Tenant databases run their own **index server** service. The index server is the most important server. It contains the actual data stores and the engines for processing the data and runs in every tenant database. The server for SAP HANA extended application services (**xsengine**) runs in a tenant database.

Services that do not persist data, such as the **compile server** and the **preprocessor server**, run on the system database and serve all databases.

SAP HANA Persistence

To ensure optimal performance, the SAP HANA database holds most of its data in memory. However, it still uses persistent storage to provide a fallback in case of failure.

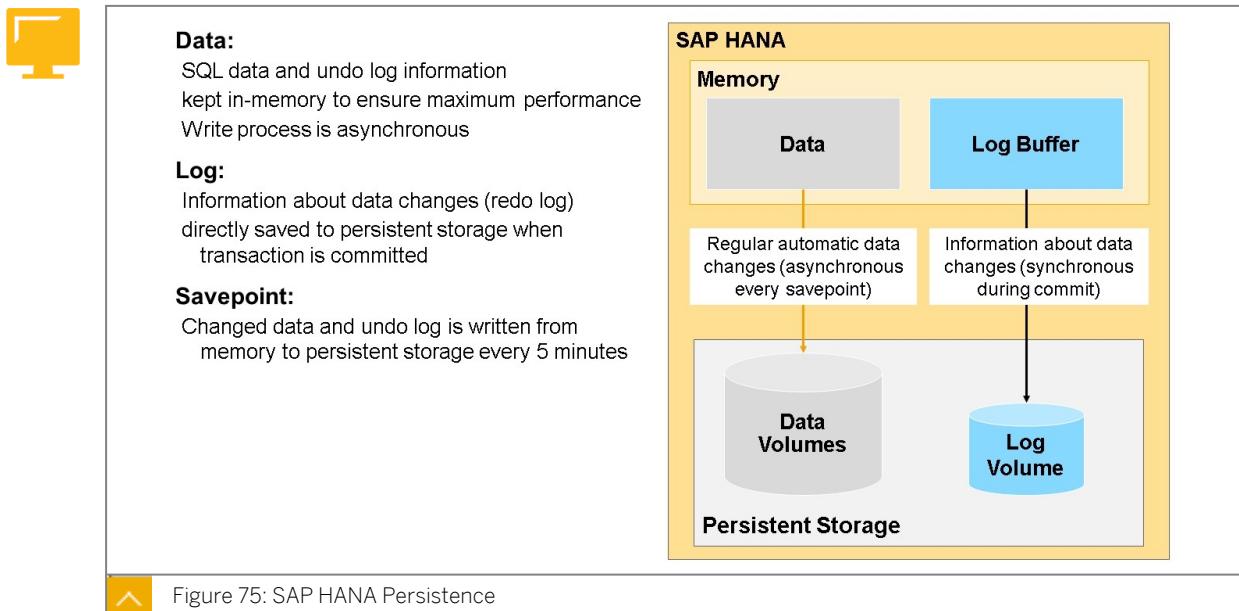


Figure 75: SAP HANA Persistence

During normal database operation, data is automatically saved from memory to disk at regular savepoints. Additionally, all data changes are recorded in the redo log. The redo log is saved from memory to disk with each committed database transaction. After a power failure, you can restart the database as you would with any disk-based database. It then returns to its last consistent state by replaying the redo log since the last savepoint.

Although savepoints and log writing protect your data against power failures, savepoints do not help if the persistent storage itself is damaged or a logical error occurs. To protect against data loss because of disk failures, or to make it possible to reset the database to an earlier point in time, backups are required. Backups typically save the payload (the actual data) of the data area and log area to different locations. Unused space in the database is not backed up.



Note:

Backups are also used for other scenarios such as database copies and they complement other high availability strategies such as system replication or storage replication.

SAP HANA Backup Concept

A backup concept for the database always includes the regular backup of the data files, the structure data, and also the log information.

Backups are performed while the database is running. The impact of backups on system performance is negligible, and users can continue to work while the backup is running.

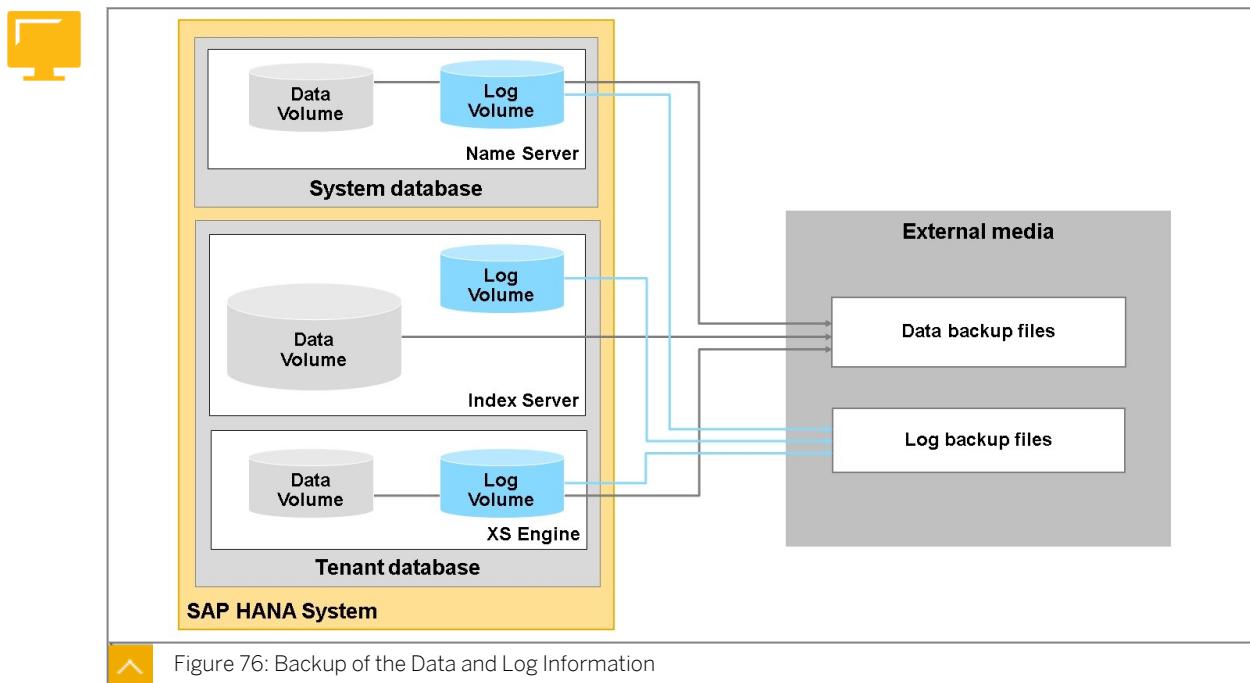
The following principles for data backups apply for an individual database in a multitenant database container system.

A data backup includes all the data that is required to recover the database to a consistent state.

With a data backup, only the actual data is backed up; unused space in the database is not backed up.

The data area is backed up in parallel for each of the SAP HANA services.

While a data backup is running, some data integrity checks are performed. If these checks are successful, the data is written to the backup destination.



Types of Backups

Three types of data backups are available in SAP HANA, as follows:

Complete

A complete data backup includes all the data structures that are required to recover the database.

Differential

Differential backups store all the data changed since the last full data backup.

Incremental

An incremental backup stores the data changed since the last data backup - either the last complete data backup or the last delta backup (incremental or differential).

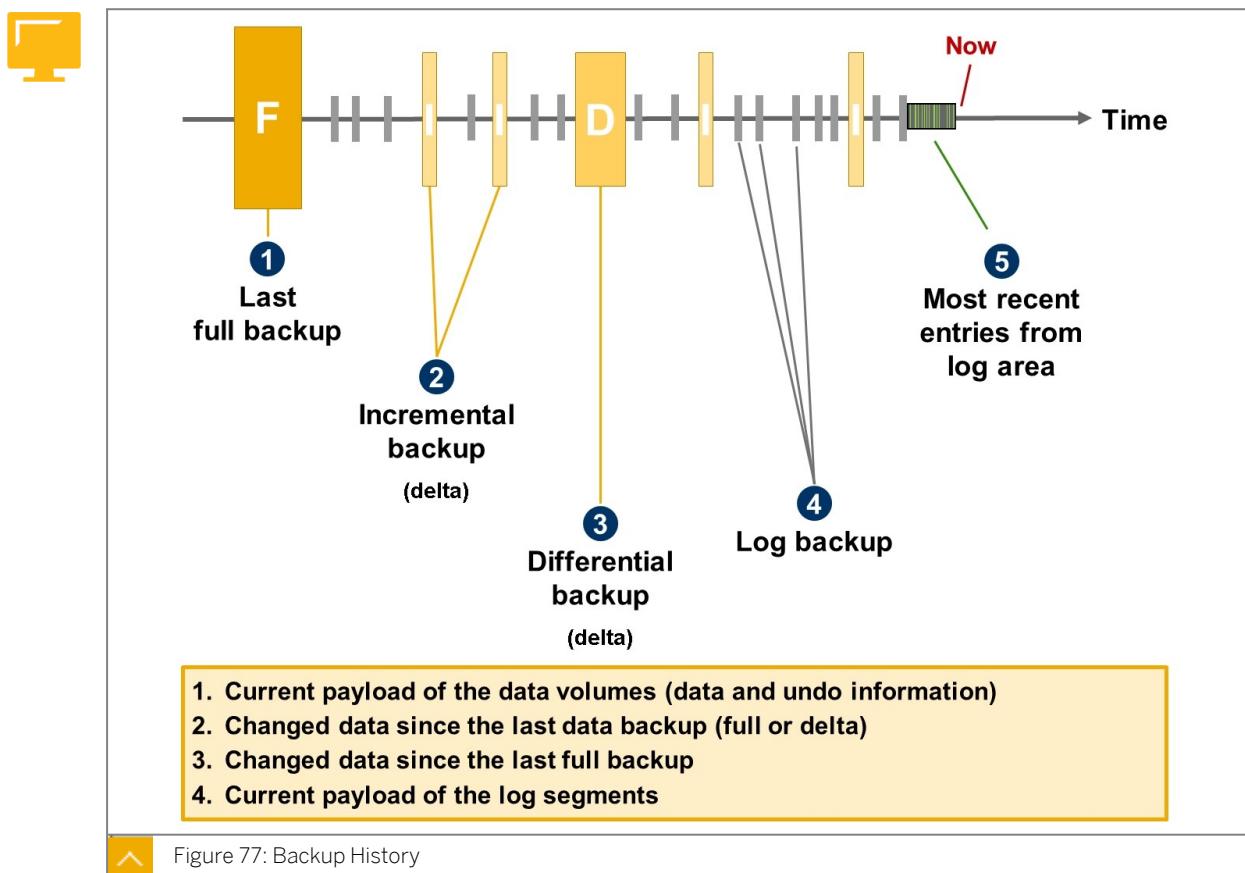
By default, SAP HANA log segments are backed up automatically.

Log segments are backed up for each service that has persistence. When a log backup is created, only the actual data (the "payload") of the log segments is written from the log area to service-specific log backups in the file system or to a third-party backup tool.

Scenarios for Recovering a Database

Recovery Types

The figure, *Backup History*, shows an overview of the possible backup types during normal operation.

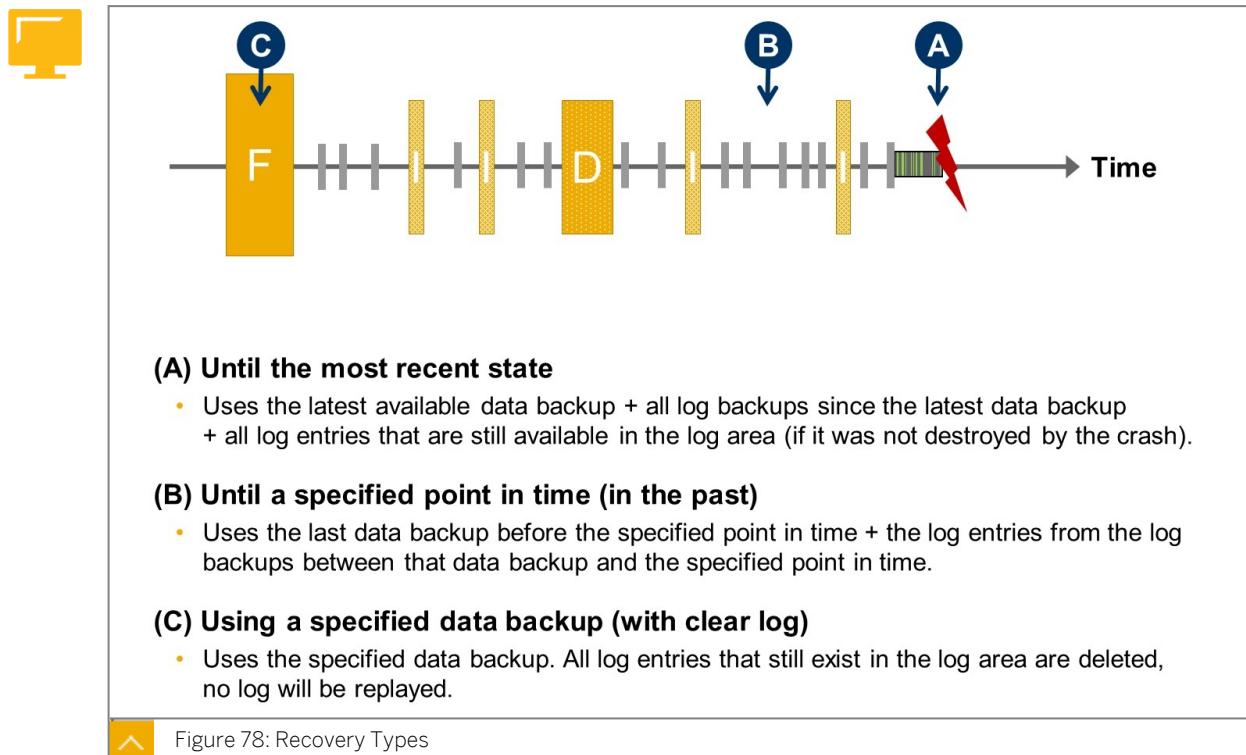


To recover an SAP HANA database, the database needs to be shut down. During recovery, the database therefore cannot be accessed by end users or applications.

It may be necessary to recover the SAP HANA database in the following situations:

- A disk in the data area is unusable.
- A disk in the log area is unusable.

- As a consequence of a logical error, the database needs to be reset to its state at a particular point in time.
- You want to create a copy of the database.



There are different types of recovery:

- **Most Recent State:** Used for recovering the database to the time as close as possible to the current time. For this recovery the data backup and log backup have to be available since last data backup and log area are required to perform this type of recovery
- **Point in Time:** Used for recovering the database to a specific point in time. For this recovery the data backup and log backup have to be available.
- **Specific Data Backup:** Used for recovering the database to a specified data backup. A complete data backup is required to successfully execute this recovery option.
- **Specific Log Position:** This recovery type is an advanced option that can be used in exceptional cases where a previous recovery failed.

Requirements for Performing a Recovery:

- The SAP HANA database must be shut down.
- The user must have the system privilege BACKUP ADMIN.
- The <sid>adm operating system user is required.
- At least one data backup is required.
- Before a recovery can start, all data and log backups must be either accessible in the file system or available through a third-party backup tool.

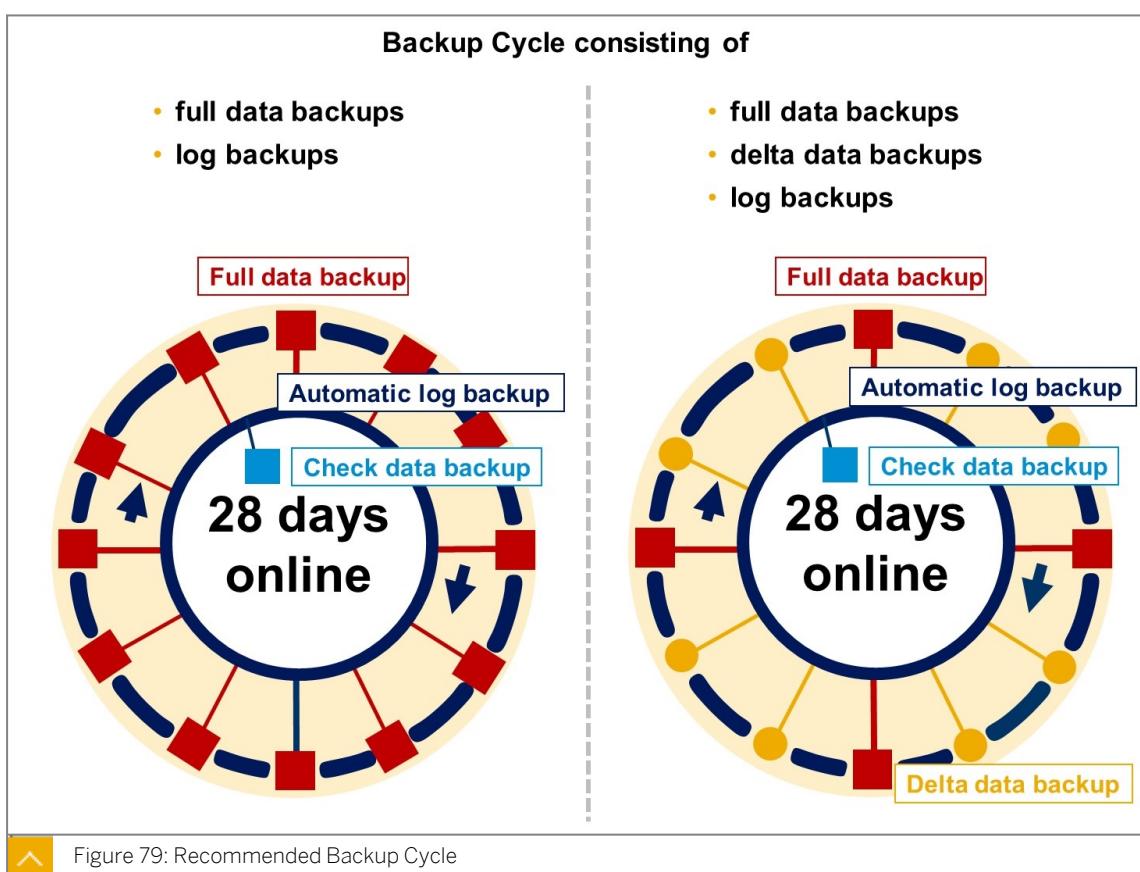
- The number and type of the SAP HANA services in source and target systems must be identical.

**Note:**

An SAP HANA database cannot be recovered to an SAP HANA database with a lower software version. The SAP HANA database software version used for the recovery must always be the same version or higher than the SAP HANA database used to create the backup.

Backup Cycle

As the figure *Recommended Backup Cycle* shows, we recommend a backup cycle of at least 28 days in duration. This means you should keep the backup media for the data and log information for at least 28 days before overwriting them.



We recommend that you perform a full data backup (of all data) at least once a week or, if possible, daily.

Delta data backups (incremental or differential) should be done after more extensive system activities during the week.

The SAP HANA database system executes automatic log backups at least every 15 minutes (default configuration setting).

Doing so, this creates four backup generations in a 28-day cycle.

We recommend that you always have two different backup media containing the same information.

Perform a data and log information backup with verification of the backup media at least once in the backup cycle.



LESSON SUMMARY

You should now be able to:

- Explain the purpose and architecture of databases
- Describe SAP HANA backup and recovery principles

Scheduling and Monitoring Database Actions

LESSON OVERVIEW

This lesson presents the DBA Cockpit, the central point for the monitoring and administration of databases in the SAP environment.

The DBA Cockpit Planning Calendar is used to schedule regular backups of data and log information.

Business Example

To protect the SAP system against data loss if an error occurs, the administrator regularly performs backups.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the use of the DBA Cockpit to centralize database administration tasks
- Schedule and monitor a database backup with DBA Cockpit
- Perform a batabase backup with SAP HANA Cockpit

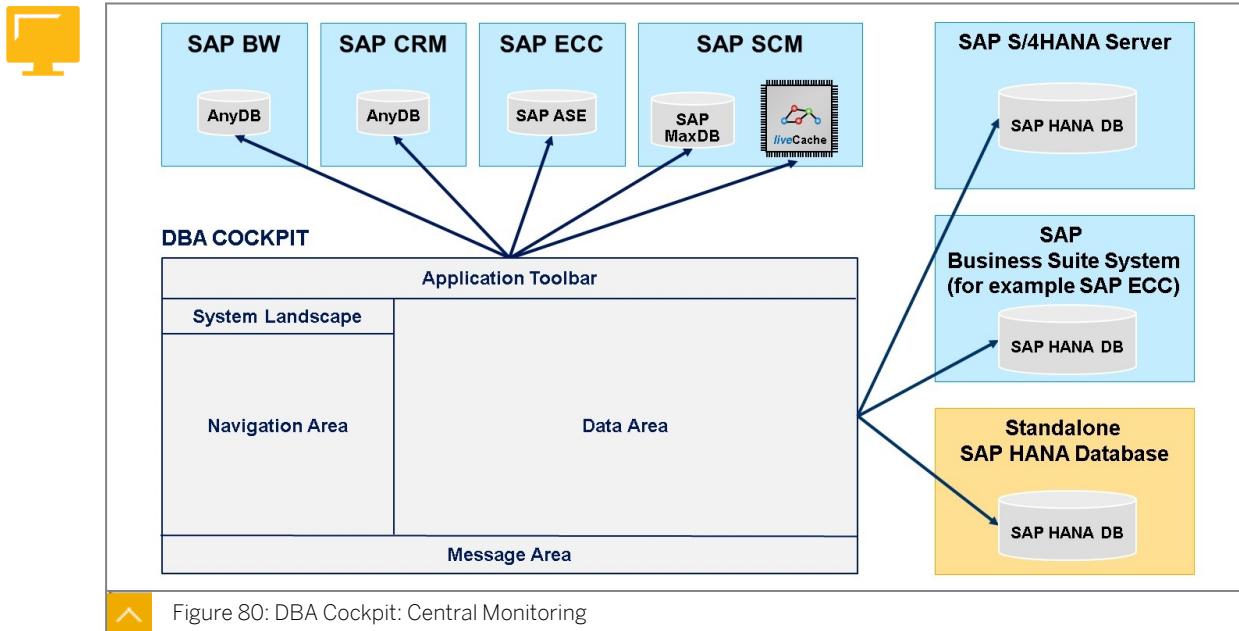
DBA Cockpit - Central Monitoring

The DBA Cockpit is a platform-independent tool that you can use to monitor and administer your database. It provides a graphical user interface (GUI) for all actions and covers many aspects of handling a database system landscape.

You access the DBA Cockpit by calling the transaction **DBACOCKPIT**. Then, the initial screen *DBA Cockpit: System Configuration Maintenance* appears.

As the figure *DBA Cockpit: Central Monitoring* shows, you can monitor and manage external databases, including both ABAP and non-ABAP systems, in the DBA Cockpit. All database systems supported by SAP can be connected as external databases.

A functioning secondary database connection is required to connect external databases. It must be configured beforehand.



The DBA Cockpit screen is comprised of the following panes:

- **Application Toolbar:** Common used functions like refresh the screen, or to display or hide the system landscape toolbar and the navigation frame.
- **System Landscape Toolbar:** Central functions to manage the database system landscape, like:
 - Managing database connections and DBA Cockpit configurations
 - Choosing the database to be monitored
 - Setting up and configuring the system landscape
- **Navigation:** Provides quick access to database monitoring and administration functions.
- **Data Area:** Information about the currently selected function.
- **Message Area:** Messages provided by the framework.



Hint:

We recommend using SAP Solution Manager to connect the DBA Cockpit, to ensure that you can use all the database monitoring functions in SAP Solution Manager. Also see SAP Note 1027146 – Database administration and monitoring with the DBA Cockpit – and the SAP Notes referenced there.

The DBA Cockpit functionalities

To minimize system downtime and improve performance, you schedule regular administrative jobs such as backups, database checks, and monitor your database daily basis. The DBA Cockpit is the central point for the monitoring and administration of all SAP-supported databases in the SAP system.

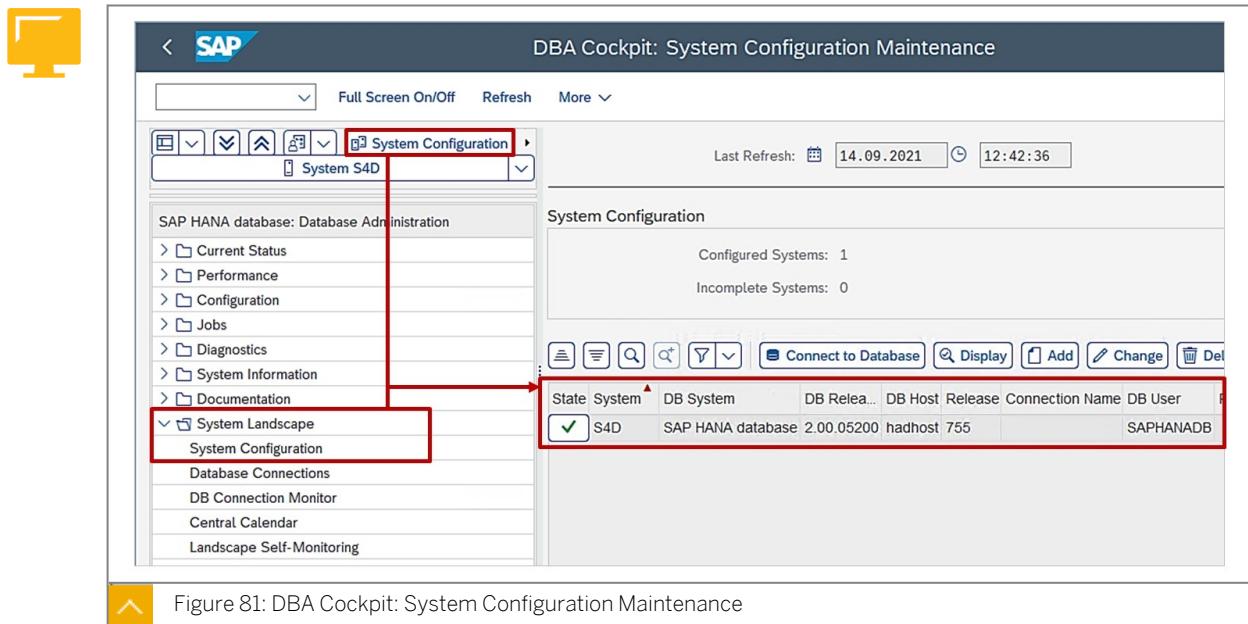


Figure 81: DBA Cockpit: System Configuration Maintenance

After a new SAP ABAP system has been installed, a connection to the local database is automatically set up in DBA Cockpit. The local database can be monitored and administered without any additional configuration steps.

To display the available database connections, from DBA Cockpit, choose *System Configuration*.

The screen *DBA Cockpit: System Configuration Maintenance* displays a list of all systems available, showing the current system status.

The above figure shows one configured system, which is the local SAP HANA tenant database.

The Connection Name column is empty because the "primary" connection is used here.



Note:

For more information please see *SAP Help Portal* → *SAP S/4HANA* → *DBA Cockpit for SAP HANA*.

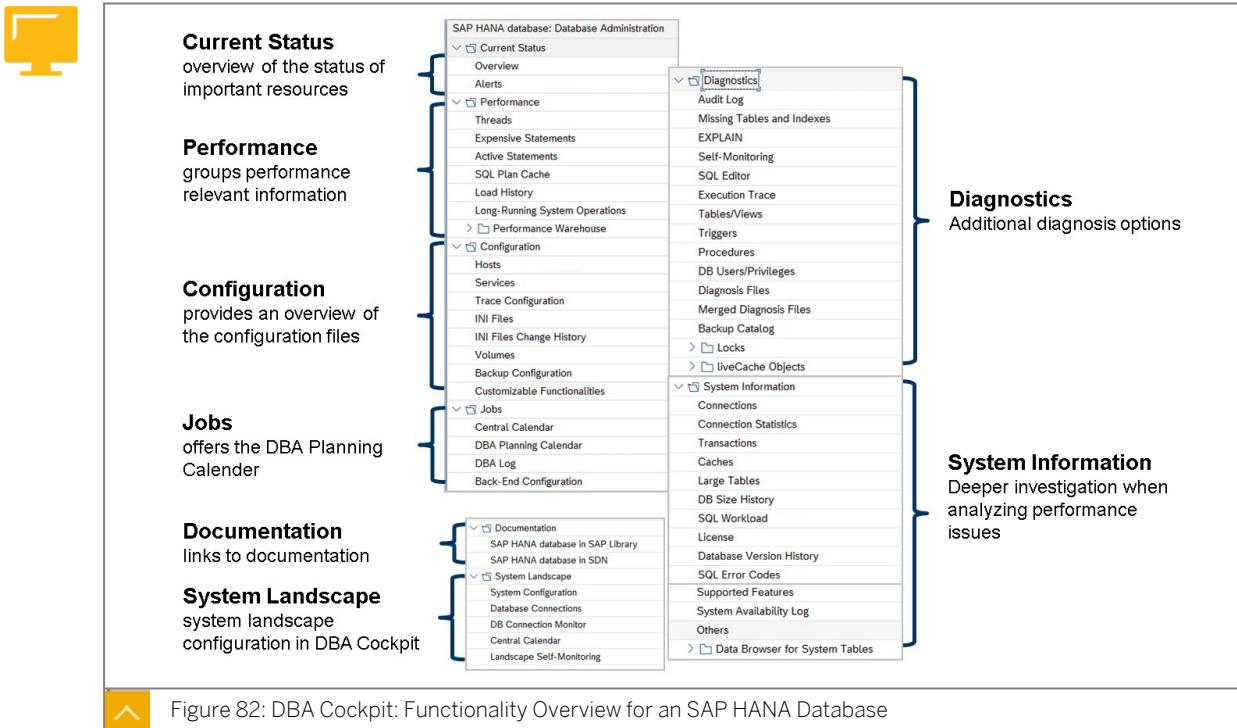


Figure 82: DBA Cockpit: Functionality Overview for an SAP HANA Database

As the figure DBA Cockpit: Functionality Overview for an SAP HANA Database shows, the menu structure of the DBA Cockpit (transaction `DBACOCKPIT`) is tailored for monitoring, administering, configuring and optimizing the relevant installed database. Its look and feel is similar to all SAP supported databases, but its functionality depends on the underlying database.

The functions of the DBA Cockpit can be divided into distinct areas, as follows:

- You get an overall status of your database and usage of important resources like memory, CPU and disk consumption using the navigation area *Current Status* → *Overview* folder item.
- You schedule backups and perform other administrative tasks in your database system using the *Jobs* → *DBA Planning Calendar* and the *Central Calendar*. With these calendars, you can schedule and monitor backups and administrative activities locally or centrally for several SAP systems and databases.
- The configuration settings of your database, services, traces, data and log volume statistics and backups can be administered and monitored, in the *Configuration* folder.
- The most important key figures for database performance display in the *Performance* and *System Information* folders.
- You monitor the storage behavior of the database (such as space statistics showing the history of the database or size and free space) in *System Information*. Monitoring the status of the database objects can be done mainly in *Diagnostics*.



Note:

For more information about using the SAP ABAP transaction DBACOCKPIT in SAP HANA environments, please see SAP Note: [2222220](#) - FAQ: SAP HANA DBACOCKPIT – and the SAP Notes referenced there.

Backup Scheduling and Monitoring using DBA Cockpit

In the SAP system, you can schedule and monitor regular database backups with the DBA Cockpit Planning Calendar.

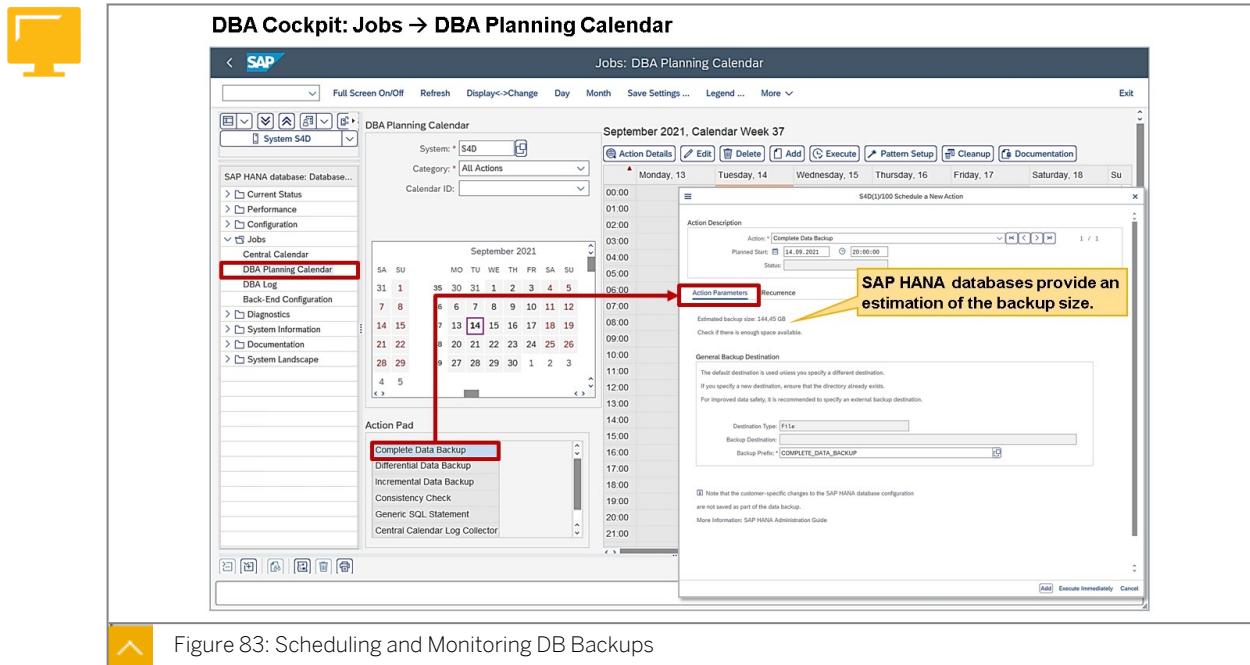


Figure 83: Scheduling and Monitoring DB Backups

The figure *Scheduling and Monitoring DB Backups* shows the interface of the DBA Planning Calendar. Here, you select a suitable data backup procedure for your scenario, and determine the repeat pattern and interval. We recommend you use the complete data backup procedure that is specific to your system, as shown in the figure *DBA Planning Calendar Action Pattern*. Choose the start time or an immediate start.

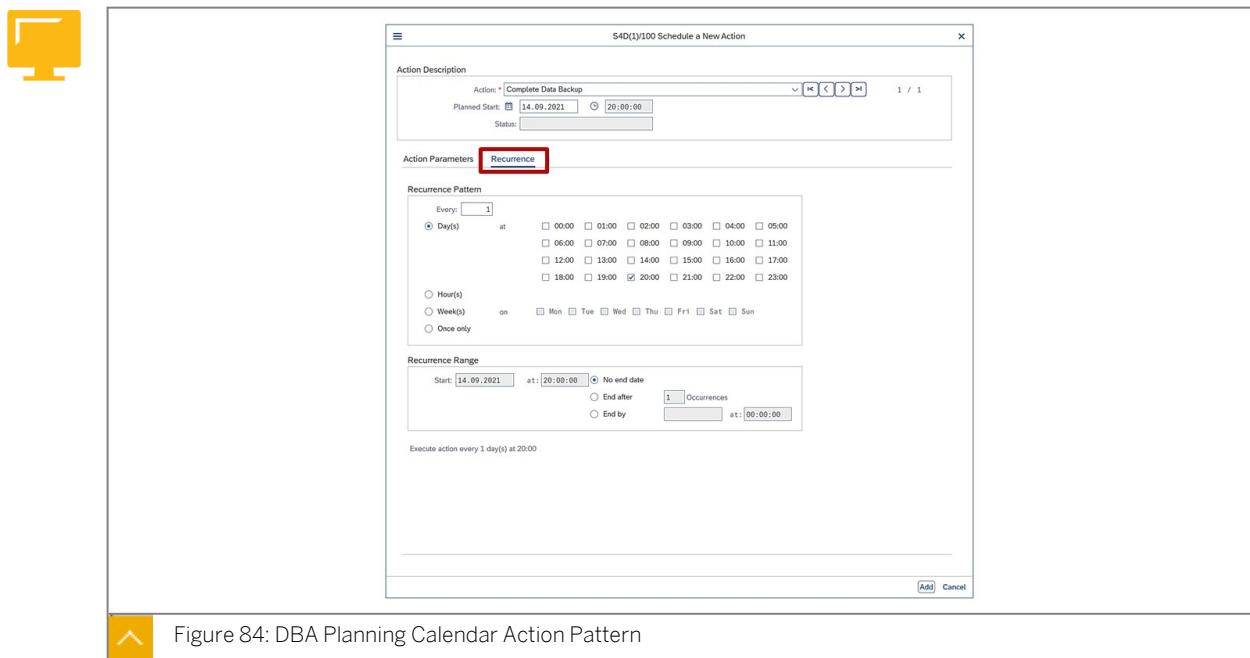


Figure 84: DBA Planning Calendar Action Pattern

Check every day to see if your backups have been successfully completed. In the planning calendar, a completely successful backup displays with a green label. A light yellow label means the backup completed with warnings. A red label indicates an erroneous, and therefore unusable backup. For more information, see *Jobs → DBA Log* in the DBA Cockpit.

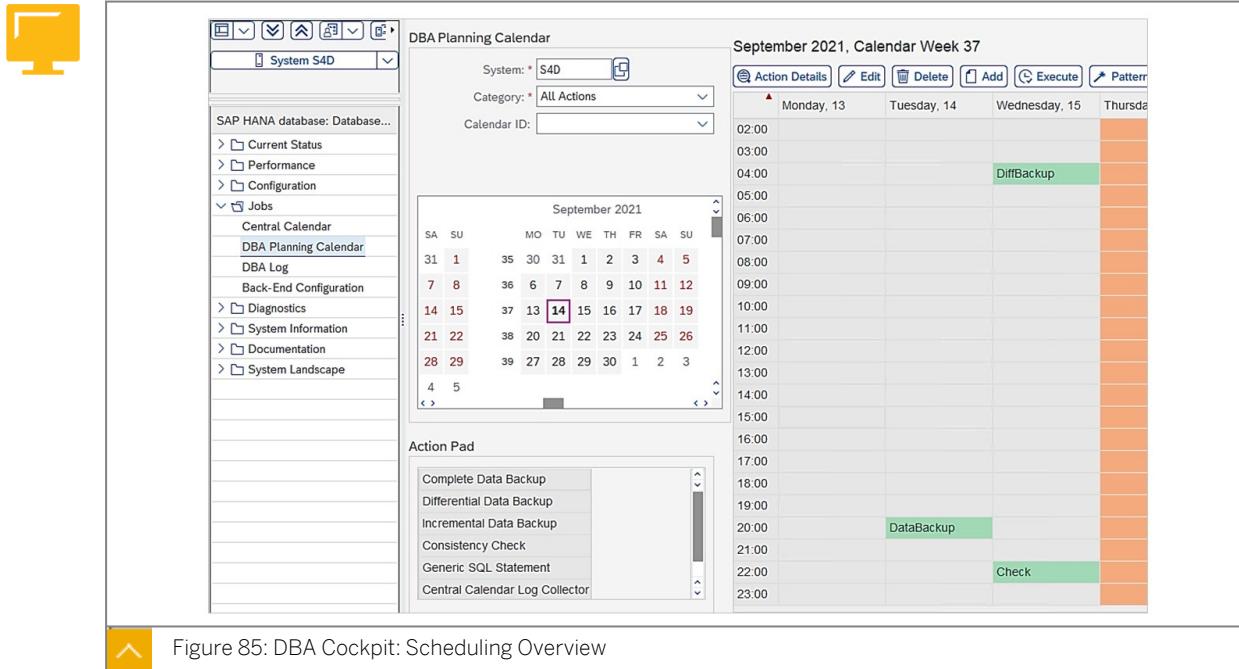


Figure 85: DBA Cockpit: Scheduling Overview

You should also regularly check the status of your scheduled periodic actions, such as backups, consistency checks or in case you are using a non SAP HANA database system, for example statistics generation executions. In the event of problems or warnings, analyze the respective logs either in the Planning Calendar or in the DBA Cockpit log overview, as shown in the figure *DBA Cockpit: DBA Log*.

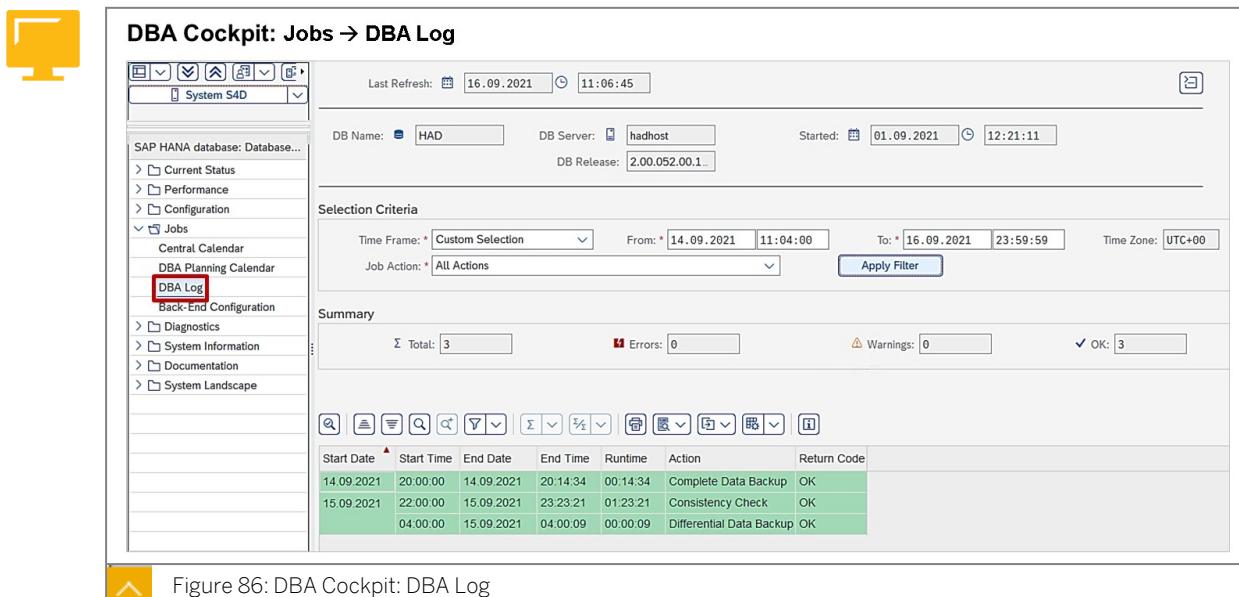


Figure 86: DBA Cockpit: DBA Log



Note:

Every database system also has its own trace and log files to check for possible errors.

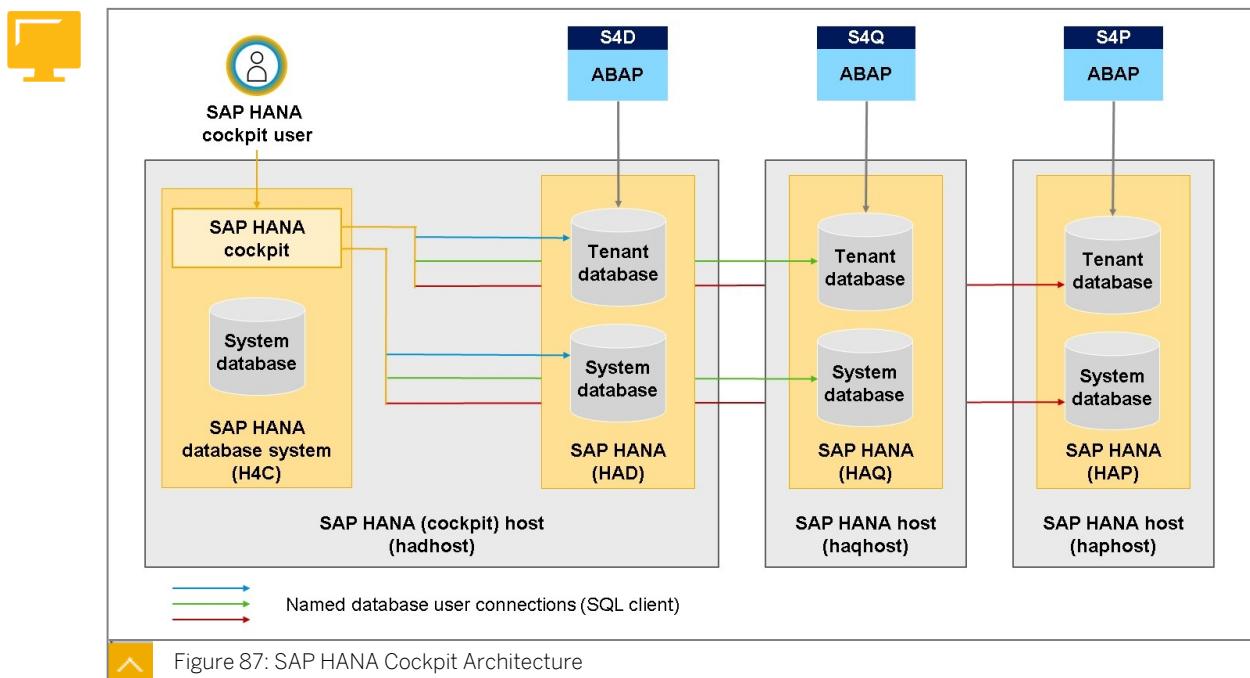
Performing Backups Using SAP HANA Cockpit

Introduction to SAP HANA Cockpit 2.0

SAP HANA cockpit is the main native administration tool for SAP HANA.

The SAP HANA Cockpit provides functions for aggregation, system and database administration, for example, database monitoring, user administration, and data backup. Administrators can use the SAP HANA Cockpit to start and stop services, monitor the system, configure system settings, and manage users and authorizations in multitenant database container (MDC) systems.

The SAP HANA cockpit administrator uses the SAP HANA cockpit manager tool to register databases, create SAP HANA cockpit users and database groups that other cockpit users can access within the SAP HANA cockpit.

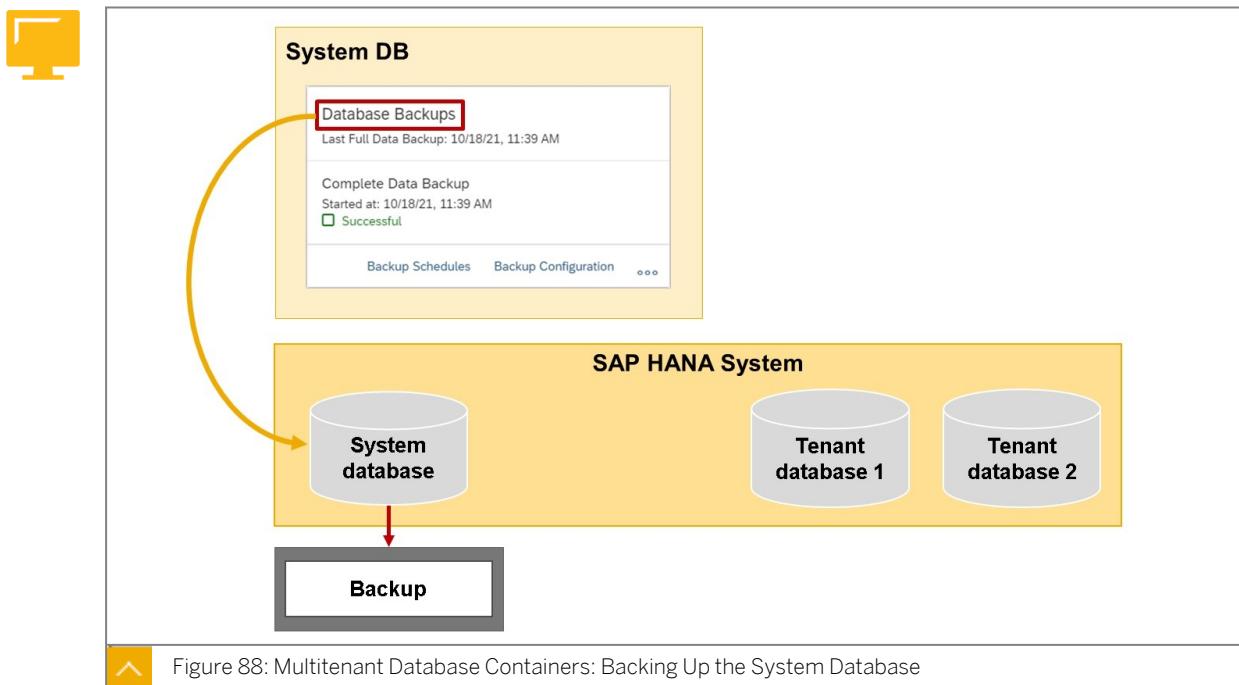


Backup of a Multitenant Database Container System

Using SAP HANA cockpit, you can create full data backups and delta data backups (differential backups and incremental backups). You need to create a backup for the system database and all tenant databases.

Data backups of the system database are needed on a regular basis.

The system database contains information about the system as a whole and all tenant databases. It is used for central system administration.



To perform a data backup of the system database with SAP HANA cockpit, choose the system database in the databases list of SAP HANA cockpit and choose *Database Backups*. Specify your backup settings and start the backup.

Because data backups of the system database only contain information about the system as a whole, data backups of the tenant databases are also needed on a regular basis.

The tenant databases contain the business data.

A data backup of a tenant database with SAP HANA cockpit, can be performed either using the system database or depending on the system configuration, it can also be initiated directly from a tenant database.

Performing Data Backups Using SAP HANA Cockpit

As you can see in the figure *Backup with SAP HANA Cockpit*, the backup wizard also shows the estimated backup size.

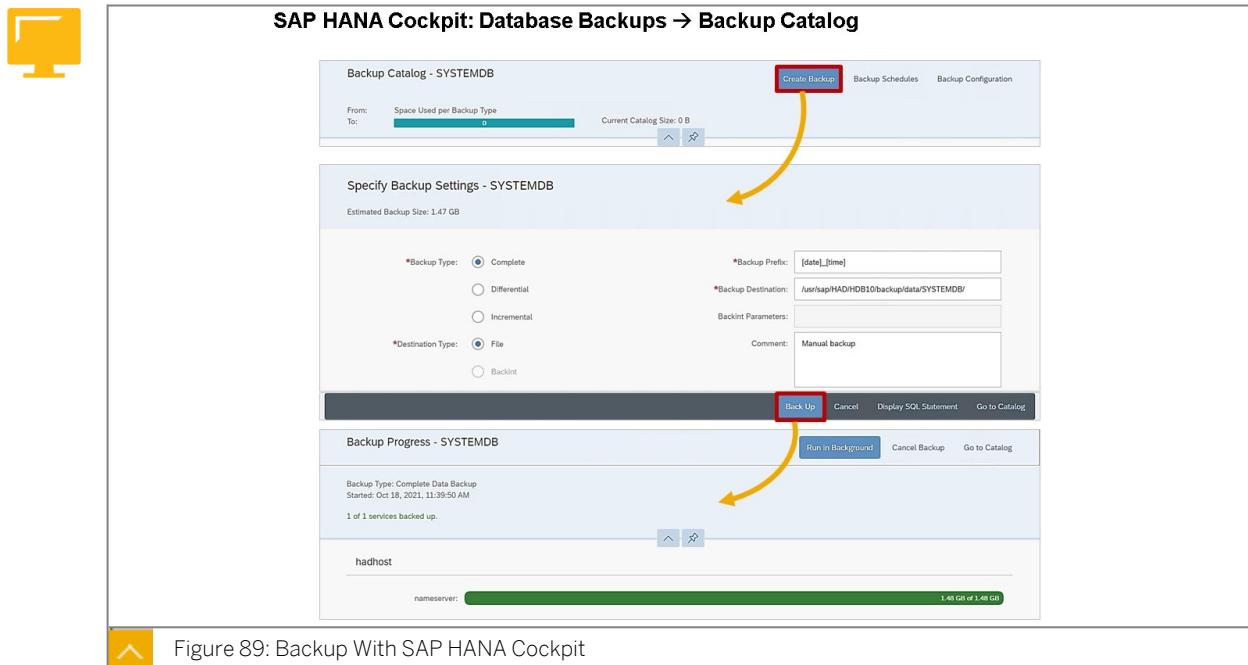


Figure 89: Backup With SAP HANA Cockpit

To create a data backup using SAP HANA Cockpit, perform the following steps:

1. In the SAP HANA cockpit, choose the *Database Backups* card.
2. To open the backup settings page, choose the *Create Backup* button at the top of the *Backup Catalog* screen.
3. Select the type of the data backup from the following options:
 - Complete Data Backup
 - Differential Data Backup
 - Incremental Data Backup
4. Specify the backup destination and the backup file prefix.
 - By default, SAP HANA cockpit uses the date and time stamp as the backup prefix.
 - The default backup destination shows the path specified in the SAP HANA global.ini file under the backup parameter basepath_databackup.
5. Select the *Back Up* button, to start the backup.

Once you have started the backup, the progress is displayed.

Overview of Backup Operations

After the backup is finished, the backup details are shown in the Backup Catalog.

Backup Catalog - SYSTEMDB

From: Oct 18, 2021, 11:37:05 AM To: Oct 18, 2021, 11:39:57 AM

Space Used per Backup Type: 100% Current Catalog Size: 4.41 KB

Filtered by: Start Time (Last four weeks)

Status	Start Time	Backup Type	Duration	Size	Destination Type	Comment												
Successful	10/18/21, 11:39 AM	Catalog Backup	< 1 second	4.41 KB	File	> X												
Successful	10/18/21, 11:39 AM	Complete Data Backup	6 seconds	1.48 GB	File	Manual backup > X												
Success	Backup Details - SYSTEMDB																	
Success	Backup ID: 1634557190478																	
Success	Status: Successful Type: Complete Data Backup Backup ID: 1634557190478 Size: 1.48 GB Backup Prefix: 2021_10_18_11_39_50 Destination Type: File Comment: Manual backup																	
Started: Oct 18, 2021, 11:39:50 AM Finished: Oct 18, 2021, 11:39:57 AM Duration: 6 seconds Throughput: 253.33 MB/s Location: /usr/sap/HAD/HDB10/backup/data/SYSTEMDB/ System ID: SYSTEMDB@HAD Encryption Key Hash: Additional Information:																		
Backup Parts <table border="1"> <thead> <tr> <th>Host</th> <th>Service</th> <th>Size</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>hadhost</td> <td>nameserver</td> <td>1.48 GB</td> <td>2021_10_18_11_39_50_databackup_1_1</td> </tr> <tr> <td></td> <td>nameserver</td> <td>5.64 KB</td> <td>2021_10_18_11_39_50_databackup_0_1</td> </tr> </tbody> </table>							Host	Service	Size	Name	hadhost	nameserver	1.48 GB	2021_10_18_11_39_50_databackup_1_1		nameserver	5.64 KB	2021_10_18_11_39_50_databackup_0_1
Host	Service	Size	Name															
hadhost	nameserver	1.48 GB	2021_10_18_11_39_50_databackup_1_1															
	nameserver	5.64 KB	2021_10_18_11_39_50_databackup_0_1															

Figure 90: Overview of Backup Operations



LESSON SUMMARY

You should now be able to:

- Explain the use of the DBA Cockpit to centralize database administration tasks
- Schedule and monitor a database backup with DBA Cockpit
- Perform a batabase backup with SAP HANA Cockpit

Unit 4

Lesson 3

Performing Regular Database Checks

LESSON OVERVIEW

There may be a large number of periodic checks that need to be performed in addition to making backups, depending on which database you are using.

Business Example

To ensure good database performance and, therefore, good SAP system performance, the administrator performs additional checks of the database, which can be scheduled regularly.



LESSON OBJECTIVES

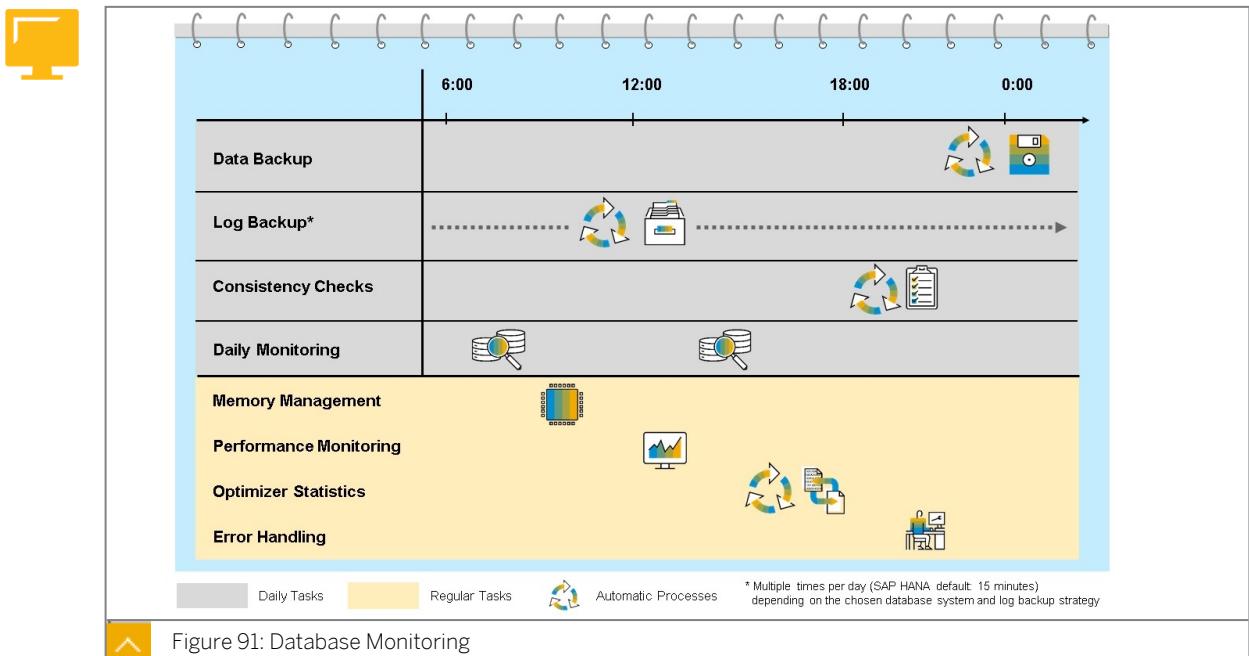
After completing this lesson, you will be able to:

- Describe examples of periodic database checks

Regular Monitoring of the Database

Monitoring databases

As the figure *Database Monitoring* shows, in addition to daily monitoring of database backups, other checks must be performed regularly. Some of these checks must be scheduled in the planning calendar in the DBA cockpit. The specific tasks depend on the database.



Examples of database checks include the following:

- Monitor database growth (for example, free space).

- Check for general database problems or errors.
- Monitor scheduled periodic actions (for example, backups).
- Check the consistency of your database.
- Generate statistical data to ensure good performance when accessing data records.

**Note:**

The SAP HANA database server decides the best refresh strategy for statistical data based on the data source. For example, for data statistics objects on column store data sources, the data statistics object is refreshed automatically when underlying data changes.

The figure *SAP HANA Database Size History*, shows the aggregated database size. You can adjust the aggregation period between day, month, and year. In addition to monitoring the database growth, you can also monitor performance indicators like the used memory, which is beside the CPU usage, one of the most important KPIs in an SAP HANA database. For a comprehensive insight into more monitoring aspects of the SAP HANA database system, we recommend that you use the SAP HANA cockpit or the SAP Solution Manager.

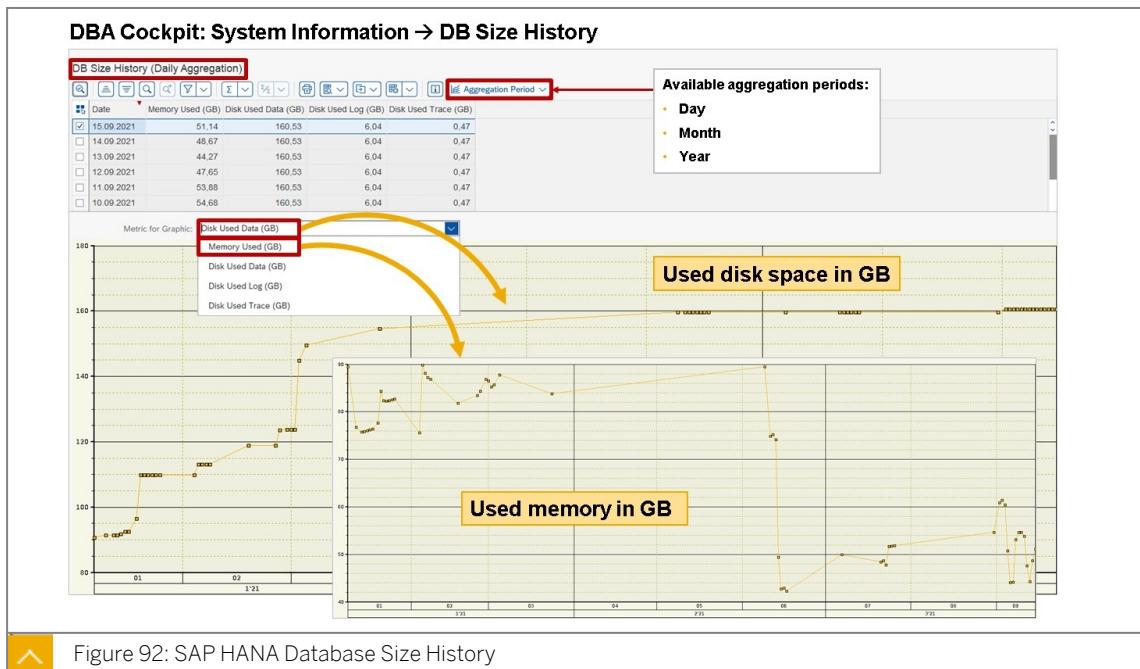


Figure 92: SAP HANA Database Size History

The statistics service is part of the internal monitoring infrastructure of the SAP HANA database. It is continuously collecting and evaluating information about status, performance, and resource usage from all components of the SAP HANA database.

DBA Cockpit provides an overview of current alerts that the statistics service reports when usage thresholds for database resources are exceeded.

The screenshot shows the DBA Cockpit interface with the following sections:

- DBA Cockpit: Current Status → Alerts**: Shows a table of current alerts with columns: Category, Alert ID, Description, and User Action. One alert (Alert ID 37) is highlighted with a red border.
- Indicates a possible solution to the problem**: A callout points to the User Action for Alert ID 37, which suggests performing a backup as soon as possible.
- Provides all Alert definitions**: A callout points to the Alert Check Information section, which lists various alert categories and their descriptions.
- Alert Check Information**: A table with columns: Category, Alert ID, Alert Name, Last Check Time, Interval, Next Check Time, Max. Priority, Description, and User Action. The table includes rows for Availability, Backup, and other system-related alerts.

Figure 93: Database: Alert Monitoring

As shown in the above figure *Database: Alert Monitoring*, you can examine *Current Alerts*, change to *Display Old Alerts*, where you can select a time frame for which to display alerts. Double-clicking on a row in both views provides you with more information about a specific alert.

The *Alert Check Information* section represents an overview of the checks performed in the system.

All alerts are supported by green, yellow, and red traffic light indicators to help identify the severity of an alert status.

Checking database consistencies

The SAP HANA database system provides different types of technical consistency checks which can be scheduled or immediately executed using the DBA Planning Calendar.

Let us consider the following scenario as an example:

You are interested in a pro-active SAP HANA consistency check, or you already face symptoms that can indicate SAP HANA inconsistencies and want to check if and to what extent corruptions exist.

The consistency check, as shown in the figure *Performing an SAP HANA Column Tables Consistency Check*, is a procedure available in the SAP HANA database with which you can perform a range of consistency check actions on the database column tables. The frequency with which you do this depends on your scenario.



Caution:

It is not recommended to run catalog checks and perform DDL operations (for example, dropping users) simultaneously, since this may cause the check to return multiple errors. Either run the catalog check on the system copy or wait until other operations have completed. Only if you continue to receive errors, should you contact the SAP Support.

To execute the check you specify:

- The *Action*, choose the (F4) key to display an overview of available checks
- The *Schema*
- The *Table* to check

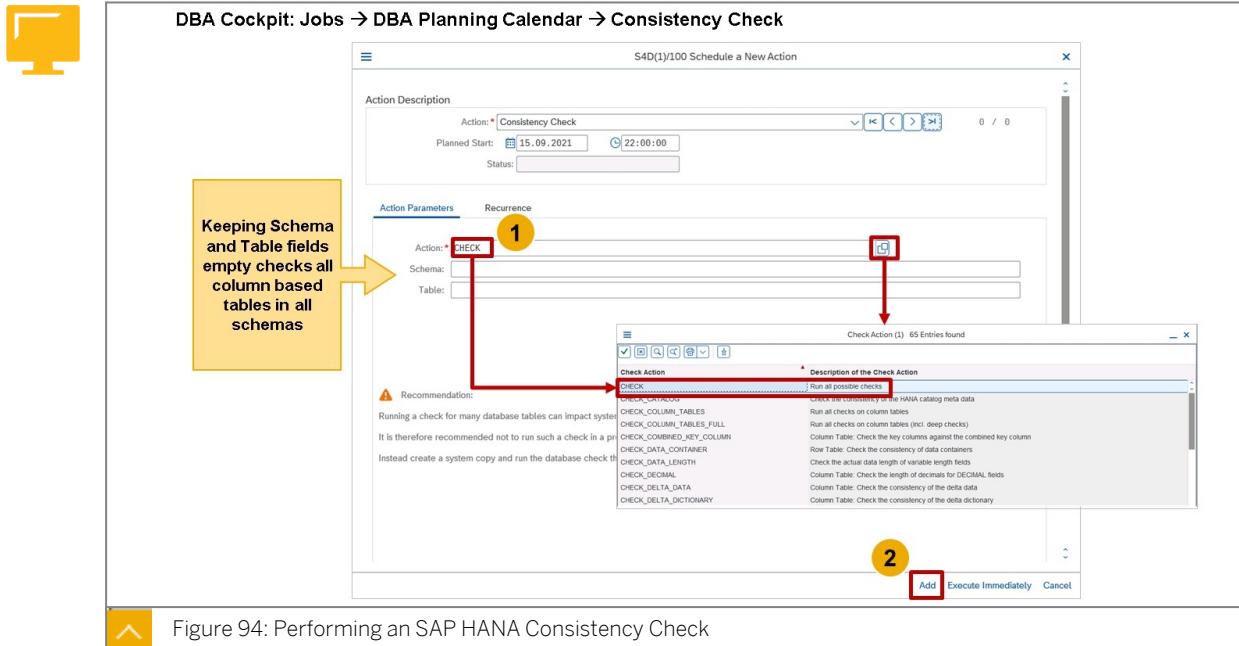


Figure 94: Performing an SAP HANA Consistency Check

After the check is completed, the status of the **Check** is shown in color.

- **Green** - No inconsistencies were detected.
- **Red** - Either: an error occurred, and the action was canceled OR: the action was completed, but one or more inconsistencies were detected in the SAP HANA system.

If the status of a completed consistency check is red, examine the job log.



Hint:

Performing a consistency check can negatively impact database performance. For this reason, you should consider scheduling a consistency check at times when the database is subject to a lower workload or performing the consistency check in a copy of a production SAP HANA system.

Updating table statistics for AnyDBs

When an SQL statement is executed, the database must select one of the possible access paths to the requested data. The WHERE clause in the SQL statement specifies the number of results. The database now has to find the relevant data records as quickly as possible, that is, with few read accesses. A database process, which is called "Optimizer" chooses which access method is used.

Under an AnyDB, the regular generation of statistics data is a primary prerequisite for the optimizer's decision, and thus for high-performance access to the data records. The database can either read the entire contents of a table (full table scan) or access a table through an index (index scan). Using the statistics, the cost-based optimizer of the database calculates the respective number of read accesses for all the possible access paths, and chooses the

best (most economical) access path. The figure *Determining the Optimal Access Method under AnyDB* illustrates this optimization process.

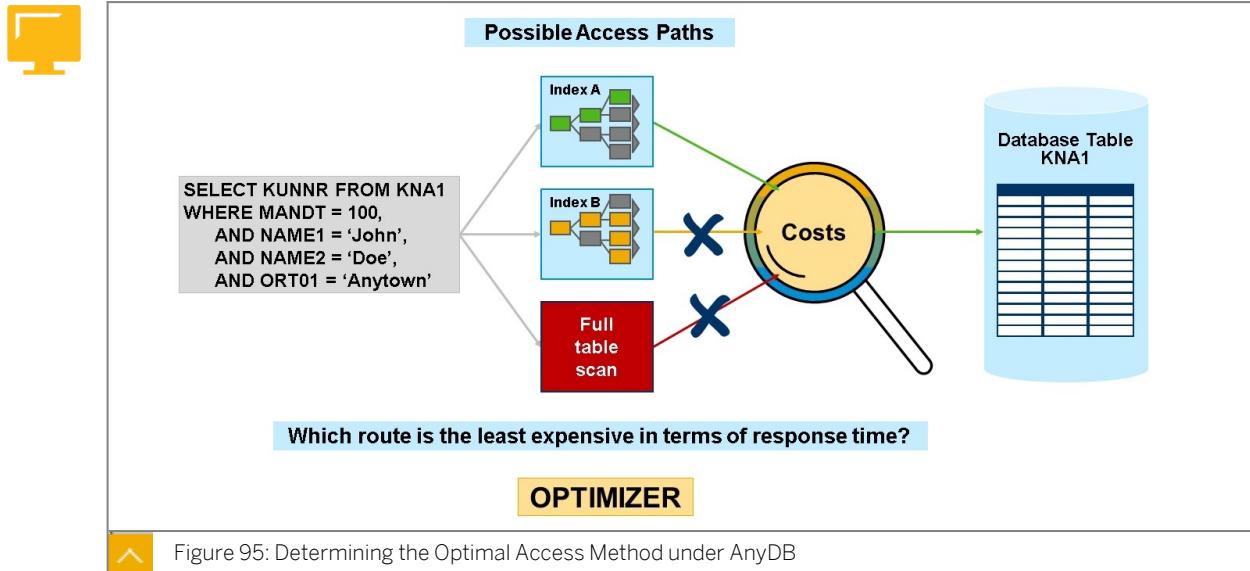


Figure 95: Determining the Optimal Access Method under AnyDB

The statistics contain information about the number of entries in the table, the number of blocks that are occupied by the table and the indexes, and the selectivity of the values of the individual table fields.

The recommended frequency for generating statistics (that is, how often) can vary depending on the database or the release you are using. In general, new statistics only have to be generated when a table has noticeably grown or shrunk in size. This is why the statistics are generated in two steps in the SAP environment. In the first step, a check is performed to determine whether statistics need to be generated for the table. To do this, the current number of data records is compared with the number of data records that existed the last time that statistics were generated. In the second step, statistics are generated for all tables whose size has changed noticeably. Depending on the database you are using, both of these steps are scheduled in one job or in two separate jobs.

The generation of statistics is extremely important for efficient access to data and should therefore be checked regularly by the administrator.

This lesson merely provides an initial overview of the various checks that the administrator has to perform regularly. For more information about these checks, contact your database vendor, consult the technical documentation, contact an SAP consultant, or attend one of our administration courses for the database that you are using.

SAP Notes for Database Consistency Checks

- SAP Note [1977584](#): Technical Consistency Checks for SAP HANA Databases
- SAP Note [2116157](#): FAQ: SAP HANA Consistency Checks and Corruptions
- SAP Documentation [DBA Cockpit for SAP HANA - Schedule Action: Consistency Check: SAP Help Portal](#)



LESSON SUMMARY

You should now be able to:

- Describe examples of periodic database checks

Appendix: Describing the Concept of SAP Data Archiving

LESSON OVERVIEW

Data archiving is used to remove large quantities of data that is no longer required in the system, but which is to be retained in an analyzable format. This lesson defines the term data archiving, and explains the reasons for implementing an archiving project. The process of data archiving will also be outlined.

Business Example

The management team at your company has decided to implement SAP data archiving. As a member of the system administration team, you want to learn about the fundamentals of data archiving.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the idea and benefits of data archiving
- Outline the technical process of data archiving and list options for accessing archived data

Removal of Data

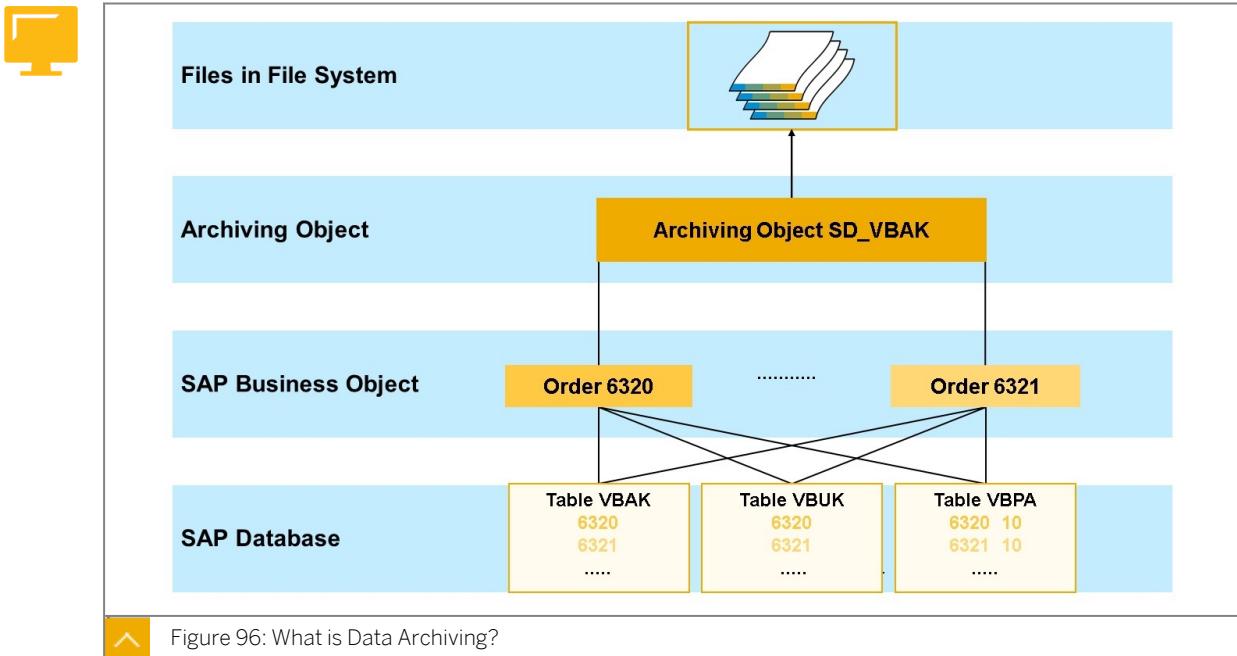
With current database systems, a large data volume can often lead to performance bottlenecks, which cause poor performance on the user side, and increased resource costs on the administration side. Data that is no longer required in the database for applications should therefore be removed from the database. However, it is often not possible simply to delete the data, as read access to this data is often still needed. The data must therefore be stored outside the database on external storage media so that it can later be read again.

Definition of Data Archiving

Data archiving is the consistent removal of data objects from database tables of the SAP system database, where all table entries that characterize a data object are written to an archive file outside the database. Business consistency is ensured by SAP archiving programs that store all relevant table entries together, on the basis of archiving objects.

The contents of business objects in the SAP system are distributed across multiple tables in the database. Archiving objects bundle logically connected tables of business objects. This ensures, in the context of data archiving, that **all** information for a business object is stored outside the database and therefore deleted from the database.

The figure *What is Data Archiving?* clarifies the connection between database tables, business objects, and archiving objects. The application data removed from the SAP system database is compressed and stored in a different location (such as a file system).


Hint:

From a technical point of view, archived data can be loaded back into the database. However, because it is historical data that has not been affected by changes to the database and its contents since it was removed, reloading the data could cause inconsistencies. You should only reload data into production systems in exceptional circumstances. For more information, see SAP Note [53064 – What is important when reloading from archives?](#).

The data is archived in online mode, which means you do not need to shut down the system during the archiving.

Since the term “archiving” is used in an undifferentiated way in the IT world, this lesson examines other important terms in the data archiving environment to create a distinction between these terms and data archiving. In particular, data archiving is **NOT** any of the following activities:

- Reorganization
- Backup / restore
- Document storage (usually using SAP ArchiveLink)
- Deletion of test data

The term reorganization has a dual meaning within the SAP world (historically, due to SAP R/2). Reorganization can mean the physical deletion of application data from the database. However, its actual meaning is reorganization of the database. During reorganization of the database, the data is unloaded from the database and then reloaded, to optimize the physical data distribution on the hard disk.

A backup is the backing up of the database contents in case of system breakdowns. The aim is to restore the database status to as close to the status that was current before the breakdown. Backups are usually performed at fixed time intervals and follow a fixed procedure. The process of reloading the backed-up data is called a restore.

Database backups are independent of the data archiving and should be scheduled and performed at regular intervals.

Document storage means the general electronic storage and management of documents on storage systems outside the SAP system. The documents are transferred to a document storage system. Only a pointer that links to the stored documents and allows access is administered in the SAP system.

SAP ArchiveLink is a data interface integrated in the SAP NetWeaver Application Server that controls communication with storage systems.

Only data from completed business processes can be archived. Test data rarely achieves this status.

Reasons for Data Archiving

The larger a database is, the higher the cost for database administration with regard to recovery or backup. Ensuring delivery of the SAP system services requires increasingly more resources. Upgrades of the system or database also become disproportionately more expensive with an increasing database size.

For end users, an increasing database size means a higher system load for individual database queries; that is, increasing response times for reporting and access to individual database records.



Hint:

Full table scans, in particular, are accelerated by data archiving, but access using indexes can also be appreciably accelerated by data archiving. The main effect in this case is that a reduction of the disk requirement for an index improves the quality of the database's data buffer. This effect is especially pronounced for indexes that are likely to be accessed using any of their fields. This includes all indexes that are not created in a chronologically sorted format (using the document number, for example).

The reasons for data archiving are summarized in the following list:

Why Archive Data?



- Improve response times or ensure good response times.
- Reduce costs for database administration.
- Reduce system downtime for software upgrades, recoveries, and (offline) data backups.
- Take into account legal and business requirements, including the following:



- Country-specific retention rules
- Data must be available to auditors
- Potential reusability of data

Data must be archived so that it can be called at any time for queries, such as those from tax authorities. Every country has statutory data retention requirements. In addition, only data from completed business processes can be archived.

With data archiving, data is archived independently of the hardware and software release statuses (metadata is stored). In this way, it is still possible to call archived data without any problems even after a system upgrade.

Cooperation Between the System Administration and User Departments

You should begin planning the archiving of application data during the implementation of SAP software. The volume of transaction and master data can increase very quickly when live operation starts.

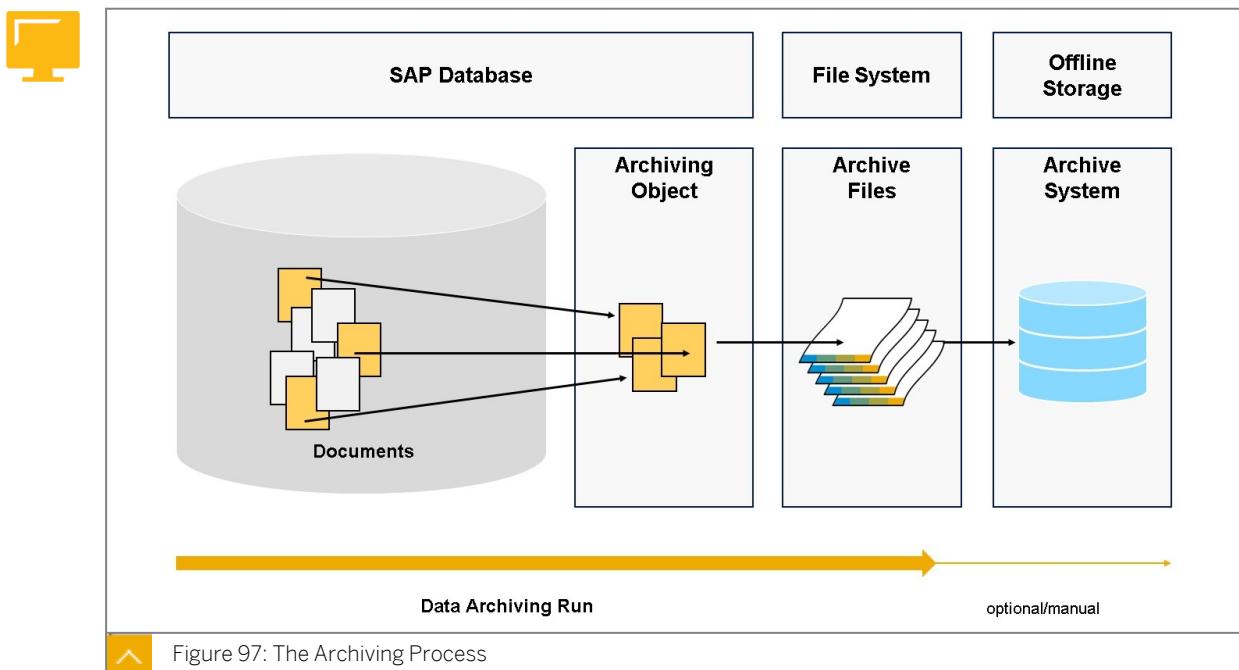
System administration's aim is to keep the database as small as possible; that is, to delete as many data objects as possible to reduce the database load. The aim of user departments, in contrast, is to be able to access as many documents as possible online for queries, lists, and reporting.

Data archiving is about finding a compromise between these two positions that satisfies both sides. The long-term aim must certainly be to maintain as constant as possible a volume of data in the database and to archive data proactively instead of reactively.

Schematic Process of a Data Archiving Run

The actual data archiving is a process in three steps:

1. Creating the archive file(s): In the first step, the write program creates one (or more) archive file(s). The data to be archived is then read from the database and written to the archive file(s).
2. Storing the archive file(s): After the write program has completed the creation of the archive file(s), they can be stored.
3. Deleting the data: The delete program first reads the data in the archive file and then deletes the corresponding records from the database.



You must also consider the fact that it is usually not sufficient to write the data to be archived to archive files and to delete it from the database. The archive files must be stored securely and be managed to ensure that they are accessible later if required.

As the figure *The Archiving Process* illustrates, there are several options for the storage of archive files (see step 2 in the list):

- Hierarchical Storage Management (HSM) systems: If an HSM system is used, it is sufficient to maintain the appropriate file path in Customizing (transaction FILE). It is up to the HSM system to store the files on an appropriate storage medium, depending on frequency of access and disk space utilization.
- Optical archiving: Documents are electronically stored and managed on storage systems outside the SAP system landscape. Documents are usually physically stored on optical media, such as CDs. This is where the name originates.
- Manual storage: After the deletion program has processed the relevant archive file, it can be manually recorded to tape.

