mySAP™ Business Suite

# SAPTEC

# SAP NetWeaver: Fundamentals of the Application Platform

Date ______________________

Training Center ______________________

Instructors ______________________

______________________

Education Website ______________________

## Instructor Handbook

Course Version: 2003 Q2
Course Duration: 3 Day(s)
Material Number: 50062876
Owner: Christoph Maerker (D038226)

*An SAP Compass course - use it to learn, reference it for work*

## Copyright

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

## Trademarks

- Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.
- IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.
- ORACLE® is a registered trademark of ORACLE Corporation.
- INFORMIX®-OnLine for SAP and INFORMIX® Dynamic ServerTM are registered trademarks of Informix Software Incorporated.
- UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.
- Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.
- HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- JAVA® is a registered trademark of Sun Microsystems, Inc.
- JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- SAP, SAP Logo, R/2, RIVA, R/3, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

## Disclaimer

THESE MATERIALS ARE PROVIDED BY SAP ON AN "AS IS" BASIS, AND SAP EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR APPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THESE MATERIALS AND THE SERVICE, INFORMATION, TEXT, GRAPHICS, LINKS, OR ANY OTHER MATERIALS AND PRODUCTS CONTAINED HEREIN. IN NO EVENT SHALL SAP BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND WHATSOEVER, INCLUDING WITHOUT LIMITATION LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS OR INCLUDED SOFTWARE COMPONENTS.

# About this Handbook

This handbook is intended to complement the instructor-led presentation of this course, and serve as a source of reference. It is not suitable for self-study.

## Typographic Conventions

American English is the standard used in this handbook. The following typographic conventions are also used.

| **Type Style** | **Description** |
|---|---|
| *Example text* | Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths, and options.<br>Also used for cross-references to other documentation both internal (in this documentation) and external (in other locations, such as SAPNet). |
| **Example text** | Emphasized words or phrases in body text, titles of graphics, and tables |
| EXAMPLE TEXT | Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example SELECT and INCLUDE. |
| `Example text` | Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, and passages of the source text of a program. |
| **`Example text`** | Exact user entry. These are words and characters that you enter in the system exactly as they appear in the documentation. |
| **`<Example text>`** | Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries. |

## Icons in Body Text

The following icons are used in this handbook.

| Icon | Meaning |
|---|---|
| 💡 | For more information, tips, or background |
| ➡ | Note or further explanation of previous point |
| ⚠ | Exception or caution |
| 1<br>2<br>3 | Procedures |
| 🖥 | Indicates that the item is displayed in the instructor's presentation. |

# Contents

# Course Overview

This course provides an introduction to working with the functions of the SAP Web Application Server. The SAP Web Application Server is the central module of the application platform of SAP NetWeaver. Once you understand and have mastered the functions of the SAP Web Application Server, you understand the technical basis of all mySAP Business Suite solutions that are based on SAP NetWeaver. The **SAPTEC** course is a required prerequisite for attending (and understanding) almost all other training courses in the areas of SAP system administration and (SAP) business programming.

## Target Audience

This course is intended for the following audiences:

- Project team members
- Persons working in the area of system technology, such as system administrators, Technology Consultants, developers in the SAP environment, and so on

## Course Prerequisites

### Required Knowledge

- A basic understanding of data processing

## Course Duration Details

| | |
|---|---|
| **Unit 1:**<br>**SAP Solutions** | |
| mySAP Business Suite | 30 minutes |
| Definition of SAP NetWeaver | 30 minutes |
| **Unit 2: Navigation** | |
| Logon and Screen Design | 30 minutes |
| Calling Functions | 25 minutes |
| Getting Help | 20 minutes |
| Personalizing the User Interface | 20 minutes |
| **Unit 3: The System Kernel** | |
| Client/Server Architecture | 35 minutes |
| Presentation and Database Interface | 30 minutes |
| Structure of an Instance | 35 minutes |
| Dialog Processing | 30 minutes |
| The SAP Transaction | 20 minutes |

| | |
|---|---|
| Enqueue Processing in SAP Systems | 35 minutes |
| Update Processing | 40 minutes |
| Spool Processing | 25 minutes |
| Background Processing | 40 minutes |
| **Unit 4: The Workbench** | |
| Data Structure of SAP Systems | 30 minutes |
| System Landscape and Transports | 30 minutes |
| Accessing and Editing Repository Objects | 55 minutes |
| Working with the ABAP Dictionary | 75 minutes |
| Transporting and Verifying Changes | 25 minutes |
| **Unit 5: Communication and Integration Technologies** | |
| Cross-System Business Processes | 20 minutes |
| Remote Function Calls and BAPIs | 45 minutes |
| SAP Business Workflow | 20 minutes |
| Internet Technologies | 50 minutes |
| Data Migration | 20 minutes |
| **Unit 6: System Administration Tools and Aids** | |
| User Administration | 35 minutes |
| Daily Tasks in System Management | 25 minutes |
| SAP Service Marketplace | 20 minutes |

## Course Goals

This course will prepare you to:

- Describe the basic concepts behind SAP Netweaver and mySAP Business Suite
- Demonstrate your ability to use navigation options in the system efficiently
- Describe the architecture of the SAP Web Application Server and understand its foundations
- Describe the essential features of software development in the SAP environment
- Name and correctly evaluate interface technologies
- Name basic administrative tasks and demonstrate how to perform some of these tasks

## Course Objectives

After completing this course, you will be able to:

- List the solutions offered by SAP
- Navigate in the system

- Describe the system architecture using the appropriate terminology
- Describe the process for development work in the SAP environment
- Name interface technologies in the SAP environment
- Name central administrative functions of SAP systems

## SAP Software Component Information

The information in this course pertains to the following SAP Software Components and releases:

**Caution:** Since this instructor guide can only be updated for new releases, a separate area, **Instructor Guides**, has been set up at http://intranet.sap.com/epf-adm. Up-to-date tips for instructors are collected in a document (SAPTEC_32_IG.doc) in this area. The documents you have in front of you now represent the status as at the end of April 2003. For more recent information, see the supplementary instructor guide document mentioned above.

You must be well prepared to teach this challenging course. Why is this a challenging course to teach?

- Your audience is likely to be very mixed: some participants will be seeing an SAP system for the first time, while others, who have actually worked with SAP R/3 for some time, will have changed roles within their companies, and are therefore attending this “beginners’ course”. Ensuring that all participants attain the same level of knowledge, without “losing” some of them or boring others, is no easy task.
- The other difficulty lies in the very different areas of responsibility of the participants: some are developers, others are administrators, Technology Consultants, or decision makers (who may not yet have decided to buy SAP). It can be tricky to achieve a balance between these different perspectives.
- In addition to that, the course really does require a thorough understanding of the inner workings of the system, particularly in the area of system architecture. You may well find the “Blue Books” a useful tool for preparation: http://ency.wdf.sap-ag.de:1080. Although the blue book “Dispatcher and Taskhandler” was written in 1996, the information it contains is still relevant and of fundamental importance.
- This course also requires instructors to keep up with the latest developments on a broad range of topics. For example, certain terms are no longer valid, but they are still relevant in the context of clarifying the relationship “old term ↔ new term” for the participants.

For example, SAP Basis ↔ mySAP Technology ↔ SAP NetWeaver, or application server ↔ SAP Web Application Server. Since these terms are subject to constant change, it is perfectly possible that even these examples will no longer be valid by the time you read this. You should therefore always read the Development News and check SAPNet regularly.

As part of your preparation for teaching this course, you should attend the following courses: SAP01, ADM100, ADM102, ABAP Workbench Basics, and possibly ADM325. You should also attend SAPTEC itself at least once, though twice would be better.

An additional source of information you could use for your preparation is the online documentation, particularly for the BC-CST, BC-CCM, BC-CTS, BC-SEC, BC-MAS, and BC-ABA areas.

- This course is taught using a copy of the SAP R/3 Enterprise Master System ID3 that is recreated every week. You may find artefact entries from the master system in various places in the training system.
- There is a template user with the name **SAPTEC** and the password **tecsap**. Use this user as a template for copying users for the participants in transaction BC_TOOLS_USER. It provides all the appropriate authorizations for this course. The user name does not need to be unique throughout the system for the purposes of this course.
- There is no longer an optional second instance available for SAP R/3 Enterprise. This means that the course uses a central system.
- The course system is the domain controller of a two system landscape. The second system is virtual and is called CL5. You must not change this setup during the SAPTEC course /- other courses would be negatively affected.
- If lock management is not active, you can use the report *ZSENQON* to reactivate lock management. You should not share this fact with the course participants.
- Transaction SM12 no longer displays any “shared” locks for transaction SU3. This means that transaction SU3 no longer calls any displayable shared locks.
- The behavior of the foreign key check has changed from SAP R/3 4.6C to SAP R/3 Enterprise. There is an instructor note in the relevant lesson.

# Unit 1

1

## SAP Solutions

This unit gives the participants an idea of the importance of SAP NetWeaver as the technical foundation of all components of the mySAP Business Suite family of solutions. The instructor's preparation for this section should include the One Voice document and the initial page for SAP NetWeaver in SAPNet or the SAP Service Marketplace.

http://intranet.sap.com/onevoice

http://intranet.sap.com/netweaver

Be careful not to overload the participants with too much new terminology in the first lesson of this course. After you have shown the first graphic ("SAP's Product Strategy"), it is a good idea to summarize the content of the remaining graphics. The participants can easily refer back to their documents for the content later.

### Unit Overview

This unit introduces you to the structure of the **mySAP Business Suite** family of solutions.

### Unit Objectives

After completing this unit, you will be able to:

- Name some of the advantages that your company gains by using mySAP Business Suite
- State the basic concept of SAP NetWeaver
- Describe the central role played by the SAP Web Application Server

### Unit Contents

# Lesson: mySAP Business Suite

2 Lesson Duration: 30 minutes

## Lesson Overview

This unit provides an introduction to the structure of SAP's software solutions, at the center of which is the mySAP Business Suite.

## Lesson Objectives

After completing this lesson, you will be able to:

- Name some of the advantages that your company gains by using mySAP Business Suite

This introductory lesson aims to give the participants an overview of the benefits available to companies using the mySAP Business Suite. Only some of these benefits are presented here. You must make sure that you show the participants the SAP homepage as part of this lesson: http://www.sap.com.

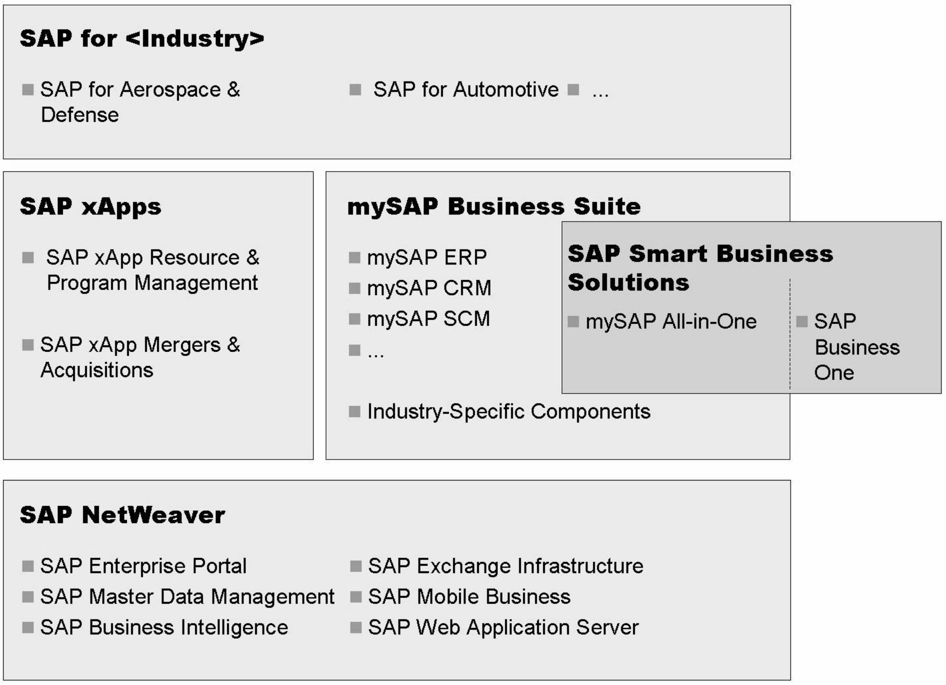
## Business Example

You want to obtain an overview of SAP solutions.

## SAP's Software Portfolio

SAP's software portfolio is constantly extended, optimized, and tailored to the needs of the market and of customers. SAP has therefore, for example, developed many industry-specific solutions over the years. The topic of integrating different business systems (even from different vendors and across company boundaries) has become increasingly important recently. SAP's software portfolio has been adapted to this trend. Products for small to midsize businesses have also been added to SAP's family of software solutions.

The graphic provides you with an overview of the structure of SAP solutions.



**SAP for <Industry>**

- SAP for Aerospace & Defense
- SAP for Automotive
- ...

**SAP xApps**

- SAP xApp Resource & Program Management
- SAP xApp Mergers & Acquisitions

**mySAP Business Suite**

- mySAP ERP
- mySAP CRM
- mySAP SCM
- ...
- Industry-Specific Components

**SAP Smart Business Solutions**

- mySAP All-in-One
- SAP Business One

**SAP NetWeaver**

- SAP Enterprise Portal
- SAP Master Data Management
- SAP Business Intelligence
- SAP Exchange Infrastructure
- SAP Mobile Business
- SAP Web Application Server

**Figure 1: SAP product strategy**

The basic elements of SAP solutions are listed below.

- SAP NetWeaver
- mySAP Business Suite
- SAP Smart Business Solutions
- SAP xApps (SAP Cross Applications)
- Industry Solutions

Each element is introduced briefly in the following.

**SAP NetWeaver**

SAP NetWeaver is the technological infrastructure for all SAP solutions.

For more information, see http://www.sap.com/netweaver.

**mySAP Business Suite**

**mySAP Business Suite** bundles all cross-industry, SAP NetWeaver-based SAP solutions.

For more information, see http://www.sap.com/solutions.

### SAP Smart Business Solutions

**SAP Smart Business Solutions** are SAP solutions for small and midsize businesses. The **mySAP All-in-One** solution is an SAP R/3-based system that has been extended with business functions from other SAP solutions. This means that each combination of functions results in a different "mySAP All-in-One package". This product is suitable for customers with a few dozen to a few hundred end users.

**SAP Packaged Solutions** offer a similar approach. These solutions are packages of functions from different solutions of the mySAP Business Suite combined individually for each customer. SAP Packaged Solutions are not shown in the above graphic.

**SAP Business One** is a completely new product that can be easily linked to existing SAP landscapes (by XML). The software is programmed in C++ and can be installed under various Microsoft Windows operating systems. SAP Business One contains functions important for running your business, such as financial accounting, customer management, purchasing, and warehouse management. Even the smallest companies can use this software profitably.

For more information, see http://www.sap.com/smb.

### SAP xApps

**SAP xApps** (SAP Cross Applications) allow you to integrate existing applications with each other by accessing existing datasets and functions using open interfaces. This means that you can implement new applications based on an existing infrastructure.

For more information, see http://www.sap.com/xapps.

### Industry Solutions

Industry Solutions provide business functions for selected industries. These functions are implemented in an SAP R/3 (Enterprise) System and are available as Add-Ons.

For more information, see http://www.sap.com/industry.

SAP solutions are shown from left to right in order of increasing functionality.

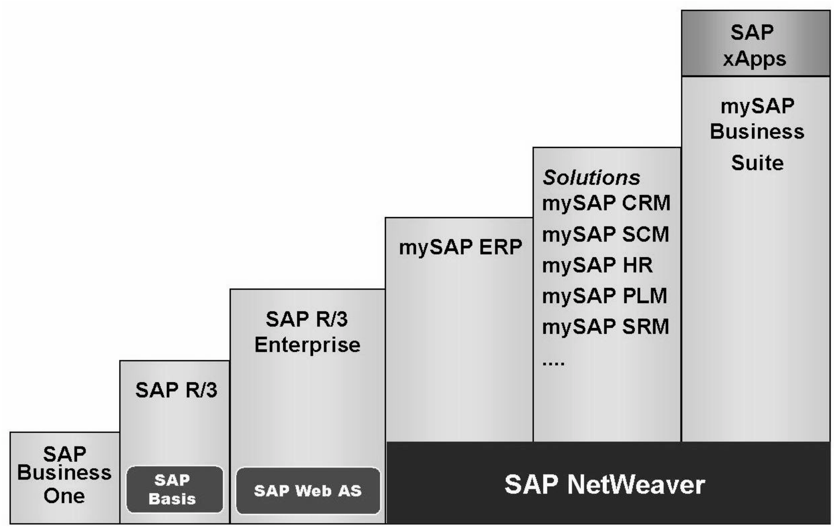


**Figure 2: SAP solution hierarchy**

**Hint:** Note that the classic SAP R/3 System up to SAP R/3 4.6C (inclusive) runs on "(SAP) Basis"; that the runtime environment of SAP R/3 is SAP Web AS, and that the mySAP ERP solution consists of an SAP R/3 Enterprise System that is running on the SAP NetWeaver platform.