Access to Archived Data

A major prerequisite for archiving application data is that this data must belong to completed business processes or periods and is therefore no longer needed for current business processes. However, it may be necessary to access this data again even after it has been archived, for example, in the case of a complaint, for evaluations, or for an internal or external audit. In this section, you will obtain an overview of the options for accessing archived data.

Access Options

The Archive Development Kit (ADK) stores data in such a way that read access to it is possible at any time. The prerequisite for this is the existence of appropriate read programs, which are made available by the relevant archiving object. They are used to read the archived data objects in accordance with the selection criteria and to display it in a suitable format for the user. In general, two types of access or display are used:

Ways to Access Archived Data

- Sequential access
- Direct access

Sequential (read) access is the simplest form of access to archived data. With this type of access, the read program first opens the archive files from an archiving run sequentially, reads the contents of all data objects, and lists the data that meets the entered selection criteria sequentially. You can use this method, for example, to list all data objects for a particular posting period or for a particular document number group.

Direct access to individual archived data objects, such as an order or billing document can only be achieved using an index table. The data object to be displayed is first selected in the index table using search terms. If this is successful, the archive file that contains the data object concerned is localized and opened using the index. The read program can now directly access and display the data object for which you have searched. This type of direct access is relatively costly from a technical point of view, and is, therefore, only offered by a small number of archiving objects.

The SAP Archive Information System provides much more extensive and easier to use functions for quick direct access to archived data objects.

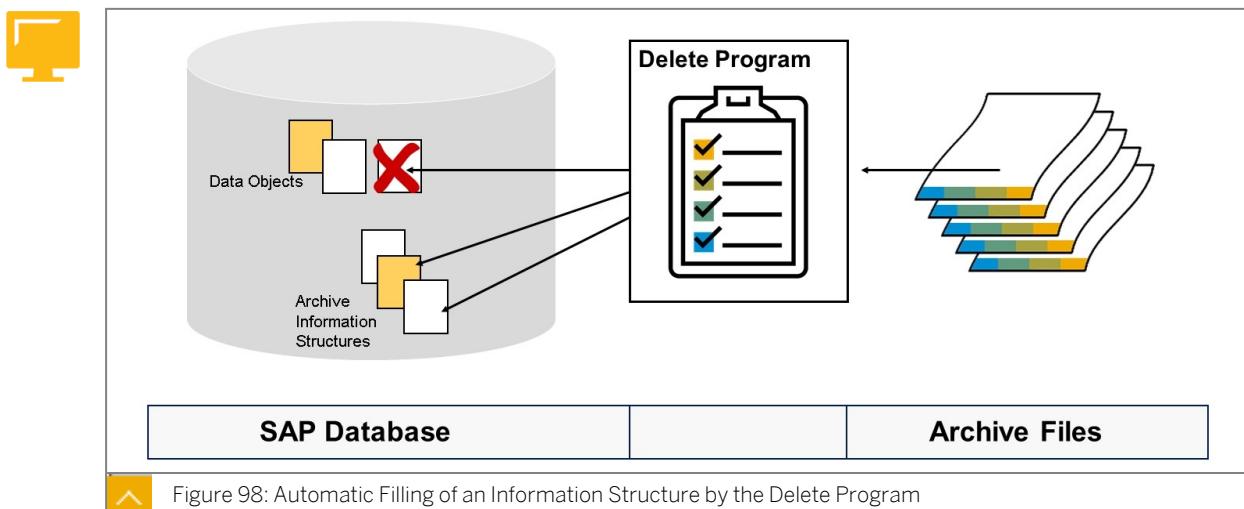
The SAP Archive Information System

The SAP Archive Information System (Archive Information System, SAP AS) is a tool for searching in SAP data archives that is fully integrated in the archiving environment. It offers you support when searching for archived data and provides functions for displaying this data.

The SAP Archive Information System is a generic tool. The available functions can be used for all existing archiving objects.

Data retrieval is always based on archive information structures. These are transparent database tables that are supplied with data from the archive. To search for archived data for an archiving object, there must be at least one archive information structure for this archiving object. An information structure of this type always includes the structure itself, the corresponding transparent database table, and an evaluation program.

Before it is created by filling it with data from the archive, an archive information structure is activated. The system generates a transparent database table and the evaluation program in the background. At the start of the delete run for an archiving object, all of the active information structures for this archiving object are filled. The process of filling the information structure is illustrated in the figure *Automatic Filling of an Information Structure by the Delete Program*.



When displaying archived data, the user can usually choose between views in which the data objects can be displayed, such as the technical view and the business view.

Related Information

For more information about data archiving, see:

- The online documentation, in the Introduction to Data Archiving section (CA-ARC)
- SAP course BIT660 (Data Archiving)
- SAP Support Portal: <https://www.sap.com/products/information-lifecycle-management.html>.



LESSON SUMMARY

You should now be able to:

- Explain the idea and benefits of data archiving
- Outline the technical process of data archiving and list options for accessing archived data

Learning Assessment

1. What is the minimum recommended cycle for full database backups for SAP databases?

Choose the correct answer.

- A Create two backup generations in a 28-day cycle.
- B Create four backup generations in a 28-day cycle.
- C Create six backup generations in a 28-day cycle.
- D Create eight backup generations in a 28-day cycle.

2. Which activities can you schedule using functions of the transaction DBACOCKPIT?

Choose the correct answers.

- A Restarts of the database system.
- B Recovery of the database system, for example, in sand box environments.
- C Backups of the database system.
- D Consistency checks of the database system.

3. What information does the optimizer of a database of type “anyDB” consider to decide on how to access data in the database?

Choose the correct answer.

- A Log data
- B Last change date
- C Statistics data
- D Number of previous accesses

4. Every archiving object in the SAP system is connected 1:1 with the content of a single database table.

Determine whether this statement is true or false.

- True
- False

Learning Assessment - Answers

1. What is the minimum recommended cycle for full database backups for SAP databases?

Choose the correct answer.

- A Create two backup generations in a 28-day cycle.
- B Create four backup generations in a 28-day cycle.
- C Create six backup generations in a 28-day cycle.
- D Create eight backup generations in a 28-day cycle.

You are correct! We recommend that you perform a full data backup (of all data) at least once a week, with a recommended backup cycle of at least 28 days. This means that you should keep the backup media for the data and log information for at least 28 days before overwriting them. Learn more about this topic in the lesson “Describing the Architecture of Database Systems” of the course ADM100.

2. Which activities can you schedule using functions of the transaction DBACOCKPIT?

Choose the correct answers.

- A Restarts of the database system.
- B Recovery of the database system, for example, in sand box environments.
- C Backups of the database system.
- D Consistency checks of the database system.

You are correct! You can use functions offered by transaction DBACOCKPIT to schedule backups and consistency checks for the database used by AS ABAP. It is not possible to schedule restarts or recovery activities of the database system. Learn more about this topic in the lesson “Scheduling and Monitoring Database Actions” of the course ADM100.

3. What information does the optimizer of a database of type “anyDB” consider to decide on how to access data in the database?

Choose the correct answer.

- A Log data
- B Last change date
- C Statistics data
- D Number of previous accesses

You are correct! The optimizer of a database of type “AnyDB” considers the statistics data to decide on how to access the data in the database. Log data, Last date of change and the number of previous accesses are not considered for this purpose. Learn more about this topic in the lesson “Performing Regular Database Checks” of the course ADM100.

4. Every archiving object in the SAP system is connected 1:1 with the content of a single database table.

Determine whether this statement is true or false.

- True
- False

You are correct! Different business objects could belong to each archiving object in the SAP system. The data for these business objects could, in turn, be distributed across multiple database tables. Learn more about this topic in the lesson “Appendix - Describing the Concept of SAP Data Archiving” of the course ADM100.

Lesson 1

Creating, Copying, and Maintaining User Master Records

177

Lesson 2

Maintaining User Authorizations with Roles and Profiles

187

Lesson 3

Define Profile Parameters to Secure Passwords and Logins

199

Lesson 4

Evaluate Users and Authorizations

209

UNIT OBJECTIVES

- Outline the different users in the SAP environment
- Explain the elements of user master records
- Describe the different user types
- Explain the importance of user groups
- Create users using the user maintenance transaction
- Describe the authorization concept of AS ABAP based SAP systems
- Explain the relation between authorization objects and authorization checks
- Create an authorization role using the role maintenance transaction
- Discuss the SAP Fiori content model
- Explain the authorization concept for SAP Fiori
- Identify profile parameters to secure user passwords and user logins
- Implement security policies
- State how to restrict user logons during maintenance work
- Name standard users in the SAP system

- Use the User Information System to obtain information about authorizations and users
- Execute the system trace for authorization checks to locate authorization problems

Creating, Copying, and Maintaining User Master Records

LESSON OVERVIEW

This lesson explores the administration of user master records. Creating, copying, and maintaining master records of this type will be described in more detail.

Business Example

The users of the SAP system require their own user IDs with appropriate authorizations to log on. The administrator sets up a user ID in the system for each user.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Outline the different users in the SAP environment
- Explain the elements of user master records
- Describe the different user types
- Explain the importance of user groups
- Create users using the user maintenance transaction

Users in the SAP Environment

The concept of user administration and the creation of a user master record are explained in detail in this lesson. The understanding of the concept is important to obtain a better understanding of SAP systems.

The figure *Users in the SAP Environment* illustrates how users interact with the SAP system.

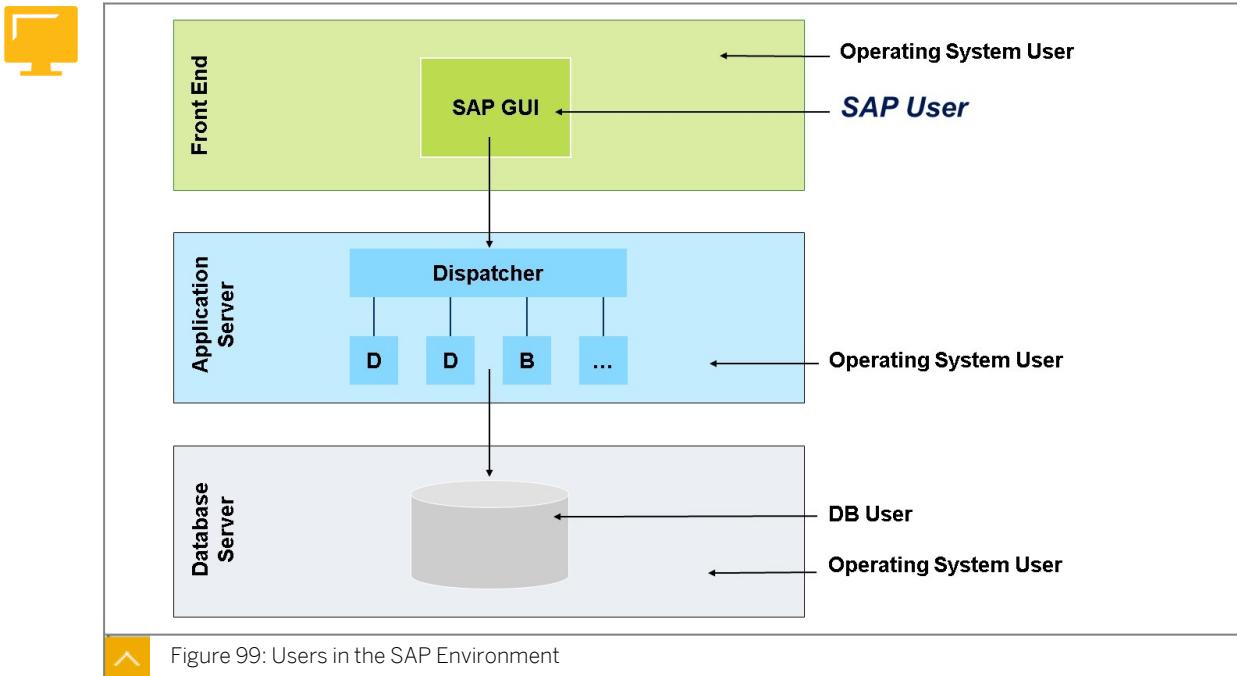


Figure 99: Users in the SAP Environment

Users can log on to an operating system, a database, or an SAP system using a combination of a user ID and a password. The operating system level, the database level, and the SAP system level usually have different authorization concepts.

In the SAP environment, a “user” usually refers to a user master record consisting of a **user ID**, **password**, and other user-related information.

Access to the operating system level of the application server and database server must also be protected to prevent damage to data and ensure that the SAP systems can be used. A user/password combination in the SAP system does not guarantee that you can log on to the operating system of a host or to the database server with the same user/password combination. However, identical user/password combinations can be created for SAP systems and operating systems.



Note:

In the SAP system, user requests are processed by SAP work processes. These work processes all use a common user to access the database.

This lesson describes how users can log on to a client of an ABAP-based SAP system. Users and authorization data are client-dependent.

Options for Creating a User Master Record

A user master record allows users to log on to an ABAP-based SAP system. You create user master records for each user with the user maintenance (transaction SU01).

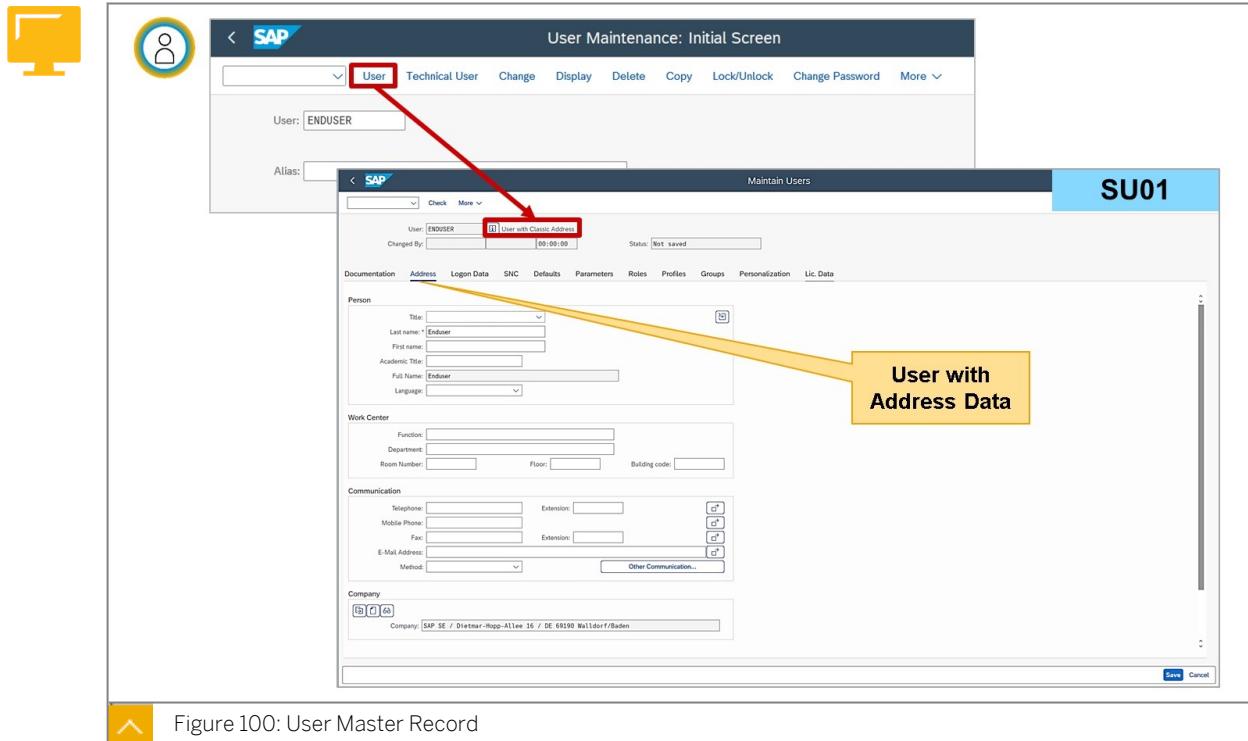


Figure 100: User Master Record

You can create a new user master record by creating a new master record or by copying an existing one. The user master record contains all data and settings that are required to log on to a client of the SAP system.

To create a user master record, at least the following information is required in SU01:

- *Last name* on the *Address* tab page.
- *New (initial) password* and identical repetition of the password on the *Logon Data* tab page.

Furthermore, you can provide additional data in SU01, as shown in figure *User Master Record*. This data is divided into the following tab pages:

- The *Documentation* tab contains a brief description and detailed documentation of the user. A person that has technical or functional responsibility for this user, for example, can also be entered on this tab.
- The *Address* tab displays address data.

**Note:**

With SAP S/4HANA, a new identity model for business users has been introduced. A business user is defined as a natural person who is represented by a business partner and a link to a user in the system. Business users interact with the software in the context of a business process, for example, in the role of a purchaser, a sales representative, or a production planner. A user in SAP S/4HANA is now related (1:1) to a corresponding business partner (person) from whom essential data is referenced. This reduces redundant maintenance and prevents the disadvantage you have with outdated information.

However, it is still possible to create users without business partner assignment. They are displayed as *User with Classic Address*.

For more details, see SAP Note [2570961](#) – *Simplification item S4TWL - Business User Management*.



The screenshot shows the SAP User Maintenance interface for transaction SU01. At the top, there's a navigation bar with tabs like 'User', 'Change', 'Display', etc. Below it, a sub-header says 'Maintain Users'. The main area displays a user record for 'RFC_USER' with the alias 'Technical User'. A red box highlights the 'User' field in the header and the 'User' field in the record. A yellow arrow points from the 'Documentation' tab in the sub-area to a callout box labeled 'User without Address Data'.



Figure 101: Technical User

As of release SAP S/4HANA 1909, in transaction SU01, there is a distinction between business users and technical users. The *Technical User* does not have a business partner associated with it and does not have the address data tab. You should use the *Documentation* tab for the description.

Technical users are typically users who work in the background or work in RFC interfaces between the systems. When creating a technical user, user type *Dialog* is selected by default. However, it makes more sense to choose *System* or *Communication Data* as user type for technical purposes. Further details on user types are described in this lesson.

**Note:**

SAP recommends that you convert all SU01 users (not the technical users) into business users after an SAP S/4HANA conversion. This is possible as of SAP S/4HANA 1709 FP02.

Elements of User Master Records

Elements of User Master Records

- The *Logon Data* tab displays the password and validity period of the user, as well as the user type. For further information about the password rules for special users, refer to SAP Note [622464 – Change: Password change requirement for user type "SYSTEM"](#).

**Hint:**

Some of the following sections in this lesson refer to some fields on this tab in more detail.

- The *Database Management System* (DBMS) tab enables the SAP system to manage users and their authorizations in the DBMS (available for SAP systems running on SAP HANA DB only – see SAP Note [1836006 – Requirements for DBMS User for DB connection](#)).
- The *Secure Network Communications* (SNC) tab is used for managing security functions (external product) that are not directly available but have been prepared in SAP systems. Consider the usage regulations for the country in which you want to use this function.
- The *Defaults* tab displays default values such as the default printer, the default logon language, and so on.
- The *Parameters* tab contains user-specific values for standard fields in SAP systems.
- The *Roles and Profiles* tab shows the roles and profiles that are assigned to the user. Users normally get authorizations via roles. The authorizations are combined in authorization profiles that are automatically generated with a corresponding role. Roles are then entered in the user master record.

**Hint:**

Lesson 2 of this unit refers to authorizations and role maintenance in more detail.

- The *Groups* tab shows the grouping of users for mass maintenance.
- The *Personalization* tab displays personalization objects. In some transactions, personal settings are required that have some effect on the appearance of the transaction. These settings can be stored, or pre-populated using personalization objects.
- The *License Data* tab allows you to specify the contractual user type of the user. It is evaluated during system measurement.

DBMS User Administration

In a typical SAP system installation, you maintain users who execute applications on the AS ABAP in transaction SU01. There, you can also maintain several technical users for the DBMS, but this is only required in certain specific use cases.

- For SAP Business Warehouse (SAP BW), a 1:1 assignment of users is required to grant analytics permissions in the database to virtual analysis users in SAP BW.
- Users run applications that access the database directly. You have to assign database authorizations to these users.

When your SAP system is running on SAP HANA DB, to simplify DMBS user administration, you can define a connection between user administration in AS ABAP and the DBMS. When you create users in AS ABAP, users in the DBMS are created automatically with the same user IDs and the same passwords. When an administrative lock is set for a user in AS ABAP, the corresponding DBMS user is also locked. You can also add or remove DBMS authorizations for the DBMS user, as long as the DBMS supports this.



Hint:

The option to synchronize users created with transaction SU01 in ABAP with the SAP HANA database, must be configured first. This includes adding a database connection in table DBCON (transaction DBCO) for the database user and database type *HDB* and entering the name of the database connection and the client in the *USR_DBMS_SYSTEM* view with transaction SM30 (*Maintain Table View*).

The necessary configuration steps are described in the online documentation (*Product Assistance*) for SAP S/4HANA, area *Enterprise Technology* → *ABAP Platform* → *Securing the ABAP Platform* → *Security and User Administration* → *User Administration and Identity Management in ABAP Systems*. From here, choose the link to *User Maintenance Functions* and from here choose *User and Role Administration of Application Server ABAP* → *Administration of Users and Roles* → *User Administration* → *User Administration Functions* → *Creating and Editing User Master Records* → *DBMS Tab* → *(Link to) DBMS User Management* → *Configuring DBMS User Management for SAP HANA*. When you have made the necessary configuration, the DBMS tab page is also displayed in transaction SU01.

User Types

The *user type* is an important attribute of a user. Different user types are available for different purposes:

Dialog

A normal *dialog* user is used for all logon types by just one person. During an interactive dialog logon, the system checks for expired/initial passwords, and the user can change his or her own password. The usual settings for the validity period of a password apply to this user type. The dialog user can log on to the system multiple times, but be aware that multiple dialog logons are checked and logged.

System

You use the *system* user type for dialog-free communication within a system or for background processing within a system. System users are also used for RFC communication in various applications, such as ALE, Workflow, Transport Management System, and Central User Administration. You cannot use this user type for a dialog logon. The usual settings for the validity period of a password are not valid for system users. Only user administrators sets the productive password and can change the password.



Note:

For more information, also see SAP Note [622464](#) – Change: Password change requirement for user type "SYSTEM".

Communications Data

You use the *communications data* user type for dialog-free communication between systems. You cannot use this user type for a dialog logon. The usual settings for the validity period of a password apply to users of this type.

Service

A user with type *service* is a dialog user that is available to a larger, anonymous group of users. In general, you should only assign highly restricted authorizations to users of this type. Service users are used, for example, for anonymous system accesses using an ICF service. The system does not check for expired/initial passwords during logon. Only the user administrator can change the password. Multiple logons are permitted.

Reference

A *reference* user is a general user, not specific to a particular person. You cannot use a reference user to log on. A reference user is used only to assign additional authorizations. You can specify a reference user for a dialog user for additional authorization on the *Roles* tab page.

The figure *User Types* provides an overview of the types and effects of the password rules for users.



User Type	Enabled for Interaction (SAP GUI / ICF)	Not Enabled for Interaction
Dialog		
System		
Communications Data		
Reference		
Service		
Password Change: Yes	Dialog	Communications Data
Password Change: No	Service	System

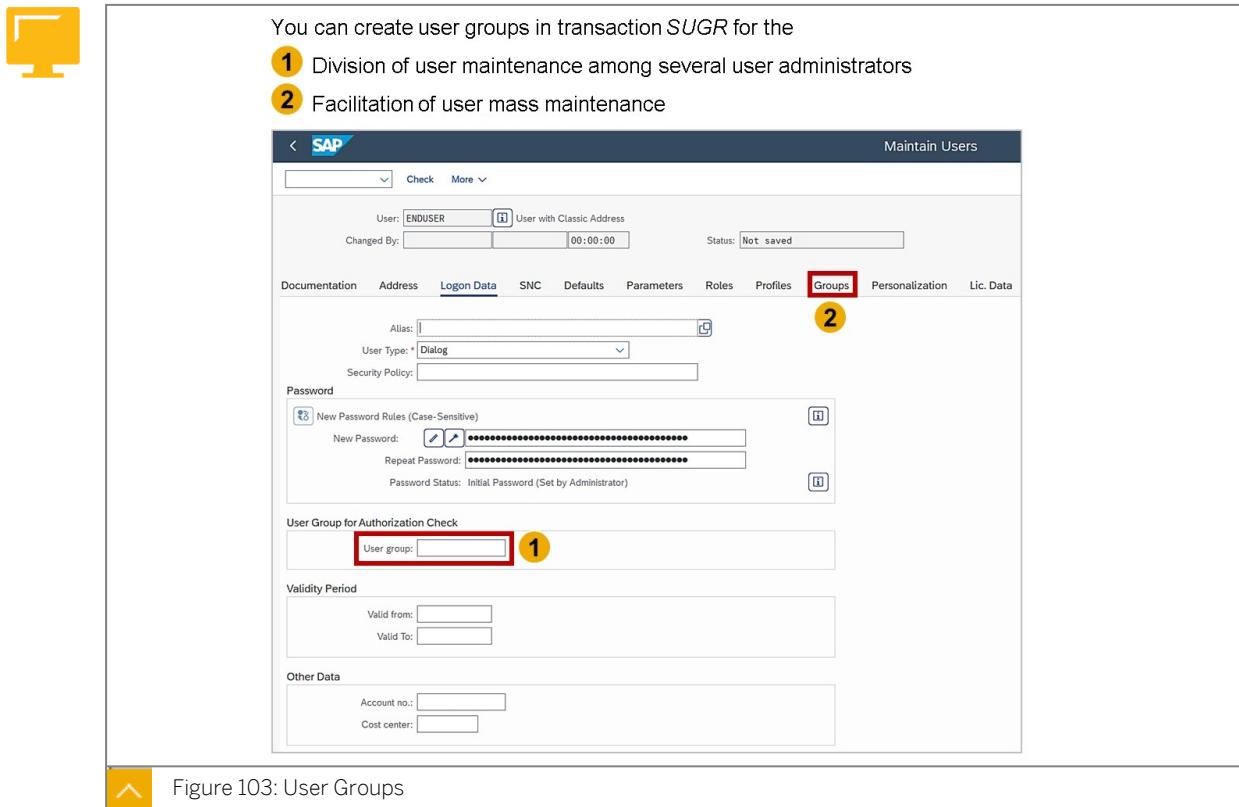
Figure 102: User Types

User Groups

An entry in field *User Group for Authorization Check* (on tab *Logon Data*) is required if you want to divide user maintenance among several user administrators. Only the administrator that has authorization for this group (authorization object *S_USER_GRP*) can maintain users of this group. If you leave the field blank, the user is not assigned to any group (concerning authorization checks). This means that the user can be maintained by any user administrator allowed to maintain any group. This assignment is part of the logon data in the user master record.

For mass maintenance of user data (transaction *SU10*) users can be assigned to a user group on the *Groups* tab page. Assignments that you make on the *Groups* tab page are not used for authorization checks that are specified on the *Logon Data* tab page using the *User Group* field. This grouping is only suitable for mass maintenance.

You can create user groups in the transaction *Maintain User Groups* (*SUGR*).



Additional Information

More information on *User Administration* topics can be found in the online documentation (*Product Assistance*) for SAP S/4HANA, area *Enterprise Technology* → *ABAP Platform* → *Securing the ABAP Platform* → *Security and User Administration* → *User Administration and Identity Management in ABAP Systems*. From here, choose the link to *User Maintenance Functions* and from here choose *User and Role Administration of Application Server ABAP* → *Administration of Users and Roles* → *User Administration* → *User Administration Functions*.

In addition, the following SAP Notes might be helpful:

- SAP Note [2570961](#) – *Simplification item S4TWL - Business User Management*

- SAP Note [622464](#) – Change: Password change requirement for user type "SYSTEM"



LESSON SUMMARY

You should now be able to:

- Outline the different users in the SAP environment
- Explain the elements of user master records
- Describe the different user types
- Explain the importance of user groups
- Create users using the user maintenance transaction

Maintaining User Authorizations with Roles and Profiles

LESSON OVERVIEW

In this lesson, the terms *authorization object*, *authorization profile*, *authorization check*, and *role* are discussed in a common context. The focus here is on role maintenance; that is, on creating a role.

Business Example

Authorizations for users are created using *roles* and *profiles*. Administrators create the roles and the system supports them in creating the associated authorizations.

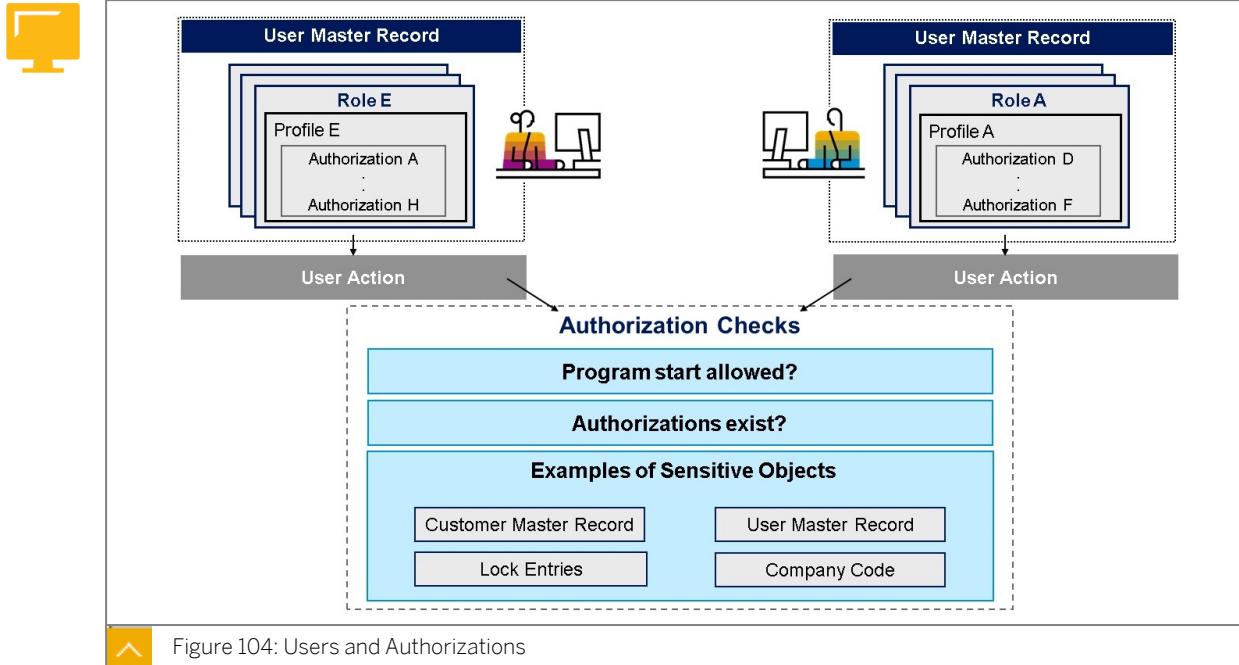


LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe the authorization concept of AS ABAP based SAP systems
- Explain the relation between authorization objects and authorization checks
- Create an authorization role using the role maintenance transaction
- Discuss the SAP Fiori content model
- Explain the authorization concept for SAP Fiori

Authorization Concept



Users can log on to a client of an SAP system if they have a user master record and they know their user name and password, and if the user type is authorized for the logon type. For example, a *communication* or *system* user cannot log on using SAP GUI.

After the users have logged on to the SAP system, they will start working productively by calling applications. As the figure *Users and Authorizations* shows, in the SAP system, there is an authorization check every time an application is called.

If users attempt to start an application they are not authorized to, the system rejects the user by displaying an error message. If the user starts an allowed application, the system displays the initial screen of this application. Depending on the application called, the user enters data and performs actions on this screen. There may be additional authorization checks for the data and actions to be protected.

Authorization Objects and Authorization Checks

To understand the ABAP authorization concept, and to create your own roles and authorizations, you need a basic knowledge of roles and authorization profiles in the user master record.

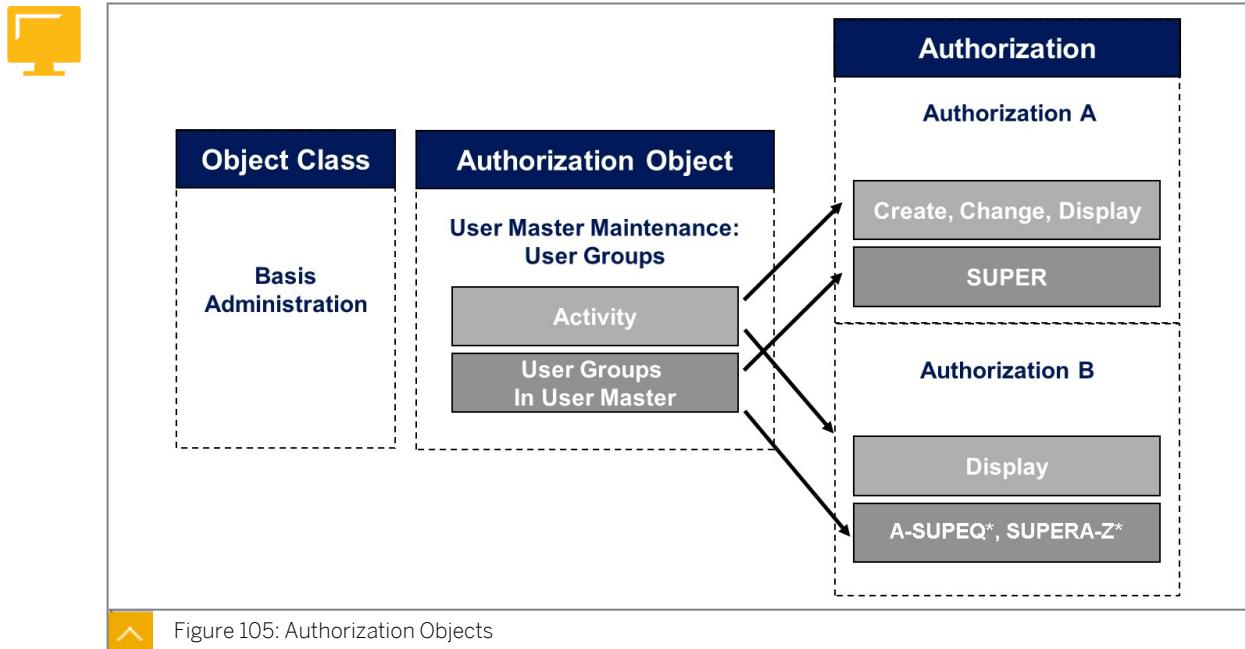


Figure 105: Authorization Objects

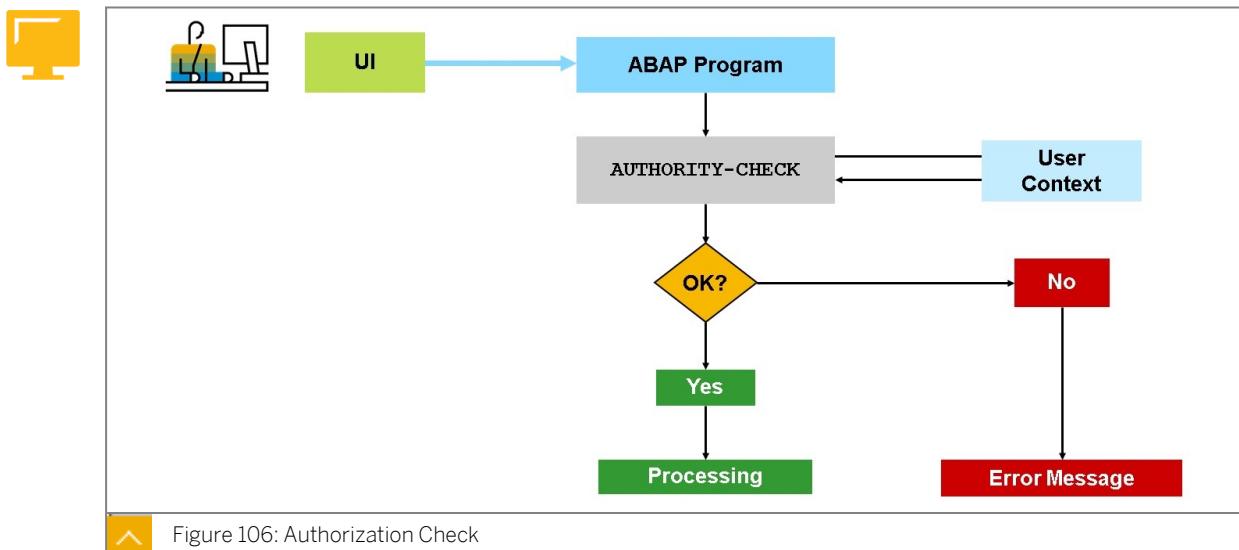
In an ABAP-based SAP system, actions and access to data are protected by authorization objects. The authorization objects are delivered by SAP and are included in SAP systems. To provide a better overview, authorization objects are grouped into **object classes**.

Authorization objects allow complex checks that involve multiple conditions that allow a user to perform an action. The conditions are specified in authorization fields for the authorization objects and are linked for the check. Authorization objects and their fields have descriptive and technical names. An authorization object can include up to ten authorization fields.

The figure *Authorization Roles* shows an example of authorization object *User Master Maintenance: User Groups* (technical name: *S_USER_GRP*) that protects user master records. The authorization object *S_USER_GRP* contains two fields: *Activity* (technical name: *ACTVT*) and *User Group in User Master Record* (technical name: *CLASS*).

An **authorization** is always associated with exactly one authorization object and contains the value for the fields of an authorization object. An authorization is a permission to perform a certain action in the SAP system. The action is defined on the basis of the values for the individual fields of an authorization object. In the example, *Authorization A* for the authorization object *S_USER_GRP* allows creating, changing, and displaying of all user master records that are assigned to the user group *SUPER*. *Authorization B*, in contrast, allows only the display of user master records for user groups that start with *A-SUPEQ* and *SUPERA-Z*.

Multiple authorizations can be generated out of one authorization object. Some authorizations are already delivered by SAP, but the majority are created specifically for the customer's requirements.



As illustrated by the figure *Authorization Check*, when a user logs on to a client of an SAP system, their authorizations are loaded in the user context. The user context is in the user buffer (in the main memory) of the application server. It can be displayed using transaction SU56.

When a user calls a transaction, the system checks whether the user has the required authorization in the individual user context for calling the selected transaction. Authorization checks use the authorizations in the user context. If you assign new authorizations to the user, the user may need to log on to the SAP system again to be able to use these new authorizations (for more information, see SAP Note [452904](#) – *Loss of authorization after profile generation* and the documentation for the parameter *auth/new_buffering*).

If the authorization check for calling a transaction is successful, the system displays the initial screen of the transaction. Depending on the transaction, the user can create data or select actions. When the user completes the dialog step, the data is sent to the dispatcher, which passes it to a dialog work process for processing. Authority checks (command *AUTHORITY-CHECK*) that are checked during runtime in the work process are built into the coding by the ABAP developers for data and actions that are to be protected. If users have all required authorizations in their individual user context for the checks (return code = 0), the data and actions are processed and the next screen is displayed. If the required authorization is missing, the data and actions are not processed and the user receives a message that the assigned authorizations are insufficient. This is controlled by the evaluation of the return code. In this case, it is not equal to 0.

All authorizations are permissions. There are no authorizations that prohibit actions. Everything that is not explicitly allowed is forbidden. You could therefore describe this as a **positive authorization concept**.

Role Maintenance with Menus and Authorizations

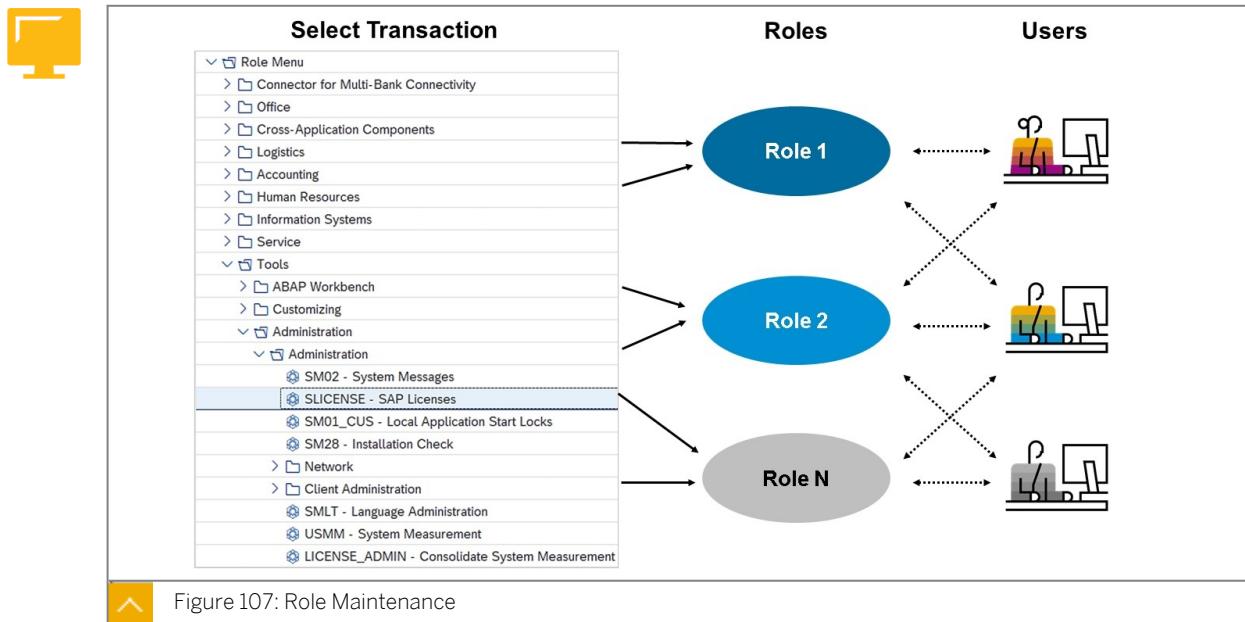


Figure 107: Role Maintenance

Role maintenance (transaction PFCG, formerly also called *Profile Generator*) simplifies the creation of authorizations and their assignment to users. As the figure *Role Maintenance* shows, in PFCG, transactions that belong together from the company's point of view are selected. Role maintenance automatically creates authorizations with the required field values for the authorization objects that are checked in the selected transactions.

A role can be assigned to a number of different users. Changes to a role therefore have an effect on multiple users. Users can be assigned various roles.

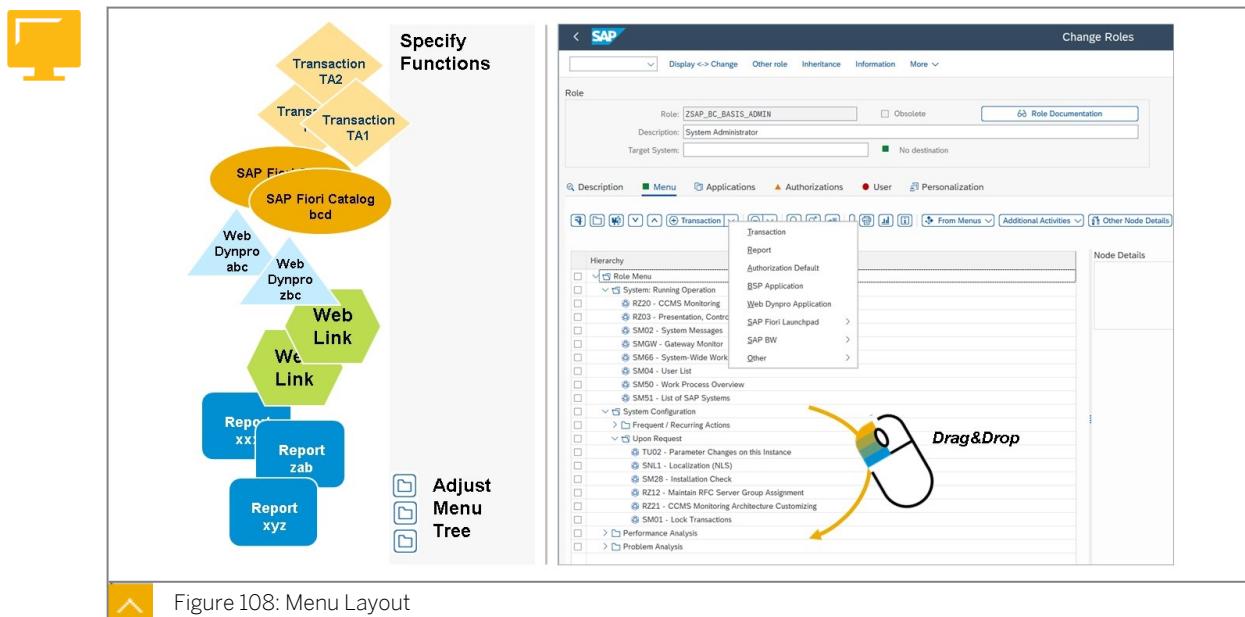


Figure 108: Menu Layout

As the figure *Menu Layout* shows, the user menu comprises the role menu(s) and contains the entries (startable applications, URLs, reports, and so on) that are assigned to the user through the roles.

To edit the menu of a role in PFCG, enter the name of the role and choose the icon for Create or Change. Choose the *Menu* tab page.

Here you can select and change functions: The menu tree can be adjusted for the individual roles as required.

- You can insert startable applications such as **transactions**, **Web Dynpro Applications** or **SAP Fiori objects** (for example, catalogs or groups) in the tree structure and delete existing entries.
- If you choose **Report** in the dropdown menu for inserting, you can also integrate reports. In this case, *Role Maintenance* creates transaction codes (if they do not already exist) with which the reports can be called.
- If you choose the function **Web address or file** in the dropdown menu of the *Insert Node* button, you can add Internet addresses or links to files (such as tables or text files). When integrating files, you must use the storage paths instead of URLs. You can also specify SAP Business Warehouse web reports and links to external mail systems and Knowledge Warehouse.
- The authorizations for SAP Fiori applications are also created using the *Role Maintenance*. You can maintain authorizations for **SAP Fiori Catalogs**, **SAP Fiori Groups** and starting with SAP S/4HANA 2020 also for **SAP Fiori Spaces** by choosing the option **SAP Fiori Launchpad** in the dropdown menu of the *Transaction* button.



Note:

The content model and the authorization model are explained further in this lesson. However, new SAP Fiori Launchpad objects 'spaces' and 'pages' are not discussed in this course.

You can also change menus by creating, moving, deleting, or renaming directories and subdirectories as required. You can use the drag and drop function in role maintenance.



The screenshot shows the SAP Role Maintenance interface. A red arrow points from the 'Edit Authorization Data and Generate Profiles' button (labeled 1) to the 'Generate' button (labeled 2) in the top right corner of the main window. A yellow arrow points from the 'Generate' button to a yellow oval labeled 'Authorization Profile' at the bottom right. The main window displays a list of authorizations under the 'SAP Object Class BC_1' node, with various maintenance status values like 'Maintained' and 'Changed'. The top navigation bar includes tabs for 'Change', 'Delete inactive', 'Selection criteria', 'Manually', 'Organizational levels..', 'Trace', 'Information', 'Versions', and 'More ..'.

Role Maintenance automatically creates the authorizations that are associated with the transactions specified in the menu tree. However, all authorization values must be **manually**

checked and adjusted if required in accordance with the actual requirements and authorities. The system administrator is responsible for this task, together with the appropriate user department. When using organizational levels, you do not carry out maintenance directly in the field, but by means of the *Organizational levels...* button.

As shown in the figure *Generating Authorization Profiles*, you choose the *Authorizations* tab page and then *Display Authorization Data* or *Change Authorization Data*, depending on the maintenance mode. Check the scope and contents of the authorizations.

If there are system proposals, a **green traffic light** in the authorization overview indicates that *Role Maintenance* has supplied at least one proposal for each authorization field. A **yellow traffic light** indicates that the authorization must be maintained manually after it has been created. *Role Maintenance* does not provide a default value for the authorization. In the example shown above, which deals with user maintenance with respect to user groups, *Role Maintenance* offers no suggestion about which user groups should be maintained by a user that will be assigned to this role.

Some fields appear in many authorizations. A number of important fields have therefore been combined into organizational levels, such as the company code. When you maintain an entry for the organizational level using the *Organizational levels...* button, you maintain all the fields that appear there at the same time. A **red traffic light** indicates, therefore, an unmaintained organizational level.

Once all authorizations are maintained as required, the authorization profile can be generated by choosing *Generate*.



Note:

The second character of the profile name must not be an underscore (_). See SAP Note [16466 – Customer name range for SAP objects](#).

After creation, the profile name cannot be changed. The authorizations are grouped together in profiles. The profiles must be entered in the user master record (by role maintenance) for the authorizations to take effect for the user. This is called *User Comparison*.

Users and Roles

The assignment of users to roles is performed in the *Role Maintenance* transaction (PFCG) or in the *User Maintenance* transaction (SU01). Select the *User* tab page and the user IDs to be maintained there. When selecting user IDs, the system uses the current date as the start of the validity period of the assignment; it sets 12/31/9999 as the end date. You can change both values.

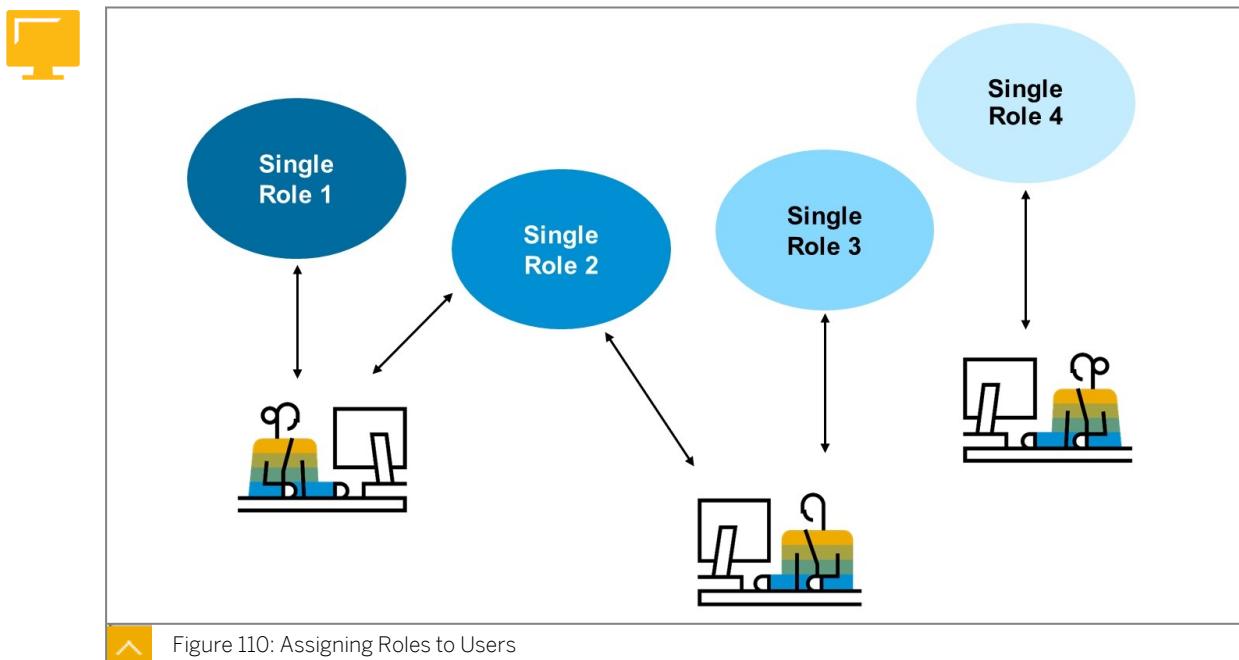


Figure 110: Assigning Roles to Users

As the figure *Assigning Roles to Users* shows, users can be linked to more than one role. This can be useful, for example, if some activities (such as printing) are to be permissible across roles.

The assignment of roles to users does not automatically grant the corresponding authorizations to the users. To assign the authorizations, you must first perform a **user comparison**, during which the role's profile is entered in the user master record. This is illustrated in the figure *User Comparison*.

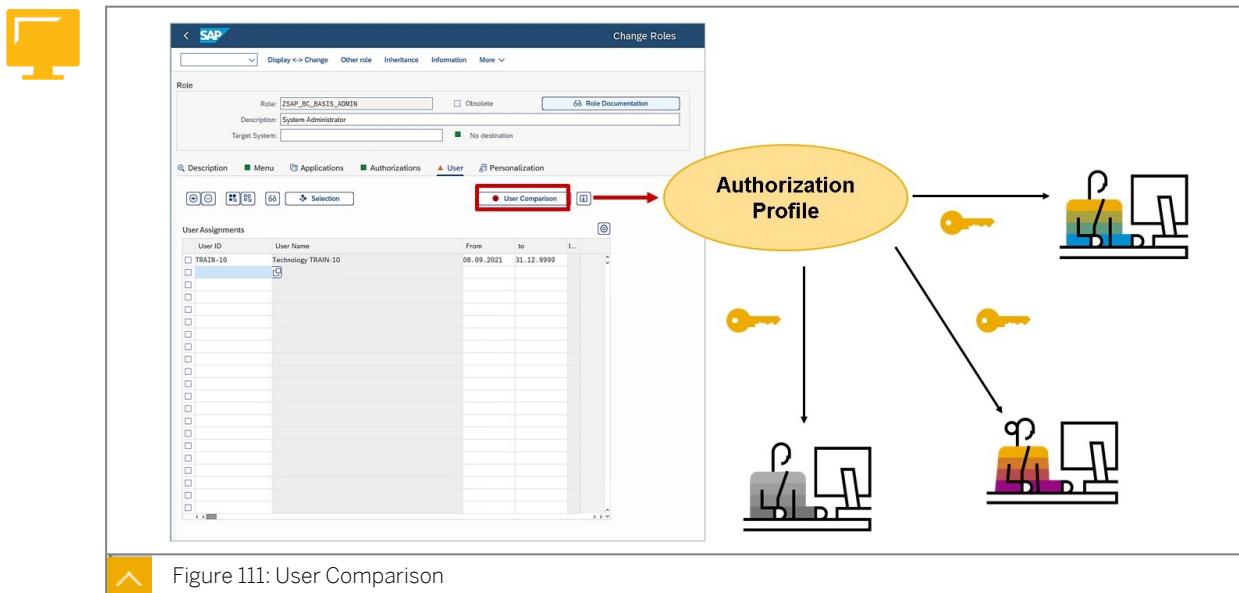


Figure 111: User Comparison

A **user comparison** determines whether authorization profiles should be added to or removed from the current user on the basis of the individual role assignment. During a comparison, profiles are added to a user master record due to roles that have been added. If role assignments are removed manually or time-dependently, the corresponding authorization profiles are deleted from the user master record.

The comparison can be performed individually for each role. For this, select the role in *Role Maintenance* (transaction `PFCG`), choose the *User* tab page, and choose *User comparison*. In the dialog box that the system displays, choose *Complete comparison*.

If you have to update multiple role assignments, you can perform a corresponding comparison in *Role Maintenance* by choosing *Utilities → Mass comparison* (transaction `PFUD`). You can specify the desired roles individually, or update all assignments by entering the asterisk (*) character.

You can also activate the periodic user master record comparison in transaction `PFCG` by choosing *Utilities → Mass comparison*. Choose the option *Schedule or check job for full reconciliation*. The system then displays a search window for the background job `PFCG_TIME_DEPENDENCY`. If it does not find an appropriate job, you can create a new one. The default value is that all user master records are compared once each day.



Hint:

If you want to proactively ensure that the user `SAP*` is no longer required for most emergency situations and that the general profile `SAP_ALL` no longer has to be assigned to the emergency user, see SAP Note [76829 – Emergency role for user administration](#).

SAP Fiori Content Model

SAP Fiori is the design language that brings great user experience to enterprise applications based on SAP User Experience. At the point of SAPPHIRE in 2013, the first 25 apps for managers and employees with request and approval functions had been released. Since then the number of apps has increased greatly. SAP Fiori 2.0 was introduced with SAP S/4HANA 1610, taking the idea of SAP Fiori to the next level. Today SAP Fiori 3 is the current target design, which evolves the SAP Fiori design language for all SAP products to fully support the Intelligent Suite.

A role-based user experience provides end users with all information and functions they need for their daily work - but no more. The classic SAP user interface (UI) often offers a single complex transaction for many user roles. The tasks of a single user then make it necessary to use multiple transactions or additional applications. SAP Fiori decomposes these big transactions into several discrete apps suited to the user roles. All apps are connected to each other so all the tasks of the transaction are still possible. However, they are only performed if the user really wants. The SAP Fiori launchpad then serves as the central entry point for all of the apps of a user.

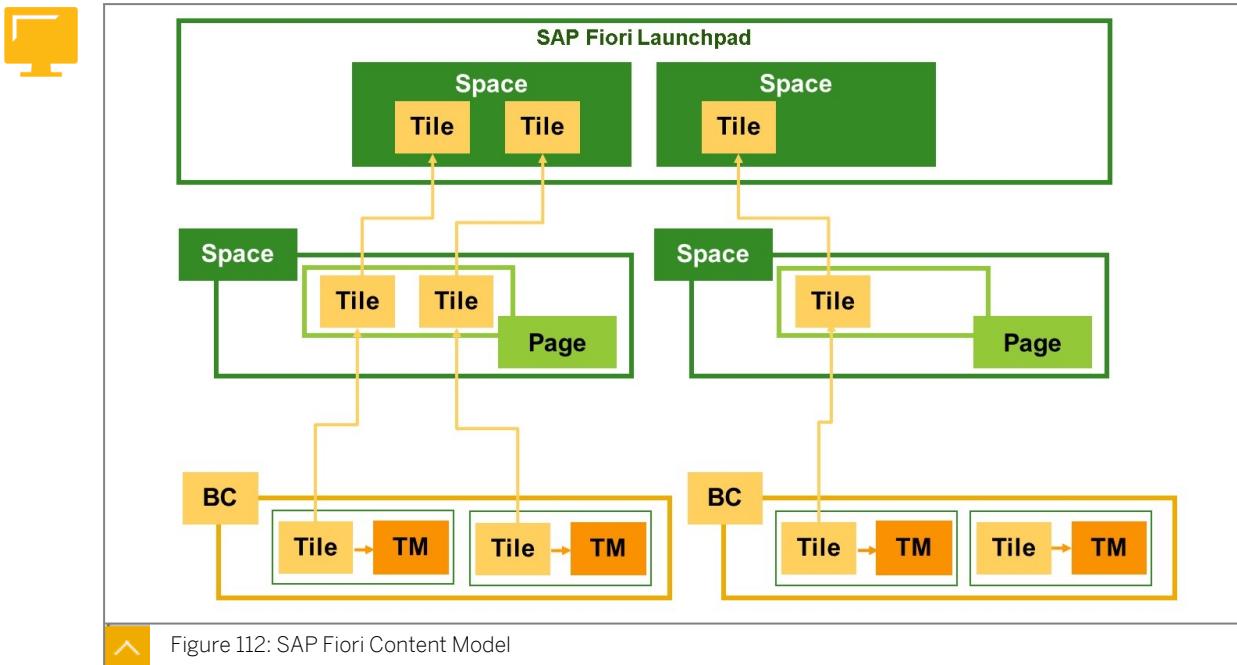
The SAP Fiori Launchpad (FLP) offers several features to make it easier for end users to do their work. Depending on the role of the user, certain groups consisting of apps are displayed. Each app is represented by a tile and a target mapping.

Tiles represent the visual part of an app such as a title, subtitle, information, icon and the semantic object and action for the intent-based navigation. Target mappings define the target application which is launched when an intent (semantic object/action) is triggered.

Tiles can be clicked or touched to start the app. Tiles can offer summarized information about the data available for the user in the app. This information can be visualized using a simple number, color coding based on Key Performance Indicators (KPIs), and even charts showing comparisons, trends, or contributions.

Tiles and their corresponding target mappings are organized in catalogs in the system. These catalogs hold all the technical information to start an application. To show a tile in the SAP

Fiori Launchpad, it must be embedded in a space. Additionally spaces can be structured in pages.



A tile connects the FLP with the app implementation in the system. This includes starting the correct app with the correct parameters and the information shown on the tile. Tiles are created in catalogs that collect all apps of one solution area. Apps, tiles, and catalogs are delivered by SAP. However, they can also be created by customers.

Tiles of catalogs can be assigned to groups and then visualized as tiles or links. Spaces do not add any additional settings to the tiles and have no direct connection to the apps. Both catalogs and spaces can then be assigned to user roles. This works similarly to adding transactions to user roles and grants the user role access to all tiles of the catalog or group.

SAP also delivers template catalogs and roles. However, these should be adapted to the needs of the customer, or, to be more precise, the needs of the users of the customer.

Authorization Concept in SAP Fiori

SAP Fiori apps adopt the user management and authorization concepts provided by ABAP Platform (Application Server ABAP).

The security recommendations and guidelines for user and role administration and authorization as described in the SAP NetWeaver ABAP Platform Security Guide also apply for the SAP Fiori apps.

To use SAP Fiori apps, users need app-specific SAP Fiori user interface (UI) entities and authorizations. You use PFCG roles to assign these types of entities to users.

SAP Fiori launchpad is the access point to apps on mobile or desktop devices. To use SAP Fiori apps, users need the following app-specific types of entities:

- UI:

The SAP Fiori UI entities that define which SAP Fiori apps are displayed to the user. The apps are organized through catalogs and spaces.

- Authorizations:

The authorizations that are required to use SAP Fiori launchpad, to start SAP Fiori apps, and to use the business logic and data of the apps.

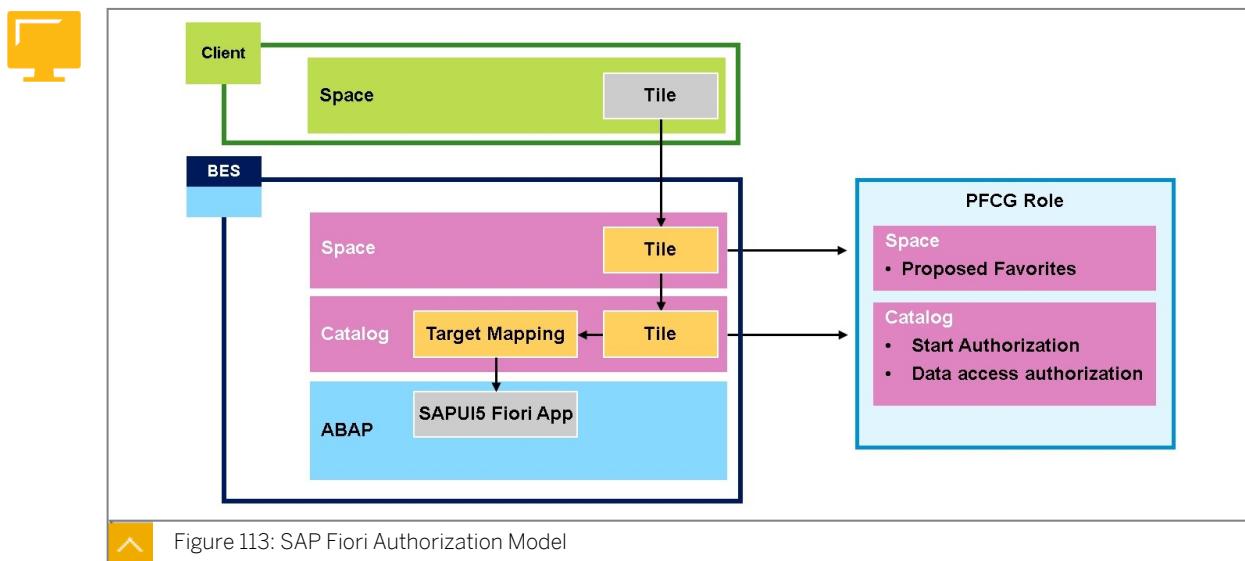


Figure 113: SAP Fiori Authorization Model

You use PFCG roles to assign the UI entities and authorizations to the users:

To get the authorizations for SAP Fiori Apps, you add the SAP Fiori Catalog to the PFCG role menu. By adding the catalogs to the role menu, you include the apps in the catalog that are available to the users. The system determines the services for a catalog and automatically includes the authorizations.

The services that the SAP Fiori apps use are implemented in the SAP S/4HANA system. The users need to have start authorizations for the services and all the business authorizations for accessing the business data that is displayed in the app.

By adding spaces to the role menu, you define the SAP Fiori launchpad home page



LESSON SUMMARY

You should now be able to:

- Describe the authorization concept of AS ABAP based SAP systems
- Explain the relation between authorization objects and authorization checks
- Create an authorization role using the role maintenance transaction
- Discuss the SAP Fiori content model
- Explain the authorization concept for SAP Fiori

Define Profile Parameters to Secure Passwords and Logins

LESSON OVERVIEW

In this lesson, you learn about system parameters that are important in user administration, for example, for logon behavior. You can use the information system to obtain information about any failed logon attempts. Failed authorization checks are analyzed with the system trace.

Business Example

Users are having problems due to missing authorizations. An administrator can use the system tools to analyze these problems.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Identify profile parameters to secure user passwords and user logins
- Implement security policies
- State how to restrict user logons during maintenance work
- Name standard users in the SAP system

Login Parameters

This section considers authorizations in the SAP system from an operational perspective. Among other things, the following questions are considered:

- Which system settings can be used to influence logon behavior?
- How can errors and problems be analyzed?



Profile Parameters	Default	Value Range
Minimum Password Length <i>login/min_password_lng</i>	10*	3-40 characters
Password Expiration Time <i>login/password_expiration_time</i>	0	0-1000 days
Expiration Time for Unused Initial Passwords <i>login/password_max_idle_initial</i>	0	0-24000 days
Expiration Time for Unused User PW <i>login/password_max_idle_productive</i>	0	0-24000 days
Minimum Difference in PW Characters <i>login/min_password_diff</i>	1	1-40 characters

* as of SAP S/4HANA 2020,
for releases up to SAP S/4HANA 1909: 6

Figure 114: Profile Parameters for User Passwords

As the figure *Profile Parameters for User Passwords* shows, the minimum length for passwords is defined with the *login/min_password_lng* parameter. The parameters *login/min_password_digits*, *login/min_password_letters*, *login/min_password_lowercase*, *login/min_password_uppercase*, and *login/min_password_specials* specify the minimum number of digits, letters (number of upper and lower case), and special characters that a password must contain, respectively.

The parameter *login/password_expiration_time* specifies the number of days after which users must set a new password. If the parameter is set to 0, the users do not need to change their password.

The following rules apply to all passwords:

- Passwords cannot start with “?” or “!”.
- Passwords cannot be “pass”.



Hint:

The setting that determines that users must create a new password that differs from the previous 5 passwords they have entered is no longer mandatory. You can use the *login/password_history_size* parameter to set the history from between 1 and 100. The proposed standard value remains 5.

You can define additional password restrictions in table *USR40*.

The *login/password_max_idle_initial* parameter indicates the maximum length of time during which an initial password (a password selected by the user administrator) remains valid if it is not used. Once this period has expired, the password can no longer be used for authentication. The user administrator can reactivate the password logon by assigning a new initial password.

The parameter *login/password_max_idle_productive* indicates the maximum length of time a production password (a password chosen by the user) remains valid when it is not used. Once

this period has expired, the password can no longer be used for authentication. The user administrator can reactivate the password logon by assigning a new initial password.

With the parameter *login/min_password_diff*, the administrator can determine the number of different characters a new password must possess in comparison with the old one when users change their passwords. This parameter has no effect when a new user is created or passwords are reset (for the latter, the parameters for initial passwords apply).



Profile Parameters	Default	Value Range
Terminate the Logon Process <i>login/fails_to_session_end</i>	3	1-99
Maximum Number of Failed Logons <i>login/fails_to_user_lock</i>	5	1-99
Deactivate the Automatic Unlock <i>login/failed_user_auto_unlock</i>	0	0-1
Deactivate Multiple Dialog Login <i>login/disable_multi_gui_login</i>	0	0-1
Exception User (Multiple Login) <i>login/multi_login_users</i>	Alphanumeric	

Figure 115: Profile Parameters for User Logons

As the figure *Profile Parameters for User Logons* shows, you can set the number of failed logon attempts after which SAP GUI is terminated with the *login/fails_to_session_end* parameter. If users want to try the logon again, they must restart SAP GUI.

You use the *login/fails_to_user_lock* parameter to configure the number of possible failed logon attempts before the user is locked in the SAP system. The failed logon counter is reset after a successful logon attempt.



Hint:

At midnight (server time), the users that were locked as a result of incorrect logon attempts are no longer automatically unlocked by the system (default value since AS ABAP 7.0). You reactivate this automatic unlocking with the parameter *login/failed_user_auto_unlock = 1*.

The administrator can unlock, lock, or assign a new password to users in user maintenance (transaction SU01).

If the *login/disable_multi_gui_login* is set to **1**, a user cannot log on to a client more than once. This can be desirable for system security reasons. If the parameter is set to **1**, when users attempt to log on to the system in a new logon session, they can either continue with the new session by ending the previous one or terminate the logon attempt. Users to whom this should not apply should be specified in the parameter *login/multi_login_users*, separated by commas and with no spaces.

Security Policies

In addition to login parameters, you can create security policies to control the logon behavior and the system behavior for password rules and changes. Security policies replace the behavior that is defined by profile parameters.

Individual users sometimes need a non-standard security policy regarding logon behavior and passwords. For example, users with extensive authorizations, such as using administrator passwords, should have stronger protection than standard users. These users should be prompted to change their passwords more often and should be subject to more complex password rules. Intensive use of these requirements can lead to an increase in help desk inquiries, however, if standard users have to meet such requirements as well.

A security policy is a collection of security attributes and their values. You create security policies in transaction **SECPOL** where you define the attributes with specific values for a desired logon and system behavior. The advantage of security policies is that you can assign them on a user and client-specific basis.



Note:

Creating security policies is a customizing operation, this is why you need a customizing transport request.



Figure 116: Defining and Assigning Security Policy

As shown in figure *Defining and Assigning Security Policy*, you need to perform the following steps to create and assign a security policy:

1. In transaction **SECPOL**, you create a new security policy by choosing *New Entries*.
2. Specify the attributes for the security policy by choosing the *Attributes* node.
3. Assign the security policy to a user in the user maintenance (transaction **SU01**).

On the *Logon Data* tab, enter the created security policy in the field *Security Policy*.

You can also assign it to multiple users using mass user maintenance (transaction SU10).

As soon as you assign a security policy to a user master record, it defines the desired behavior. If you leave the field blank, the standard behavior defined with profile parameters applies.

Restricting User Logons During Maintenance Work

The SAP system administrator must manage system downtime manually. This means that the administrator is responsible for locking the SAP system, logging off the users, stopping batch processing, interrupting RFC connections and other interface communication, and so on.

When, for example, maintenance work is being performed on the system, only specific administrators should be able to log on to the system.

Options to Restrict User Logons to the Application Server



- Setting the profile parameter *login/server_logon_restriction*
- Defining a maintenance period using transaction SMAINTENANCE

Using one of these options you can restrict logons to the application server. In general users are not allowed to log on to the system, only users with a specific security policy assigned can log on to the system.

The security policy must either contain the attribute *SERVER_LOGON_PRIVILEGE* with value **1** or the attribute *TENANT_RUNLEVEL_PRIVILEGE* with value **1**.

Restricting User Logons by setting the parameter *login/server_logon_restriction*

- . The parameter *login/server_logon_restriction* can be changed dynamically in RZ11 without restarting the SAP system.

Possible values for parameter *login/server_logon_restriction*

The following values are possible:

- **0: No restriction**
All users can log on to the application server.
- **1: Logon to the application server only allowed with special authorization**
Only users with a specific security policy assigned can log on to the system. The security policy must contain the attribute *SERVER_LOGON_PRIVILEGE* with value **1**. Users who try to log on to the system without this special authorization will see an error message:
Server is currently not generally available (restricted logon).
- **2: No logons allowed on the application server**
Users who try to log on to the system see an error message: **Server is currently not generally available (logon not possible)**.
- **3: An external logon to the application server is allowed only with special rights**
Only those users whose assigned security policy contains the attribute *SERVER_LOGON_PRIVILEGE* with the value **1** can log on to the system externally. Users who try to log on to the system externally without special authorization see the following

error message: **Server is currently not generally available (restricted logon).**



Note:

SAP Note [2065596 – Restricting logons to application server](#) describes which logon types are not external logons.

- 4: No external logon to the application server is allowed

Users who try to log on to the system externally without special rights see the following error message: **Server is currently not available (logon not permitted)**.



Hint:

Setting this dynamic profile parameter does not log active users off from the application server. Use transaction RZ10 to save this value permanently. You can use the where-used list in the user information system (transaction SUIM) to find out to which users the security policy with the *SERVER_LOGON_PRIVILEGE* attribute has been assigned. To do so, start transaction SUIM and choose: *Where-Used List → Security Policies → In Users*.

If you have activated the emergency user, SAP*, a logon to the system with the SAP* user is always possible. The emergency user is active if the profile parameter *login/no_automatic_user_sapstar* is set to 0 and a user master record for user SAP* user does not exist in transaction SU01.

Defining a Maintenance Period Using Transaction SMAINTENANCE

Transaction SMAINTENANCE allows you to define a maintenance period. During this maintenance period, normal users will not be able to log on.

Only users with a specific security policy assigned can log on to the system. The security policy must contain the attribute *TENANT_RUNLEVEL_PRIVILEGE* with value 1.

Users who try to log on to the system externally without special rights see the following error message: **Server is generally not available at this time (maintenance only)**

In addition, only admin batch jobs will be executed (all other jobs will be on hold). For more information, see the SAP Online Documentation for SAP S/4HANA (Product Assistance), area *Enterprise Technology → ABAP Platform → Administrating the ABAP Platform → Administration Concepts and Tools → Administration of Application Server ABAP → Maintenance Mode*.

Lock inactive users

You can use the report RSUSR_LOCK_USERS to lock inactive users. The report lets you select and lock the users automatically. Select the criteria for locking the users in the selection screen of the report RSUSR_LOCK_USERS. You can choose to check the results of the selection and display the users first or to lock the users immediately. Note that only a local user lock is set here. You can run the report both online and in the background.

Initial Passwords for Standard Users



Client	000	001 ¹	066 ¹	Client without user SAP ^{*2}
User(s)	SAP*, DDIC	SAP*, DDIC	EARLYWATCH	SAP*
Password	(set during installation)	(set during installation)	support	pass

! Since these users are well known, they must be protected against unauthorized access.

¹ For deleting client 001 and 066 see SAP Note 1749142 – *How to remove unused clients including client 001 and 066*.

² Log on with coded user SAP* only works, if profile parameter *login/no_automatic_user_sapstar* is set to 0.

Figure 117: Standard Users

There are two basic types of standard users: those created by installing the SAP system and those created when you copy clients.

As the figure “Standard Users” shows, during the installation of the SAP system, the clients 000 and (depending on the system / release) 066 are created. Client 001 is not always created during an SAP installation, but it is created in other circumstances, such as an SAP ECC installation. Standard users are predefined in the clients. The standard names and passwords of these users may be known to other people, so you must protect them against unauthorized access.

- **The SAP system standard user, SAP***

SAP* is defined in the system code, so it is the only user in the SAP system for which no user master record is required. By default, SAP* has the password **PASS** and unrestricted access authorizations for the system.

When you install the SAP system, a user master record for SAP* is created automatically in client 000 (and 001, if it exists). During the installation process, the administrator is prompted to enter a password (in the prompt for the *master password*). The installation process will proceed once the password is entered. The master record created here deactivates the special properties of SAP*, so that only the authorizations and password defined in the user master record now apply.



Hint:

Special characteristics of the SAP* user.

In case you proactively want to ensure that the user SAP* is no longer required for most emergency situations and that the general profile *SAP_ALL* no longer has to be assigned to the emergency user, see SAP Note [76829](#) – *Emergency role for user administration*.

**Note:**

Starting with SAP_BASIS 754, the client copy procedures have been changed and enhanced, see SAP Note [2962811](#) – *New Client Copy Tool: General Information for details*. This new tool includes improved security and reduced manual effort, that is, the user SAP* is no longer needed to perform client copies, providing the benefit the SAP system does not have to be restarted. Additionally, for better process automation it is integrated into the ABAP task manager via task lists.

- **The DDIC user**

This user is responsible for maintaining the ABAP Dictionary and the software logistics.

When you install the SAP system, a user master record is automatically created in client 000 (and 001, if it exists) for the user DDIC. During the installation, you will be prompted to enter a password for this user, similar to the master password prompt for the SAP* user. Certain authorizations are predefined in the system code for the DDIC user, meaning that it is, for example, the only user that can log on to the SAP system during the installation of an upgrade.

- **The EarlyWatch user**

Depending on the system / release, the *EarlyWatch* user is delivered in client 066 and is protected by the password **SUPPORT**. The *EarlyWatch* experts at SAP used to work with this user. This user should not be deleted. Change the password. This user should only be used for *EarlyWatch* functions of monitoring and performance.

**Note:**

Client 066 is not used any more, you can delete it. For details, refer to SAP Note [1749142](#) – *How to remove unused clients including client 001 and 066*.

**Caution:**

To protect the system against unauthorized access, we recommend that you assign these users to the user group **SUPER** in client 000 [001]. This user group is only assigned to superusers.

To counter this problem and protect the system against misuse, you can deactivate the special properties of SAP*. To deactivate SAP*, set the system profile parameter *login/no_automatic_user_sapstar* to a value of > 0. If the parameter is active, SAP* no longer has any special properties. If you now delete the SAP* user master records, the login with **PASS** no longer works.

To restore the former properties of SAP*, you need to reset the parameter and to restart the system.

**LESSON SUMMARY**

You should now be able to:

- Identify profile parameters to secure user passwords and user logins

- Implement security policies
- State how to restrict user logons during maintenance work
- Name standard users in the SAP system

Unit 5

Lesson 4

Evaluate Users and Authorizations



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use the User Information System to obtain information about authorizations and users
- Execute the system trace for authorization checks to locate authorization problems

Options for Locating User Information

To obtain an overview of authorizations and users in an SAP system, you can use the user information system.

You start the user information system with transaction **SUIM**. To start it from user maintenance (transaction **SU01**), choose the menu path *Information → Information System*.



Figure 118: User Information System

As the figure *User Information System* shows, the user information system provides you with a collection of different reports that help you to evaluate information on users, roles, profiles, authorizations, and so on.

You can display lists that answer various questions, such as the following:

- Which users have been locked in the system by administrators or failed logon attempts?
- When did a user last log on to the system?
- What changes were made in the authorization profile of a user?
- In which roles is a certain transaction contained?

- Which users have authorizations assigned to them that are classified as critical?

**Hint:**

We recommend that you regularly check the various lists that are important for you. Define a monitoring procedure and corresponding checklists to make sure that you continually review your authorization plan.

We especially recommend that you determine which authorizations you consider critical and regularly review which users have these authorizations in their profiles.

Records of Authorization Checks

Failed authorization checks can be displayed in transaction SU53. The system shows authorization objects for which the authorization check failed along with the checked values.

In many cases, end users are authorized for SU53 so they can find out themselves which authorizations they are missing.

**Hint:**

Users can only display values for the checked object if they have authorizations for the object S_USER_AUT. Otherwise, the message **You are not authorized display authorization values** is displayed.

The system administrator can use transaction SU53 to check which authorizations other users are also missing while executing an action. If system administrators have authorizations for S_USER_AUT too, they can also display the values that the user has for the checked object.

System Trace for Authorization Checks

You can record authorization checks in your own and other sessions using the system trace function *System Trace for Authorization Checks* (transaction STAUTHTRACE). Alternatively, you can also use the authorization check function in the system trace (transaction ST01).

**Caution:**

By default, the system trace only works if the instance (application server) is the same.



Figure 119: System Trace for Authorization Checks

All checked authorization objects including the checked field values are recorded here. The system trace is suited to finding multiple missing authorizations. In this case, you can activate the system trace for the authorization check of a special user who has all required authorizations for the actions to be checked. The actions are then performed with this special user. The trace records all authorization checks. You can then evaluate them.



Note:

If you want to use values from traces easily when maintaining authorization defaults or roles, see SAP Note [1631929](#) – *Using trace evaluation to maintain menus and authorizations*.

Related Information

More information on *User Administration* topics can be found in the [SAP S/4HANA online documentation](#), path *Product Assistance* → *English* → *Enterprise Technology* → *ABAP Platform* → *Securing the ABAP Platform* → *Security and User Administration* → *User Administration and Identity Management in ABAP Systems*.

In addition, the following SAP Notes cover topics presented in this lesson:

SAP Note [2467](#): *Password rules and preventing incorrect logons*

SAP Note [862989](#): *New password rules as of SAP NetWeaver 2004s (NW ABAP 7.0)*

SAP Note [1631929](#): *Using trace evaluation to maintain menus and authorizations*



LESSON SUMMARY

You should now be able to:

- Use the User Information System to obtain information about authorizations and users
- Execute the system trace for authorization checks to locate authorization problems

Learning Assessment

1. How are authorizations assigned to a user?

Choose the correct answers.

- A Users are assigned authorizations using profiles.
- B Users are assigned authorizations using roles.
- C Users are assigned authorizations using user names.
- D Users are assigned authorizations by a certification authority (CA).

2. Users of which types are able to log on to an AS ABAP using SAP GUI?

Choose the correct answers.

- A Dialog
- B Communication
- C Service
- D System

3. Identify the restrictions that you can set with regard to passwords in AS ABAP.

Choose the correct answers.

- A Minimum password length
- B User is not allowed to change password
- C Password expiration time
- D Password needs to be changed during each logon, except when using single sign-on
- E Number of failed logons until user becomes locked

Learning Assessment - Answers

1. How are authorizations assigned to a user?

Choose the correct answers.

- A Users are assigned authorizations using profiles.
- B Users are assigned authorizations using roles.
- C Users are assigned authorizations using user names.
- D Users are assigned authorizations by a certification authority (CA).

You are correct! Authorizations are grouped together in profiles. The roles assigned to users contain profiles with appropriate authorizations for the role. Authorizations are not assigned using user names or a CA.

2. Users of which types are able to log on to an AS ABAP using SAP GUI?

Choose the correct answers.

- A Dialog
- B Communication
- C Service
- D System

You are correct! Users of type *Dialog* and *Service* are able to use SAP GUI to log on to an AS ABAP. Users of type *Communication* and *System* are not allowed to log on to an AS ABAP using SAP GUI. Read more on this in the lesson “Creating, Copying, and Maintaining User Master Records”.

3. Identify the restrictions that you can set with regard to passwords in AS ABAP.

Choose the correct answers.

- A Minimum password length
- B User is not allowed to change password
- C Password expiration time
- D Password needs to be changed during each logon, except when using single sign-on
- E Number of failed logons until user becomes locked

You are correct! You can use parameters to define the minimum password length, the password expiration time, the number of failed logons until user becomes locked. You cannot disallow the user from changing the password, also you cannot force a password change during each logon. Read more on this in the lesson “Defining Login Parameters and Using the User Information System”.

UNIT 6

RFC Communication

Lesson 1

Describing the Communication Between SAP Systems Using RFC

219

Lesson 2

Managing RFC Resources

225

Lesson 3

Creating RFC Destinations

231

UNIT OBJECTIVES

- Explain the principles of remote function calls (RFC)
- List the different types of RFC
- Identify profile parameters for RFC configuration
- Outline the tools for RFC resource management and monitoring
- Create an RFC destination

Describing the Communication Between SAP Systems Using RFC

LESSON OVERVIEW

In this lesson, you will get an introduction to the Remote Function Calls (RFC) as interface technology available to you in the SAP system.

Business Example

SAP systems can communicate with each other using Remote Function Calls. To do so, the administrator must define the connection.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

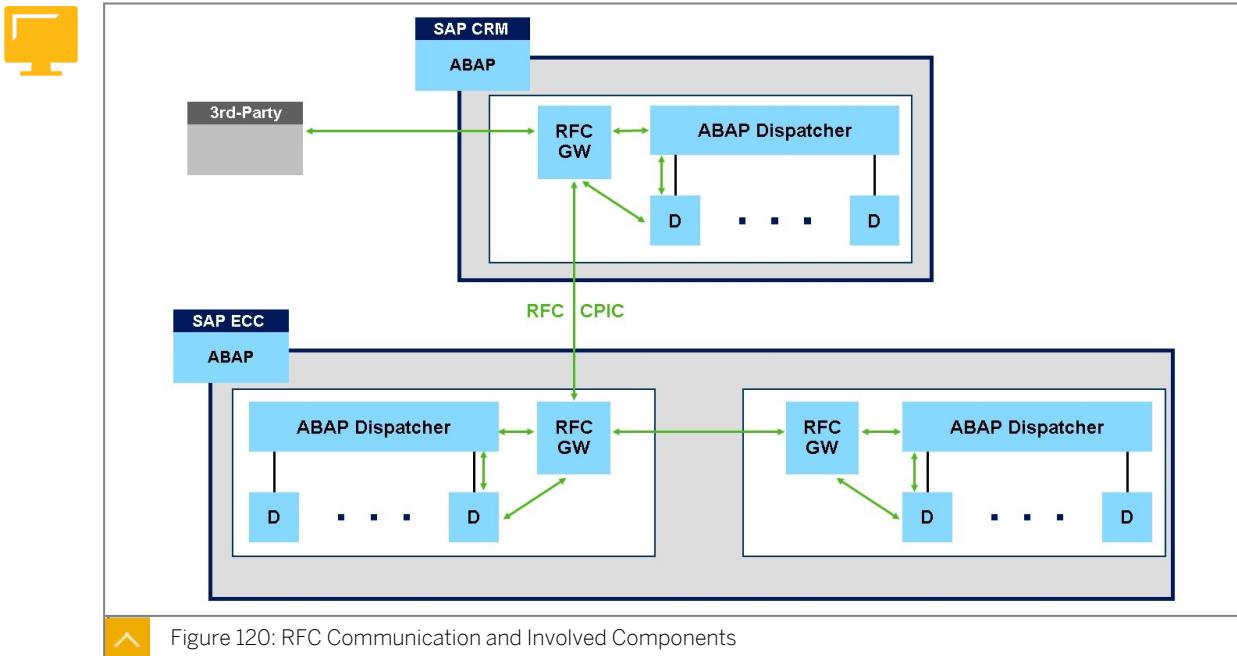
- Explain the principles of remote function calls (RFC)
- List the different types of RFC

RFC Fundamentals

SAP systems have interfaces at different communication levels. These interfaces range from highly technical connection options, for example, using the TCP/IP protocol or CPI-C, to highly specialized interfaces designed for business objects, such as BAPIs or the IDoc interface used in the Application Link Enabling (ALE) environment. All higher interfaces, that is, those that access business objects or processes, use the same technology, the *Remote Function Calls(RFC)*.

The Remote Function Call is an SAP interface protocol, not the programming language, and it is based on CPI-C and TCP/IP. It simplifies the programming of communication processes between different SAP systems. RFCs enable you to call and execute predefined functions, such as a module, in a remote SAP system – or within the same SAP system. Remote Function Calls can also be used for communication with non-SAP software or with programs at the operating system level. RFCs manage the communication process, parameter transfer, and error handling.

The procedure for RFC communication between two SAP systems is that the calling SAP system uses an RFC definition in the SAP system that is called to access a specific function. The Remote Function Calls operate whether the data exchange is synchronous or asynchronous, periodic or demand-driven, or transactional. Many conceivable variants are supported.



As the figure *RFC Communication and Involved Components* illustrates the RFC communication between instances of an SAP system and between different AS ABAP-based SAP systems. During this communication, the RFC gateway is always involved. If a dialog work process has to establish an RFC connection to a remote system in the context of a request (for example, retrieving customer data), it uses the RFC gateway (RFC GW) to communicate with the remote system. The local RFC gateway forwards the request to the gateway of the remote system. The remote RFC gateway transfers the request to the dispatcher, which, in turn, forwards the request to one of its work processes, which then communicates, as a result, directly with its local RFC gateway (without involving the dispatcher further).

Inbound RFC connections are therefore always received by the RFC gateway. Outbound RFC connections are initiated by the work process.

To call an RFC module from an SAP system:

- There must be a technical connection between the two systems – this connection is called an *RFC Destination*.



Hint:

RFC connections are often referred to as RFC destinations. These terms can be used synonymously.

- A developer needs to know the import and export parameters (defined in the *Function Builder*) to perform the Remote Function Call.

RFC is an SAP interface protocol that is based on the Common Programming Interface for Communication (CPI-C) and allows cross-host communication between programs. It also enables external applications to call ABAP functions and SAP systems to contact external applications, provided they are RFC-enabled.

RFC means that ABAP programmers do not have to write their own communication routines. In a Remote Function Call, the RFC interface performs the following tasks:

- Converts all parameter data to the format required in the remote system
- Calls the communication routines that are required to communicate with the remote system
- Handles errors that occur during the communication

The RFC interface is easy for ABAP programmers to use. The processing steps for calling external programs are integrated in the `CALL FUNCTION` statement.

Common RFC Types

RFC is an umbrella term for various technical implementation variants. As an SAP system administrator, you should be familiar with different RFC variants.

RFC Usage Variants

Synchronous RFC (sRFC)

For direct communication between different systems and between Application Server ABAP and SAP GUI.

Asynchronous RFC (aRFC)

For direct communication between different systems and for parallel processing of selected tasks.

Transactional RFC (tRFC)

For true asynchronous communication. Transactional RFC ensures “transaction-like” processing of processing steps that were originally autonomous.

Queue(d) RFC (qRFC)

Queued RFC is an extension of tRFC. It also ensures that individual steps are processed in sequence.

Background RFC (bgRFC)

bgRFC is the successor technology to tRFC and qRFC. We strongly recommend using bgRFC instead of tRFC and qRFC.



Note:

Administrators have to provide an RFC destination, so that developers can execute remote function calls. The defined RFC destination can be used in several different ways. The detailed usage of the existing connection is decided and implemented in the program code to be executed.

Synchronous RFC (sRFC) is the synchronous call of function modules. In this process, the client waits until the server has completed its processing. The two systems involved must be addressable at the time of the call.

**Caution:**

When you use synchronous RFC in ABAP, you can, from the context of the remote (called) function module, use the RFC destination "BACK" to execute remote-enabled modules in the context of the RFC calling program. To do so, the user that executes the RFC calling program must have the necessary RFC authorizations and the callback should not be prohibited by the previous call of the function module RFC_CALLBACK_REJECTED (see SAP Note [1515925](#) - Preventing RFC callbacks during synchronous RFC).

Configurable positive lists can control which function modules can be called via RFC callback for each RFC destination (see SAP Notes [1686632](#) - Positive lists for RFC callback and [1971118](#) - No RFC callback check).

Despite its name, **asynchronous RFC** (aRFC) is not really an asynchronous communication type because it does not fully satisfy the conditions for this type of communication.

Consequently, the called system must be available during the call, as for sRFC. Control over the function returns directly to the calling program in this case, however. aRFC is always recommended if real-time communication is established with a remote system, where processing in the calling program should not be interrupted until the results of the called function module have been obtained (the term asynchronous is used in this sense here). The two systems involved must be available at the time of the call.

Unlike aRFC, **transactional RFC** (tRFC, also formerly known as asynchronous RFC) is a genuine asynchronous communication method that executes the called function module just once on the RFC server. The remote system in which the RFC client program executes a tRFC does not need to be available in this case. The tRFC component saves the called RFC function together with the corresponding data in the SAP database in a unique transaction ID (TID). If a call is sent while the receiving system is unavailable, the call remains in the local queue. The calling dialog program can continue without having to wait for confirmation of whether the function module was executed successfully or not. If the receiving system does not become active within a certain amount of time, the call is scheduled as a background job. tRFC is used whenever a function is to be executed as a logical unit of work (LUW). In this case, all calls within an LUW are executed as follows:

- In the sequence in which they are called
- In the same program context in the target system
- In a single transaction, that is, they are either fully written to the database (known as a COMMIT) or fully reset (known as a ROLLBACK)

tRFC is always recommended if you want to ensure the transactional sequence of the calls.

The disadvantages of tRFC are as follows:

- It handles all LUWs independently of one another. Due to the number of activated tRFC processes, this procedure can significantly reduce performance in both the sending system and target system.
- The sequence of LUWs defined in the application cannot be retained. Consequently, there is no guarantee that the transactions will be executed in the sequence specified by the application program. The only guarantee is that all LUWs are transferred sooner or later.

To guarantee that multiple LUWs are processed in the sequence specified by the application, tRFC can be serialized using (inbound or outbound) queues. This RFC type is called **queued**

RFC (qRFC). Therefore, qRFC represents an enhancement to tRFC. In this process, an LUW (transaction) is not transferred unless it does not have any predecessors (with regard to the sequence defined in the various application programs) in the involved queues. It makes sense to use qRFC whenever you want to ensure that a number of different transactions are processed in a given sequence.

Background **RFC** (bgRFC) is the successor to tRFC and qRFC, with significant improvements in terms of performance and functional capability. We therefore strongly recommend using bgRFC instead of tRFC and qRFC.

For more detailed information RFC usage variants, see the official documentation for ABAP Platform on the SAP Help Portal - RFC Variants: https://help.sap.com/docs/ABAP_PLATFORM_NEW/753088fc00704d0a80e7fdb6803c8adb/4899b53cee2b73e7e1000000a42189b.html.



Caution:

Familiarize yourself with the measures for securing RFC communication recommended by SAP. For more information, see the following:

- SAP Security Recommendations: Securing Remote Function Calls (RFC): Security white paper on <http://support.sap.com/securitywp>.
- Consider using Unified Connectivity (UCON) to block access to function modules that can be accessed remotely but are not required for the business processes of your company: <http://scn.sap.com/docs/DOC-53844>.
- RFC/ICF Security Guide - <https://help.sap.com/viewer/c495ada972d045b2be2869f5573af8e7/201909.000/en-US/4892486caa6b17cee1000000a421937.html>



LESSON SUMMARY

You should now be able to:

- Explain the principles of remote function calls (RFC)
- List the different types of RFC

Managing RFC Resources



LESSON OBJECTIVES

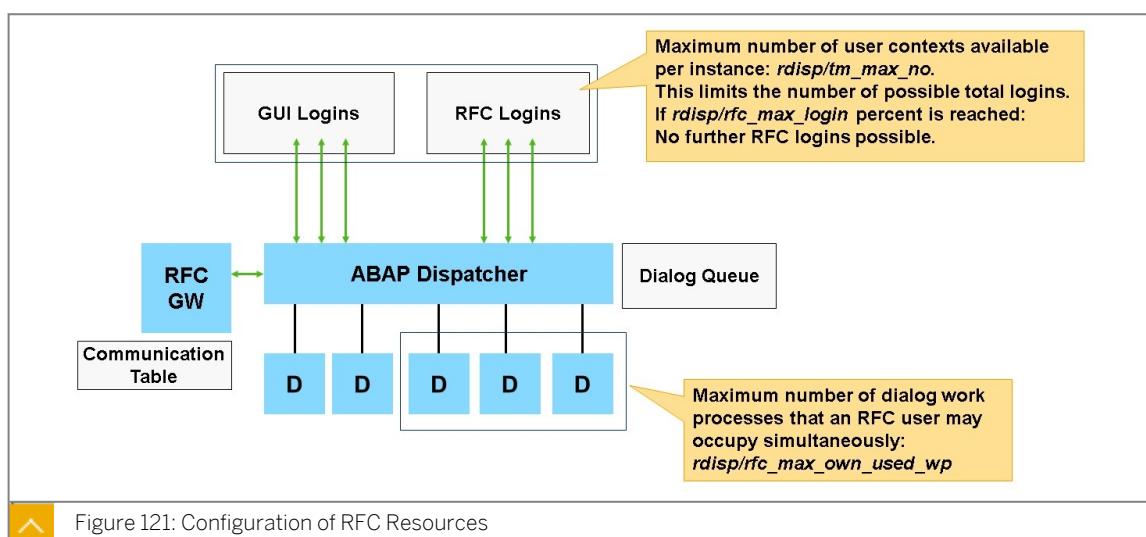
After completing this lesson, you will be able to:

- Identify profile parameters for RFC configuration
- Outline the tools for RFC resource management and monitoring

RFC Resources in the SAP System

Remote Function Calls are processed in dialog work processes. Therefore, every RFC request occupies a dialog work process on the application server on which the RFC is executed. You can optimize the system configuration to ensure an optimum RFC functional capability. Thus, you can also avoid that the RFC communication occupies all dialog work processes that are especially crucial for dialog processing and for end users.

Configuration of RFC Resources



To configure RFC resources, you must maintain various system parameters. The figure *Configuration of RFC Resources* illustrates, which parameters you can use to manage RFC resources, for example, specify the maximum number of work processes that can be occupied by RFC.

RFC resources can be configured for the following areas:

- Logons to the SAP system
- RFC requests in the dialog queue
- Entries in the communication table

- Occupation of dialog work processes

**Caution:**

The information in this lesson on profile parameters and their default values may vary between releases.

Logons to the SAP System

A total of *rdisp/tm_max_no* users can be logged on (via GUI or RFC) to the server concurrently. If *rdisp/rfc_max_login* percent (default value: 90 [%]) of the maximum number of possible users are logged on, no more RFC logons are accepted. A single RFC user can occupy a maximum of *rdisp/rfc_max_own_login* percent of the logon slots (*rdisp/tm_max_no*) (default value: 25).

RFC Requests in the Dialog Queue

The dispatcher dialog queue contains *rdisp/elem_per_queue* entries. Maximum *rdisp/rfc_max_queue* percent of the entries in the dialog queue can be filled with RFC requests.

Entries in the Communication Table

The communication table contains one entry for each CPIC communication on each of the involved sides. It can contain a maximum of *rdisp/max_comm_entries* entries, of which a maximum *rdisp/rfc_max_comm_entries* percent can be used for RFC communication.

Occupation of Dialog Work Processes

When the server starts, it has *rdisp/wp_no_dia* dialog work processes. This number can change while the server is running, such as through an operation mode switch or the addition of dynamic work processes. A single RFC user can occupy a maximum of *rdisp/rfc_max_own_used_wp* percent of the dialog work processes. With this, you can prevent one user (or an application that is sending RFCs) from occupying all the dialog work processes on the target server.

Management of the Dynamic RFC Resource Quotas

Most of the RFC resource parameters represent quotas that can be activated on each application server. These parameters are dynamic parameters. You can use transaction RZ11 to dynamically configure the RFC quotas. You can also use transaction SARFC or the program RSARFCLD to dynamically configure the RFC quotas and to monitor the resources on all servers in the SAP system.

In transaction SARFC you will see the following:

You are shown which servers are currently available in the system, and how the system is handling resources for asynchronous RFCs on the servers.

If you double-click on a server name, a dialog box appears in which you can change the values for your server.

If you have the necessary authorization, you can enter change mode and change the values of the quota parameters described in the previous section.

**Note:**

The parameters are not displayed with their technical names but rather with more user-friendly names. You can double-check what is what in transaction RSPFPAR.

The values that you set here overwrite (until the next restart) the values of the parameters set in the profile file.

A prerequisite for the activation of RFC quotas is that the parameter *rdisp/rfc_use_quotas* is set to the value 1.

**Note:**

Note that the settings you make using the report only apply to the instance to which you are currently logged on, and are lost when the instance is next restarted. The parameter settings in the profile file then apply again.

RFC Server Groups for Internal RFC Load Distribution

Asynchronous RFC can be used to achieve load distribution. When programming an application, the developer can use an RFC destination to execute function modules in parallel background tasks, which run in more than one work process. Long-running SAP reports can implement parallel processing; this reduces the application's runtime considerably.

Parallel processing is implemented in ABAP reports and programs, not in the background processing system itself. That means that jobs are only processed in parallel if the report that runs in a job step is programmed for parallel processing. In the system, parallel-processed background jobs are executed by available dialog work processes.



Logon Group	Instance	Status
390	s4dhost_S4D_11	[checkbox]
390	s4dhost_S4D_12	[checkbox]

RFC Server Groups are only used within an SAP system

Figure 122: RFC Server Groups for Internal Load Balancing

The parallel processing can be executed on any application server. The parallel processing system is built the way that a parallel job cannot occupy all of the resources in an SAP system and cause performance problems for other jobs or other users. However, to achieve even load balancing and optimize the sharing of resources, you can predefine application server groups, called RFC server group. For each RFC server group, you can also maintain separate values for RFC resource parameters.

RFC server groups

RFC server groups are created in a similar way to logon groups. To create them, choose *Extras → RFC Groups* in transaction **SM59**.

You can also use transaction **RZ12** directly, to manage RFC server groups.



Note:

RFC server groups cannot be used for inbound RFC communication; they are only used for RFC calls within the same system, for example, when generating ABAP loads using transaction **SGEN**.

The RFC server groups cannot be used for external RFC communication. The logon groups from transaction **SMLG** are used for this instead.

External RFC Load Balancing



FavTyp	Short Descript.
B	Best Performance
R	Round Robin
W	Weighted Round Robin

Figure 123: External RFC Load Balancing

Logon groups should ensure that the load is distributed as evenly as possible over all available application servers. This is valid not only for dialog user logons but also for external RFC logons. Here, logon groups in transaction **SMLG** can be used for incoming RFC connection from remote systems.

You need to create the logon group in the target system using transaction **SMLG**. In the logon group configuration, you can set the indicator for *External RFC-enabled*.

The logon group can use different method to determine the favorite server:

- *Best Performance*: The server for which the best quality was calculated is used for the next logon.
- *Round Robin*: Each server in the group is used in turn for the next logon.
- *Weighted Round Robin*: The RFC requests are distributed according to the weight defined for an instance. The default weight is the number of dialog work processes configured for the instance

Read more about the weighted round robin method in the SAP Note: [1112104 - Weighted round robin procedure](#)

When this logon group is used, and you select this checkbox, client-side load balancing is performed in the RFC server in the RFC source system. Normally, when consecutive RFC calls are placed, the message server of the RFC system does not determine new logon favorites. This can result in unwanted load balancing in the RFC target system. Therefore, the RFC client system can perform load balancing for the involved instances independently if you select this checkbox.

RFC Activity Monitoring

For monitoring and troubleshooting activities for RFC communication, the following tools are available:

Tools for RFC Monitoring and Troubleshooting

Server resources (SARFC)

Displays the RFC resources currently available to all application servers. Dynamic change of the RFC parameters.

List of transactional RFCs (SM58)

You access this transaction through *Tools → Administration → Monitor → Transactional RFC*. This tool lists those transactional RFCs that could not be carried out successfully or had to be scheduled as background jobs. The list contains the LUW ID and the corresponding error message from the target system.

RFC monitor for the outbound queue (SMQ1)

Here, you can monitor the status of your LUWs in the outbound queue and manually start any queues that hang.

RFC monitor for the inbound queue (SMQ2)

Here, you can monitor the status of your LUWs in the inbound queue.

Gateway Monitor (SMGW)

The Gateway Monitor is used to analyze and manage the gateway in the SAP system.

bgRFC monitor (SBGRFCMON)

You can use the bgRFC Monitor to display the units recorded for the bgRFC. A unit comprises one or more function modules that must be processed as an indivisible unit.

Related Information

SAP Help Portal documentation:

[SAP S/4HANA online documentation](#)

RFC documentation link: https://help.sap.com/docs/ABAP_PLATFORM_NEW/753088fc00704d0a80e7fdb6803c8adb/4888068ad9134076e10000000a42189d.html

SAP Notes

SAP Note [74141](#): Resource management for tRFC and aRFC

SAP Note [593058](#): New RFC load balancing procedure

SAP Note [597583](#): Performance improvement using RFC parallel processing

SAP Note [986373](#): RFC load distribution

Information about RFC security is available in the white paper “Securing Remote Function Calls (RFC)” which is attached to SAP Note [2008727](#) – Securing Remote Function Calls (RFC).



LESSON SUMMARY

You should now be able to:

- Identify profile parameters for RFC configuration
- Outline the tools for RFC resource management and monitoring

Unit 6

Lesson 3

Creating RFC Destinations

LESSON OVERVIEW

In this lesson, you will learn how to set up an RFC destination.

Business Example

As part of an e-commerce scenario, functions from different SAP systems must be linked with each other. Order data, for example, has to be processed further in another system.

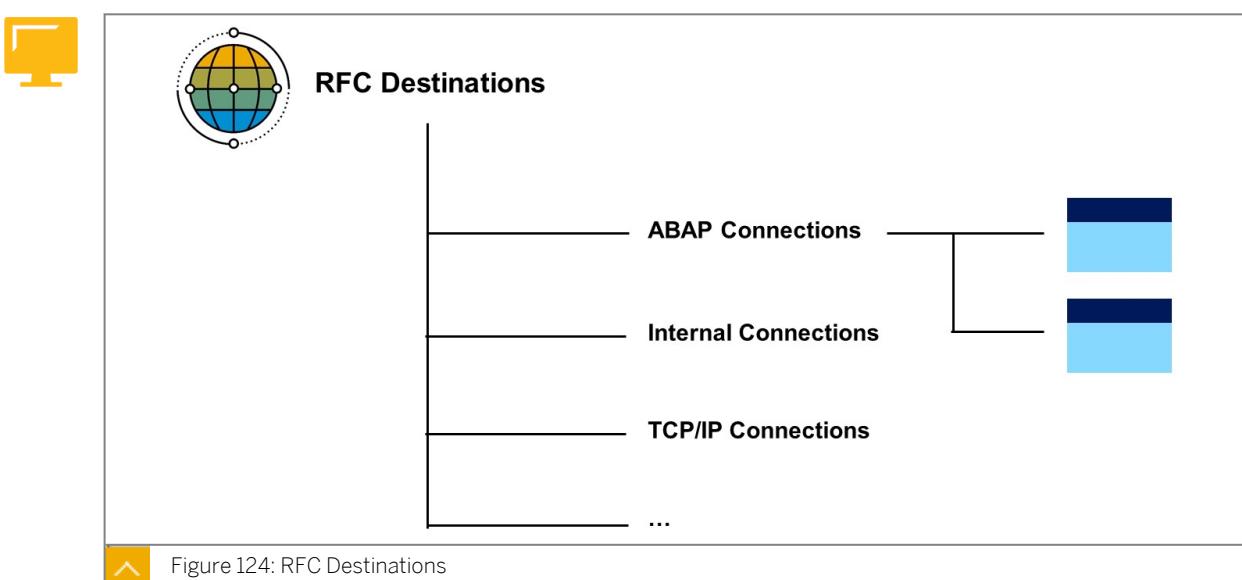


LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Create an RFC destination

Creation of RFC Destinations



You create new RFC connections in transaction SM59.

Before calling a function module on a remote system, you must create a “bridge” over which the calls can be executed. Therefore, you must define the remote system as a destination in your calling system.

You manage RFC destinations always in the source system. Those “bridges” have always a one-way street. A destination points at a specific target system, however, if you want to enable calls from the target system to the source system, the created destination cannot be used. You have to create another destination in the target system that points to your source system.

You create and configure destinations in transaction SM59. The connection types and all existing destinations are displayed in a tree structure on the initial screen.



Hint:

For details about all available connection types, see the online documentation. For this, open SAP Help Portal : https://help.sap.com/docs/ABAP_PLATFORM_NEW/753088fc00704d0a80e7fb6803c8adb/4888068ad9134076e10000000a42189d.html

In SM59, there is a search function for destinations that have already been set up. To search for a destination, choose *Search* and enter your selection. The system displays a list of all matching entries. You can display all available information for each entry.



Note:

The search function is case-sensitive.

To change an existing RFC destination, select the relevant RFC destination in the menu tree and then choose *Change*.



Hint:

To copy an existing RFC connection, you have to switch to the edit mode for the RFC connection you want to copy. Then choose *Connection → Copy*.



Figure 125: Creating an RFC Destination

The figure *Creating an RFC Connection* shows the screen in SM59 where you enter the data to create a new RFC connection. You need to provide at least the following information:

- Connection Type (for example, type 3 for connection between ABAP systems)

- Short Description

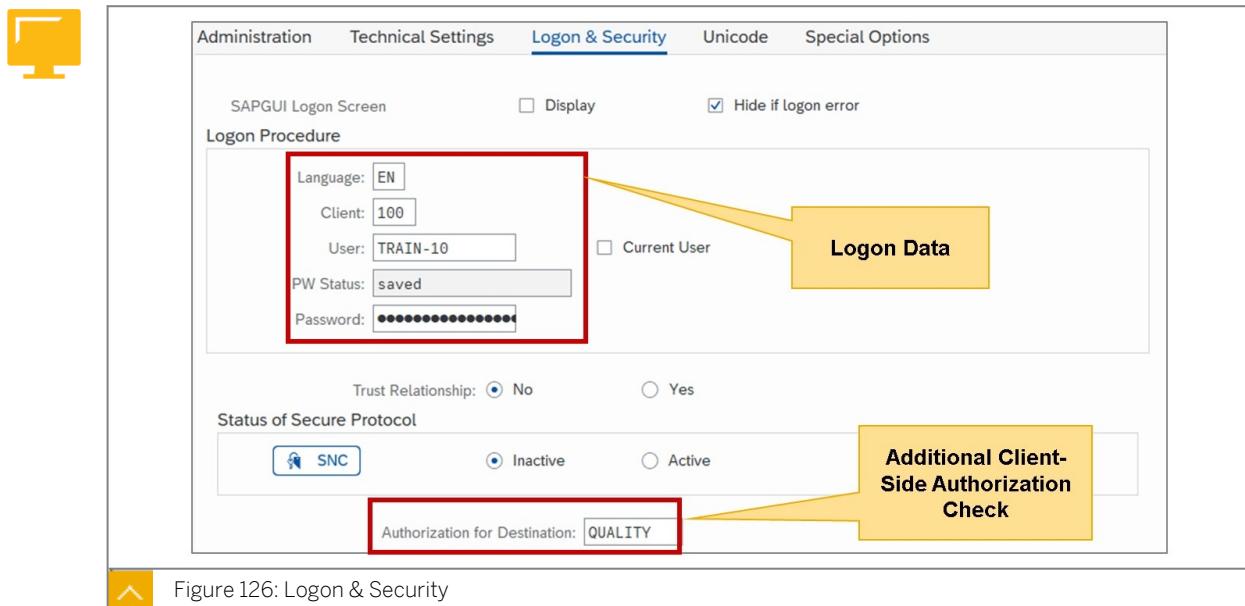
Technical Settings

Other than that, on the *Technical Settings* tab, provide a *Target Host*, which is the name of the system host the connection is pointing at. Also, the *Instance Number* of the remote system is required in case you do not want to use load balancing and the remote function call should be processed on a specific application server instance.

To communicate with the RFC Gateway of the target system, the source system determines and uses the instance number of the corresponding service **sapgw\$\$** and the communication port that is stored in the services file at the operating system level.

Example: If the instance number is 11, the service is **sapgw11** and the port is **3311**.

Logon & Security



On the *Logon & Security* tab, you can define a logon language, a client, and enter logon data that is then used for establishing the RFC connection to the target system.

In the source system, RFC connections can always be used system-wide. This means that an RFC connection you have defined in client 000 can also be used from client 100 (without any difference).

The target of an RFC connection is always a **single client**, and not an entire system.

Hint:

The RFC destination always points at one specific client in the target system. If you don't specify the client in the RFC destination, the system will try to execute the remote function call in the same client as in the source system. However, the same client in the target system may not exist. It is recommended that you specify the client in the destination.

Ensure that you make an entry in the *Client* field for the following reasons in particular:

- If you fail to specify a target client, you might be unable to use your defined RFC connection as expected, despite testing it successfully.
- If the RFC connection fails to work correctly, it can be hard to find the cause of the error, because error messages do not point to a missing entry in the *Client* field.
- RFC connections between ABAP-based SAP systems always target a client within a system, rather than a whole system.
- Strictly speaking, when the connection has been established, the source system always communicates with a single instance of the remote system, even if this instance was found through a logon group.

For security reasons, you should generally enter a user of type **Communication** or **System** with very restrictive authorizations, that is, adjusted to your requirements. In addition, the user needs access authorizations for the remote system, along with authorizations to use RFC functions in the local and remote systems.



Note:

The *PW Status* field informs you whether you have already stored a password in the masked *Password* field or not.

Or you should leave the *User* and *Password* fields blank. If you leave the *User* and *Password* fields empty, the system displays an input prompt for logon when you later open a connection.

If you enter a communication or system user, dialog logons to the system are not possible, although programs can use the connection to communicate. Nonetheless, this requires the careful assignment of authorizations for the non-SAP GUI-enabled users in the target system.



Caution:

Do not use your own user data here, but instead generic user data, because every user (with the appropriate RFC authorizations) can use the RFC destination that you create. Think carefully about the scenarios where this procedure is unavoidable and be careful when granting authorizations to the users in remote systems. Be sure to read the information about securing RFC connections in SAP systems. This should be documented in detail.

You can also use the trusted RFC connection instead. To do so, you must configure a trust relationship between the two systems that should communicate with each other.

For more information, please check the official documentation: https://help.sap.com/docs/ABAP_PLATFORM_NEW/753088fc00704d0a80e7fb6803c8adb/48956eb194cc73e9e1000000a42189b.html?q=trusted%20RFC

Authorization for destination

The *Authorization for Destination* field gives you another option for a client-side authorization check to use in an existing RFC connection. To use this check, the user has to have the authorization object S_ICF with filled out fields. In the field ICF_FIELD, you need to enter the value DEST. In the field ICF_VALUE, you enter the string. You should enter this string then into *Authorization for Destination* in the SM59 to allow only those user with the value to use the RFC destination.

Example

Assume, for example, that you have three RFC connections (such as S4H_FI, S4H_CO, and S4H_SD) that you only want to be used by users who has the S_ICF object in their user buffer and the value for ICF_VALUE field has the value **S4H_PRD**. You need to assign authorizations to the users for the object S_ICF and the fields ICF_FIELD and ICF_VALUE with the values DEST and **S4H_PRD**, respectively. You also enter the value **S4H_PRD** in the *Authorization for Destination* field for each of the three RFC connections in transaction SM59. All users with the appropriate authorizations can now use these three RFC connections successfully.

Unicode

Information concerning the target system code page is stored on the *Unicode* tab page. Perform a *Unicode Test* to check whether the target system is a Unicode system. If the target system is a Unicode system, select the *Unicode* option for the destination. Otherwise, errors may occur when exchanging data or during the use of other means of communication.

For the source system, you can check the Unicode settings by choosing *System → Status*.

Tests for RFC Destinations

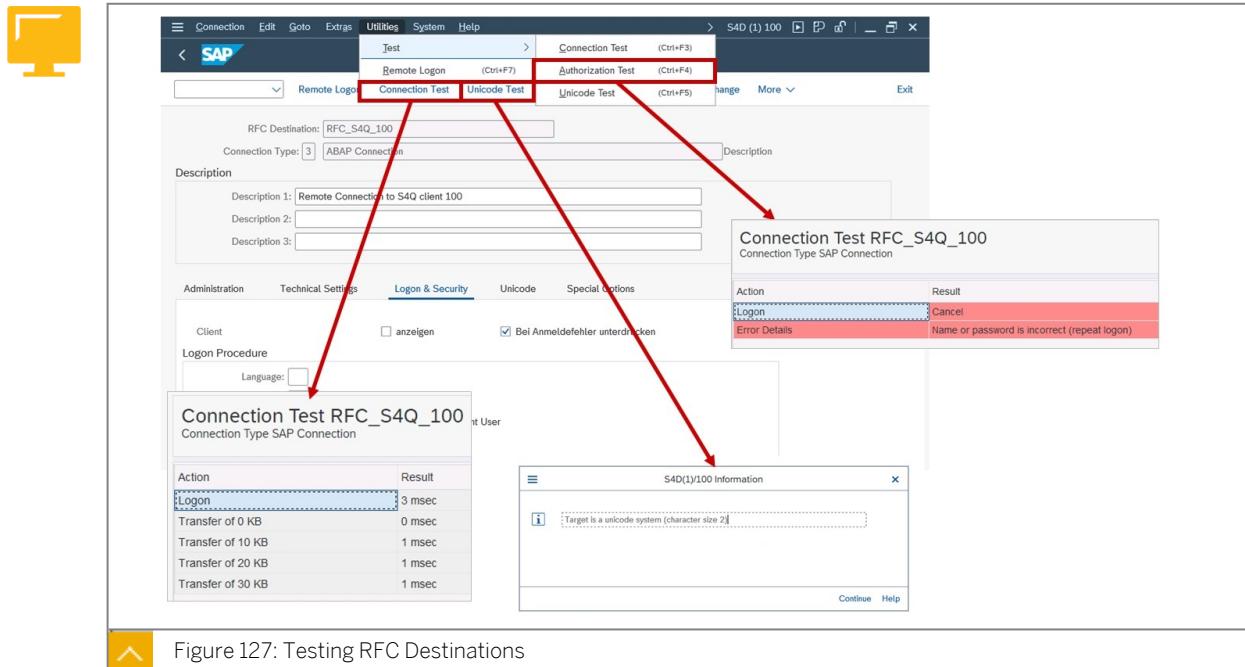


Figure 127: Testing RFC Destinations

The figure *Testing RFC Connections* shows three different functions for testing a destination: *Connection Test*, *Authorization Test*, and *Unicode test*.

Connection Test

During the connection test (*Utilities → Test → Connection Test*), the system attempts to set up a technical connection with the target system and displays a table with the response times. If an error message appears, check your settings. This test is strictly a technical connection test. It only checks whether a partner system can be reached with your information. In particular, it does not query or use any logon data (if such data has been saved in the connection definition).

Unicode Test

When performing a unicode test (*Utilities → Test → Unicode Test*), the system checks if the target system is a Unicode system, and displays the result of the test immediately. Here, as well, no logon data is needed.

Authorization Test

If you have entered logon data for the RFC destination, you can perform an authorization test (*Utilities → Test → Authorization Test*) to test whether the provided logon data is correct.



Hint:

The **Authorization Test's** reaction to incorrect login data is dependent on the checkboxes selected in the *Logon & Security* tab. Choosing *Bei Anmeldefehler unterdrücken* (Hide in Case of Logon Error) will result in an error message during the **Authorization Test**, as illustrated in the figure above. If no boxes are selected or only *anzeigen* (Display) is selected, you will be directed to the remote system's SAPGUI Logon Screen.

Remote Login

You can attempt to log on to the remote system. To do so, choose *Remote Login*. A new session opens for the remote system. Enter the client, your user name, and your password. If you have entered a dialog user with a password in the connection, a dialog logon is performed.



LESSON SUMMARY

You should now be able to:

- Create an RFC destination

Learning Assessment

1. Which RFC variant can you use to process work steps in parallel?

2. RFC communication takes places via dispatcher processes that communicate directly with each other when two instances of AS ABAP exchange data.

Determine whether this statement is true or false.

- True
 False

3. When you configure RFC connections using transaction SM59, which check/test options are available to you in this transaction?

Choose the correct answers.

- A Authorization Test
 B Connection Test
 C Configuration Completeness Test
 D Unicode Test
 E Kernel Compatibility Test

Learning Assessment - Answers

1. Which RFC variant can you use to process work steps in parallel?

You can use asynchronous RFC to process program steps in parallel, as long as there are work processes available in the system.

You are correct! Read more on this in the lesson “Communicating between SAP Systems using Remote Function Calls” of the course ADM100.

2. RFC communication takes places via dispatcher processes that communicate directly with each other when two instances of AS ABAP exchange data.

Determine whether this statement is true or false.

True

False

You are correct! The communication between two instances of AS ABAP involves two (RFC) gateway processes that exchange data between instances. Read more on this in the lesson “Communicating between SAP Systems using Remote Function Calls” of the course ADM100.

3. When you configure RFC connections using transaction SM59, which check/test options are available to you in this transaction?

Choose the correct answers.

A Authorization Test

B Connection Test

C Configuration Completeness Test

D Unicode Test

E Kernel Compatibility Test

You are correct! When configuring RFC connections, you can execute an Authorization Test, a Connection Test and an Unicode Test. Configuration Completeness Test and Kernel Compatibility Test are not offered in this transaction. Read more on this in the lesson “Setting Up RFC Connections” of the course ADM100.

UNIT 7

Maintaining SAP Software

Lesson 1

Describing Maintenance Levels

241

Lesson 2

Implementing SAP Notes

245

Lesson 3

Preparing Software Maintenance using Maintenance Planner

253

Lesson 4

Describing Support Package Manager (SPAM) and Software Update Manager (SUM)

271

Lesson 5

Importing Support Packages with the Support Package Manager (SPAM)

277

Lesson 6

Importing Support Packages with the Software Update Manager (SUM)

285

Lesson 7

Explaining SAP System Upgrade and SAP S/4HANA Conversion

289

UNIT OBJECTIVES

- Describe maintenance levels
- Implement SAP Notes in the development environment and transport them into the production system
- List the features of System Recommendation
- Describe the technical realization of System Recommendation
- Use System Recommendation to find a SAP correction note to resolve a program error
- List the goals of Maintenance Planner

- List the key features of Maintenance Planner
- Describe the Landscape Management Process
- Prepare a maintenance transaction
- Perform a maintenance transaction
- Describe the additional features of Maintenance Planner
- Describe Support Package Manager (SPAM) and Software Update Manager (SUM)
- Import Support Packages with the Support PackageManager (SPAM)
- Describe the idea of importing Support Packages with the Software Update Manager (SUM)
- Explain the idea of an SAP System Upgrade and an SAP S/4HANA Conversion

Describing Maintenance Levels



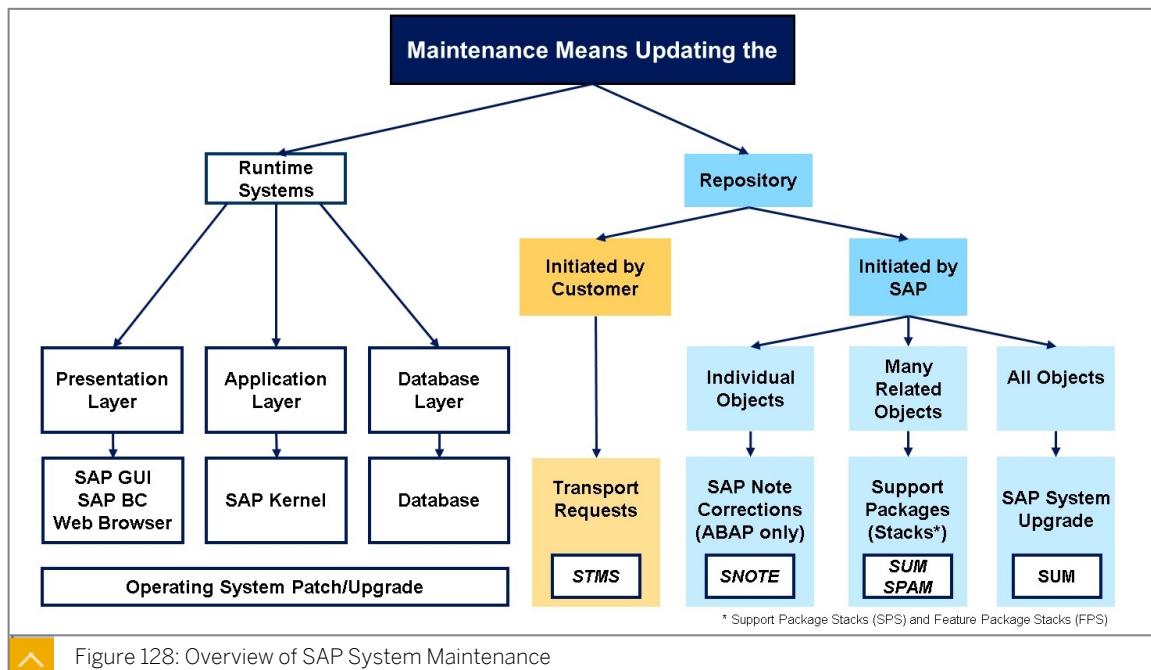
LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe maintenance levels

Types of Maintenance

One important element of SAP Software life cycle management is the maintenance and update of SAP systems. A number of different terms are used in the SAP context, depending on the type of maintenance involved. They are delimited from one another – together with the names of the tools used – in the figure below.



The individual types of changes to the repository initiated by SAP and the tools provided for making them are described in the following lessons. This unit does not cover changes to the runtime environment.

**Note:**

Maintenance requires downtime of the SAP system or parts of the SAP system. This downtime must be coordinated between SAP system administrators and the departments – for example, neither end user jobs should be active in the SAP system during this downtime, nor should data transfers from other systems take place.

SAP Notes and Support Packages

An SAP system consists of various software components. As shown in the figure *SAP Notes and Support Packages*, these software components receive regular updates through SAP Notes and Support Packages. SAP Notes and Support Packages are used to import adjustments based on changes in legal requirements. They are also used to correct errors in the standard SAP software, for security reasons, to enhance existing functions, and make minor new functions available.

The SAP system should always have the most recent Support Package level, to guarantee compliance with legal requirements and to eliminate security problems and errors.



Example: SAP S/4HANA Server

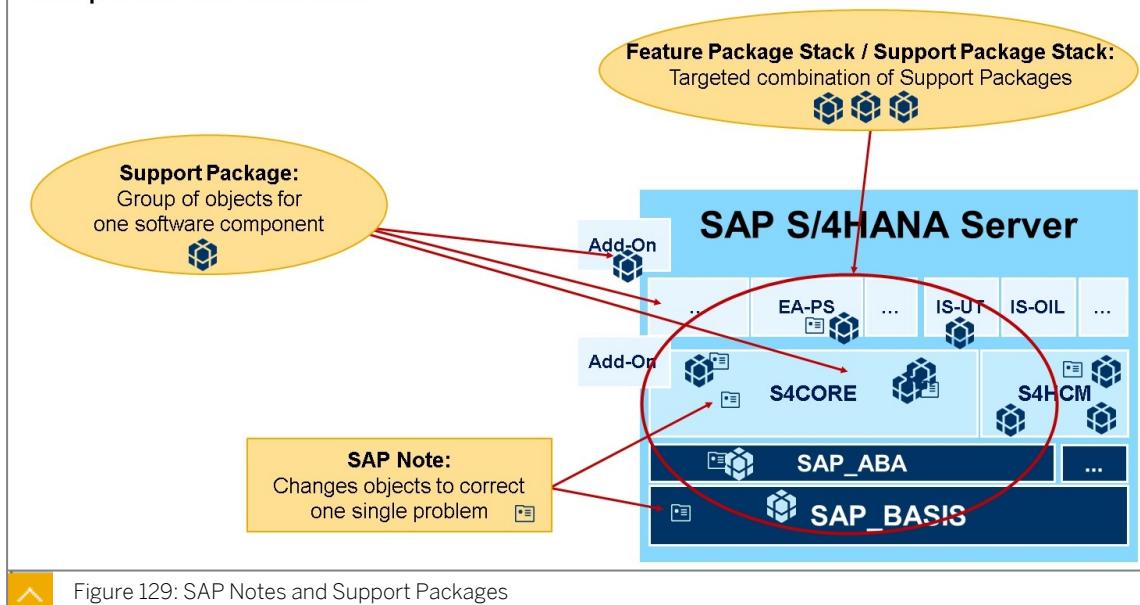


Figure 129: SAP Notes and Support Packages

Some SAP Notes contain general information, hints and tips, or recommendations from SAP. Others describe a problem and the solution to errors in standard functions of SAP software. This second type contains the solution to an individual problem, which is often the solution to a programming error, in the form of corrected lines of source code. Only this second type of SAP Notes can automatically be applied.

Support Packages are bundles of repository objects and customizing. Each release of each software component has its own Support Packages. Technically speaking, Support Packages are a kind of transport request, but a kind that cannot be imported as a normal transport request. A Support Package contains all content from SAP Notes that have been created to the creation date of the Support Package, as well as objects that were not delivered by an SAP Note. A Support Package can also contain new customizing that has been corrected since the last Support Package. Support Packages are not cumulative; they are based on their predecessors.

In principle, importing Support Packages for one particular software component is independent of the level of Support Packages for other software components. Individual components are independent of one another, except when the import of individual Support Packages leads to side effects. To import an S4HCM Support Package, for example, you may have to import an SAP_BASIS Support Package or S4CORE Support Package first. These side effects are documented in a composite SAP Note as soon as they are identified. This SAP Note is referenced during the import of Support Packages.



Hint:

You should use Maintenance Planner to determine the Support Packages to be imported - known dependencies are highlighted there.

To enable you to implement patches consistently in different software components of an SAP system, SAP recommends importing Support Packages using Support Package stacks. Support Package stacks are recommended target combinations of Support Packages for different software components. Support Package stacks are not separate patches, but recommended combinations of normal Support Packages. Support Package stacks are available for the various SAP systems. In addition to Support Packages, stacks mostly contain recommendations for patching other components, such as kernel patches for patching the ABAP runtime environment.

The problem of how to patch a complex SAP system landscape frequently arises, as well as the question of how to actually use the Support Package stack for orderly, documented patching. What is the current patch status of my SAP system landscape? Where can I find the necessary Support Packages and Support Package stack information? The Maintenance Planner provides the solution to these questions. With the Maintenance Planner, you can request the Support Package stacks that are required for the SAP systems of your landscapes defined in SAP Solution Manager in a controlled and manageable way.



Hint:

You should always use the Maintenance Planner for calculating the newest Support Package stack for your SAP system.

For SAP systems based on AS ABAP, a **Support Package** is a bundle of software corrections available as a support package for the ABAP programming language

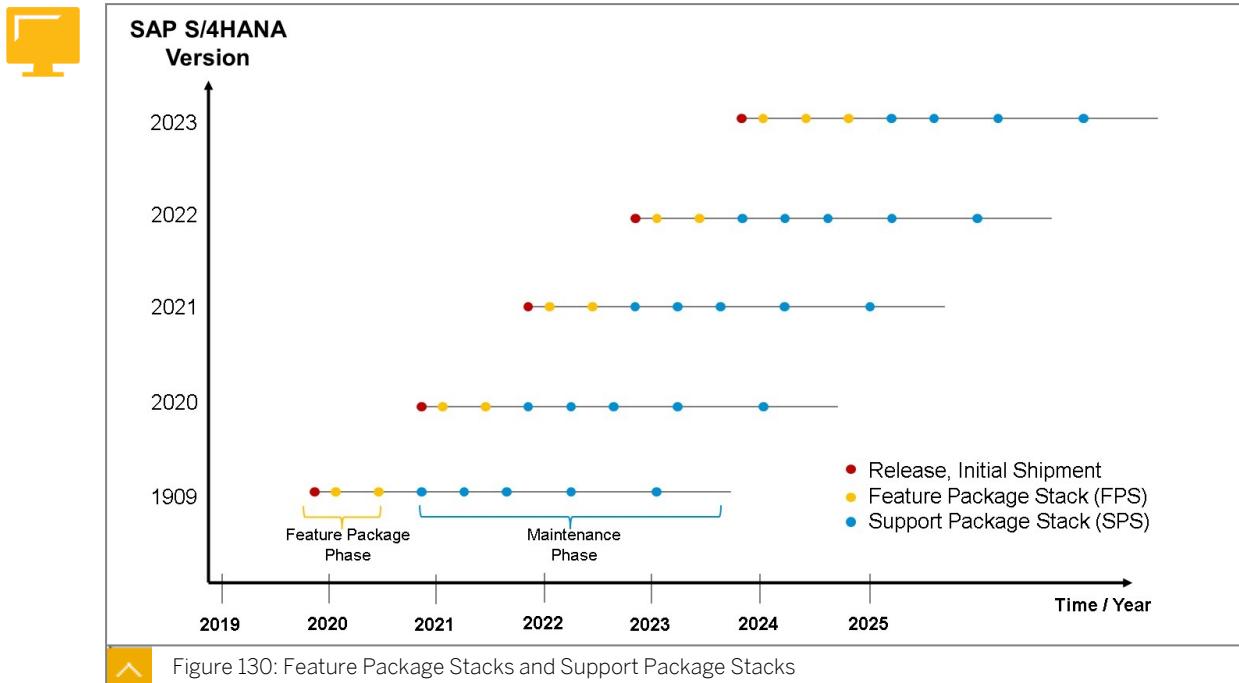
Feature Packages in turn are packages that deliver non-disruptive innovation for generally available product versions, also bundling software corrections and legal changes. A feature package delivery comes as a Feature Package stack. Like a Support Package stack, it must be used in the given combination. Feature Package stacks should be seen as an entity in themselves – customers must heed the minimum requirements and dependencies between individual components and apply the Feature Packages and patches specified in the Feature Package stack together.



Note:

Feature Packages are not provided for all product versions. SAP decides case by case whether a Feature Package phase is offered after general availability of a product version for a defined time period.

For on-premise software products, SAP S/4HANA provides innovations on a regular basis through releases and feature packages.



The last Feature Package stack for a given release before a new release is shipped is usually the starting point of the maintenance phase. This Feature Package stack is also called *Go-to feature package*. During a maintenance phase, SAP provides corrections but no new innovations for the release.

- During the *feature package phase* of a product version, the focus is on *Time to Impact*. In this phase, the functional scope of the product version evolves based on customer and market demands. Therefore moving to the latest product version is recommended for customers that regularly deploy Support Package updates and want to follow the quick innovation track.
- During the *maintenance phase*, the focus is on *maintenance*. Not moving to the latest product version is recommended for customers that do not plan for regular Support Package updates and want to keep their system running as-is.



LESSON SUMMARY

You should now be able to:

- Describe maintenance levels

Implementing SAP Notes

LESSON OVERVIEW

This lesson provides an introduction to the subject of SAP Notes.

Business Example

You want to import changes that were developed by SAP due to new legal requirements into your SAP system. You also want to correct potential errors in the SAP software. SAP provides SAP Notes for this purpose.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Implement SAP Notes in the development environment and transport them into the production system
- List the features of System Recommendation
- Describe the technical realization of System Recommendation
- Use System Recommendation to find a SAP correction note to resolve a program error

Note Assistant

Support Packages are imported using the Support Package Manager (transaction SPAM) or the Software Update Manager (SUM). SAP Notes are implemented with the Note Assistant (transaction SNOTE).

The Note Assistant can implement various types of SAP Notes: changes to SAP programs, the creation of new SAP programs, and changes to SAP function modules, and several other types of repository objects. By implementing SAP Notes, it cannot, however, change other objects, such as dictionary objects. Furthermore, the Note Assistant can only change repository objects, but not other types of objects, such as customizing.

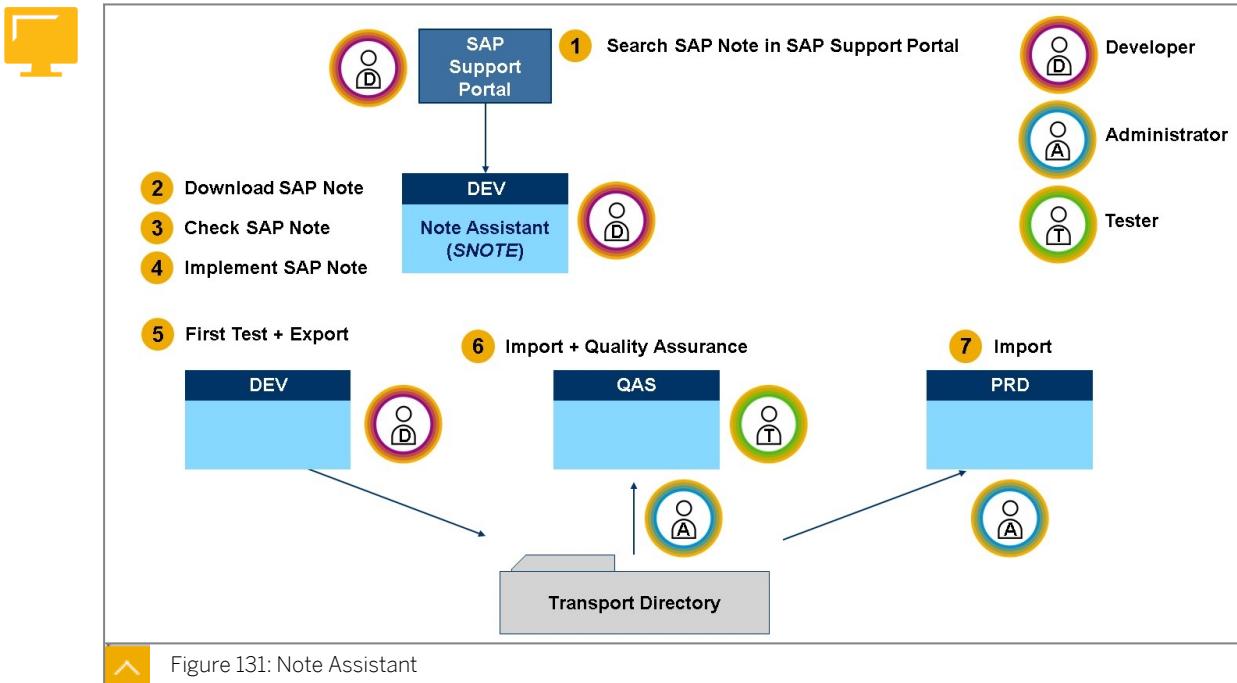
To overcome these restrictions, see section *Transport Based Correction Instructions (TCI)* below.



Note:

SAP Note [3051466](#) – Note Assistant Made Easy : Revamped Note Assistant (available as of SAP_BASIS 750) completely revamps Note Assistant with an aim of making it more simplified, intuitive, customizable and scalable, along with providing a new user experience.

SAP Notes with Correction Instructions



As the figure Note Assistant shows, the Note Assistant implements SAP Notes as follows:

1. You locate the required SAP Notes on the SAP Service Marketplace by searching for keywords, or selecting the unique SAP Note number, if you know it.
2. You download these SAP Notes to the development system with the Note Assistant (transaction SNOTE).
3. The Note Assistant checks the SAP Notes. It checks whether the release named in the SAP Note and the Support Package level are correct, whether the SAP Note requires other SAP Notes as predecessors, whether it can be implemented due to other modifications that may exist for the affected repository object, and so on.
4. You implement the SAP Note by choosing *Implement SAP Note*. This creates a transport request.
5. You run an initial test of the result of implementing the SAP Note in the development system.
6. If this test is successful, you import the resulting transport request into the quality assurance system. The quality assurance test, or technical acceptance, is performed here.
7. If the quality assurance test also is successful, you import the transport request into the production system.



Note:

From January 15, 2020, the download process stopped working unless Note Assistant, transaction SNOTE, is enabled to work with digitally signed SAP Notes. For digitally signed SAP Notes, check SAP Notes [2408073 – Handling of Digitally Signed notes in Note Assistant](#) and [2537133 – FAQ - Digitally Signed SAP Notes](#).

Transport Based Correction Instructions (TCI)

Extensive changes to one functional area typically require a huge amount of SAP Notes and/or multiple manual changes. *SAP Note transport-based correction instruction (TCI)* is a new way to deliver ABAP correction instructions in a flexible manner.

From technical point of view, there are the following type of SAP Notes:

- SAP Notes without any correction instructions (such as *Consulting notes* or *Information for Installation notes*)
- SAP Notes with correction instruction(s)
- SAP Note Transport-Based Correction Instructions (TCIs)

SAP Notes with correction instruction(s) contain specific, **single** corrections. They can be implemented automatically for all supported objects. However, objects that are not supported must be implemented manually, which in turn might require developer knowledge.

SAP Note Transport-Based Correction Instructions (TCIs) in contrast contain corrections of only one specific functional area, which is an encapsulated subcomponent. They can contain **multiple** corrections, which reduces dependencies to other corrections and update artifacts. Technically, they contain an ABAP transport similar to Support Packages deliveries, but they are delivered together with SAP Notes as a new type of correction instructions.



Hint:

SAP Note [2489679 – How to identify TCI Note](#) helps to differentiate a TCI Note from a non-TCI Note.

A TCI Note is a note which contains ABAP corrections in a package (such as *COMM SAPK70000LCPSAPBASIS*). In contrast, a non-TCI Note is a note which contains ABAP corrections to a object / new object (such as *REPS RSBDCOSO*). In order to identify the kind of SAP Note, you need to check the delivered correction type.

TCIs support all ABAP objects that have a transport connection. A single TCI consists of exactly one transport request and a set of installation attributes, for example, software component version, minimal Support Package, languages.

The validity of a TCI is defined by the software component version and a range of Support Package levels.



Note:

As prerequisite, the system landscape has to be enabled by following the information in SAP Note [2187425 – Information about SAP Note Transport based Correction Instructions \(TCI\)](#). This SAP note contains a PDF as attachment that explains the configuration process in detail.

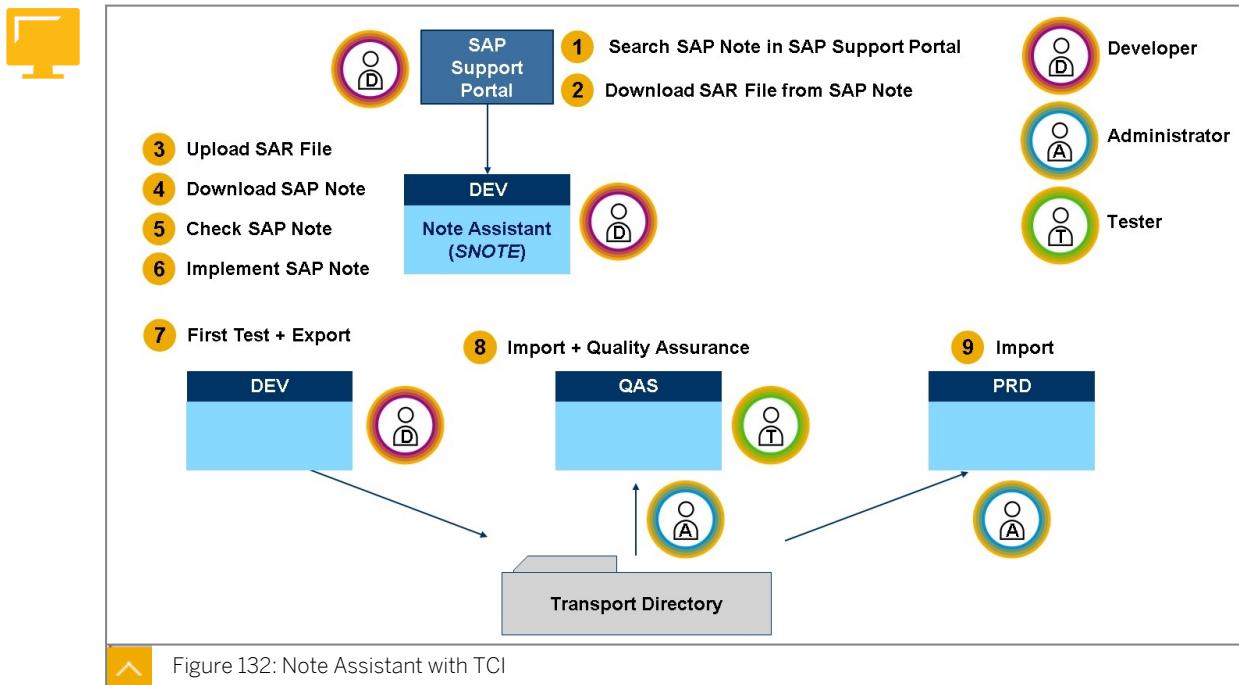
Benefits

Compared to SAP Notes with correction instructions, SAP Note transport-based correction instructions (TCI) support all transport-enabled SAP ABAP objects including Data Dictionary objects, table content and MIME objects. In addition, they allow a fast consumption of consolidated correction instructions and no development skills are needed for implementation.

due to an automated installation process. If needed, TCIs can also be de-implemented or rolled back. For details, see SAP Note [2408383](#) – TCI - Enabling System for SAP Note Transport-Based Correction Instruction (TCI) Rollback.

Implementation Process

The following figure outlines the implementation process for TCIs:



- In SAP ONE Support Launchpad, open the SAP Note that contains the TCI.
- Choose *Correction Instructions* and select the relevant software component.
- Choose *Download*. The system provides a SAR file. You have to save the downloaded SAR file into a directory of your front end computer or on operating system level of your SAP system.
- To implement the SAP Note with the corresponding TCI in your development system, use the *Note Assistant* through transaction SNOTE. In the *Note Assistant*, choose *Goto* → *Upload TCI* and select the SAR file that you have downloaded.
- Then choose *Goto* → *Download SAP Note* and enter the SAP Note number that matches the SAR file. Finally choose *Execute* and proceed as usual.



Note:

The manual download and upload of the SAR file is not necessary if you use the ABAP Download Service to download SAP Notes with all of their dependencies and relevant transport-based correction instructions. See SAP Note [2554853](#) – SAP NetWeaver download service for SAP Notes.

- Record your changes in a transport request. After a first test in the development system, release the transport request.

**Hint:**

Warnings about objects without directory entry can be ignored, since TCIs can contain deletions.

- Import the transport request into the quality assurance system and perform an (integration) test.
- After the integration test has successfully been passed, you can import the transport request into the production system.

For more information on TCIs, see the PDF that is attached to SAP Note [2187425 – Information about SAP Note Transport based Correction Instructions \(TCI\)](#).

Excusus: System Recommendations Features

SAP Solution Manager hosts an app called System Recommendations. This app allows you to manage the SAP Notes and Support Package Patches that are not yet implemented and installed on your technical systems. Once an SAP Note has reached the completely Implemented status or a Support Package Patch has been registered as being fully installed by SAP Solution Managers Landscape Management Database (LMDB), you can no longer see them in the application. The app automatically calculates the SAP Notes and Support Package Patches for the desired technical system.

The following features are available:

- Calculate, display, and assign statuses to SAP Notes that are available for installation on one or more of your technical system
- Display SAP Notes and their object list as well as the prerequisites SAP Notes
- Manage and track the workflow of an SAP Note implementation
- Analyze the effect that implementing an SAP Note has on your systems and business processes
- Create a request for change
- Select a Java patch and add it to the Download Basket
- Enable background services to automatically update SAP Note information

**Note:**

In this Lesson we focus only on the SAP Notes for ABAP based Systems.

The System Recommendation comprises the following SAP Notes categories:

Table 2: Examples of SAP Notes which are supported

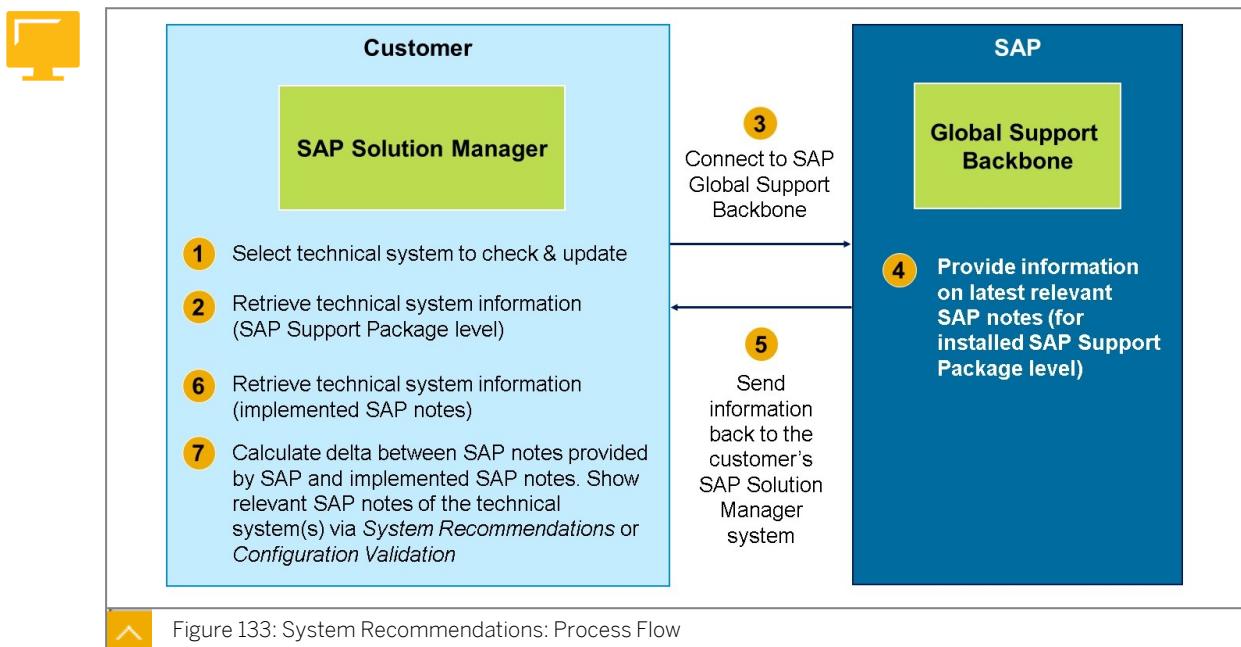
SAP Note type	Explanation
SAP Security Notes	Important SAP Notes in the security category.

HotNews	SAP HotNews, which are SAP Notes with priority 1 (very high). These SAP Notes provide information to help avoid and/or solve problems that can result in data loss or crashes of the SAP system.
Performance-Relevant Notes	SAP Notes from the performance category improve the performance of your system.
Legal Change Notes	SAP Notes from the Legal Change, Announcement of Legal Change, and Correction of Legal Function categories respond to changes in legal requirements.
System Measurement (License Auditing Notes)	SAP Notes related to Licensing SAP Software, like new license checks, usage information, and so on.
SAP Notes for Corrections/Patch Notes	All SAP Notes that contain corrections to program source code (ABAP) or contain patches (Java).

System Recommendation – Technical Realization

To simplify data entry and speed up delta calculation, background jobs are scheduled during the configuration of SAP Solution Manager that automatically collect and store all the required information from the managed systems as well as from the SAP Global Support Backbone.

The next figure simplifies this process:



The SAP Solution Manager sends Support Package level for each managed system to SAP Global Support Backbone, it performs a delta calculation and delivers all relevant SAP Notes back to the SAP Solution Manager. The SAP Solution Manager subtracts off all SAP Notes and Support Package Patches already implemented and you get only the relevant SAP Notes for the technical system.

Use System Recommendation to find SAP Notes for Program Errors

Start the System Recommendation app

To start the System Recommendations start the transaction `SM_WORKCENTER` within the SAP Solution Manager to start the SAP Solution Manager Launchpad. Then navigate to the *Change Management* area and select *System Recommendations* to start the application.



Note:

To find the system in the list the following activities including their required activities need to be performed: *SAP Solution Manager Mandatory Configuration* and the *Managed System Configuration*.



SM_WORKCENTER → Change Management → System Recommendations

The screenshot shows the SAP Solution Manager Launchpad interface. At the top, there is a navigation bar with links for Home, Change Management, SAP Solution Manager Configuration, Root Cause Analysis, Project and Pro, and a search icon. Below the navigation bar is a grid of application tiles. One tile, labeled "System Recommendations" with a count of "3 Systems" and a red border around it, is highlighted. Other tiles include "Change and Release Management", "Administration Cockpit", "Change Control Management Analytics", "Quality Gate Management", "Approve Q-Gates and Urgent Changes", "License Management System Overview", "SAP Readiness Check", "Scope and Effort Analyzer", "Maintenance Planner", and "Download Service Maintenance Planner".

Figure 134: System Recommendations: Starting the Application

Use the System Recommendation app

When you start the application, the System Overview is displayed. On the System Overview you can select the system and choose *Display SAP Notes*, the SAP Note Overview screen displays, as shown in the next figure.



SAP System Recommendations - SAP Note Overview

NewStandard1

Technical System: S4H-ABAP	Release Date: dd.MM.yyyy - dd.MM.yyyy
Note Type:	Priority:
Implementation Status:	Processing Status:
Application Component: FI	Category: A - Program error

SAP Notes for selected technical systems: 2

Technical System	Note Number	Short text	Release Date	Application Component	Priority	Support Package	Category	Security Category	Implementation Status	Processing Status	Correction Types	Attributes
<input type="checkbox"/> S4H-ABAP	2819883	When using update task: Clearing information not deleted after reversal or clearing reset	1/15/2020	FI-CA	1 - HotNews	SAPK-10401INS4 CORE	A - Program error		New	Undefined	Automatic	No Kernel, Dependent
<input checked="" type="checkbox"/> S4H-ABAP	2820009	CFIN maps TAX_COUNTRY in BSET and not in BSEG	2/11/2020	FI-GL-GL-F	1 - HotNews	SAPK-10401INS4 CORE	A - Program error		New	To Be Implemented	Automatic	No Kernel, Dependent

★ Save as Tile Actions Integrated Desktop Actions

Once you find the SAP Note to solve the issue you could use one of the **Integrated Desktop Actions** to perform the change in your Landscape:

- Start Impact Analysis
- Download the SAP Note
- Create Request of Change

 Using **Actions** you find the following features: Show Object List, Show prerequisite Notes, Show Side Effect Notes, Change Status

Figure 135: System Recommendations: SAP Note Overview

Use the filter options to find right SAP Note to fix your issue. In our case the ABAP System S4H is effected. We filter for the Application Component FI (Financial Accounting) and the Category *Program Error* as we get informed by the Business Operation there is an error on CFIN maps.

By choosing *Display Detail Page*, the *Show SAP Note Details* screen displays.

With the *Integrated Desktop Actions* you could for Example *Download the SAP Note* or *Create Request for Change*.

For more information, consult the following resources

- System Recommendation: <https://support.sap.com/sysrec>
- SAP course: SM100 – SAP Solution Manager Configuration for Operations.



LESSON SUMMARY

You should now be able to:

- Implement SAP Notes in the development environment and transport them into the production system
- List the features of System Recommendation
- Describe the technical realization of System Recommendation
- Use System Recommendation to find a SAP correction note to resolve a program error

Preparing Software Maintenance using Maintenance Planner



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- List the goals of Maintenance Planner
- List the key features of Maintenance Planner
- Describe the Landscape Management Process
- Prepare a maintenance transaction
- Perform a maintenance transaction
- Describe the additional features of Maintenance Planner

Goals of Maintenance Planner

Maintenance Planner Motivation

Usually your organization has a complex SAP system landscape with different SAP systems, software products, versions, and components installed. SAP offers different Support Packages, Support Package Stacks, SAP Enhancement Packages or SAP system upgrades for maintenance. Maintenance Planner guides you through the procedure for planning and downloading Support Packages and patches for your SAP systems.

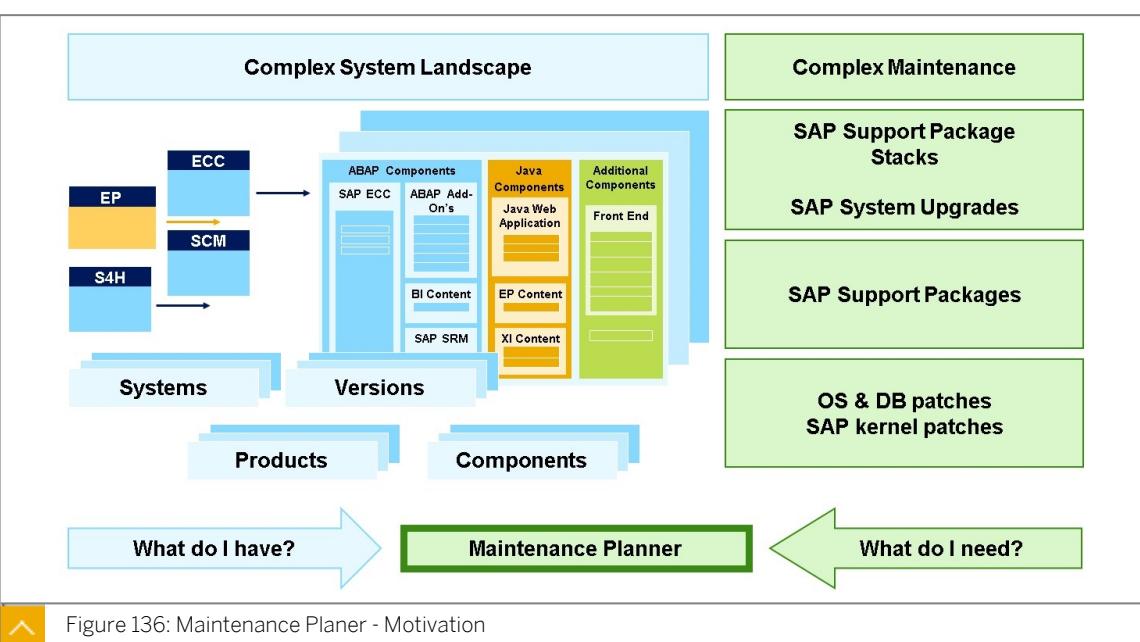


Figure 136: Maintenance Planer - Motivation

Maintenance Planner Introduction

The Maintenance Planner allows planning changes in the landscape based on data provided by SAP Solution Manager Landscape Management Database (LMDB). Maintenance Planner is a hosted solution offered in SAP Support Portal and lets you effectively plan and maintain your landscape.

The Maintenance Planner consolidates critical tasks such as defining product maintenance dependencies, generating stack configuration files, and selecting archives in a single tool.

You can download all corrective software packages using SAP Download Manager,

Maintenance Planner supports all SAP product releases and automatically determines files required for maintenance operation, confirms the file selection, and adds them automatically to the download basket of the registered S-user.

Key Features of the Maintenance Planner

Maintenance Planner – Features

With Maintenance Planner, you can plan a new SAP system installation or an update or upgrade. You can analyze the effect on dependent SAP systems and specify the deployment date.

The Maintenance Planner is an essential administrative tool that helps you complete a major part of your application lifecycle management. Complex maintenance can be planned and scheduled to be deployed at later date.

Maintenance Planner is a hosted application on SAP Support Portal. You could access the tool as follows:

1. Use the tile Maintenance Planner in SAP Solution Manager. Optional use <https://apps.support.sap.com/sap/support/mp>.
2. Log in with your SAP credentials (S-user).



Hint:

One customer number can be assigned to multiple S-users.



Caution:

- Ensure that the S-users you use for Maintenance Planner and the connection with the Support Portal are mapped to the same customer number.
- The destination in the SAP Solution Manager System used to upload the SAP system data to the Support Portal is configured during Mandatory Configuration of SAP Solution Manager.
- You cannot currently use multiple tabs of the same browser instance.

With Maintenance Planner you can perform the following tasks:

- Plan a new SAP system installation.

- Explore all the SAP systems and SAP system tracks in your landscape.
- Plan update or upgrade activities for an existing SAP system.
- Plan an integration of a scenario using Cloud Integration Automation Scenario (CIAS) Tile
- Group SAP systems into tracks and perform collective maintenance.
- Analyze dependent SAP systems impacted by your change.
- Identify and evaluate changes to the landscape.
- Plan a new SAP S/4HANA system or convert an existing SAP ERP system to SAP S/4HANA

and offers the following additional capabilities:

- Visualize all types of systems and connection between them in canvas using different entities of the library
- Identify all the products of your landscape and third-party add-ons deployed in your products
- Identify all products that are out of maintenance



Note:

Since July 2020 the Maintenance Planner offers this additional capabilities. For details please check SAP Note: [2921927](#)- Enhanced Maintenance Planner Onboarding.

The following functions (regarding automatic delta calculation of all relevant maintenance files) are available:

- Installation of Support Package Stacks (SPS)
- Upgrade of SAP Solutions
- Installation of SAP_HR Support Packages
- For AS Java-based SAP systems the existing Java patches for the software components are detected and added to the selection
- Installed add-ons on the SAP system are supported and can be included in the maintenance

Let's have a look at the Maintenance Planner Web UI:

New features available since July 2020 - see SAP Note 2921927

System tracks are groups of SAP systems

Overview of all transactions you created

Determines SAP systems in your landscape

Figure 137: Maintenance Planner – Landing Page

Choose *Explore Systems* to determine the SAP systems in your landscape. The tool displays all the SAP systems in your landscape. It displays name, type, product, verification status of the technical system, and the SAP system type (ABAP, Java, HANA DB, and so on). You can filter to find an SAP system.

Choose *Explore Systems Tracks* to update or upgrade a whole group of technical SAP systems, or a system track, at once. To create a track, use *Explore Systems* → *System details* → *Tracks*.

Choosing *Transactions* gives you an overview of all the transactions you created, ordered by ID, name, status, and so on. For example, when you create a track, the check is carried out in this section.

Choose *Plan a New System* to add a new SAP system to your landscape.

The search field on top (not in the screen shot) lets you find SAP systems, transactions, and tracks. You can search by the name or description of the entity, and the result is grouped into SAP systems, transactions, and tracks.

The Landscape Management Process

Maintenance Planner Prerequisites

Performing a maintenance transaction, which then selects the required files for the update or upgrade, you ensure the prerequisites are in place. The main steps are:

1. Depending on your SAP Solution Manager release, the following *SOLMAN_SETUP* guided procedures have been finished successfully: *SAP system preparation*, *infrastructure preparation*, and *basic configuration*.
2. Your user in SAP Solution Manager is assigned to your S-user for the SAP Service Marketplace. S-users are managed in SAP Support Portal at the URL <https://support.sap.com/user-admin> or <https://support.sap.com/en/my-support/users.html>.

3. To connect the S-user with the SAP system user within the SAP Solution Manager System, the transaction **AISUSER** is used. You do this for the user **SOLMAN_BTC**, which is responsible for the periodic update, and for your SAP Solution Manager System user in case you want to use the direct upload function. For more information, see SAP Note [1822202 - Authorization to upload LMDB data to SAP Support Portal](#).
4. You need to make sure, that the installation number and SAP system number are known in LMDB and SAP Support Portal.
5. Your technical SAP systems are synchronized, from SAP Solution Manager to SAP Support Portal.
6. Check SAP Note [2174410 - Maintenance Planner: Known limitations](#). This note shows if limitations exist which will be resolved as soon as possible. This note will be updated regularly.



Note:

The selected Support Packages can be stored in the Download Basket of the S-user. The Download Basket needs to be installed locally. For details, please refer to SAP Note [330793 - Download patches](#). The Support Packages will be automatically approved during the maintenance transaction so that they can be downloaded from the SAP Software Download Center. Optionally they can be downloaded using the download service in SAP Solution Manager 7.2.

Data and Topology of SLD, LMDB, and SAP Support Portal

Landscape data is uploaded to SAP Support Portal - for example, to enable the EarlyWatch Alert or Maintenance Planner. For this reason, you must guarantee that you get reliable data to the SAP Support Portal. The topology consists of three layers of landscape data tools: The SAP System Landscape Directory (SLD) - optional, SAP Solution Manager Landscape Management Database (LMDB), and the SAP Support Portal Customer Profile.

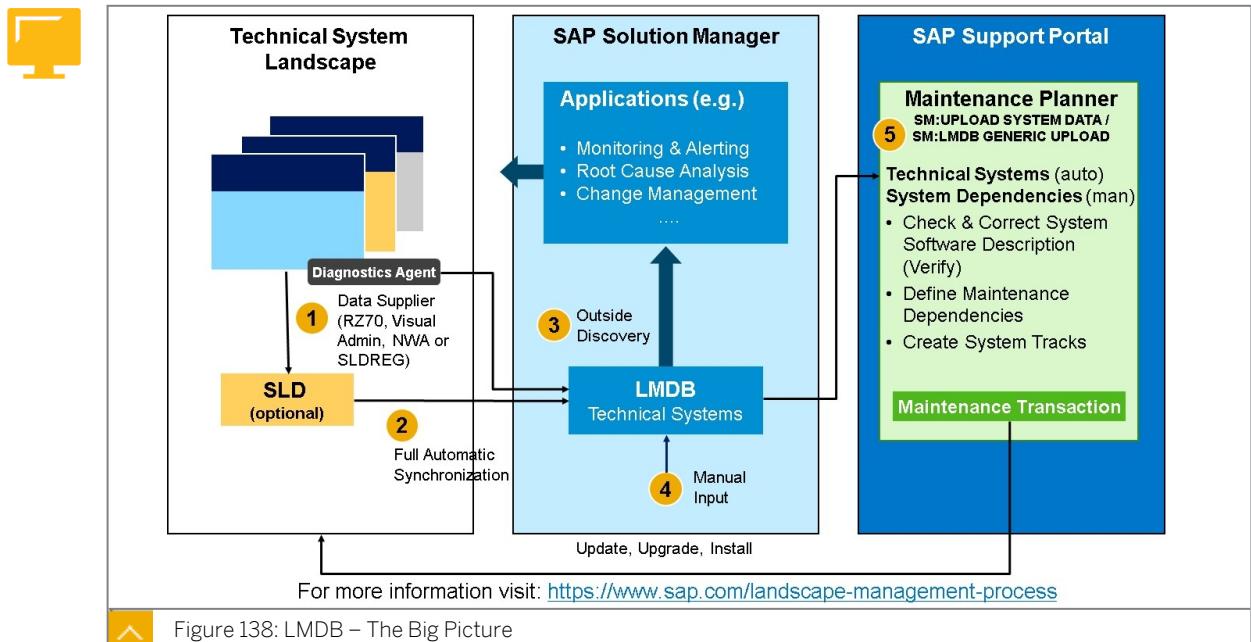


Figure 138: LMDB – The Big Picture

The figure *LMDB: The Big Picture* gives a short introduction on how the SAP system data finds its way to the SAP Support Portal to be used by the Maintenance Planner to perform a Maintenance Transaction. The process is as follows:

1. Data Suppliers automatically register technical SAP systems in the SAP system Landscape Directory (SLD).
2. During the setup of SAP Solution Manager, the landscape description, CIM model, and SAP Software Catalog (SAP CR content) are copied from the SLD to the LMDB by an initial, full synchronization. After this, the incremental synchronization propagates SAP system changes from the SLD to the LMDB every 10 minutes.

Most of the SAP system information in the LMDB is provided by this full automatic synchronization with an SLD in the SAP system landscape, which continuously gathers data sent by the data suppliers.

Since SAP Solution Manager 7.2 SPS 8 a scenario is also available without using the SLD. For more information please visit: <https://blogs.sap.com/2012/10/08/how-to-reduce-manual-effort-in-cim-model-and-cr-content-update-of-sld-systems/>

3. Some information, such as host information, is written by the Diagnostics Agent directly into the LMDB. This is called Outside Discovery.
4. Manual input is included here, if necessary.



Caution:

Some information, like business partner information or additional attributes, can only be created manually in the technical SAP system editor of the LMDB. Apart from these exceptions, do not create SAP system information manually. If the SAP system is subsequently registered by an SLD data supplier, manual information cannot be overwritten because the LMDB has a higher rank than the SLD. So you would have to update manually created data manually, in the future, and it can become obsolete.