## Functions of mySAP Business Suite

mySAP Business Suite, previously known as mySAP.com, is a complete package of open enterprise solutions that link all people involved, information, and processes, and therefore improve the effectiveness of your business relationships. The various SAP solutions are combined into the mySAP Business Suite and further developed individually. For more information about all SAP solutions, see http://www.sap.com/solutions. The mySAP ERP solution is based on an SAP R/3 Enterprise System with functions from mySAP HR and mySAP Financials. All SAP solutions, including mySAP ERP, are based on the complete SAP NetWeaver.

mySAP Business Suite provides:

- A complete spectrum of business solutions
- A technological infrastructure that combines openness and flexibility with maturity and stability
- Interfaces for integrating non-SAP products
- Components that can be adapted to meet multiple business requirements
- Numerous industry-specific functions

### Context of Solutions and Components

Numerous solutions for business challenges are provided in the context of the mySAP Business Suite. However, many solutions have similar or identical requirements for business functions in subareas. Different solutions therefore contain, in part, similar components. Examples of components are an SAP R/3 (Enterprise) System, an SAP APO System, or an SAP Business Information Warehouse System. The graphic for the example of the mySAP SCM solution provides an overview of this hierarchy (components as building blocks of solutions).

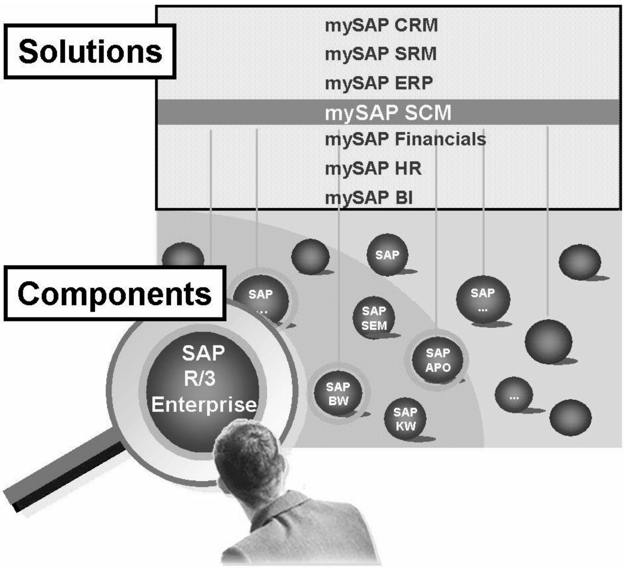


**Figure 3: Solutions and components**

### Possibilities with mySAP Business Suite

To give you a clearer impression of the possibilities of mySAP Business Suite, you can see some of the possible business scenarios that are possible using mySAP Business Suite from the following graphic.

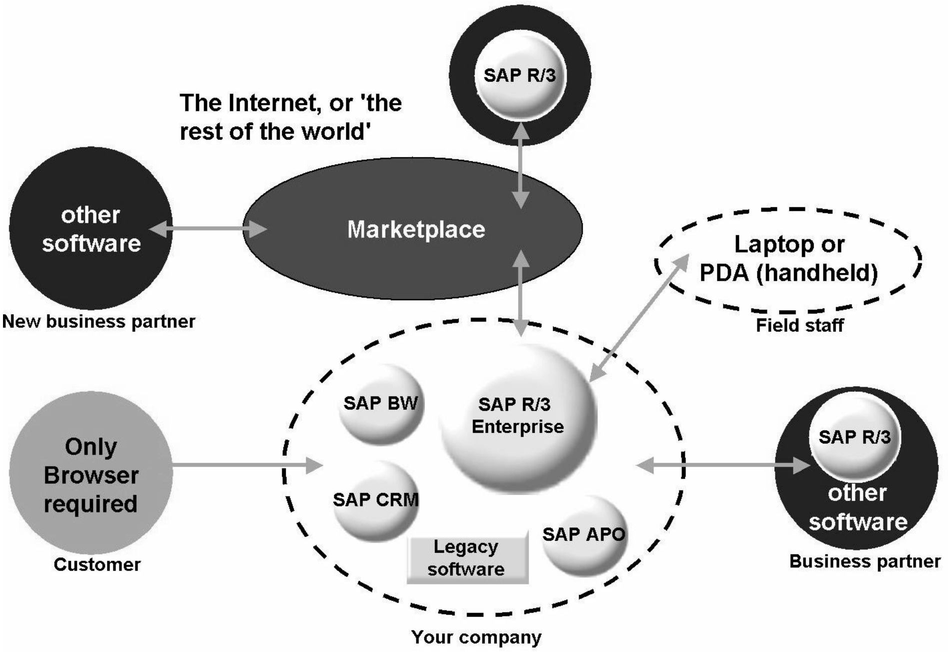


**Figure 4: Possible business scenarios with mySAP Business Suite**

mySAP Business Suite offers many options for integrating your company-internal business processes and, likewise, integrating business partners' systems. You can implement the integration of cross-enterprise processes using different technologies, such as with Application Link Enabling, using Electronic Data Interchange (EDI), through XML data exchange, or also using "Cross-Applications" (xApps) and Web Services.

## Facilitated Discussion

Highlight the possibilities offered by mySAP Business Suite.

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

Which of the opportunities offered by mySAP Business Suite will you actually use in your company?

## Lesson Summary

You should now be able to:

- Name some of the advantages that your company gains by using mySAP Business Suite

### Related Information

For more information, see the following SAP Internet sites. All of these sites are freely available, although a few require you to register free of charge.

Please draw the participants' attention to the Web site of your local SAP subsidiary, for example: http://www.sap.de.

- http://www.sap.com
- http://help.sap.com
- http://service.sap.com
- http://www.sap.info
- http://www.sapinsideronline.com

# Lesson: Definition of SAP NetWeaver

9

Lesson Duration: 30 minutes

## Lesson Overview

This lesson introduces the central role of SAP Web Application Server in the context of SAP NetWeaver.

## Lesson Objectives

After completing this lesson, you will be able to:

- State the basic concept of SAP NetWeaver
- Describe the central role played by the SAP Web Application Server

This lesson is very important, since it shows the participants how they can use the knowledge they gain from this course to work in any system based on the technology of SAP NetWeaver or SAP Web Application Server. Participants do not merely learn about the “old” SAP Basis, but find out what they need to know to work in any system based on SAP NetWeaver.

## Business Example

You want to know what SAP means by “SAP NetWeaver”, and what role SAP Web Application Server (SAP Web AS) plays in SAP NetWeaver.

## SAP NetWeaver

SAP NetWeaver is the technical foundation on which almost all SAP solutions are currently based. SAP NetWeaver provides core functions for the infrastructure of your business solutions in four subcomponents.

Not all SAP solutions are based on SAP NetWeaver, in particular SAP Business One.

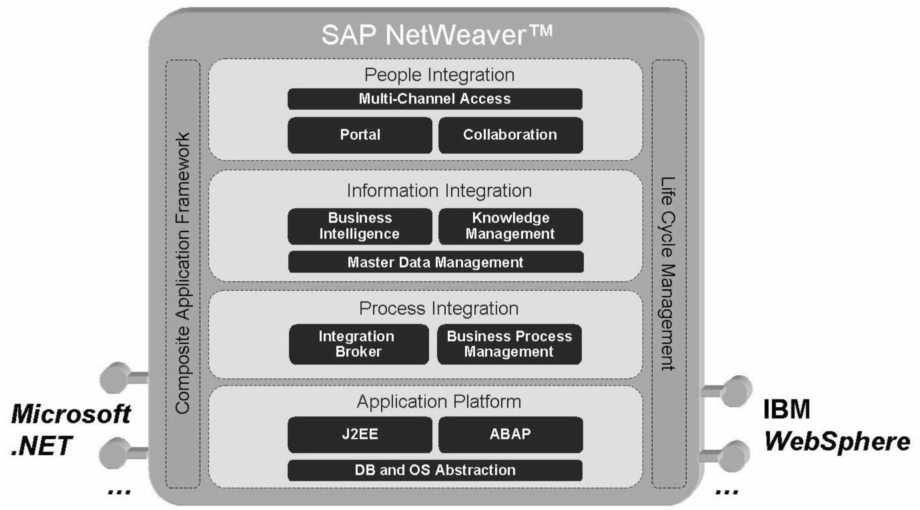


Figure 5: SAP NetWeaver

**Elements of SAP NetWeaver:**

- **People Integration**

  People Integration ensures that your employees have the information and functions that they require to perform their work as quickly and efficiently as possible. The functions of the **SAP Enterprise Portal** are of key importance here.

- **Information Integration**

  The "Information Integration" level provides access to all structured and unstructured information in your company. The core component here is the **SAP Business Information Warehouse**, which provides data from many different systems for evaluation. *Knowledge Management*, a component of the SAP Enterprise Portal, and *Master Data Management* also provide functions for central data storage.

What is meant by "structured" information? This generally means SAP data or data from SAP systems. All other information is described as unstructured. This does not mean structureless; the data could very well have its own structure, but not a SAP-specific structure.

- **Process Integration**

Process Integration ensures that business processes run across system boundaries in a heterogeneous system landscape. This is achieved by using XML data packages and workflow scenarios, among other things. The **SAP Exchange Infrastructure** (SAP XI) plays a central role here.

- **Application Platform**

  With **SAP Web Application Server**, the Application Platform has J2EE and ABAP runtime environments and therefore supports Web applications and Web services in an open development environment.

SAP NetWeaver is therefore an extensive, strategic integration and application platform that allows you to reduce the Total Cost of Ownership (TCO) of your SAP landscape. SAP NetWeaver helps you to bring together and organize people, information, and business processes across system and organization boundaries. All SAP solutions will be based on SAP NetWeaver in the future. At the same time, it supports cross-application software, known as xApps. xApps are the new class of integrated standard software. xApps extend existing (heterogeneous) IT landscapes with new business processes. SAP NetWeaver therefore provides the technical foundation for an Enterprise Services Architecture. Different software interfaces ensure full interoperability of applications that are running on Microsoft .NET and IBM WebSphere.

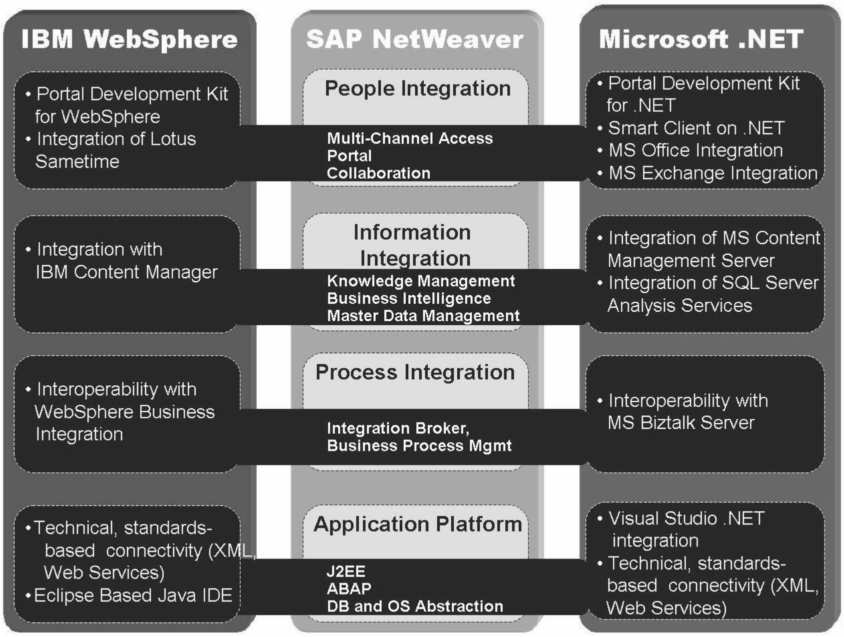

IBM WebSphere
SAP NetWeaver
Microsoft .NET
• Portal Development Kit for WebSphere
• Integration of Lotus Sametime
People Integration
Multi-Channel Access
Portal
Collaboration
• Portal Development Kit for .NET
• Smart Client on .NET
• MS Office Integration
• MS Exchange Integration
• Integration with IBM Content Manager
Information Integration
Knowledge Management
Business Intelligence
Master Data Management
• Integration of MS Content Management Server
• Integration of SQL Server Analysis Services
• Interoperability with WebSphere Business Integration
Process Integration
Integration Broker,
Business Process Mgmt
• Interoperability with MS Biztalk Server
• Technical, standards-based connectivity (XML, Web Services)
• Eclipse Based Java IDE
Application Platform
J2EE
ABAP
DB and OS Abstraction
• Visual Studio .NET integration
• Technical, standards-based connectivity (XML, Web Services)

**Figure 6: Interoperability of SAP NetWeaver**

## SAP Web Application Server (SAP Web AS)

In addition to the traditional runtime environment for ABAP programs, SAP Web Application Server also has a runtime environment for J2EE-based Java programs, the *SAP J2EE Engine*. Together with the database, SAP Web AS forms the Application Platform of SAP NetWeaver.

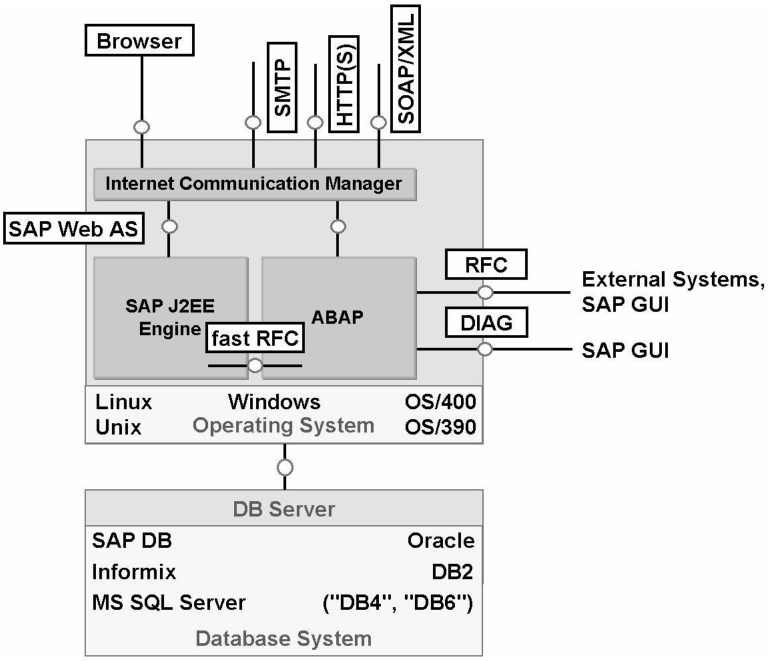


**Figure 7: SAP Web Application Server (SAP Web AS)**

**SAP Web AS** is the logical result of further development of the SAP Application Server Technology (formerly also known as “SAP Basis”), with particular attention being paid to Web-based applications.

SAP Web Application Server offers:

- A reliable and thoroughly tested runtime environment, evolved over more than 10 years
- A framework for executing complex business processes that meets the highest security standards
- A reliable and user-friendly development environment
- Support for open technical standards, such as: HTTP, HTTPS, SMTP, WebDAV, SOAP, SSL, SSO, X.509, Unicode, HTML, XML, and WML
- High scalability, inherited from SAP Basis
- Support for various operating systems and database systems

## Facilitated Discussion

The discussion should demonstrate the “power” of SAP NetWeaver and its advantages.

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

What advantages do SAP customers gain from using SAP NetWeaver?

## Lesson Summary

You should now be able to:

- State the basic concept of SAP NetWeaver
- Describe the central role played by the SAP Web Application Server

### Related Information

- For more information, see http://www.sap.com/netweaver, for example.

## Unit Summary

You should now be able to:

- Name some of the advantages that your company gains by using mySAP Business Suite
- State the basic concept of SAP NetWeaver
- Describe the central role played by the SAP Web Application Server

15

# Test Your Knowledge

1. The following solutions are provided by SAP:

   *Choose the correct answer(s).*

   □ A mySAP Business Change Enabler (mySAP BCE)
   □ B mySAP Product Lifecycle Management (mySAP PLM)
   □ C mySAP Customer Relationship Management (mySAP CRM)
   □ D mySAP Supply Chain Management (mySAP SCM)
   □ E mySAP Strategic Organization Planning (mySAP SOP)
   □ F mySAP Human Resources (mySAP HR)

2. The following statements about the SAP Web Application Server are correct.

   *Choose the correct answer(s).*

   □ A SAP Web Application Server is an evolutionary development of the previous SAP runtime environment.
   □ B SAP Web Application Server can run only applications written in ABAP.
   □ C SAP Web Application Server can run only on Microsoft Windows platforms.
   □ D SAP Web Application Server provides an SMTP interface, among others.
   □ E SAP Web AS is not scalable.

16

## Answers

1. The following solutions are provided by SAP:

   **Answer:** B, C, D, F

   mySAP BCE and mySAP SOP are imaginary products. The others are real SAP solutions.

2. The following statements about the SAP Web Application Server are correct.

   **Answer:** A, D

   SAP Web Application Server is a further development of the previous SAP Application Server. SAP Web AS can run applications written in Java using the SAP J2EE Engine and is fully scalable. It can also be run on all previously supported platforms and offers a large number of interfaces, such as an SMTP interface.

# Unit 2

17

## Navigation

Since many course participants will be seeing an SAP system for the first time, it is important that they all receive a good grounding in the basics that will enable them to access system functions efficiently. You should pay particular attention to the developers and system administrators during this unit. They are the ones who will often need to call up functions that are unfamiliar to them in the course of their work. To find these functions, administrators will need to know how to use the command field (for example, “search_sap_menu”) and table TSTCT. Table TSTCT is not mentioned explicitly in the documents, but if the instructor does refer to it, the participants normally respond with great interest.

### Unit Overview

In this unit you learn how to navigate in SAP systems using SAP GUI. You will also learn how to personalize the SAP system and get help when you are working in the SAP system.

### Unit Objectives

After completing this unit, you will be able to:

- Log on to the system successfully
- Name and use elements of a SAP GUI screen
- Use various methods to call SAP system functions
- Describe the standard menus **System** and **Help**.
- Use the basic functions of the F1 help
- Use the F4 help
- Display the information in the online documentation
- Use the various options for personalizing the SAP system

## Unit Contents

# Lesson: Logon and Screen Design

18

Lesson Duration: 30 minutes

## Lesson Overview

In this lesson, you will learn about logging on to the system and about the significance of the various SAP GUI screen elements.

## Lesson Objectives

After completing this lesson, you will be able to:

- Log on to the system successfully
- Name and use elements of a SAP GUI screen

After completing this lesson, the participants should be able to log on to the system without errors and name and use the basic elements of a SAP GUI screen. Course ADM100 covers how to configure SAP logon.

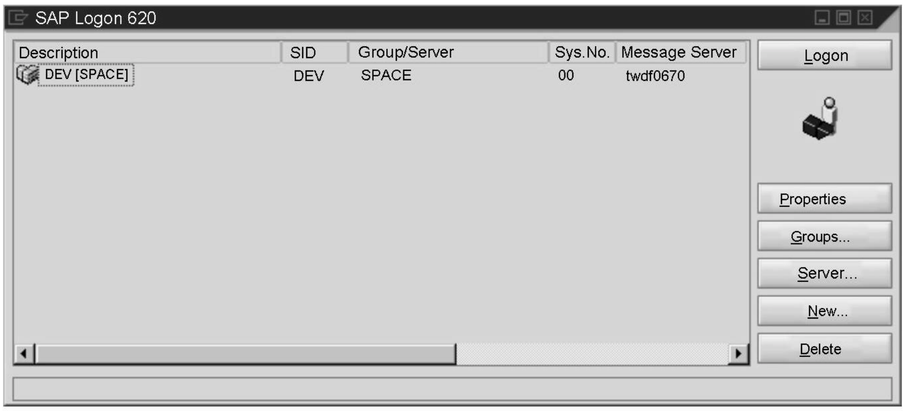
## Business Example

How do I log on to SAP systems and what are the uses of various SAP GUI screen elements?

## Logging on to the System

You can use differently structured front end programs to access SAP systems, in this way, for example,the SAP Business Information Warehouse (SAP BW) has the Business Explorer (BEx) as its front end. All SAP solutions are, however, accessible using a general front end program, the SAP GUI (SAP Graphical User Interface). The SAP GUI is the standard program for accessing almost all SAP solutions. Several variants of the SAP GUI are available; these variants are all graphically equivalent, but adapted for use in different environments. For the sake of simplicity, further descriptions refer to the SAP GUI for the Windows environment.

The SAP GUI program connects the front end computer with SAP systems. Theoretically you can specify the SAP system required at the command-line level when calling the SAP GUI program; in practice, you never need to do this. For starting SAP GUI, SAP provides another program: SAP Logon. When you call up SAP Logon, it displays a list of SAP systems for which you can start the logon process. This list is taken from a file on the front end: *saplogon.ini*. This file is usually centrally preconfigured and provided for end users. During logon, the SAP Logon program also enables“logon load distribution” using the resources available for the system selected.



| Description | SID | Group/Server | Sys.No. | Message Server |
|---|---|---|---|---|
| DEV [SPACE] | DEV | SPACE | 00 | twdf0670 |

Figure 8: The SAP Logon program

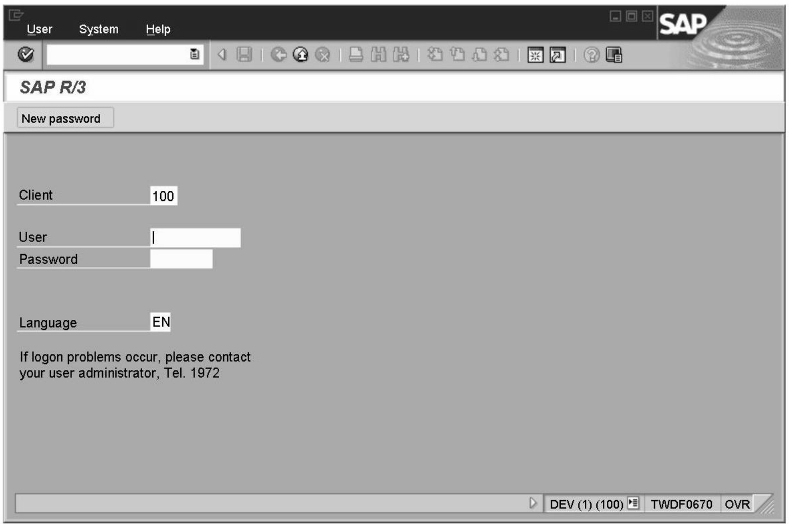
When logging on to an SAP system, you will be prompted to enter the user and password, among other things. If you have implemented a Single Sign-On (SSO) solution, you may not need to enter this information. You also have the option of specifying a client when logging on; the client field usually already contains an appropriate default value.

**Hint:** A client usually represents a company in an SAP system. This means that if an SAP system has several clients, then several companies can be represented and simultaneously active in that system. The client has a corresponding key field in the tables of the database used by that SAP system. If you are in a client, you can only access data for that specific client. Clients therefore correspond to independent business entities.

When logging on, you can select a logon language supported by that system. SAP systems can support a large number of languages, the minimum being English and one other selected language. The number of installed languages determines which languages your system supports. On the logon screen, you also have the option of using the *New password* pushbutton to change your password, at the most once per day. Your system administrator can add additional text to your logon screen. For more information about how to do this, see SAP Note 205487.

**Note:** SAP Notes contain detailed information on certain functions or corrections for known errors in SAP products. You can access SAP Notes through, for example, the SAP Service Marketplace on the Internet, using a valid user (S-user) and password: http://service.sap.com/notes

**Hint:** In the course of **one** logon to a system, you can work in several sessions (processing windows of an SAP system) simultaneously. Your system administrator can use a system parameter to define how many sessions are permissible for each logon to the SAP system. This parameter (`rdisp/max_alt_modes`) applies to all users of a system and can be set to values between 2 and 6 (SAP Web AS 6.10 and earlier) or 2 and 16 (SAP Web AS 6.20 and higher).



Client 100

User

Password

Language EN

If logon problems occur, please contact your user administrator, Tel. 1972

DEV (1) (100) TWDF0670 OVR

**Figure 9: The logon screen for an SAP system**

User data is stored in the system by client, that is, you can, for instance, log on to client 100 in your system, but you will not have a user for client 200, even if your system has a client 200. The data for a particular user within a client is called the "user master record".

Multiple logons are logged as of SAP R/3 4.6. This is for reasons of both security and licensing. If the same user logs on more than once, then for each subsequent logon, the system displays a warning message that offers the user the following three options:

- Continue with this logon and end any other logons in the system

- Continue with this logon without ending any other logons in the system (this is logged)
- Terminate this logon

Once you have logged on successfully, the initial screen of the SAP system appears; this screen is also known as the "SAP Easy Access" screen.

## Screen Structure

The SAP Easy Access screen is the default initial screen in SAP systems. The left side of the screen contains a tree hierarchy of the menus available to you in the SAP system; you can use the right side of the screen to display your company logo. This graphic is made available centrally by your system administrator and cannot be customized by individual users.

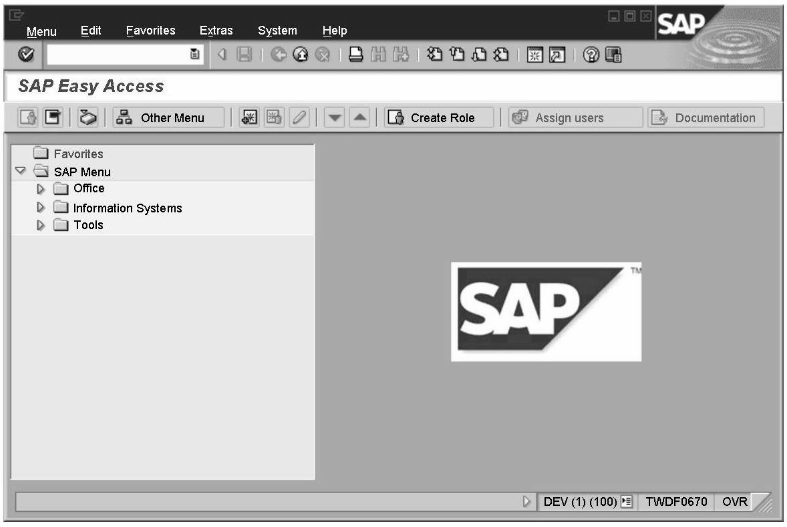
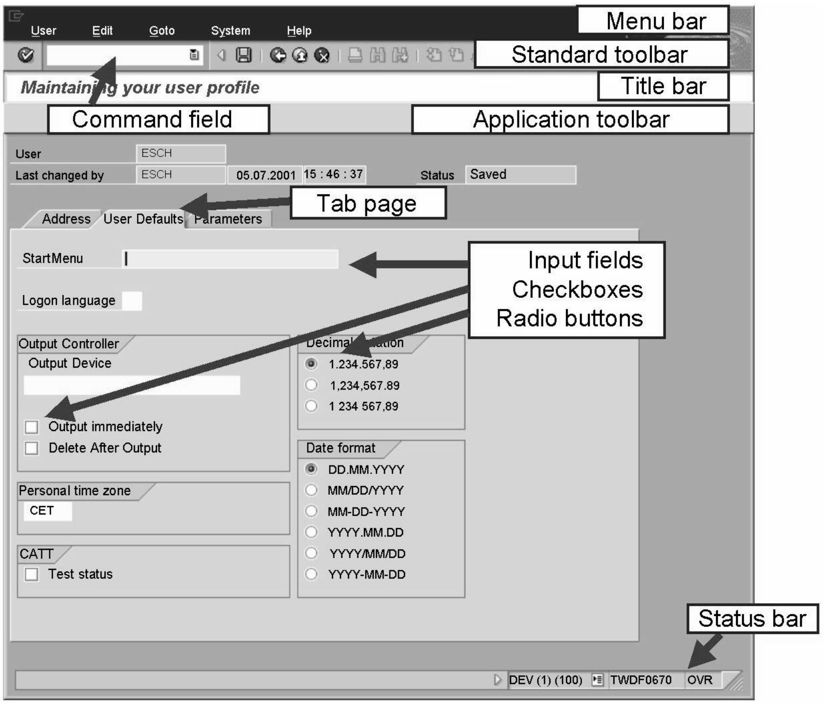


**Figure 10: SAP Easy Access**

You can find a detailed description of how to configure the graphic in SAP Easy Access under *Extras* → *Administration information* (if you have the correct authorization). Please note that the graphic is stored in the system and transported to the front end every time a user logs on. Although it is transported in compressed form, the graphic should not exceed 20 KB. You can also prevent the graphic from being displayed by either choosing the setting *Low Speed Connection* in the SAP Logon program (see SAP Note 161053), or by using *Extras* → *Settings* in the SAP Easy Access screen.

You can view a typical screen (with various elements), for example, by choosing *System* → *User Profile* → *Own Data* and then choose the *Defaults* tab page. You will then see this screen with the data for your own user:



*Maintaining your user profile*

User ESCH
Last changed by ESCH 05.07.2001 15 : 46 : 37 Status Saved

StartMenu

Logon language

Output Controller
Output Device
□ Output immediately
□ Delete After Output

Personal time zone
CET

CATT
□ Test status

Decimal notation
● 1.234.567,89
○ 1,234,567.89
○ 1 234 567,89

Date format
● DD.MM.YYYY
○ MM/DD/YYYY
○ MM-DD-YYYY
○ YYYY.MM.DD
○ YYYY/MM/DD
○ YYYY-MM-DD

DEV (1) (100) TWDF0670 OVR

**Figure 11: Maintaining your user profile**

An SAP screen can contain the following simple screen elements:

- **Command field**: you can start applications directly by entering their transaction code in the command field (this is hidden as default). You can find the transaction code for an application either in the overview menu on the SAP Easy Access screen, in the status bar (see below), or in the application itself under *System → Status*.
- **Menu bar**: the menu bar is the top line of any dialog window in the SAP system. The menus shown here depend on which application you are in. These menus may also contain cascading menus.
- **Standard toolbar**: The pushbuttons in the standard toolbar are shown on every SAP screen. If they are not available in an application, pushbuttons that you cannot use on that screen are deactivated. If you place the cursor over a pushbutton for a short time, the system displays a flag with the name or function of the pushbutton. If appropriate, the corresponding function key setting is also displayed.
- **Title bar**: the title bar names the function that you are currently in.
- **Application toolbar**: this shows the pushbuttons available in the application that you are currently in.
- **Checkboxes**: checkboxes allow you to select several options from a group of fields.
- **Radio buttons**: you can only select one option.
- **Tab**: this organizes several subscreens to improve clarity.

- **Status bar**: the status bar displays information on the current system status, such as warnings and errors. You can also change the display variant to show, for example, the transaction code of the transaction you are currently in.

Other elements include **input fields** and **pushbuttons**.

# 25 Exercise 1: Logon and Screen Design

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- To learn how to log on to an SAP system

## Business Example

You want to work in an SAP system.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:** None

## Task Logging on to the System and Initial Tasks

Log on to the training system and carry out some basic exercises in the system. The menu paths refer to the SAP standard menu.

1. Start the SAP Logon program and choose the entry for the SAP system specified by your instructor. Choose the *Logon* pushbutton.

   Enter the client, user name, initial password, and the logon language specified by the instructor. When you log on for the first time, enter a new password of your choice twice in the window that appears.

2. What is the maximum number of parallel sessions (windows of the SAP system) that you can open using *System* → *Create session*?

3. What is the name of the function you reach if you choose *Tools* → *Administration* → *Monitor* → *System Monitoring* → *User Overview*? What transaction code could you use to call this transaction instead of the menu?

# Solution 1: Logon and Screen Design

## Task Logging on to the System and Initial Tasks

Log on to the training system and carry out some basic exercises in the system. The menu paths refer to the SAP standard menu.

1. Start the SAP Logon program and choose the entry for the SAP system specified by your instructor. Choose the *Logon* pushbutton.

   Enter the client, user name, initial password, and the logon language specified by the instructor. When you log on for the first time, enter a new password of your choice twice in the window that appears.

   a) Follow the instructions in the exercise.

2. What is the maximum number of parallel sessions (windows of the SAP system) that you can open using *System → Create session*?

   a) Depending on the relevant system parameter setting, you can open between 2 and 16 sessions. This training system should allow you to open up to 6 sessions.

3. What is the name of the function you reach if you choose *Tools → Administration → Monitor → System Monitoring → User Overview*? What transaction code could you use to call this transaction instead of the menu?

   a) The function is called *User List* (see the entry in the title bar). You can find the relevant transaction code using *System → Status*; in this case, the code is SM04. You can also choose to display the transaction code for an active transaction in the status bar.

## Lesson Summary

You should now be able to:

- Log on to the system successfully
- Name and use elements of a SAP GUI screen

### Related Information

- You can find additional information in the online documentation under *Help* → *SAP Library* → *Getting Started*.

# Lesson: Calling Functions

28

Lesson Duration: 25 minutes

## Lesson Overview

During this lesson you will learn about various ways of calling functions in SAP systems.

## Lesson Objectives

After completing this lesson, you will be able to:

- Use various methods to call SAP system functions
- Describe the standard menus **System** and **Help**.

During the course of this lesson, you should explain to the participants the various options for calling functions using either the mouse or the keyboard. Pay particular attention to navigation using transaction codes. There is a simple method for administrators to create their own lists of transaction codes: select the entries required for the list from the table TSTCT and save them in Excel. The easiest way of doing this is to use the function *System* → *List* → *Save* → *Local file*. Participants frequently ask about transaction code lists: this is one of the simpler answers to that question. End users do not need to create transaction code lists in this way, since they have their role-based menus at their disposal, which contain the functions required for that role normally about 10-15.