5. From the LMDB, technical SAP system information is sent to SAP Support Portal to be used by the SAP Support Portal Application, as Maintenance Planner, for example.

Scenario: High Security Customers and SAP S/4HANA Adoption

- If your organization prohibits external RFC connections (due to security reasons), because of which you do not have an RFC connecting your SAP Solution Manager system to the SAP Backbone systems.
- Or if you want to move to SAP S/4HANA, you can use the System Info XML. Refer to knowledge base article (KBA) [2287046](#) – How to Generate the System Info XML and upload to Maintenance Planner



Hint:

Before a maintenance transaction can be performed, you need at least to verify the SAP system software description stored within the SAP Support Portal Customer Profile.

Prepare a Maintenance Transaction

Verify Systems, Define Tracks and Create Dependencies

Before starting the maintenance activity the System(s) needs to be error free. This means, your system(s) must match with the SAP product model. If this is not the case you need to **verify** the system before you are able to update or upgrade such system(s).

Also keep in mind, you are working with system landscapes. Usually you have more systems connected together to one system landscape. For example three systems running the product instance SAP Solution Manager ABAP and three other systems running the product instance SAP Solution Manager Java.

Also the systems could be connected to each other and performing a maintenance to one system also affects the connected system(s).

Therefore we recommend that you build system **tracks** and **dependencies** to reflect this before starting a maintenance transaction. The next slide illustrates these tree steps:

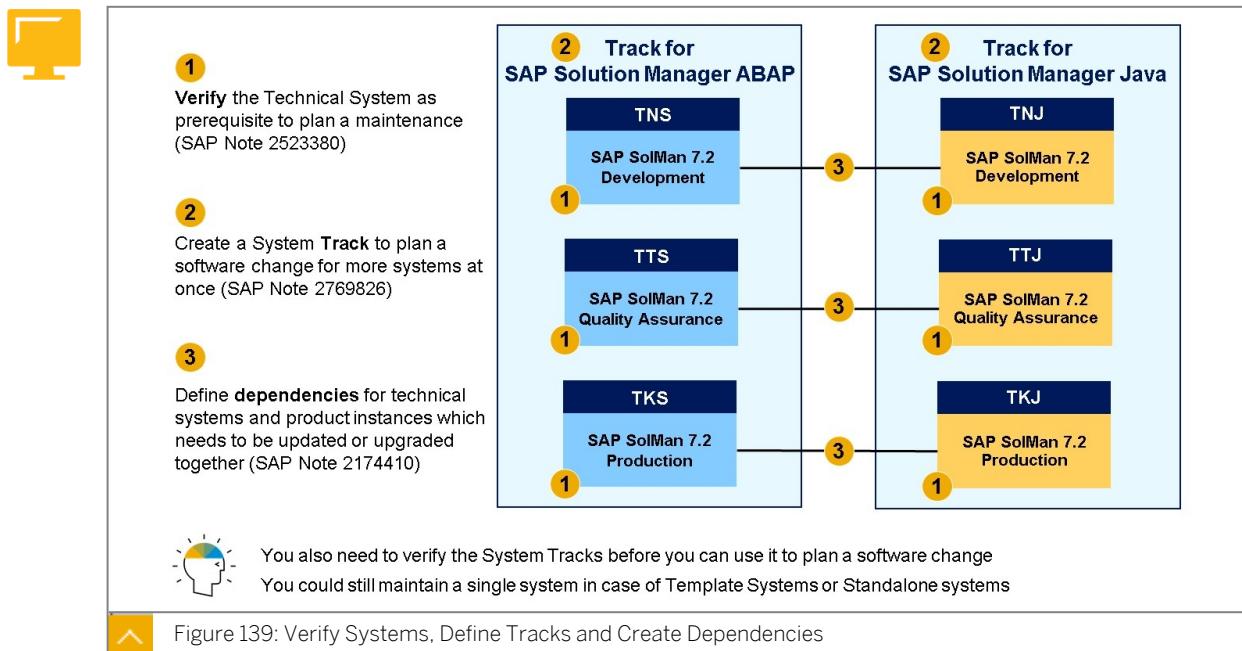
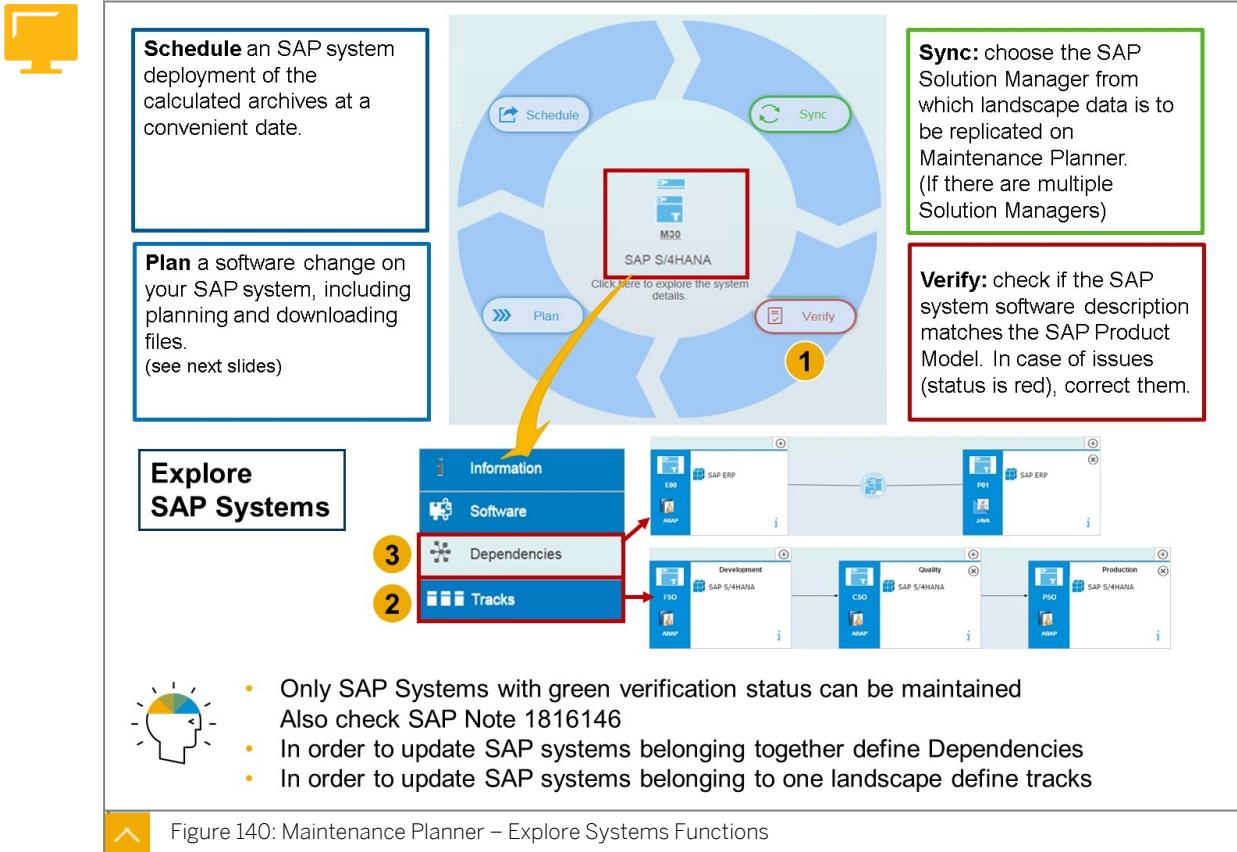


Figure 139: Verify Systems, Define Tracks and Create Dependencies

Using the Explore System functions

When you start the *Explore Systems* function within the maintenance planner you see a list of all the systems stored for your customer number. The resulting list can search for a system identifier or you can use the filter options to filter by System Type, Product or other elements (not shown in the screen shot).

After you select a system you see the following screen (the discussed functions *Verify*, *Dependencies* and *Tracks* are marked red):



First we take a closer look at the functions in the outer circle starting on the top right of the screen with the focus on Verify, Dependencies and Tracks in our case:

Sync Function

In the SAP system maintenance cycle, the Sync button is green if there is only one SAP Solution Manager, which is chosen by default. If the Sync button is yellow, multiple SAP Solution Managers are present.

Verify Function

Your SAP system may be marked with *Error* if the SAP system description does not comply with the SAP product model. You cannot update or upgrade such an SAP system.

Caution:

Ensure that your SAP system complies with the SAP product model. Your SAP system may be erroneous if its description does not comply with the SAP product model.

A system is erroneous because of one or more of the following reasons:

- The system description shows a different software component than the actual software stack installed on the system.
- The system is modeled incorrectly in Landscape Management Database.
- Overlap of data from two or more SAP Solution Managers to maintenance planner.
- Product maintenance dependency not verified for this system.

Example: The software component version of SAP Basis is 740 but the assigned software product version is EHP1 of SAP NetWeaver 7.3. For more information also take a look to the section: Excursus: The SAP product model

Plan Function

The Maintenance Planner allows you to plan an update or upgrade on any SAP system in your landscape. It provides all the update or upgrade options available for the SAP system that you choose. This is discussed in detail during this lesson.

Schedule Function

Schedule a system deployment of the calculated archives at a convenient date.

When you select the System identifier (here M30) you go to another page where you can use the following options:

Dependency Function

A product maintenance dependency defines which technical SAP systems and product instances need to be updated and upgraded together in the maintenance processes.

When you try to add a dependent SAP system, Maintenance Planner proposes all possible maintenance dependencies, based on the product versions.

Tracks Function

A system track is a logical grouping of related SAP systems and contains two or more SAP systems in your landscape on the same target software level. A system track helps you to calculate an update or upgrade to all the SAP systems at once.

Excursus: The SAP Product Model

Most customers have a clear understanding of the term Product and Product Version. For example, SAP NetWeaver 7.5, EHP7 FOR SAP ERP 6.0, SAP S/4HANA 2021 or SAP SOLUTION MANAGER 7.2.

Also, the fact that an SAP System consists of different Software Components is well known. Like SAP BASIS 7.40, SAP ABA 7.40 in case of AS ABAP based SAP Systems or J2EE ENGINE APPLICATIONS 7.50, LM NWA BASIC APPS 7.6 for AS Java based SAP Systems.

But this knowledge is not enough to calculate the maintenance requirements of an SAP System. Here the Product Instance comes into play.

Product Instance

A Product Instances bundles Software Components that have dependencies at runtime and therefore need to run on the same SAP System (could also be called Technical System). One prominent example is SAP ECC. Product Instances are required when you calculate system updates or upgrades using the SAP Maintenance Tools.

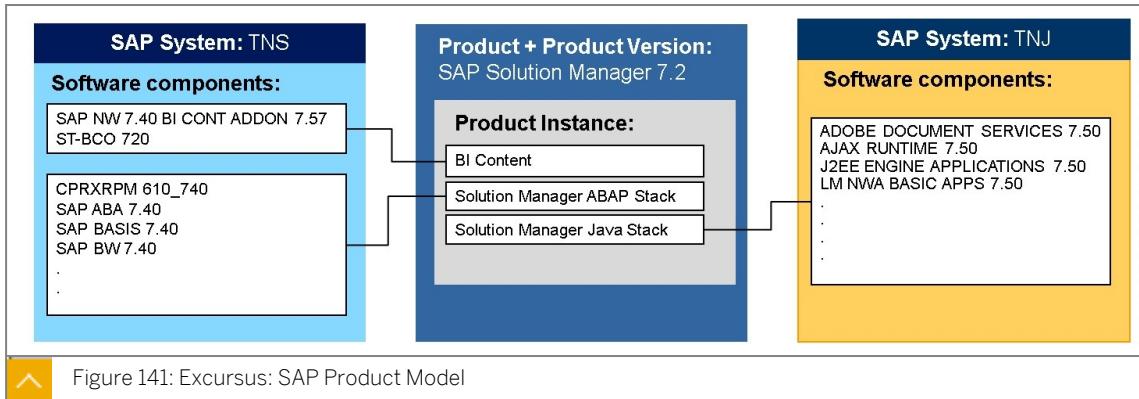
One Software Component can be part of several Product Instances; one Product Instance can contain other Product Instances.

Product Instances have inter-dependencies.



Note:

To reduce complexity, included Product instances are not shown on the following slide. Because Software Components can be part of several Product instances, some Software Components could be part of more than one Product Instance. For example, the Software Component SAP BASIS belongs to the Product Instance Application Server ABAP **and** Solution Manager ABAP Stack. The Product Instance Application Server ABAP is not shown on the slide.



In this example there is an AS ABAP based SAP System called *TNS* on the left-hand side. This system consists of about 20 single Software Components. Only six of them are mentioned on the slide.

On the right-hand side, you see an AS Java based SAP System called *TNJ*. This system consists of about 70 single Software Components, of which four are mentioned on the slide.

In the middle part of the slide, the Product / Product Version of SAP Solution Manager 7.2 is built by various Product Instances like BI Content, Solution Manager ABAP Stack and Solution Manager Java Stack. These Product Instances can be related to one or even many Software Components. The Product Instances run in different SAP Systems in this case.

Only if the systems reflect the SAP Product Model, a maintenance could take place. If the definition is wrong the SAP System gets Erroneous. In this case you need to use the verification tool as part of the Maintenance Planner which helps you to correct the system description.

**Note:**

There could be many reasons why an SAP System gets Erroneous. The verification tool gives suggestions on how to solve the issue. The aim is to get a system description, which reflects exactly the SAP Product Model. This means no Software Component should be missing to define one Product Instance. The dependencies between the Product Instances need to be correct. In case of questions, ask your local Business Process Operation Team, development manager, project lead or create a ticket at SAP.

Please take this seriously, a wrong system definition and a subsequent maintenance may lead to an unresponsive and/or corrupted system, which cannot be used anymore for your Business.

Once the System is verified don't forget to perform the Correction of installed software information (CISI) described in SAP Note: [1816146](#).

Perform a Maintenance Transaction

Planning an SAP system Update or Upgrade

The Maintenance Planner allows you to plan an update or upgrade on any SAP system in your landscape. It provides all the update or upgrade options available for the SAP system that you choose

Prerequisites

For you to perform any kind of update or upgrade, the SAP system has to be in a verified state, indicating that the SAP system description complies with SAP product modeling. A verified SAP system status is green.

For an SAP system that is not in a verified state, the verification status is red in the maintenance cycle for the SAP system; this needs to be corrected before you proceed.

**Note:**

Depending on your update strategy you can update or upgrade a single SAP system using *Explore Systems* → *Plan* or you could update or upgrade a system track which includes two and more SAP systems using *Explore Systems Tracks* → *Plan*. In both cases, SAP system dependencies are recognized.

We recommend that you perform maintenance at the system track level where possible.

**Hint:**

If, for some reason, you need to repeat the maintenance transaction after you apply the software changes to the first SAP system within your system track, you can change the software calculation as the Maintenance Planner is able to calculate the required files for the update or upgrade for each SAP system separately.

**Hint:**

For complex upgrades, it might be useful to copy the development SAP system to a sandbox SAP system. In this case, you can perform the maintenance transaction first for your sandbox SAP system and afterwards for the complete system landscape (System Track).

The next slide shows all the steps you need to go through when performing a maintenance transaction:



Start Maintenance Transaction using
Explore System Track → Plan or Explore Systems → Plan

**Define Change:**

Select update option like *Minimum Software Change*, *Plan a Maintenance*, *Plan a <Product Version> Upgrade*, *Install or Maintain an Add-on*, etc..

The following symbols are used: = to be installed = to be updated

**Select Files (OS/DB dependent files):**

Select OS/DB depending files like SAP Kernel, SAP Host Agent, SAP IGS, Software Update Manager (SUM), including Data Base Migration (DMO)...

**Select Files (stack dependent and independent files):**

Check selected stack dependent and independent files, optional add SAP_HR and EA-HR Support Packages, add Java patches...

**Download Files:**

Download Stack XML file (one file for all SAP systems in case of a system track), push files to Download Basket of S-user. Optional: generate side effect report, check upgrade dependencies, check SAP security notes, add additional files, download stack text file.

**Complete:**

Save if the maintenance is still ongoing (in order to change it, if it fails) or set to *Complete* if SAP system or all SAP systems within the track are updated.



Figure 142: Maintenance Planner – Steps to perform a Maintenance Transaction

Step Define Change

Keep in mind that the options are calculated automatically, based on your SAP system landscape. Therefore the options available are dependent on your SAP system landscape and the software level of the SAP Support Portal application Maintenance Planner.

**Hint:**

The current software stack is displayed in the left panel as you make your planned changes.

Based on the option you choose, you will have to select other parameters, like the target software levels and instances.

1. In our example you want to update SAP S/4HANA On-Premise 1511 from Support Package Stack (SPS) 0 to Support Package Stack (SPS) 02. Usually, the SAP AS ABAP 7.5 is also updated automatically, from Support Package Stack (SPS) 01 to Support Package Stack (SPS) 03 in your case.

2. You also want to update all Add-Ons, so you select *all* Add-Ons and choose which components you want to update to which level.

**Hint:**

The options are calculated automatically, based on your SAP system landscape. The options available in your individual case may vary.

3. Each selection that has been made needs to be confirmed using the *Confirm Selection* button.
4. Finish the process if all Product Instances you want to maintain are on the right Target Software Level. As an indicator, you may use the green symbols *to be installed* and *to be updated* in the *Target Software Details* section. If this is not the case, repeat step 1 and maintain the missing Product Instances.
5. If you have selected all Product Instances you want to update or upgrade choose *Next*.

Step Select Files (OS/DB Depending files)

In this step you have to manually select the correct kernel files for your SAP system. Note that there might be application servers running on different operating systems.

**Note:**

In an AS ABAP system, you can find the release information for each instance using transaction SM51. Then choose *Go to → Server → Information → Release Information*.

1. If you now choose a database-dependent part of your SAP kernel, the system automatically selects the appropriate database-independent kernel. Also choose the SAP IGS and the Software Update Manager.

Depending on your SAP Host Agent, the update strategy may or may not choose the SAP Host Agent.

**Note:**

As an administrator, you may want to use the automatic update feature of the SAP Host Agent as described in SAP Note: [1479374 - Using the SAP Host Agent Auto Upgrade Feature](#), to keep all SAP Host Agents in your SAP system Landscape on the same patch level. However, SUM may request a newer version of the SAP host agent. Therefore you could download the newest SAP Host Agent version here in case the SUM requests a newer version.

2. To confirm your selection, choose *Confirm Selection*.
3. Choose *Next* to proceed to the next step.

Step Select Files (stack-dependent and stack-independent files)

1. Check the stack-dependent and stack-independent files. Make sure that in the case of an AS-ABAP-based SAP system, the entry SAP SPAM / SAINT update is selected. This should be done automatically.



Note:
Stack-independent files are not deployed automatically. You need to maintain them manually.

2. In Systems where HR is used (for example SAP ECC) you can select the relevant HR Support Packages by choosing the button *Add HR Packages*. This option is available on your SAP system in the following circumstances:

- You have a valid RFC destination to the managed SAP system maintained in SAP Solution Manager.
- SAP HR or EA-HR or HR-CEE software component is installed in the SAP system.

3. Choose *OK*.



Note:
In AS Java based SAP Systems, proceed as follows:

- a. Choose *Add Java Patches*.
- b. Select the Java files to be downloaded.
- c. Include the Java patches separately or include them in the stack XML.
- d. Choose *OK*.

4. If you have finished your selection choose *Next*.

Step Download Files

1. Choose *Download Stack XML*.
2. Choose *Push to Download Basket*. The Maintenance Planner confirms all files in the download basket of the S-user with which you logged on.

Download Files, you now take the following steps:



Note:
These steps only simulate the changes to your SAP system. Actual changes will be made to your SAP system when you perform the steps in the section implementing an SAP system change using SUM.



Note:
You can now download the installation files from your download basket (for example at <https://launchpad.support.sap.com/#/downloadbasket>).

**Hint:**

In the case of SAP AS ABAP 7.0 or lower, you might need to download the Stack text file.

All files (this means the files within the download basket and the Stack.XML files) must be distributed to the appropriate directories of the SAP system(s). Only then you can start the implementation of these file with the appropriate tool (SPAM, SAINT, SUM, ...).

For all other options available here please consult the Maintenance Planner User Guide.

Step Complete

Finally, you complete the process:

1. Check the target status and set to complete. Optionally download PDF for documentation reasons.
2. If the maintenance of the involved SAP systems is not complete, choose Save. If the maintenance has been completed choose Set to Complete.

Additional Features of Maintenance Planner**Maintenance Planner – Additional features**

Maintenance Planner allows you to do more than simply calculate which updates and upgrades are possible. A lot of additional features are also available. This section gives a quick overview about the different functions.

Planning a New System Installation

Maintenance Planner helps you plan a new system installation on your landscape.

At the end of this procedure you will be able to generate a Stack XML file and delta archives that will contain every detail of the system and the software components. These are used by Software Provisioning Manager for the installation process.

**Note:**

To use the generated Stack XML file by SWPM use the following command syntax to call SWPM: <path_to_directory_where_you_extracted_SWPM>/sapinst SAPINST_STACK_XML=<absolute_path_to_location_of_STACK_XML_file>

Execute Plan

You can also execute a pre-planned system installation.

Installation of SAP Fiori Apps with Maintenance Planner

SAP Fiori apps require front-end, back-end and if applicable, SAP HANA components. The required front-end and back-end components are delivered in separate products, and have to be installed in the system landscape.

Through integrated SAP Fiori installation planning procedure, Maintenance Planner addresses the need to install front-end and back-end components delivered in separate products, by calculating all the system requirements for selected SAP Fiori Apps in SAP Fiori Apps Library.

Procedure:

1. Log in to the SAP Fiori app library: <https://www.sap.com/fiori-apps-library>.
2. Log in with your S-User.
3. Choose the SAP Fiori app to install on your SAP system.
4. Choose *Find out more*.
5. Read the product features from the app library. To proceed with the installation, choose the *Implementation Information* tab. SAP system details, such as Front-End Components, Back-End Components and Prerequisite for installation, are displayed.
6. To use Maintenance Planner to install the above components, choose the *Maintenance Planner* link.
7. The Maintenance Planner Launchpad for Fiori installation is displayed.

Upgrade Dependency Analyzer

With Upgrade Dependency Analyzer integrated into Maintenance Planner, you can identify how an upgrade on any one of your systems in the landscape affects another system in your landscape.

Cloud Integration Automation Service for SAP Cloud Integration Scenario

The scenario *Plan for Cloud Integration* provides a guided workflow to integrate SAP cloud solutions. The list of integration scenarios you see are based on the tools support based on your licensed solutions.

These additional services and content are voluntarily provided by SAP and can be withdrawn, postponed, or suspended at any time. For more information, see: [SAP Note 2608492](#)

Hybrid Landscape Visualization

Hybrid Landscape Visualization lets you view and create landscape pictures to graphically visualize systems in your landscape. There are no landscape pictures available by default, but you can create a landscape picture by choosing any entity from your library. Here, you can also search existing landscape by their names.

Product Analytics

This section lets you quickly analyze the systems in your landscape. You can use one of the following queries available:

- *Provisioned Products*: Displays the top provisioned products
- *Out Of Maintenance Product Versions*: Displays all the product versions scheduled to go out of the maintenance
- *Third Party Add-Ons*: Displays all the third party add-ons present in your landscape.

View Recommended Notes

This new feature identifies SAP Notes with automatic correction instructions for ABAP systems for your SAP system landscape. The following are some of the key functionalities:

- Tailored recommendations for each ABAP system, based on system landscape and SAP Notes metadata
- SAP Notes recommendation, which includes Security Notes, Hot News, Performance Notes, Legal Changes, and License Audit Notes

- Options to filter and sort systems to get the total number of recommendations for each system
- Display SAP Notes and its link to view its details and implementation prerequisites
- Link to the system details

Reference: Find More Information About Maintenance Planner

Related Information Maintenance Planner

- SAP Note [2174410](#) - *Maintenance Planner: Known limitations*
- Maintenance Planner in Help Portal: https://help.sap.com/docs/MAINTENANCE_PLANNER
- Maintenance Planner in Support Portal: <https://support.sap.com/solution-manager/processes/maintenance-management/maintenance-planner.html>
- Maintenance Planner Wiki Page: <https://wiki.scn.sap.com/wiki/display/SM/Maintenance+Tools%3A+Maintenance+Planner+and+Maintenance+Optimizer>
- Troubleshooting Guide: <https://wiki.scn.sap.com/wiki/display/SL/Maintenance+Planner+-+Troubleshooting+Guide>
- FAQ: <https://blogs.sap.com/2016/01/25/maintenance-planner-frequently-asked-questions/>
- Landscape Management Process: <https://www.sap.com/landscape-management-process>
- IT Management Blog: <https://scn.sap.com/community/it-management/blog/2015/04/30/topology-of-sld-lmdb-and-customer-profile-how-to-get-reliable-landscape-data-in-sap-support-portal-as-a-basis-for-planning>
- Component in SAP Support Portal: BC-UPG-MP



LESSON SUMMARY

You should now be able to:

- List the goals of Maintenance Planner
- List the key features of Maintenance Planner
- Describe the Landscape Management Process
- Prepare a maintenance transaction
- Perform a maintenance transaction
- Describe the additional features of Maintenance Planner

Describing Support Package Manager (SPAM) and Software Update Manager (SUM)

LESSON OVERVIEW

In this lesson, you will learn about the Support Package Manager. You will learn what a SPAM/SAINT update is and how to apply it. A SPAM/SAINT update may be required before you can import new Support Packages. In any case, you should always apply the latest SPAM/SAINT update before importing Support Packages.

Business Example

Functional enhancements to an SAP system can affect not only the applications, but also the actual tools that you use to import the Support Packages.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe Support Package Manager (SPAM) and Software Update Manager (SUM)

SPAM versus SUM

Support Packages can be imported either with the Support Package Manager (SPAM) or with the Software Update Manager (SUM). The Add-on Installation Tool (SAINT) is used to install and update Add-ons. It is a second interface of the SPAM.

The figure below lists possible cases where you might prefer one tool over the other.



SPAM/SAINT	SUM
Easy handling	Short downtime possible, depending on scenario strategy
Short runtime	Shows point of no return with reset option
Test mode, showing modification preview	Performs several checks, e.g. database free space
Does not require Maintenance Planner	Stack.xml from Maintenance Planner required for most scenarios
Runs as transaction, inside SAP system, possible as dialog or batch	Runs separately, outside SAP system
Can work with third party add-ons unknown to Maintenance Planner	Can restart the SAP system, can update the kernel

Figure 143: SPAM versus SUM



Note:
A stack.xml file is required to import Support Packages for the SUM for AS ABAP-based SAP system. This stack.xml file is generated by the Maintenance Planner.

In short, you can say: **SPAM** should be used for easy handling and the **SUM** should be used when you have to import a large number of Support Packages into a downtime-critical SAP system.



Note:
For restrictions of transaction **SPAM** in SAP S/4HANA, see SAP Note [1803986](#): *Rules to use SUM or SPAM/SAINT to apply SPs for ABAP stacks.*

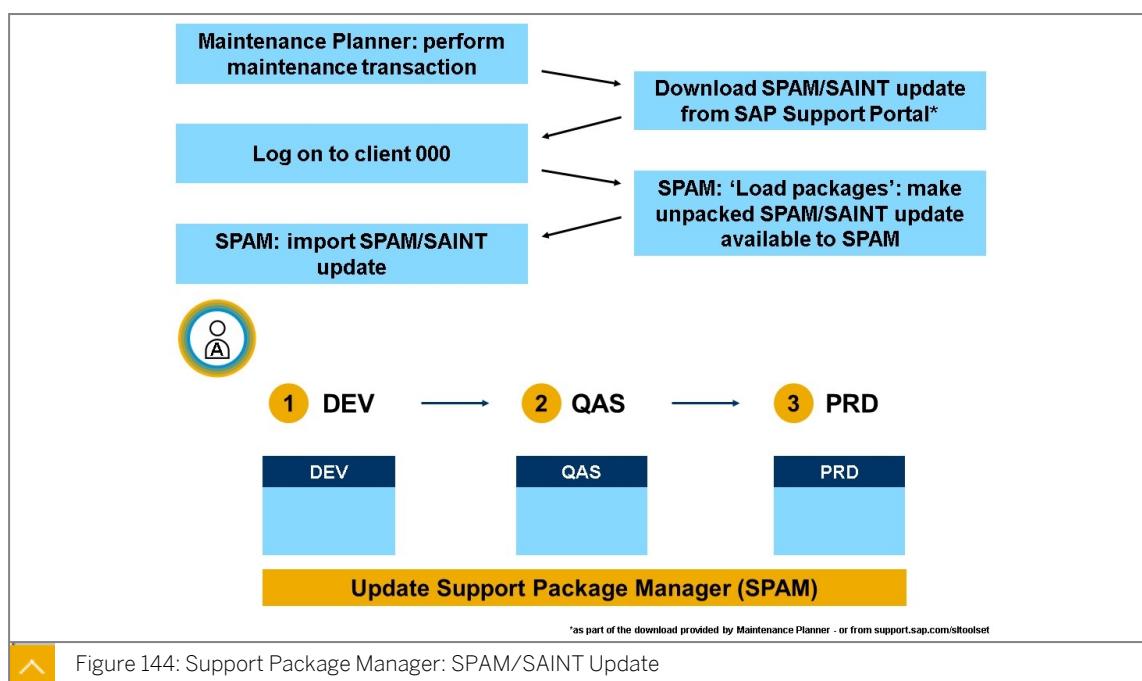
SPAM/SAINT Updates

Most standard software components of an SAP system are installed during the initial installation process. Additional software components can be imported afterwards, if required. These are referred to as add-ons. Examples of add-ons are plug-ins or industry-specific components that are not part of the standard installation. These add-ons are imported and updated to a newer release using the Add-On Installation Tool, transaction **SAINT**. The Add-On Installation Tool and add-ons themselves will not be dealt with in detail in this lesson.

Support Packages are imported with the Support Package Manager, transaction **SPAM**.

Transactions **SPAM** and **SAINT** are two different interfaces for the same functions. So they are updated together, if necessary.

The figure *Support Package Manager: SPAM/SAINT Update* illustrates how you can update the Support Package Manager and the Add-On Installation Tool itself via transaction **SPAM**.



The Support Package Manager offers the following functions:

You can use the Support Package Manager (transaction SPAM) to import individual Support Packages for individual components. You can also import several consecutive Support Packages for a software component. Additionally, you can import several Support Packages for various software components (a vector). If you want to import a vector of Support Packages, you can switch the queue to vector selection, with *All Components* in the selection dialog box, or from the start with *Extras → Settings → Define Queue*.



Hint:

You should always import all the Support Packages of a Support Package stack – not only Support Packages of a single software component.



Note:

It is also possible to use the stack XML file that the Maintenance Planner calculated during the maintenance transaction. This imports the Support Packages that were calculated by the Maintenance Planner.

When you import Support Packages into the SAP system with the Support Package Manager, a fixed sequence of steps is processed. If the import process for the Support Package terminates, processing can be restarted at the break-off point. The process restarts at the point at which it was terminated.

You can also use a special import procedure to reduce the downtime, by choosing *Extras → Settings → Import Queue → Import Mode: Downtime-minimized*. The individual phases of the Support Package Manager are grouped into four modules. You can determine the start times of the modules individually. The individual modules can also be scheduled for background processing.

The Support Package Manager can identify dependencies (but not side effects) between the different Support Packages and take them into account. To remain up to date and also be able to use the new functions of the Support Package Manager, you should update the Support Package Manager itself before you import Support Packages. Since transactions SPAM and SAINT are closely related from a technical point of view, a shared patch, the SPAM/SAINT update, is supplied at irregular intervals to upgrade the tools. You can find out the correct version from the short description of transaction SPAM, for example, *Support Package Manager - Version 7.58/0088*.



Hint:

The update for the Support Package Manager (SPAM/SAINT update) is only delivered in the English and German languages. It is therefore recommended that you log on in English or German when working with the Support Package Manager.

The SPAM/SAINT update is imported using transaction SPAM.

You can only import a SPAM/SAINT update if there are no terminated Support Packages in the SAP system. If terminated Support Packages exist, a dialog box notifies you of this fact, and you can take one of the following actions:

- You can import the queue of Support Packages completely and then run the SPAM/SAINT update.

- You can reset the status of the queue, import the SPAM/SAINT update first, and then the Support Package queue. To reset the status of the queue, choose *Extras → Reset Status → Queue*.

The procedure for importing a SPAM/SAINT update is as follows:

1. Check whether the current SPAM/SAINT update on the SAP Support Portal is newer than the SPAM/SAINT update in your SAP system (the SPAM/SAINT version of the SAP system is displayed in the title bar of the Support Package Manager).
2. Download the newest SPAM/SAINT update from the SAP Support Portal. The newest SPAM/SAINT update is also calculated by the Maintenance Planner during a maintenance transaction.
3. There are three different methods to provide the SPAM/SAINT update to the Support Package Manager:
 - copy the SPAM/SAINT archive to the front end host
 - copy the SPAM/SAINT archive to the transport directory, subdirectory *EPS*, subdirectory *download*
 - Unpack the downloaded SPAM/SAINT archive to the transport directory, subdirectory *EPS*, subdirectory *in*

The unpack method is described in the following steps:

4. Log on to client 000 of the SAP System and call transaction **SPAM**.
5. Register the patch (the SPAM/SAINT update) to the Support Package Manager using *Support Package → Load packages → EPS Files from Application Server*. Despite the menu description, this does not import the SPAM/SAINT update directory, nor does it load anything from the application server. It merely registers the packages to the Support Package Manager. No packages are loaded. They are searched for on the host where the application server is running, not on the application server.
6. To import the SPAM/SAINT update, choose *Support Package → Import SPAM/SAINT Update*.



Note:

Instead of manually extracting the archive files, there is also the possibility to load and decompress the files from the front-end host or the SAP system host via transaction **SPAM**.

In a multi-system landscape, you have to perform this import process individually in each SAP system. In a three-system landscape, first into the development system, then into the quality assurance system, then into the production system.

Since the SPAM/SAINT update is a full (100%) delivery of transactions **SPAM** and **SAINT**, the latest available SPAM/SAINT update can be imported immediately. If your current version is 7.58/0088, for example, you can import version 7.58/0092 directly.



Hint:

Importing a SPAM/SAINT update only affects transaction SPAM and SAINT – it does not require any downtime for the SAP system nor a maintenance window.

Software Update Manager (SUM)

You can also import Support Packages with the Software Update Manager (SUM). The SUM was originally designed as a tool for performing SAP system upgrades. The central SUM tool is SAPUp. You can also use the SUM to import Support Packages and in many other scenarios.

In contrast to the Support Package Manager (SPAM), the SUM is not a transaction, but a standalone tool.

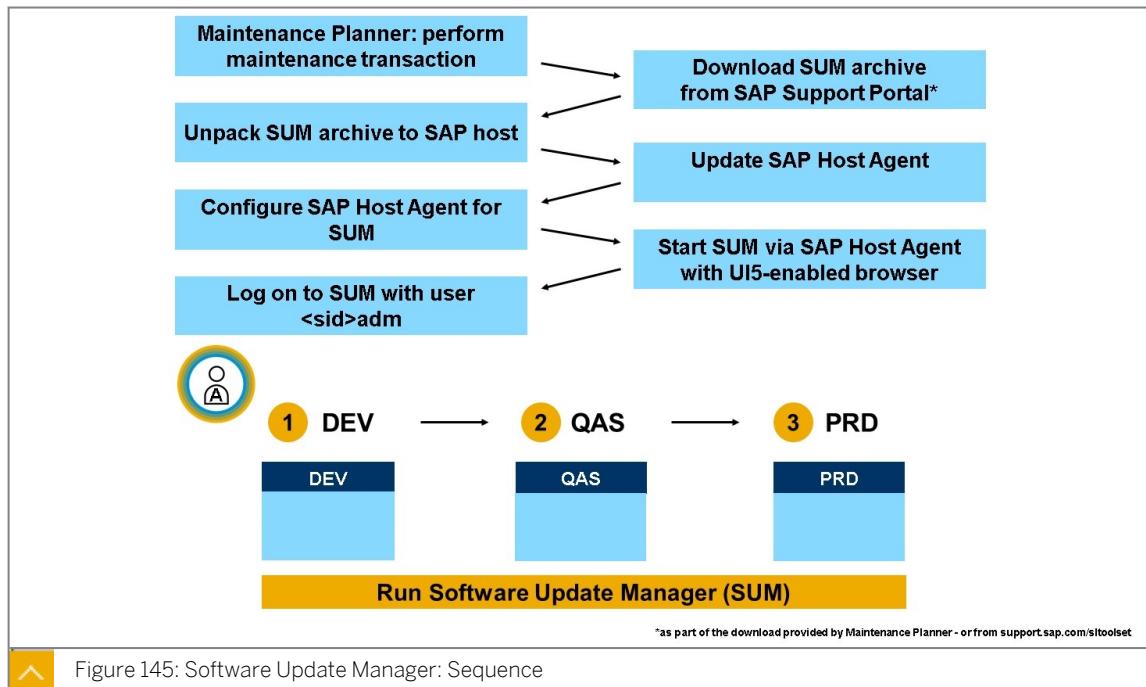


Figure 145: Software Update Manager: Sequence

You can download the SUM archive from support.sap.com/sitoolset.

There are two versions of SUM available:

- SUM 1.1 for AS ABAP based SAP systems with a target of SAP_BASIS 740 and below
 - SUM 2.0 for AS ABAP based SAP systems with a target of SAP_BASIS 750 and above
- SUM 1.1 is also used for AS Java based SAP systems, independent of the release.



Note:

In the year 2022 SUM 1.1 replaced SUM 1.0.

To start the SUM, perform the following steps, as shown in the figure *Software Update Manager: Sequence*:

1. Copy the SUM archive to the host where the SAP system is running, such as directory `/usr/sap/SYS/<SID>` - or any other directory. The directory should have around 20-200 GB of free space - depending on the action you are performing.
2. Unpack the SUM archive - this creates several thousand files.
3. If necessary, update the SAP Host Agent, which is required for the connection between the SUM UI and the SUM.
4. Configure the SAP Host Agent from within the SUM directory via `STARTUP confighostagent <SID>` (SUM 1.1) or from within the subdirectory `abap` from the SUM directory via `SUMSTART confighostagent <SID>` (SUM 2.0). For a Windows operating system, the SID of the SAP system can be omitted. Use a user with root permission on a Linux or Unix operating system, use user `<sid>adm` on a Windows operating system.
5. Use a UI5-enabled browser to connect to `https://<host name>:1129/lmsl/sumabap/<SID>/sluigui` (SUM 1.1) or `https://<host name>:1129/lmsl/sumabap/<SID>/slui` (SUM 2.0) - this connects the UI (the browser) via SAP Host Agent to the SAPup.
6. Log on to the SUM with the user `<sid>adm`.

When using the SUM, you perform these steps on the development system first, then on the quality assurance system, and then on the production system – just like with transaction SPAM.



LESSON SUMMARY

You should now be able to:

- Describe Support Package Manager (SPAM) and Software Update Manager (SUM)

Importing Support Packages with the Support Package Manager (SPAM)

LESSON OVERVIEW

In this lesson, you will learn how Support Packages are imported into your SAP systems.

Business Example

Importing Support Packages is the responsibility of the SAP system administrator. The new versions of SAP objects that are contained in the Support Packages stabilize the SAP system and enhance its functional scope. They are also needed to adapt the SAP system to legal changes.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Import Support Packages with the Support PackageManager (SPAM)

Import Support Packages



Hint:

Support Packages can be imported using either the Support Package Manager (transaction SPAM) or the Software Update Manager (SUM). In the lesson *Describing Support Package Manager (SPAM) and Software Update Manager (SUM)*, you compared the differences, advantages, and disadvantages of both tools. In this lesson, you will learn about how to import Support Packages with the Support Package Manager.

The Support Package stacks represent combinations of Support Packages recommended by SAP for the individual SAP system. SAP recommends to import them regularly.

When importing Support Packages, you should observe the instructions for the Support Package stack and import all the Support Packages of the Support Package stack in one step, if possible. A kernel patch is often part of a Support Package stack. You should implement it before you import the Support Packages (not covered in this lesson).

There is no common, generally valid Support Package stack. The Support Package stack depends – among other things – on the releases of all software components of the individual SAP system and on the current Support Package levels of the individual software components. A Support Package stack must be calculated individually for a specific SAP system with the Maintenance Planner.

**Hint:**

Support Packages always belong to a specific release of a specific software component. This means, for example, there is not just a Support Package 4, but instead S4CORE 104, Support Package 4. S4CORE 106, Support Package 4 and SAP_HR 608, Support Package 4 would be something completely different.

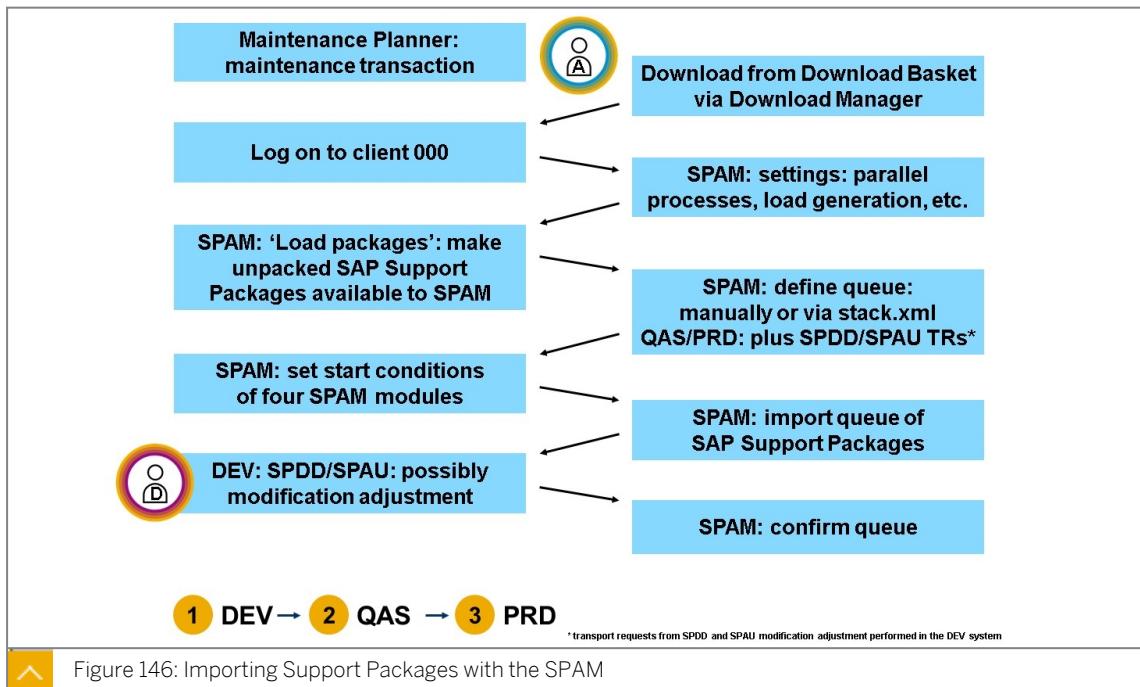


Figure 146: Importing Support Packages with the SPAM

SAP recommends the following procedure for importing Support Packages, as shown in the figure above:

1. Start the Maintenance Planner. The Maintenance Planner calculates the necessary combination of Support Packages (Support Package stack) for the SAP system you want to update. You should also implement the current SPAM/SAINT update. The Maintenance Planner can approve the selected Support Packages for the download and place them in the download basket for the Download Manager.
2. You can download the required Support Packages from SAP Support Portal using the Download Manager.
3. The downloaded files must be unpacked to the transport directory of the relevant SAP system, subdirectory *EPS* subdirectory *in*.
4. Log on to client 000 of the SAP system and call transaction SPAM.
5. The new Support Packages can be announced to the Support Package Manager through *Support Package* → *Load Packages*

Here you will find three alternatives to search and load Support Packages, where you can use one of the following options:

- *EPS Files from Application Server* to search for unpacked Support Packages in the directory *trans* → *EPS* → *in*

- SAR Archives from Frontend to search on the front-end PC for SAR files
- SAR Archives from Application Server to search in the directory *trans → EPS → download* for SAR files

In any case: This does not import the Support Packages; it is simply a registration in the Support Package Manager.

6. Configure the settings for the import process.
7. The queue of Support Packages to be imported is defined. You can select either Support Packages of an individual software component or Support Packages from multiple software components - or just provide the stack XML file from the Maintenance Planner. The Support Package Manager checks the queue for consistency. The Support Package Manager ensures that only the Support Packages appropriate for the specific SAP system are displayed in the queue. Support Packages that are intended for a different release or a software component that is not installed, or whose predecessor relationships are not fulfilled, do not appear in the queue, even if they have been registered in the SAP system.



Hint:

You should always import all the Support Packages of a Support Package stack in one step, that is, in a single queue.



Note:

If it exists, you should provide the stack XML file, calculated by Maintenance Planner.

8. To import the Support Package queue, choose *Support Package → Import Queue*.
9. Choose the start conditions for the four import modules.
10. Start the import process. One or two modification adjustments may arise during the import. The developers have to carry them out at the required times. You cannot perform them later: SPDD modification adjustment for the dictionary objects, and SPAU modification adjustment for the non-dictionary repository objects.
11. As soon as the Support Packages have been imported successfully, check the logs with the path: *Goto → Import Logs → Queue*.
12. When the Support Packages have been imported into the development system, the developers can check the objects. After the Support Packages have been imported into the quality assurance system, an acceptance test should be carried out by subject matter specialists. This will result in several transport requests, that contain the necessary adjustments. You can then import the Support Packages into the production system, followed by the several transport requests, containing the adjustments from the acceptance test.
13. If the import of the Support Packages into the SAP system has been completed successfully, the imported queue has to be confirmed using *Support Package → Confirm*.

There are a number of possible settings for the import process (*Extras → Settings*). Choose *Import Queue → Scenario* to determine which actions you want to carry out during the import:

- You use the standard scenario to import Support Packages completely. All steps are performed. The import mode Downtime-minimized can be used here, to slightly reduce the downtime.

This downtime reduction option of SPAM is - by far - less effective than using SUM to import Support Packages.

- You can use the test scenario to observe whether a modification adjustment is required before you import the Support Packages, or to check whether conflicts occur that must be resolved before the import. No data or objects are imported to the SAP system during the test scenario.

There is no test scenario for SPAM/SAINT updates. The selection you make here is ignored when importing a SPAM/SAINT update.

There are many steps, or phases, to the import procedure. They are subdivided into four modules.

- The modules can be executed individually in the correct sequence.
- The modules can run in the background.
- The start time of the modules can be controlled flexibly.

The four modules perform the following steps:

1. Preparation module: All the preparatory steps and check steps (such as the test import and add-on conflict check) are performed in this module. The module can run during production operation. Once this module has been executed, you still have the option of resetting or deleting the queue. If you continue with the Import 1 module, data is changed in the database and the queue can no longer be reset or deleted.
2. Import 1 module: In this module, the dictionary objects of the Support Packages are imported, but without activating them. If the import procedure is carried out in Downtime-minimized mode, the ABAP programs are also imported inactively. The running SAP system cannot "see" these changed ABAP programs yet. The manual modification adjustment of the dictionary objects is performed at the end of this module, if required (SPDD modification adjustment). If you can guarantee that no manual changes to the repository are required for the SAP system and no transports are imported into the SAP system during this time, this module can also run during production operation.
3. Import 2 module: Now, the dictionary objects are activated. The remaining import steps are carried out. If you use the Downtime-minimized import mode, inactive ABAP programs are activated in this module as well. During this module, production operation is not permitted, to avoid inconsistencies and data loss. Also, the modification adjustment of the non-dictionary repository objects (SPAU modification adjustment) is performed in this module. This module will always require a downtime for the SAP system, which means you need a maintenance window.



Hint:

The SAP system administrator is responsible for adhering to the downtime window: All users must be logged off, background processing must be completed, and RFC communication and other interface communication must be prevented. These actions are not carried out automatically by SPAM - they are the duty of the SAP system administrator.

**Note:**

When selecting *Downtime-minimized*, the downtime of the SAP system can be reduced. But the runtime will be extended, because the ABAP programs have to be activated in module *Import 2*. Module *Import 1* will run longer, also, because the imported objects have to be kept in parallel to the active objects with methods of version history.

4. Post Processing module: In this module, all the clean-up steps are processed. Production operation can continue.

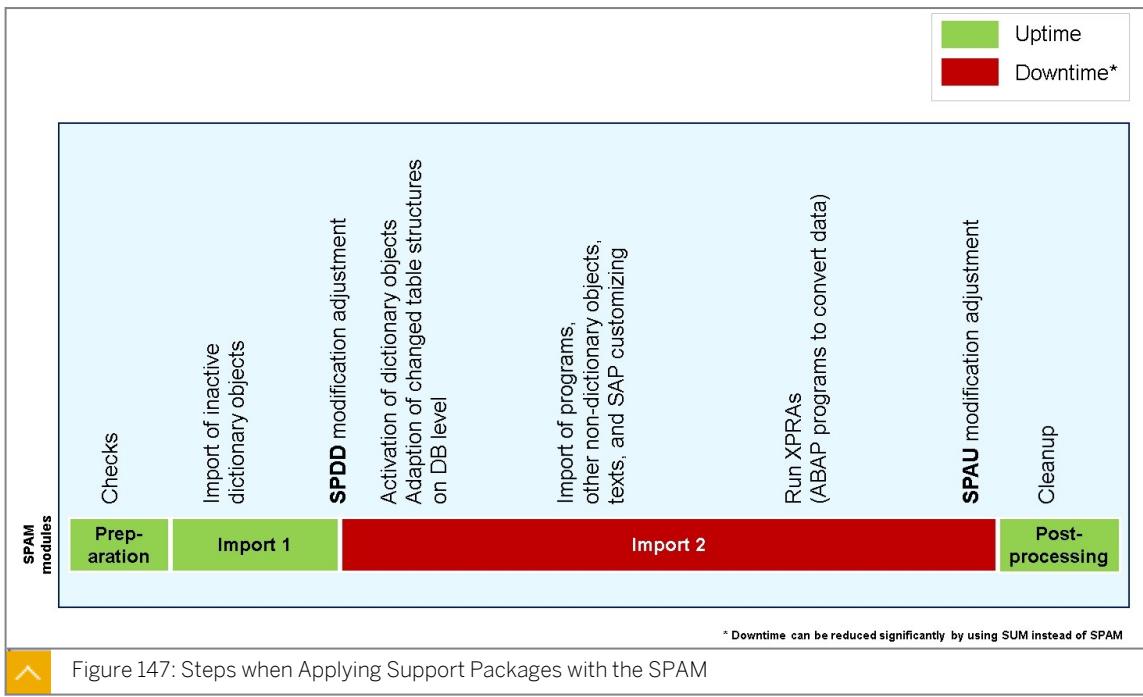


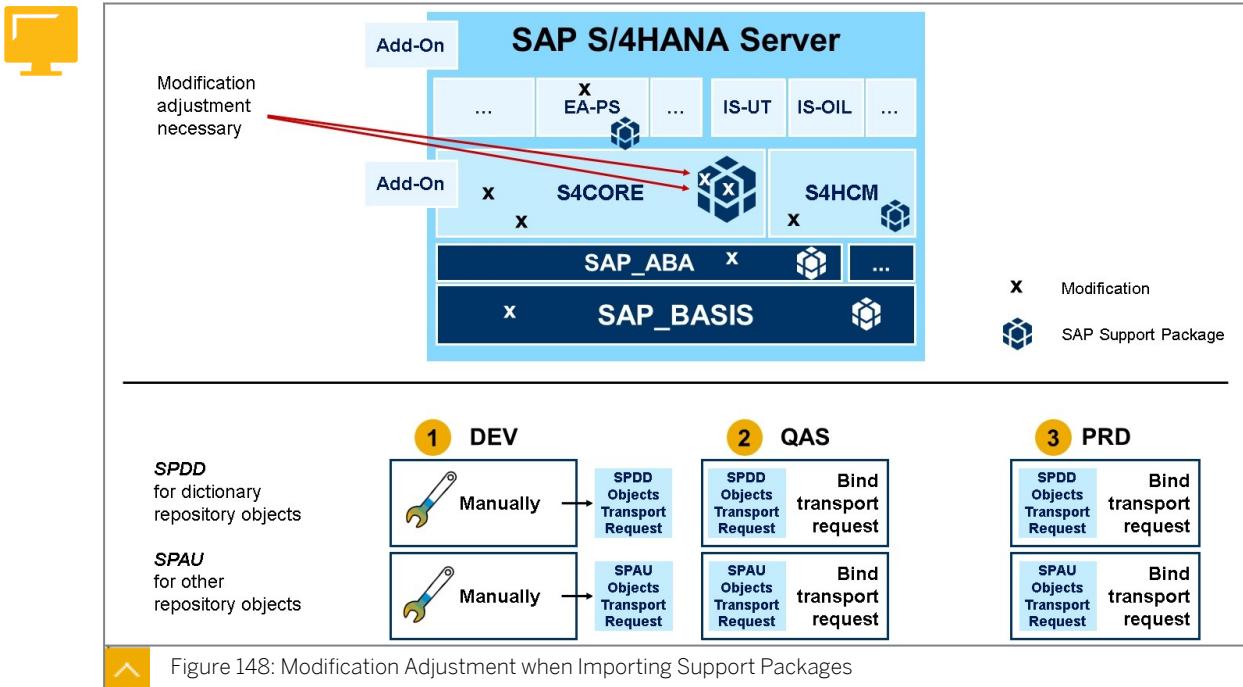
Figure 147: Steps when Applying Support Packages with the SPAM

**Caution:**

The SAP system administrator must manage the downtime manually. This means that the administrator is responsible for locking the SAP system, logging off the users, stopping batch processing, interrupting RFC connections and other interface communication, and so on.

With SAP S/4HANA 2020, transaction **SMAINTENANCE** allows you to define a maintenance period. During this maintenance period, normal users will not be able to log on. Only system administrators with a special security policy can connect (transaction **SECPOI**, attribute **TENANT_RUNLEVEL_LOGON_PRIVILEGE**). In addition, only admin batch jobs will be executed (all other jobs will be on hold). For more information, see the SAP Online Documentation for SAP S/4HANA (*Product Assistance*), area *Enterprise Technology* → *ABAP Platform* → *Administering the ABAP Platform* → *Administration Concepts and Tools* → *Administration of Application Server ABAP* → *Maintenance Mode*.

Transaction **SMAINTENANCE** focuses on cloud-based SAP systems.



If SAP objects were modified by the customer and then imported with a Support Package, the objects have to be adjusted when they are imported, as indicated in the figure *Modification Adjustment when Importing Support Packages*. If these objects are not adjusted, the customer's modifications that have been made to the SAP objects are reset to the SAP delivery status. Use transaction SPDD to adjust dictionary repository objects and transaction SPAU to adjust non-dictionary repository objects.

The modification adjustment for dictionary repository objects is performed using transaction SPDD. This modification adjustment must be carried out completely before you can continue with the import. Otherwise, the modifications to the dictionary repository objects are lost, leading to possible data loss.

Caution:

The SPDD modification adjustment should be performed manually in the development system and by binding the resulting transport request in the subsequent SAP systems of the landscape. It is not possible to cover the SPDD modification adjustment of the subsequent SAP systems by importing a transport request!

The modification adjustment for non-dictionary repository objects is performed using transaction SPAU.

You can perform the SPAU adjustment in each SAP system of a multi-SAP system landscape manually. Alternatively, this modification adjustment can be performed once in the development system and imported to the downstream SAP systems using the resulting transport request (which is not possible in transaction SPDD). To do this, however, you should compare the modification status of the individual SAP systems in the landscape beforehand. You can check these at any time, for example, with the Modification Browser (transaction SE95). Better than importing the transport request from the development system is to bind the transport request to the subsequent SAP systems!

To perform an SPDD or SPAU modification adjustment, proceed as follows:

- To enable developers to perform the modification adjustment, create a transport request in the Transport Organizer, and under the transport requests, create tasks for the developers. Only create one transport request for the adjustment of dictionary repository objects (SPDD) and a second transport request for the adjustment of non-dictionary repository objects (SPAU).



Hint:

SPDD and SPAU are development activities. Accordingly, they are carried out in the development client (such as client 100), not in client 000.

- The developers can now perform the modification adjustment. To do so, the developers can use transactions SPDD and SPAU in the development client. After the adjustment is complete, the developers must release their tasks. The administrator releases the transport requests only when the import process is complete.
- You flag the modification adjustment transport requests (choose *Modification Adjustment* → *Assign Transport* in transactions SPDD and SPAU). When you import the Support Packages to the downstream SAP systems of the landscape, these modification adjustment transports can be included in the respective Support Package queue. This means manual modification adjustments are no longer required in the downstream SAP systems.
- Choose *Confirm Adjustment* and confirm the prompt.
- To continue the import process, choose *Support Package* → *Import Queue*.

The Support Package Manager now continues processing and displays the status.



The downtime must be managed manually by the SAP system administrator.



It is also possible to import SAP Support Packages with the Software Update Manager (SUM).



Figure 149: Additional Hints



LESSON SUMMARY

You should now be able to:

- Import Support Packages with the Support PackageManager (SPAM)

Unit 7

Lesson 6

Importing Support Packages with the Software Update Manager (SUM)



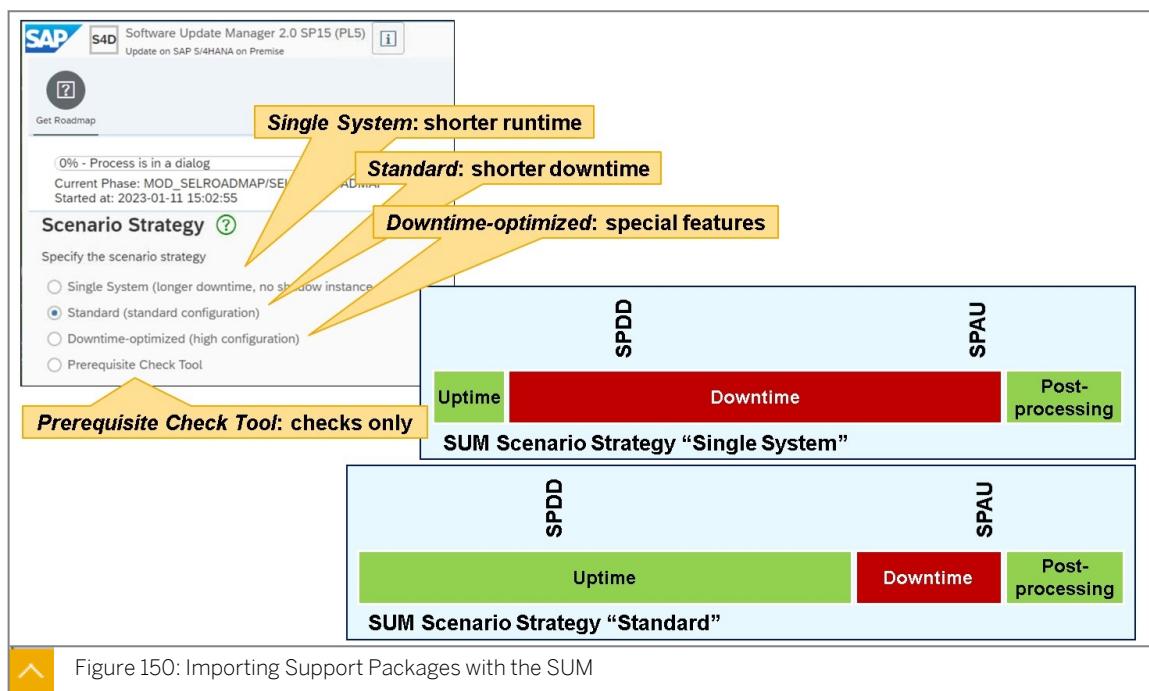
LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe the idea of importing Support Packages with the Software Update Manager (SUM)

Software Update Manager (SUM)

You can also import Support Packages with the Software Update Manager (SUM). Importing Support Packages with the SUM is more difficult and more complex for the SAP system administrator, but results in shorter system downtime – assuming the SUM is configured properly. Therefore, SAP recommends this method when you have to import a large number of Support Packages into a downtime-critical SAP system.



When selecting the scenario strategy *Single System*, the Support Packages are directly imported into the SAP system. This has the advantage, that it is the fastest method. But it causes almost 100% downtime for the entire procedure.

When choosing *Standard*, the SUM works with a second, temporary SAP system, the so called *shadow system*. The advantage is, that many steps can be processed in the shadow system, while the real SAP system is running. This saves most of the downtime. But it takes more

time, to build up the shadow system, and it costs temporarily more database space, more disc space, and more main memory to run the shadow system.

Depending on the SUM scenario, sometimes the option *Downtime-optimized* is available. Special additional actions are performed in order to decrease the downtime even more. An example is the *nZDM* - the near Zero Downtime Maintenance capabilities of SUM.

As the figure *Importing Support Packages with the SUM* shows, you can reduce the SAP system downtime significantly by choosing the scenario strategy *Standard*.



Note:

In older versions of SUM 2.0 and in all versions of SUM 1.1 there is a scenario strategy *Advanced*. This is the same as *Standard*. In SUM 2.0 there is no scenario strategy *Advanced* anymore.

For a comparison of transaction SPAM and the SUM, see the lesson *Describing Support Package Manager (SPAM) and Software Update Manager (SUM)*.

Although the stack XML file is provided to SUM, SUM suggests to import more Support Packages, than the calculated Support Package Stack. For this, these Support Packages have to be provided to SUM before starting SUM.

Component	Release	Start Level	Minimum Level	Equivalence Level	Maximum Level	Calculated Level	Target Level	Status
Software Components			3	3			3	
EA-HR	608	105	114		114		114	
...								
S4CORE	106	2	3	3	3		3	
...								
SAP_BASIS	756	2	3	3	3		3	
SAP_BW	756	2	3	3	3		3	
SAP_GWFND	756	2	3	3	3		3	
SAP_HR	608	105	114		114		114	
SAP_UI	756	3	5	5	5		5	

Figure 151: Support Package Selection

SUM can import the Support Packages from the calculated Support Package Stack - according to the stack XML file. These Support Packages are displayed in column *Target Level*. If SUM finds additional Support Packages, provided in the download directory for SUM, SUM suggests to apply these additional Support Packages, also. For this you can choose *Take over calculated level*. These additional Support Packages are displayed in column *Calculated Level* - if they fulfill all restrictions. After choosing *Take over calculated level* and *Next*, these additional Support Packages appear in column *Target Level*.

The restrictions are:

- *Minimum Level*: the minimum level to fulfill the dependencies between this Support Package and the Support Packages of all other software components.
- *Equivalence Level*, applies only in case of an upgrade: minimum level to avoid a downgrade of this software component.
- *Maximum Level*: the maximum level to fulfill the dependencies between this Support Package and the Support Packages of all other software components.



LESSON SUMMARY

You should now be able to:

- Describe the idea of importing Support Packages with the Software Update Manager (SUM)

Explaining SAP System Upgrade and SAP S/4HANA Conversion

LESSON OVERVIEW

In this lesson, you will learn about the concept of an SAP system upgrade and an SAP S/4HANA conversion.

Business Example

You need new business functions for an existing SAP system. Patching the SAP systems with Support Packages is not sufficient. Therefore you have to bring the SAP system to a new release or even convert your SAP ECC system to an SAP S/4HANA system.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the idea of an SAP System Upgrade and an SAP S/4HANA Conversion

SAP System Upgrade and SAP S/4HANA Conversion

Support Packages are imported for the following purposes:

- To correct errors in the SAP standard.
- To implement legal changes.
- To eliminate system vulnerabilities.
- To (sometimes) implement small functional enhancements.

Support Packages do not change the release level of a software component in the SAP system. An upgrade is required to increase the release level of the software components of an SAP system.



Hint:

In old releases, an upgrade was sometimes called *SAP Enhancement Package installation*.

During an upgrade, software components – of an SAP S/4HANA Server system, for example – are exchanged by newer versions. Afterwards, individual business processes can be activated. Depending on the amount of activated business functions, the project effort required for testing, technical acceptance, adjustments, and re-implementation of business processes can differ significantly.

The Switch Framework is a way to deliver new functions in an inactive way. The Switch Framework allows SAP to make changes to repository objects and roll them out, without these changes becoming effective immediately in the customer system. If you want to use the

new, changed business processes, you have to activate them in the customer system by activating the corresponding business function.



Hint:

The Switch Framework is only available in AS ABAP-based SAP systems with AS ABAP 7.00 and above.

The targeted activation of business processes, or business functions, enables you to keep the project effort required for updates and upgrades within reasonable limits.



Caution:

In practice, you always have to include Support Packages in the upgrade process. The changes delivered by Support Packages cannot be activated. Therefore, you always have to perform a full regression test, even if you do not activate a single business function – almost as if you imported Support Packages, only.

The result of an upgrade of an SAP ECC system is a newer version of the SAP ECC system. The result of an upgrade of an SAP S/4HANA Server system is a newer version of the SAP S/4HANA Server system.

You can convert your SAP ECC system to an SAP S/4HANA Server system by performing an SAP S/4HANA Conversion.



Examples:

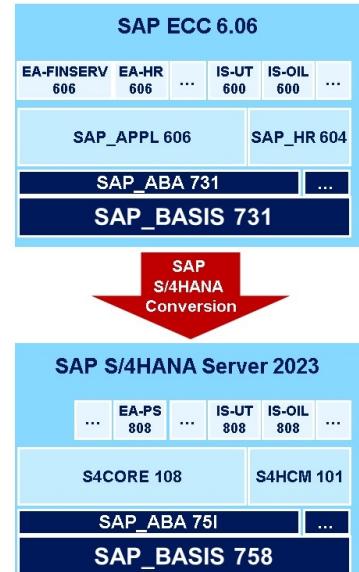
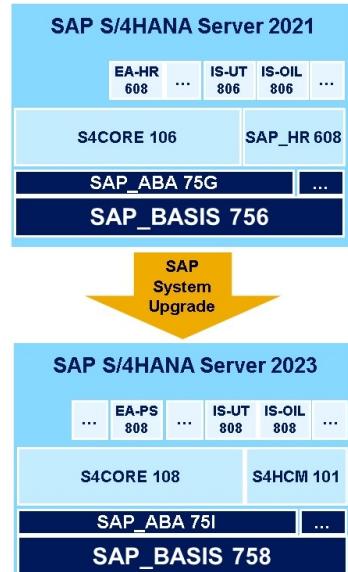
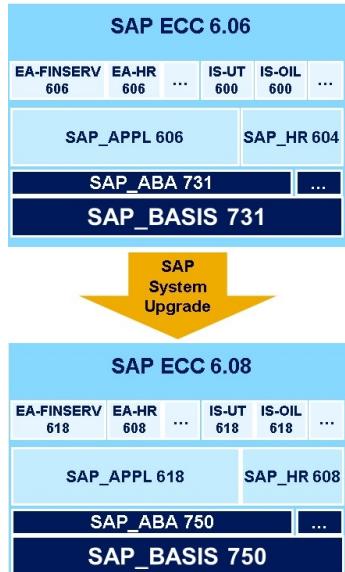


Figure 152: SAP System Upgrade and SAP S/4HANA Conversion

As the figure above shows, an SAP system upgrade is a release change of individual software components. To reduce effort, new, changed, and enhanced business processes (business functions) do not have to be activated until they are needed. When performing an SAP S/4HANA conversion, some business functions are always active – the *always on* business functions.

**Note:**

The technique of selective activation of business functions is available in AS ABAP-based SAP systems, only.

Since all software components are newly installed during an upgrade, you can upgrade from any supported start release directly to the newest target release. For example, you can upgrade directly from the very old SAP ECC 6.00 to the current SAP ECC 6.08.

You can perform an SAP S/4HANA conversion for an SAP ECC system with release 6.00 and above to the current SAP S/4HANA Server release.

The SAP system upgrade and the SAP S/4HANA conversion are performed using the Software Update Manager (SUM).

**Note:**

An SAP system upgrade and the SAP S/4HANA conversion causes a temporary technical downtime in the SAP system.

**Hint:**

During an SAP system upgrade and the SAP S/4HANA conversion, you always have to select Support Packages for the target release of your software components. Otherwise you risk the loss of data by performing a partial downgrade of functionality! This is being assured by Maintenance Planner and SUM.

Related Information

SAP course ADM328 - SAP S/4HANA Conversion and SAP System Upgrade.



LESSON SUMMARY

You should now be able to:

- Explain the idea of an SAP System Upgrade and an SAP S/4HANA Conversion

Learning Assessment

1. How can you maintain an AS ABAP based SAP system?

Choose the correct answers.

- A By generating ABAP loads.
- B By applying SAP Notes.
- C By importing Support Packages.
- D By installing the SAP Support Portal.

2. What are Support Packages used for?

Choose the correct answers.

- A To maintain application data.
- B To import SAP objects to keep the SAP system up to date.
- C To implement adjustments in the SAP system due to changes in legal requirements.
- D To transport programs developed by the customer.
- E To import all the transport requests that SAP makes available on the SAP Support Portal.

3. What characterizes a Support Package Stack best?

Choose the correct answer.

- A It contains one or several changes for a single object
- B It resolves mismatch issues between SAP objects and customers modifications
- C It contains a validated target combination of Support Packages

4. The Maintenance Planner can be used to plan a new installation of an SAP system.

Determine whether this statement is true or false.

- True
- False

5. Identify the activity that you can execute using transaction SPAM (Support Package Manager).

Choose the correct answer.

- A Updating the kernel.
- B Importing a new add-on.
- C Importing Support Packages.
- D Splitting off the ABAP Central Services Instance (ASCS).

6. You always have to install all available SPAM/SAINT updates in the sequence of their release, because they are built on one another.

Determine whether this statement is true or false.

- True
- False

7. The queue for importing the SPAM/SAINT update must be confirmed manually.

Determine whether this statement is true or false.

- True
- False

8. A requirement for importing a SPAM/SAINT update is that the SAP system does not have any terminated import processes for Support Packages.

Determine whether this statement is true or false.

- True
- False

9. Support Packages can be imported in any order.

Determine whether this statement is true or false.

- True
- False

10. To obtain a short downtime, you choose the SUM strategy Standard.

Determine whether this statement is true or false.

- True
- False

11. During the conversion of an SAP ECC system to an SAP S/4HANA Server system the SAP system will receive the software component *S4CORE* which will replace the software component *SAP_APPL*.

Determine whether this statement is true or false.

True

False

Learning Assessment - Answers

1. How can you maintain an AS ABAP based SAP system?

Choose the correct answers.

- A By generating ABAP loads.
- B By applying SAP Notes.
- C By importing Support Packages.
- D By installing the SAP Support Portal.

You are correct! SAP Notes and Support Packages are used to maintain an AS ABAP based SAP system. Generating ABAP loads is not maintenance. You cannot install the SAP Support Portal. Read more in unit 7, lesson 1 of the course ADM100.

2. What are Support Packages used for?

Choose the correct answers.

- A To maintain application data.
- B To import SAP objects to keep the SAP system up to date.
- C To implement adjustments in the SAP system due to changes in legal requirements.
- D To transport programs developed by the customer.
- E To import all the transport requests that SAP makes available on the SAP Support Portal.

You are correct! Support Packages are not used to enter application data. Customer developments are transported to other systems using transport requests. Transport requests from SAP are imported with the Transport Management System (TMS). Read more in unit 7, lesson 1 of the course ADM100.

3. What characterizes a Support Package Stack best?

Choose the correct answer.

- A It contains one or several changes for a single object
- B It resolves mismatch issues between SAP objects and customers modifications
- C It contains a validated target combination of Support Packages

You are correct! Support Packages Stacks contain a validated target combination of Support Packages. Individual objects will be corrected using SAP Notes, while Modification Adjustment Transport resolve modification issues. Read more in unit 7, lesson 1 of the course ADM100.

4. The Maintenance Planner can be used to plan a new installation of an SAP system.

Determine whether this statement is true or false.

- True
- False

You are correct! In spite of its name, the Maintenance Planner can also be used to plan a new installation of an SAP system. Read more on this in the lesson “Preparing for Software Maintenance” of the course ADM100.

5. Identify the activity that you can execute using transaction SPAM (Support Package Manager).

Choose the correct answer.

- A Updating the kernel.
- B Importing a new add-on.
- C Importing Support Packages.
- D Splitting off the ABAP Central Services Instance (ASCS).

You are correct! Read more on this in the lesson “Preparing for Software Maintenance” of the course ADM100.

6. You always have to install all available SPAM/SAINT updates in the sequence of their release, because they are built on one another.

Determine whether this statement is true or false.

True

False

You are correct! During a SPAM/SAINT update, all the repository objects for SPAM and SAINT are imported (100% patch), so you do not have to follow a given sequence. Read more on this in the lesson “Describing Support Package Manager (SPAM) and Software Update Manager (SUM)” of the course ADM100.

7. The queue for importing the SPAM/SAINT update must be confirmed manually.

Determine whether this statement is true or false.

True

False

You are correct! It is neither necessary to define a special queue, nor to confirm this non-existent queue. Read more on this in the lesson “Describing Support Package Manager (SPAM) and Software Update Manager (SUM)” of the course ADM100.

8. A requirement for importing a SPAM/SAINT update is that the SAP system does not have any terminated import processes for Support Packages.

Determine whether this statement is true or false.

True

False

You are correct! This is true: there must not be any terminated import processes for Support Packages in your SAP system. Read more on this in the lesson “Describing Support Package Manager (SPAM) and Software Update Manager (SUM)” of the course ADM100.

9. Support Packages can be imported in any order.

Determine whether this statement is true or false.

True

False

You are correct! Support Packages must be imported in ascending order, without any gaps. Read more on this in the lesson “Importing Support Packages with the Support Package Manager (SPAM)” of the course ADM100.

10. To obtain a short downtime, you choose the SUM strategy *Standard*.

Determine whether this statement is true or false.

True

False

You are correct! You choose SUM strategy *Standard* for a shorter downtime, you choose SUM strategy *Single System* for a shorter runtime. Read more on this in the lesson “Importing Support Packages with the Software Update Manager (SUM)” of the course ADM100.

11. During the conversion of an SAP ECC system to an SAP S/4HANA Server system the SAP system will receive the software component *S4CORE* which will replace the software component *SAP_APPL*.

Determine whether this statement is true or false.

True

False

You are correct! During the conversion of an SAP ECC system to an SAP S/4HANA Server system the SAP system will receive the software component *S4CORE* which will replace the software component *SAP_APPL*. Read more on this in the lesson “Explaining SAP System Upgrade and SAP S/4HANA Conversion” of the course ADM100.

Lesson 1

Exploring SAP Spool System

303

Lesson 2

Creating Output Devices

307

Lesson 3

Working with Device Types

317

Lesson 4

Working with Spool Servers

325

Lesson 5

Monitoring and Troubleshooting Printing Issues

331

Lesson 6

Planning the SAP Print Architecture

343

Lesson 7

Appendix: Managing Output Devices with Printing Assistance for Landscapes (PAL)

349

UNIT OBJECTIVES

- Illustrate the information flow from the print document to printout
- Explain central terms in SAP Printing and explain their meaning
- Classify Printing Methods
- Create Output Devices
- Select a suitable device type
- Illustrate the idea of Unicode Printing Enhancement
- Explain the use of logical spool servers

- Create a logical spool server
- Summarize the classification of spool servers and output devices
- Use monitoring transactions for printing
- Perform housekeeping activities for spool
- Analyze printing errors
- Develop a Printing Concept
- Explain the basic idea of Printing Assistance for Landscapes (PAL)
- List the steps to set up PAL
- Describe the distribution of printer groups to target system groups

Unit 8

Lesson 1

Exploring SAP Spool System



LESSON OBJECTIVES

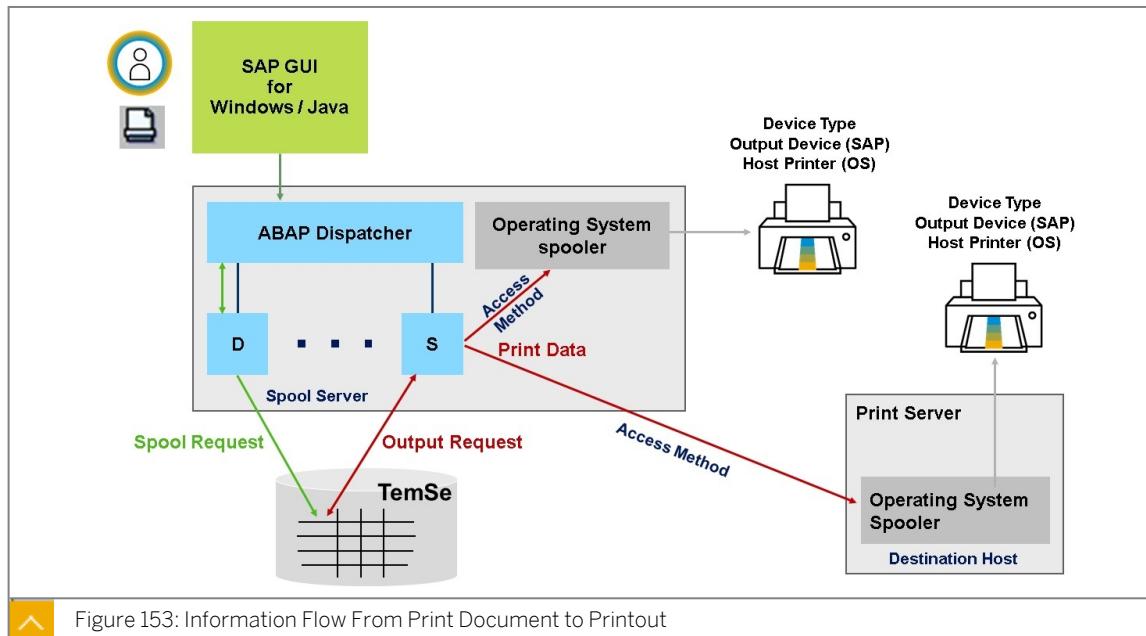
After completing this lesson, you will be able to:

- Illustrate the information flow from the print document to printout
- Explain central terms in SAP Printing and explain their meaning

The Printing Process

There are various document classes in the SAP system (such as report lists and SAPscript or SAP Smart Forms documents). Although the way documents are created may be completely different, the output on paper is always performed using a common mechanism. The SAP spool system creates a spool request and an output request.

- The **spool request** contains device-independent print data, such as administrative information (author, date, number of copies), and the actual print data.
- The **output request** is created only when the document is sent to a particular output device. In the output request, the device-independent print data from the spool request is converted to the printer language that the selected output device understands.



As the figure *Data Flow During Printing* shows, when the user releases a document for printing, a spool request is first created. Once the document is sent to the output device, an output request is generated. This procedure allows the user to display a spool request before output. One spool request can have several output requests. This can prevent the user from

recreating (possibly with a great deal of effort) a spool request, if, for example, the toner in a printer is exhausted or there was the wrong paper in the tray. The user can also create a spool request and an output request at the same time by choosing the *Print immediately* option.



Hint:

By choosing the options for *spool control* on the *Defaults* tab in transaction SU3, end users can choose to print their spool requests immediately by default.



Note:

The print data of a spool request is stored in TemSe (a store for temporary sequential objects). You can define whether the objects are to be stored by TemSe in the SAP database or in the file system by setting the profile parameter *rspo/store_location*.



Caution:

Do not mix the terms *Printing* (explained in the figure and the text above) and *Output Control*:

SAP S/4HANA introduces a new style of output management. The new output management is going to be the successor of all other output management frameworks (*SD Output Control*, *FI Correspondence*, *FI-CA Print Workbench*, *CRM Post-Processing*). The output management for SAP S/4HANA comprises all functionalities and processes that are related to the output of documents.

Among others, this includes the integration of output management functions into the business applications as well as the technical infrastructure.

For more information, see the online documentation for SAP S/4HANA (*Product Assistance*), area *Cross Components* → *SAP S/4HANA Output Control* and

- SAP Note [2228611 – Output Management in SAP S/4HANA](#)
- SAP Note [2791338 –FAQ: SAP S/4HANA output management](#)
- the blog *Output Management in SAP S/4HANA* (<https://blogs.sap.com/2021/04/26/output-management-in-sap-s-4hana/>) and

Basic Terms in SAP Printing

The configuration of the SAP spool system is a system administration task. You must create a device definition in the SAP system for every output device on which you want to print a document, so that the SAP system recognizes this device. This is done in transaction SPAD.

The device definition contains all of the important data for addressing the device. When creating a new output device, (among others) you need to provide the following information:

- The **output device** is the name of the printer in the SAP system. It can be up to 30 characters long, and is case-sensitive.
- Creating an output request prompts the SAP spool system to send a (usually) printer-dependent, completely formatted data stream to a printer through an operating system

spooler (OS spool). This means that the addressed printer model must be known by the SAP system. These definitions of printer models in the SAP system are called **device types**.



Note:

The device type **SWIN** transfers the spool data from the SAP system to the Microsoft Windows printer driver. This is useful, for example, when different printers are used for front-end printing in a Microsoft Windows environment.

- A **spool server** denotes an SAP application server with spool work processes or a logical server that is to prepare the output for this device.
- To be able to administer the printer from within the SAP system, it must be known at the operating system level. A spool work process can reach an operating system spooler in several ways using different **host spool access methods**. The access method is identified by a letter and informs the system which architecture is in use (local printing / remote printing, with / without *SAPSprint* service program, ...) and which operating system you are using.
- The **host printer** is the name of the printer at the operating system level (in the host system).



Note:

The name of the host printer is case-sensitive. Under Microsoft Windows, there should be no blank in the host printer name, and network printers are addressed using their UNC names (in the format \\R12345\P42).

- The **destination host** is used only for remote printing and is the name of the host on which the OS spooler is running and which receives the data for output. For local printing, the field for the destination host is replaced by the field **host name**, which is filled automatically when you specify the spool server.



LESSON SUMMARY

You should now be able to:

- Illustrate the information flow from the print document to printout
- Explain central terms in SAP Printing and explain their meaning

Creating Output Devices



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Classify Printing Methods
- Create Output Devices

Introduction to Access Methods

All hardware and software components involved in printing are combined in the print architecture.