The significance of the user menus and the option of switching to the SAP menu are also covered in this lesson.

## Business Example

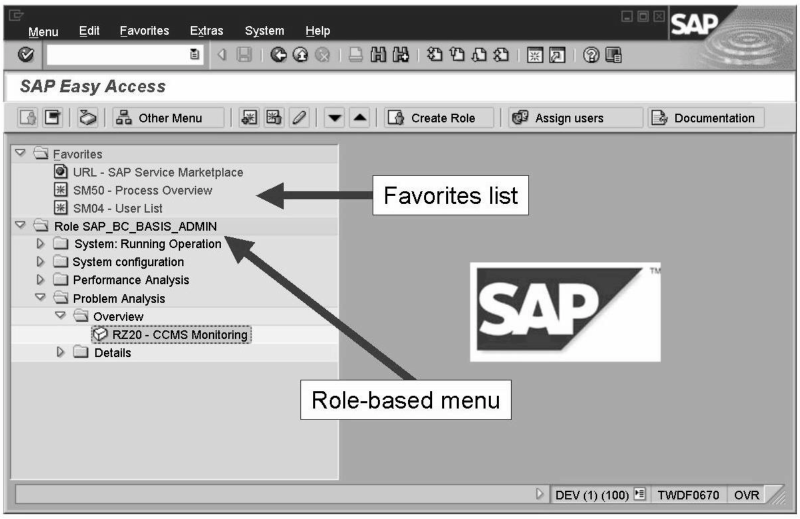
You want to know how to call functions in the SAP system.

## Favorites List and User Menu

Once you have logged on, on the left side of screen you have two overview trees at your disposal for selecting functions:

- The user-defined favorites list
- The role-based user menu or the SAP menu

The favorites list contains SAP system functions or links to Internet content or to files on the end user's front-end computer. The favorites list, which is initially empty, can be edited by each end user in accordance with his or her preferences; you can only view your own favorites list. Since data on the favorites is stored within the SAP system, each user might have different collections of favorites in different systems. Favorites can be organized into folders. You can edit your favorites in the SAP Easy Access screen using the *Favorites* menu.



Favorites list

Role-based menu

**Figure 12: Favorites list and user menu**

The *Favorites* menu gives you the option of adding a function from the user (or SAP) menu to your personal favorites list by selecting the function and choosing *Favorites* → *Add*. You also have the option of using the mouse to "Drag&Drop". You can add URLs or links to files to your list of favorites by choosing *Favorites* → *Add other objects*. Click on the triangle symbol to the left of the file symbol to expand or collapse the list of favorites.

The end user can switch from the role-based user menu to the SAP standard menu (if the system settings allow this).

**Hint:** System administrators can use table *USERS_SSM* to determine whether or not users are allowed to switch between the SAP menu and their user menus. If both menus are disallowed, then the user menu is still displayed. The **availability** of the user or SAP menu does not affect a user's authorizations. In other words, users can always use the transaction code (see below) to call transactions that they have authorization for.

The role-based user menu is created on the basis of the role(s) assigned to the user and transferred to the front end. This means that it makes sense to keep roles as small as possible because large roles could take a long time to transfer to the front end in certain circumstances. If a user has several roles assigned, then some functions can be repeated in different parts of the role-based user menu. If you want to prevent this, please see SAP Note 357693 (and the other SAP Notes to which it refers) for information on finding and deleting duplicates, and on other related matters.

**Hint:** As long as you have the appropriate authorization, you can also display user menus other than the ones assigned to you in your user master record. To do this, choose *Other menu*. Use *Create role* to start the role maintenance transaction, PFCG.

## Calling Functions

You have several options for navigating in an SAP system:

- By entering transaction codes in the command field
- By choosing items from menus in the menu bar
- By choosing items from the favorites list or from the user or SAP menus

You can use the keyboard to get to the SAP Easy Access screen or the command field and call system functions from there.

**Hint:** Use the following keys to navigate using the keyboard:

- “TAB”: always moves from one field element to the next within a field group
- “Ctrl + TAB”: goes from one field group to the first element of the next field group
- “Ctrl + /”: goes directly to the command field

You can find further shortcuts using the pushbutton *Customizing of local layout (Alt+F12)→ SAP GUI Help*.

When the command field is active, you can use the F1 key to display possible entries for this important field. The following entries are possible:

- `/n` to cancel the current transaction
- `/nXXXX` to call transaction **XXXX** directly from another transaction. Without the prefix you can only call XXXX from the SAP Easy Access screen.
- `/o` to display an overview of sessions
- `/oXXXX` to call transaction XXXX in a new session directly from another transaction
- `/nend` to end the logon session with a confirmation dialog box
- `/nex` to end the logon session without a confirmation dialog box

- `/i` to delete the session you are currently using

**Note:** The command field is an exception in that choosing the F4 key does not display the input help. The F4 key in the command field lists the last 15 different entries in the command field. This list is stored in the registry on the front end and is valid for all sessions on the front end (regardless of the system used).

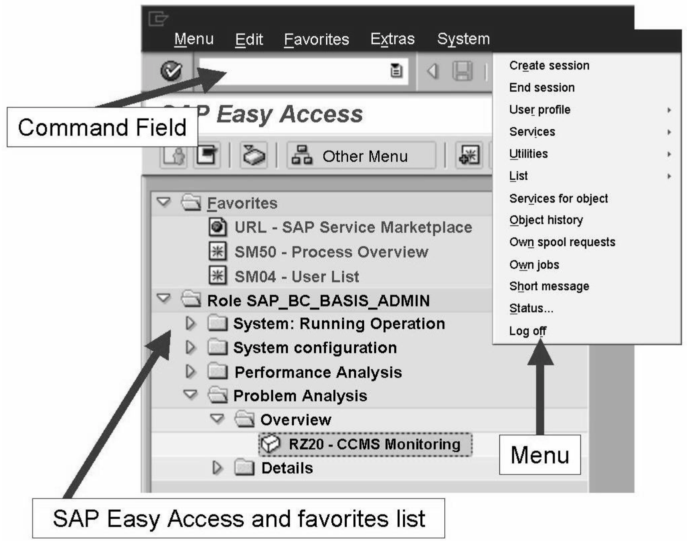


**Figure 13: Various navigation options**

**Hint:** By entering `search_sap_menu` or `search_user_menu` in the command field, you can search either the SAP standard menu or your user menu for a string or transaction code of your choice. The result of such a search is a list of hits that shows you how to navigate to the functions found using either menus or transaction codes.

You can navigate to the menus in the menu bar using the key combination `Alt + <underlined letter of the menu option required>`.

## The “System” and “Help” Menus

The *System* and *Help* menus are always available with the same options on every screen in an SAP system.

The *System* menu allows you to access various system functions. This menu also contains functions that you can only access using this menu (and not in any other way). You can log off using *System* → *Log off*, and display useful information on your system and the function you are currently using, such as the transaction code, by choosing *System* → *Status*.

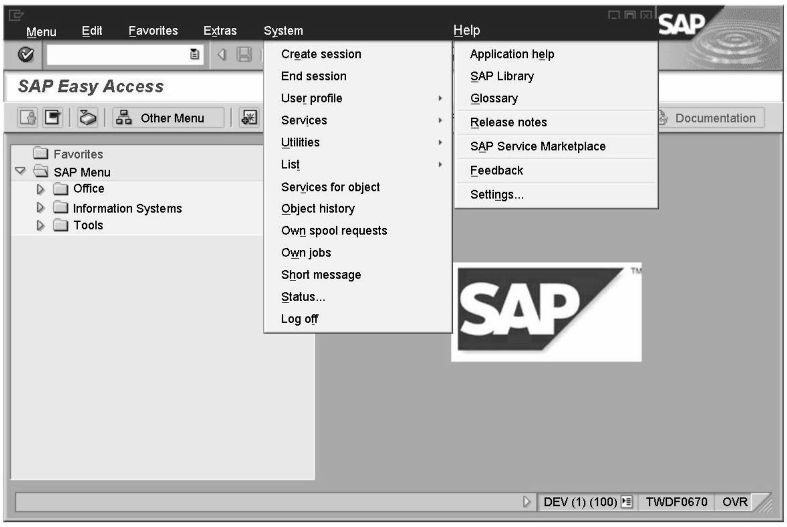


**Figure 14: The System and Help menus**

You can use the *Help* menu to access the online documentation. You can also display the "Release Notes" and configure the standard settings for your F4 Help. You will probably find that the most useful option for you here is the context-sensitive link to the section of the online documentation relevant to the function you are using. The context-sensitive help is also called "Application Help". The tool for accessing the online documentation as a whole is the "SAP Library".

# 33 Exercise 2: Calling Functions

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Familiarize yourself with various ways of calling functions and navigating in the system

## Business Example

You want to call functions.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task 1 Various Ways of Calling Up the User Overview

Who is currently logged on to the training system?

1. Call the transaction for displaying a list of users who are logged on. To do this, follow

   *Tools* → *Administration* → *Monitor* → *System Monitoring* → *User Overview* (in the SAP standard menu!).

2. Quit the user overview by using either the F3 key or the *Back* button in the standard toolbar. Now enter `sm04` in the command field on the SAP Easy Access screen. This takes you directly to the user overview.

## Task 2 Some possible entries in the command field

Which of the following are valid entries in the command field?

1. Start this exercise in the SAP Easy Access screen. Make the following entries in the command field without returning to the SAP Easy Access screen in between. Make a note of your observations (you need not make any other entries on each screen that appears other than after you have entered `/nend`: please choose *No* here):

*Continued on next page*

| Entry | Result |
|---|---|
| SU3 | |
| SM04 | |
| /nsm04 | |
| /nend | |
| /nex | |

# Solution 2: Calling Functions

## Task 1 Various Ways of Calling Up the User Overview

Who is currently logged on to the training system?

1. Call the transaction for displaying a list of users who are logged on. To do this, follow

   *Tools → Administration → Monitor → System Monitoring → User Overview* (in the SAP standard menu!).

   a) See exercise for solution

2. Quit the user overview by using either the F3 key or the *Back* button in the standard toolbar. Now enter **`sm04`** in the command field on the SAP Easy Access screen. This takes you directly to the user overview.

   a) See exercise for solution

## Task 2 Some possible entries in the command field

Which of the following are valid entries in the command field?

1. Start this exercise in the SAP Easy Access screen. Make the following entries in the command field without returning to the SAP Easy Access screen in between. Make a note of your observations (you need not make any other entries on each screen that appears other than after you have entered **`/nend`**: please choose *No* here):

| Entry | Result |
|---|---|
| SU3 | |
| SM04 | |
| /nsm04 | |

*Continued on next page*

| | |
|---|---|
| /nend | |
| /nex | |

a)

| Entry | Result |
|---|---|
| SU3 | Calls the screen for maintaining your own user settings. |
| SM04 | Fails to call the user overview. You can only call a transaction directly, that is, without a prefix from the SAP Easy Access screen. |
| /nsm04 | Calls the user overview successfully. "**/n**" first ends the active transaction and then calls up the specified transaction. |
| /nend | A logoff confirmation dialog box appears. You could now exit the system. |
| /nex | No logoff confirmation dialog box appears and your session is simply terminated. |

## Lesson Summary

You should now be able to:

- Use various methods to call SAP system functions
- Describe the standard menus **System** and **Help**.

### Related Information

- Online documentation: extended help on SAP Easy Access. To display this documentation, choose *Help* → *Application Help* on the SAP Easy Access screen

# Lesson: Getting Help

37

Lesson Duration: 20 minutes

## Lesson Overview

This lesson covers the help options provided by the F1 and F4 keys.

## Lesson Objectives

After completing this lesson, you will be able to:

- Use the basic functions of the F1 help
- Use the F4 help
- Display the information in the online documentation

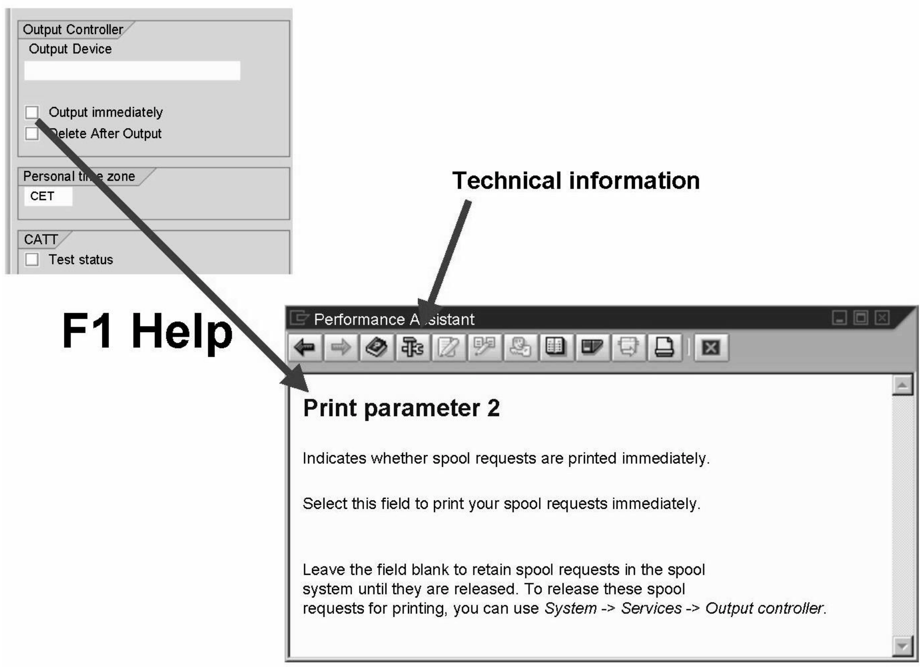
You need to have plenty of experience of using the F1 and F4 help functions. You must be able to explain how to adjust the maximum number of hits in the F4 help, and what the technical information in the F1 help means, for example.

## Business Example

You need more information on fields that are ready for input (for example, what the purpose of the field is, or what values you can enter).

## F1 Help

You can use the F1 key to display an explanation of fields, menus, functions and messages. The F1 help also displays technical information on the relevant field. For example, this information includes the parameter ID that you can use when setting user-specific default values for input fields that are also linked to the parameter ID. You can also display interesting information by, for example, calling the F1 help for the command field.



Print parameter 2

Indicates whether spool requests are printed immediately.

Select this field to print your spool requests immediately.

Leave the field blank to retain spool requests in the spool system until they are released. To release these spool requests for printing, you can use *System -> Services -> Output controller*.

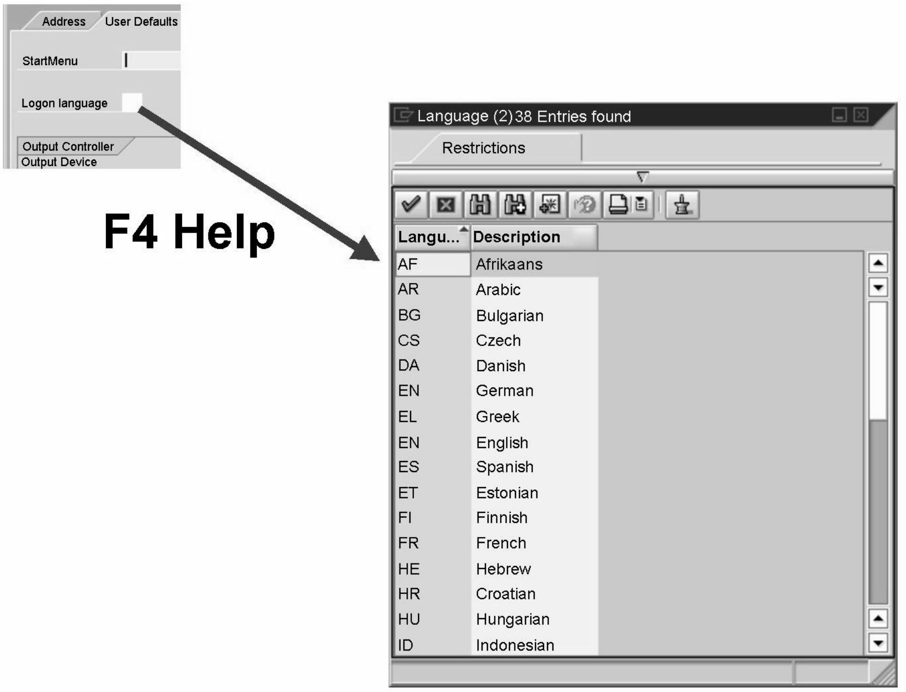
**Figure 15: The F1 help**

You can also use other buttons in the "Performance Assistant" dialog box to display information on the selected field. One of the most important pieces of information, besides the link to the context-sensitive *Application Help*, is the link to the *Technical Information*. Choose this to display detailed information, such as the parameter ID assigned to the field. You can use parameters to predefine frequently used input fields with values; to do this, the parameters are stored with the desired values in the user master record.

**Note:** You may need to activate the Performance Assistant first using the menu *Help* → *Settings...* → *F1 Help*.

## F4 Help

You can choose F4 to display possible input values. You can also call up the F4 help for a field using the button immediately to the right of the selected field. If a field contains a "checkmark" symbol, you can only proceed to the next step in that application after entering a permitted value (required entry). You can use transaction or screen variants, or Customizing to mark fields as mandatory or optional, hidden or filled with hidden standard values.



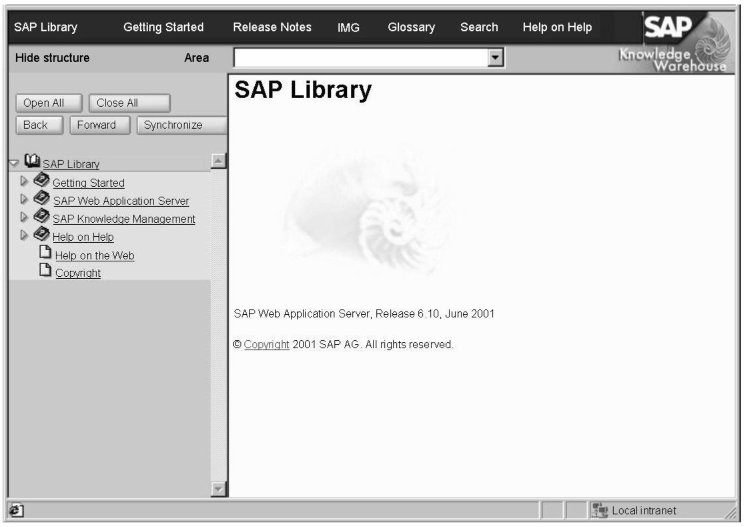
| Langu... | Description |
|---|---|
| AF | Afrikaans |
| AR | Arabic |
| BG | Bulgarian |
| CS | Czech |
| DA | Danish |
| EN | German |
| EL | Greek |
| EN | English |
| ES | Spanish |
| ET | Estonian |
| FI | Finnish |
| FR | French |
| HE | Hebrew |
| HR | Croatian |
| HU | Hungarian |
| ID | Indonesian |

**Figure 16: The F4 help**

The F4 help displays a list of possible entries for a field. If there are a large number of possible entries, a separate selection screen is displayed. If there is a long hit list, the F4 help only displays as many entries as the user has specified on the *F4 Help* tab page under *Help* → *Settings...* . The default value for the maximum number of displayed hits is 500. Take the time to familiarize yourself with the other settings you can make here.

## SAP Library (Online Documentation)

The SAP Library is a useful aid to getting to know system functions. It allows you to access the online documentation. The information stored here does not simply describe how to use system functions, it also explains system architecture concepts, gives examples of how you can configure various processes, and warns you about possible user errors and their consequences. The online documentation also contains several tips and tricks to help you carry out common tasks more easily and quickly.



SAP Library

SAP Web Application Server, Release 6.10, June 2001

© Copyright 2001 SAP AG. All rights reserved.

Figure 17: The SAP Library

**Hint:** You can also access the SAP Library on the Internet. To do this, go to http://help.sap.com , or go to http://www.sap.com and choose → *Education* → *SAP Help Portal*. You can access the full product documentation for all releases here; a user-friendly full-text search by solution is also available.

41

# Exercise 3: Getting Help

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Use the F1 help and the F4 help

## Business Example

You need more information about a field on your screen.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task Using the F1 Help and the F4 Help

In transaction SU3, call the F1 help and the F4 help.

1. Use either the system menu or transaction code SU3 to call the function for maintaining your own user data. Call the F1 help for various input fields. Find the parameter ID for the “Logon language” field using *Technical Information*. To do this you need to use the Performance Assistant display.

2. Call the F4 help for the *Logon language* field. What is the language code for Ukrainian?

# Solution 3: Getting Help

## Task Using the F1 Help and the F4 Help

In transaction SU3, call the F1 help and the F4 help.

1. Use either the system menu or transaction code SU3 to call the function for maintaining your own user data. Call the F1 help for various input fields. Find the parameter ID for the "Logon language" field using *Technical Information*. To do this you need to use the Performance Assistant display.

   a) The "Logon language" field is on the *Defaults* tab page. The *Technical Information* shows that the parameter ID for this field is `SPR`.

2. Call the F4 help for the *Logon language* field. What is the language code for Ukrainian?

   a) Carry out the exercise as described. The language code for Ukrainian is `UK`.

## Lesson Summary

You should now be able to:

- Use the basic functions of the F1 help
- Use the F4 help
- Display the information in the online documentation

### Related Information

- You can find additional information at http://www.sap.com → *Education* → *SAP Help Portal*.

# Lesson: Personalizing the User Interface

44 Lesson Duration: 20 minutes

## Lesson Overview

During this lesson, you will learn about various options for personalizing the appearance of the SAP GUI.

## Lesson Objectives

After completing this lesson, you will be able to:

- Use the various options for personalizing the SAP system

There are many options available for personalizing the system and the appearance of the front end. You need to be aware which settings depend on the front end and which depend on the user.

## Business Example

You want to personalize your access to SAP systems.

## Methods for Personalizing the SAP GUI

End users have several personalization options at their disposal. Some are described here. You can use *Extras* → *Settings* to change the appearance of the initial screen, for example, by turning off the graphics display on the right side of the screen, or by displaying technical names (transaction codes) on the SAP Easy Access screen.

Use this button in the standard toolbar for *Customizing of local layout*:

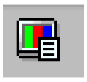


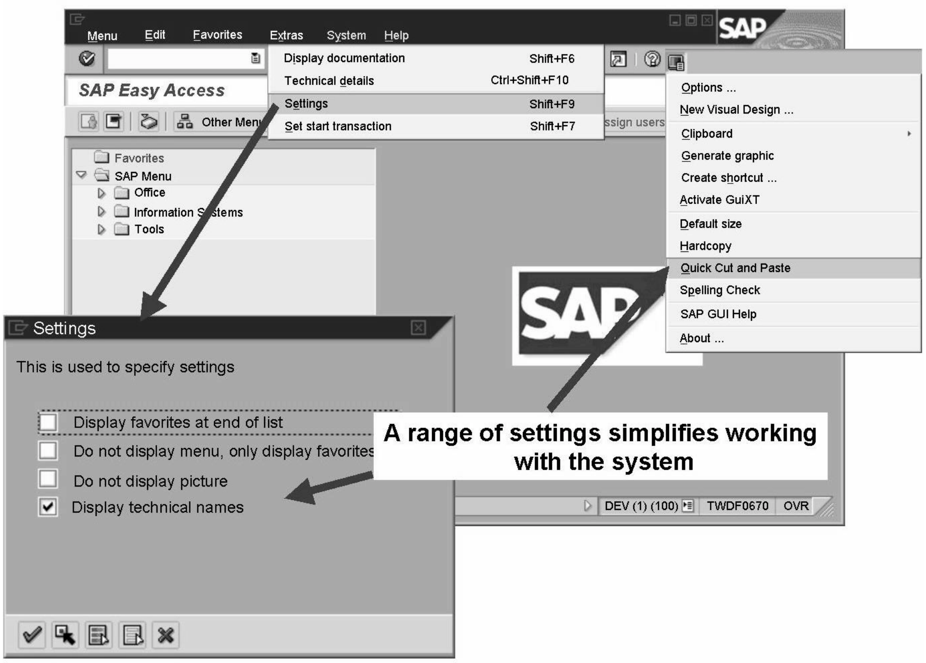
**Figure 18:**

Using the *Customizing of local layout* pushbutton, you can, for example, manage the input history by choosing *Options...* → *Local data*. When the input history is active, it builds a small front end database that contains the last n number of entries made in input fields in transactions. You can define the value of "n" yourself. These entries are then provided as input help for fields that are appropriately declared. There is a certain delay for the entries in the input history. You can minimize this delay by choosing *Options...* → *Local data* → *History* → *Immediately* .

The *Options...* also enable you to set the speed of quick info, and to display system messages in dialog boxes (*Options...* → *Messages*). Various other aids and settings are available here, for instance, you can choose a color scheme for your GUI.

Demonstrate the options mentioned. Show the class how to display system messages in dialog boxes (popups), for example. This setting comes up again in one of the exercises.

**Note:** Under *Customizing of local layout*, you can also choose *New Visual Design...* → *General* to have the option of **changing the font size** in your SAP GUI window. However, you will need to close the SAP Logon program, call it up and log on to the system again before your changes to the settings take effect.



Settings

This is used to specify settings

- □ Display favorites at end of list
- □ Do not display menu, only display favorites
- □ Do not display picture
- ☑ Display technical names

A range of settings simplifies working with the system

Figure 19: Two of the many personalization options

You can set personal standard values using *System* → *User profile* → *Own data*. You can do this on the tab pages Address, Defaults, and Parameters.

**Hint:** Use parameters to fill frequently-used input fields with default values. The prerequisite for you to be able to do this is that the input field has been assigned a parameter ID. To find the parameter ID, select the input field for which you want to define a default value and choose the F1 help, followed by the *Technical Information* pushbutton. This calls up a dialog box that displays the corresponding parameter ID under *Field data* (as long as a parameter ID is assigned to the field). Enter this parameter ID, such as, *XUS*, in the *Parameters* tab page in your user data, and assign a value to it, in this case, your user name. The result of this is that all fields with XUS as their parameter ID now have the value you entered as a default.

System table *TPARA* contains all parameter IDs.

The favorites list on the SAP Easy Access screen and the status bar display variants provide additional personalization options.

**Hint:** Use the information displayed in the status bar on the bottom right edge of the GUI window. For example, use it to permanently display the transaction code of the active transaction.

# 47 Exercise 4: Personalizing the User Interface

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Learn to use easy personalization options.

## Business Example

You would like to personalize your front end, that is, adapt it to your requirements.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task Easy Personalization Options

Carry out some simple adaptations to your front end.

1. Choose the *Customizing of local layout* pushbutton and activate *Quick Cut and Paste*. Now test this function on some fields in your *User Profile* (SU3). You can then use the same method to deactivate this function.

2. Display the transaction code of the active transaction on the status bar.

# Solution 4: Personalizing the User Interface

## Task Easy Personalization Options

Carry out some simple adaptations to your front end.

1. Choose the *Customizing of local layout* pushbutton and activate *Quick Cut and Paste*. Now test this function on some fields in your *User Profile* (SU3). You can then use the same method to deactivate this function.

   a) Choose the layout menu pushbutton and select the item *Quick Cut and Paste* from the menu. An explanation of how to use this function appears in the status bar.

2. Display the transaction code of the active transaction on the status bar.

   a) Choose the button for selecting the display variant in the status bar and select *Transaction*. The transaction code of every transaction that you now call is displayed in the status bar of this session.

## Lesson Summary

You should now be able to:

- Use the various options for personalizing the SAP system

### Related Information

- You can find additional information in the online documentation, in the "Getting Started" section.

## Unit Summary

You should now be able to:

- Log on to the system successfully
- Name and use elements of a SAP GUI screen
- Use various methods to call SAP system functions
- Describe the standard menus **System** and **Help**.
- Use the basic functions of the F1 help
- Use the F4 help
- Display the information in the online documentation
- Use the various options for personalizing the SAP system

51

# Test Your Knowledge

1. You can only work in one window (session) at a time in an SAP system.

*Determine whether this statement is true or false.*

□ True

□ False

2. Which of the following statements accurately describe a client in an SAP system?

*Choose the correct answer(s).*

□ A A client represents a completely independent business entity.

□ B A client has its own database.

□ C A client corresponds to a customer.

□ D A client may represent an entire company.

3. The following strings are valid entries in the command field:

*Choose the correct answer(s).*

□ A `/nend`

□ B `/nex`

□ C `?SM04`

□ D `/nsm04`

□ E From SAP Easy Access: `SM04`

□ F From SAP Easy Access: `search_sap_menu`

4. You can use the F1 key to:

*Choose the correct answer(s).*

□ A Display an explanation of a field's uses

□ B Display a list of possible input values

□ C Change your user data (for example, the logon language)

□ D End the session you are in

□ E Display detailed technical information on a field

5. You can use the F4 key to:

   *Choose the correct answer(s).*

   □ A Start another session
   □ B Display detailed technical information on a program
   □ C Call the SAP Library
   □ D Display possible entries for an input field
   □ E Create a print request of the screen currently displayed

6. You can also call the context-sensitive SAP Library using the F1 key.

   *Determine whether this statement is true or false.*

   □ True
   □ False

7. The following personalization options are available in the SAP GUI:

   *Choose the correct answer(s).*

   □ A You can change the font size (within limits) in the SAP GUI
   □ B You can have system messages displayed in a dialog box
   □ C You can change the size of input fields
   □ D You can suppress the display of pictures in the SAP GUI
   □ E Every user can display a picture of his or her choice on the SAP Easy Access screen
   □ F You can use a personal input history on your front end

53

# Answers

1. You can only work in one window (session) at a time in an SAP system.

   **Answer:** False

   You can work in up to 16 windows (sessions) simultaneously per logon (as of SAP Web AS 6.20).

2. Which of the following statements accurately describe a client in an SAP system?

   **Answer:** A, D

   The data for all clients in an SAP system is stored within a single common database. This data is nevertheless divided by client so that different companies can be administered and controlled in different clients. A client is "not" a customer within an SAP system.

3. The following strings are valid entries in the command field:

   **Answer:** A, B, D, E, F

   You can enter valid transaction codes directly on the SAP Easy Access screen. If you are in another function, you need to enter **`/n`** in front of the transaction code. **`/nend`** and **`/nex`** are two different logoff options. **`?SM04`** is not a valid entry. **`search_sap_menu`** is a valid entry; it creates a search screen for entries in the SAP menu.

4. You can use the F1 key to:

   **Answer:** A, E

   The F1 help provides documentation and technical information on a field.

5. You can use the F4 key to:

   **Answer:** D

   The F4 key provides a list of possible entries for a selected field.

6. You can also call the context-sensitive SAP Library using the F1 key.

   **Answer:** True

   You can call the context-sensitive help using the *Application Help* pushbutton in the F1 help window.

7. The following personalization options are available in the SAP GUI:

   **Answer:** A, B, D, F

   Numerous personalization options are available, using the *Extras* menu, the *Customizing of local layout* pushbutton, and using the status bar and *Help → Settings....* End users cannot, however, change the size of input fields. You also cannot use the SAP GUI to display a picture of your choice on the SAP Easy Access screen.

# Unit 3

55

## The System Kernel

This is a pivotal unit in this course. It describes in detail how SAP systems process user requests. It discusses SAP Web Application Server process types and explains important concepts, such as the SAP transaction, asynchronous updates and the lock mechanism. Participants need the information in this unit if they are to understand the content of more advanced courses. You should therefore make sure that you plan enough time for this unit. You should get at least as far as finishing the "Structure of an Instance" lesson on the first day of the course.

### Unit Overview

This unit gives you a detailed insight into how the SAP system processes user requests – from SAP GUI via the SAP Web Application Server to the database and back to SAP GUI. The various process types and interfaces of the SAP Web Application Server are introduced and their significance is explained.

### Unit Objectives

After completing this unit, you will be able to:

- Outline simple client/server configurations
- Describe the processing flow for user requests in SAP systems
- List various types of SAP GUI
- Outline how SAP Web Application Server and SAP GUI communicate, and how SAP Web Application Server and the database communicate
- Name the most important processes on an SAP Web Application Server
- Define the term instance and recognize the characteristics of a central instance
- Outline the processing flow for a dialog step in the SAP system
- Describe the concept of work process multiplexing

- Define the term transaction
- Outline differences between a database transaction and an SAP transaction
- Recognize the significance of a lock mechanism for the consistency of business data
- Outline the flow for a lock request
- Outline the principle of asynchronous updates
- Explain how updates are processed in the system
- Describe the print process flow
- Use various concepts fundamental to the spool environment correctly
- Explain the difference between dialog processing and background processing
- Name areas where background processing is used
- Schedule simple jobs using the Job Wizard and monitor their execution and results

## Unit Contents

# Lesson: Client/Server Architecture

57

Lesson Duration: 35 minutes

## Lesson Overview

This lesson introduces the client/server concept on which SAP systems are based. Simple client/server configurations are discussed. On the basis of this discussion, you will gain an overview of how user requests are handled in SAP systems.

## Lesson Objectives

After completing this lesson, you will be able to:

- Outline simple client/server configurations
- Describe the processing flow for user requests in SAP systems

This lesson explains the concepts of client and server in the context of SAP systems; participants should also gain an overview of the processes involved in processing a user request. You can also refer to the client and server concepts while explaining the processing of user requests (a work process is both client and server).

## Business Example

In the course of implementing an SAP system, you need to establish how you are going to distribute the required SAP system processes among the available hardware.