Depending on the number of printers in your landscape and on your printing requirements, you may want to use different printing methods, such as:

- Local printing
- Remote printing
- Front-End Printing

Local Printing

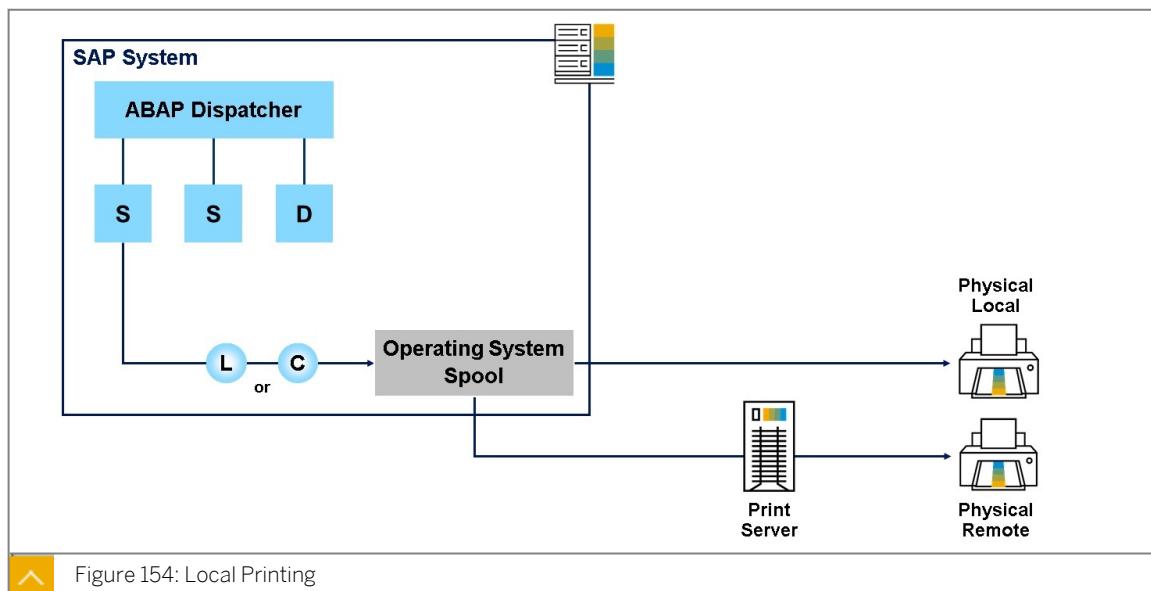


Figure 154: Local Printing

As the figure *Local Printing* shows, it is irrelevant whether the printer is directly connected to this host, or is reached over a network (and possibly another print server). The spool work process passes on its data locally to the OS spooler that is on the same host.

- On Linux/Unix systems, the print data with access method **L** is output using operating system methods, such as the commands **lp** or **lpr**. The specific syntax is defined in profile parameters (see SAP Note [15355](#) – Command line parameter for access method “L”).
- Under Microsoft Windows, access method **C** passes the data directly to the operating system print API.

Local printing is the fastest and most reliable connection from the SAP system to the OS spooler – the data does not need to be transferred over a network connection to reach the host spool system. As soon as the spool work process has transferred its data, it can deal with new output requests, even if the operating system spooler is still occupied.

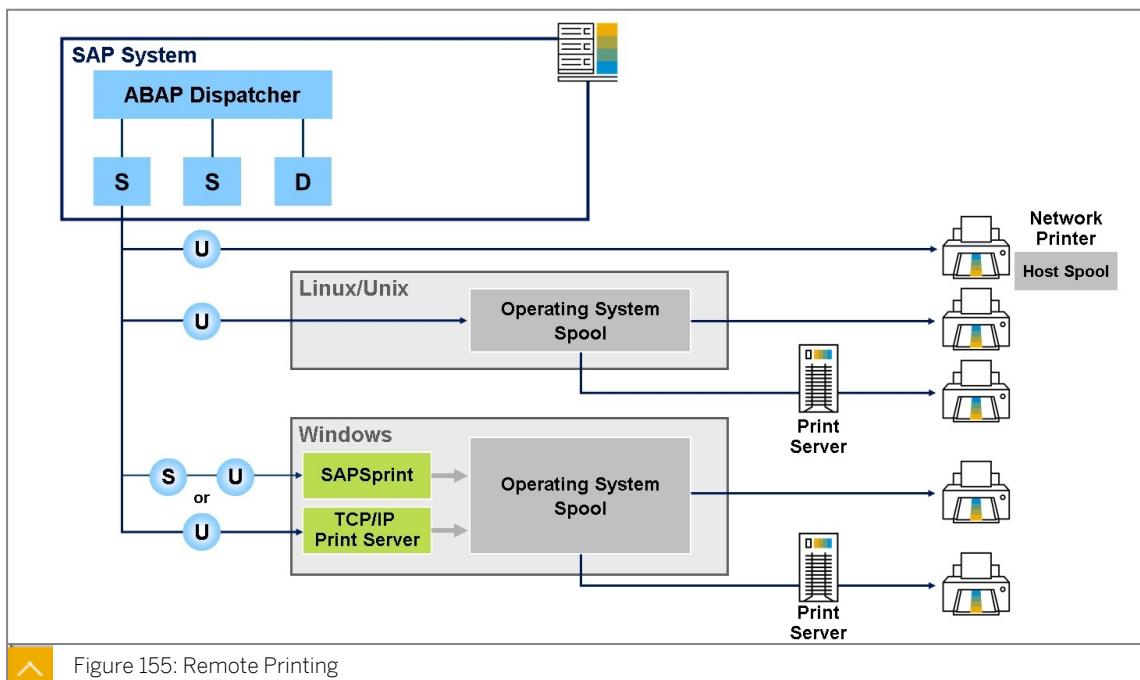


Note:

You can configure multiple spool work processes for an SAP instance. Regardless of the access method, the output sequence is affected, and different spool requests may print in a different order from the one in which they were created. If you require output in sequence, you can specify this for individual printers. However, a setting of this type reduces the ability to process printouts in parallel. SAP Note [108799](#) – How many spool work processes for each instance? contains further information about this topic.

Remote Printing

With remote printing, the spool work process runs on a different host than the OS spooler. The SAP spool work process sends the formatted data stream to the output device over a network connection. In exactly the same way as with local printing, it is irrelevant from the SAP system's point of view whether the printer is directly connected to the remote host or is reached using a network connection.



As shown in the figure *Remote Printing*, the following are typical scenarios for remote printing:

- Network printers provide their own OS spoolers, and are directly connected to a computer network. Printers of this type are directly addressed from the SAP system using their names via access method **U** (UNIX Berkeley protocol BSD).

In this scenario, you should consider the capacity of the network printer and the size of the output requests.

- Access method **U** is also used if the remote host is a Linux/Unix system – the data is forwarded to the *Line Printer Daemon* (lpd). SAP Note [39405](#) – *Print on UNIX LPD: Malformed Address* describes how access method U can be used for the various Linux/Unix versions.
- If the remote host is a Microsoft Windows system, access method **S** is usually used (SAP protocol with SAPSSprint), but access method **U** is also supported when you want to use the TCP/IP Print Server of Microsoft Windows (with or without SAPSSprint).

For all hosts running on Microsoft Windows operating systems SAP provides the SAPSSprint program. SAPSSprint is a Windows service that is capable of multi-threading. Each output request is processed in a separate, isolated thread. The output requests that SAPSSprint receives from the SAP system can thus be transferred to a particular printer individually. If the printer is not working, this does not disturb the printing of other output requests on the other printers.

SAPSSprint is available on the SAP Support Portal under the quick link /swdc (<https://support.sap.com/swdc>). Here choose *Access Downloads* (for *Support Packages & Patches* in area *Types of Software*). From there, choose path S → SAPSSPRINT .



Hint:

SAPSSprint 7.60 contains components for printing PDF files. These are based on .NET and are an integral part of the SAPSSprint installation. It is no longer necessary to install SAPPDFPRINT as a separate component, see SAP Note [277220](#) – *PDFPRINT 750 and 760 - current information*.



Hint:

For performance reasons, you should only use remote printing in a LAN environment (not WAN), and ensure that the operating system spoolers are available.

Front-End Printing

Normally, all printers on which you want to print from the SAP system must be created in the SAP system as output devices. Front-end printing is available to reduce the significant administration effort associated with this. This means that – with the help of front-end printing – a user logged on to the SAP system through the SAP GUI can use the printers that are set up at his or her front end PC.

Unlike remote printing, you do not require any fixed IP addresses for front-end printing.

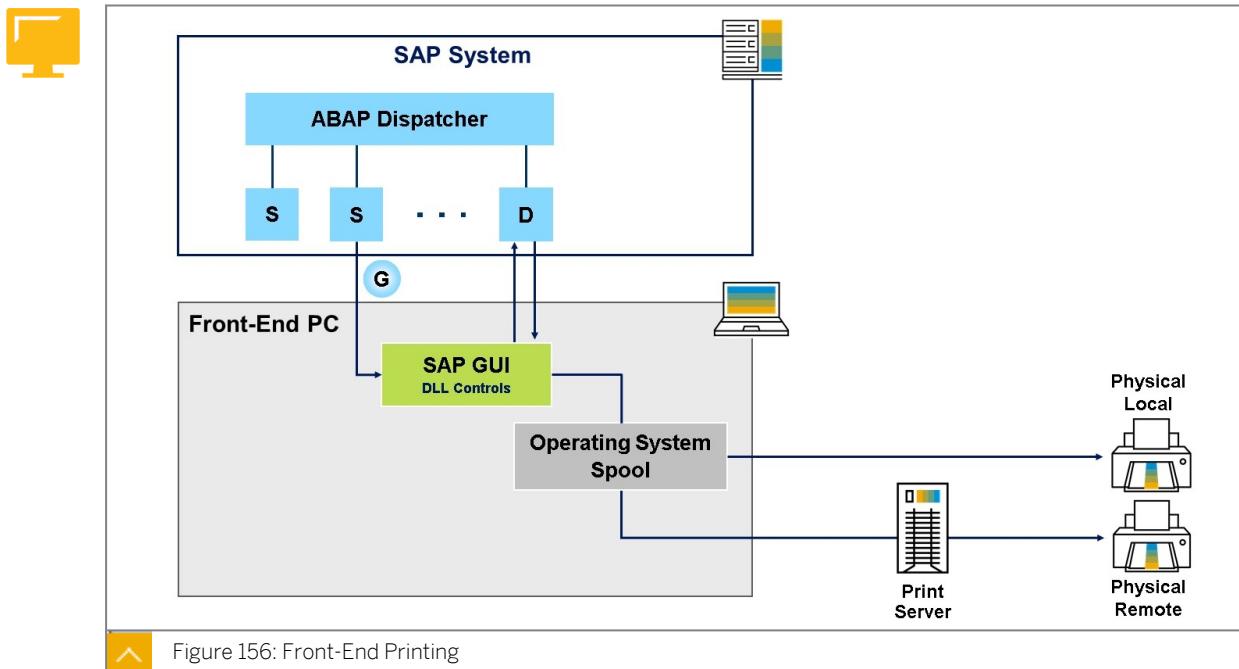


Figure 156: Front-End Printing

As shown in the figure *Front-End Printing*, you can use front-end printing with control technology and access method **G**. Controls are DLLs that run in the process context of the SAP GUI. The print control receives the print data and transfers it to the host spool system (OS spooler).



Note:

Access method **G** allows you to set up front-end printing using SAP GUI for Java irrespective of the platform. Printing using the Windows Terminal Server is also made easier with front-end printing using control technology.



Table 3: Output Devices for Front-End Printing

	SAP GUI for Windows	SAP GUI for Java	SAP GUI for HTML
Device type	SWIN	<Select a suitable entry>	PDF1 or PDFUC
Host spool access method	G	G	G
Host printer	DEFAULT	<irrelevant>	<irrelevant>

As the table *Output Devices for Front-End Printing* shows, when using front-end printing with control technology and access method **G** on SAP GUI for Windows, you define a generic host printer in the SAP system by assigning it to the physical device **DEFAULT** (starting with two underscores). Since the models used as front-end printers can vary considerably, the device type **SWIN** is often assigned for a Microsoft Windows front end, but you can also enter a specific device type.

When printing with SAP GUI for Java on other operating systems, use a corresponding device type, such as **POST2** for PostScript, or **PDF1** or **PDFUC** for PDFs.

If the front-end printing is used on SAP GUI for HTML, you can use the device types *PDF1* and *PDFUC*. The print data is then transferred to the front-end browser as a PDF document and can be printed locally.



Hint:

Front-end printing is suitable for output on local printers, but unsuitable for production or mass printing. Front-end printing is also unsuitable for output lists in background processing because it requires a connection to the front-end PC.

The processing of front-end printing is performed in the same way as with the other printing methods that use a spool work process. There can, therefore, be conflicts between regular and front-end print requests. With the profile parameter *rdisp/wp_no_spo_Fro_max*, you can define a maximum number of spool work processes for front-end printing for each SAP instance (the default value is 1).

We recommend that you configure at least one spool work process on each application server that users log on to. If no spool work process is running on the application server that a user wants for front-end printing, the system selects the available spool server that has the least load to process the print request. To override this behavior, you can define a spool server specifically for front-end printing by setting the profile parameter *rspo/local_print/server* to the required server name.



Hint:

If you expect a significant workload due to front-end printing, you should configure at least one additional spool work process for each front-end printing spool server for other tasks.



Note:

In addition to the general print authorizations, the following authorizations are required for front-end printing:

- authorization object *S_SPO_DEV*, authorization *SPODEVICE = <SAP name for the output device for front-end printing>* and
- authorization object *S_RFC*, authorization *RFC_TYPE = FUGR, RFC_NAME = LPRF* (function group for local printing RFC), *ACTVT = 16* (Execute).

**Note:**

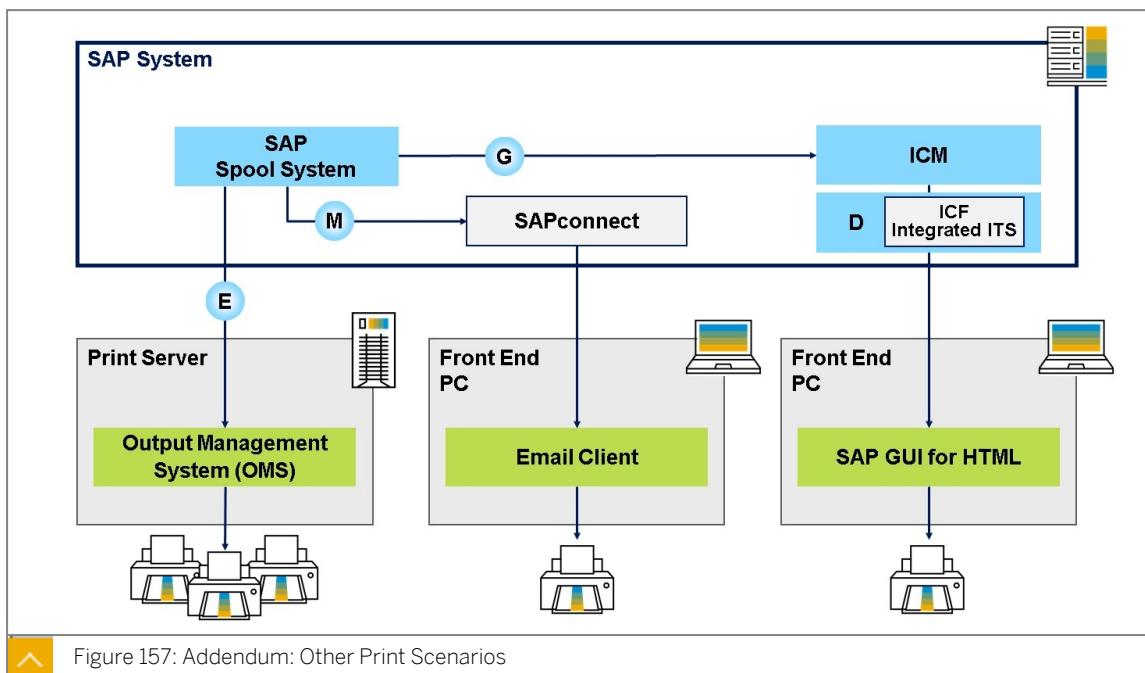
For more information as well as some restrictions on front-end printing such as:

- you cannot perform front-end printing in background processing, as there is no connection to the front end,
- front-end printing does not support printing from BSP, Web Dynpro or SAP Fiori applications,
- a free session is required for front-end printing, so it must not be the case that all sessions are occupied.

see SAP Notes [128105 – Front-end printing \(composite SAP Note\)](#) and [616762 – Front-end printing from BSP, Web Dynpro, or SAP Fiori applications](#).

Addendum: Other Print Scenarios

The figure Addendum: Other Print Scenarios provides an overview of other print scenarios.



External Output Management Systems (OMS)

External Output Management Systems (OMS) can be addressed from the SAP system using a defined interface (BC-XOM) (access method **E**). This access method is of particular interest if you have a very large output volume, or the output of multiple systems (SAP systems and others) is to be centrally controlled and monitored.

Email

You can send print output by e-mail to a user (access method **M**). The e-mail is sent using SAPconnect.

All device types can be used to print the e-mail, however, you must ensure that the relevant display program or printer is available for the device type. The use of a device type that generates PDF files is recommended. This PDF file is then sent as an e-mail

attachment. You can then display this with a PDF display program, such as Adobe Acrobat Reader, and print it as required.

For more information, see SAP Notes [311037 – Printing using e-mail](#) and [513352 – Printing by e-mail \(update\)](#).

SAP GUI for HTML

A user can also be connected to an SAP system from a web browser using SAP GUI for HTML. This type of connection can be set up in the SAP system with the integrated ITS.

What options does this user have for print output? One option is the previously described output by email. In contrast, Web printing does not need an email client. To use Web printing, the system administration must provide an output device with device type **PDF1** or **PDFUC** and access method **G**. When a user selects this output device, a PDF document is displayed in the Web browser window, which the user can then print locally.

For more information, see SAP Notes [771683 – Front-end printing for SAP GUI for HTML \(as of Release 6.40\)](#) that lists the exact requirements and provides additional details about web printing and [2489982 – Font options for PDF preview on SAP GUI for HTML](#).

Create Output Devices

The SAP term for printers is **output device**. This is because SAP may not only print to printers but also to external Output Management Systems, archiving systems, and so on.

When creating output devices in transaction SPAD, to some extent, you need to enter different data. For details, please take a look at the table *Creating Output Devices*.



Table 4: Creating Output Devices

		Local	Remote	Front end (SAP GUI for Windows)
Device Attributes	Device Type	<Select the appropriate type in each case>		SWIN
	Spool Server			–
Access Method	Host Spool Access Method	L (on Linux/Unix) resp. C (on Windows)	S or U	G
	Host Printer	<Printer name at OS level>		— DEFAULT
	Host Name	<Determined by entry for Spool Server, fix>	–	–
	Destination Host	–	<Specify>	–



Hint:

Note that a printer can never be renamed, and that the name of the printer in the SAP system is case-sensitive.

The configuration of the SAP spool system is a system administration task. You must create a device definition in the SAP system for every output device on which you want to print a document, so that the SAP system recognizes this device. This is done in transaction SPAD.

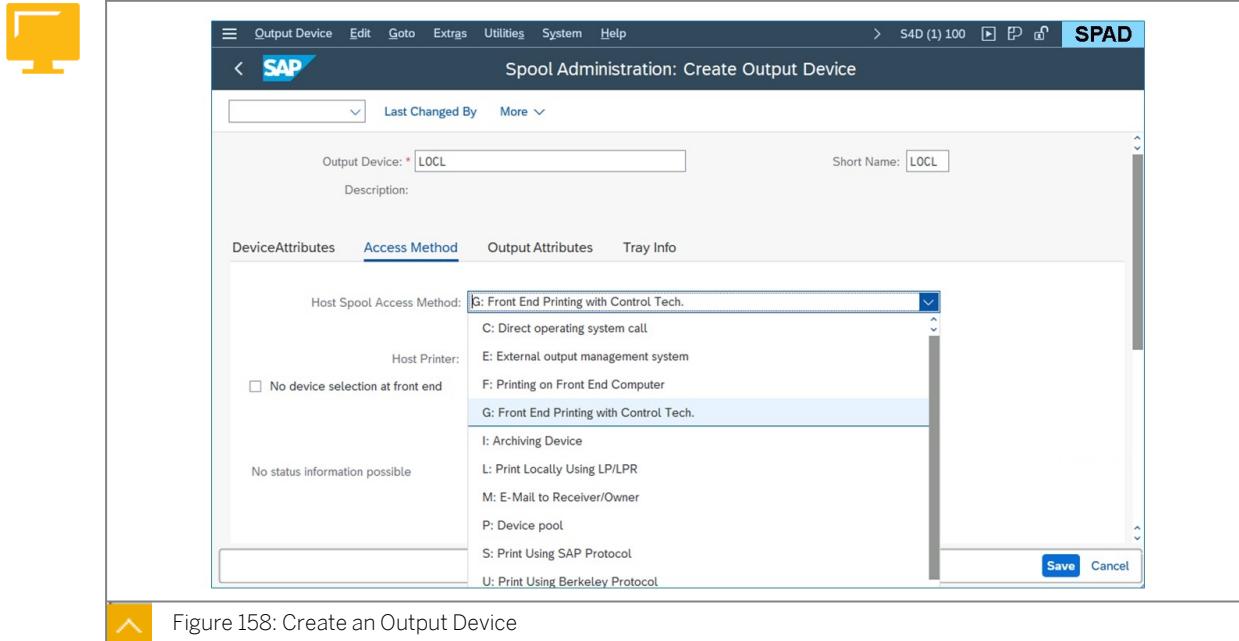


Figure 158: Create an Output Device

The device definition contains all of the important data for addressing the device. When creating a new output device, you need to provide the following information.

Output Device

The name of the printer in the SAP system. It can be up to 30 characters long, and is case-sensitive. Users of the SAP system use this name to select an output device.

Short Name

This is used for internal system purposes and can be automatically generated.

Device Type

The device type indicates the type of printer to be addressed to format the device-specific output. When you define an output device, choose the name of the device type that was defined in the SAP System for your printer model, (such as *Post2* for a *PostScript* printer).

Note:

The device type **SWIN** transfers the spool data from the SAP system to the Microsoft Windows printer driver. This is useful, for example, when different printers are used for front-end printing in a Microsoft Windows environment.

Spool Server

An SAP application server with spool work processes or a logical server that is to prepare the output for this device.

**Note:**

SAP Note [3129597](#) – Using `_AUTOSPOOLSERVER` as spool server explains the idea of the assignment option of the spool server `_AUTOSPOOLSERVER`.

Authorization Group

You can enter the name of an authorization group instead of an output device in a user authorization (field `SPODEVICE` for authorization objects `S_SPO_DEV` and `S_SPO_PAGE`). The user is then authorized to use all devices in this authorization group.

Location

The location of the printer, for example, building and room number (so that users can find their output).

Message

Additional information that, if entered, temporarily overrides information from the `Location` field (for example, “Currently in maintenance.”).

Lock Printer in SAP system

Output requests for printers for which this indicator is selected are created but will not be transferred to the printer. The user receives the message . . . **no immediate printing.**

Host Spool Access Method

Specifies how the spool work process transfers data from SAP system to the host spool system (OS spooler) for a specific printer.

Host Printer

The name of the printer at the operating system level (in the host system).

**Note:**

The name of the host printer is case-sensitive. Under Microsoft Windows, there should be no blank in the host printer name, and network printers are addressed using their UNC names (in the format `\\\R12345\P42`). The entry `_DEFAULT` (starting with two underscores) controls the Windows default printer (for front-end printing using SAP GUI for Windows).

Host Name

This field is used only for local printing. It is filled automatically when you specify the spool server.

Destination Host

This field is used only for remote printing and is the name of the host on which the OS spooler is running and which receives the data for output.

Do not query host spooler or printer for print requests

By default, the spool work process keeps track of print requests sent to the host spool system to be able to update the status of the requests in the SAP system. To do this, the host spooler must be queried periodically.

If a host spool system is slow or there are problems querying network or remote printers, the query may take a long time. This may affect printing performance as the spool work

process cannot process any other print requests while waiting for a response from the query. In cases like this, you can deactivate querying using this switch.



LESSON SUMMARY

You should now be able to:

- Classify Printing Methods
- Create Output Devices

Unit 8

Lesson 3

Working with Device Types



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Select a suitable device type
- Illustrate the idea of Unicode Printing Enhancement

Device Types for SAP Output Devices

A device type describes the type of the printer to be addressed. When you define an output device, choose the name of the device type that was defined in the SAP system for your printer model. The SAP system then uses the device type to format the output device-specific print output.

When the spool work process generates an output request, it formats the data according to the specifications of the device type.

The device type is separate from a printer driver. The device type is the set of all properties of an output device that the SAP system has to know to control the output device correctly, such as control commands for font selection, page size, character set selection, and so on.

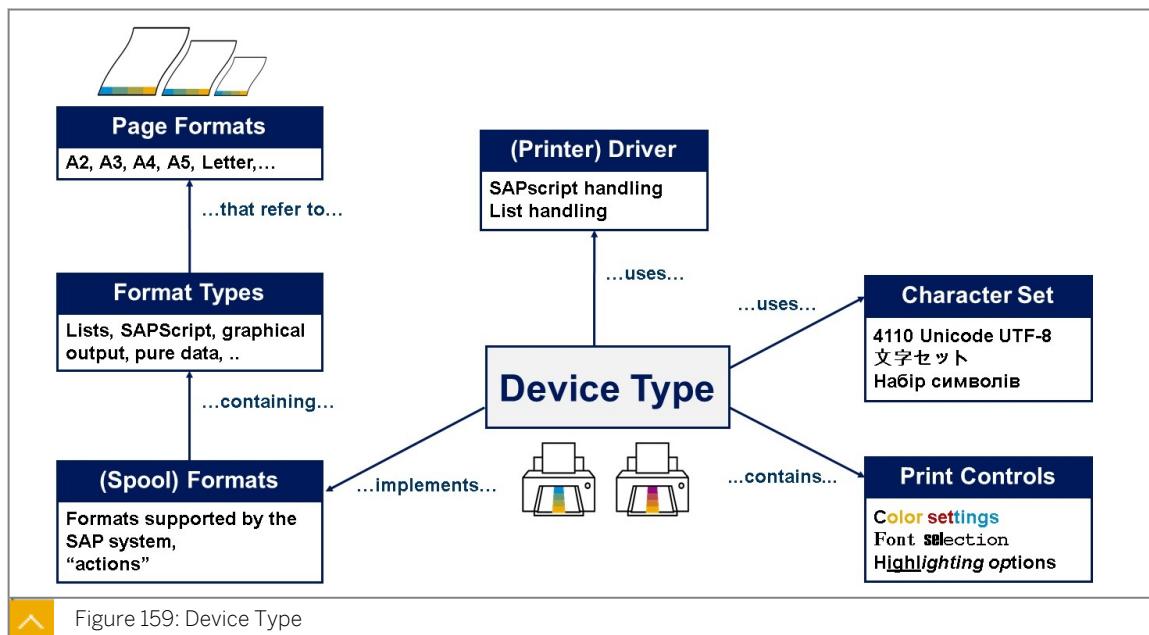


Figure 159: Device Type

The figure “Device Type” illustrates the structure of a device type. The following terminology is useful in understanding device types:

Character Set

A character set contains the characters that can be output by a particular output device. To be able to use a particular character set for a selected device type in the SAP system, it must contain this character set.

Print Control

Print controls are general print control statements that can be embedded into texts as formatting commands. This allows you to control particular display options of output devices, such as boldface, changing the font and font size, and so on. They are converted into printer-specific escape sequences for the output.

(Spool) Format

Spool formats specify the format supported by the SAP system. The system differentiates between SAPscript formats (such as *DINA4* and *LETTER*) and ABAP list formats (such as *X_65_132* meaning 65 rows / 132 columns).

 Note:

Actions are output device-specific commands that are required for the implementation of a format. The action *printer initialization*, for example, can contain a printer command with which the number of rows on a page is defined. There is a set of actions for every format supported by a device type.

Format Type

A format type describes how output should appear on paper. It primarily contains the formatting of the page format. Among others, there exist a format type for lists (spool formats with this format type need to have the guaranteed number of lines and columns on the output defined as an additional attribute) or SAPScript (spool formats with this format type need to have a page format defined as an additional attribute).

Page Format

A page format describes the format of a printable page in the SAP system. A large number of standard page formats are predefined in the system. For a device to support additional formats that are not predefined, you can define new formats. Consider when doing so that your output device must, of course, be able to use the new format.

Printer Driver

There are two different printer drivers assigned to a device type: an SAPscript printer driver (which is only used by SAPscript) and an printer driver for printing ABAP lists with this device type.

 Note:

SAPScript is a tool for text management and form printing.

For more information about device types, check the online documentation on SAP Help Portal for SAP S/4HANA (*Product Assistance*), area *Enterprise Technology* → *ABAP Platform* → *Administrating the ABAP Platform* → *Administration Concepts and Tools* → *Solution Life Cycle Management* → *SAP Printing Guide (BC-CCM-PRN)* → *Device Types for SAP Output Devices*.

Selecting an Appropriate Device Type

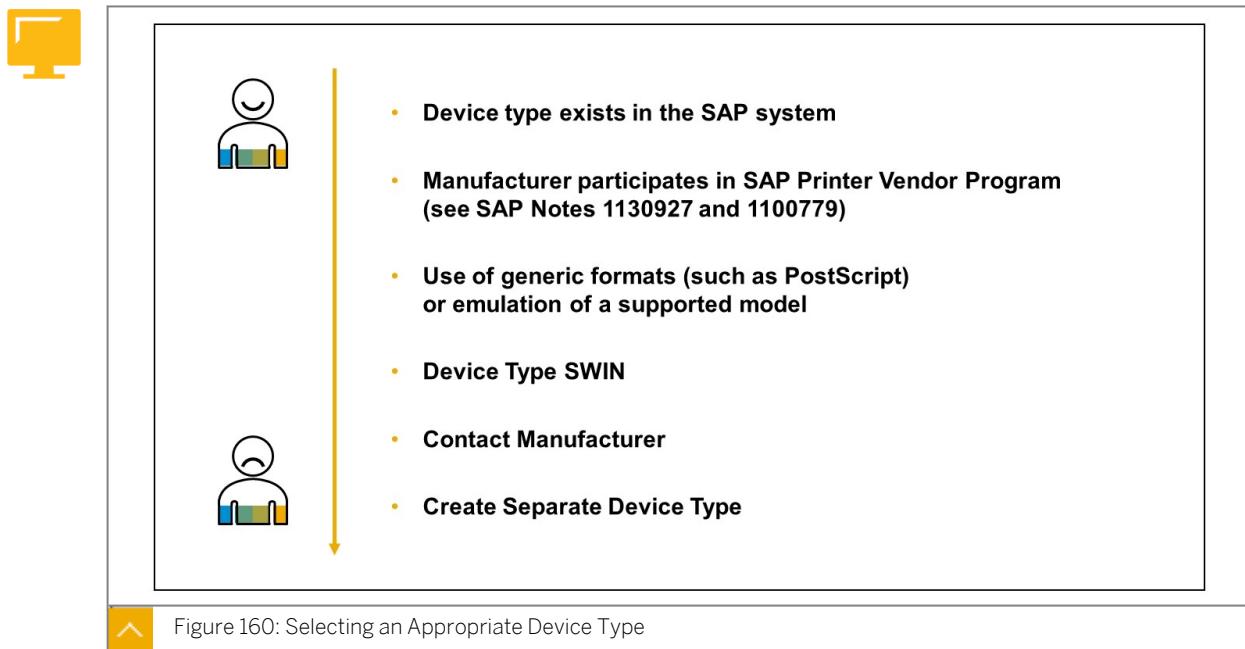


Figure 160: Selecting an Appropriate Device Type

- In the ideal situation, a device type for your printer model exists in your SAP system.
- Note:**
A separate device type is not required for every printer model of a vendor, but just for each model family.
- Several printer manufacturers participate in the *SAP Printer Vendor* program. If there is no device type for a specific printer model in your SAP system, you can easily import device types for the supported printer models provided by those manufacturers using the *device type selection wizard* (available in transaction *SPAD* for output devices).

Note:
For more information about *SAP Printer Vendor Program*, see SAP Note [1130927](#) – *SAP Printer Vendor Program* which (among others) contains a link to the *SAP Integration and Certification Center - SAP Printer Vendor Program (PVP)* document.

- If the device type does not exist in the SAP system, many printer models can be addressed by a generic format, such as PostScript. It is also possible to address a printer model in the same way as another model that is supported by SAP (compatibility mode).
- There are drivers for the Microsoft Windows operating systems for practically all printers on the market. You can use these printers as output devices with the generic device type **SWIN**. This device type allows you to use any printer or fax that the Microsoft Windows Print Manager can use, regardless of whether the SAP system has a device type for the corresponding device. In this case, it is actually the Microsoft Windows Print Manager and Windows driver that format a document and make an output-ready data stream out of it.

- If there is no device type for your printer model and none of the options above can be used, you can create your own device types. The SAP system contains all of the tools required to create your own device types or to adjust copies of existing device types.

**Caution:**

To do this, detailed knowledge of the SAP spool system and the printer controls is required. You should weigh up the expense against the purchase of supported printers.

Wizard for Device Type Selection

In most cases, the SAP System already provides the appropriate device type for the printer type of the printer model that you want to use. These standard device types are completely defined and need no modification or extension before you use them in device definitions.

When you create a new output device or change an existing one, it is sometimes hard to choose the appropriate device type. To help you with the selection, you can use the *Device Type Selection* (wizard) in transaction SPAD. The wizard leads you step-by-step through the selection of the most suitable device types for your output devices.

The wizard requires a set of information such as the name of the printer manufacturer and the language in which the documents are to be printed. Since new printers are being introduced day by day, new printer models and the corresponding device types must be 'introduced' to the *Device Type Selection* first to become selectable. This is why the selection data needs to be updated on a regular basis.

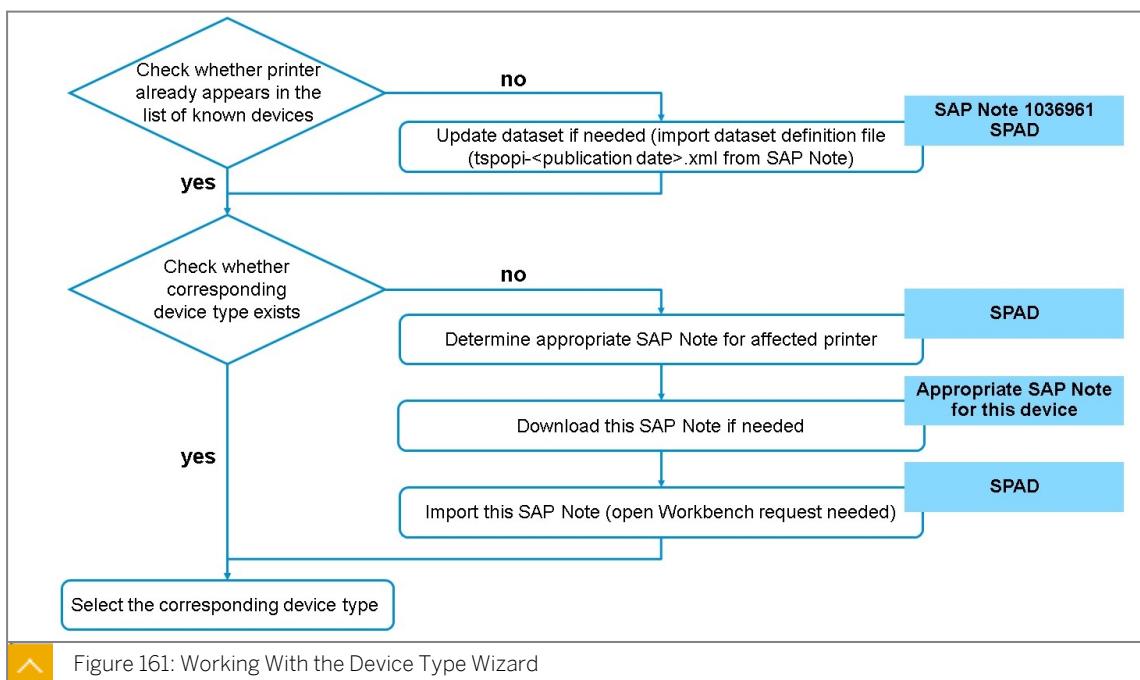


Figure 161: Working With the Device Type Wizard

The figure “Working With the Device Type Wizard” lists the major steps to provide your SAP system with new device types for printer models by manufacturers that participate in the SAP Printer Vendor program.

To start the device type wizard, start transaction SPAD and navigate to the list of *Output Devices*. Double-click an existing output device or create a new one. On the *Device Attributes* tab, choose *Device Type Selection* (magic wand icon) next to the field *Device Type*.

1. On the selection screen, choose the manufacturer and check if your printer model shows up in the list of known printer models.
2. If necessary, update the printer information by downloading the attachment `tspopi-<time_stamp>.zip` from SAP Note [1036961](#) – *Device type selection wizard in transaction SPAD*.
3. Unzip the downloaded attachment. You will get a file `tspopi-<date_stamp>.xml` that contains the current set of information about printer manufacturers and printer model/device type combinations. To import the XML file, choose *Update Printer Information* in the entry screen of the *Device Type Selection Wizard* and select the XML file. As a result, the entries for the manufacturer and the printer models should have been updated.



Note:

The information about what printer manufacturers and printer models/device types are available is stored in two database tables. If you use the wizard for the first time in a system, you need to fill them once and update them later on a regular basis. This is done with the upload of the XML file to the system.

If the database tables are already filled, the date of the last update of the *dataset version* is displayed and you should be able to select the manufacturer and the required printer model.

4. Now you can select the required printer manufacturer and printer model and choose *Continue with Language Selection*. After you have selected the language, you can display the available device types (button *Show Available Device Types*).
5. You should now be able to assign the selected device type to an output device (button *Assign Device Type to Output Device*).
6. If the warning *Device Type is not yet installed!* is displayed, the selected device type does not yet exist in the system. It must be installed before you can assign it to the output device. To do so, you need to download the Reference SAP Note that is displayed.

To download the Reference SAP Note, double-click the note number that is displayed in the *Details* section. You will be forwarded to the SAP Note. From there, download the attached device type.

7. Back in the *Device Type Selection*, choose *Install Device Types* (truck icon) in the application toolbar and follow the instructions in the appeared window.

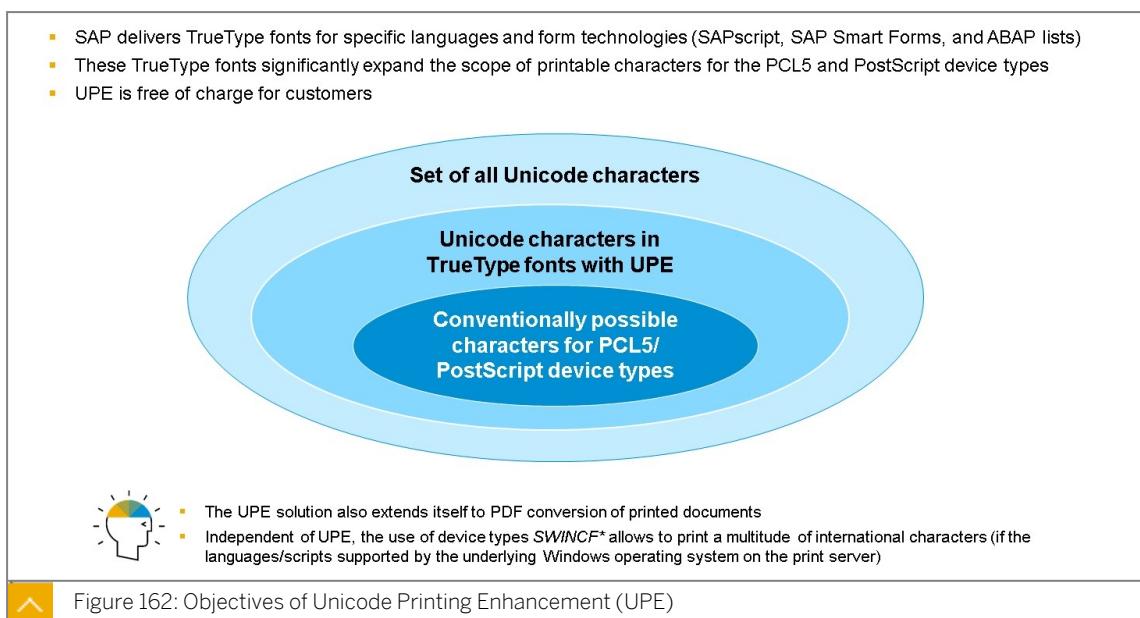
You can now either create new output devices that match the new device type or change the existing output device definitions.

The Idea of Unicode Printing Enhancement

Typical printers are printers driven with the printer description languages PCL5 or PostScript and which are connected directly to the SAP backend, that is, the SAP application server running ABAP programs. Printing German, English or French documents was generally possible without any issues, but as soon as the customer ventured into more exotic languages respective scripts, for instance Greek, Cyrillic or even Chinese, the printout would contain a lot of hash characters (###) instead of the expected foreign characters. Since a SAP Unicode system (in theory) allows you to enter, store and display arbitrary Unicode characters, this limitation to the business forms processing seemed unacceptable to many customers.

One way to overcome this issue is to connect printers to the back end via a Windows print server running the SAPscript software and employing device types **SWINCF*** when defining the printer in transaction **SPAD**. In this case, the actual text rendering and PDL generation is done on the Windows print server and not in the SAP back-end system and thus, a multitude of international characters (in effect those languages/scripts supported by the underlying Windows operating system on the print server) can be printed from the SAP system.

In 2012, SAP launched an initiative aimed at solving the typical problems associated with printing documents with Unicode characters: **Unicode Printing Enhancement (UPE)** (see the following figure). Due to UPE SAP customers can enhance the bandwidth of possible printable characters for output devices by providing a solution for PCL5 and PostScript printing from SAPscript/Smart Forms that is comparable to the **SWINCF*** solution for Windows printing. In addition, the existing PDF converter should be enhanced so it would preserve all “exotic” characters from the print documents in the generated PDF file. As a last requirement, printing ABAP lists via “UPE-enabled” output devices should also support the output of foreign characters, similar to SAPscript/Smart Forms.



As listed in the figure *Objectives of Unicode Printing Enhancement (UPE)*, the characteristics of UPE include:

- Support for PCL5 and PostScript-enabled printers, including PDF conversion
- Support for SAP Smart Forms, SAPscript, and ABAP lists
- TrueType fonts licensed from SAP
- UPE is free of charge for customers

The UPE solution helps to overcome this general limitation with most PCL5/PS device types:

- The UPE solution adds a set of TrueType fonts to the SAP system, which can be used by the PCL5 and PostScript printer drivers to download additional Unicode characters required by a document to the printer.
- In order to make additional characters (from the TrueType Fonts) usable for SAPscript/Smart Forms and ABAP list printing, the UPE solution adds so-called *Unicode Reference Device Type* (URDT) to the system.

- URDTs basically serve as font and character mapping tables during the formatting/print process: The data stored in a URDT reveals to the SAPscript/Smart Forms formatter that a requested Unicode character is available in a TrueType font and which TrueType font contains the character
- The URDTs also contain the character width of the Unicode characters printable via TrueType fonts, so that the SAPscript and Smart Forms formatters can correctly format the document

The UPE solution also extends itself to PDF conversion of printed documents, for example, for the purpose of optical archiving via SAP Archive Link. This means that when you archive a SAPscript or Smart Forms print form which prints via an UPE-enhanced printer, the resulting PDF file is automatically also UPE-enhanced, meaning it will contain any additional Unicode characters which would be downloaded from TrueType fonts during the print process. In the PDF data format this is accomplished via embedding subsetted portions of these TrueType fonts which contain only those characters needed in the document.

Prerequisites

A primary requirement for using UPE is an AS ABAP-based SAP system with Unicode.



Note:

To identify if your system is a Unicode system, choose menu path **System → Status**.

For more information, see SAP Note [1812076 – Unicode Printing Enhancement \(UPE\)](#) including its attached PDF.

Activate Unicode Printing Enhancement

In AS ABAP-based systems that meet the described prerequisites, you activate UPE for an output device by assigning a **Unicode Reference Device Type** (URDT) to this device. A URDT contains fonts that are required to print a certain group of languages (such as Latin-2 for Eastern European languages) or an alphabet (such as Cyrillic).



The **Unicode Print Settings** button appears if the system prerequisites are met ...

... for output devices with device type PCL5 or PostScript

Flavor

Supported languages

Figure 163: Assigning Unicode Reference Device Types (URDT) to Output Devices

As shown in the figure “Assigning Unicode Reference Device Types (URDT) to Output Devices”, the *Unicode Print Settings* button is displayed in transaction SPAD for output devices that use a PCL5 or a PostScript driver. You use this function to assign the URDT.

When you assign a URDT to an output device, you define which additional languages and fonts can be printed on that output device in addition to the languages supported by the device type. When combined with UPE, this improves both the scope of possible printable characters and performance because the datastream that is sent to the printer only contains characters that are not in the printer's own fonts.

SAP provides the following URDTs (see the figure above):

- URDTC for Simplified Chinese
- URDTJ for Japanese
- URDTK for Korean
- URDTT for Traditional Chinese



Note:

You can only assign one URDT to an output device. The same Unicode character can have different appearances in the above languages. If you expect to output documents in one of the four languages, select the corresponding URDT. If you expect to output documents in several of these four languages, create several output devices (each with the “matching” URDT). If you do not expect to output documents in any of the four languages, select any URDT.

Additional Information

The SAP Community blog *UPE – Unicode Printing Enabling* (<https://blogs.sap.com/2013/07/09/upe-unicode-printing-enabling/>) might be helpful for a first overview.

In addition, the following reports are relevant in the UPE context:

- *RSPO_CHANGE_PRINTERS* permits mass changes to output devices (including UPE activation)
- *RSPOUPE_URDT_UTIL* for the detailed analysis of the fonts contained in a URDT
- *RSPOUPE_URDT_TOOL* enables customers to modify URDTs



LESSON SUMMARY

You should now be able to:

- Select a suitable device type
- Illustrate the idea of Unicode Printing Enhancement

Working with Spool Servers



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the use of logical spool servers
- Create a logical spool server
- Summarize the classification of spool servers and output devices

The Concept of Logical Spool Servers

In this lesson, you will learn about the concept of logical spool servers (or – for short – *logical servers*) and how to set them up.

In the SAP printing concept, an output device is assigned to just one spool server. A spool server, on the other hand, can have multiple output devices assigned to it, which raises the risk of this server becoming overloaded. You should, therefore, have a mechanism for load balancing across multiple servers. Having groups of spool servers is also recommended for reliability reasons. Both of these aspects are taken into account in the implementation of logical spool servers, also called logical servers.

A logical server stands for one or more other logical and/or real SAP spool servers. Logical servers can be used in place of real SAP spool servers anywhere in the spool system. The logical server is a *placeholder* for the spool server assigned to it, while a real server is an actual application server with a spool work process and can output print requests.



Hint:

Include logical servers from the outset when planning your printer landscape. When scaling your SAP system later (additional instances with spool work processes), logical servers make it easier to adapt the printer landscape.

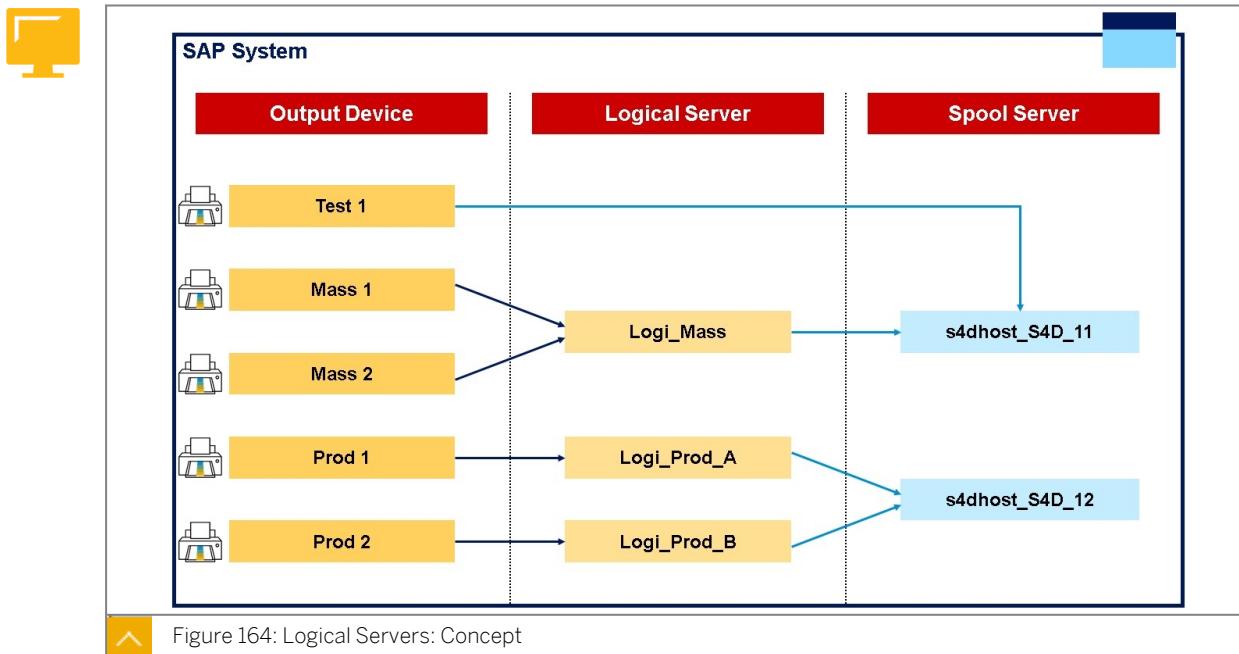


Figure 164: Logical Servers: Concept

The figure *Logical Servers: Concept* illustrates a basic scenario involving logical servers. A (real) spool server is an SAP application server with at least one spool work process. Every output request is processed on a real spool server of this type. An output device created in the SAP system can be assigned to a real spool server directly. However, there are many advantages associated with an additional logical (spool) servers for this purpose. These stand for a hierarchy of other logical servers and/or real spool servers.



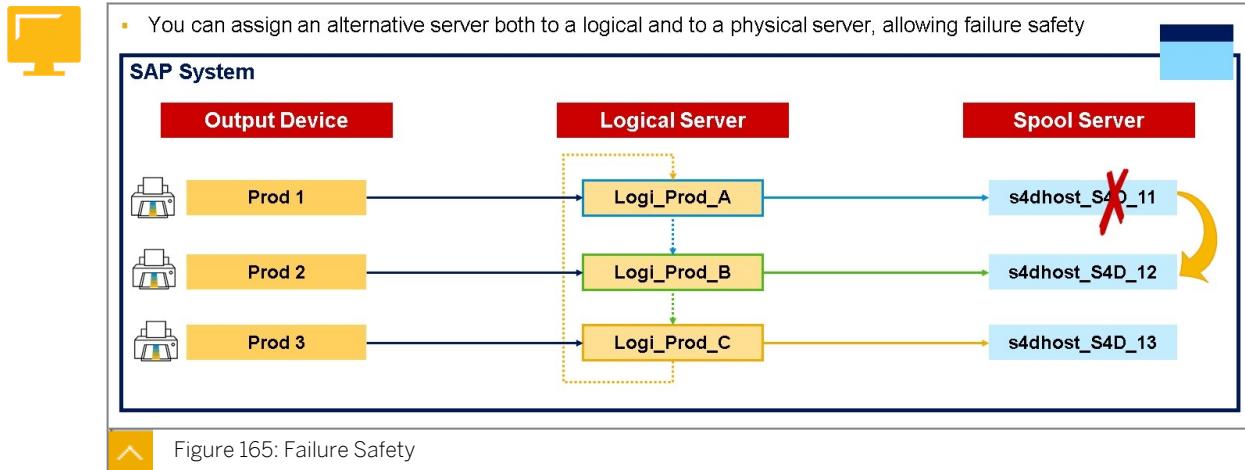
Hint:

Another idea might be to group the SAP output devices by defining different logical spool servers for devices connected over a network and for local devices. Both logical spool servers could point to the same real SAP spool server. However, you can manipulate the devices separately for such purposes as switching devices to another spool server.

Features of Logical Spool Servers

Failure Safety

When you create a spool server (both logical and real spool servers), you can specify an alternative server. If the normal server is not available, the SAP system attempts to use this alternative.



You cannot define more than two spool servers for a logical server. However, because a logical server can reference other logical servers, you can also create extensive spool server hierarchies.

In the figure “Failure Safety” above, output device *Prod 1* is assigned to logical spool server *Logi_Prod_A* which in turn is mapped to spool server *s4dhost_S4D_11* and has logical spool server *Logi_Prod_B* as alternative server. In the same way *Logi_Prod_B* is mapped to *s4dhost_S4D_12* and has alternative server *Logi_Prod_C*. *Logi_Prod_C* in turn is mapped to *s4dhost_S4D_12* and has alternative server *Logi_Prod_A* (closing the loop). If now, for a document coming from *Prod 1*, *s4dhost_S4D_11* is not available, the spool system will redirect the output request to the (available) spool server *s4dhost_S4D_12*.



Note:

If this spool server was not available as well, it would be redirected to *s4dhost_S4D_13*.

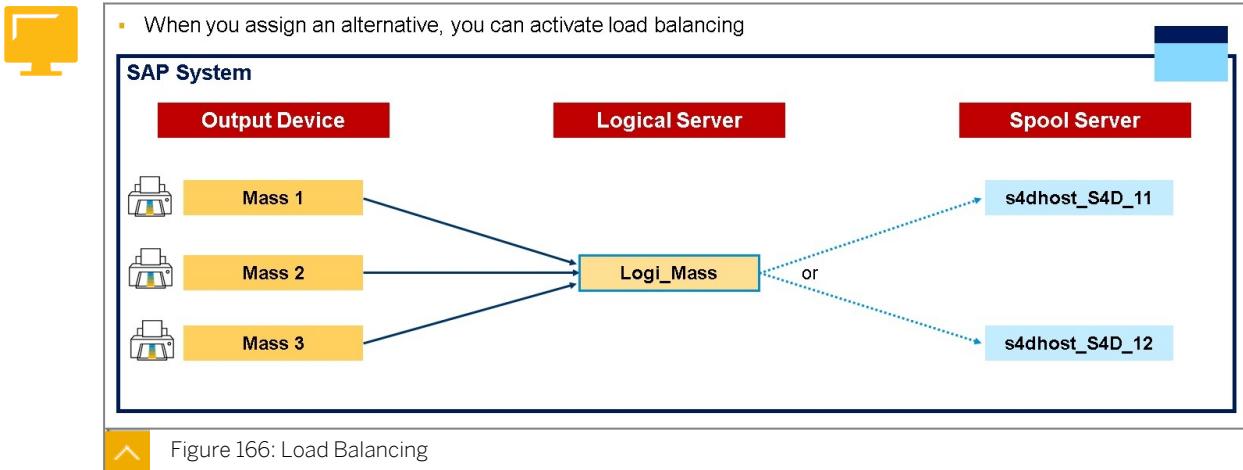


Caution:

You must ensure that all output devices that may be used by a different spool server can be controlled in the same way by every spool server. For example, in the figure “Failure Safety”, if the output device *Prod 1* points at operating system level to a printer *Printer1* that is controlled locally, an operating system printer *Printer1* must be available on server *s4dhost* and be accessible as a local printer from all possible application servers (that is both from *s4dhost_S4D_11* and *s4dhost_S4D_12* and *s4dhost_S4D_13*).

Load Balancing

You can allow load balancing for every spool server with an alternative server by selecting the field *Allow Load Balancing*. The load of a spool server is calculated from the number of spool work processes, output requests, and printed pages.



You can create an output request for a spool server with load balancing on either a logical or a real spool server, as the figure “Load Balancing” shows. The system determines which server has the smallest load. The algorithm is recursive: The same selection criteria are used on the mapping and the alternative server, which both could be logical servers themselves.

If load balancing is activated, then output requests for a particular device may be processed by different spool servers. Potentially, a short output request that was generated later could overtake an earlier but longer request.



Note:

The spool system does allow you to enforce print in order of generation for particular output devices. To do this, on the *Output Attributes* tab for the device definition, select *Sequentially Request Processing*. This option forces “print in order of generation” for output requests for the device, even if the definition of the assigned *Spool Server* definition allows dynamic server selection: Sequential request processing (property of an output device) has priority over the load balancing shown here (property of a spool server). This option makes it possible to use load balancing for most of your printers, but keep the “print in order of generation” for critical printers.

Setup of Logical Spool Servers

You manage spool servers in transaction SPAD, on the *Devices / servers* tab in the *Spool Servers* area. The following terminology is relevant when setting up a spool server (see figure “Creating a Logical Server”):

Server name (field AS Instance)

The name of the spool server can have a maximum of 20 characters and is case-sensitive. You can use an additional field a *short description*.

Server class

You can classify the spool server here (for high-volume printing, for example).

Logical server

Select this field when you create a logical server.

Allow load balancing

You use this field to control which method the system uses to determine a spool server for processing of output requests. When the field is set, the system searches for the server with the lowest load among the hierarchy of logical and/or alternative servers you have defined (valid both for logical and physical spool servers).

Mapping

The name of a real or logical server to which this logical server refers.

Alternative server

The server that should be used if the current server is down or not available.

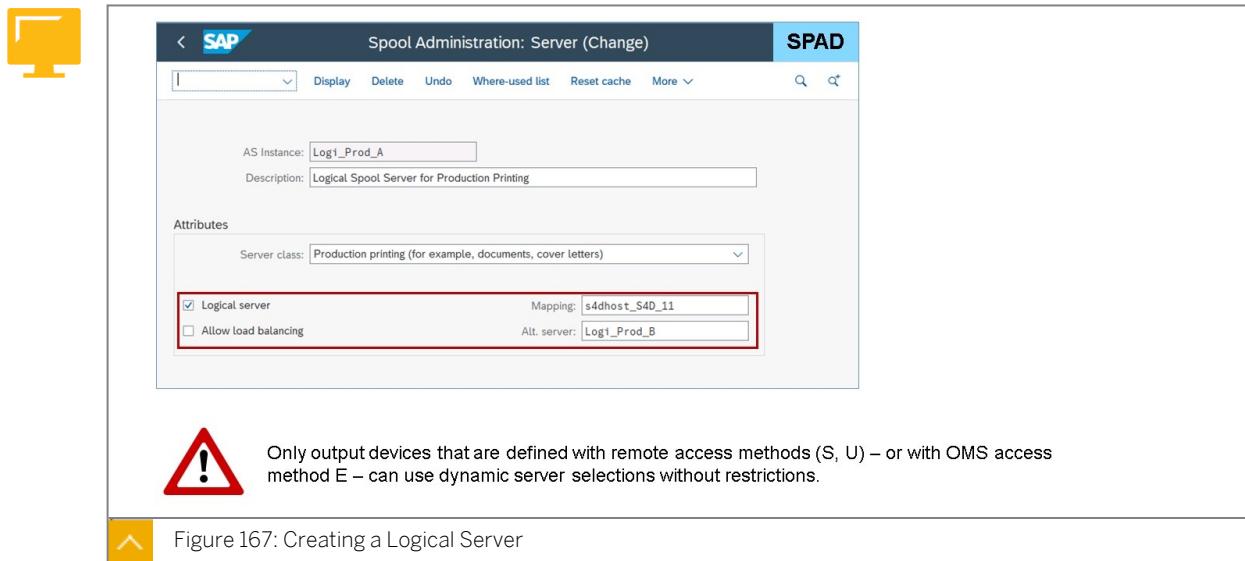


Figure 167: Creating a Logical Server

You can use dynamic server selection without restrictions only for output devices that are defined with remote access methods S and U or with the local OMS access method E. The reason for this is that the SAP system accesses remote printers using a network link. This network link is likely to be usable without adjustment by any spool server. Output requests for a remote printer can therefore be moved from one spool server to another without causing any problems. Output Management System (OMS) printers are accessed by a print command that can be executed locally from any spool server. Requests for OMS printers can therefore also be moved from spool server to spool server without problems.

For local printers (access methods C and L), on the other hand, dynamic server selection requires some preparation at the operating system level. In a local connection, the SAP spool server passes an output request directly to the spooler or print manager in the host on which it is running. Such an output request can be processed at another server only if the local spooler or print manager at the second spool server can identify the target output device. This means that the target output device must be defined under the same name in both local spoolers (in the host of the first spool server, and also in the second host). Otherwise, switching an output request for such a device to another spool server could generate the error *Device unknown* from the local spooler or print manager.

Classification of Spool Servers and Output Devices

You can classify spool servers according to their intended use, such as a spool server for mass printing or for production printing. This classification helps to make your print architecture clearer. It also helps you to realize your planned print architecture, such as assigning classified output devices to the correct spool server.

You can classify output devices and spool servers, for example, for test printing or production printing. To classify an output device, select it (in transaction SPAD under *Output Devices*) and choose the menu path *Edit → Classification*.

Classify your printers and spool servers by print type

Production Printing **Mass Printing** **Test Printing**

(Logical) Spool Server

Output Device: Edit → Classification → <Classification>

Note: Make sure that Server Class (spool server) and Classification (output device) match

The SAP system checks the classifications when saving, and displays a warning message if there are deviations: For example, if you attempt to assign a *mass printing* output device to a *production printing* spool server, the spool system compares the classification of the output device to the classification of the server. In the case that the classifications do not match, the SAP system informs you.



LESSON SUMMARY

You should now be able to:

- Explain the use of logical spool servers
- Create a logical spool server
- Summarize the classification of spool servers and output devices

Unit 8

Lesson 5

Monitoring and Troubleshooting Printing Issues



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use monitoring transactions for printing
- Perform housekeeping activities for spool
- Analyze printing errors

Monitor Spool Requests and Output Requests

In this section you will learn about managing spool and output requests.

To maintain spool and output requests, call the *Output Controller* (transaction SP01 or choose *System → Services → Output Control*).



Note:

If you only want to check the status of your own spool requests, choose *System → Own Spool Requests* (transaction SP02).



Figure 169: Selecting Spool or Output Requests

As the figure *Selecting Spool or Output Requests* shows, transaction SP01 provides many **selection criteria** for selecting particular spool or output requests. The *Further selection criteria...* function is simple to use and allows each user to define their own selection criteria.

You can select to display either spool requests or output requests (according to your selection criteria):

- The list of spool requests is a list of the spool requests that match the selection criteria that you specified in the initial screen of transaction SP01 (*Output Controller*).

Information, such as spool number, spool request type, creation date and time, creator, number of pages, and the processing status is displayed for each request. The data for a spool request is displayed in a row.

Among others, the following functions are available to edit the displayed spool requests:

- *Print Request* (not to be confused with the general print function): An output request is generated from the spool request. The print settings of the spool request are used.
- *Print with changed parameters*: Change the print settings, such as the printer name, before you print the spool request.
- Display and (to some extend also) change the *request attributes* (such as number of pages or output device).
- *Delete requests*.
- *Display the contents of a request*.
- *Display output requests* for spool requests.
- Forward the contents of a request (send using SAPOffice or export as text or PDF) – using the menu path *Spool Request → Forward*.
- The list of output requests is a list of output requests that were selected using certain criteria. Every time that the system prints a spool request, that is, that it sends an output request, a new entry for this output request is added to the list of output requests.

The system displays information about the individual requests, such as creation time, output device, format, and processing status.

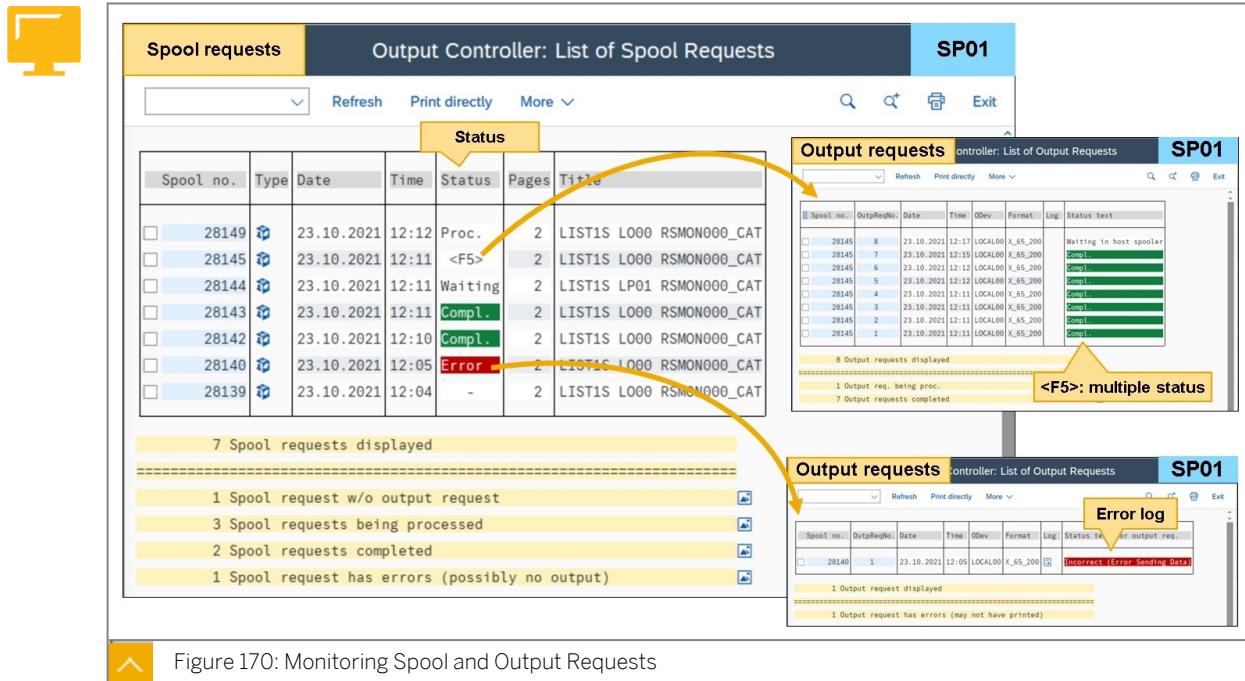


Figure 170: Monitoring Spool and Output Requests

As an example, the list displayed in the figure *Monitoring Spool and Output Requests* shows all spool requests that match your selection criteria. The list is created using the SAP List Viewer (ALV). This allows you to change the appearance of the list as you desire. This means that you can show and hide columns, sort columns, or create variants.

The following indicators specify the status of a spool request (or an output request):

- Not yet sent to the operating system (no output request exists).

+ Spool request is just being created (stored in spool system).

Waiting

The output request has not yet been processed by the spool system.

Waiting in host spooler

The output request is intended for the spool work process or formatting in the host spool system or is currently being processed there.

In Proc.

A spool work process is formatting the output request for printing.

Printing

The request has been passed to the host spooler or to an output device, and a final status has not yet been returned to the SAP spool system.

Compl.

The output request has been printed. If the query at the host spooler is not activated, the status changes to *Compl.* as soon as the output request is transferred to the host spooler.

<F5>

There are (at least two) output requests with various statuses. To display the status of the individual requests, choose *Output requests* or double-click the request status.

Problem

Indicates an error, not of a serious nature, such as incomplete character set. This request was still printed.

Error

Indicates a serious error, such as a network error.

Time

A particular time was specified for the output of the request by the request creator.

Archive

The spool request has been processed by the spool system and is waiting to be archived. The archive status applies to spool requests sent to an archiving device.

To display the output requests associated with a spool request, you double-click the status of the spool request. To select output requests from several spool requests, mark them and choose *Output Requests* (or F5).

For every unsuccessful output request, a log is written that you can use for error analysis.



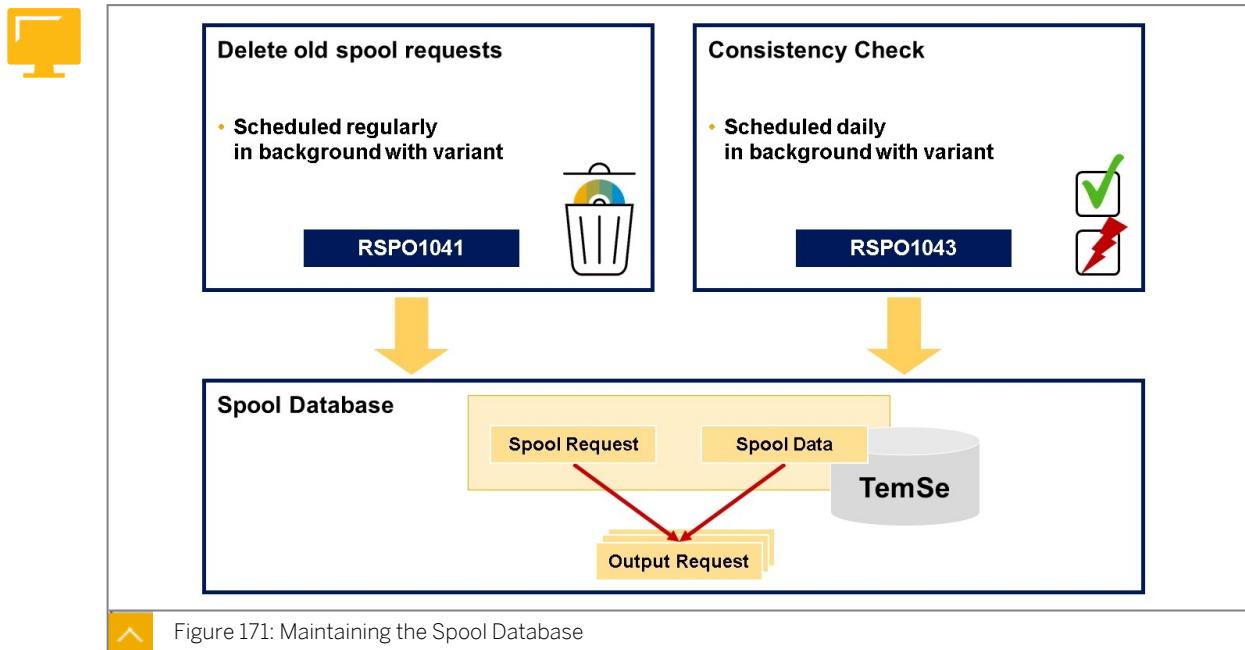
Note:

With SAP S/4HANA 2022, SAP ships an SAP Fiori app (*Maintain Print Queues* app) to set up print queues to manage the printing of documents and monitor the print jobs in each queue. This helps you to identify and analyze errors and gives you a sense of direction for doing troubleshooting to solve them.

A print queue consists of a name, a description, a format (such as *PDF* or *PCL 5c (Color)*) and an output device.

For more information, see the online documentation for SAP S/4HANA 2022 (*product assistance*), area *Enterprise Technology* → *ABAP Platform* → *Administrating the ABAP Platform* → *Administration Concepts and Tools* → *Output Management* → *Maintain Print Queues*.

Housekeeping Activities for Spool



The system administrator is responsible for regularly deleting old spool requests and ensuring the consistency of the spool database.

As listed in the figure *Maintaining the Spool Database*, you delete old spool requests by scheduling the ABAP program *RSPO1041* with an appropriate variant as a periodic job. For more information, see SAP Notes [130978](#) – *RSPO1041 - alternative to RSPO0041* and [3042383](#) – *Example settings for RSPO1041 run*.

To check the consistency of the spool database, schedule the ABAP program *RSPO1043* with an appropriate variant as a daily job. For more details of the program *RSPO1043*, see SAP Note [98065](#) – *Spool consistency check with RSPO1043 as of 4.0A*.

You can also start spool administration tools (in dialog) in transaction **SPAD** on the **Admin.** tab page. Note, however, that the functions for deletion and consistency checks in **SPAD** are limited, compared to the specified programs.

Additional Information

For more information on housekeeping for spool, see the online documentation for SAP S/4HANA (product assistance), area *Enterprise Technology* → *ABAP Platform* → *Administrating the ABAP Platform* → *Administration Concepts and Tools* → *Solution Life Cycle Management* → *SAP Printing Guide (BC-CCM-PRN)* → *Maintaining the Spool Database*.

When you have found an inconsistency in the spool database or in the database for temporary sequential objects (TemSe), SAP Note [48400](#) – *Reorganization of TemSe and spool* may be helpful. In addition, when the creation of new spool requests takes an extremely long amount of time, SAP Note [48284](#): *System cannot generate any more spool requests* may be helpful.

Tables, Authorizations and Transactions for Spool

The following figure shows selected tables, authorization objects and selected transactions that are used in the spool environment:



Selection of tables for spool

- TST01 – TemSe objects
- TST03 – TemSe data
- TSP01 – Spool requests
- TSP02 – Output requests
- TSP03 – Printer declaration (output devices)
- TSP0A – Device types

SE16

Authorization objects

- S_SPO_DEV – allows access to named output devices
- S_SPO_PAGE – restricts print quantities for requests to a certain maximum number of pages
- S_SPO_ACT – spool actions (e.g. for transaction SP01)
- S_ADMI_FCD – system authorization

PFCG

Selected transactions for spool

- SPAD – Spool administration
- SP01 – Output controller – display spool requests and output requests
- SP11 – Display TemSe objects
- SP12 – TemSe administration
- SPOOL – Overview of important spool reports and transactions



Figure 172: Tables, Authorizations and Transactions for Spool

Tables for Spool

TemSe is a storage for temporary sequential data; that is, objects that are not normally permanently held in the system are stored in the TemSe. Every TemSe object consists of a header entry in table **TST01** and the actual object. With spool requests, you can decide whether the object is stored in the file system or in table **TST03**. By default, a spool request is stored in table TST03.

Spool requests also have entries in table **TSP01** (Spool Requests) and possibly in table **TSP02** (Output Requests).

Tables **TSP03** and **TSP0A** finally contain information on output device and device types.

Authorization Objects

Authorization object **S_SPO_DEV** is used to control access to a specific printer or other output device. This authorization object contains field *SPODEVICE* (*Spool: Long device names*) in which you enter the SAP names of the output devices for which a user is to be authorized (for example **A*** for all output devices starting with A).

Using authorization object **S_SPO_PAGE**, (with fields *SPODEVICE* – device name for which the restriction is to apply and *SPOPAGES* – maximum number of pages allowed) you can restrict the maximum number of pages of a request that can be printed on a particular printer.



Note:

This authorization check is only active if profile parameter *rspo/auth/pagelimit* is set to **1**.

S_SPO_ACT is the authorization object for the output controller (transaction **SP01**) to define who is allowed to perform actions on spool requests that are protected with authorization character strings. Authorization for authorization object **S_SPO_DEV** are checked when users attempt to access the spool request of **another** user. The authorization is, therefore, mainly of interest for users who administer spool requests in the output controller. These users must access the spool requests of other users. This requires a **S_SPO_ACT** (spool: actions) authorization, in addition to the spool administration authorization (**S_ADMI_FCD**, system functions).

**Note:**

The authorization is not checked if users access their own unprotected spool requests.

`S_SPO_ACT` consists of the fields

- `SPOACTION` (specifying which operations are permitted on protected spool requests, for example `DISP` – display contents of a protected spool request, `REPR` – output protected spool request more than once, or `DELE` – manually delete request)
- `SPOAUTH` (authorization value for with the user is authorized)

A user is permitted to perform a spool action if the value stored for this action in the user master matches the value in the authorization field in the spool request. If no value is specified in the authorization field of the spool request, all actions are permitted.

**Hint:**

An authorization key can be entered at the time a spool request is created, for example when a user selects *Print* (using the *Properties* button, then switch to folder *General Properties* → *Spool request*). If no value is specified, the spool system automatically uses the ID of the user creating the request as an implicit authorization value.

Finally, system administrators require one or more values for the authorization object `S_ADMIN_FCD` (with field `S_ADMIN_FCD`) to be able to administer the spool system in certain clients. For example, the authorization to administrate spool requests is controlled using the values `SP01` (cross-user, cross-client) and `SP0R` (cross-user, client-specific) or the authorization for spool administration is controlled using the two groups of authorization values `SPAD` (cross-client) and `SPAR` (client-specific). In the same way, `SPTD` (cross-client) and `SPTR` (client-specific) are TemSe authorizations.

Selected Transactions

Transaction `SPAD` is used to create a device definition in the SAP system for every output device. and to create logical spool server.

Using transaction `SP11`, you have the option to display a list of TemSe objects (according to certain selection criteria).

Transaction `SP12` (*TemSe – Administration of Temporary Sequential Data*) can be used for example to list all TemSe objects, to check the memory allocation for TemSe and to perform a consistency check of TemSe. This consistency check program `RSTS0020` checks the header entries in table `TST01` and the objects that are associated with the header entry (stored either on file system level or in database table `TST03`) – it neither checks entries in table `TSP01` nor in table `TSP02`.

**Note:**

To avoid timeouts when running in dialog, you can use program `RSTS0020` to determine inconsistencies in the background (and delete them in dialog with report `RSTS0030`).

Transaction **SPOOL** finally provides an overview of important spool reports and transactions.

Transaction SPOOL

The screenshot shows the SAP interface for Transaction SPOOL. The title bar reads "Overview of Important Spool Reports and Transactions" and "SPOOL". The menu bar includes "Expand Subtree", "Collapse Subtree", "Execute", "Expert Mode", "Restrict Values", "More", "Search", and "Exit". A toolbar at the top right contains icons for "Additional expert tools", "Tool", "Transaction / Program for direct access", and "SAP Note for further information". The main area displays a table with columns "Description", "Report / Transaction", and "Note". The table lists various spool-related transactions and programs, such as SPAD, RSPO_CHANGE, RSPO0040, RSPO0029, RSPO_FIND_DEL, and RSPO0067. The "Tool" icon points to the "Tool" column, the "Transaction / Program for direct access" icon points to the "Report / Transaction" column, and the "SAP Note for further information" icon points to the "Note" column. A legend at the bottom left identifies these icons.

Figure 173: Transaction SPOOL

In the area of output management, many transactions and reports are available. As an alternative starting environment, you can use transaction **SPOOL**. This transaction displays a list of important transactions and reports (together with relevant SAP Notes) in the spool area. The transaction allows you to directly call the displayed transactions and reports by double-clicking them.

Procedure for Analyzing SAP Printing Problems

The following figure shows a procedure for analyzing a printer problem.

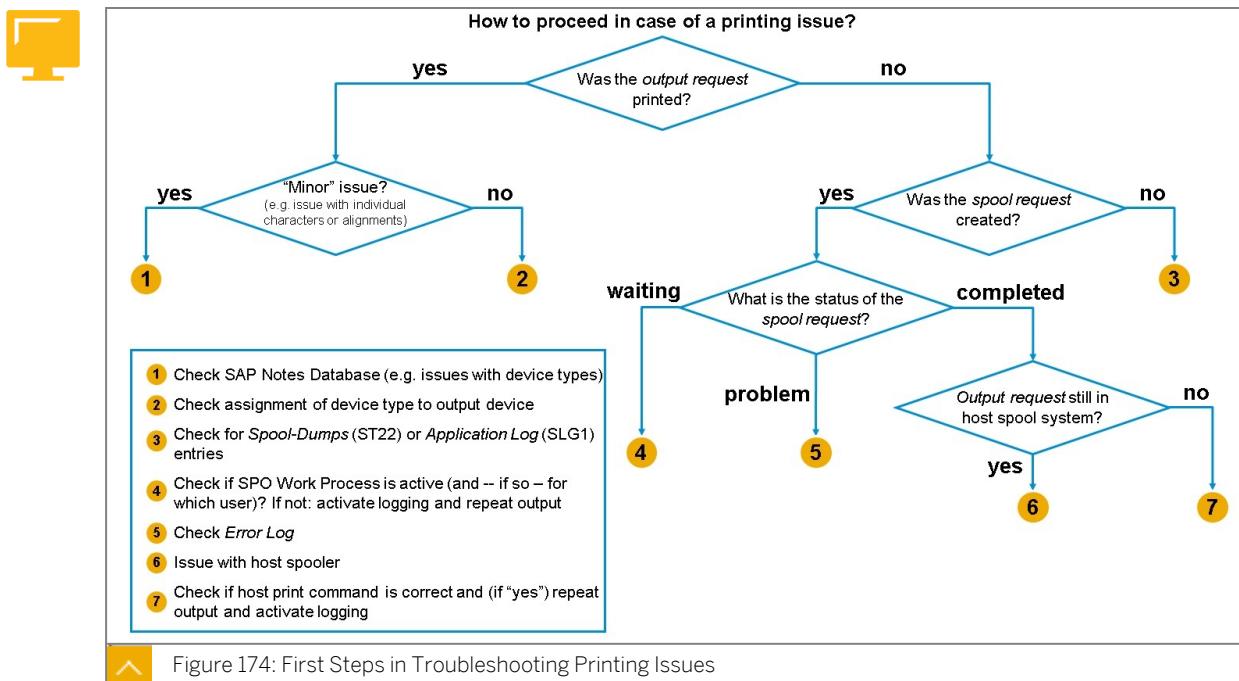


Figure 174: First Steps in Troubleshooting Printing Issues

In addition to checking physically whether output has appeared, you can use the spool output controller (transaction SP01) with its selection options to check on the status of a job. With the help of this transaction you can check the status of both the spool request and the output request (if it exists) to determine how to proceed.

In case that the output request has been printed, but it contains errors, it is important to distinguish between minor problems and major problems with the appearance of output that has actually been printed.

- A *minor* problem has occurred when the print out is legible and generally correct (option (1) in the figure above). Examples could be that some characters are missing or are represented by the # character (because they may not exist in the character set of the output device). For solutions of issues like this, see the SAP Notes database on SAP Support Portal (quick link [/notes](#)).
- A *major* problem has occurred when an output request has been printed but is not readable (option (2) in the figure above). For example the output is in the incorrect character set, or lines breaks and formatting are severely incorrect. In this case, you need to check whether the correct device type specified was in the output device definition and / or whether the emulation specified at the printer match the emulation specified in the device definition.

In case that you neither find an output request nor a spool request (and you made sure that the spool request has not been deleted already for example due to the *Delete after printing* option), the spool system seemed to be unable to generate a spool request when the user has selected to print (option (3) in the figure above). In this case, the spool program logic terminates the program and records all the details about the abnormal termination in an ABAP short dump. You can use transaction ST22 to analyze the program termination or to obtain additional information before you create an SAP ticket. Maybe you can also find additional information in the application system log (transaction SIG1)

 Note:

For example you want to create more than 32,000 spool requests in the SAP system. In the SAP default, the number of print requests is restricted to 32000, however. If you reach this limit, no free number can be found and the short dump *SPOOL_INTERNAL_ERROR* appears where the message "Spool Full" or "Spool Overflow" appears in the dump itself or in the system log. You have two options for reacting to the message: Either you have your print requests deleted regularly in the background (with the help of program *RSPO1041*) or you extend the number assignment as described in SAP Note [48284](#) – System cannot generate any more spool requests.

When a spool request is sent to a printer, then the spool system formats the output data for the printer. This creates an output request. It then passes the resulting output request to a host spool system or print manager for output (which should be performed within a couple of seconds). In case that the spool request has been generated but is still in status *Waiting* (option (4) in the figure above), the spool request has been sent to a printer, but it has not yet been processed which means that the output request is still waiting in the SAP spool system and has not yet been passed on to a host spool system and output device. In this case, the root cause may be:

- If you find a lot of spool requests with the status *Waiting* in the output controller, then there may be a problem with the communication pathways within the SAP system. You can

eliminate this possibility by checking that the message server of your SAP system is functioning correctly.