## Client and Server: Terminology Definition

The participants need to be able to differentiate between hardware-oriented and software-oriented views. To ensure that the participants have completely understood this section, you could, at the end, ask if the sentence “Several SAP Web Application Servers can run on one application server” makes sense, and what exactly it means.

Before we discuss various client/server configurations in the context of SAP systems, we first need to define the concepts “client” and “server”. There are basically two ways of doing this:

In the **hardware-oriented** view, the term “server” means the central server in a network that provides data, memory, and resources for the workstations (clients).

In the **software-oriented** view, client and server are both defined at the process level (service). A service in this context is a service provided by a software component. This software component can consist of a process (such as a work process) or a group of processes (such as a SAP Web Application Server) and is then called a server for that service. Software components that use this service are called clients. At the same time, clients can also be servers for other specific services.

The following graphic clarifies the two approaches to the definitions.

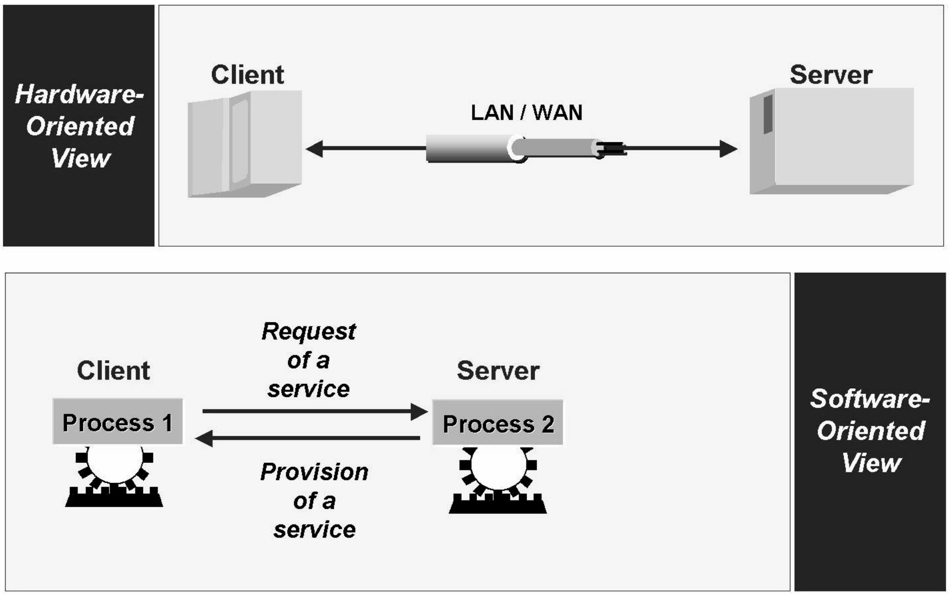


Figure 20: Hardware-oriented view – software-oriented view

In the context of SAP systems, the terms client and server are generally used as defined in the software-oriented view.

## Simple Client/Server Configuration for SAP Systems

The participants should get to know the various configuration options; specifically, they should understand the uses of a special hardware layer for application processes (scalability, logon load balancing). You may also want to include a brief summary of the concept of logon groups.

The following processes are required for operating business application software:

- Presentation processes (for example, for displaying screens)
- Application processes (for example, for executing application programs)

- Database processes (for example, for managing and organizing database data)

When you are configuring an SAP system, you need to decide how you are going to distribute the required processes among the available hardware. There are various ways of doing this, some of which are described below in more detail. Configurations are either single-level or multilevel, depending on the number of computer layers used (see the following graphic). The SAP R/3 system is an example of business application software.

- In single-tier configurations, all processing tasks (database, application and presentation processes) are performed by one computer. This is classic mainframe processing.
- Two-tier configurations are usually implemented using special presentation servers that are responsible solely for formatting the graphical interface. For example, many SAP users run SAP GUI processes on Microsoft Windows PCs.
- In a three-tier configuration, each layer runs on its own host. Several different application servers can use the data from a database server at the same time.

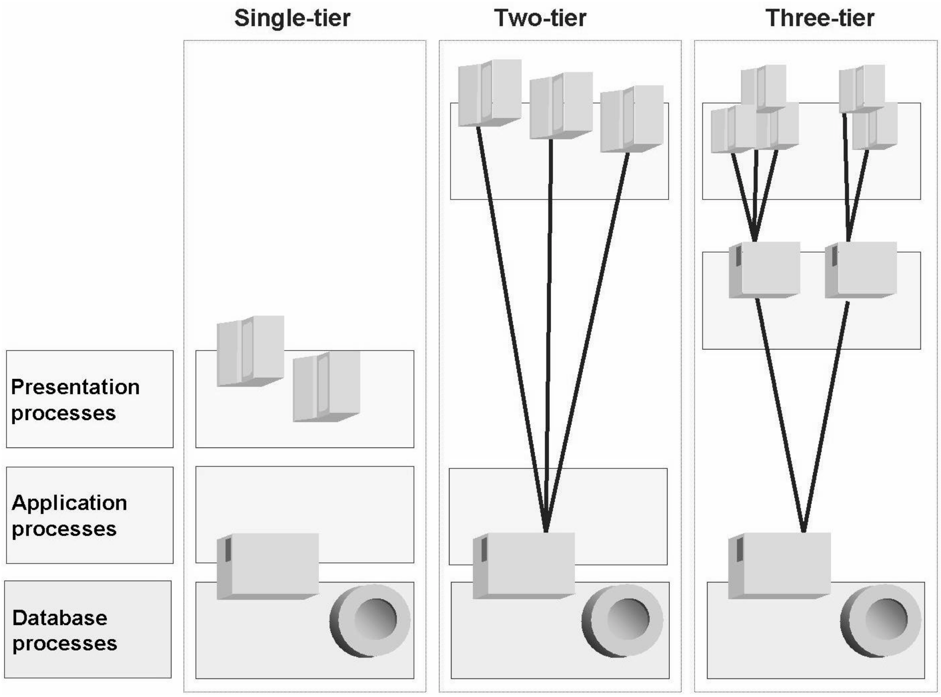


**Figure 21: Simple client/server configurations**

Single-tier configurations are generally used for tests and demonstrations (for example, an SAP system on a laptop). If several users work on a system configured in this way, then the extra hardware costs for each additional user become greater than the costs associated with implementing additional hardware layers (for example, moving the presentation processes to other hosts).

The two-tier configuration with distributed presentation processes (as shown in the previous graphic) can maintain good performance for a significantly higher number of users, without substantially increasing the hardware costs. The load resulting from the presentation processes is distributed to the various front-end computers and so does not influence the performance of the database host.

However, if the number of users exceeds a certain upper limit, the central host, on which both application and database processes run, risks becoming a bottleneck. To prevent this, you can improve the performance of the SAP system by distributing the application-layer processes to several hosts. Another advantage of adding a hardware layer specifically for application processes is that it facilitates scalability. If the number of SAP users in a system increases over time, negatively affecting system performance, then this problem can, in most cases, be solved simply by adding another host for application processes.

An alternative two-tier configuration is to install powerful desktop systems and to use these for presentation and applications (two-tier client/server). These configurations are especially suited to applications with high processor demands (for example, simulations or for software developers), but are not implemented in the SAP environment, other than for test purposes, due to the the additional administration required.

In the mySAP Business Suite environment, more complex client/server configurations consisting of more than three tiers are both theoretically possible and used in practice. Additional layers could be, for example: an Internet Transaction Server (ITS), or a Web server.

At this point you could show the participants which configuration is used for the course (normally a two-tier configuration with distributed presentation). To do this, use transaction SM51 followed by the menu path *System* → *Status*, and the message that the SAP GUI process is running locally on the front end, for example.

You could also discuss the uses of logon groups at this point. It is also helpful for participants if you draw their attention to the following fact: if there are several application servers (from the software point of view) available, then the system (or rather, the message server, which the participants do not yet know about) establishes at logon which of these servers the user is logged on to. The user remains logged on to the server selected until he/she logs off.

## Processing of User Requests in the SAP System

While explaining buffers and their usage, you could also, if necessary, talk about the need for buffer synchronization and the procedures used for this.

During this lesson, the participants will learn about the basic flow for a user request in an SAP system. Only user requests for internal users (with SAP GUI) are discussed here. User requests for Internet users connected through the ICM are not covered in this section.

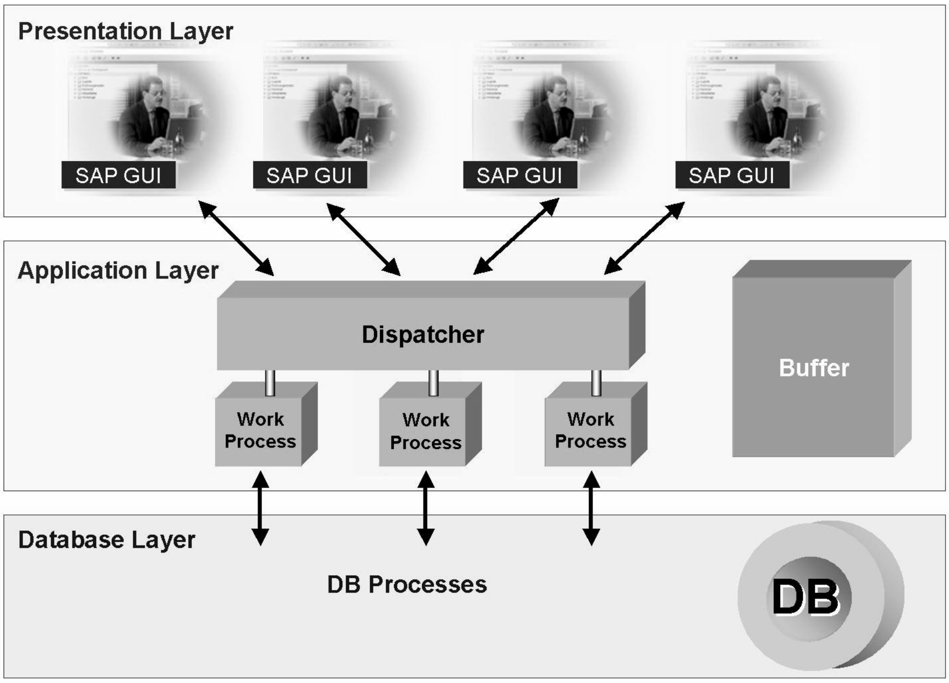


**Figure 22: Processing user requests**

A user request in an SAP system is processed, as you can see from the graphic, by various processes on all three levels (presentation, application, and database level):

The screen entries of a user are accepted by the SAP presentation program **SAP GUI** (SAP Graphical User Interface), converted to an internal format and forwarded to the SAP Web Application Server (software-oriented view).

The central process on an SAP Web Application Server is the **dispatcher**. The dispatcher, in association with the operating system, manages the resources for the applications written in ABAP. The main tasks of the dispatcher include distributing transaction load to the work processes, connecting to the presentation level and organizing communication.

The processing requests are first saved in request queues and processed according to a “first in - first out” principle.

The dispatcher distributes the requests one after the other to the available work processes. Data is actually processed in the **work process**, although the user who created the request using the SAP GUI is not always assigned the same work process. **There is no fixed assignment of work processes to users.** Processing user requests sometimes requires data to be read from the **database** or written to it. For this, every work process is connected directly to the database.

Once the process is complete, the processing result from the work process is sent via the dispatcher back to the SAP GUI. The SAP GUI interprets the received data and generates the output screen for the user with the help of the operating system on the front end computer.

The buffers help to speed up processing of user requests. Data that is often read but seldom changed (for example, programs or customizing data such as clients, currencies or company codes) can be kept as a copy of the database content in the shared memory of the application server. This means that the data does not have to be read from the database every time it is needed, but can be called very quickly from the buffer.

**Note:** During the initialization of the SAP system (during startup) the dispatcher carries out several actions, including: reading the system profile parameters, starting work processes, and logging to the message server.

## Facilitated Discussion

This discussion is intended to enhance your understanding of the way in which a user request is processed in an SAP system.

First ask the participants if they have any questions relating to this section. If they do, then you can have a group discussion of those questions. If there are hardly any or no questions, then the questions above give you a starting point for a discussion that is intended to reinforce what the participants have learned in this lesson. Feel free to substitute your own questions for the ones above and discuss these with the participants instead.

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

Which process on the application server receives the user request?

Is the assignment of users to work processes fixed?

What is a buffer used for?

## Lesson Summary

You should now be able to:

- Outline simple client/server configurations
- Describe the processing flow for user requests in SAP systems

# Lesson: Presentation and Database Interface

63 Lesson Duration: 30 minutes

## Lesson Overview

SAP systems can be used with a variety of operating systems and databases made by different manufacturers. Depending on the operating system used for the presentation layer, you will, however, need to use a special presentation program (SAP GUI). Also, depending on the database, the data from platform-independent ABAP programs needs to be translated into a database language that the database used can understand. This lesson provides an overview of the interfaces from the SAP Web Application Server to the presentation layer and to the database layer.

## Lesson Objectives

After completing this lesson, you will be able to:

- List various types of SAP GUI
- Outline how SAP Web Application Server and SAP GUI communicate, and how SAP Web Application Server and the database communicate

This lesson explains the communication processes between the various layers of an SAP system (in this case: presentation, application and database layers). You also need to cover the various types of SAP GUI (if you know how to access them, you could even demonstrate them on the system). When discussing the database interface, you should refer to the problems related to the use of native SQL (ABAP coding no longer platform-independent). You could also discuss the issue of data buffering (what kind of data should be buffered) and the need for buffer synchronization.

## Business Example

You want to learn more about the different types of SAP GUI available. You are also interested in the database interface between the application server and the database.

## The Presentation Interface

The presentation interface, SAP GUI (GUI = Graphical User Interface) enables the user to interact with the SAP system and enter or display data.

SAP GUI implements the graphical user interface using the functions provided by the relevant presentation environment. SAP GUI is based on the Windows Style Guide and is available for several platforms, providing the same functions for each. If you learned to use SAP GUI on one platform, you can use the system on another platform exactly the same as before, with the exception of a few small platform-specific GUI attributes.

The following types of SAP GUI are available:

- SAP GUI for the Windows environment
- SAP GUI for the Java environment
- SAP GUI for HTML

SAP GUI for the Windows environment (abbreviated to **SAP GUI for Windows**) is the SAP GUI implementation in a Microsoft Windows environment. SAP GUI for Windows is written in C / C++, runs on Windows-based platforms, and supports all transactions in an SAP system. The flow of data between the presentation level and the application level does not consist of prepared screens, but rather logical, compact information about control elements and user input (DIAG protocol). The volume of data transferred for each screen change is generally a few KB. You can therefore also easily connect presentation servers over WANs.

SAP GUI for the Java Environment (abbreviated to: **SAP GUI for Java**) is written in Java and is the platform-independent implementation of SAP GUI. Like SAP GUI for the Windows environment, this GUI also uses the DIAG protocol, so the volume of data transferred for each screen change is also generally only a few KB and you can easily connect presentation servers over WANs.

You can either install SAP GUI on the front end, or alternatively you can install it on a terminal server and access it using terminal clients.

**SAP GUI for HTML** consists of the Internet Transaction Server (ITS) on the server side and a Web browser on the client side. The ITS is required to convert the SAP GUI data flow from the SAP Web Application Server into HTML and back. An HTML data flow is exchanged (using the Web server) between the WGate of the ITS and the front end. The primary advantage of SAP GUI for HTML is that almost no installation is required on the front end, all you need is a browser (currently: Microsoft Internet Explorer, Version 4.0 or higher). SAP GUI for HTML does not support all the functions in an SAP system, it is, however, sufficient for the majority of users in a company.

The following graphic shows the various alternatives and the flow of communication with the SAP Web Application Server dispatcher.

If it is available and you know how to access it, you could at this point call up SAP GUI for the Java Environment (for example on a WTS) for the training system you are using and, once you have logged on, carry out a transaction in it (for example, SM50). You could do the same for the HTML GUI in the browser, so that the participants get to see the various types of GUI.