- Status *Waiting* may also indicate that there is a problem with the spool work process that is assigned to a printer. So you may also check the status of the spool work process. For this, you should first check the definition of the output device in question (transaction SPAD) to find the spool server involved. Then you can use transaction SM50 to check whether the spool work process(es) of that application server are active (and check the name of the active user).
 - An active user SAPSYS indicates that the spool work process is busy with internal spool activities or is requesting the status of a print request from a host spool system. For example, if the spool work process requests status and the target host system does not reply immediately, then the spool work process must wait for the query to time out. During this time, the spool work process cannot work on any other output requests. The time-out wait suggests that either the host system is not active or the network link to the host system is not working. If this problem occurs frequently, then you may think about turning off status querying for this output device.
 - An active “real user” indicates that the spool work process is processing an output request for a user. If this takes a long time, the spool work process either is processing a large output request and/or is sending the output to a slow communication partner (for example due to a slow WAN link). Or multiple output requests are queued up for processing in the spool work process.



Note:

If you are having frequent problems with delayed processing of spool requests and/or spool work processes waiting for communications time-outs, then you may want to run your spool work processes with the developer trace activated (profile parameter *rdisp/TRACE* – for more specific trace options, see SAP Note [2420274 – How to set Level 2 Tracing for Spool](#)).

If the status of a spool request has been set to *Compl.* or *Problem* but no output is printed (options (5) to (7) in the figure above), then the SAP spool system finished processing the spool request and is working normally but problem with the data in the spool request, with the printer definition in the SAP System, or with the host system spool or printer prevented any output from appearing. With the help of transaction SP01, you can then check the status of the assigned output request:

- If the output request is in status *Problem* (option (5) in the figure above), check the error log, which may contain additional information from the spool work process, from the SAPlpd transfer program (for Windows) and from the host spool system or print manager. If you cannot isolate and resolve the printing problem on the basis of the output request log, repeat the print request after switching on the logging of output data – see SAP Note [2428307 – How to retain temporary print data file](#) for more details.
- If the output request is in status *Completed* and the output request is still in the host spool system (option (6) in the figure above), this may indicate an issue with the host system.
- Finally, if the output request is in status *Completed* and the output request is not in the host spool system any more (option (7) in the figure above), you may check whether the host print command was correct and – if “yes” – repeat the output with activated output data logging according to SAP Note [2428307 – How to retain temporary print data file](#).

**Hint:**

To check whether the host print command was correct, check the value of the `rspo/host_spool/print` profile parameter. You may try to execute this command at the host system command line. If you are not able to print successfully using this command, check your host system documentation to find out what is incorrect in the command.

Additional Information

For more information on troubleshooting, see the online documentation for SAP S/4HANA (*product assistance*), area *Enterprise Technology* → *ABAP Platform* → *Administrating the ABAP Platform* → *Administration Concepts and Tools* → *Solution Life Cycle Management* → *SAP Printing Guide (BC-CCM-PRN)* → *Analyzing SAP Printing Problems*.

In addition, the composite SAP Note [504952](#) – *Composite SAP Note for spool and print* lists many important notes that may be helpful when analyzing print problems.

SAP Note [118057](#) – *Flexible Configuration of the Spool Service* provides additional background information on the SAP spool architecture.



LESSON SUMMARY

You should now be able to:

- Use monitoring transactions for printing
- Perform housekeeping activities for spool
- Analyze printing errors

Unit 8

Lesson 6

Planning the SAP Print Architecture



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Develop a Printing Concept

Introduction to Planning the SAP Print Architecture

You should plan your SAP print architecture especially carefully if you must deal with the following print tasks:

- Printing time-critical documents (such as delivery notes: if these are not added to the delivery, the product cannot be sent)
- Printing large print requests (for example, printing long monthly lists).

The following aspects are intended to support you in planning your optimal print architecture.

TemSe Storage Location

The print data of a spool request is stored in TemSe (a store for temporary sequential objects). You can define whether the objects are to be stored by TemSe in the SAP database or in the file system by setting the profile parameter *rspo/store_location*.

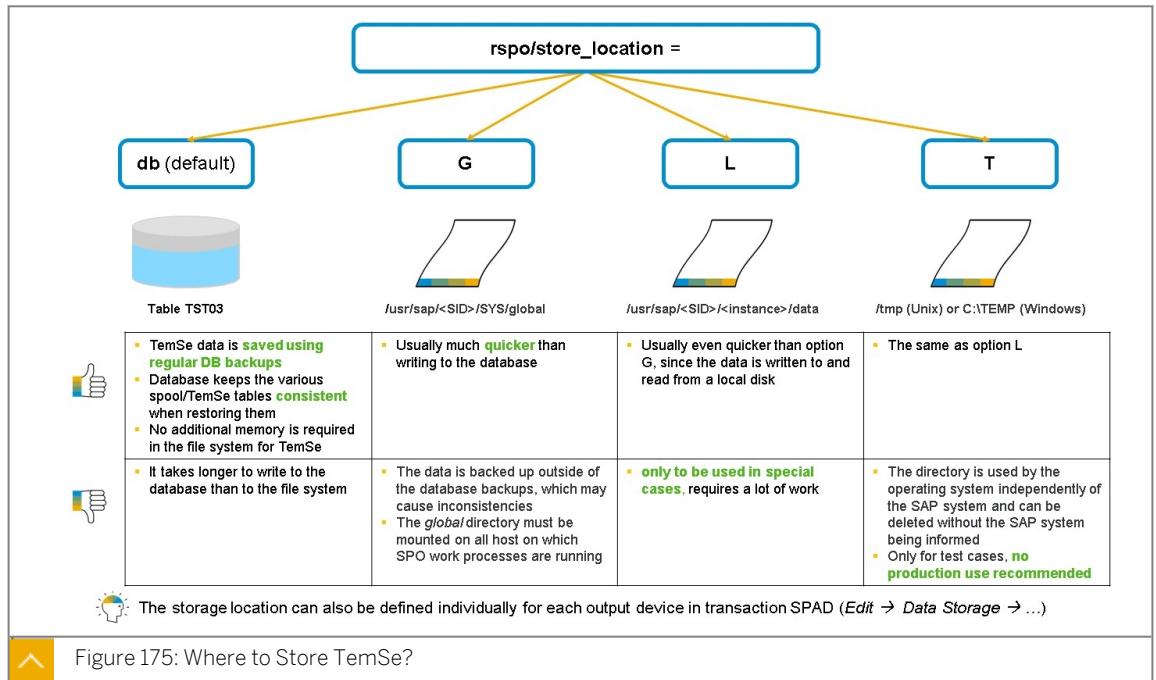


Figure 175: Where to Store TemSe?

- When you use value **db** (the default value), the print data of a spool request is stored in the database table *TST03*. Advantage: The spool/TemSe data is saved using regular DB backups and then restored. The database keeps the various spool/TemSe tables consistent when restoring them. No additional memory is required in the file system for TemSe. Disadvantage: It takes longer to write to the database than to the file system (generally also longer than writing to NFS file systems). The temporary spool data is always saved too, which makes the backups longer.
- When you use value **G**, the print data of a spool request is stored at the operating system level in the (global) directory `/usr/sap/<SID>/sys/global`. Advantage: Usually much quicker than writing to the database. Disadvantage: The data is backed up outside of the database backups, which means that the administration tables *TSP01* and *TST01* might be inconsistent with the files in the file system and need to be made consistent again using consistency checks after restoring a backup in the case of disk crashes. If the SAP system consists of multiple instances on multiple machines, the 'global' directory must be mounted on all machines using NFS; if not, inconsistencies occur and spool operations cannot run without errors. Enough disk space must be available, since spool data can easily reach volumes of several GB in the file system.
- When you use value **L**, the print data of a spool request is stored at the operating system level in the (local) directory `/usr/sap/<SID>/<Instance>/data`. Advantage: Usually even quicker than option G, since the data is written to and read from a local disk. Caution: This value should only be used in special cases. If misused, it could cause inconsistencies in spool/TemSe and produce printing errors. L specifies that the creator of the spool data and the spool work process are part of the same instance and run on the same physical machine (to which the printer is also connected). The printer must also be assigned to this spool work process. The spool requests are created in this way so that they can only be edited on this instance. This type of configuration requires a lot of work and is therefore not recommended.
- When you use value **T**, the print data of a spool request is stored at the operating system level in the temp directory `/tmp` (Linux/UNIX default), `C:\Temp` (Windows default). This directory is used by the operating system independently of the SAP system and, more specifically, is deleted without the SAP system being informed. This can produce data loss, inconsistencies, and other errors when printing. This means it should never be activated in production environments.

For more information, see SAP Notes [20176 – Where is the spool request saved?](#) and [11070 – Space requirements of TemSe and spooler](#).



Note:

You can also define the storage location individually for each output device in transaction **SPAD** (to do so, choose *Edit → Data Storage* in the list or details of the output device).

Criteria to Choose the Access Method

Local Printing (Access Methods C and L)

Local printing can be used if the host spool system (operating system spooler) and the spool work process of the SAP system are on the same host. It can be used for operating systems Windows and UNIX.

Local printing is the fastest and most reliable form of printing from the point of view of SAP systems. As a prerequisite, it requires that the application server on the host which contains the operation system spooler has at least one spool work process (profile parameter *rdisp/wp_no_spo*).

For local printers, on the other hand, dynamic server selection (with the help of logical spool servers) requires some preparation at the operating system level. For example, the target output device must be defined under the same name in all possible local spoolers (in the hosts of all possible spool servers).

Remote Printing (Access Methods S and U)

Remote printing needs to be used if you are printing over a network; that is, when the host spool system (operating system spooler) and the spool work process of the SAP System are on different hosts.

As a prerequisite, there must be a network to transfer the data to the print server (and for mass printing, this should be a LAN network). In addition, remote printing requires fixed IP addresses and reliable communications partners (so that timeouts do not occur).



Note:

For Windows-based hosts, the use of SAPSprint is recommended because it can also interpret data streams that were generated with the generic, that is device-independent, device type SWIN, and call the appropriate Microsoft Windows drivers that generate device-specific data streams. In contrast, an TCP/IP Print Server can only forward data that has already been formatted for the printer (that is, device-specific data) directly to Microsoft Windows without using Microsoft Windows printer drivers

Front-End Printing

Front-end printing is available to reduce the significant administration effort associated with this. This means that a user logged on to the SAP system through the SAP GUI can use the printers that are set up at his or her front end PC. A generic output device needs to be created in the SAP system to allow this.

Constraints for front-end printing are:

- you cannot perform front-end printing in the background
- a free session is required for front-end printing, so it must not be the case that all sessions are occupied
- since front-end printing uses the local GUI connection, it is generally unsuitable for the output of large or particularly time-critical documents
- when printing multiple spool orders, the sequence cannot be kept because all spool orders are processed in parallel in different threads

If you are using front-end printing, set up at least two spool work processes on each application server, if possible (profile parameter *rdisp/wp_no_spo*). In addition, check and set the maximum number of spool work processes that can be used for front-end printing (profile parameter *rdisp/wp_no_Fro_max*).

Summary

The following figure summarizes the statements from the text above.



	Local Printing C L	Remote Printing S U	Frontend Printing G
	<ul style="list-style-type: none"> ▪ Fastest and most reliable form of printing 	<ul style="list-style-type: none"> ▪ Can use dynamic server selections without restrictions 	<ul style="list-style-type: none"> ▪ Reduces administration effort significantly
	<ul style="list-style-type: none"> ▪ Additional effort needed for dynamic server selection (logical spool servers) 	<ul style="list-style-type: none"> ▪ Requires network and reliable communication partners (no timeouts) 	<ul style="list-style-type: none"> ▪ Cannot be used in background ▪ Generally unsuitable for mass printing and time-critical production printing

 Figure 176: Which Access Method to Choose?

Steps to Plan the Print Architecture

The first task to plan a print architecture is to determine the most important print requirement. For this, you should identify which users print time-critical documents (for example, delivery notes), which users prints large documents (such as ABAP list reports) and what is the expected total quantity of output request in your SAP system.

Next you need to determine how important the respective printers are (important printers are for example printers whose output is time-critical).

Then you may think about classifying the printers. In case that you have set up three or more spool servers in your SAP system, you may then finally assign every print group to one or more spool servers.

To get an optimal printer throughput, you may distinguish (see the following figure):

- For **production printing** (time-critical printing), you should usually use local printing as it is probably faster than remote printing. Remote access methods should be taken account for production printers in case of high reliability (failure safety with the help of logical spool servers) in case that the network between the spool work process and the host spool system allow a high data throughput.
- Similar to production printing, for **mass printing**, local access methods also have the advantage of better performance – when you use remote access methods, you require a fast network and a reliable printer server. You may also think about setting up a separate spool server for mass printing so that processing of long lists does not affect the output of other output requests.
- For **non-critical printers**, you may also setup a separate spool server so that they do not affect and are not affected by print requests for other printers. You should never use a spool server for both time-critical and non-critical printers, as the non-critical output requests could block the processing of the time-critical output requests. For example, if a non-critical printer that uses access method U or S cannot be accessed, time-critical printing may be seriously delayed.

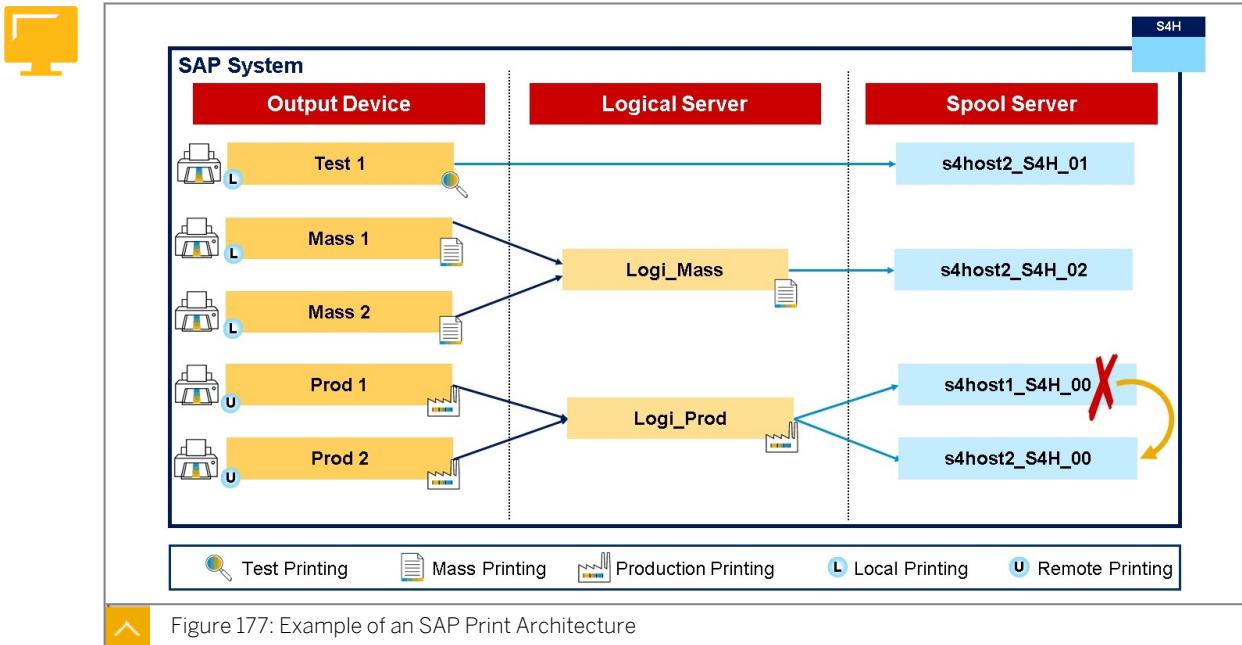


Figure 177: Example of an SAP Print Architecture

**Hint:**

To optimize the output speed, you can also deactivate the status query through the host spool system and the print manager. To do this, select the *Do not query host spooler or printer for print requests* option in the relevant device definition (transaction SPAD).

Distributing the Print Architecture

As you typically have multiple SAP systems in use that use the same printers, after defining the print architecture there comes the question of how to distribute the print landscape to the single SAP systems (and how to redistribute it after you have performed some changes).

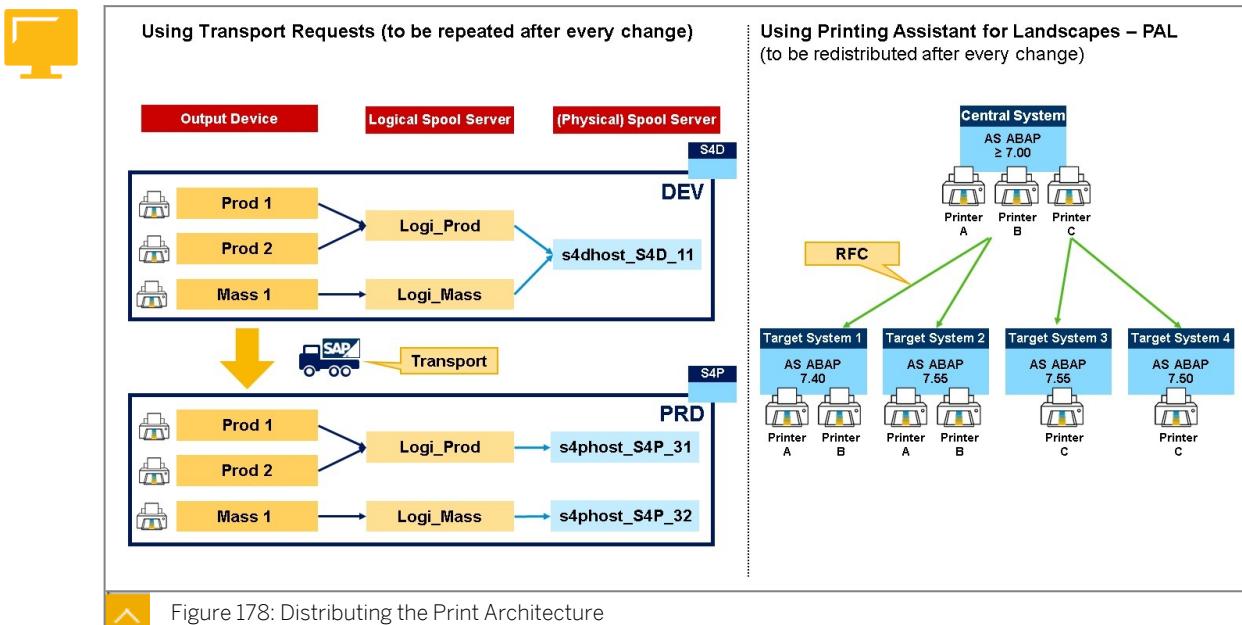


Figure 178: Distributing the Print Architecture

The concept of logical servers supports you when defining a consistent, transportable print landscape. For example, unlike real spool servers, logical servers can have the same name in different SAP systems. Therefore, you can define a consistent SAP print architecture in the development system and then transport it to other systems. All output devices and logical spool servers are transported. After the transport, all you need to do is adjust the mapping of the logical servers to the (physical) spool servers of the new system, as shown in the left part of the figure *Distributing the Print Architecture*.



Note:

There are functions for the manual transport of output devices and logical servers in transaction SPAD.

The *Printing Assistant for Landscapes* (transaction PAL) gives you another option for distributing output devices within an SAP system landscape (see the right part of the figure *Distributing the Print Architecture*).

You can use PAL to configure output devices in one system instead of configuring them separately in each system of your system landscape. After you have decided which system shall serve as the “Central System”, you define all output devices of your system landscape as PAL Printers in this system. Afterwards you distribute the PAL Printer definitions from the “Central System” to other systems (“Target Systems”) via RFC. When the distribution is successfully finished, all PAL Printers can be used immediately on the Target Systems for actual printing.



Note:

For more information on PAL, see the online documentation for SAP S/4HANA (product assistance), area *Enterprise Technology* → *ABAP Platform* → *Administering the ABAP Platform* → *Administration Concepts and Tools* → *Solution Life Cycle Management* → *SAP Printing Guide (BC-CCM-PRN)* → *Printing Assistant for Landscapes (PAL)*.



LESSON SUMMARY

You should now be able to:

- Develop a Printing Concept

Appendix: Managing Output Devices with Printing Assistance for Landscapes (PAL)



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the basic idea of Printing Assistance for Landscapes (PAL)
- List the steps to set up PAL
- Describe the distribution of printer groups to target system groups

Introduction to Printing Assistant for Landscapes

You can simplify the administration of output devices in ABAP-based SAP systems using the Printing Assistant for Landscapes (PAL). To do so, you maintain output devices in one system (the “central system”), and distribute their definition across any number of additional systems, called “target systems” with the help of RFC destinations— instead of configuring them separately in each system of your system landscape.

We recommend using PAL if at least one of the following criteria applies:

- You manage many output devices in a few SAP systems.
- You manage a few output devices in many SAP systems.
- You manage many devices in many SAP systems.

Before you can start using the Printing Assistant for Landscapes, you must define your PAL landscape which means that you have to decide which system shall be used as central system and which systems shall be used as target systems. The central system is the only system in your PAL landscape where you actually use transaction PAL.

All AS ABAP based SAP systems that are still in mainstream (or extended) maintenance can be used for PAL – for more details, see SAP Note [964501](#)– *Printing Assistant for Landscape (PAL)*.



Note:

Since SAP Solution Manager Systems also enables central administration of other functions, you can use an SAP Solution Manager system as the central system for PAL.

The figure *PAL Landscape* illustrates a small PAL Landscape to indicate the release requirements of the systems involved.

**Hint:**

While transaction `SPAD` uses the term “output device” to refer to the name of a print device in the SAP System, transaction `PAL` uses the term “printer”. Each `PAL` Printer requires an associated output device as a source when you create it. A printer in transaction `PAL` therefore corresponds to an output device in transaction `SPAD`.

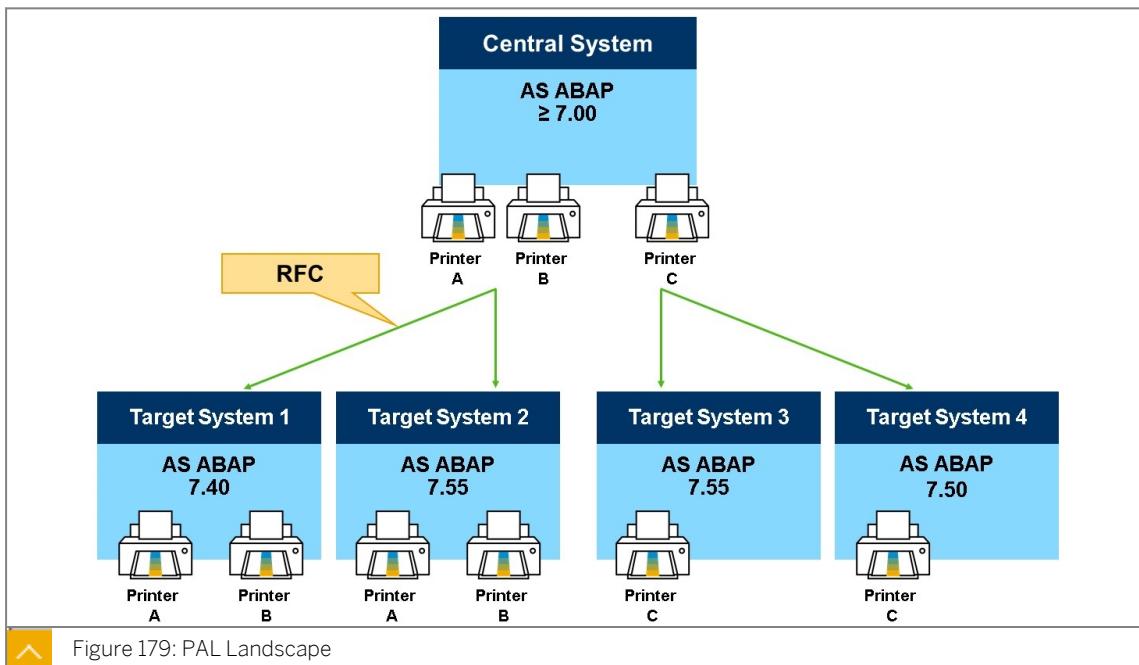


Figure 179: PAL Landscape

Supported Access Methods

Each output device you want to use in your PAL landscape must exist in the spool administration (transaction code `SPAD`) of the Central System or of the Target Systems in order to become available in `PAL`.

In general, `PAL` supports the following access methods:

- L and C: Local printing
- U and S: Remote printing
- G: Front-end printing
- M: Printing via e-mail
- E: External output management systems

However, depending on the access method, there are some prerequisites for local printing and for external output management systems, for example:

- Output devices using access method C must have an – for example – Windows spool server assigned. The `PAL` Printer is accessed from the spool server by a direct Operating System API call. During the distribution process, `PAL` will make sure that a Windows spool server will be selected in the target system which the `PAL` Printer will be distributed to but `PAL` can not make sure that the selected spool server knows the access method C `PAL`

Printer on OS level. It is therefore likely that PAL will select a wrong spool server. Therefore you will receive a system warning before a distribution of an access method C PAL Printer takes place. If no Windows spool server can be found in a particular target system, the distribution is not possible.

- Printer jobs are sent to output devices linked with access method L using an operating system command. On Unix this is either the *LP* or *LPR* command. During the distribution process, PAL cannot check whether the name of the PAL printer under which the printer of the operating system is known is identical to the name that is used in the printer spool server of the target systems and in the printer spool server of the central system (field *Host Printer* in transaction *SPAD*). And it can also not check whether command used to send the print request and query the job status is the same in the target systems and in the central system and is able to be executed there. As a conclusion, PAL can therefore not guarantee that PAL printers work without errors in the target system, even if the distribution was successful. Therefore you will also receive a system warning before a distribution of an access method L PAL Printer takes place.

For additional information, see SAP Note [1240633](#) – *PAL: Transfer of spool server*.



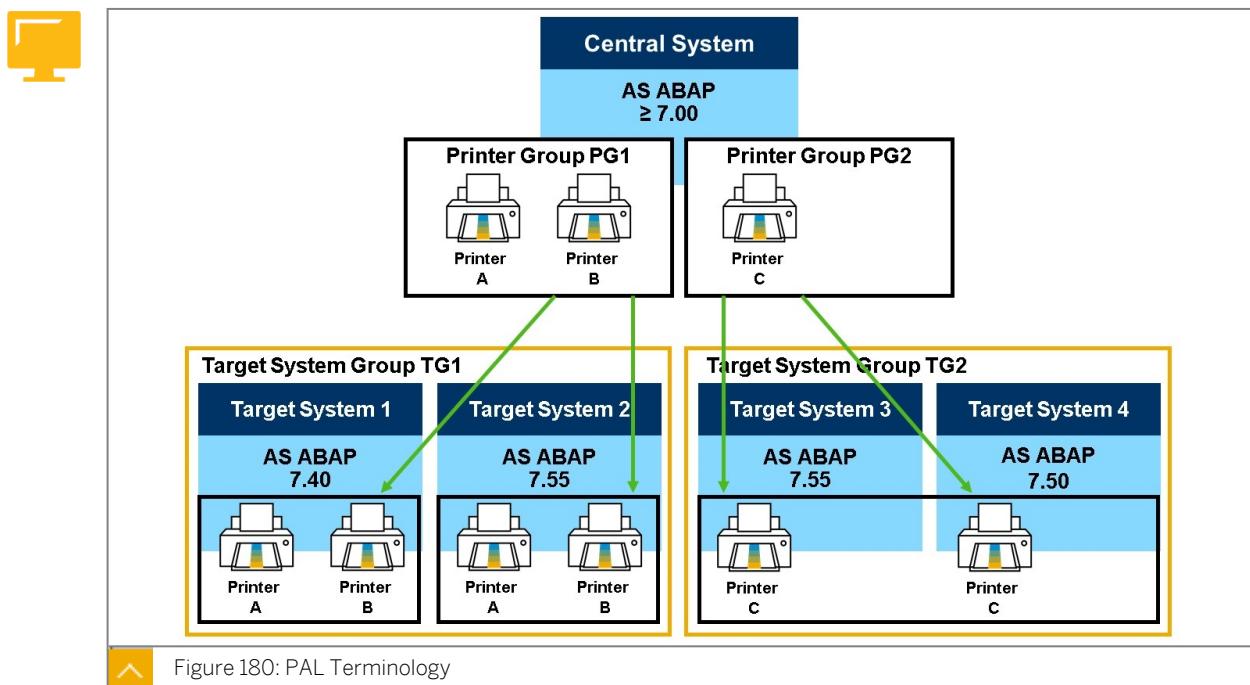
Note:

As it can be necessary to change the spool server of one or more PAL Printers after the distribution of PAL Printers with access method C or access method L in the target system (because PAL cannot ensure that the correct spool server has been selected during distribution), SAP ships program *SPCPC_SPOO_SERV_MASS_CHANGE*.

Set Up the Printing Assistant for Landscapes

Central Terms for PAL

Figure *PAL Terminology* shows some central terms in the context of PAL:



(PAL) printer

The starting point for PAL is an output device (created with transaction SPAD). This output device must then be designated as a PAL printer (often simply referred to as a “printer” here).

Printer group

A printer group normally comprises several printers that are jointly distributed later.

Target system

A target system in a PAL environment refers to an RFC destination (which is maintained in the central system). This RFC destination contains connection and logon data for a suitable ABAP-based SAP system in the PAL landscape.



Note:

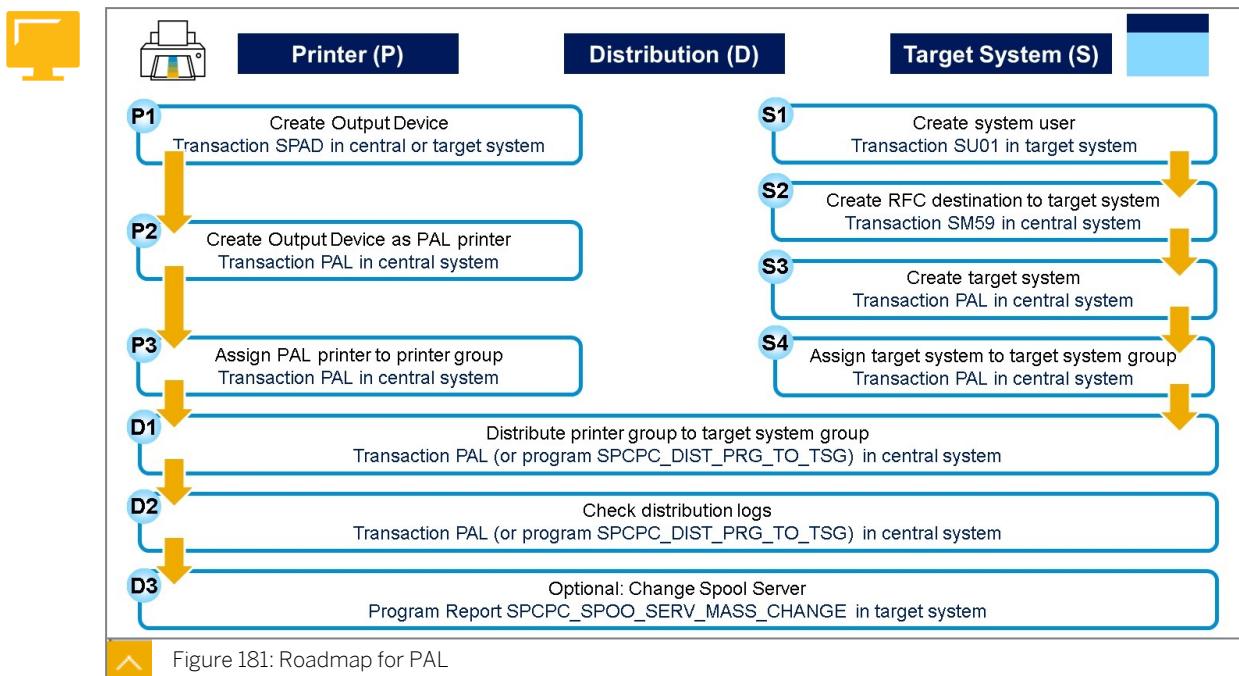
RFC destinations which will be used for PAL must have connection type 3 (ABAP connection). It is recommended to enter a system user (**not** to use the *Current User* option) and to leave the *Language* field empty.

Target system group

Printer groups are always distributed in a target system group, which usually consists of several target systems.

Steps in Planning the PAL Landscape

The figure *Roadmap for PAL* shows the steps for defining and using a PAL print landscape (here S indicate steps to be defined per system, P indicate steps to be performed per printer and D finally indicate steps related to the distribution of printers):



First you need to decide which system in your system landscape should act as the *central system* and which systems you want to use as *target systems*. Then you need to create RFC destinations from the central system to the target systems including system users (steps S1 and S2 in the figure above).

Then you define your output devices as *PAL Printers* (step P2) and put them together into *printer groups* (step P3). You can choose from the list of output devices created in the Spool Administration (transaction SPAD – step P1) in the central system, and you can also retrieve output device information from the target systems. As a result, all PAL Printer definitions are finally located in the central system.



Note:

PAL Printers can only be administered in the central system. Changing or deleting PAL Printers in a target system is not possible.

Next, you need to form *target system groups* out of the target systems (steps S3 and S4).

Finally you can start distributing *printer groups* to *target system groups* (step D1). Distribution can take place between *printer groups* and *target system groups*, but not between single PAL printers and target systems. You can distribute one *printer groups* to several *target system groups*.

The remaining part of this lesson explains individual steps of the roadmap in more detail.

PAL Landscape Maintenance – System Related Steps

Printer definitions are distributed from the central system to the target systems using RFC. You therefore need an RFC destination in your central system to each target system. It is recommended to set up a dedicated system user for spool administration in every target system.



Note:

Concerning the required authorization, see SAP Note [1573994 – User role for PAL client system](#).

To ensure a target system is recognized by PAL, it must be created with transaction `PAL` in the central system. Printers are then always distributed using *target system groups* that define a logical collection of target systems. A target system group could, for example, consist of all production systems.



Note:

A target system cannot belong to multiple *target system groups*.

The main steps in the initial setup are as follows:

1. In the central system, create an RFC destination for each target system (including logon data – this corresponds to steps S1 and S2 in the roadmap above).
2. Call transaction `PAL` in the central system.
3. Next to *Show*, select *Target System* and then *Create* (step S3). Enter a *Target System Name* and select the corresponding *RFC Destination*.
4. Next to *Show*, select *Target System Group* and then *Create*. Enter a *Target System Group Name*. You can select the target systems you want to add now or later (step S4).

Create and Distribute PAL Printers

After you have defined printers in the central system or a target system as output devices in transaction SPAD (step P1), PAL now allows you to manage and distribute those printers as *PAL printers*. You can then combine these PAL printers in *printer groups*. For example, one *printer group* can incorporate all test printers. You can then distribute *printer groups* to *target system groups*.



Note:

A PAL printer cannot belong to multiple *printer groups*.

The main steps in the setup are as follows:

1. Call transaction PAL in the central system.
2. Next to *Show*, select *Printer* and then *Create*. The wizard prompts you to enter whether you want to publish the output devices as printers from the central system or a target system (step P2).



Note:

If you set the *Overwrite* flag, the new PAL Printer will overwrite the original output device in the central system.

3. Next to *Show*, select *Printer Group* and then *Create*. Enter the *Printer Group Name*. You can select the PAL printers you want to add now or later (step P3).
4. Now you are ready to distribute your PAL printers to your target systems (step D1). For this select a printer group in the *Printer Group* field and choose *Distribute*. Select one or more target system groups.
5. After distribution, check the *Distribution Status* (step D2).

When a printer is distributed to a target system for the first time, an arbitrary (real) spool server is assigned to the output device. Using the report SPCPC_SPOO_SERV_MASS_CHANGE (which can be accessed from transaction SPOOL (folder *Spool Administration* → *Printing Assistant for Landscapes (PAL)* → *Spool Server Change in Target System*) in the target system, you can assign a (logical or real) spool server to some or all PAL printers (step D3). The only mandatory input field is *New Spool Server*. The checkbox *Confirm before change* is selected by default. If you use the default setting, a list of affected PAL printers is displayed. You can check first if the new spool servers are correct before you execute the report.



Note:

PAL printers can be changed in the target system with the help of transaction SPAD. Most of the properties are overwritten during the next (re-)distribution, however. Some fields (such as *Spool server*) are excluded from this, they are displayed on a popup as soon as you start to change an output device in the target system.

Change of PAL Objects

When you create a printer from a target system in transaction `PAL`, this output device is then also known in transaction `SPAD` on the central system. All further maintenance of this output device is then performed in the central system. This enables you to take changes to one PAL printer and transfer them to the connected target systems.



Note:

If you want to transfer a PAL printer that was copied from a target system back to that same target system, you have to delete the corresponding output device in the target system first. Otherwise, PAL will report that the target system already has an output device with the same name in the central system, but as a non-PAL printer.



Hint:

The error message `Lock failed, unknown lock` (or a slightly different message) indicates that the selected output device is currently being edited in change mode. To release the lock, leave transaction `SPAD` (or switch to *display mode*).

It is also possible to schedule the distribution as a background job (via report `SPCPC_DIST_PRG_TO_TSG`). For the minimum system requirements, see SAP Note [1476793](#) – PAL Printer Distribution in Background.

Additional Information

For more information on the Printing Assistance for Landscape, see

- SAP Note [964501](#): *Printing Assistant for Landscape (PAL)*
- SAP Note [1240633](#): *PAL: Transfer of spool server*
- SAP Note [1476793](#): *PAL Printer Distribution in Background*
- SAP Note [1054170](#): *FAQ: Frequently Asked Questions Regarding PAL*



LESSON SUMMARY

You should now be able to:

- Explain the basic idea of Printing Assistance for Landscapes (PAL)
- List the steps to set up PAL
- Describe the distribution of printer groups to target system groups

Learning Assessment

1. What is the main task of spool work processes?

Choose the correct answer.

- A Generating Spool Requests
- B Generating Output Requests
- C Generating Spool Data

2. Which of the following access methods exist in AS ABAP-based SAP systems?

Choose the correct answers.

- A Remote printing
- B Front-End printing
- C Instance printing
- D Local printing

3. Identify features of using logical spool servers.

Choose the correct answers.

- A Failover functions.
- B Load Balancing.
- C Make spool work processes available across different SAP systems.
- D Allow an enhanced authorization concept.

4. Which transaction provides an overview of important spool reports and transactions?

Choose the correct answer.

- A SPAD
- B SP01
- C SPOOL

5. Where should you store the TemSe if you focus on data consistency rather than on performance?

Choose the correct answer.

- A In the database, table *TST03* (*rspo_store_location* = **db**)
- B On file system level, directory */usr/sap/<SID>/SYS/global* (*rspo_store_location* = **G**)
- C On file system level, directory */usr/sap/<SID>/<instance>/data* (*rspo_store_location* = **L**)

6. The Printing Assistant for landscapes (PAL) allows the central maintenance of selected printer definitions across several SAP systems.

Determine whether this statement is true or false.

- True
- False

Learning Assessment - Answers

1. What is the main task of spool work processes?

Choose the correct answer.

- A Generating Spool Requests
- B Generating Output Requests
- C Generating Spool Data

You are correct! A spool work process creates output requests, based on spool requests. Spool requests and spool data can be created, for example, by dialog work processes.

2. Which of the following access methods exist in AS ABAP-based SAP systems?

Choose the correct answers.

- A Remote printing
- B Front-End printing
- C Instance printing
- D Local printing

You are correct! Instance printing does not exist.

3. Identify features of using logical spool servers.

Choose the correct answers.

- A Failover functions.
- B Load Balancing.
- C Make spool work processes available across different SAP systems.
- D Allow an enhanced authorization concept.

You are correct! Logical spool servers allow failover functions and load balancing. Spool work processes cannot be made available across different SAP systems and logical spool servers also don't allow an enhanced authorization concept.

4. Which transaction provides an overview of important spool reports and transactions?

Choose the correct answer.

A SPAD

B SP01

C SPOOL

You are correct. Transaction SPOOL provides an overview of important spool reports and transactions. Transaction SPAD can be used to create output devices and logical spool servers. Transaction SP01 provides an overview of spool requests and output requests.

5. Where should you store the TemSe if you focus on data consistency rather than on performance?

Choose the correct answer.

A In the database, table TST03 (*rspo_store_location = db*)

B On file system level, directory /usr/sap/<SID>/SYS/global (*rspo_store_location = G*)

C On file system level, directory /usr/sap/<SID>/<instance>/data
(*rspo_store_location = L*)

You are correct. When you decide to store the TemSe in the database (*rspo_store_location = db*), TemSe data is saved using regular DB backups and the database keeps the various spool /TemSe tables consistent when restoring them. When you store TemSe on file system level (*rspo_store_location = G or L*), access to it is usually much faster than accessing to the database.

6. The Printing Assistant for landscapes (PAL) allows the central maintenance of selected printer definitions across several SAP systems.

Determine whether this statement is true or false.

True

False

You are correct! The central maintenance of selected printer definitions is the core feature of the Printing Assistant for landscapes (PAL). Read more on this in the lesson “Managing Output Devices with Printing Assistant for Landscapes” of the course ADM100.

Lesson 1

Understanding the Concept of Background Processing

363

Lesson 2

Monitoring Background Processing

371

Lesson 3

Setting Up the Background Processing Environment

377

Lesson 4

Scheduling Background Jobs

387

Lesson 5

Troubleshooting Background Issues

399

UNIT OBJECTIVES

- Explain the basic concepts of background processing
- Explain the status flow of a job
- Use the Job Overview to monitor jobs
- Evaluate the need for reserving work processes for high-priority jobs
- Create Job Server Groups and explain their use
- Create RFC groups for the parallel processing of jobs
- Schedule housekeeping jobs
- Perform simple tests to check the consistency of the background processing environment
- Create a time-based job
- Create an event and a job based on that event
- Create a job that contains external commands or external programs

- List options to extend the standard background system
- List utility programs and tables for background processing
- Analyze problems in background processing

Understanding the Concept of Background Processing



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the basic concepts of background processing

Introduction to Background Processing

In this lesson, you will learn about the basics of background processing.

Business Example

Reports that are to run on a regular basis and long-running programs are scheduled as jobs in the SAP system. The administrator schedules jobs and monitors the system's background processing to ensure that it is running correctly.

The Idea of Background Processing

Dialog work processes for processing end-user requests should generally be made available immediately, or at least without a noticeable wait time. Therefore, dialog work processes should only be used for requests with short processing times to ensure the maximum possible throughput and rapid availability. Up to SAP BASIS 754, the system profile parameter *rdisp/max_wprun_time* exists for this reason. It limits the maximum runtime of a dialog step within a dialog work process. This should ensure that dialog work processes are not blocked by long-running programs, thus interfering with online operation. After the maximum runtime has elapsed, the program is terminated.

The operation of the parameter *rdisp/max_wprun_time* is described in detail in SAP Note [25528: Configuration of maximum work process runtime - parameter rdisp/max_wprun_time](#).

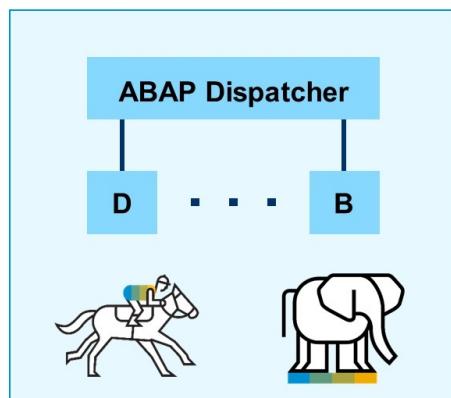
**Note:**

As of SAP_BASIS 740, a more precise setting is possible: The maximum runtime can be set using the following parameter depending on the kernel priority (high, medium, low):

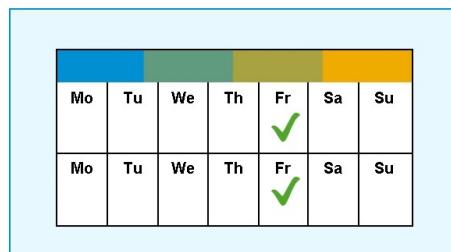
- *rdisp/scheduler/prio_high/max_runtime*: Maximum runtime for requests executed with priority **high** (for example, GUI requests).
- *rdisp/scheduler/prio_normal/max_runtime*: Maximum runtime for requests executed with priority **medium** (for example, RFC requests).
- *rdisp/scheduler/prio_low/max_runtime*: Maximum runtime for requests executed with priority **low** (for example, dialog requests triggered by background requests).

However, the following applies for compatibility reasons: If the parameter *rdisp/max_wprun_time* has been configured explicitly, its value overwrites the more specific settings.

As of SAP_BASIS 755, the parameter *rdisp/max_wprun_time* is removed from the ABAP kernel. Also the upgrade procedure removes it from the profiles during upgrade to SAP_BASIS 755 and above. See SAP Note [2918906: Deprecation of Profile Parameter "rdisp/max_wprun_time"](#) for details. The set of the three new parameters above provide much more flexibility. Moreover, SAP assumes that the default values of the new profile parameter should meet the needs of most customers and their applications. But because of the mentioned compatibility rules of releases prior to SAP_BASIS 755, their values are often overwritten and the benefit is lost. Also, a switch from an old profile where *rdisp/max_wprun_time* has been set to the new priority-based defaults might cause incompatibilities if the ABAP code of the application is not well written.



- **Scheduling of long-running tasks**
- **Relief for dialog resources**



- **Scheduling of regular tasks**



Figure 182: Purpose of Background Processing

As the figure *Purpose of Background Processing* illustrates, you can use the background work processes (sometimes also called batch work processes) for long-running tasks. You can also use background processing for recurrent tasks, such as the daily database backup or the month-end work for financial accounting.

You can define new jobs in transaction SM36. You can also call the *Job Wizard* (transaction SM36WIZ) from here. Alternatively, you can schedule background tasks in transaction SA38, along with numerous application transactions.



Note:

In most cases, background tasks are scheduled by experts in the user departments. As the system administrator, you ensure that the scheduled jobs can actually run in the system (by ensuring a suitable system configuration, for example).



Figure 183: Job Scheduling

As shown in the figure above, when you define a job, you have to enter the following information:

- General specifications such as job name, job class / priority (default: "C") and (optional) execution target / target server (group)
- Definition of one or more job steps
- Definition of a start condition (time-dependent or event-dependent)

Job steps and start criteria will be discussed in more detail in the remaining part of this lesson.

The *Job Wizard* supports you when defining jobs by guiding you easily through the creation process.

**Note:**

The way in which a job is created ("classic" or using the Job Wizard) has no influence on the result. Some functions, such as specifying an SAP user for each step, or swapping steps, are not available with the Job Wizard.

Jobs and Job Steps

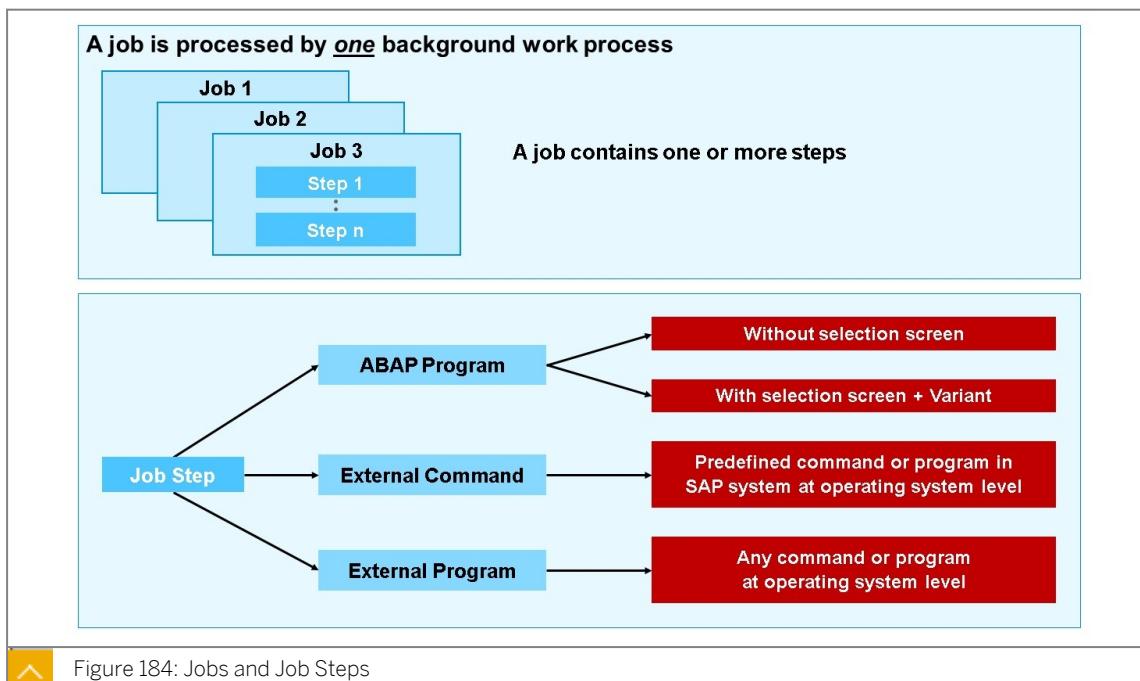


Figure 184: Jobs and Job Steps

Every job is processed without interruption by one single background work process. As outlined in the figure *Jobs and Job Steps*, a job comprises one or more steps. A step within a job can be the following:

- **ABAP Program:** Every ABAP program in the SAP system can be scheduled as a step of a background job. If the ABAP program has one or more selection screens, it might be required to create the necessary input in advance in the form of a *variant*. A variant makes it possible to run an ABAP program in the background even if the program requires an input. The values stored in the variant are then used during the execution of the program. If an ABAP program has a screen output as its result, this will be saved in a spool list. You can specify an (e-mail) recipient for this spool list when defining the job. This recipient then receives the job output by e-mail after the job has been executed. You must also specify a printer for the creation of spool lists even though, as a result of background processing, there is not necessarily any direct output to a printer (this depends on the printer's access method). This may have to be explicitly started later.
- **External Command:** An external command is a call of a predefined script, a command, or a program at the operating system level. With external commands, you can "mask" operating system calls and store them in the SAP system under an elective name. You can also use the SAP authorization concept to protect the execution of an external command. This enables you to determine which users are allowed to execute which external commands (and on which target hosts and/or operating systems).

- **External Program**: An external program is any operating system command. The SAP authorization concept only specifies whether a user can call external programs or not. A more detailed authorization assignment, for example, at the level of program names, is not provided. You should use external commands for this.

Step User

With the classic definition of jobs using transaction SM36, you can assign an (execution) user to each step of the job (see the following figure). The specified user is used for the authorization checks during the execution of the step, as shown in the figure below. By default, the name of the logged-on user appears in this field, and your job is performed using your authorizations. Enter a different user name, if your job should not be performed using your own authorizations.



Hint:

To be able to do this, you must, however, have the appropriate authorization (authorization object S_BTCH_NAM) to enter names other than your own in the *User* field. An enhancement to this is authorization object S_BTCH_NA1 which controls not only the *User* field but also the *ABAP program name* field.

Figure 185: Step Users

It is useful to set up technical step users (also known as “background users”) for various work areas that have the necessary authorizations for the required activities, and that can be used by users with the same authorizations to schedule background tasks in this work area (such as for system administration). These users have separate user master records and are created specifically for use in jobs (this means it may not be possible to use them in dialog, which is just fine).

**Note:**

End users should not be allowed to use technical step users that have more authorizations than their dialog user; otherwise this could pose a security risk.

Use the *System* user type when creating step users. A dialog log on with a user of this type is not possible. In the same way, users of this type are exempt from the usual settings for the validity period of a password. The password of a system user can only be changed by user administrators using transaction SU01 (*Users → Change Password*).

**Note:**

The *Job Wizard* does not offer the option to specify a dedicated user to each step: When you use the *Job Wizard* to define a job, the name of the user who is logged on is included in the authorization check for each step automatically.

Priorities



Priorities of a background job

In addition to the planned start time, 6 additional job priorities are possible:



- Class A job with target server (group)
- Class A job without target server (group)
- Class B job with target server (group)
- Class B job without target server (group)
- Class C job with target server (group)
- Class C job without target server (group)



Figure 186: Priorities of a Background Job

As the figure above shows, background jobs can be scheduled with different priorities:

- Class A (highest priority)
- Class B (medium priority)
- Class C (normal priority)

If a job should be executed on a particular server or job server group, it is handled preferentially compared to other jobs with the same class. This preference only applies if multiple jobs of various priorities request background processing at the same time, for example, because they are scheduled for the same start time.

**Hint:**

You should ensure that a large share of all background tasks (about 90% of all jobs) are normally scheduled as a job with class C without target server specification. "System" background tasks are also executed in this way, such as the (database-related) tasks that you schedule with transaction DBACOCKPIT.

Start Options for Jobs



A job can be triggered:

-  **At a specific time ("scheduled")**
-  **When a defined event occurs ("event-driven")**



Figure 187: Start Criteria for a Job

As shown in the figure above, a background job can be triggered in one of the following ways:

- By scheduling it on a particular date at a particular time. This includes the start time "immediately" as well. If there are no free background work processes available when the job is released, the job will be scheduled with *<actual date / time>* as a start condition.
- By the occurrence of a particular event defined in the SAP system. This includes jobs that should be started after other jobs or at operation mode changes. If there are no free background work processes available when the event is triggered, the job will be scheduled with *<actual date / time>* as a start condition.

Related Information on the Concept of Background Processing

For more information on the concept of the background processing, see the online documentation for background processing which (for example) is available on the SAP Help Portal (<http://help.sap.com>) for SAP S/4HANA, under *Product Assistance*, area *Enterprise Technology* → *ABAP Platform* → *Application Server ABAP - Infrastructure* → *Components of Application Server ABAP* → *Background Processing in the SAP System*.

In addition, the following SAP Notes may be helpful:

- SAP Note [25538](#) – Configuration of maximum work process runtime - parameter *rdisp/max_wprun_time*
- SAP Note [2918906](#): Deprecation of Profile Parameter "*rdisp/max_wprun_time*"

- SAP Note [519059](#) – FAQ: *Background processing system*



LESSON SUMMARY

You should now be able to:

- Explain the basic concepts of background processing

Monitoring Background Processing



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the status flow of a job
- Use the Job Overview to monitor jobs

Status of a Job

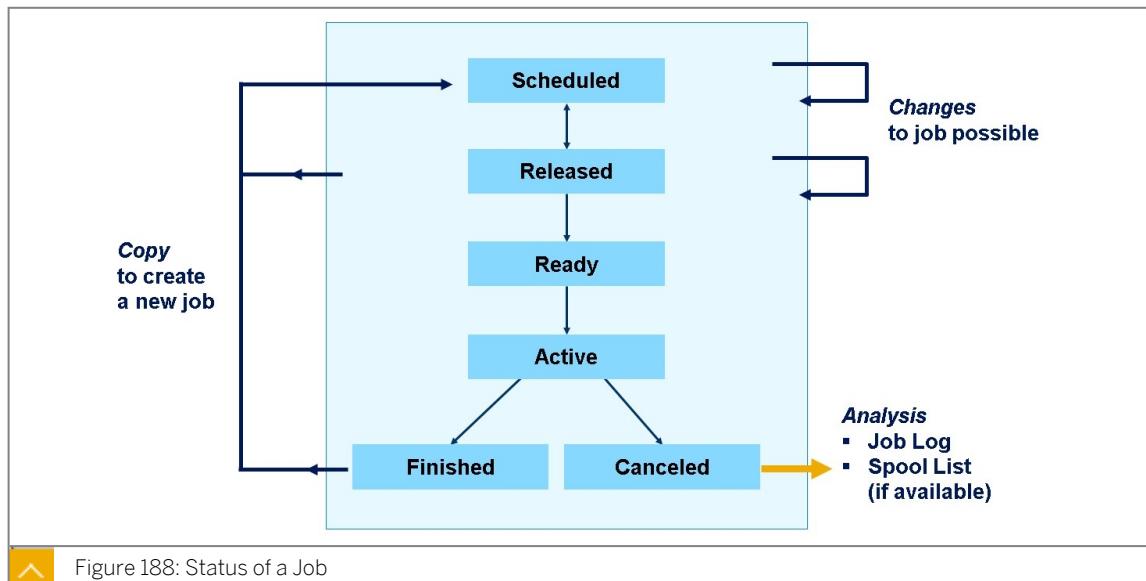


Figure 188: Status of a Job

As listed in the figure above, a job can have the following statuses:

Planned

The steps that make up the job have already been defined, but the start condition must still be defined. This job will not run without further activities on your part.

Released

The job has been completely defined, including its start condition. A job cannot be released without a start condition. Only an administrator or a user with the relevant authorizations for background processing can release a job. This ensures that unauthorized users cannot execute jobs without approval.

Ready

The start condition of a released job has been fulfilled. A job scheduler has placed the job in the request queue of background work processes.

Active

The job is currently being executed and cannot be deleted or changed.



Hint:

To capture a job, you must be logged on to the instance where the job is running.

Finished

All steps of the job have been completed successfully.

Canceled

The job has terminated. There are two ways this can happen:

- An administrator deliberately terminates the job in transaction SM37 by choosing *Job → Cancel active job*.
- A job step terminates with an error.



Note:

If a background job remains in state *Ready* or *Active* for an unexpectedly long time and you want to check whether the display is still current for the job status, you can use the following function in transaction SM37: *Job → Check status*. If necessary, the check changes the job status to the correct status.

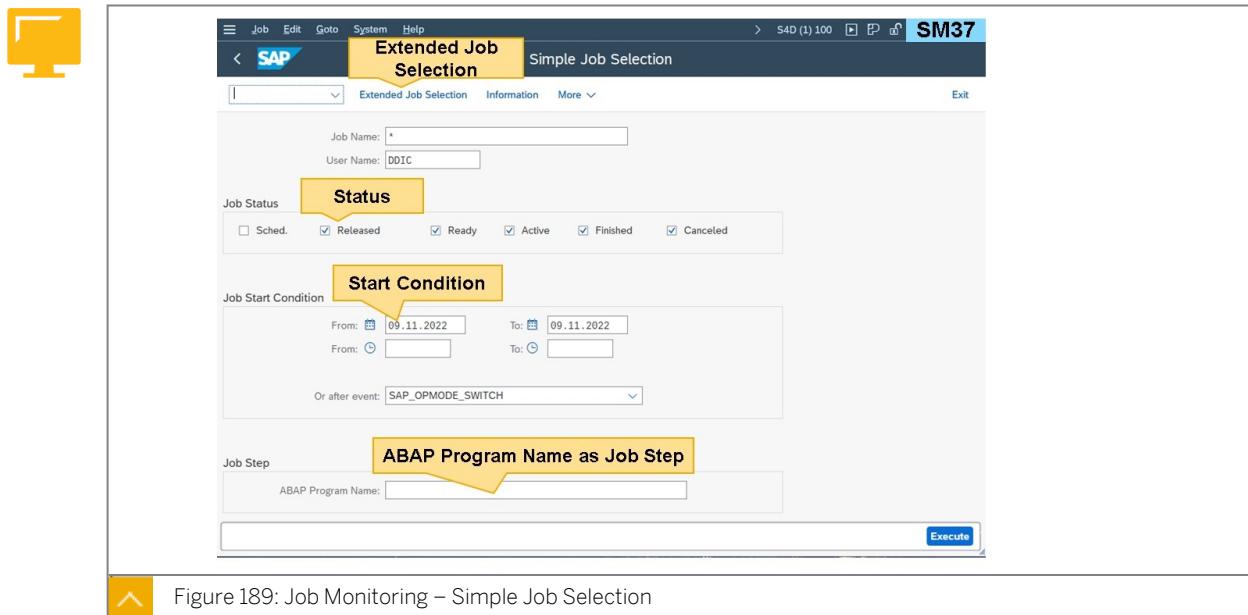
You can change a job as long as the job still has the status *Scheduled* or *Released*.

If job execution has already begun, you can monitor its processing in the *Job log* (see the following section).

If the job contains ABAP programs that generate output lists, they are saved in spool lists.

Monitoring Jobs

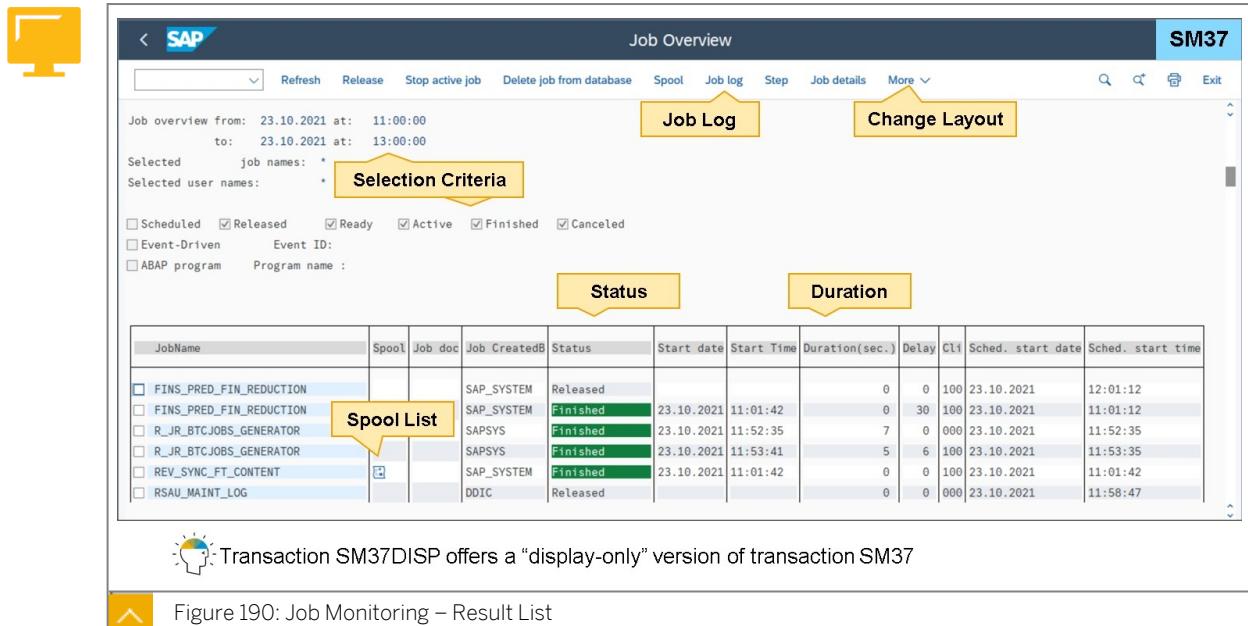
You can monitor jobs in transaction SM37 (or transaction SM37DISP which offers only display options (and no change options)).



You can select jobs using a diverse range of criteria on the initial screen of this transaction. In this way, you can display all jobs that contain a particular job step, that have a particular status, or that react to a defined event (for this, you must select a particular event or “*”).

Note:

For a job selection with more selection criteria, you can use the *Extended Job Selection* button in the application toolbar (see figure above). Any selection criteria that you have defined in the *Simple Job Selection* screen will be transferred to the *Extended Job Selection* screen.



After you have chosen *Execute*, a *Job Overview* screen (see figure above) appears that is implemented in SAP List Viewer (ALV). By choosing *Settings* from the menu, (among other

things) you can determine the columns that are displayed, and the sorting. You can set this layout as the standard layout (for the current user or for all users).



Hint:

The transaction **SM37** offers the following two ALV-based display types:

- **LIST:** Offers a quick list structure (standard setting).
- **GRID:** Enables you to change the layout easily (required for accessibility).

A batch administrator can use the program **BTC_SWITCH_LIST_GRID** to define the display type for individual users or all users.

You can navigate from the *Job Overview* shown in the figure to other job-specific views:

- The **Spool List** contains the output lists for ABAP programs (if any).
- The **Job Details** contain, among other information, the job definition, duration of the processing of the job, and the start time of the job.
- The **Job Log** stores all messages output by a background program. You can display this log to obtain information about a program that terminated with errors or to perform a detailed investigation about a background processing run.



Note:

For SAP systems belonging to SAP Business Suite, job logs are stored in the file system (as subdirectories of **DIR_GLOBAL**). For SAP S/4HANA Server systems, the job log is now no longer written to a file, but instead to a database table through a service connection. This ensures that a job log is rollback proof. You can switch off this new behaviour by maintaining an entry in table **BTCOPTIONS**. For details, see SAP Note [2360818](#) – Job log in the database and its attachments.

Outlook: Job Monitoring in SAP Solution Manager and SAP Cloud ALM

Both SAP Solution Manager and SAP Cloud ALM offer job monitoring as well:

- *Job Management* in SAP Solution Manager is a process that supports management of solution-wide background operations that automate your business processes. It allows you to request, document, analyze, and schedule background jobs in your system environment from a central location, with a standardized workflow.

As a prerequisite, your system landscape and the *Job Management* scenario must be configured in the SAP Solution Manager Configuration.

- *Job & Automation Monitoring* in SAP Cloud ALM ensures business operations without disruptions and increases the business process execution quality and performance:
 - by monitoring job executions health and detection of anomalies during job execution
 - by directly alerting both business and IT users on disruptions and exceptions
 - by providing details at job execution level to enable fact based root cause analysis
 - by reducing configuration effort by making use of historical execution information.

For more information, see SAP Support Portal (<https://support.sap.com>), area *ALM* → *SAP Solution Manager* and then (in section *Overview of ALM Processes*) enter section *Business Process Operations*.

For SAP Cloud ALM, see area *ALM* → *SAP Cloud ALM* → *Operations* and then (in section *Core Capabilities*) enter section *Job & Automation Monitoring*.



LESSON SUMMARY

You should now be able to:

- Explain the status flow of a job
- Use the Job Overview to monitor jobs

Setting Up the Background Processing Environment



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Evaluate the need for reserving work processes for high-priority jobs
- Create Job Server Groups and explain their use
- Create RFC groups for the parallel processing of jobs
- Schedule housekeeping jobs
- Perform simple tests to check the consistency of the background processing environment

Reservation of Work Processes for Class A Jobs

In normal operation, every background work process processes jobs of every priority.

You can, however, reserve some of the configured background work processes as desired for high priority jobs; that is, class A jobs. The reservation of work processes for class A jobs does not reserve any **particular** work processes. Rather, it ensures that a particular number of background work processes are kept free for class A jobs whenever possible.

Jobs of classes B and C can only be started if the defined number of work processes remains free for possible class A jobs, as shown in the figure below.

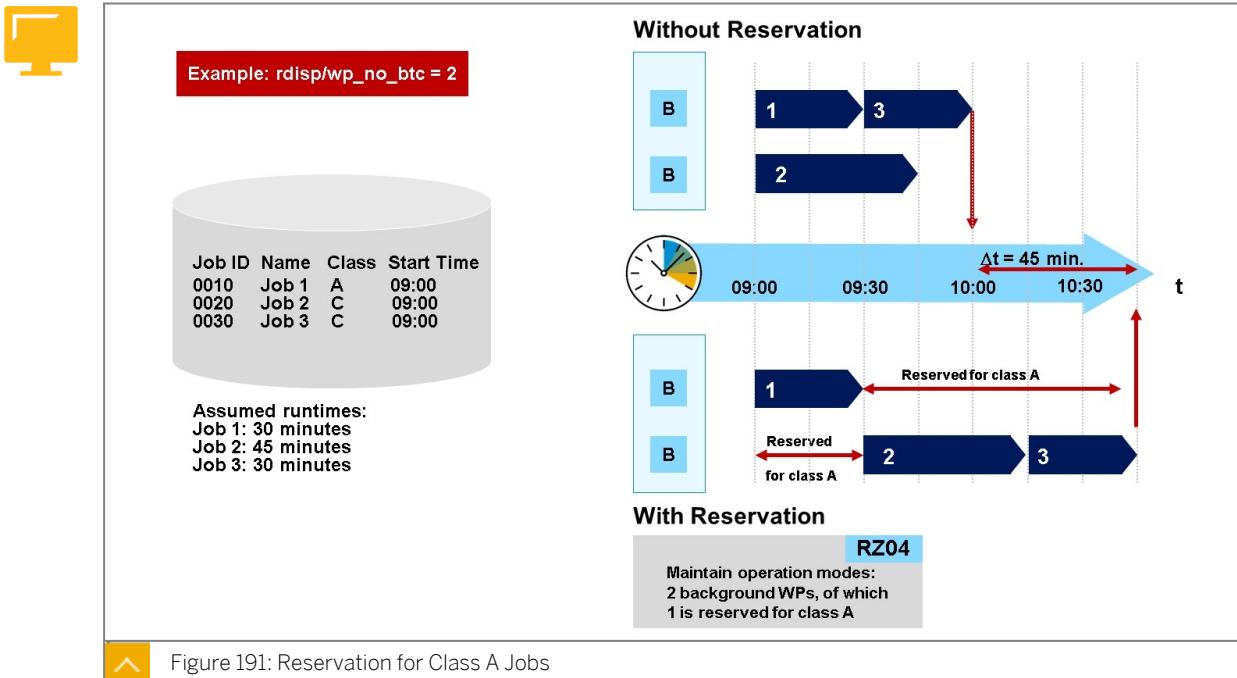


Figure 191: Reservation for Class A Jobs

This option allows a choice between two prioritization strategies for your background jobs:

- *Critical track (class A) and two-level priority (classes B and C)*: If you have mission-critical jobs whose immediate execution must always be ensured, you can set up class A as your “critical jobs” category. In this case, reserve a work process for class A jobs and assign class A only to these mission-critical jobs. In this scenario, class B and C jobs are very unlikely to block a class A job.
- *Three-level priority for allocation of free work processes*: You can also decide to run with a straight three-level priority scheme for allocating free work processes. In this case, reserve no work processes for class A. If the class A workload is small, or bottlenecks rarely occur in background processing, that is, at least one background work process is usually free, the reservation of a work process for class A probably provides no advantages. In this case, reservation will simply mean that a work process is seldom used.

To set the number of reserved background work processes for class A, you can define an operation mode in transaction RZ04 (CCMS: *Edit Operation Modes and Instances*) and maintain the work process distribution for this operation mode. When doing so, you have the option of reserving work processes.



Hint:

If you want to use the function for reserving background processes for class A jobs, it is usually enough to define one background work process for class A jobs in each system. We recommend that you do not reserve more than one background work process for processing class A jobs for each application server.

Target Server and Job Server Groups

Only instances with background work processes or a job server group can be used for the targeted scheduling of a background task.

A **job server group** contains one or more instances with available background work processes. You can use groups of this type in the same way as logon groups for dialog users. You can also process background tasks from different applications on selected instances.

Job server groups can be used to separate jobs from application servers being used by dialog users. In this case, the job server group used by a job would contain application servers other than those contained in the logon groups (transaction **SMLG**) for dialog users.

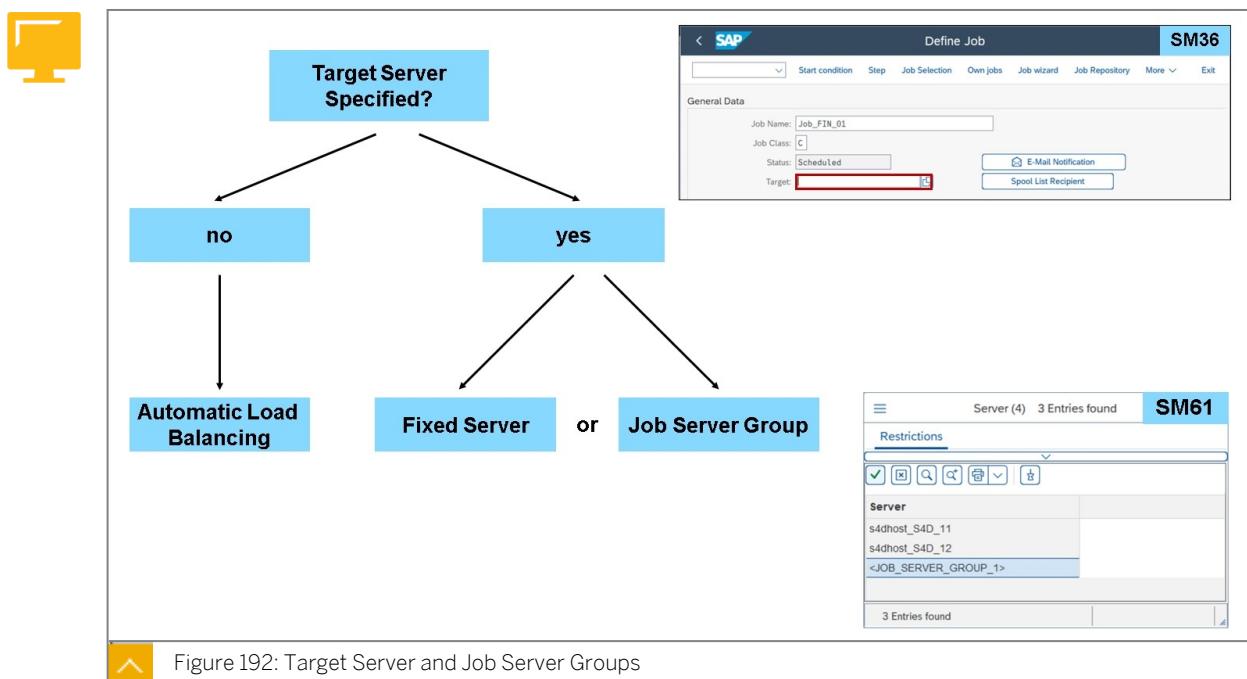


Figure 192: Target Server and Job Server Groups

To set up a job server group, start transaction **SM61** (*Display control object list*) and then choose *Job Server Groups*. You can then define your job server group and assign your instances (see the figure above).

For more information, see SAP Note [612838](#) – *Administration of job server groups*.



Hint:

There is a default job server group that you can create and use as described in SAP Note [786412](#) – *Determining execution server of jobs w/o target server*.



Note:

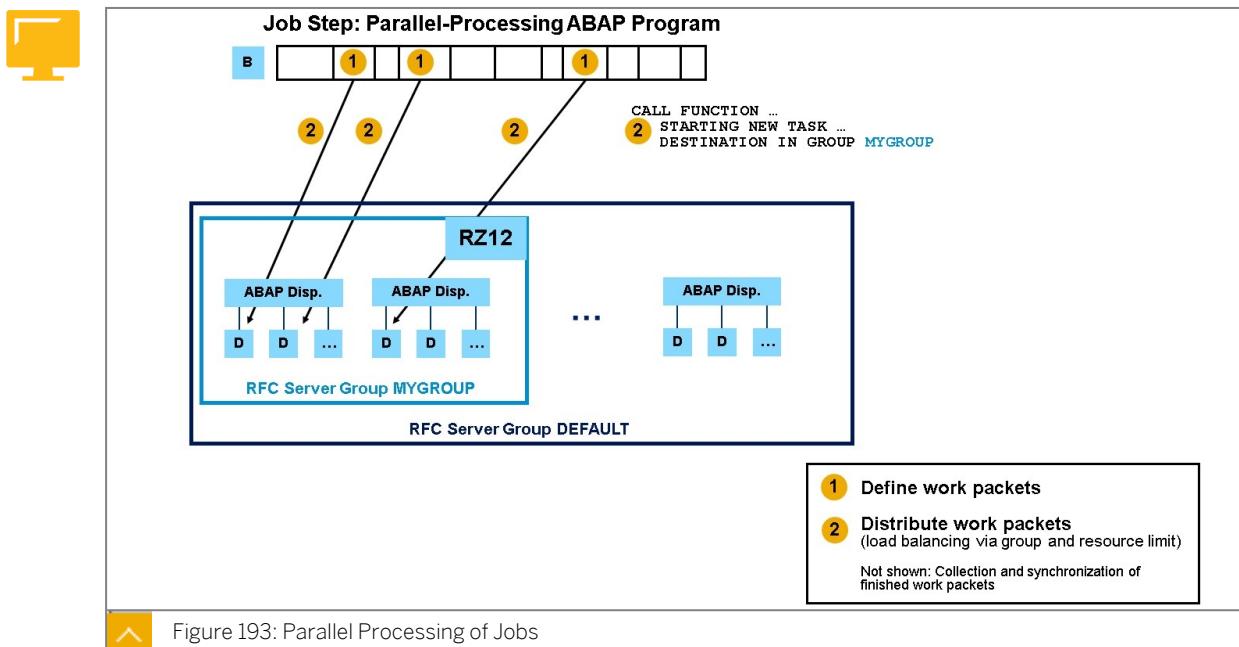
In transaction **SM61**, you can also perform health checks for application servers (leading to transaction **SM65**) and (by choosing *Modify control object list entry* in the application toolbar) you can change the trace level for background control objects.

Parallel Processing of Jobs

For some long running SAP reports, the nights are just getting too short. Especially at customers with large volumes of data, some reports that customarily run in the background processing system (such as material planning runs) may have run times of many hours.

These jobs can exceed the “night-time” that is available, particularly if the dialog users are distributed across multiple time zones.

SAP offers a solution to the “short nights” problem: parallel-processed background jobs. Long-running reports can now implement parallel processing where the work to be done can be distributed into smaller packages for the available work processes. The results can then be collected and synchronized.



In parallel processing, a job step is started as usual in a background processing work process. With the help of the ABAP command `CALL FUNCTION <Function Module> STARTING NEW TASK <Task> DESTINATION IN GROUP <Group>`, a program that runs in a job step can be programmed to have portions of the data to be processed run in parallel in other work processes (see the figure above). The calling program is continued as soon as the remotely called function has been started in the target system, without waiting for its processing to be finished.



Note:

While the job itself runs in a background process, the parallel processing tasks that it starts run in dialog work processes. Such dialog work processes may be located on any SAP server.

**Note:**

By default, a parallel-processed job uses all qualified servers in an SAP System according to automatic resource-allocation rules. However, by defining RFC groups, you can control which servers can be used for parallel-processed jobs. An RFC group specifies the set of allowed servers for a particular parallel-processed job.

By default, the group of servers eligible to be used for parallel processing is all qualifying servers in your SAP System (`CALL FUNCTION <Function Module> STARTING NEW TASK` with the argument `DESTINATION IN GROUP DEFAULT`). However, by defining RFC groups, you can control which servers can be used for parallel-processed jobs.

Definition of RFC Groups for Parallel Processing Jobs

An RFC group specifies the set of allowed servers for a particular parallel-processed job.

To define an RFC server group, use transaction RZ12. For RFC server groups, the descriptions of the individual parameters for resource allocation of the can be found in the F1 help in the system. Limitations concerning the SAP system parameters are explained in SAP Note [99284](#) – *RFC exception: RESOURCE_FAILURE*. Also, see SAP Note [2001276](#) – *Changed configuration as of 7.40 SP2* for changes as of AS ABAP 7.40. It is recommended to deviate from the predefined SAP standard values for the parameter in very exceptional situations only (and must be tested with the relevant load on the system).

**Note:**

When a server belongs to more than one group, jobs that use the group will compete for free work processes on the shared server(s). You must ensure that the group you create is also specified in the ABAP program that will use the group.

Standard Jobs (non SAP S/4HANA)

As a system administrator, you are going to work with a lot of batch jobs in your business, and want to know what you have to keep the batch functionality in an optimal state.

Standard jobs are background jobs that should run regularly in a production SAP system. These jobs mainly perform certain clean-up tasks in the system, such as the deletion of obsolete spool requests.

**Note:**

For example, the batch system is an operative execution environment for jobs, therefore SAP does not reorganize the jobs by default but it keeps online the earlier executed jobs in the database, too. As a consequence, it is very important to work with an optimal reorganization strategy in your system in order not to let grow the table sizes for jobs in the system (tables `TBCO`, `TBTCP`, ...) too big. This is why it is important to execute corresponding housekeeping jobs in your system regularly (for example `SAP_REORG_JOBS`).



- Standard jobs are to run regularly in a productive non-SAP S/4HANA Server system
- They usually deal with specific 'clean-up' tasks in the system, for example, deleting obsolete spool requests

The screenshot shows the SAP transaction SM36 interface for scheduling standard jobs. The main area displays a table of standard jobs with columns for Co. Background Job Name, SchedStart, Start date, Start time, Status, and Job ID. Below the table, there's a form for scheduling a standard job, including fields for SAP Component, Job name, Job info, Report Name, Variant, Server Name, and Step User. A note 'Basis Components' highlights the SAP Component column, and 'F4 help' points to the Job name field. In the bottom right, a note 'Schedule Jobs manually according to SAP Notes' is displayed.

Figure 194: Scheduling Standard Jobs (non SAP S/4HANA)

In AS ABAP-based SAP systems that are **not** SAP S/4HANA Server systems, you can access a selection of important standard jobs that you can schedule, monitor, and edit in the transaction SM36. To do so, you should choose *Standard jobs* in the application function bar. As the figure *Scheduling Standard Jobs* displays, you will see the table with all standard jobs (table REORGJOBS). If there are no standard jobs scheduled, the table will appear empty.

- If you want to schedule all standard jobs, choose *Default Scheduling* and provide a system user under which the jobs should be executed. All standard jobs will be scheduled with the specified variant and period. If other jobs exist that execute one of the programs of the standard jobs, you will see an information message.
- To schedule individual jobs, choose the selected job with the input help and specify an execution period.
- To define an additional standard job that is not yet available in the selection (table REORGJOBS), choose *Predefine new job*.



Note:

SAP recommends scheduling the jobs in client 000. However, some of the jobs specified work with client-specific objects (for example, jobs). For more information, see SAP Note [16083: Standard jobs, reorganization jobs](#).

Due to changes to standard jobs over time, schedule the standard jobs manually according to the SAP Notes listed below which also provide more information on standard jobs:

- SAP Note [16083: Standard jobs, reorganization jobs](#)
- SAP Note [1034532: Changes for standard jobs](#)
- SAP Note [1411877: New Standard Jobs](#)
- SAP Note [1440439: New Standard Jobs \(2\)](#)

Technical Job Repository (SAP S/4HANA)

In SAP S/4HANA Server systems, the *Technical Job Repository* serves the purpose to automatically schedule so-called technical background jobs (both in the “ADMIN” and “BUSINESS” clients) which must be running in order to keep the system “healthy”. It replaces the “Standard Job” functions in transaction SM36 and consists of a design-time component and a runtime component.