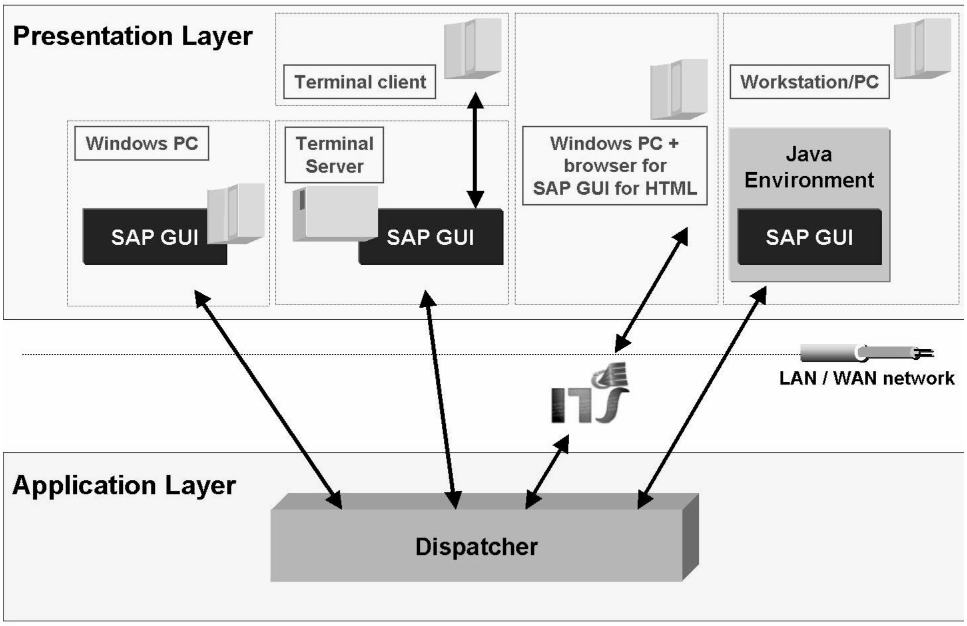


Figure 23: Alternative types of SAP GUI

**Note:** You can find further information on the various SAP GUI alternatives in the following SAP Notes:

- SAP Note 66971: front-end platforms supported
- SAP Note 314568: functions not available with SAP GUI for HTML
- SAP Note 146505: SAP GUI for the Java environment
- SAP Note 138869: SAP GUI on the Windows Terminal Server

## The SAP Web Application Server Database Interface

This section should give the participants an understanding of Open SQL as platform-independent SQL. You must draw their attention to the fact that anything they write using native SQL will be platform-dependent (see also the facilitate discussion at the end of this lesson).

Relational Database Management Systems (RDBMS) are generally used to manage large sets of data. An RDBMS saves data and relationships between data in the form of two-dimensional tables. These are known for their logical simplicity. Data, tables, and table relationships are defined at database level in the database catalog (the data dictionary) of the RDBMS.

Within the SAP programming language ABAP, you can use SAP Open SQL (SQL = Structured Query Language, database query language) to access the application data in the database, regardless of the RDBMS used. The database interface, which is part of every work process on the SAP Web Application Server, translates Open SQL statements from ABAP into the corresponding SQL statements for the specific database used ("Native SQL"). This allows ABAP programs to be database-independent.

**Note:** Open SQL is a database query language based on the (ISO) SQL standard but also containing enhancements that are not included in the standard.

When interpreting Open SQL statements, the SAP database interface checks the syntax of these statements and automatically ensures optimal utilization of the SAP buffers that are in the shared memory of every SAP Web Application Server. Data that is frequently required by the applications is stored in these buffers so that the system does not have to access the database server to read this data. In particular, all technical data, such as ABAP programs, screens, and ABAP Dictionary information, as well as a number of business administration parameters, usually remain unchanged in an operational system and are therefore ideally suited to buffering.

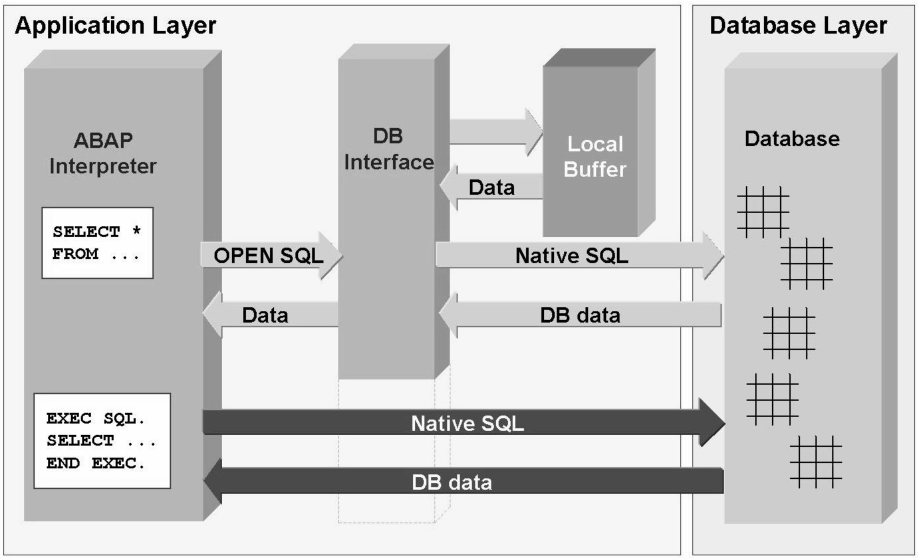


**Figure 24: Database query flow**

Furthermore, native SQL commands can be used directly in ABAP, that is, without using the local buffers and without the database interface interpreting the commands. You can do this by including the commands in a EXEC SQL. - END EXEC. bracket in the ABAP program. The ABAP Interpreter does not check the syntax of any commands within this bracket. If you use native SQL, you can no longer maintain the platform independence of the affected programs.

If you have more than one SAP Web Application Server, you will also have more than one buffer. You may therefore need synchronize buffers from time to time. You can discuss the issue of buffer synchronization with the participants at this point (optional): what causes inconsistencies between buffers? How often should you synchronize the buffers? What possible solutions are there? (Only buffer data that is changed rarely, if at all.)

## Facilitated Discussion

Participants should understand the effects of programming using native SQL commands.

You need to ensure that the participants understand that if they use native SQL commands in the programs they write, then they will lose the platform independence of the ABAP coding.

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

If you use native SQL commands in your programs, how does this affect system upgrades?

## Lesson Summary

You should now be able to:

- List various types of SAP GUI
- Outline how SAP Web Application Server and SAP GUI communicate, and how SAP Web Application Server and the database communicate

### Related Information

For more information on the various types of SAP GUI (including a collection of frequently asked questions and answers), see the SAP Service Marketplace under the quick link */sapgui*.

# Lesson: Structure of an Instance

68

Lesson Duration: 35 minutes

## Lesson Overview

Various processes are configured on an SAP Web Application Server. This lesson gives you an overview of the most important processes on an SAP Web Application Server and the tasks they fulfill. The term instance is also introduced and explained.

## Lesson Objectives

After completing this lesson, you will be able to:

- Name the most important processes on an SAP Web Application Server
- Define the term instance and recognize the characteristics of a central instance

The lesson is intended as an introduction to the processes on an SAP Web Application Server. To clarify the processes, you may find it helpful to configure an SAP system on a pin-board using prepared labels (you can find further information on this in the instructor guide area at http://intranet.sap.com/epf-adm). First you could, for example, set up a three-tier client/server architecture with several instances on different hosts. You can then assign a dispatcher to each instance. Then you can give an overview of all the work process types and the services with the tasks they perform. You could let the participants decide which processes to configure on which instance. Once you have completed the configuration on the pin-board, you can discuss the configuration that the participants have chosen (for example, taking downtime prevention into account), demonstrating the advantages of configuring a central instance (if the participants have not already configured the system in this way).

## Business Example

You want to learn about the structure of an instance of an SAP system.

## SAP Web Application Server Processes

The SAP runtime system consists of a number of parallel processes that work together.

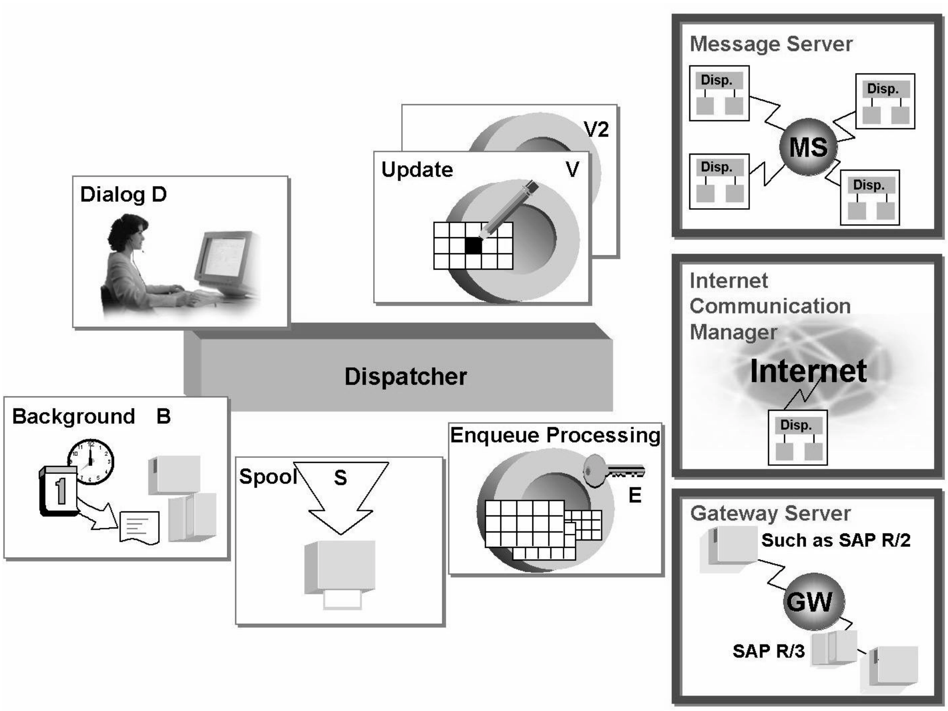


**Figure 25: SAP Web Application Server Processes**

On each application server these processes include the dispatcher as well as work processes; the number of work processes depends on the available resources:

- Dialog work processes fulfill all requests for the execution of dialog steps triggered by an active user. Every dispatcher requires at least two dialog work processes.
- Spool work processes pass sequential data flows on to printers. Every SAP system requires at least one spool work process, you can also have more than one spool work process per dispatcher.
- Update work processes execute update requests. Similarly to spool work processes, you need at least one update work process per SAP system, and you can have more than one per dispatcher.
- Background work processes execute programs that run without interacting with the user. You need at least one per SAP system (at least two for upgrades), and you can configure more than one background work process per dispatcher.
- The enqueue work process administers the lock table in the shared memory. The lock table contains the logical database locks for the SAP system. Only one enqueue work process is needed for each system.

At this point you can demonstrate the current configuration of the local instance using SM50. You could also, if appropriate, give a brief explanation of the (optional) V2 update work process.

In addition to these work processes, the runtime system provides additional systems (these are **not** work processes) for internal and external communication:

- The message server (MS) handles communication between the distributed dispatchers within an SAP system, thereby enabling scalability of several parallel application servers. The message server is configured only once per SAP system.
- The gateway server (GW) enables communication between SAP systems, or between SAP systems and external application systems. There is one per dispatcher.
- The Internet Communication Manager (ICM) is a process added with SAP Web AS 6.10. The ICM enables SAP systems to communicate directly with the Internet. The ICM receives requests from the Internet and forwards them to the SAP system for processing. It can also direct HTTP requests from an SAP system to a Web server and send the response back to the SAP system. You can configure a maximum of one ICM process per dispatcher.

The previous graphic shows an overview of SAP Web Application Server processes.

**Note:** You can use the transaction SM50 (work process overview) to display an overview of the work processes on the SAP Web Application Server that you are logged on to; you can also display this overview by choosing *Tools* → *Administration* → *Monitor* → *System Monitoring* → *Process Overview* on the SAP Easy Access Screen.

## The instance

An **instance** is an administrative unit that combines SAP system components providing one or more services. The services provided by an instance are started or stopped together. You use a common instance profile to set parameters for all components of an instance. Each instance has its own buffer areas.

**Hint:** The terms (SAP) instance and (SAP Web) Application Server are often used as synonyms.

The application layer of an SAP system (from the software point of view) generally consists of several instances; a dispatcher, work processes and the services listed above are configured on each of these instances. If, however, the application layer of an SAP system consists of a single instance, then all processes required to operate the SAP system must be configured on this instance. A system with this configuration is known as a "central system".

The graphic "The instance" shows all the processes that are required for error-free operation of an SAP system, collected on one instance. This instance is distinct from the other instances of an SAP system and is called the **central instance**. This graphic also shows other configured instances. These instances, which provide specific services, generally run on separate servers, but can also run on the same server, if required.

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

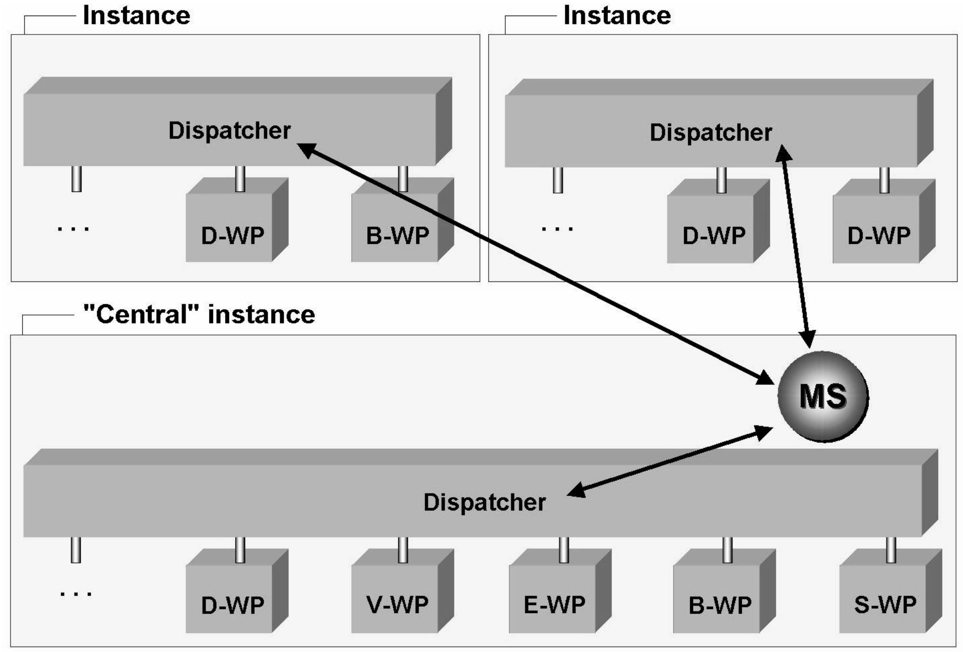


**Figure 26: The instance**

At this point you could show the participants transaction SM51 and briefly discuss the SAP Web Application Servers available, or, if appropriate, you could draw the participants' attention to the link from SM51 to SM04 and SM50.

The message server provides the SAP Web Application Servers with a central message service for internal communication (for example, for starting updates, requesting and removing locks, triggering background requests). The dispatchers for the individual SAP Web Application Servers communicate via the message server that is installed once in each SAP system (it is configured in the system profile files).

Presentation servers use the message server to log on to an SAP Web Application Server. This means that you can use the message server performance database for automatic load distribution. This load distribution, which takes place during the logon procedure, is also known as logon load balancing.

**Note:** You can display an overview of the SAP Web Application Servers configured on your SAP system using transaction SM51 (SAP Servers), or by choosing *Tools* → *Administration* → *Monitor* → *System Monitoring* → *Servers* on the SAP Easy Access screen. In this transaction, pushbuttons provide links to the user list (transaction SM04) and to the work process overview (SM50) for the selected SAP Web Application Server.

# 73 Exercise 5: Structure of an Instance

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- List important functions of transaction SM51, which displays an overview of the active instances and the work processes configured on them.

## Business Example

As the system administrator, you need to have an overview of the instances and processes of your SAP system.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:**

1. The participants must have the authorization to execute transaction SM51

## Task Overview of the instances configured on the training system

Use transaction SM51 to answer the following questions (*Tools* → *Administration* → *Monitor* → *System Monitoring* → *SAP Servers*).

1. What is the name of the instance that you are logged on to? (Hint: **instance** and [SAP Web Application] **Server** (software-oriented view) are used synonymously here.)
2. Which work process types are configured on your SAP Web Application Server?

# Solution 5: Structure of an Instance

## Task Overview of the instances configured on the training system

Use transaction SM51 to answer the following questions (*Tools* → *Administration* → *Monitor* → *System Monitoring* → *SAP Servers*).

1. What is the name of the instance that you are logged on to? (Hint: **instance** and [SAP Web Application] **Server** (software-oriented view) are used synonymously here.)

   a) The name of your instance is displayed as an entry in the *Server Names* column. The *Host name* column shows the name of the host computer on which the SAP Web Application Server is configured. If several instances are displayed in transaction SM51, then you can also find which instance you are logged on to using the menu path *System* → *Status*, followed by the *Server name* field.

2. Which work process types are configured on your SAP Web Application Server?

   a) The work process types are displayed in transaction SM51, in the *Type* column. Please note that the Internet Communication Manager (ICM) is a service, **not** a work process. Alternatively, you can answer the question using transaction SM50 (if you are in transaction SM51, you can use the *Processes* pushbutton), which lists the work processes in detail.

## Lesson Summary

You should now be able to:

- Name the most important processes on an SAP Web Application Server
- Define the term instance and recognize the characteristics of a central instance

### Related Information

SAP Note 39412 contains additional information on the topic of how many work processes you can and should configure.

# Lesson: Dialog Processing

76 Lesson Duration: 30 minutes

## Lesson Overview

In this unit, you learn how your (dialog) requests are processed by the SAP system, step-by-step. The process for a dialog transaction that consists of several screens is also outlined.