- In **SAP S/4HANA Server** systems, the Technical Job Repository **automatically** schedules technical background jobs

Not in Scope: the background job is scheduled only under certain circumstances (for example, only when a special application is active) which are not met

Not relevant: The job shall not run in the current client

Job Definition Status					
JD Status	JD Status Reason	Package	Background Job Name	Job Status	Scope Dependent
Not In Scope	SP0PQ	SAP_PRINQUEUE_FROM_SPOOL	SAP_PRINQUEUE_FROM_SPOOL	Not In Scope	☒
Not In Scope	SP0PQ	SAP_PRINQUEUE_NOTIFY	SAP_PRINQUEUE_NOTIFY	Not In Scope	☒
Not In Scope	SP0PQ	SAP_PRINQUEUE_REORG	SAP_PRINQUEUE_REORG	Not In Scope	☒
Not In Scope	PS_PPM_INTEGRATION_UTILS	SAP_PS_CREATE_PPM_PROJ	SAP_PS_CREATE_PPM_PROJ	Not In Scope	☒
Not In Scope	PS_BILLING_PROPOSALS	SAP_PS_FILL_EXTRACT_TABLES	SAP_PS_FILL_EXTRACT_TABLES	Not In Scope	☒
Not In Scope	CATF	SAP_PS_TIMESHEET_TRANSFER	SAP_PS_TIMESHEET_TRANSFER	Not In Scope	☒
Not In Scope	SREAD_ACCESS_LOG_TRANSFER	SAP_RAL_REPLICATION_OP	SAP_RAL_REPLICATION_OP	Not In Scope	☒
Not Relevant	SABP_RABAX	SAP_REORG_ABADUMPS	SAP_REORG_ABADUMPS	Not Relevant	☒
Not Relevant	SBTI	SAP_REORG_APPLJOBS	SAP_REORG_APPLJOBS	Not Relevant	☒
Not Relevant	SZAL	SAP_REORG_APPLLOG	SAP_REORG_APPLLOG	Not Relevant	☒
Not Relevant	SBDC	SAP_REORG_BATCHINPUT	SAP_REORG_BATCHINPUT	Not Relevant	☒
Not Relevant	SICM_WD_SUPPORT_INFO	SAP_REORG_ICM_WD_SUPPORT_INFO	SAP_REORG_ICM_WD_SUPPORT_INFO	Not Relevant	☒
Not Relevant	SBTI	SAP_REORG_JOBS	SAP_REORG_JOBS	Not Relevant	☒
Not Relevant	SBTC	SAP_REORG_ORPHANED_JOBLOGS	SAP_REORG_ORPHANED_JOBLOGS	Not Relevant	☒
Not Relevant	STMS	SAP_REORG_ORPHANED_TEMSE_FILES	SAP_REORG_ORPHANED_TEMSE_FILES	Not Relevant	☒
Not Relevant	SABP_LIST	SAP_REORG_PRIPARAMS	SAP_REORG_PRIPARAMS	Not Relevant	☒
Not Relevant	SPOO	SAP REORG SPOOL	SAP REORG SPOOL	Not Relevant	☒
Not Relevant	SXMI	SAP REORG XMLLOG	SAP REORG XMLLOG	Not Relevant	☒
Not In Scope	WRPL	SAP_RFMS_REPLSLMENT_RUN_REORG	SAP_RFMS_REPLSLMENT_RUN_REORG	Not In Scope	☒

Figure 195: Job Repository for SAP S/4HANA Server

- The **designtime component** allows SAP to create *Technical Job Definitions* (in short: *Job Definitions*) which are basically templates for recurring background jobs.
- The **runtime component** is automatically active in every SAP S/4HANA Server system. This can be checked with the help of program *R_JR_UTIL_1* with default parameter settings.

The runtime component is responsible for actual scheduling (or unscheduling) of background jobs as mandated by the set of delivered *Technical Job Definitions*. The runtime component is triggered by an hourly action which occurs automatically in an SAP S/4HANA system.



Hint:

The profile parameter responsible for this is *rdisp/job_repo_activate_time* with default value *60M*. The hourly *Job Generation* action stores its last execution time in table *BTCCTL*. So it is possible to receive the last *Job Repository* job generation time by displaying table *BTCCTL* with transaction *SE16* and selecting the entry with key value *CTLOBJ = JOBDSDL*.

Almost every job definition contains the value “(DEFAULT)” in field *User Name*. This is actually a placeholder for the execution user that the runtime component will use when scheduling the background job. As default behaviour, the *Technical Job Repository* will use *SAP_SYSTEM* as batch job step user if this user is available in the current client. If no such user exists, user *DDIC* will be used as batch job step user as a fallback solution.

The *job definition status* (see figure above) differs from the background job status. A job definition can be consistent, but nevertheless the job is not scheduled, for example because it should not run in the current client.

The *job definition status reason* explains why no job is scheduled for a job definition, for example

- if the user of the job definition does not exist in the current client (status reason “BTC User Error”) or



Note:

As of release SAP_BASIS 7.56, if you select one job definition from the list and choose button *All clients* in the application toolbar, you will get an overview of the selected job definition details in all clients.

- if the job will not run in the current client (status reason “Not Relevant”) or
- if the job definition is scope dependent (see column *Scope Dependent*) but the scope conditions are not fulfilled (status reason “Not In Scope”).

Transaction `SJOBREPO_STEPUSER` allows you both to create a suitable technical user for *Job Repository* for the current client and to assign a technical user to the current client that is to be used by *Job Repository*. With the help of this transaction you can also display currently assigned *Job Repository* technical users for all existing clients.

For more information about the *Technical Job Repository*, see SAP Notes [2190119](#): *Background information about S/4HANA technical job repository* and [3236399](#) – *FAQ - Technical Job Repository (SJOBREPO)*.

Analyzing the Configuration of Background Processing

Certain requirements have to be met before the background processing can work. For example, profile parameters and authorizations must be set. A simple function is available for checking the consistency of the background processing environment: the *analysis tool for background processing* (transaction `SM65`). This tool is displayed in the figure below.

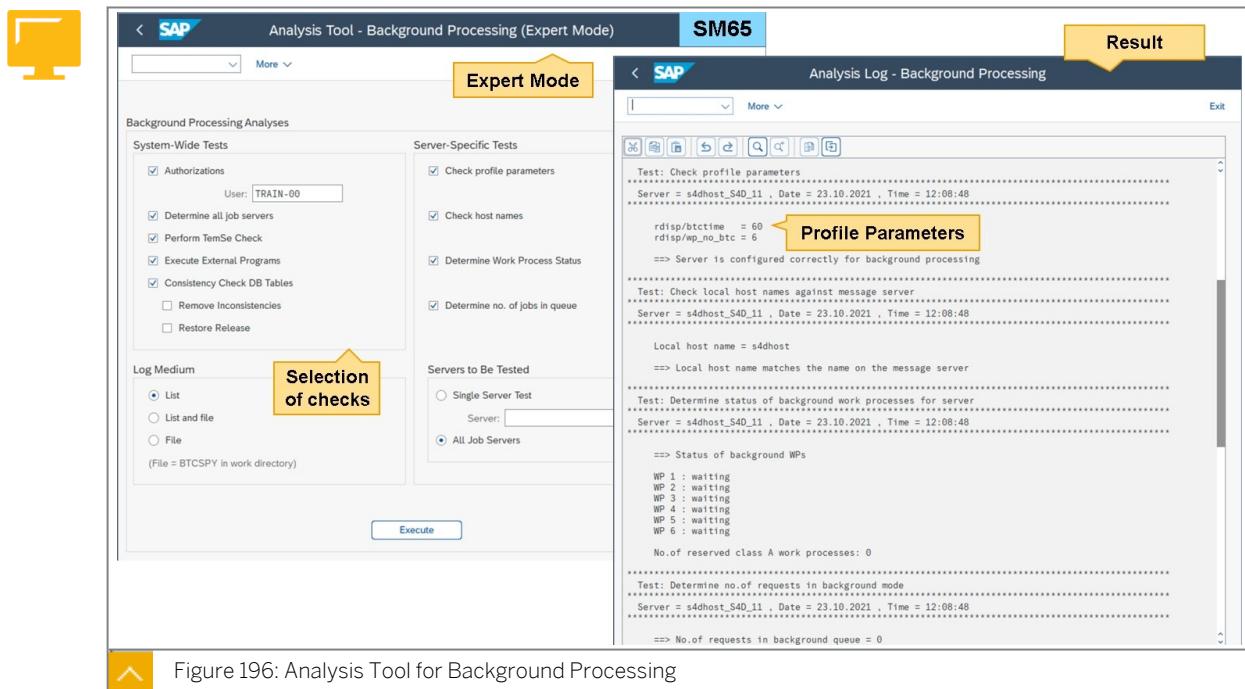


Figure 196: Analysis Tool for Background Processing

There are two test types available:

- *Basic tests* (initial screen): Here you can check whether the profile parameters are correctly set and whether the user ID SAPCPIC exists (required for starting external programs). You can perform this check both for a particular background server and for all servers. It is required for starting external programs.
- *Expert tests* (from the start screen, choose Goto → Additional Tests): In addition to the basic tests described above, you can use this option to also check the
 - user authorization for background processing
 - name of all job servers
 - function and status of the TemSe subsystem, which is required for writing the job logs
 - consistency of the database tables.

The background processing stores job data in multiple database tables. These tables can be checked for consistency. This test is especially important if problems occur with the database and you need to determine whether all job data is still available.



Note:

In addition, the program *BTC_OPTIONS_SET* offers you versatile options for background processing in your system. For each option in this report, the SAP Note that provides additional information for the relevant function is identified. For example, you can allow to specify a target server or a target server group for execution in the background using transaction SA38 and from an application as well.



LESSON SUMMARY

You should now be able to:

- Evaluate the need for reserving work processes for high-priority jobs
- Create Job Server Groups and explain their use
- Create RFC groups for the parallel processing of jobs
- Schedule housekeeping jobs
- Perform simple tests to check the consistency of the background processing environment

Unit 9

Lesson 4

Scheduling Background Jobs



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Create a time-based job
- Create an event and a job based on that event
- Create a job that contains external commands or external programs
- List options to extend the standard background system

Time-Based Scheduling



Immediate

Once or periodically

Exceptions can be defined for periodic scheduling



At a Specific Date/Time

Once or periodically

Exceptions can be defined for periodic scheduling

`rdisp/btctime`

Mo	Tu	We	Th	Fr	Sa	Su
30	31	1	2	3	4	5
Mo 6	Tu 7	We 8	Th 9	Fr 10	Sa 11	Su 12

Example: 4th working day of the month

One Day Relative to Start/End of Month

- Once or periodically

Figure 197: Time-Based Start of a Job

A job can be started in a time-based or event-based manner. As the figure above shows, a time-based start allows you to choose between the following options:

- The job should be executed immediately.
- The job should be executed at a particular date/time.
- The job should be executed on a particular work day.

You can choose to make all time-based start conditions for jobs recurrent. This means that the job is executed again after a defined period of time has elapsed. You can specify exceptions (such as postponement to the following work day in the case of holidays in the factory calendar).

The job is started at the specified date and time, in accordance with the job priority and availability of background work processes.

You can also specify a time period during which the job should be executed. To do this, you specify a time after which the job should no longer be executed (*No Start After*). With this function, you prevent periodic jobs being executed at an undesirable time, among other things. For example, suppose your company has a reorganization job that should only be executed at night, and the job becomes delayed. With a start time window, you can avoid this job being executed during the day, when the dialog users are active and there are fewer system resources available.



Note:

If you use periodic job scheduling, it is not possible to define an end to the periodic execution – for example, a scheduling condition such as “every day to the end of this year” is currently not supported.

Load Balancing

As the figure below shows, the profile parameter *rdisp/btctime* specifies the time period in which the time-driven job scheduler is active. To avoid the time-driven job scheduler, you can execute jobs with the start condition *immediate*. For jobs with an immediate condition, the dialog work process of the scheduling user performs the job scheduling. However, if no free resources are found, the job is scheduled in a time-based way. The scheduled start time then corresponds to the date and time at which it should have started.

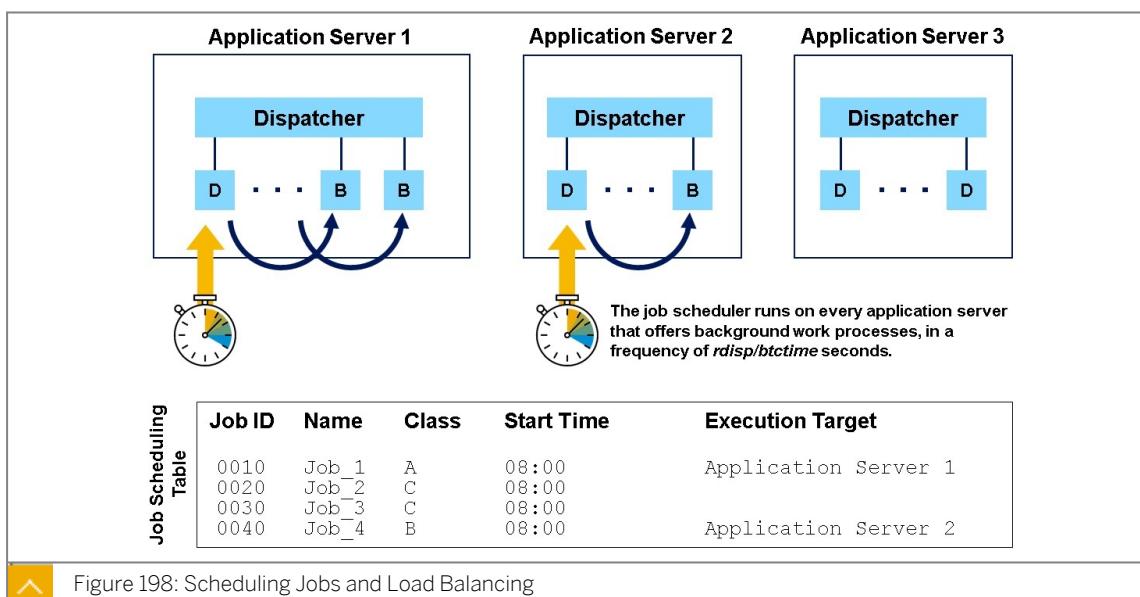


Figure 198: Scheduling Jobs and Load Balancing

Background work processes can be configured on every application server of the SAP system using the profile parameter *rdisp/wp_no_btc* or, for example, when configuring operation modes.

The number of background work processes required in the SAP system depends on the number of tasks to be performed in the background. If the *Transport Management System* is used to perform imports of transport requests, there must be at least two background work processes in the system.

The combination of *job ID* and *job name* defines the job uniquely in the system.

On every SAP application server on which background work processes are defined, the time-driven job scheduler runs every `rdisp/btctime` seconds (default value: 60 seconds). This is an ABAP program (`SAPMSSY2`, an auto ABAP program) that runs automatically in a dialog work process. While the job scheduler is supposed to run every 60 seconds (by default), these runs usually do not take place exactly on the minute. Instead, they could run the 42nd second of the minute on one application server and the 9th second of the minute on another application server. You cannot influence this start time. It changes when the application server is restarted, at the latest, and depends on when certain system activities are performed for the first time after an instance restart – and then every 60 seconds from that point.



Hint:

The job scheduler is also started every time a job ends, which can increase the throughput of background processing significantly (depending on how many jobs there are).

SAP Note [923228](#): *Background job scheduler: Use of processes that have become free* describes this changed behavior and its requirements.

The time-driven job scheduler checks the job scheduling table in the database for jobs that are waiting for processing. These jobs are transferred to free background work processes on the SAP application server, in accordance with their priority and execution target.

- Jobs that are not assigned any particular execution target can be executed with any free background work process. This means that the workload is automatically distributed between the SAP instances.
- If a job is explicitly assigned an execution target (such as a selected instance or a group of instances), the special properties of the execution target can be used. For example, you can ensure that a job is performed on a particular operating system, or that the job is executed by a background work process that is running on the same host as the database. This means, however, that you lose the advantages of automatic load balancing.

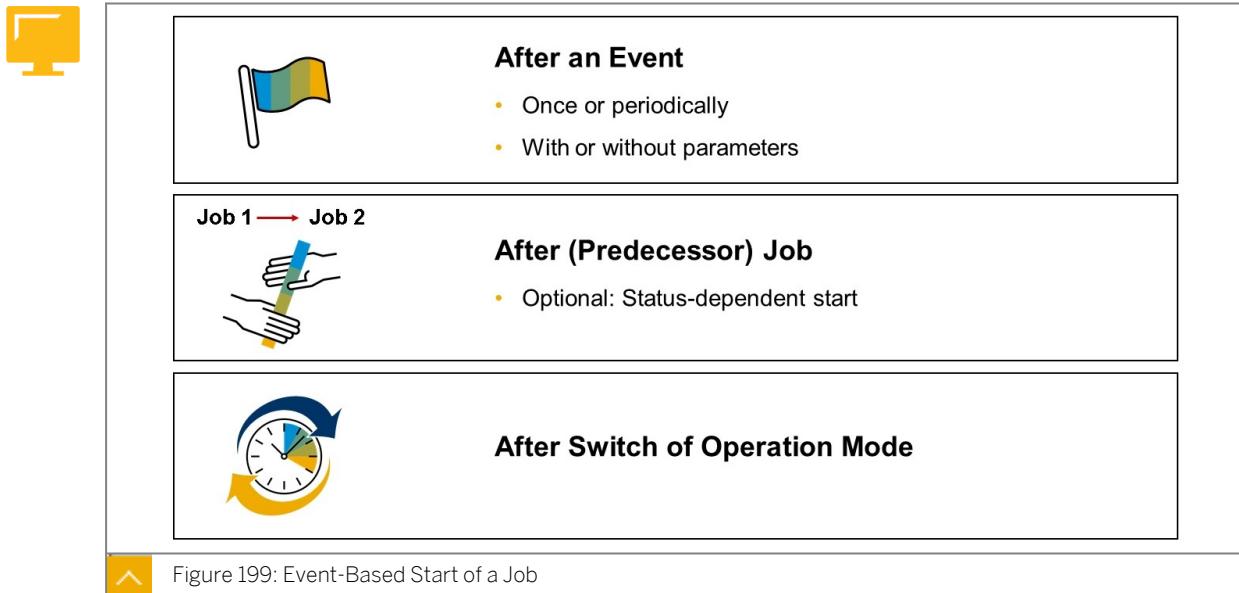


Note:

For more information, see SAP Note [24092](#) – *Distribution of background jobs on application server*.

Event-Based Scheduling

An **event** is a signal to the background processing system that a particular status has been achieved in the SAP system. The background processing system receives events and then starts all jobs that are linked to this event.



As listed in the figure above, event-based jobs can be scheduled with one of the following three start conditions:

After event

The job starts after a defined event is received by the SAP system.

At operation mode

With this option, you can link a job to the activation of an operation mode when scheduling the job.

After job

With this option, you can create simple job chains where the execution of the successor job can be made dependent on the status of the predecessor job.



Hint:

The profile parameter `rdisp/btcname` is no longer evaluated and is therefore no longer relevant. For more information, see SAP Note [1035355: Background processing events: Handling of target server](#).

Events

You can trigger events with the function *Background Events: Overview and Administration* (transaction SM64). There is a distinction between **system events** and **user events**. System events are events predefined by SAP that you cannot modify and, if possible, you should not trigger manually.

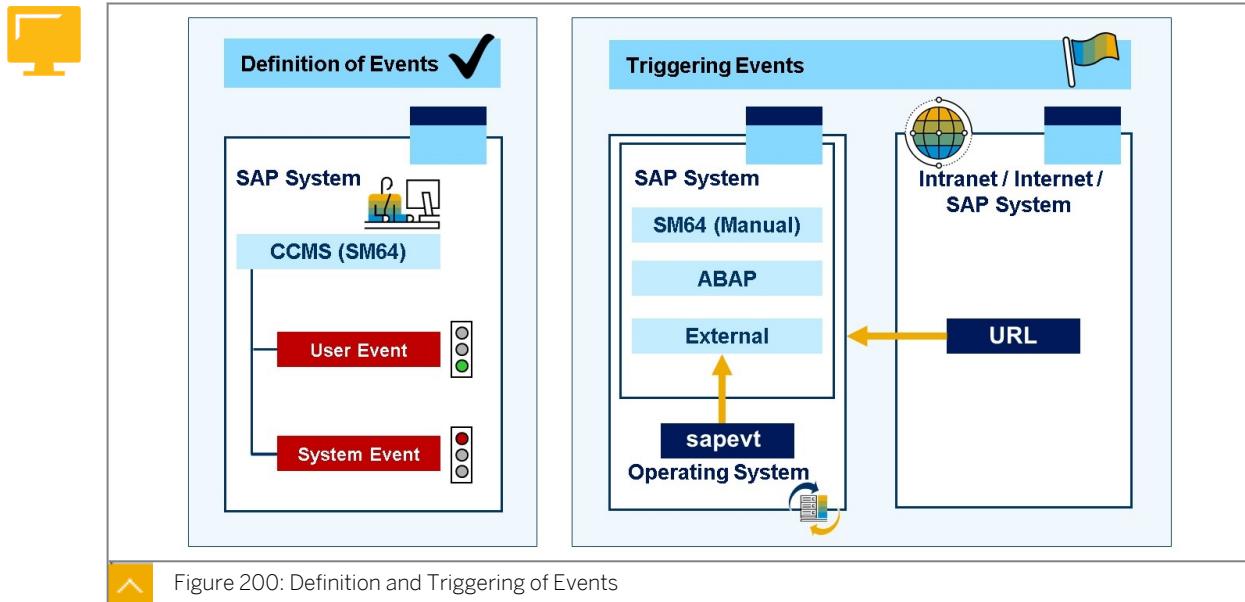


Figure 200: Definition and Triggering of Events

As the figure above shows, events can be triggered in several ways:

- Manually, using transaction SM64.
- By the ABAP program *BTC_EVENT_RAISE* (see SAP Note [919458](#) – Program '*BTC_EVENT_RAISE*') or by calling the function module *BP_EVENT_RAISE*.
- At the operating system level, through program *sapevt*.
- Using HTTP(S) by calling the ICF service */default_host/sap/bc/batch/event_raise*.



Note:

There are even different options (not included in the list above), such as to use the *RAISE* method of class *CL_BATCH_EVENT*.

When an event is triggered, a parameter can also be provided. In this way, you can define jobs that wait for the occurrence of the event together with the specified parameter.

You can display a history of events in the *Event History* (transaction SM62).



Note:

For error analysis, SAP Note [1645120](#): *Trigger of a background event is unknown* may help.



```

wdflbmt0102:s4dadm> sapevt -help
wdflbmt0102:s4dadm> more dev_evt

Sat Nov 14 16:25:07 2020
Trace File of External Event Raiser (Level=2, Append=0)

***** Complete Call: *****
sapevt
-help
***** End of Call *****

DIR_GLOBAL: /usr/sap//SYS/global
Usage: sapevt <parameters>
<Parameters> are some options from the followings:
  {<EventID> | event=<EventID>} [{-p <EventParam>} | 
param=<EventParam>]
  [-t[0|1|2][a]]
  [-v]
  {[name=<SystemName>] [msserv=<MsServ>] [mshost=<MsHost>]
[pf=<Profile>]}
  {[timeout=<TimeOut>]}
  [-? | /? | -help | /help | -VERSION]
rdisp/mshost read from profile:
EventID: -HELP
*** ERROR ***: Cannot determine mshost ←
because sapevt was called without instance profile

When sapevt -help is called, a file called dev_evt is generated, which contains the trace from sapevt - in this example the help manual for sapevt

```

Figure 201: Syntax for the sapevt Program

sapevt is a program located in kernel directory which is used to trigger an event from a program running outside the SAP system. The syntax for the *sapevt* program is shown in the figure above.

Example: **sapevt myevent name=S4D pf=/usr/sap/S4D/SYS/profile/ S4D_D11_s4dhost.**

sapevt writes its output in a trace file *dev_evt* in the directory from where *sapevt* was called.



Note:

SAP Note [802172](#): *sapevt (Version 4): Release-independent version* describes the individual parameters in detail. In addition, SAP Note [826779](#) – *sapevt (Version 5): Security settings in the Msg.* might be helpful if the *sapevt* program must use a different port to communicate with the message server due to increased security requirements.

To be able to react to external events, the SAP system must be active. Otherwise an event triggered by an external program is lost.



Hint:

If the name of the event contains a blank character, you must specify the event in quotation marks when you call *sapevt*.

For example: **sapevt "my event" name=S4D pf=/usr/sap/S4D/SYS/ profile/S4D_D11_s4dhost.**



Note:

If you want to call *sapevt* from a remote host, refer to SAP Note [374379](#): *Triggering SAPEVT from a remote host.*

The figure *Examples of sapevt Calls* shows further details.



Example for a valid call of **sapevt**, triggering event **EVENT1** with parameter **PAR1** and providing the **instance profile**

```
wdflbmt0102:s4dadm> sapevt EVENT1 -p PAR1
pf=/usr/sap/S4D/SYS/profile/S4D_D11_s4dhost

Entering ssfPkiGetInstancePSE
ssfAuxGetInstancePSE: Profile Parameter SECUDIR = /usr/sap/S4D/D11/sec
ssfAuxGetInstancePSE: Sec directory /usr/sap/S4D/D11/sec
ssfAuxGetInstancePSE: Get SYSTEM_PKI/PIN from SecStoreFS
ssfAuxGetInstancePSE: Instance PSE
/usr/sap/S4D/D11/sec/sap_system_pki_instance.pse
ssfPkiGetInstancePSE: Instance PSE - last changed on disk Tue Feb 11 11:31:15
2020
ssfPkiGetInstancePSE: Instance PSE - last changed in memory Thu Jan 1 00:00:00
1970
ssfAuxCreateMemoryPSE: Memory PSE container created
#_MemPSE_#650704630276126000000001
ssfAuxCreateMemoryPSE: Instance PSE loaded in container
#_MemPSE_#650704630276126000000001
ssfAuxCreateMemoryPSE: Reset PIN for container
#_MemPSE_#650704630276126000000001
Exit ssfPkiGetInstancePSE: Instance PSE #_MemPSE_#650704630276126000000001
wdflbmt0102:s4dadm>
```



- This call is successful when it is executed by the user **<sid>adm** on the host where the application server is running.
- The event name is not case-sensitive – but the event parameter is.



Figure 202: Examples of sapevt Calls

Due to unavoidable changes as of Kernel 7.40 in the internal SAP message format, there is a break in **sapevt** in kernel release 7.40 when used on a cross-release basis. In addition, since **sapevt** communicates unauthorizely with the message server, the program may no longer function in general if secure message server communication is active.



Note:

When **sapevt** is called on the application server with the user **<sid>adm** the parameter **pf=<instance profile>** must be provided. This behavior depends on the type of the operating system and the release and patch level of the kernel of the SAP system. Newer SAP systems on Windows require this input, on Linux it might be that an error message will be returned, but the event might have been raised successfully.

So if in doubt, provide **pf=<instance profile>**.

SAP provides two more options to trigger a batch event externally.

- ICF service **/default_host/sap/bc/batch/event_raise** (see the figure below and SAP Note [2001336](#) – *Downport: ICF service for triggering batch event and its attached PDF*).
- RFC client **sapevt_rfc** on operating system level which calls function module **BP_REMOTE_EVENT_RAISE** (see SAP Notes [2015788](#) – *Program "sapevt_rfc"* and [2014139](#) – *Function BP_REMOTE_EVENT_RAISE*).



- The ICF service `/default_host/sap/bc/batch/event_raise` can be called by an HTTP client and triggers a batch event in the SAP system
- Syntax for the URL: `http(s)://<host name>:<port>/sap/bc/batch/event_raise?sap-client=<client>&sap-user=<user ID>&sap-password=<password>&eventid=<event to be triggered>`

The screenshot shows the SAP Define Services interface with the SICF tab selected. In the Filter Details section, the ServiceName is set to 'EVENT_RAISE'. Below this, the Virtual Hosts/Services tree shows 'default_host' expanded, with 'sap' and 'bc' children, and 'batch' expanded further to show 'event_raise'. To the right of the tree, there is documentation for 'VIRTUAL DEFAULT HOST' and 'BASIS TREE (BASIS FUNCTIONS)'.



Parts of the URL shown in green are optional.

Figure 203: Triggering Events Using ICF Service event_raise

Regarding the ICF service `/default_host/sap/bc/batch/event_raise`, you have several options:

- A user and password is stored in the section *Logon Data* of the ICF service. Then it is sufficient to pass to the http client a parameter string like the following to trigger the event `SAP_TEST: http://<host name>:<ICM-Port>/sap/bc/batch/event_raise?sap-client=<client>&eventid=SAP_TEST`. This call is basically equivalent to a call of the old program `sapevt`, because it does not require authentication.
- Specify the logon data in the parameter string (at least for testing). This would result in `http://<host name>:<ICM-Port>t/sap/bc/batch/event_raise?sap-client=<client>&sap-user=<user name>&sap-password=<password>&eventid=SAP_TEST`.
- It is also possible to logon with X.509 certificates.

Additional Information

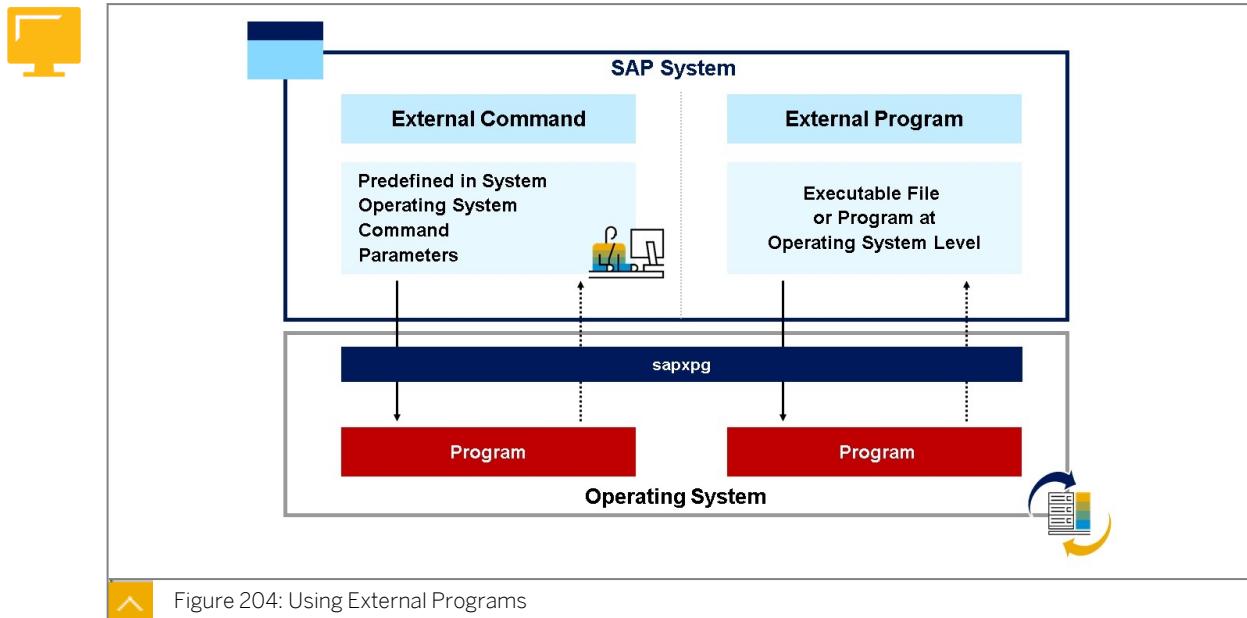
The following SAP Notes might be helpful:

- SAP Note [802172](#): *sapevt (Version 4): Release-independent version*
- SAP Note [1021079](#) : *Event history*
- SAP Note [1645120](#): *Trigger of a background event is unknown*
- SAP Note [2000417](#): *Problems with SAPEVT as of kernel release 7.40*
- SAP Note [2001336](#): *Downport: ICF service for triggering batch event*
- SAP Note [2015788](#): *Program "sapevt_rfc"*

External Commands and External Programs

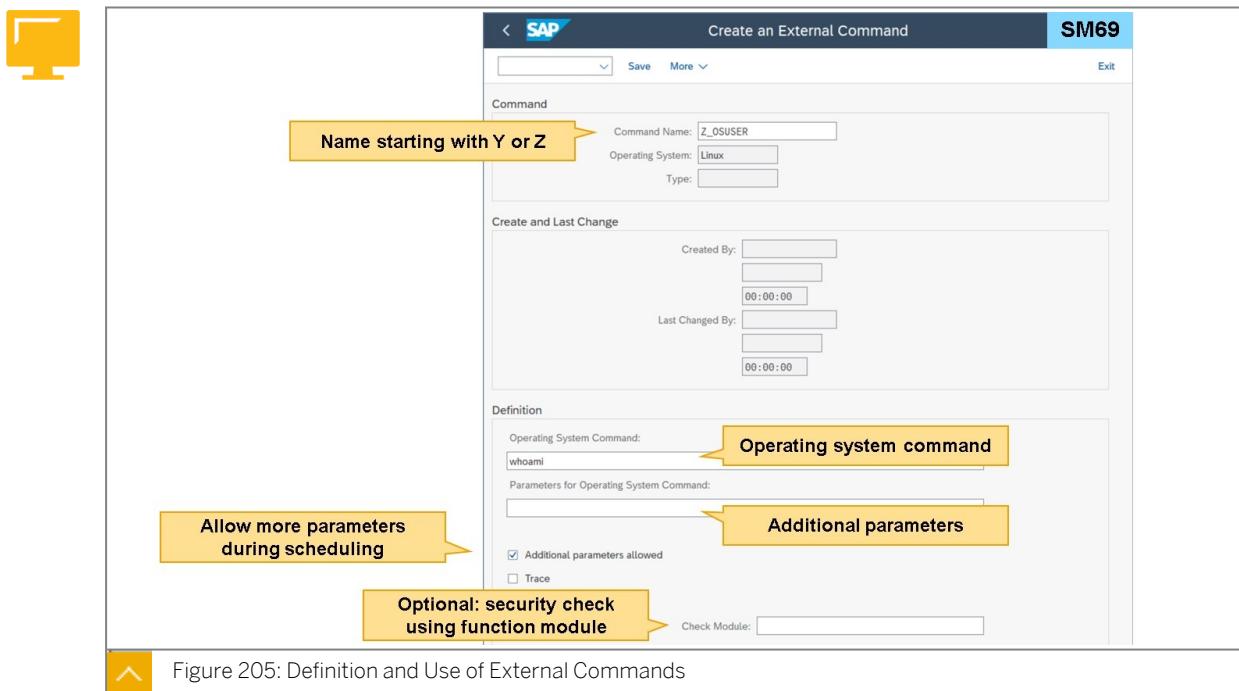
The background processing system differentiates between *external commands*, which are intended for “normal” users, and *external programs*, which should primarily be used only by system administrators. This differentiation gives administrators the ability to run any required external program, while normal users are restricted to external commands for which there are name-specific authorization checks. In both cases, the program `sapxpg` is called at operating

system level, which in turn starts the relevant program at operating system level, as shown in the figure below.



External commands are host system commands or programs **predefined** in the SAP system by an administrator. They are protected by authorizations so that end users can only schedule the external commands for which the system administrator has explicitly granted them authorization. You can, therefore, provide functions outside the SAP system, at operating system level, for users of the SAP system. The successful execution of external commands requires authorization for authorization object *S_LOG_COM: Authorization to execute external operating system commands*. This authorization object consists of three fields: *COMMAND*, *OPSYSTEM*, and *HOST*.

External programs are **unrestricted commands** that are neither predefined nor restricted by specific authorizations. A user with administration authorization can enter any external program in a job step. No further SAP authorization check is performed before the execution of the command. External programs provide an administrator (user with administration authorization for external commands (authorization object *S_RZL_ADM: Background Processing: Background Administrator*)) the flexibility to run **any** required host system command in the SAP system without administrative preparation.



The figure above illustrates the steps required in creating external commands:

1. Call transaction SM69 (*Tools → CCMS → Configuration → Display/Change External Commands*).
2. Choose *Create*.
3. Enter the information for the new command, using the following rules:
 - External commands are uniquely identified with a *command name* (beginning with Y or Z) and an *operating system* type. The *Type* field is filled automatically.
 - Specify an executable *operating system command* (if required: with the complete path) and specify any additional required or optional parameters.
 - Select the *Additional Parameters Allowed* checkbox if users are allowed to specify argument parameters when they execute the external command later or schedule it in a job. The additional parameters are added to parameter strings that may already be specified under *Parameters for Operating System Command*.
 - The *Trace* field is usually left blank. To follow the execution of an external command, use the trace parameter for the function module *SXPG_COMMAND_EXECUTE*.
 - If you have defined an additional authorization check, enter the name of the function module that performs it in the *Check module* field. This is usually a customer copy of the function module *SXPG_DUMMY_COMMAND_CHECK*. The system calls the function module automatically when a user tries to execute the external command or schedule it in a job.
4. Choose *Save*. To return to the command overview, choose *Back*.

Additional Information

The following SAP Notes might be helpful:

- SAP Note [677435](#): Overview: external programs and external commands

- SAP Note [686765](#): Security check when you execute external commands
- SAP Note [854060](#): Authorizations for external commands
- SAP Note [859104](#): Authorizations for external programs

Control Flags

As the figure below illustrates, you can use control flags to make specifications about the task and other runtime options. You do not normally need to change the default values.

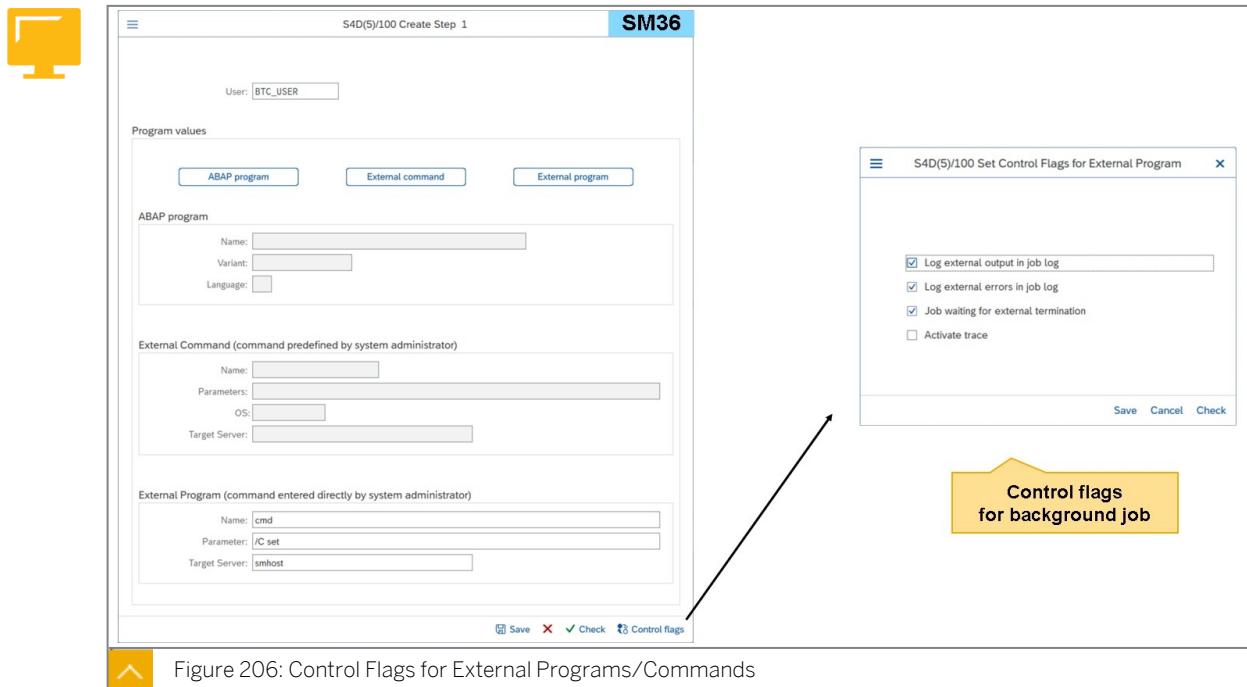


Figure 206: Control Flags for External Programs/Commands

You can, for example, specify the following:

- Whether the process is to be logged. The output data is written to the log as it is output by the external program. The language here can differ from your logon language. You can also call additional information about the data of an external program in the job log.
- Whether the SAP control program should wait for the end of the external program on the target computer (use this option for example if you want to receive the return value from the external program).
- Whether you want to switch on the more detailed level of the trace.



Hint:

You can display additional information about the meaning of the control flags using the F1 help on the Set Control Flags for External Programs screen.

Options for Extending the Standard in Background Processing

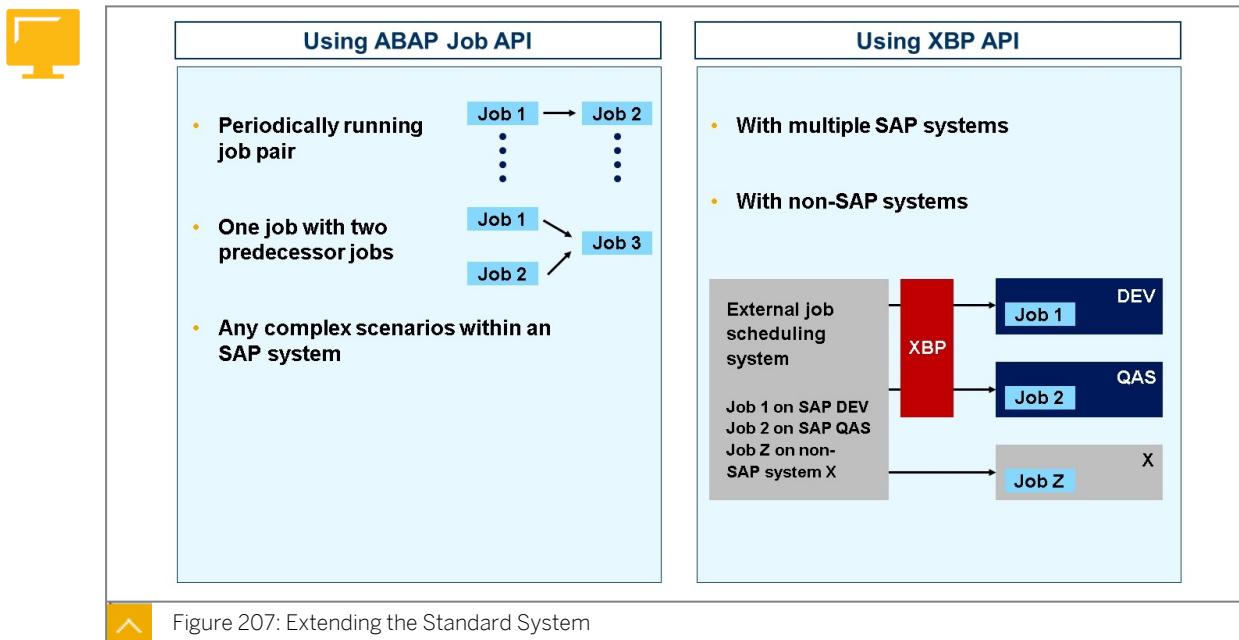
The job scheduling options described in the previous sections of this lesson do not cover all possible requirements. The following options – aside from using *SAP Business Process Automation by Redwood* (SAP BPA) – are available for realizing more complex scenarios:

The SAP system has various internal function modules that support you when defining your own job processes. These function modules are in the function groups *BTCH* and *BTC2*. You can define any complex scenarios with the help of these function modules.

SAP provides a set of interfaces that enable other system management environments to be connected to the SAP Computing Center Management System (CCMS).

- The *eXternal Monitoring Interface API* (XMI-API) interface logs the activities of external users and programs.
- The *eXternal interface for Background Processing* (XBP-API) interface allows the use of external job scheduling programs.

With these tools, you can schedule background jobs beyond the boundaries of the SAP system and include non-SAP systems, as shown in the figure below.



In addition to conventional, local SAP background processing (transactions *SM36* and *SM37*), you can use a *job scheduler* for landscape-wide background processing.

For information about *SAP Business Process Automation by Redwood* (SAP BPA; formerly known as *SAP Central Process Scheduling by Redwood* – SAP CPS), see SAP Note [2005087](#) – *SAP BPA V9*.

SAP Solution Manager also provides some central functions for job scheduling in the *Job Management* work center. SAP Cloud ALM also offers central functions in the *Job & Automation Monitoring* tile as part of *SAP Cloud ALM for Operations*.



LESSON SUMMARY

You should now be able to:

- Create a time-based job
- Create an event and a job based on that event
- Create a job that contains external commands or external programs
- List options to extend the standard background system

Troubleshooting Background Issues



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- List utility programs and tables for background processing
- Analyze problems in background processing

Tables, Authorizations and Utility Programs in Background Processing

The following figure shows selected tables, authorization objects and utility programs that are used in background processing:

	Selection of tables in background processing	SE16
	<ul style="list-style-type: none">▪ TBCO – job header table (job management data)▪ TBTP – job step table (individual processing steps in jobs)▪ TBTS – control table for the time-driven job scheduler▪ BTCEVTJOB – table for jobs scheduled to wait for an event▪ BTCCCTL – control table for background processing▪ BTOPTIONS – configuration table for background processing	
	Authorization objects	PFCG
<ul style="list-style-type: none">▪ S_BTCH_ADM – authorizes a superuser to manage background processing▪ S_BTCH_JOB – permits the user to perform different operations on jobs▪ S_BTCH_NAM / S_BTCH_NA1 – authorizes to schedule programs for (other) users▪ S_LOG_COM – authorization for execution of logical commands		
	Selection of utility programs BTCAUX* and SXBP_* for certain tasks within background processing	SA38
<ul style="list-style-type: none">▪ BTCAUX01 – display jobs with missing print parameters▪ BTCAUX04 – identify periodic jobs scheduled more than once▪ BTCAUX14 – listing jobs according to their frequency▪ SXBP_TEST_GET_JOB_DEFINITION – evaluate job definition using XBP interface▪ SXBP_TEST_TRANSFER_SPOOLIST – test transfer of spool lists using XBP interface		

Figure 208: Tables, Authorizations and Utility Programs in Background Processing

Tables in Background Processing

Table TBCO is often called the batch main table. Each batch job has exactly one entry in this table.

Job steps data are stored in table TBTP. There is a 1:n relationship between TBCO and TBTP. TBTP contains one entry for each job step.

Table TBTS contains information of all time based jobs (job, with start condition *Date / Time*), which have not started yet (job, with start condition *Date / Time*). There is one entry for each such job. The time based scheduler, which is a part of the batch runtime system, select jobs from this table in small periods of time and start it.

Similar to TBTS table, table BTCEVTJOBS contains information of each event based job, which is still “waiting” for its event to start.

Table TBTCCNTXT is used for storing the application specific return code of a job and the parent child information of jobs.

Certain activities of the batch runtime system are recorded in table BTCCCTL .For example: the time stamp of the last execution of the time scheduler and event scheduler.

The table BTCOPTIONS finally can be considered as a configuration table.

Authorization Objects in Background Processing

A user with BTCADMIN = **x** for authorization object **S_BTCH_ADM** can

- Access background jobs in all clients of an SAP system. In the job overview, the system displays all background jobs throughout the system.



Note:

Without this authorization, users can only work on background jobs in the clients in which they are logged on.

- Perform all functions on background jobs.
- Execute external programs in job steps.

Other values for BTCADMIN for this authorization object may for example allow to copy other users' jobs or create periodic jobs.

The authorization object **S_BTCH_JOB** consists of the authorization fields JOBACTION and JOBGROUP. JOBGROUP must always have the value *****. For JOBACTION, you can assign the values: PLAN (copy or repeat other users' jobs), DELE (delete other users' background jobs.), RELE (release jobs (including your own)), SHOW (display definitions of other users' jobs), and MODI (modify other users' jobs.).



Note:

A user without any specific authorization for jobs may perform the following actions:

- Schedule jobs for which the job class is C and cannot be changed.
- View and change steps of his or her own jobs.
- Delete his or her own jobs.
- View the job details of his or her own jobs.

Authorization object **S_BTCH_NAM** (with authorization field BTCUNAME – user name that a user can specify as an authorized user) determines the authorized users, which users can choose from when scheduling a background job.



Note:

A user can always enter himself or herself as an authorized user when scheduling a job. Thus, an authorization for this object is only required if your users need to use other users as authorized users. This could be the case, for example, if a user requires special authorizations for background jobs.

With authorization object **S_BTCH_NA1**, an enhancement is available that also checks the program to schedule as well as the user (additional authorization field PROGNAME for the name of the program they want to run in the background).



Hint:

Programs can be scheduled without a start authorization check as a job step if they are not assigned to an authorization group. SAP Note [1946079](#) introduced the supplementary authorization check of the authorization object **S_PROGNAM** for the dialog-based generic program start. SAP Note [2272827](#) – *Check of S_PROGNAM for scheduling of job step* enables the additional authorization check if job steps are scheduled using transaction SM36 or the job API, too.

Finally, authorization object **S_LOG_COM** (with authorization fields COMMAND for the name of the external commands, OPSYSTEM for the type of operation system on the target host) and HOST (for the hostname of the target host) needs to be assigned to users who want to create job steps with *external commands* in background processing. For more information, see SAP Note [854060](#) – *Authorizations for external commands*.



Note:

For the execution of *external programs*, you need system administrator authorization (authorization object **S_RZL_ADM**), see SAP Note [859104](#) – *Authorizations for external programs*.

Selected Utility Programs

A number of utility programs already exist for certain tasks within background processing. SAP Note [997328](#) – *Utility programs for background processing* lists some of them together with specific SAP Notes for each of the programs. These programs have been extended over time. For example, SAP Note [1439806](#) – *Listing jobs according to their frequency* ships report BTCAUX14 which (among others) list how often a particular job runs in a certain time period.

In the same way, SAP Note [1180215](#) – *Test programs for the XBP interface* also contains a list of programs including a link to additional SAP Notes.

Components Involved in Background Processing

The following figure shows the components that are involved in processing various types of background jobs.

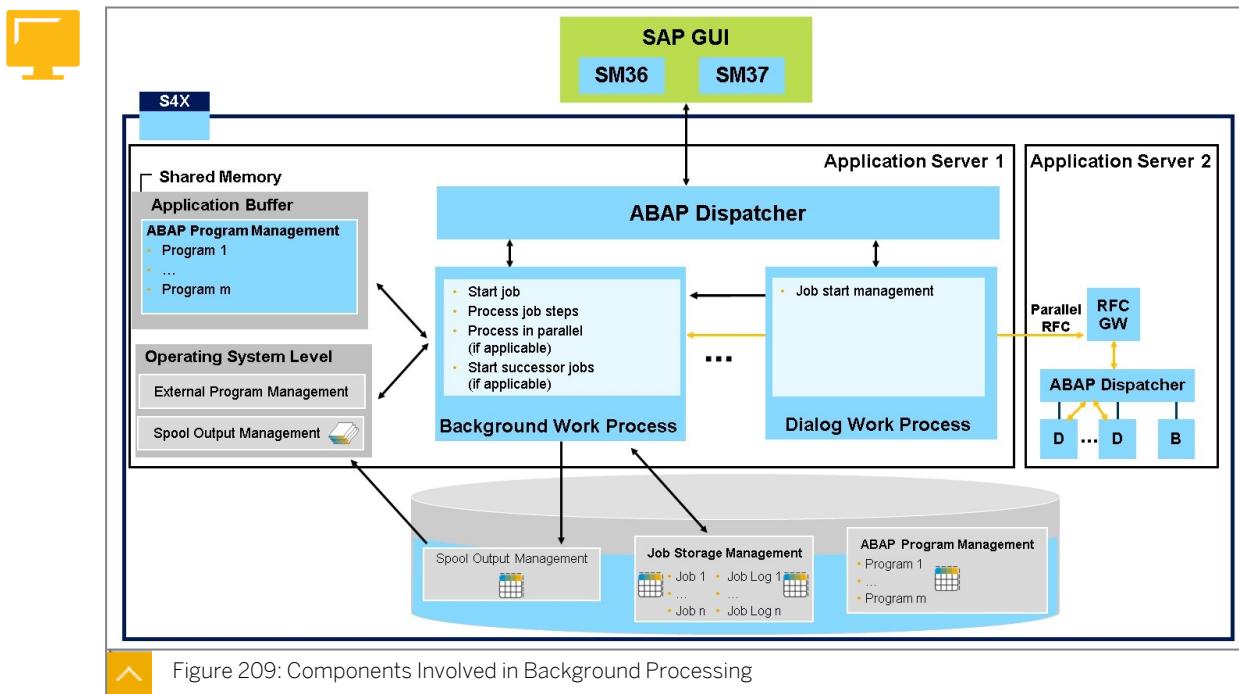


Figure 209: Components Involved in Background Processing

Troubleshooting the Background Processing System

Problems in SAP background processing can be classified as follows:

- Configuration problems
- Administration problems
- Runtime problems.

The following examples show case studies how the different analysis tools may be used for troubleshooting:

When a **job was not started**, (besides of obvious things such as the start date has not yet been reached) you may proceed as follows:

1. Check whether the user who has scheduled the job has the authorization to release the job.
2. Check whether the job has a target server assigned and – if “yes” – if there is a free background work process on the target server for the job class requested.
3. Use the analysis tool for background processing (transaction SM65) to check the profile parameters of the background processing on all applicable application servers.
4. Check the *system log* (transaction SM21) on all applicable application servers for unusual entries.
5. Record the component of the background processing which is responsible for starting the job and analyze the trace information. For this, enter transaction SM61 and switch to change mode (*Modify control object list entry*). Switch on the trace level for the control object you want to check (and do not forget to switch off the trace again later). As long as the control object has run at least once, you can have a look at the trace file for the work process on the appropriate application server.

**Note:**

In the case of control objects that run in dialog work processes, it is not possible to predict in which of the available work processes the object will be processed. However, it is quite likely to be work process 0 or 1. The best approach is to sort the list of trace files according to date and time (transaction AL11). The trace entries should be found in one of the most recently written trace files. Look through the files for entries that have an "L" in the first column.

For control objects that run in a background work process, you can find the work process number in the *job detail* data in the job overview (transaction SM37).

**Troubleshooting Example:** Job was not started although start date / time has reached

Check Authorization of the user who has scheduled the job
Transactions SU01 and PFCG

Check free background work processes on target server(s)
Transactions SM37 and SM50

Check profile parameters for background processing
Transactions SM37 and SM65

Check system log for unusual entries
Transaction SM21

Activate trace and record again – then check developer trace
Transactions SM61, SM50 and AL11



For additional information, see also SAP Note 165084 – *Why does my job not start?*

Figure 210: Troubleshooting Background Processing - Example

For additional information, see SAP Note [165084](#) – *Why does my job not start?*.

When the **job terminates unexpectedly**, you may proceed as follows:

1. Display the job log and check for any error messages. For example there may be a variant for an ABAP program that cannot be found, a runtime error in an ABAP program has occurred or an external program could not be started.
2. Check whether the job has a target server assigned and then check the *system log* (transaction SM21) on all applicable application servers for unusual entries.

When the **job remains in status 'active'**, you may proceed as follows:

1. Check whether the job really is still running. For this, select the job in the job overview (transaction SM37) and choose *Job → Check Status*. If the job is still active, you may try to capture it (*Job → Capture: active job*). This can be used to pause and analyze an active job which is in the middle of executing an ABAP program. An ABAP debugger window is opened, showing the ABAP program code at the point at which it is being executed. The code can then be analyzed.

**Note:**

However, this only works for ABAP programs that are executed as job steps. It is not possible to analyze external programs in this way. For an external program, the check for whether it is still active has to be made at the operating system level.

2. Check whether the job waits for an external program that is being executed in the job to be terminated. In this case it might be helpful to check whether the user SAPCPIC exists in the client in which the job is running.
3. Check whether the job server was restarted while the job was running. For this, you need to
 - find out the start time and the execution host of the job (for example, by checking the job log)
 - find out when the job server was last started (by checking the *system log* of the job server).
4. When the job server was restarted while the job was running, the runtime environment of the background processing system may not have been able to update the job status in the database. In this case, you may try to manually set the job status to *canceled* (*Job → Cancel active job* in transaction SM37).

Additional Information

For more information on trouble shooting, see SAP Note [3008195 – FAQ: Background Processing BC-CCM-BTC-*](#) and the *Support Wiki* for SAP Background Processing System (<https://wiki.scn.sap.com/wiki/display/SI/SAP+Background+Processing+System>).

Additional Information on Background Processing

The following section contains some typical question related to background processing, together with some SAP Notes.

Where can I get more information about background jobs?

Note the answers provided in SAP Notes [31503: FAQ: Background jobs](#) and [3008195 – FAQ: Background Processing BC-CCM-BTC-*](#).

How can I debug programs in the background?

Note the procedure described in SAP Note [573128: Debugging programs in the background](#).

Identifying jobs with deleted users

Note the solution described in SAP Note [1533612: Identify jobs with deleted users](#) which ships program BTCAUX09.

How often does each job run?

Note the solution described in SAP Note [1439806: Listing jobs according to their frequency](#) which ships program BTCAUX14. Transaction SM37C (*Extended Job Selection*) gives you another way to access this valuable information under *Goto → Job Statistics*.

Why does my job not start?

Note the solution described in SAP Note [165084: Why does my job not start?](#).

How can I see who has deleted which job?

Note the solution described in SAP Note [850885](#): *Logging the deletion of jobs.*

How can I log calls of external commands?

Note the solution described in SAP Note [1663158](#): *Logging calls of external commands.*

How can I select jobs with the step user, SM37C?

Note the solution described in SAP Note [1742282](#): *Job selection by step user referring to transaction SM37C.*

What is the meaning of program BTC_OPTIONS_SET?

The program *BTC_OPTIONS_SET* offers you versatile options for background processing in your system. For each option in this report, the SAP Note that provides additional information for the relevant function is identified.

Which transaction can I use to only display jobs?

Note the solution described in SAP Note [1665057](#): *Display transaction for background jobs.*

Errors occur during the multiple scheduling of periodic jobs. What can I do?

Note the solution described in SAP Note [1130883](#): *Periodic job is scheduled several times which ships program BTCAUX04.*

My job uses a printer that does not exist anymore and reports errors.

Note the solution described in SAP Note [1226171](#): *Identifying jobs with deleted printer which ships program BTCAUX02.*

In which job has this spool request been created?

Note the solution described in SAP Note [1145265](#): *In which job has this spool request been created?* which ships program *BTCAUX05*. Alternatively, you can also access this function in transaction *SM37C* with the menu path *Goto → Further Selections → Jobs for Spool Request*. This path is only available in AS ABAP-based SAP systems as of *SAP_BASIS 731*.

Why are background processing jobs not generally transportable?

Note the information described in SAP Note [51135](#): *Why are bkgd processing jobs not generally transportable?.*



LESSON SUMMARY

You should now be able to:

- List utility programs and tables for background processing
- Analyze problems in background processing

Learning Assessment

1. What can be executed as a job step

Choose the correct answers.

- A ABAP program
- B External Command
- C SAPUI5 application
- D External Program

2. What status can a job have?

Choose the correct answers.

- A Running
- B Released
- C Waiting
- D Finished
- E Error
- F Active

3. If one of the system's ten background work processes is reserved for class A jobs, and jobs are being processed in the other nine work processes, the reserved work process remains free even if a class A job is waiting for execution.

Determine whether this statement is true or false.

- True
- False

4. Which of the listed options for time-dependent scheduling of a job can be implemented with the standard resources of the SAP system?

Choose the correct answers.

- A Daily at 15:00
- B The second-last working day of every month
- C Every nine days, but not on Sundays
- D Only tomorrow at 23:55
- E Every year on April 1st at 08:00

5. Which of the listed options for event-dependent scheduling of a job can be implemented with the standard resources of the SAP system (AS ABAP)?

Choose the correct answers.

- A Whenever the system switches to the *Night* operation mode
- B After *Job1* has run successfully
- C Whenever a particular event has been triggered
- D Whenever the periodic *Job2* has run successfully

Learning Assessment - Answers

1. What can be executed as a job step

Choose the correct answers.

- A ABAP program
- B External Command
- C SAPUI5 application
- D External Program

You are correct: You can use an ABAP program, an external command or an external program as a job step.

2. What status can a job have?

Choose the correct answers.

- A Running
- B Released
- C Waiting
- D Finished
- E Error
- F Active

You are correct! A job can have the status *Released*, *Finished*, and *Active*. In addition to the three correct answers, a job can also have status *Planned*, *Ready*, and *Canceled*. Read more on this in the lesson “Explaining Background Processing”.

3. If one of the system's ten background work processes is reserved for class A jobs, and jobs are being processed in the other nine work processes, the reserved work process remains free even if a class A job is waiting for execution.

Determine whether this statement is true or false.

True

False

You are correct! The work processes that are reserved for class A jobs process class A jobs. The option to reserve work processes for class A jobs should only prevent all work processes being occupied with class B or C jobs.

4. Which of the listed options for time-dependent scheduling of a job can be implemented with the standard resources of the SAP system?

Choose the correct answers.

A Daily at 15:00

B The second-last working day of every month

C Every nine days, but not on Sundays

D Only tomorrow at 23:55

E Every year on April 1st at 08:00

You are correct! You can implement all the options mentioned with standard AS ABAP tools.

5. Which of the listed options for event-dependent scheduling of a job can be implemented with the standard resources of the SAP system (AS ABAP)?

Choose the correct answers.

A Whenever the system switches to the *Night* operation mode

B After *Job1* has run successfully

C Whenever a particular event has been triggered

D Whenever the periodic *Job2* has run successfully

You are correct! Option D cannot be implemented directly in the standard system. An enhancement to the standard system, for example, with function modules or using *SAP Central Process Scheduling* (SAP CPS), can also implement this option, as well as many others.

UNIT 10

System Monitoring and Troubleshooting

Lesson 1

Using Local Monitoring Tools

413

Lesson 2

Explaining the Concept of Monitoring Tools in SAP Solution Manager

427

Lesson 3

Tracing and Logging in the SAP System

437

Lesson 4

Troubleshooting Procedures

445

UNIT OBJECTIVES

- Understand the reasons for System Monitoring
- Describe the local Monitoring Architecture
- Use SAP MC for System Monitoring
- Use Alert Monitor for System Monitoring
- Outline how to use CCMS as a Central Monitoring System
- Use SAP command line programs to monitor, test and solve events
- Explain the technical prerequisites to use Monitoring and Alerting Infrastructure (MAI)
- List the System and Application Monitoring Use Cases
- Understand the cross-application Alert Functions
- Use the System Monitoring Application
- Describe the Interface Monitoring
- List the options available for tracing and logging in the SAP system
- Perform simple traces in the SAP system
- Develop procedures for structured troubleshooting

Using Local Monitoring Tools

LESSON OVERVIEW

This lesson provides an introduction to the local monitoring architecture of an ABAP based On-Premise SAP System and discusses the available local tools to display and analyze alerts. We will also briefly discuss the possibilities to build a central monitoring environment using this infrastructure.

Business Example

You want to ensure good performance for business processes. Therefore a regular and proactive monitoring of the SAP systems should be done.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Understand the reasons for System Monitoring
- Describe the local Monitoring Architecture
- Use SAP MC for System Monitoring
- Use Alert Monitor for System Monitoring
- Outline how to use CCMS as a Central Monitoring System
- Use SAP command line programs to monitor, test and solve events

Why is monitoring important?

Monitoring: Why and How?

Nowadays, business processes can involve many components. These components (whether produced by SAP or not) must be monitored, because either a gradual reduction in performance or a sudden breakdown of a component could affect overall productivity. As a system administrator, it is your responsibility to monitor the system landscape regularly, both to deal with errors and to prevent them.

Once an event has been reported it might be useful to use local monitoring options to find the root cause for the event.



- Why?
 - To ensure the efficient processing of business processes.
 - To ensure system security and stability.
- How?



- Centrally and cross-system
- With an alert if an error occurs
- With help that provides cross-system detailed information if an error occurs

Example:

A file system where files of the SAP database are stored is 100% full. The database can no longer extend the tables in the files. A user performs a business transaction in the context of which a data record should be asynchronously added to a table. The insert fails due to the space problem on file system level. To avoid further error issues the entire asynchronous update mechanism is deactivated by the system. All affected user sessions hang with the display of the hourglass. The SAP system hangs. If the fill level of the file system had been monitored regularly, the administrator could have taken action at the right time and system downtime could have been avoided.

Local Monitoring Architecture using Data Suppliers

Technical Basics: Data Supplier

The local monitoring architecture (also called CCMS monitoring architecture) is delivered with the data suppliers for the most important components of your SAP system and its environment and can therefore be immediately used without the need of further configuration. When starting an local monitor user interface, you can see that the data suppliers for reporting for the following components are already active:

- The **host systems** on which your SAP system is running
- The **database**
- **SAP instances** (application servers) and their services
- Components outside the system

You do not need to prepare or activate the monitoring architecture. The data suppliers in your system are either started automatically when the system is started, or are started as they are required. There are two different types of data suppliers

- **Passive Data Suppliers** are started by the monitoring architecture, which is why they must be defined there. "Passive" describes the behavior of the data supplier in relation to the monitoring architecture: it does not start itself, but rather must be started by the monitoring architecture. Passive data suppliers are also known as data collection methods.
- **Active Data Suppliers** are started by the monitored application rather than the monitoring architecture. These data suppliers are active in their start behavior in relation to the monitoring architecture.

A data supplier writes the values for the monitored objects in a segment of the shared memory, known as the **monitoring segment**.

The next figure shows that the local monitoring infrastructure consists of three parts: data collection, data storage, and data analysis.

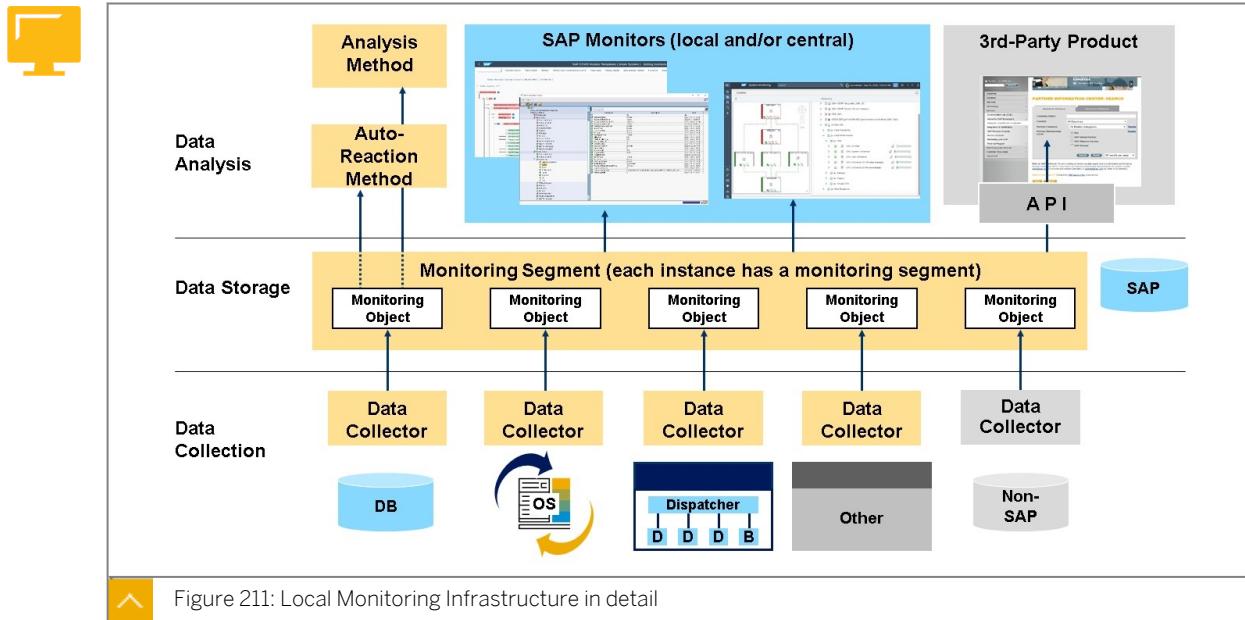


Figure 211: Local Monitoring Infrastructure in detail

At the **data collection** level, small subareas of an SAP system are monitored by special programs called data collectors. Data collectors can be ABAP, C/C++, or Java programs. In an ABAP based SAP system, hundreds of data collectors exist. Each data collector checks its sub-component at regular intervals and stores the collected monitoring data in the local monitoring segment.

Also, the monitoring infrastructure can be extended. You can integrate your own components using data collectors that you have written yourself.

The monitoring segment is the area of main memory that contains the monitoring data from the data collectors. Therefore, the monitoring segment performs **data storage**. The main memory data is continuously overwritten. Nevertheless, historical data can be kept in explicit database tables. You can then analyze the data later. The data collection and storage elements must be present on every component that is to be centrally monitored.



Caution:

Each instance of an SAP system (with the software component SAP_BASIS) has its own monitoring segment in the shared memory. This means if there is, for example, an SAP system with 8 instances, there are 8 separate monitoring segments. The number of instances determines the number of monitoring segments. Whether or not multiple instances run on the same hardware does not matter here.

If the system identifies a problem, it can execute a specifically prepared **Auto-Reaction Method**, such as informing a responsible person.

Predefined **Analysis Method** help to clarify problem situations.

There are many different tools available to display the data from the monitoring segments for **data analysis**:

- Displaying functions in SAP Management Console (SAP MC) and SAP Microsoft Management Console (SAP MMC)
- SAP command line programs like `dpmmon` or `msmon`.

- In SAP ABAP based SAP system the Alert Monitor (transaction RZ20) to display and analyze alerts. Can be used local in one SAP System or - after configuration - central for your complete IT landscape.
- Metrics could also be transferred to an SAP Application Lifecycle Management product like SAP Solution Manager, SAP Focused RUN or SAP Cloud ALM.
- Third-party vendors and partners can use various interfaces to read the monitoring data from the monitoring segment.

**Note:**

For more information about SAP Application Lifecycle Management (ALM), see <https://support.sap.com/en/alm.html>

For a list of certified partner products, check <https://www.sap.com/partner.html>.

Monitoring with the SAP Management Console (SAP MC)

Several main monitoring scenarios are available using the monitoring options provided by the SAP Management Console (SAP MC).

The SAP MC provides alert monitoring information based on predefined thresholds. With the SAP MC you can view the current state of the system as well as the most severe alerts that exist in the system with the corresponding alert colors.

System State Monitoring (for SAP Instances)

The monitoring area is divided in two parts. One area is for the *open alerts* and the other reflects the *current status*.



Alert Name	Description	Time
ResponseTime	3 msec	2022 11 04 11:28:09
ResponseTimeDialog	1 msec	2022 11 04 11:28:09
ResponseTimeDialogRFC	144 msec	2022 01 31 10:15:07
ResponseTimeHTTP	- msec	2022 11 04 11:28:09
FrontendResponseTime	3 msec	2022 11 04 11:28:09
QueueLength	0 msec	2022 11 04 11:28:09
Load+GenTime	0 msec	2022 11 04 11:28:09
RestTime	0 msec	2022 11 04 11:28:09
DBRequestTime	0 msec	2022 11 04 11:28:09
Utilization	1 msec	2022 11 04 11:27:52
PrintMode Utilization	0 %	2022 11 04 11:28:09
NumberOfRpDA	16	2022 11 04 11:28:09
ErrorRpDA	0	2021 01 26 08:49:18
ErrorRpDAQ	0.0 min	2022 11 04 11:28:09
EndeRpDA	0	2022 11 04 11:28:09
QueueLength	0 %	2022 11 04 11:28:09
LogonErrors	0 sec	2022 11 04 11:28:09
ProgramErrors	0 msec	2022 11 04 11:28:09
DialogSteps	10 msec	2021 01 26 08:49:18
UserCallTime	0 msec	2022 11 04 11:28:09
FrontendTime	0 msec	2022 11 04 11:28:09
MonitoringTime	0.268 msec	2022 11 04 11:28:09
ResponseTime(StandardTran)	16 msec	2022 11 04 11:26:52
Logon.LoadMessage	0	2022 07 01 20:12:15
Logon.LoadMessage	Preferred instance selected for logon group SPACE:s4dhost_S4D_12	2022 07 01 20:12:15
UsersLoggedIn	6	2022 11 04 11:26:51

Figure 212: Monitoring SAP Application Server with the SAP MC

Open Alerts

To monitor the most severe alerts in the system, expand the Open Alerts tree structure and choose the relevant monitor. In the result pane, the SAP MC displays the most severe alerts that have occurred in the system. The table in the result pane displays the

relevant KPIs, their values, and the time the alert last occurred. This view does not necessarily reflect the current state of the system.

Current Status

To monitor the current data reported for every monitoring tree element, expand the Current Status tree structure and choose the relevant tree element. In the result pane, the SAP MC displays the current state of the system and the most severe current alerts. The table in the result pane displays the relevant KPIs, their values, and the time the value was last measured.



Note:

If the system has not generated any alerts, the table in the result pane displays only the name of the relevant KPI.

To **acknowledge the alerts** in the system with an ABAP backend, select the Current Status or Open Alerts node from the tree structure of the relevant instance. From the context menu, choose All Alerts. The Alert Inbox appears where you can view the alerts and acknowledge them.

In the right pane choose the left mouse button on the *Alert Name* headline for choosing the different **sort criteria**. Every selection alternates between sorting “by alert”, “order of the monitoring structure” or “by reverse alert”.

You can **configure the threshold values** if the predefined levels are not suitable for your needs. For ABAP based SAP systems you can configure the threshold values in the Alert Monitor (transaction RZ20).

Additional Monitoring Options (ICM Monitoring, SAP Web Dispatcher Monitoring, Enqueue Monitoring, ...)

If applicable, you can also view basic monitoring information about **ICM** or **SAP Web Dispatcher** operation. As an example, the available ICM Monitoring options are explained below.

- To display information about the created worker threads, choose the **Thread List** sub-node from the ICM tree structure.
- To display information about the existing connections, choose the **Connection List** sub-node from the ICM tree structure.
- To display information about all objects currently in the ICM Server Cache, choose the **Cache List** sub-node from the ICM tree structure.
- To display information about the proxy server list, choose the **Proxy List** sub-node from the ICM tree structure.

In addition you can find additional monitoring information for various processes, like the **enqueue server** or **message server**.

Database Monitoring

The hierarchical tree structure of an SAP system in the SAP MC includes a node for the database server. It often appears in a blue color.

To view detailed information, select the Database node. If an authentication is requested, enter the corresponding user-ID and the respective password.

The result pane displays the type of the server, the host of the database instance, the vendor of the database, and the SAP system ID (SAPSID).

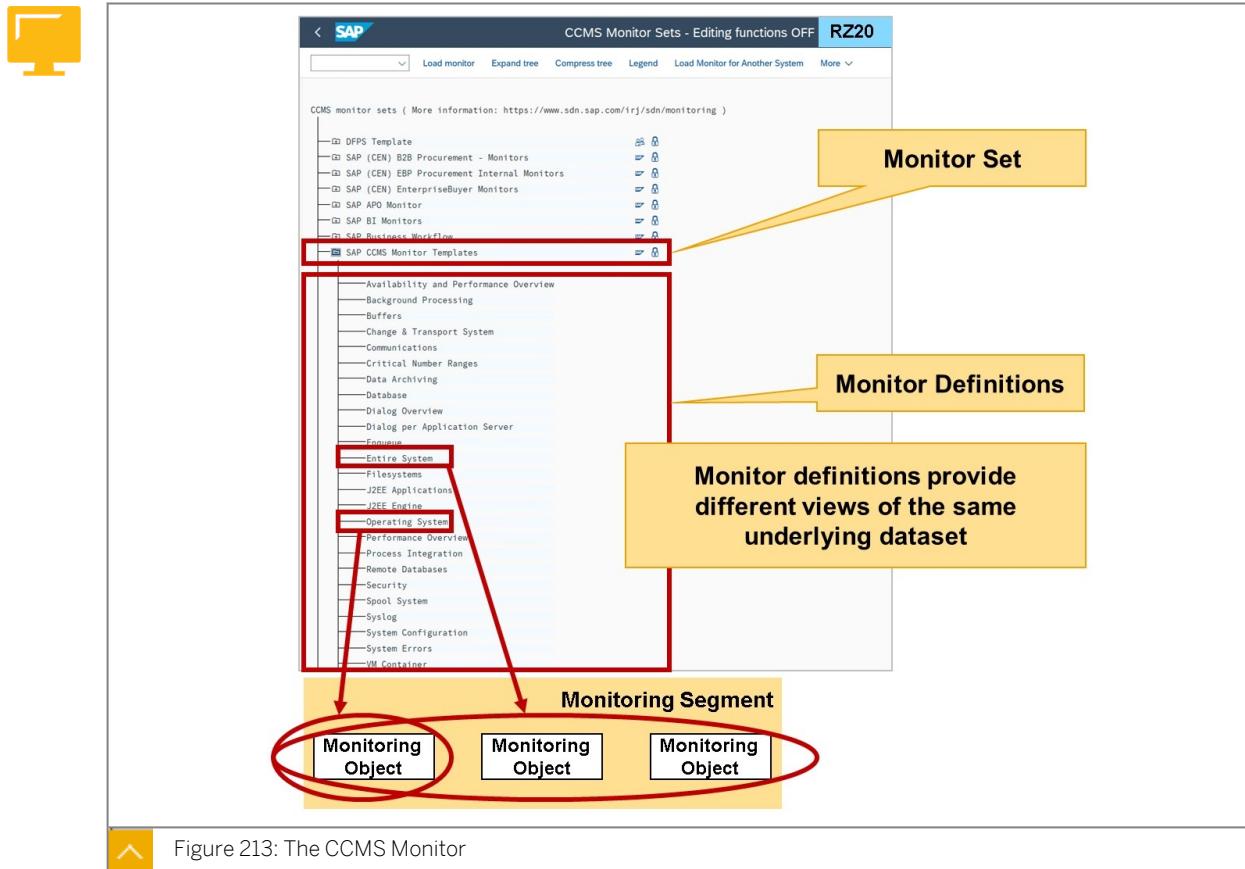
Operating the Alert Monitor

Alerts form a central element of monitoring. They quickly and reliably report errors - such as values exceeding or falling below a particular threshold value or that an IT component has been inactive for a defined period of time. These alerts are displayed in the Alert Monitor; this reduces the workload for the system administration, since they now only need to watch the error messages, instead of endless system data.

The Alert Monitor provides the following functions:

- You can use the Alert Monitor to perform complete and detailed monitoring of all SAP and non-SAP systems, the host systems, and the database.
- All errors generate alerts, which are displayed in a tree structure.
- The alerts contain a status indicator with a color and a numerical value. Yellow means a warning, red means a problem, and the numerical value shows the severity of the reported error. In the tree structure, the most severe alerts are passed upward in the display hierarchy. If a tree node is not displaying an alert, there is also no error in the entire branch below it.
- You can assign certain analysis and auto-reaction methods to the alerts, which contribute to faster processing of the error. If you double-click an alert, the monitoring architecture starts the assigned analysis method (such as the job administration transaction for a prematurely terminated job). An auto-reaction method, on the other hand, starts automatically as soon as the alert occurs. This includes executing operating system commands and sending an e-mail or an SMS message to the system administration.
- The Alert Monitor contains various view in which either the current or the open (that is, the unanalyzed) problem messages are displayed. Alerts are also archived.
- Threshold values, methods, and detailed help for many monitoring attributes and three extensive monitor sets with monitors for all aspects of system management are predefined on the basis of Best Practices in the monitoring architecture and are available in every SAP system.
- You can adjust all settings individually, and configure your own monitors.

SAP provides preconfigured monitor sets that you can use immediately. As figure *The CCMS Monitor* shows, the CCMS Monitor Sets (transaction RZ20) offer various monitor definitions that let you display the monitoring data from the monitoring segments in a tree structure. Every monitor set bundles monitor definitions that display various parts of the entire monitoring architecture by topic area. The tree structure allows a clear display when you are displaying a large number of measured values.



Monitor Set

A monitor set is a collection of different monitors. Monitor sets usually represent a product that needs to be monitored.

The provided monitor sets can be different for each system. For example, an SAP CRM system contains a special set for monitoring CRM scenarios. Of course, this also includes special data collectors that are preconfigured and delivered with an SAP CRM system.

Monitor Definition

A monitor definition describes the selection of monitor objects and monitoring attributes that you want to examine in more detail. You bundle monitor definitions into monitor sets in transaction RZ20. When you use a monitor definition, the requested data is compiled here and displayed in the corresponding monitor.

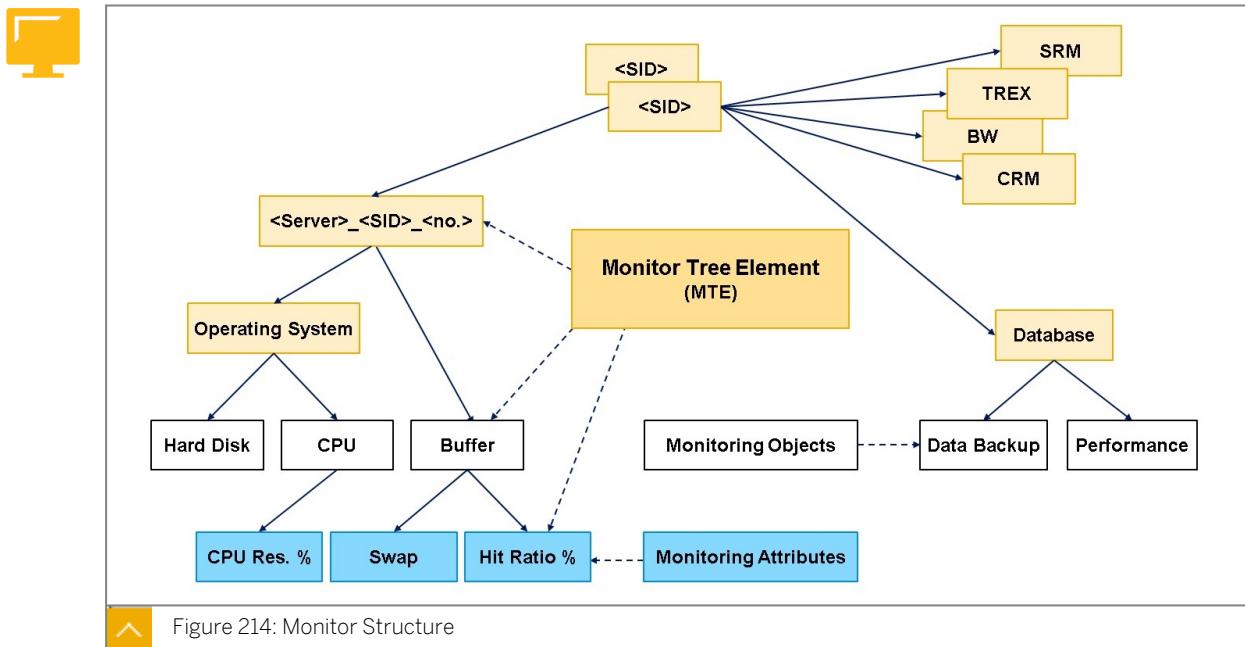
Monitor

A monitor is the display of monitoring objects and monitoring attributes that are included for display in the monitor definition.

The monitoring data for an attribute can be displayed in several monitors. The monitoring attribute *Hit Ratio* in the monitor *Program Buffer*, for example, can occur in several monitors. If you change the threshold values for this attribute in one of these monitors it is changed in all other monitors too.

Some monitors, such as the monitor *Availability and Performance Overview* in the monitor set *SAP CCMS Monitor Templates* do not display any data at first. This can be due to the fact that special settings are required to start the underlying data collectors.

At the beginning, you will use the preconfigured monitors. Later, you can also create your own monitors that display the exact data that you require for your daily monitoring work.



The figure *Monitor Structure* displays the tree structure of a monitor. A node in the monitoring tree is called a monitor tree element (MTE).

The measured values that are collected by the data collectors are displayed at the lowest level in the leaves of the tree. The leaves are called monitoring attributes.

Threshold values can be stored for a monitoring attribute. SAP supplies preconfigured threshold values. To adapt the monitor as closely as possible to your needs, however, you should check these threshold values and adjust them if required.

Monitoring attributes are grouped at the second-lowest level using monitoring objects. For example, the monitoring object program buffer contains, among others, the attributes hit rate and swap.

All other nodes in the tree structure the monitoring objects in a logical and clear way, so that you can easily find the relevant attribute.

A CCMS monitor displays different subareas of the monitoring data. A monitor can contain data from multiple SAP systems.

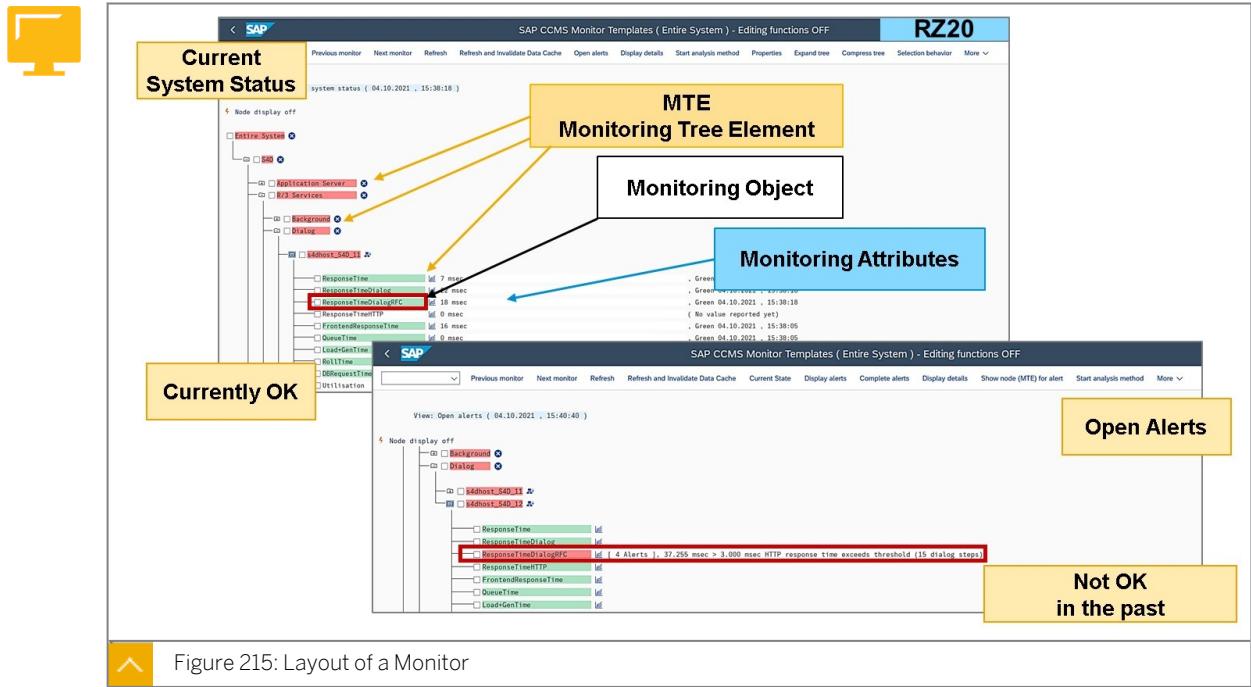


Figure 215: Layout of a Monitor

As the figure *Layout of a Monitor* shows, when you open a monitor, the corresponding monitoring data is displayed in the form of a tree. By clicking the “+” sign beside an MTE, you can expand the tree down to its leaves, which are the monitoring attributes.

Alert threshold values for triggering yellow and red alerts are assigned to monitoring attributes. If the threshold value condition is fulfilled, first a yellow, and then, if there is further deterioration, a red alert is triggered. The color of the monitoring attribute is propagated to its higher-level node in the tree, where the most severe alert is forwarded (red is more severe than yellow). As a result, the root of a tree already indicates whether an alert has occurred in that tree.

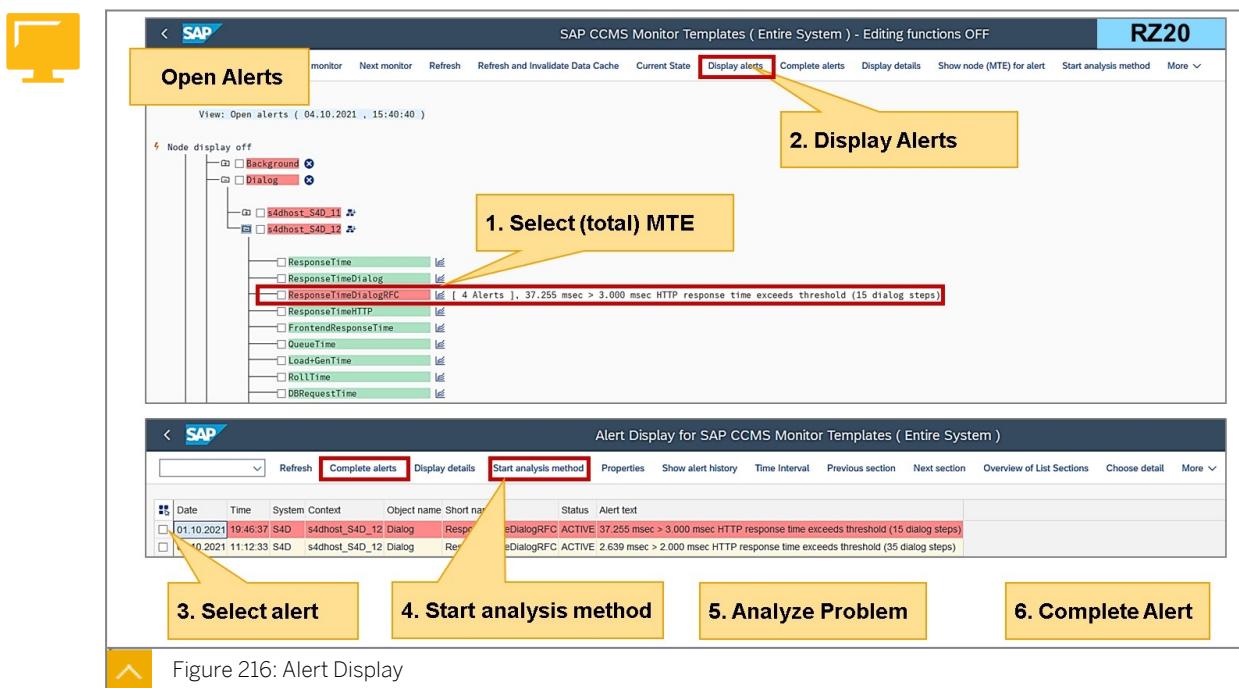
Processing Alerts

The monitor supports you in your daily work. After you have opened the monitor, the following views are available to you:

- The *Current system status* view displays the monitor with the newest reported data.
- The *Open alerts* view displays a monitor with alerts that have not been completed.

Example

During the previous night there may have been problems that are no longer occurring. In the *Current system status* view, the monitoring attribute is green, while it is displayed as red in the *Open alerts* view. After you have ensured that there are currently no problems, you can then investigate problems that have previously occurred.



As the figure *Alert Display* shows, you can see the selected view in the upper part of the monitor. You can switch between the views by choosing the *Current State* or *Open Alerts* buttons in the application function bar.

In *Open Alerts*, you can easily process the alerts that occurred in the past.

By double-clicking on an MTE in the tree, you open the *Alert Browser*, which displays a list of all alerts for the selected MTEs and all alerts below it in the tree. This means that if you double-click the root of the tree, the system displays a list of all alerts in the tree, sorted by red and yellow alerts.

If you want to process an alert, you need to select it. You choose *Start Analysis Method* to start the analysis method that is assigned to the MTE. The analysis method is a special tool that supports you when investigating problems. It can be transactions, specially programmed function modules, or URL calls. As a result, you do not need to remember all of the special tools, but simply use the *CCMS Alert Monitor* as a central point of entry.

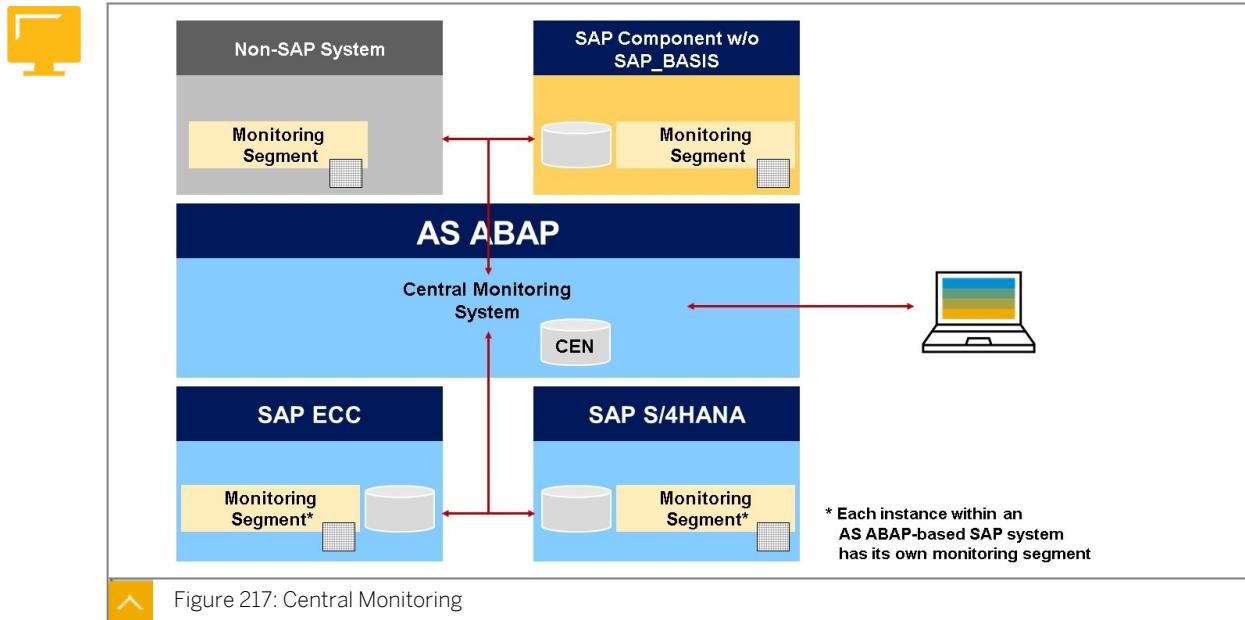
After you have clarified the problem situation, choose *Back (F3)* to return to the alert display. Then choose *Complete Alerts*. The processed alert is removed from the list and is stored in a database table.

Proceed in the same way with the remaining alerts until the list is empty. When you next use your monitor, only the newly occurred alerts will be shown.

If you want to display completed alerts again, choose *Show alert history* in the alert display. Completed alerts are displayed with the status *Done*.

Appendix: Using CCMS Infrastructure as Central System Monitoring Tool

Customers for whom the range of functions of this monitoring infrastructure is sufficient could also use it to configure a so called central monitoring system (CEN). A system landscape with up to 100 components can be monitored in this way with one instance as shown on the following slide.



Monitoring should be organized as efficiently as possible. There is not enough time for an administrator to log on to each host component to check its status. An efficient monitoring structure should be able to display the entire system landscape centrally at a glance. If an error occurs, the person responsible is automatically notified. Tools should be provided for the analysis of errors that provide cross-system detailed information about the problem.

As figure *Central Monitoring* illustrates, the CCMS monitoring infrastructure gives you the option of central and efficient monitoring for SAP systems.

This infrastructure must exist on every component that is to be centrally monitored. This is automatically the case for SAP systems with software component SAP_BASIS. The SAP start service (*sapstartsrv*) is used to connect components that do not contain SAP_BASIS.

Each component collects its own monitoring data using the infrastructure and stores it locally in the main memory. This part of main memory is called the monitoring segment and its size can be configured.

One SAP system in the system landscape is selected as the central monitoring system. Its release level should be as high as possible and it should also provide high availability.



Hint:

In large system landscapes, we recommend that you include a separate system that is used only for special tasks such as central monitoring, Central User Administration (CUA), and the transport domain controller. From the performance perspective, the workload of the central monitoring system increases only insignificantly as the collection of monitoring data is usually decentralized.

The central monitoring system collects the monitoring data for the components and displays it in various views. In this way, the administrator has a central view of the entire system landscape. If errors occur, the administrator can jump directly from the central monitoring system (via RFC) to the relevant component to analyze and solve a problem.

**Hint:**

We recommend that you use the Monitoring Infrastructure on a SAP Application Lifecycle Management system instead like SAP Solution Manager, SAP Focused RUN or SAP Cloud ALM, depending on your needs and system landscape.

Monitor, Analyze, Test and Resolve Issues Using SAP Command Line Programs

SAP Command Line Programs – Overview

This section introduces some interesting monitoring and test tools which could be used on operation system level.

Working with your `<sid>adm` user on operation system level of one of your SAP application server or central services instances and change to the folder `/usr/sap/SYS/exe/run` or a similar folder and search for programs with the name `*mon*`. You find a lot of interesting monitoring tools.

Some of them could be used to monitor processes on operation system level. Others are only for the SAP Support to use. In addition test programs are available, for example to test the logon groups.

**Note:**

Some tools have not been officially released. And they might not be available on all platforms and it is intended solely for experts to support them in their error analysis. Before you use them be sure you are allowed to and have the knowledge to do so. This is also the reason why, for some of them, there is no official documentation. But usually these programs contain a menu which is self-explanatory.

We want to start with a list of the most interesting tools:

- **dpmون**: Used to get the process overview of an instance in text mode. Dispatcher Monitor.
- **icmon**: used to monitor and manage the Internet Communication Manager (ICM). Like transaction SMICM.
- **wdispmon**: SAP Web Dispatcher Monitor program
- **gwmon**: Monitors the SAP Gateway. Like transaction SMGW.
- **msmon** and **mspprot**: Message Server Monitor Utility
- **lgtst**: Testprogram for SAP Login Info (for example logon groups)
- **mspprot**: Message Server test program
- **esmon** and **es2mon**: monitor the enqueue server and the enqueue replication servers

**Note:**

Most often there are other preferred tools provided and recommended by SAP, like existing transactions or web-based administration interfaces. The usage of command line programs can help, if the formerly mentioned tools are not available or cannot be used for a specific reason.

SAP Command Line Programs – Start the Tool

To start the program usually the following syntax is used:

```
<sapprogram> pf=<profile>
```

We want to explain the syntax using the example of calling the program **msmon** in a System called **S4D** and a central services installed called **ASCS10** on host **s4dhost** on Linux:

```
msmon pf=/usr/sap/S4D/SYS/profile/S4D_ASCS10_s4dhost
```

Another example is to call the program **dpmon** in a System called **S4D** and a instance called **D11** installed on host **s4dhost** on Linux:

```
dpmon pf=/usr/sap/S4D/SYS/profile/S4D_D11_s4dhost
```

Usually the tools have a menu with various options.



The screenshot shows the terminal output of the `dpmon` command. At the top, it says `dpmon pf=/usr/sap/S4D/SYS/profile/S4D_D11_s4dhost`. Below that is the **Monitor-Menu** with various options (d, e, p, l, t, w, b, c, m, s, o, v, z, a, f, j, x, q). To the right of the menu are two buttons: **Start dpmon** and **Monitor-Menu**. The main area shows a **Workprocess Table (short)** with four rows of data:

No	Pid	Type	State	Cause	Err	Prio	Sess-Key	Sess-Type	Locked	Sem	Time	Program	[Cli]	User
0	2893	DIA												
1	2894	DIA												
2	2895	DIA												
3	2896	DIA												

At the bottom, there is a section titled **Options for Workprocess Table (short)** with commands (a, s, k, r, d, e).

Figure 218: Example: Dispatcher Monitor (dpmon)

The tool `dpmon` is instance-specific and offers a *Monitor-Menu*, which allows to get details for the assigned work processes. In addition, further features are provided, for example, setting the restart flag, generating dump stacks, creating snapshots, or even stopping work processes.



Note:

Please consider the following SAP Note: [42074](#) - Using the R/3 dispatcher monitor 'dpmon'

SAP Command Line Programs – Usage Examples

Here are some examples under which circumstances it might be useful to use SAP command line programs:

- If you are using a standalone gateway (= gateway instance/SNA gateway), you must use the external gateway monitor **gwmon** as the transaction **SMGW** is not available
- You want to test, if the defined logon groups works correct using **lgtst**.
- You need to generate client load to check if the configuration is able to handle the amount of requests it is configured for. **icmon** or **wdispmon**.
- You forgot the password of administration user in order to use the web administration interface for ICM or SAP Web Dispatcher. **icmon** or **wdispmon**.
- You want to administrate the enqueue replication server on operation system level. **esmon** or **es2mon**.
- Because of a system standstill you need a tool on operation system level check the workprocesses of an application server. **dpmon**.

SAP Command Line Programs – Additional Information

- SAP Note: [42074](#) - Using the R/3 dispatcher monitor 'dpmon'
- SAP Note: [64016](#) - Using the SAP Gateway monitor GWMON
- SAP Note: [64015](#) - Description of test program "lgtst"
- help.sap.com: SAP Web Dispatcher → Administration of the SAP Web Dispatcher → General Administration Tools of SAP Web Dispatcher → Using the Command Line Programs icmon and wdispmon



LESSON SUMMARY

You should now be able to:

- Understand the reasons for System Monitoring
- Describe the local Monitoring Architecture
- Use SAP MC for System Monitoring
- Use Alert Monitor for System Monitoring
- Outline how to use CCMS as a Central Monitoring System
- Use SAP command line programs to monitor, test and solve events

Explaining the Concept of Monitoring Tools in SAP Solution Manager

LESSON OVERVIEW

This lesson provides a fundamental overview of the Monitoring and Alerting Infrastructure (MAI) that is the technical monitoring infrastructure in SAP Solution Manager.

Business Example

You want to obtain an overview of the technical requirements and use of some of the monitoring and analysis functions in SAP Solution Manager.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the technical prerequisites to use Monitoring and Alerting Infrastructure (MAI)
- List the System and Application Monitoring Use Cases
- Understand the cross-application Alert Functions
- Use the System Monitoring Application
- Describe the Interface Monitoring

Monitoring and Alerting Infrastructure (MAI)

Monitoring and Alerting Infrastructure (MAI) for Centralized Monitoring

The Monitoring and Alerting Infrastructure (MAI) was introduced in SAP Solution Manager 7.1 to be SAP's standard for centralized monitoring applications. The MAI allows stable, reliable operation of complex heterogeneous system landscapes.



Note:

The set of metrics that can be monitored and analyzed in SAP Solution Manager is not restricted to the metrics collected by Computing Center Management System (CCMS).

SAP Solution Manager can display additional data that does not originate in CCMS, which means it also has the following additional features:

- Additional ways of collecting data
- Additional modes of data transfer between place of origin and data storage in SAP Solution Manager

- Additional tools for displaying and analyzing the collected information



Note:

If you use the extended monitoring functions in SAP Solution Manager and the conventional CCMS monitoring functions concurrently, there is almost no negative impact, because SAP Solution Manager uses its own infrastructure for recording, storing, and displaying data whenever possible.

This section lists several monitoring and analysis capabilities of SAP Solution Manager and their technical prerequisites.



Note:

The information in this section is for introductory purposes only. The section describes the technical fundamentals of monitoring in SAP Solution Manager. License or maintenance contract-related topics are not covered here. Some of the functions described might, for example, require that your company makes use of least SAP Enterprise Support. To understand all the details, review the additional information at the end of the lesson.

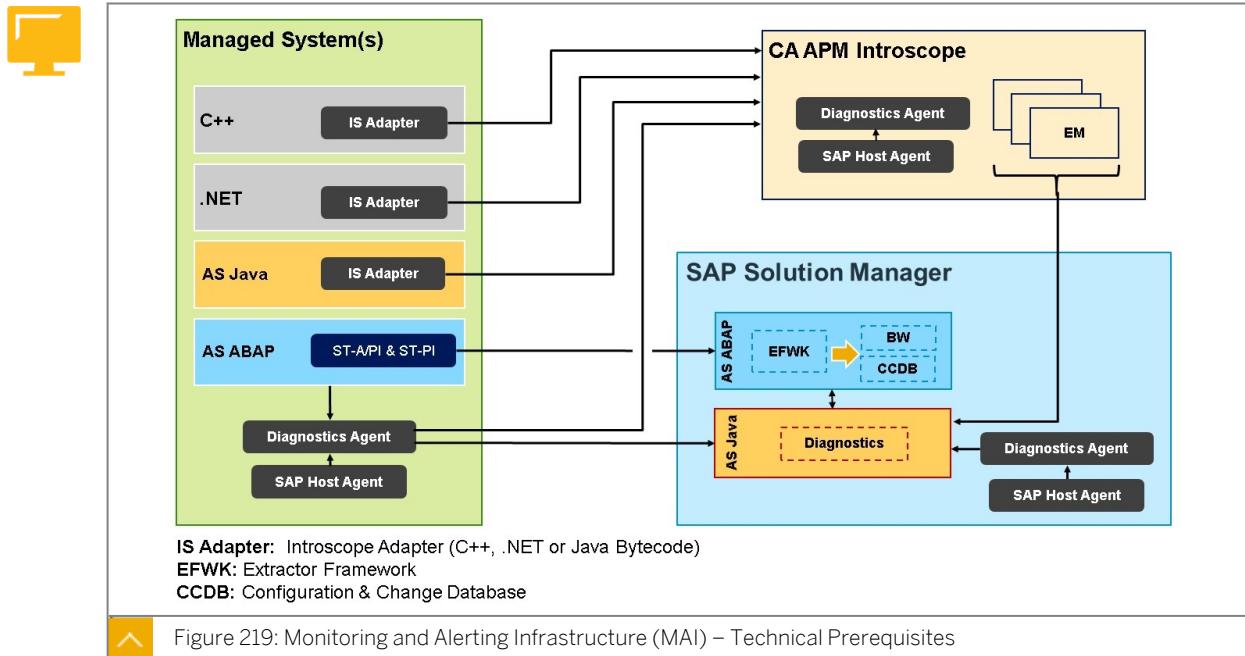


Note:

All functions shown in this lesson require that you have set up the SAP Solution Manager System using the *Mandatory Configuration* and connected the managed system using the *Managed System Configuration* using transaction *SOLMAN_SETUP* or the corresponding SAP Solution Manager Launchpad tile. In addition, you perform the configuration of the desired System & Application Monitoring application. These steps are **not** described here.

SAP Solution Manager enables you to monitor many different metrics of SAP and non-SAP systems within your system landscape.

To monitor the greatest possible number of metrics and use the full scope of monitoring capabilities in SAP Solution Manager, it is necessary to configure SAP Solution Manager and the remote system accordingly (as illustrated in the figure).



Remote systems are also called managed systems or satellite systems. SAP Solution Manager is also called the managing system.

The overview in the figure is highly simplified and depicts the data flow and potential applications.

To fully use the features of Monitoring and Alerting Infrastructure (MAI) in SAP Solution Manager, the following technical components must be installed:

CA™ Introscope Adapter (or IS Adapter)

These agents collect performance data and metrics from various components. They exist in different versions. The CA™ Introscope Bytecode Agent, for example, collects performance data and metrics from a Java server process. The collected data is then forwarded, for example, to CA™ APM Introscope Enterprise Manager.

CA™ APM Introscope Enterprise Manager (EM)

The EM serves as a central storage system, where all data collected by CA™ Introscope Agents (IS agents) is stored centrally. To enable display all the metrics of the Monitoring and Alerting Infrastructure (MAI) in SAP Solution Manager, they are also passed on to SAP Solution Manager.

SAP Solution Manager Diagnostics Agent (formerly SMD Agent)

The SAP Solution Manager Diagnostics Agent enables you to create a connection between SAP Solution Manager and the managed system and to collect information from the remote system. It must be installed once on each host or virtual host.

SAP Host Agent

The SAP Host Agent is the component that monitors the interaction between the host system and the operating system. It must be installed once on each physical host that you want to monitored.

**Hint:**

We recommend that you install the diagnostics agents and SAP Host Agent on non-SAP systems as well, to collect operating system metrics, for example. A diagnostics agent and SAP Host Agent must also be running on the host where the CA™ APM Introscope components are installed.

Extractor Framework (EFWK)

The Extractor Framework (EFWK) processes the information from the connected ABAP systems and from the CA™ APM Introscope Enterprise Manager within SAP Solution Manager. To do so, the EFWK saves the data in a product instance perspective within an InfoCube. An InfoCube is a technical component and part of SAP Business Information Warehouse.

ST-PI and ST-A/PI

An interface for the collection and transmission of data and performance metrics of ABAP components (technically, ST-PI and ST-A/PI are two SAP software components). The Extractor Framework (EFWK) calls the remote functions within the satellite systems and stores this information within the SAP Solution Manager system.

For more information, see <https://support.sap.com/en/alm/solution-manager.html>. In particular, note these two topic areas for this scenario on this page:

- Application Operations
- Setup & Configuration (which can be found under cross topics)

In particular, also read the following SAP Notes:

- [1365123](#) – Installation of Diagnostics Agents and the attached document, AgentInstallationStrategy.pdf.
- [797147](#) – Introscope Installation for SAP Customers.

For more information about configuring SAP Solution Manager and the infrastructure it used, see the SAP course SM100.

System and Application Monitoring Use Cases

Application Operation: System and Application Monitoring

The following is an overview of all available monitoring functions of the MAI (version: SAP Solution Manager Release 7.2 SPS15):

System Monitoring

The System Monitoring application provides an overview of the current status of technical systems, including their associated instances, databases and hosts.

System Monitoring is based on automated checks in regular time intervals in the four categories Availability, Performance, Exceptions, and Configuration. In each of these categories, several metrics and corresponding thresholds can be defined per managed object.

User Experience Monitoring

User Experience Monitoring (UXMon - formerly EEM) is an efficient toolbox for evaluating and reporting the availability and performance of your productive systems from a client-side perspective. As a result of the perfect integration into the E2E Diagnostics infrastructure, discovering, analysis, and resolution of occurring issues has been speeded up dramatically. Problems can often be solved before employees or customers even take notice about them, thereby yielding a lower TCO.

Interface Monitoring

Provides you with a powerful tool to monitor various interface technologies within your landscape.

All metrics are set up as Interface Channels, which allows you to abstract from a concrete technical implementation of an interface and bring together different interfaces that serve the same business purpose. Inside an Interface Channel the separate interfaces can be defined and metrics can be selected which monitor the interface in various aspects, focusing on error or backlog situations during processing. The Interface Channels can be assigned to ICMon scenarios as well as to Business Process Monitoring scenarios.

Process Integration Monitoring

Provides unified alert notifications, exceptions and status of PI components from a single-entry point. It also provides options to navigate to details of such exceptions in the respective host system. It becomes easier for addressing the issues more quickly and efficiently.

Business Intelligence Monitoring

In Business Intelligence (BI) scenarios it is critical to ensure that end users retrieve up-to-date information from the BI solution. This requires sufficient performance and availability of all involved components. Business Intelligence Monitoring has the capabilities to ensure that the involved technical systems and components are working as expected.

This includes the status overview for all technical components involved in SAP Business Intelligence Solution, the cross-system SAP BW process chains and single process chain steps, the monitoring of SAP BW queries and templates and the BusinessObjects specific jobs and correlation to system specific metrics.

Job Monitoring

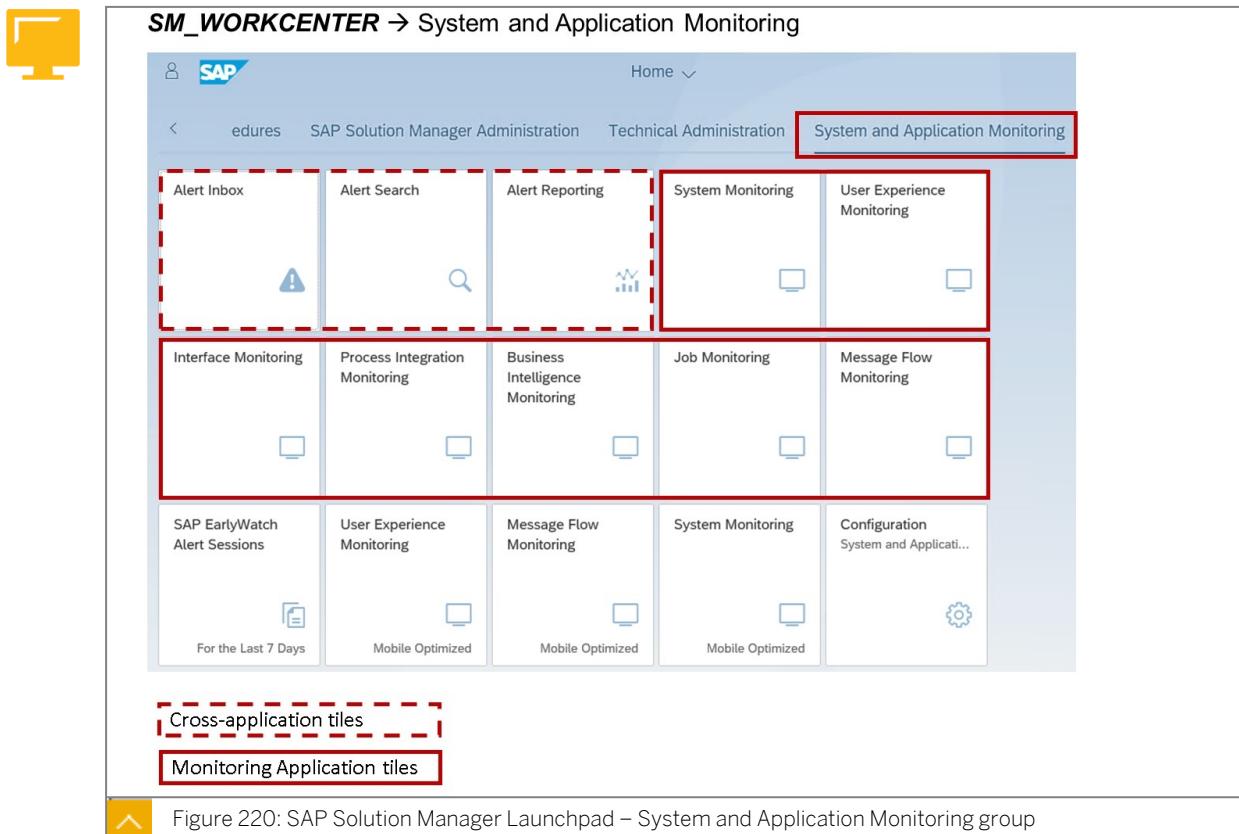
Job Monitoring belongs to important central monitoring tasks and can ensure that background jobs and job chains are running as expected. You can monitor the following types of jobs in a distributed system landscape:

- ABAP jobs
- SAP BW process chains
- SAP BusinessObjects (SBOP) jobs
- SAP Data Services (DS) jobs
- SMSE jobs: It allows you to monitor any type of externally scheduled jobs
- Custom jobs: It allows you to monitor any kind of job for which you have APIs that deliver job runtime information ("JobListGet"), that is, you can use the SAP Solution Manager job monitoring configuration and consumption UIs by implementing a couple of enhancements, for example, SAP S/4HANA cloud edition Application Jobs in the context of SAP Integrated Business Planning (SAP IBP).

Message Flow Monitoring

Message Flow Monitoring adds value to the already established Process Integration Monitoring. While PI Monitoring focuses on the technical monitoring of involved PI components, the Message Flow Monitoring provides a way to investigate and trace dedicated flows through the PI landscape on the level of single message instances. Besides company internal message flows, flows between business partners can also be monitored on a high resolution level, using technical and functional acknowledgements.

Once the monitoring scenarios you are going to use are configured you can start the transaction `SM_WORKCENTER` in the SAP Solution Manager, and navigate to the *System and Application Monitoring* section.



From here, you can access all monitoring scenarios that you have configured with transaction `SOLMAN_SETUP`. This includes, following cross-application functions (valid for all monitoring scenarios):

- **Alert Inbox:** Displays the triggered alerts for all configured scenarios.
- **Alert Search:** Once an Alert has been confirmed you could find the confirmed Alerts here.
- **Alert Reporting:** Pre Defined Reports for Alerts to use in a SAP Operations Control Center.

As well as the Monitoring Application itself, for example:

- **System Monitoring:** Graphical display of the status of selected systems.
- **Interface Monitoring:** Monitors the interfaces and connections between business-critical systems.
- **Message Flow Monitoring:** To monitor single message instances

In addition, you find the following tiles:

- **SAP EarlyWatch Alert Sessions:** SAP EarlyWatch Alert is an automatic diagnostic service that sends data regularly from the monitored system to SAP Solution Manager, which analyzes and evaluates it, to identify possible problems early, avoid bottlenecks and monitor system performance.
- **3 mobile optimized tiles for User Experience Monitoring, Message Flow Monitoring and System Monitoring**
- **Configuration System and Application Monitoring**, to jump to the *Guided Procedure for Configuration* (like SOLMAN_SETUP transaction)

Understand the Alert Inbox, Alert Search and Alert Reporting Functions

Central Alert Inbox

To explain the cross-application functions we want to introduce the *Alert Inbox* which is the central access point for analyzing and solving technical monitoring problems in an SAP Solution Manager landscape. As shown in the figure *Central Alert Inbox*, all alerts from all configured monitoring scenarios are displayed centrally in the *Alert Inbox*.

SM_WORKCENTER → System and Application Monitoring → Alert Inbox

Aggregated Alerts

Alert Name	Managed Object	Type	Ext. Sys.	TS Type	Instance	Curr.	Priority	Worst	Sold-To Party
T1E-ABAP	H24	T1E	ABAP			Medium	Medium	8	
Existence of Data Backup	H24	H24				Medium	Medium	3	
Frontend Response Time	TTJ-JAVA-wdfibmt7253...	TTJ	JAVA	wdfibmt7...		Medium	Medium	2	

Last Refresh 29.08.2019 15:09:46 CET Refresh

Alert Group

Alerts - "Frontend Response Time" at "TTJ-JAVA-wdfibmt7253_TTJ_14"

Metric Viewer

Alert Details

Central access point to handle all alerts from all Monitoring Scenarios

Figure 221: Central Alert Inbox

The *Alert Inbox* is divided into two areas by default:

- **Aggregated Alerts** (upper area): This area lists all alerts that belong to the same type of alert. Options include confirming all alerts of a given type and choosing *Change Configuration* to change the configuration of the alert. The advantage of this approach is that you can see both the current status in the *Current* column and the worst value in the *Worst* column.
- **Alert Groups** (usually the lower area): When you select an aggregated alert, all exceeded thresholds are displayed in the lower area with the date, time, and details of the selected alert.

You can enter yourself as processor here (*Assign*) or enter a comment (*Add Comment*). The *Start Date/Time* column indicates the first time the alert was reported, while the *End Date/Time* column shows the last time it was reported.

You can also use all the available analysis methods to analyze the alert and solve the problem (with the button *Show Details* you jump to the *Alert Details*).

- **Alert Details**

To display the alert details, you can choose *Display Details* or, as mentioned above, set the *Alert Details* field on the upper right to *Embedded*.

The *Central Alert Inbox* figure also displays alert details. The upper buttons (*Confirm*, *Assign*, and so on) have the same functions as described above.

When you click the icon to the right of the header, alert description and analysis, you open the section for the analysis tools that are available for examining the cause of the alert in detail. For example, you could use the *E2E Workload Analysis*, transaction SM50, or transaction SM04. Of course, you can also add customer analysis tools within the alert configuration.

The *Alert Details* section contains all the metrics from which the alert was defined, along with the status of each individual metric. You also have the option of selecting a row to display the details of the metric or clicking the icon in the *Type* column to open the **Metric Viewer**. From the *Alert Details* you are also able to show specific metrics for a longer time period using the **Metric Viewer**.

You can configure the interval for automatic updates (such as never, every 5 minutes, and so on) in the *Automatic Update* field. You can also display the alert details within the content area. To do so, use the *Alert Details* field, which you set to *Embedded* instead of *New Window*.

You could also use the following cross-application functions from the SAP Solution Manager *Launchpad* (not in the slide):

- **Alert Search:** To search for alerts from different monitoring areas. For example if somebody confirms an alert and you want to know who and why.
- **Alert Reporting:** View information about the alerts related to Application Operations, grouped by different categories and status.

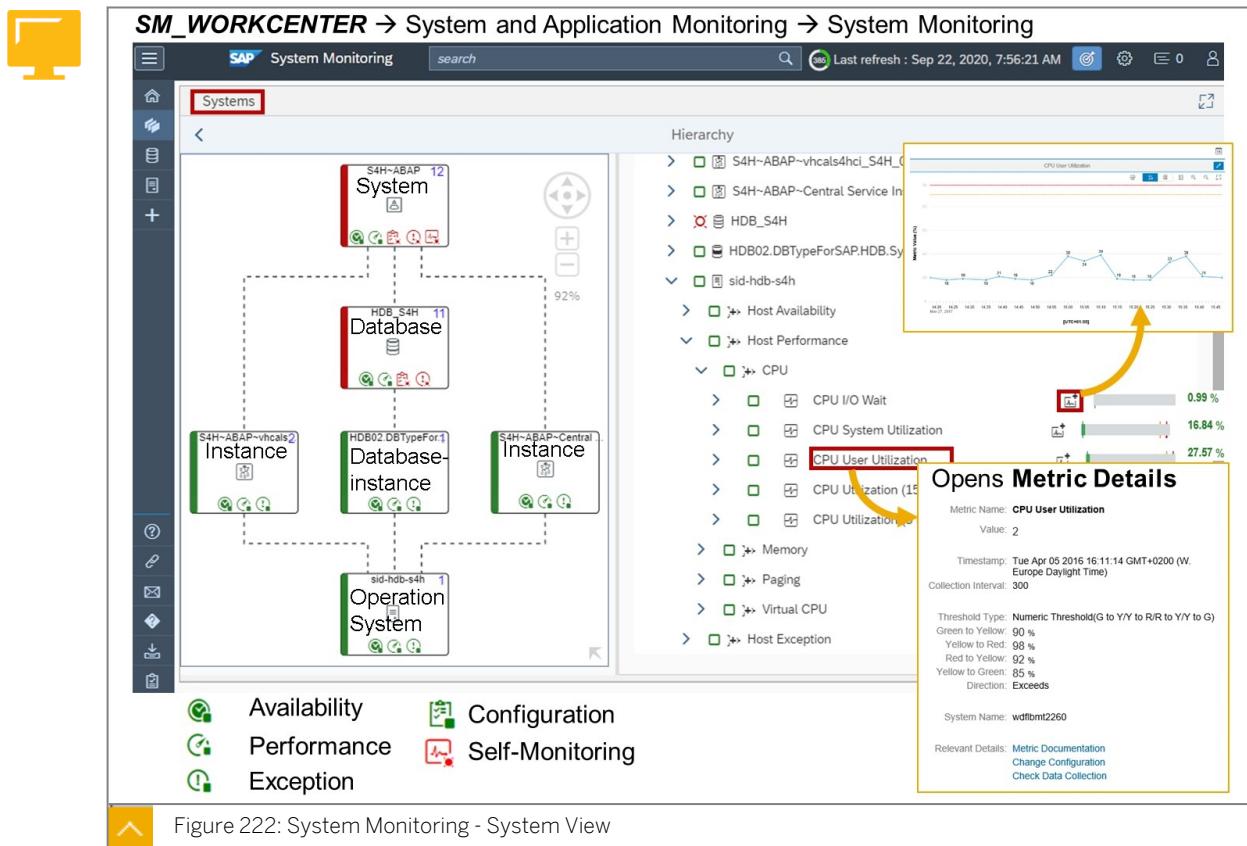
Use the System Monitoring Application

System Monitoring

System Monitoring tile shows the status of the systems, hosts, and databases in the SAP Solution Manager system landscape

There are different sections on the *Overview page*: *System Overview*, *Host Overview*, and *Database Overview*. There, you can view the total number of systems, hosts, or databases in the landscape. You can also view the status of the systems, hosts, or databases, based on their rating. Systems with grey status are displayed as per user preference. Otherwise, the grey systems are not displayed by default.

In *System Monitoring*, you can also display the status of the selected systems graphically as shown in the screen shot.



In the **System Hierarchy**, on the left-hand side of the screen it displays the hierarchy view, which shows the current status of the **system**, the **instance**, the **database**, the **database instance**, and the **operating system**. The status of the corresponding alert group is displayed as an icon and in the colors green, yellow, red, and gray. The colors have the following meanings:

- Red: Error
- Yellow: Warning
- Green: Neither error nor warning
- Gray: No data available

The right side of the screen contains the event and metrics view. From here, you can view the individual metrics of all managed objects in the sort sequence of the alert group. The last respective measurement point of each metric is displayed.

When you choose a metric name, this opens the **metric details**, where you can also display the metric documentation, change the configuration of the metric, or check the data collection.

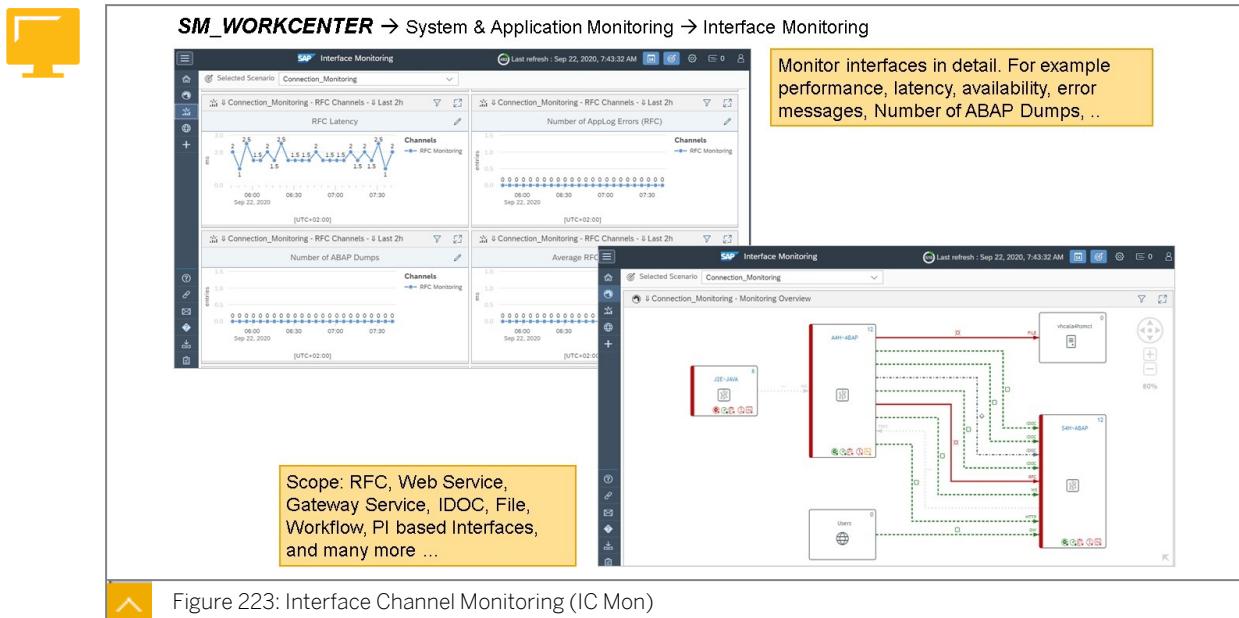
Choosing the icon before or after the metric name opens the **metrics monitor**. The metrics monitor displays the selected metric for a longer time period. You can choose a measurement point to display its details. You also have the option of configuring the required period with the slider at the bottom.

Describe the Interface Monitoring Application

Interface Monitoring

Interface and Connection Monitoring tile allows the monitoring of the interfaces and connections between business-critical systems.

You can monitor interfaces in detail, for example, the performance of connections between systems or define interfaces on the level of functions called in a system. These function calls may be, for example, a function module (in the case of an RFC connection) or methods called via a web service.



Further information on the configuration of the Monitoring and Alerting Infrastructure (MAI) and using the System and Application Monitoring functions is taught in SAP course *E2E120*.



LESSON SUMMARY

You should now be able to:

- Explain the technical prerequisites to use Monitoring and Alerting Infrastructure (MAI)
- List the System and Application Monitoring Use Cases
- Understand the cross-application Alert Functions
- Use the System Monitoring Application
- Describe the Interface Monitoring

Unit 10

Lesson 3

Tracing and Logging in the SAP System

LESSON OVERVIEW

In this lesson, you will learn about the different trace and logging options in the SAP system. You will also perform and evaluate a trace yourself.

Business Example

An unexpected, reproducible error situation occurred in your SAP system. As a system administrator, you want to find the cause of the error.



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- List the options available for tracing and logging in the SAP system
- Perform simple traces in the SAP system

Traces and Logs

You can follow the process of various operations in your SAP system with trace functions. This allows you to monitor the system and isolate problems that occur.

Traces and Logs – Test and Definition

Trace

In the SAP context, a trace usually refers to an option for logging system processes that users can activate and deactivate. It may also be possible to configure their level of detail.

A good example: System trace, transaction ST01.

Log

In the SAP context, a log is typically a continuously recorded information flow that records certain events. These logs can be managed inside or outside the SAP system.

A good example: System log, transaction SM21.

There are functions that combine the properties of traces and logs, such as developer traces.

There are many trace options in SAP systems. The main ones are listed in Trace and Log Options.

Trace and Log Options



- System log (SM21)
- Dump analysis (ABAP runtime error: ST22)
- System trace (ST01)
- Performance analysis (ST05)

- Error log files (developer trace: ST11)

Trace and Log Commands

You can use transaction SM21 to determine and correct errors that occur in your system and its environment. SAP application servers record events and problems in system logs. Every SAP application server has a local log that contains the messages, such as errors, warnings, and information, logged by this server.

If unpredictable errors occur at runtime when you call an ABAP program, a runtime error that generates a short dump is logged (transaction ST22).

If you want to record the internal SAP system activities, such as authorization checks, database accesses, kernel functions, and RFC calls, use the *System Trace* function (transaction ST01).

You can use the *Performance Trace* function (transaction ST05) to record database calls, lock management calls, and remote calls of reports and transactions in a trace file, and display the logged measurement results as lists. The *Performance Trace* also offers extensive support for a detailed analysis of individual trace records. You can find all the functions of the *Performance Trace* in the *System Trace* as well. The *Performance Trace* is a more suitable analysis tool for certain problems, since the reduced scope of functions makes it easier to handle.

Technical information about internal SAP problems is logged in the developer traces (transaction ST11).

System Log (SM21)

Events and problems are recorded locally on each application server and can be displayed in the system log (syslog) in the SAP system, as shown in the following figure.

Syslog messages

SM21

Syslog of instance s4dhost_S4D_11

Date	TIME	Instance	Type	Process No	No Cl.	User	Priority	Message ID	Message Text
04.10.2021	10:22:10	s4dhost_S4D_11	RD	000			● QOR		Failed to call function GwRead; rc: -006
04.10.2021	11:22:05	s4dhost_S4D_11	DP	000			● QOI		Operating system call recv failed (error no. 104)
04.10.2021	11:25:29	s4dhost_S4D_11	RD	000			● QOI		Operating system call recv failed (error no. 104)
04.10.2021	12:02:36	s4dhost_S4D_11	RD	000			● QOI		Operating system call recv failed (error no. 104)
04.10.2021	12:02:37	s4dhost_S4D_11	DP	000			● QOI		Operating system call recv failed (error no. 104)
04.10.2021	12:34:26	s4dhost_S4D_11	RD	000			● QOI		Operating system call connect failed (error no. 111)
05.10.2021	01:00:27	s4dhost_S4D_11	BTC	035	000	DDIC	○ ABD		Short dump deleted with default tool RSSNAPDL
05.10.2021	23:44:40	s4dhost_S4D_11	BTC	036	000	DDIC	● QOW		User DDIC is deleting RFC destination TMC#ABIFMBCOQUPOZCHCGCZ2C2BYUA
06.10.2021	01:00:28	s4dhost_S4D_11	BTC	038	000	DDIC	○ ABD		Short dump deleted with default tool RSSNAPDL
06.10.2021	05:06:37	s4dhost_S4D_11	DIA	006	000		▲ BYM		SQL error -10807 occurred; work process in reconnect status
06.10.2021	05:06:37	s4dhost_S4D_11	DIA	006	000		○ R47		Delete ABAP session T1_U24266_M0 [Info = Error Message [Error/Core]

System Log Location:
/usr/sap/<SID>/D\$\$/log/SLOG\$\$

<SID>: SAP system ID
\$\$: instance number of the application server

File System

Figure 224: System Log

**Note:**

In the context of the system log, see also the following SAP Notes:

- SAP Note [712706](#): Program RSLGVIEW - reading the SAP system log without system
- SAP Note [28665](#): Central syslog under NT.

This SAP Note provides a solution that uses CCMS monitoring functions as well as a feature for viewing the logs for all instances of a system.

You can also use the SAP Microsoft Management Console and the SAP Management Console to view the system log, even if the instances in question were not started or were started and failed.

You display a system log with transaction SM21. By default, the system reads the log for the last one to two hours. As well as the local system log, you can display system logs for other application servers. On the initial *Display the system log* screen, you can specify for which instances the syslog should be displayed. The syslog messages will then be displayed for all selected instances.

You can define the path and file names for local log files with the following system profile parameters:

- *rslg/local/file*: File name for the local log (standard: SLOG<SAPSYSTEMNUMBER>)

By default, the log files for the local system log are stored in the following directory: */usr/sap/<SID>/<instance_directory>/log*.

The log file has a maximum size that can be configured with profile parameter *rslg/max_diskspace/local*. The oldest entries will be overwritten as soon as the limit is reached.

**Hint:**

The system does not display a message when an old log file is overwritten.

Dump Analysis (ST22)

ABAP programs are checked statically when they are created and dynamically when they are running. Errors that are not statically predictable and only occur at runtime are identified dynamically by the ABAP runtime environment. States of this type lead to exceptions. If an exception is not handled or cannot be handled, a runtime error occurs. When a runtime error occurs, the ABAP runtime environment terminates the execution of the program, generates a so called short dump, and branches to a special screen for analyzing the short dump. You can access short dumps using transaction ST22.

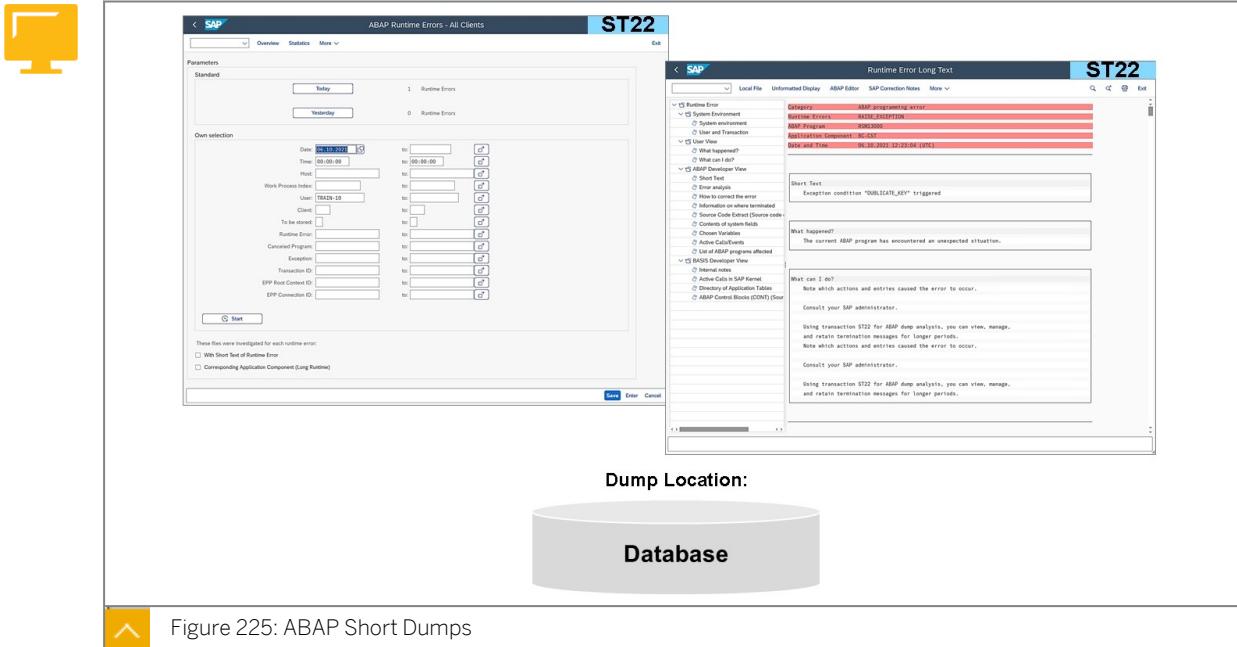


Figure 225: ABAP Short Dumps

A short dump is divided into different sections that document the error. The overview shows what other information is output in the short dump, such as contents of data objects, active calls, control structures, and so on. You can branch to the ABAP Debugger at the termination point from the short dump view. The following different error situations exist:

- Internal error

The kernel identifies an error state. In this case, send a message to notify SAP.

- Installation and environment/resource error

In this case, an error occurred that was caused by incorrect system installation or missing resources (such as the database being shut down).

- Error in application program

Typical causes of errors are:

- Content of a numerical field not in the correct format
- Arithmetic overrun
- An external procedure is not available
- Type conflict when transferring parameters to an external procedure

By default, short dumps are stored in the system for 28 days. You can delete short dumps in accordance with a time specification using the reorganize function, which you can call by choosing *Goto → Reorganize*. You can save a short dump without a time limit using the keep function, which you can choose from the *Detail View* under *Short Dump → Keep/Release*.

If problems that you cannot solve yourself occur with ABAP programs, you can send an extract of the short dump to SAP. A short dump is an important basis on which the SAP Hotline and remote consulting solve problems.

Important Features of Dump Analysis



- If a runtime error occurs, a short dump is generated. You can use transaction ST22 to analyze the short dump.
- Dump data is stored in the database.
- Dump data can be reorganized.
- Individual short dumps can be flagged for retention.

System Trace (ST01)



Trace Location:
/usr/sap/<SID>/D\$\$/log/TRACE

<SID>: SAP system ID
 \$\$: instance number of the application server

File System

Figure 226: System Trace and Performance Trace

You can use the SAP system trace (system trace for short) to record internal system activities. You can call the system trace using transaction ST01. You can also use transaction ST01 to display the non-actively displayed trace file.



Hint:

For system monitoring and problem analysis, we recommend that you use the system log or the developer trace.

The System Trace Is Used to Analyze the Following:



- Authorization checks
- Kernel functions

- Kernel modules
- DB accesses (SQL trace)
- Accesses to table buffers
- RFC calls, also *RFC Trace*
- HTTP calls
- Lock operations (client-side), also *Enqueue Trace*

You select the components to be logged on the initial screen. You can use the system trace for tracing authorization checks. If the trace is activated for the authorization check, all authorization checks performed by the system are recorded. During the evaluation, you can identify which authorizations the system checked at which times. The following detailed information is also provided: Date, time, work process number, user, authorization object, program, line, number of authorization values, and authorization values.



Hint:

For tracing authorization checks exclusively, you can use transaction STAUTHTRACE.

You can use the SQL trace to follow how the Open SQL commands in reports and transactions are converted to standard SQL commands and the parameters with which the SQL commands are transferred to the database system in use. The results of the SQL command are also logged, such as the return code and the number of records found, inserted, or deleted by the database. Logging the execution time and the call point in the application program allows you to perform more advanced evaluations.

With the enqueue trace, you can follow which lock instructions the SAP system performs on which lock objects, and which parameters the system uses for these locks. The program that triggered the lock, the owner of the lock, and the time that the enqueue server required to release the lock again are all also logged in the trace file.

You can use the RFC trace to follow which remote calls the SAP system executes, and the instance on which these calls are executed. From the trace recording, you can see which function modules were called remotely by the program to be analyzed and whether the RFC call was successfully executed. The total time required for the execution of the remote call and the number of bytes sent and received during the RFC are also logged in the trace file.

Performance Trace (ST05)

The performance trace is used for analyzing the following:



- Database calls (*SQL Trace*)
- Lock management calls (*Enqueue Trace*)
- Accesses to table buffers (*Buffer Trace*)
- Remote calls of reports and transactions (*RFC Trace*)
- HTTP communication (*HTTP Trace*)
- Single SQL statements (*Enter SQL Statement*)

The performance trace provides similar trace options to the system trace. It allows you to record database calls, calls to lock management, calls to table buffers, and remote calls of reports and transactions from the SAP system itself in a trace file. You can call the performance trace using transaction ST05.



Hint:

The functions of the performance trace are integrated into the ABAP Workbench and can therefore be called from there.

Developer Trace (ST11)

Developer traces are recordings that contain technical information and are used if errors occur. This type of process trace is especially useful to investigate host and internal SAP problems that are affecting your SAP system. Developer traces `dev_*` are written to files in the directory `/usr/sap/<SID>/<instance directory>/work` of the SAP application server that generated the trace, as shown in the figure *Developer Traces*.



The screenshot shows the SAP ST11 developer trace interface. The main window displays a list of trace files with columns for Usable, Viewed, Changed File name, Length, Owner, and Lastchange. Below the table, a large diagram illustrates the trace location path. A curved arrow points upwards from a 'File System' icon to a text box containing the path: '/usr/sap/<SID>/D\$\$/work/dev_*'. A legend at the bottom defines the symbols: '<SID>' as SAP system ID, '\$\$' as instance number of the application server, and the colored bars as different components of the path.

Trace Location:
`/usr/sap/<SID>/D$$/work/dev_*`

<SID>: SAP system ID
\$\$: instance number of the application server

File System

Figure 227: Developer Traces

You can access the developer traces in the operating system, in transaction AL11, transaction ST11, or transaction SM50 (work process overview). In transaction SM50, you can switch to the individual `dev_*` traces by choosing *Process → Trace → Display File*. You can display additional details in the displayed traces by expanding individual entries.

Trace File Configuration

You can use system profile parameters to restrict the size of the trace files and to specify an appropriate path.

The SAP system trace writes the trace data in trace files. For performance reasons, this is not done directly, but instead using a process-internal buffer. The profile parameter *rstr/buffer_size_kB* determines the size of this buffer. The SAP trace stores data in multiple files, which are written sequentially. The parameter *rstr/filename* defines the base name of these files. There is always one file with exactly this name. If the file is full (parameter *rstr/max_filesize_MB*), the file is renamed and a new file is created with the base name. When the file is renamed, a number between 00 and 99 is added to the file name. The parameter *rstr/max_files* determines the maximum number of files. If this value is exceeded, the files are overwritten.



LESSON SUMMARY

You should now be able to:

- List the options available for tracing and logging in the SAP system
- Perform simple traces in the SAP system

Troubleshooting Procedures

LESSON OVERVIEW

This lesson describes a general procedure for troubleshooting.

Business Example

Unforeseen problems can occur during the operation of your SAP systems. As the system administrator, you want to find out about procedures for structured troubleshooting.



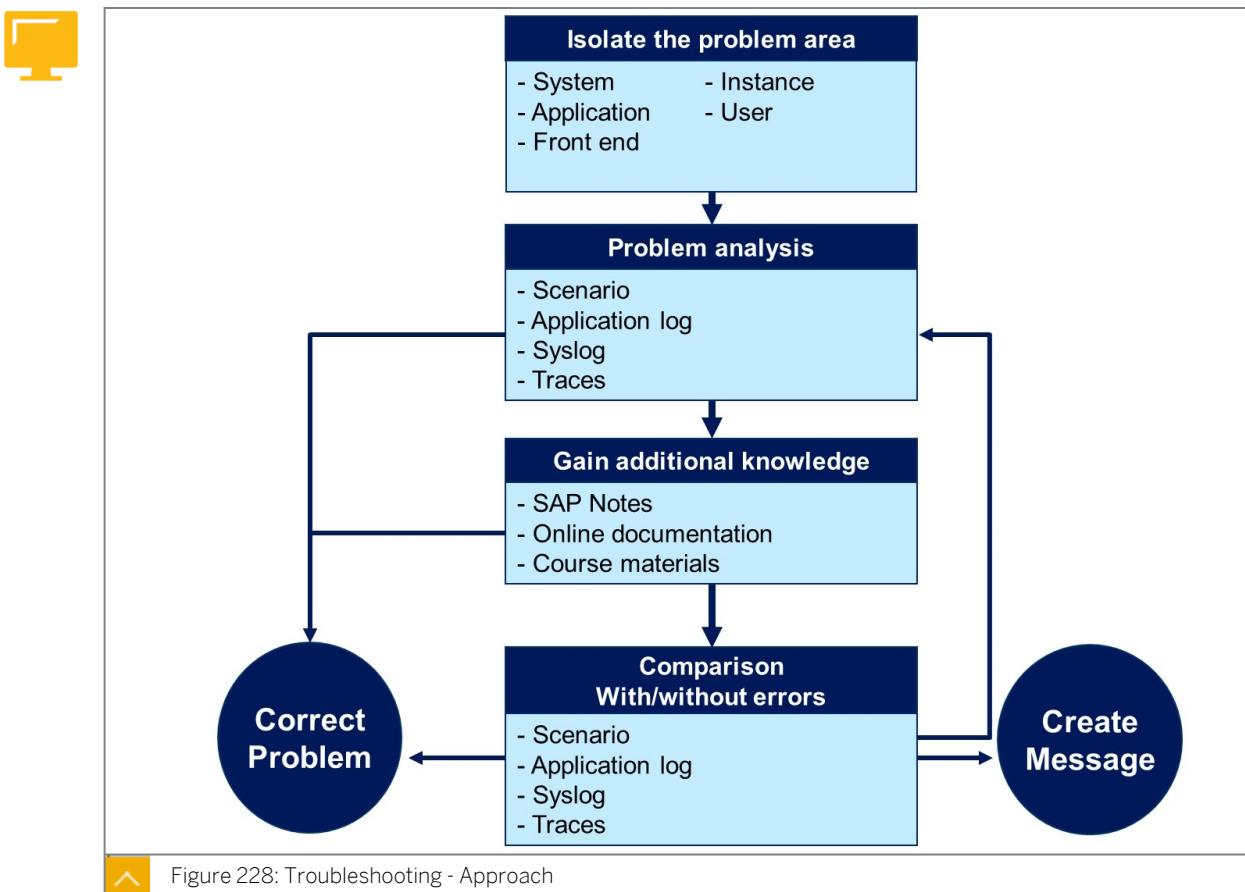
LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Develop procedures for structured troubleshooting

General Principles for Troubleshooting

Errors, by their very nature, always occur in places where they should not occur. Consequently, we can only present a general approach here.



The figure *Troubleshooting: Approach* lists steps for a possible troubleshooting approach.

Isolate the problem area: First, attempt to isolate the error. Where did it occur, when did it occur, and in what context did it occur? If an end user in the SAP system complains, that their output could not be printed, the problem description would be too imprecise. “Front-end printing on front end xyz does not work with any SAP system” is more exact. If you also know that front-end printing works on other front ends, you have already isolated the problem.

Problem analysis: Check the scenario to find out whether all required settings, and so on, are correct. Check the application logs, the system log, and the traces (the developer traces will usually be helpful here) to see whether they provide any clues for correcting the error.

Gain additional knowledge: To interpret the results from the first problem analysis, it is, of course, necessary for you to be familiar with the processes and functions of the area in which the error has occurred. If your experience and previous knowledge are insufficient, you can search for SAP Notes on the SAP Support Portal with the keywords from the system log or the trace files. You may find a solution there, or additional information that helps you find and correct the error. If you do not find any suitable SAP Notes or search terms, search for composite SAP Notes for the topic area (for example, with the terms *front-end printing* and *composite SAP Note*). For additional background information about the topic area, see the online documentation and course materials. If you still cannot solve the problem with this information, compare the process with errors against an error-free process.

Compare error-free and erroneous processes: You can use this approach to determine where there are differences between an erroneous and an error-free process. This information helps you to further isolate the problem area and may help you to solve the problem or to perform new, more targeted problem analyses. If it is not possible to perform another problem analysis, contact the SAP Support by creating an incident. Enter the information from your troubleshooting (such as a trace and/or system log information) when doing so.



Note:

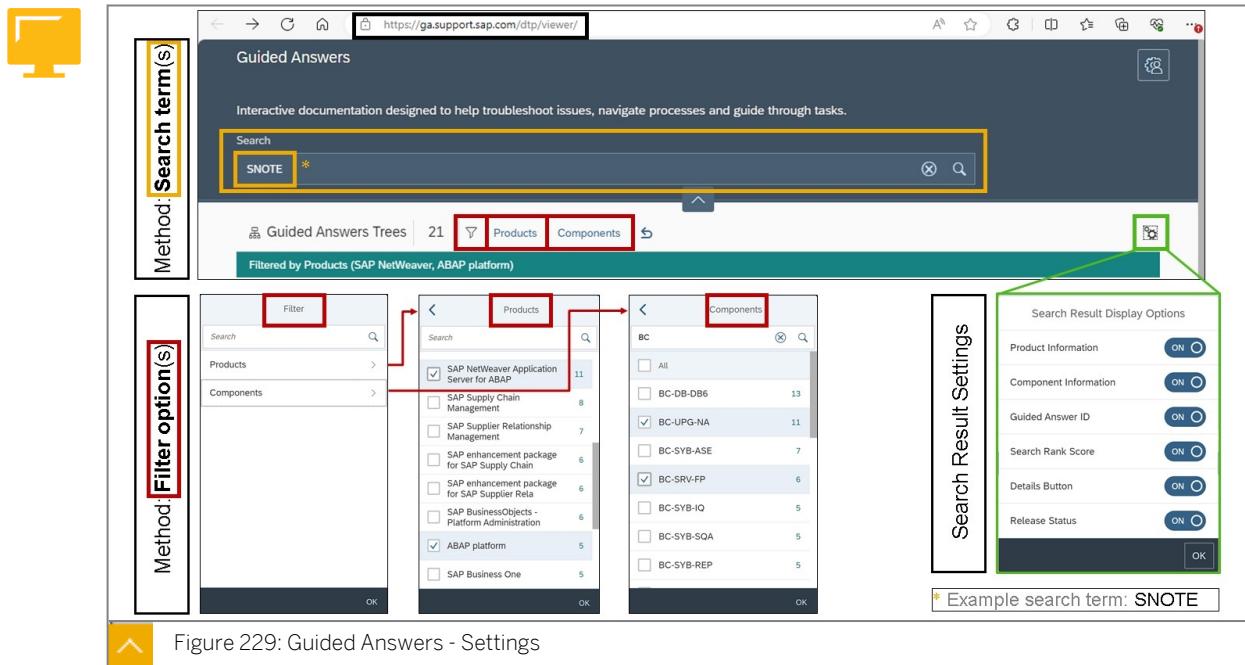
You can use the **support log assistant** self-service tool for automated problem analysis of, for example, the configuration or log and trace files. The tool allows administrators to automatically analyze support-related files right from a browser. It uses the knowledge of SAP product support experts and connects known issues found in log, trace and other text-based files with solutions and troubleshooting steps.

For more information about the support log assistant tool, see SAP Note:
[2838708 – Using the Support Log Assistant to automate support-related file analysis](#).

Use Guided Answers for Troubleshooting

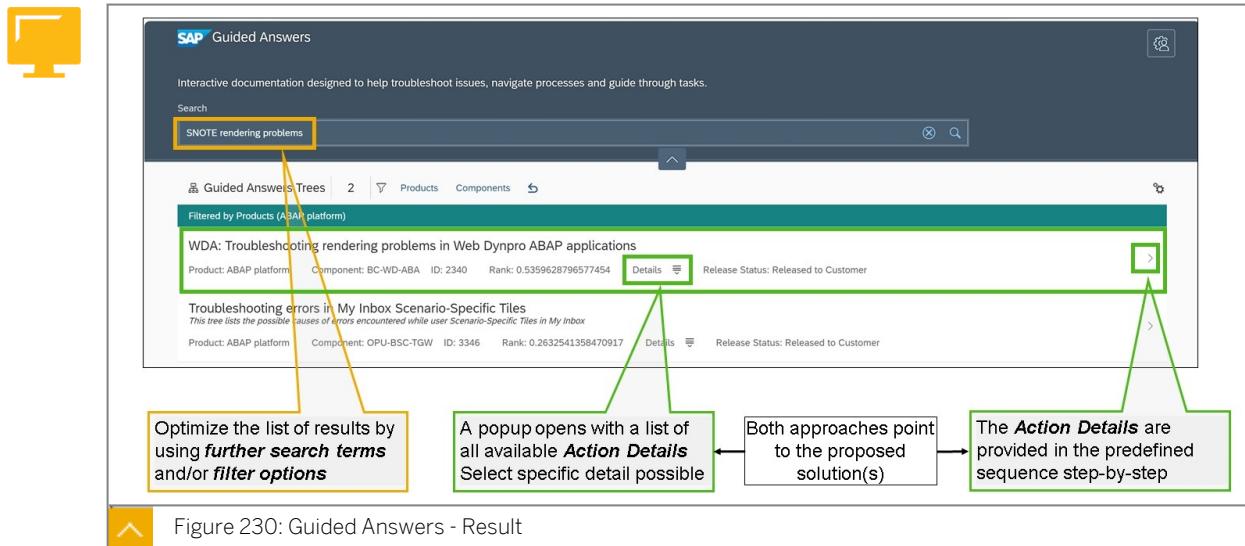
SAP provides a special web-based function called *Guided Answers* to support customers and partners with an interactive documentation. It is designed to help troubleshoot issues, navigate processes and guide through tasks.

The start page provides a *Search* field and is complemented by *Filter* options based on *Products* and *Components*. As a result of a search a *Guided Answers Trees* will be shown, which can be customized by using *Settings*.



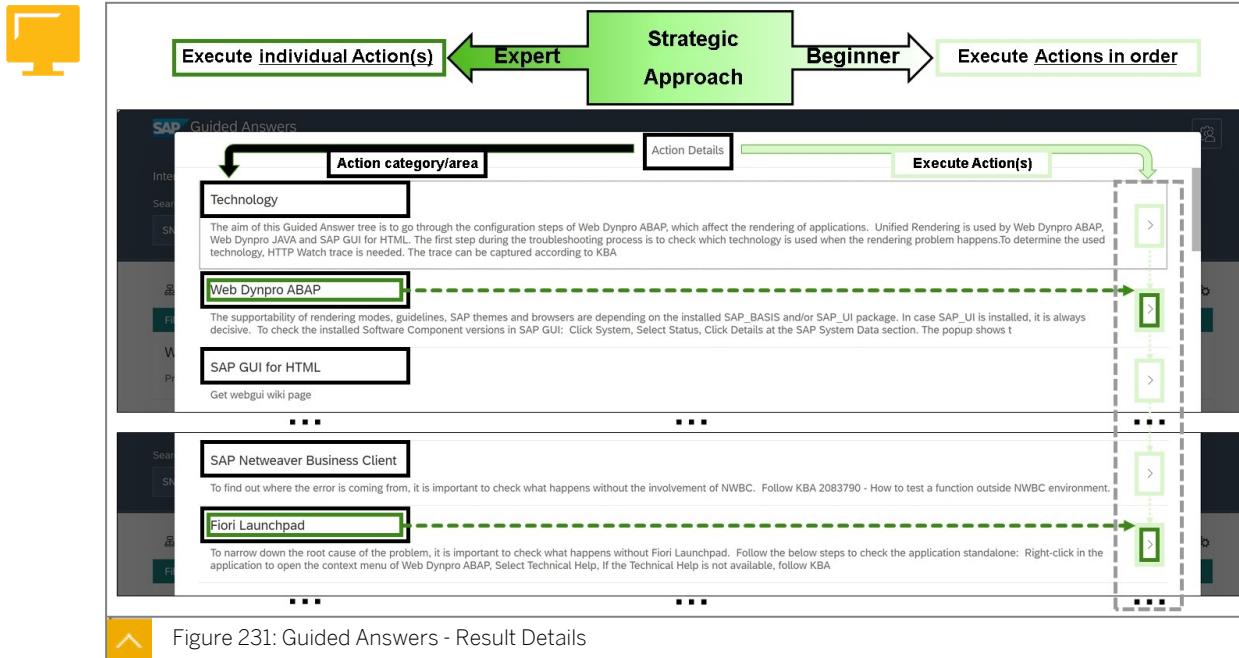
In case of a specific troubleshooting issue, use the *Guided Answers* function by providing search terms which best fit and describe the given problem. Depending on the amount of results, further filter options increase the accuracy concerning the problem and help to identify the relevant solution.

The output of a search is called *Guided Answers Trees*. By default the name of the *Guided Answer* is shown, as well as the *Product Information*, the *Component Information*, and the *Release Status*. In addition, the *Guided Answers ID*, the *Search Rank Score*, and the useful *Details Button* can be activated, individually by the user.



For experts the *Details Button* may be useful, because a specific selection of an action can be done. Beginners may want to be guided through the actions in a recommended sequence. This is done with a kind of roadmap. All actions documented by SAP to analyze and solve the problem are listed. The customer or partner runs through and answers the predefined questions by analyzing, verifying, and/or configuring specific topics. In this way you are guided through the problem-solving solutions.

At the end, the same Q&A is covered for beginners and experts. It is only a question of what you prefer, being guided through with the advantage that all relevant topics are taken into account, or selecting specific topics based on your experience.



Within the results list there are two main approaches to check the proposed solution(s):

Strategic Approach for Beginners

Use the small *arrow icon* at the right hand side of the corresponding result, which guides you step-by-step through the proposed solution(s) in a predefined sequence with links to the SAP online documentation, SAP Notes, SAP-recommended documents, and Q&A for further guidance.

Strategic Approach for Experts

Use the *Details button* (hamburger icon with an arrow) below the title of the respective *Guided Answer* in the results list. This function opens a popup listing (in most of the cases) different potentially problem-solving steps. This enables a specific selection of steps based on the given issue or your experience. For example, you can skip steps, which you have already checked before. Take care about the necessary display settings mentioned below.



Hint:

A prerequisite to visualize the *Details button* is the activation of the respective setting in the *Search Results Display Options*.

Figure 232 displays three examples of SAP Fiori pages illustrating the guided answer feature. The top screenshot shows a general troubleshooting guide for Web Dynpro ABAP applications. The middle screenshot shows a specific troubleshooting guide for Web Dynpro ABAP, including a screenshot of the SAP Fiori Launchpad. The bottom screenshot shows a troubleshooting guide for theming with Fiori guidelines, also including a Fiori Launchpad screenshot. Callouts explain the interface:

- Guided Answer**: Points to the main title of the page.
- Action answer(s)**: Points to a blue box containing a list of actions.
- Action no. and explanation**: Points to a numbered callout (1) with an explanation.
- Further Actions**: Points to a dashed box labeled "K E Y".
- Action sequence**: Points to a sequence of numbered steps (1, 2, 3, 4, 5, 6, 7) in a callout.

Figure 232: Guided Answers - Example

Every *Action* provides information as text, sometimes enhanced by screenshots and/or links. At the end of the explanation there is a question, which provides predefined answers. By selecting an answer (using a pushbutton), a follow-up *Action* is provided. In this way a complete chain of Q&A is build, which even allows to use *Back one step* or *Start over*. Moreover, an overview of the already performed steps is given at any time with the help of the navigation pane (called *command-field*) at the left.

Further information on how to use the function *Guided Answers* in an effective way, can be retrieved by attending the following *Learning Journey* online: [Demonstrating Effective Guided Answers Problem-Solving](#)



LESSON SUMMARY

You should now be able to:

- Develop procedures for structured troubleshooting

Learning Assessment

1. What can you do with transaction RZ20?

Choose the correct answers.

- A Backup database
- B Update data
- C Monitor the database and the SAP system
- D Configure and monitor the firewall

2. With which of the following transactions can you activate a trace for SQL statements in the SAP system (SQL trace)?

Choose the correct answers.

- A Performance analysis
- B System log
- C System trace
- D DBA Cockpit

3. Where is the information on the current state of the SAP system, shown in transaction RZ20, stored?

Choose the correct answer.

- A In the database.
- B In the monitoring segment.
- C In files on operating system level.
- D In local memory of the work processes.

4. What components does SAP Solution Manager use to collect monitoring data?

Choose the correct answers.

- A SAP Web Dispatcher
- B SAP Management Console (SAP MC)
- C Diagnostics Agent
- D CA Introscope

5. Identify functions in the AS ABAP that aim at analyzing error situations.

Choose the correct answers.

- A System Log
- B Dump Analysis
- C The Event Log
- D The System Trace
- E Transaction SNOTE

6. When you encounter an error situation in an SAP system, it is recommended that you involve SAP Support as soon as possible.

Determine whether this statement is true or false.

- True
- False

Learning Assessment - Answers

1. What can you do with transaction RZ20?

Choose the correct answers.

- A Backup database
- B Update data
- C Monitor the database and the SAP system
- D Configure and monitor the firewall

You are correct! You can use system monitoring in transaction RZ20 to monitor SAP systems and their databases. Read more on this in unit 10 of the course ADM100.

2. With which of the following transactions can you activate a trace for SQL statements in the SAP system (SQL trace)?

Choose the correct answers.

- A Performance analysis
- B System log
- C System trace
- D DBA Cockpit

You are correct! You can analyze SQL statements by activating the trace in transaction ST01 (system trace) or ST05 (performance analysis). Transaction SM21 (SysLog) is the system log. The DBA COCKPIT lets you analyze the database in great detail, but does not feature an SQL trace. Read more on this in unit 10 of the course ADM100.

3. Where is the information on the current state of the SAP system, shown in transaction RZ20, stored?

Choose the correct answer.

- A In the database.
- B In the monitoring segment.
- C In files on operating system level.
- D In local memory of the work processes.

You are correct! The information shown in transaction RZ20 on the current system status is being stored in the monitoring segment. Read more on this in the lesson “Using the Monitoring Architecture of the Computing Center Management System” of the course ADM110.

4. What components does SAP Solution Manager use to collect monitoring data?

Choose the correct answers.

- A SAP Web Dispatcher
- B SAP Management Console (SAP MC)
- C Diagnostics Agent
- D CA Introscope

You are correct! SAP Solution Manager uses, among others, the diagnostics agent and CA Introscope to collect monitoring data. SAP Web Dispatcher and SAP MC are not used by SAP Solution Manager to collect monitoring data. Read more on this in the lesson “Explaining the Concept of Monitoring Tools in SAP Solution Manager” of the course ADM100.

5. Identify functions in the AS ABAP that aim at analyzing error situations.

Choose the correct answers.

- A System Log
- B Dump Analysis
- C The Event Log
- D The System Trace
- E Transaction SNOTE

You are correct! The System Log, the Dump Analysis and the System Trace can help you to analyze error situations in AS ABAP. The Event Log is a function of the operating system Windows and transaction SNOTE can be used to remedy software errors, but it can't be used for analysis. Read more on this in the lesson “Tracing and Logging in the SAP System” of the course ADM100.

6. When you encounter an error situation in an SAP system, it is recommended that you involve SAP Support as soon as possible.

Determine whether this statement is true or false.

True

False

You are correct. Usually, SAP recommends that you use the tools and functions at your disposal to try to solve the situation yourself. Read more on this in the lesson "Troubleshooting Procedures" of the course ADM100.

Glossary

ABAP

Advanced Business Application Programming. SAP's programming language.

archive information structure

A central element in the SAP Archive Information System (AS). Using data fields, it enables you to find and display archived data.

archiving object

Logical object of related business data in the database that can be read from the database using a write program and then deleted from the database by the associated delete program after successful archiving.

Buffer

Area in the main memory of an instance in which data frequently used by applications can be temporarily stored.

business object

Representation of a business object in the real world, such as a purchase order. SAP business objects describe entire business processes. This type of encapsulation reduces the complexity, since the internal structure of the business object remains hidden.

CPI-C

The "Common Programming Interface for Communication" describes data exchange between different programs. CPI-C can be used to transfer "packaged" data with various technical protocols, such as TCP/IP or LU6.2.

data archiving

Reading data from the database that is no longer required for online operation, and storing this data in an archive file. Archived data is usually deleted from the system. Archiving removes load from the system, meaning that programs, database accesses, and database backups have shorter runtimes due to the lower quantity of data in the database.

Front end

Work center computer.

Message server (ABAP)

Standalone program that contains a list of all Application Servers that are part of the SAP system. In the case of a logon using logon groups, the message server determines the Application Server

that a user logs on to and organizes the communication between the Application Servers.

modification adjustment

If SAP objects were modified by the customer and then imported with a Support Package, or a release upgrade, the objects have to be adjusted when they are imported. If these objects are not adjusted, the customer's modifications that have been made to the SAP objects are reset to the new SAP delivery status.

Output request

An output request contains the data from a spool request in a format for a particular printer model.

Printing Assistant for Landscapes (PAL)

The Printing Assistant for Landscapes (PAL) lets you maintain output devices in one (central) system and then distribute them to other (target) systems using RFC destinations.

SAP Archive Information System

Generic tool fully integrated in data archiving for searching SAP data archives. The search and display of data is based on archive information structures that the user can define and fill with data from the archive.

SAP ArchiveLink

A service integrated into the SAP system technology that creates a connection between stored documents and related application documents (business objects) in the SAP system.

SAP Easy Access

SAP Easy Access is the standard initial screen in SAP systems. The left side of the screen displays the available menu in a tree structure. You can display your company's logo on the right side.

SAP GUI

SAP Graphical User Interface; software that the user can use to exchange information with the SAP system. With the user interface, you can select transactions, start programs, display files, and choose other options by pressing function keys or pushbuttons or by selecting menu options.

SAP GUI for Java

Platform-independent SAP GUI, requires a Java environment.

SAP GUI for Windows

Implementation of SAP GUI in a Microsoft Windows environment.

SAP Logon

When it is started, the SAP Logon program (only available for Microsoft Windows) provides a list of SAP systems for which the logon process can be started. This list is generated from the information in configuration files on the front end, which is usually preconfigured centrally and provided to end users. The SAP Logon program also allows logon load balancing across the resources available for the selected SAP system at logon.

SAP Note

SAP Notes enable you to make adjustments due to changes in legal requirements, correct errors in the SAP standard software, and enhance existing functions or make new functions available. An SAP Note usually involves changes to an individual object. Also see "Support Package".

SAP system upgrade

A SAP system consists of different software components with a specific release level. With a SAP system upgrade you exchange the existing software components with newer ones.

SPAM/SAINT update

100% patch for the Support Package Manager (transaction SPAM) and the Add-On Installation Tool (transaction SAINT). You use the SPAM/SAINT update to update these two tools.

Spool request

A spool request contains information about data to be output, its formatting, and the printer model used.

Support Package

Support Packages enable you to make adjustments due to changes in legal requirements, correct errors in the SAP standard software, and enhance existing functions or make new functions available. A Support Package contains a set of changed objects. Also see "SAP Note" and "Support Package stack".

Support Package Manager (SPAM)

A tool (transaction SPAM) for importing Support Packages. The Support Package Manager is also used to update this tool itself. Also see "SPAM/SAINT update".

Support Package stack (SPS)

A Support Package stack (SPS) is a recommended target combination of Support Packages and other patches for an SAP system. Also see "Support Package".

system log

Analysis option for errors in the system and its environment.

Table

Tabular array of data in the ABAP Dictionary. A table consists of columns (data values of the same type) and rows (data records). Each record can be identified uniquely by one or more fields.

TemSe

An abbreviation for Temporary Sequential database; a file that temporarily stores data from spool requests and job logs from background processing, among other information.

User Master Record

The user master record contains the definition of a user in the client. Examples of these fields include last name, first name, initial password, phone number, and so on. The user master record is used to create a user context (see this entry) when a user logs on to the system.

Variant

Pre-assignment of the input fields of a selection screen for reports, for example, to enable the execution of the report in the background.