## Lesson Objectives

After completing this lesson, you will be able to:

- Outline the processing flow for a dialog step in the SAP system
- Describe the concept of work process multiplexing

You need to explain the basic process for a dialog step to the participants (what happens, in what order, in the system when the user sees the hourglass in SAP GUI). You should emphasize that during the course of one transaction that consists of several screens, various work processes can be active for a single user. If your course also covers asynchronous updates, then you need to point out here that every work process generates an implicit Commit at the end of the dialog step.

**Caution:** This lesson does not cover the connection to the Internet through the ICM.

## Business Example

You want to be able to understand how dialog processing functions in the SAP system.

## Distribution of User Requests to Dialog Work Processes

Each SAP Web Application Server has its own dispatcher. The dispatcher is the link between the work processes and the users logged on to the SAP Web Application Server (or rather, their SAP GUIs). Its tasks include distributing all the user requests it receives to the dialog work processes on the SAP Web Application Server.

Every time a user sends a (dialog) request to the SAP system (resulting in the hourglass being displayed), this request is sent to the SAP Web Application Server that the user is logged on to. On the server, the request is first placed in a request queue for dialog requests. The dispatcher distributes requests in this queue to available dialog work processes on a "first in – first out" basis.

The dialog work process selected by the dispatcher first "rolls in" the user context (that is, the dataset that contains both the current processing status of an active program and data that characterizes the user). It then processes the user request, which may involve, for example, requesting data from the database or from the buffers in the shared memory. Once the dialog work process has processed the dialog step, the work process returns the result to the dispatcher, rolls the context back out to the shared memory, and is now available again for a new user request from the request queue. Finally, the dispatcher returns the result to SAP GUI, and the new screen is displayed for the user.

**Note: Memory management and the roll-in/roll-out process:**

The memory management system differentiates between main memory areas that are available exclusively to one particular work process, and memory areas that can be used by all work processes. The memory space used exclusively by a work process stores session-specific data that must be kept for longer than the duration of a work step. This data is automatically made available to the process at the start of a dialog step (rolled in) and saved at the end of the dialog step (rolled out). This data includes data that characterizes the user (user context), such as authorizations, administrative information, and other data for the ABAP and screen processors that has been collected in previous dialog steps for the active transaction. There are also additional memory areas for all processes in the shared memory for, among other things, the factory calendar and screen, table, and program buffers.

The execution of dialog transactions is characterized by the following :

- A program dialog step is assigned to one specific dialog work process during execution.
- The individual dialog steps for a program consisting of several screens can be executed by different dialog work processes during program runtime. This is called **work process multiplexing**.
- A dialog work process sequentially processes dialog steps for various users and programs.

The following graphic illustrates this:

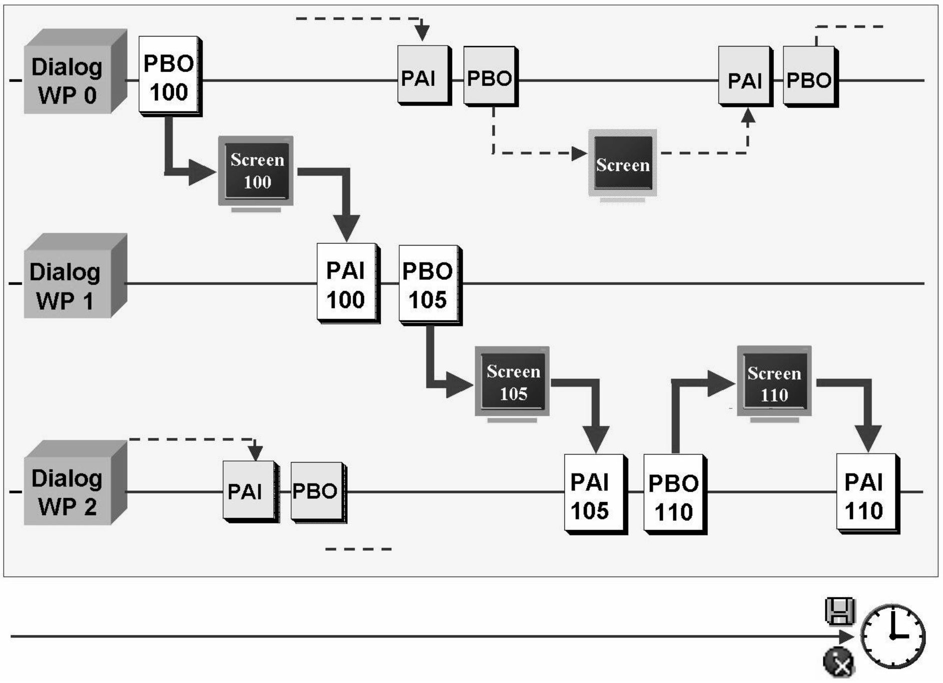


Figure 27: Work Process Multiplexing

You could also demonstrate work process multiplexing using transaction SU3 (maintaining your own user profile). You can use *System* → *Status* to check that each tab page is a separate subscreen.

## Structure of a Work Process

At this point it could be helpful for the participants if you point out that all work processes are based on the same executable, *disp+work.exe*, that is, all work processes – regardless of their type – have an identical structure.

As components of SAP Web Application Servers, work processes execute dialog steps for application programs. In addition to internal memory, a work process has a task handler that coordinates the actions within a work process, two software processors (see below), and a database interface.

SAP application programs differentiate between user interaction and processing logic.

The user actions are technically realized using screens, also called dynpros (from **dyn**amic **pro**grams), which consist of a screen image and the underlying flow logic. The screen processor executes the screen flow logic of the application program, calls processing logic modules, and transfers field content to the processing logic. The screen flow logic itself is further divided into PBO (Process Before Output), which is processed before the screen image is sent, and PAI (Process After Input), which is processed after a user interaction on the screen. The PAI part of a dialog step logically belongs to the preceding screen image, while the PBO part logically belongs to the subsequent screen image (see “Work process multiplexing” graphic).

The actual processing logic of application programs written in SAP’s programming language, ABAP, is executed by the ABAP processor. The screen processor tells the ABAP processor which subprogram needs to be executed, depending on the processing status of the screen flow logic.

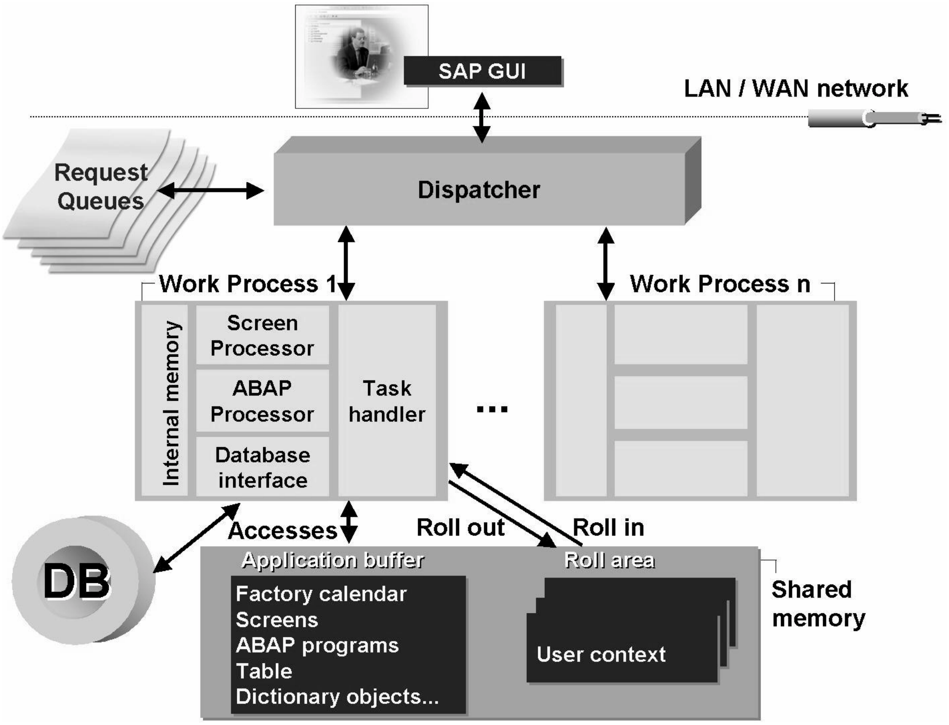


**Figure 28: Processing flow for dialog steps**

If, during a dialog step, data needs to be exchanged with the database or the buffers, then this exchange takes place through the database interface, which enables access to database tables, ABAP programs, the ABAP Dictionary, and screens, among other things.

If appropriate, you can roughly calculate, together with the participants, how many dialog processes can usefully be configured in an SAP system (taking into account the number of active users, the average processing time for a dialog step in the system and the frequency with which users send requests to the system).

81

# Exercise 6: Dialog Processing

Exercise Duration: 5 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- The participants gain an overview of the work processes configured on an instance.

## Business Example

As system administrator, you want to gain an overview of the configured work processes and their loads.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:**

1. The participants must have the authorization to execute transaction SM50

## Task Overview of the configured work processes

Use transaction SM50 to answer the following questions (*Tools* → *Administration* → *Monitor* → *System Monitoring* → *Process Overview*).

1. How many dialog work processes are there on the instance that you are logged on to?
2. Choose Refresh several times in the process overview. Are your queries always processed by the same work process?

# Solution 6: Dialog Processing

## Task Overview of the configured work processes

Use transaction SM50 to answer the following questions (*Tools* → *Administration* → *Monitor* → *System Monitoring* → *Process Overview*).

1. How many dialog work processes are there on the instance that you are logged on to?

   a) You can find the number of dialog work processes on your instance by using transaction SM50, then counting the work processes of type "DIA". Transaction SM50 displays the work processes on an instance. If you want to display the work processes for the entire SAP system, choose transaction SM66 (in this case you need to make the appropriate settings using the *Select process* and *Settings* buttons).

2. Choose Refresh several times in the process overview. Are your queries always processed by the same work process?

   a) In transaction SM50, choose the *Refresh (F8)* pushbutton. Find "your" work process by finding your user name in the *User* column. You may find – although you might not – that your queries are processed by different work processes.

      **Note:** Be aware that in this case you are executing "complete activities" and can therefore **not** see work process multiplexing in action.

## Lesson Summary

You should now be able to:

- Outline the processing flow for a dialog step in the SAP system
- Describe the concept of work process multiplexing

# Lesson: The SAP Transaction

84

Lesson Duration: 20 minutes

## Lesson Overview

Transactions are a central part of an end user's work in the SAP system. This lesson first defines the term transaction using the ACID concept, and then outlines the differences between a database transaction and an SAP transaction.

## Lesson Objectives

After completing this lesson, you will be able to:

- Define the term transaction
- Outline differences between a database transaction and an SAP transaction

During this lesson you need to distinguish clearly between database transactions and SAP transactions. Building on the participants' knowledge of work process multiplexing, you need to clarify that an SAP transaction can consist of several database transactions. You can use this lesson to prepare participants for the introduction of the SAP lock concept and asynchronous updates and explain the background to these two concepts.

## Business Example

You want to know what an SAP transaction is, and how it differs from database transactions.

## The Term Transaction

You need to explain the ACID concept. You may find it helpful to illustrate your explanation using a transaction of your choice.

Transactions are processing units grouped to provide a specific function. They have four principal characteristics. The initial letters of these characteristics together form the acronym **ACID** .

- Atomic
- Consistent
- Isolated
- Durable

This means:

- A transaction is either completely successful or remains without effect (atomic). If a transaction-oriented system goes down, you need to ensure that inconsistent, partial results are not stored.
- The system status changes from one that is accurate and consistent in business terms to another that is also accurate and consistent in business terms.
- The changes made within a transaction can only be seen by other transactions, even those that run simultaneously, after the final confirmation ("Commit").
- The results of a transaction are durable, that is, after the final confirmation they are recorded permanently in the database.

## Database Transactions and SAP Transactions

Every work process is connected to a specific communication partner at database level for the duration of an SAP instance's runtime. Work processes cannot exchange communication partners at runtime. This is why a work process can only make changes to the database within **one** database transaction.

A database transaction is, in accordance with the ACID principle, a non-divisible sequence of database operations, at the beginning and end of which the dataset on the database must be consistent. The beginning and end of a database transaction are defined by a commit command to the database system ("database commit"). During a database transaction (that is, between two commit commands), the database system itself ensures that the dataset is consistent. The database system itself takes on the task of restoring the dataset to its previous state after a transaction has terminated with an error ("rollback").

Business transactions are processing units grouped to provide a specific function; these processing units execute changes to the database that are consistent and make sense in business terms. Typical examples are credit and debit postings, which only make sense together, or creating an order and reserving the relevant materials. Correspondingly, an SAP transaction is defined as a non-divisible business process that must either be executed completely or not at all. SAP transactions are implemented as sequences of logically related dialog steps that are consistent in business terms. Every user dialog step is represented by one screen image.

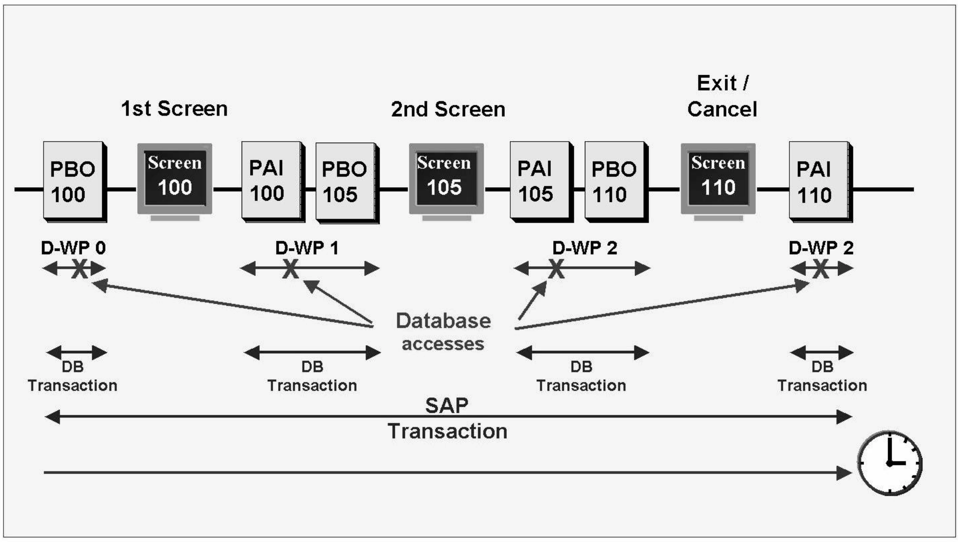


**Figure 29: Relationship between database transactions and SAP transactions**

SAP transactions are not necessarily executed within one single dialog work process. Within a transaction that changes data on the database, the user requests database changes using the displayed individual screens. Once the transaction is complete, the changes must result in a consistent database status. The individual dialog steps can be processed by different work processes (work process multiplexing), and each work process sequentially handles dialog steps for unrelated applications. Applications whose dialog steps are executed by the same work process one after the other cannot run within the same database transaction if they are not related to each other. Therefore, a work process must start a **new** database transaction for **each** dialog step.

The relationship between database transactions and SAP transactions is illustrated in the graphic “Relationship between database transactions and SAP transactions”.

## Facilitated Discussion

Discuss the meaning of the term transaction.

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

Can you undo changes to the database?

## Lesson Summary

You should now be able to:

- Define the term transaction
- Outline differences between a database transaction and an SAP transaction

# Lesson: Enqueue Processing in SAP Systems

88

Lesson Duration: 35 minutes

## Lesson Overview

To ensure data consistency within an SAP system, you must ensure that data records cannot be accessed and changed by more than one user at any one time. To do this, the SAP system has its own lock management concept. This lesson gives you an overview of how the lock mechanism works in SAP systems.

## Lesson Objectives

After completing this lesson, you will be able to:

- Recognize the significance of a lock mechanism for the consistency of business data
- Outline the flow for a lock request

You need to demonstrate to the participants the need for a lock mechanism to achieve data consistency at the database level, and also that the lock management provided by database management systems is not sufficient. You need to introduce the participants to transactions SM12 and SM04.

## Business Example

One of the system administrator's tasks is monitoring the lock entries owned by individual users in the SAP system.

## Fundamentals of Enqueue Processing in SAP Systems

Business objects must not be changed simultaneously by different users if consistency is to be maintained.

From the database point of view, every dialog step forms a physical and logical unit: the database transaction The database lock administration can only coordinate this type of database transaction. From an SAP point of view, however, this is not sufficient, because SAP transactions, which are formed from a sequence of logically related work steps that are consistent in business terms, are generally made up of several dialog steps. SAP systems need to have their own lock administration. This is implemented using the enqueue work process. This also ensures that the platform-independence of the lock management is maintained.

The SAP lock concept works on the principle that SAP programs make lock entries for data records to be processed in a lock table. Lock entries can only be made if none already exist for the table entries to be locked.

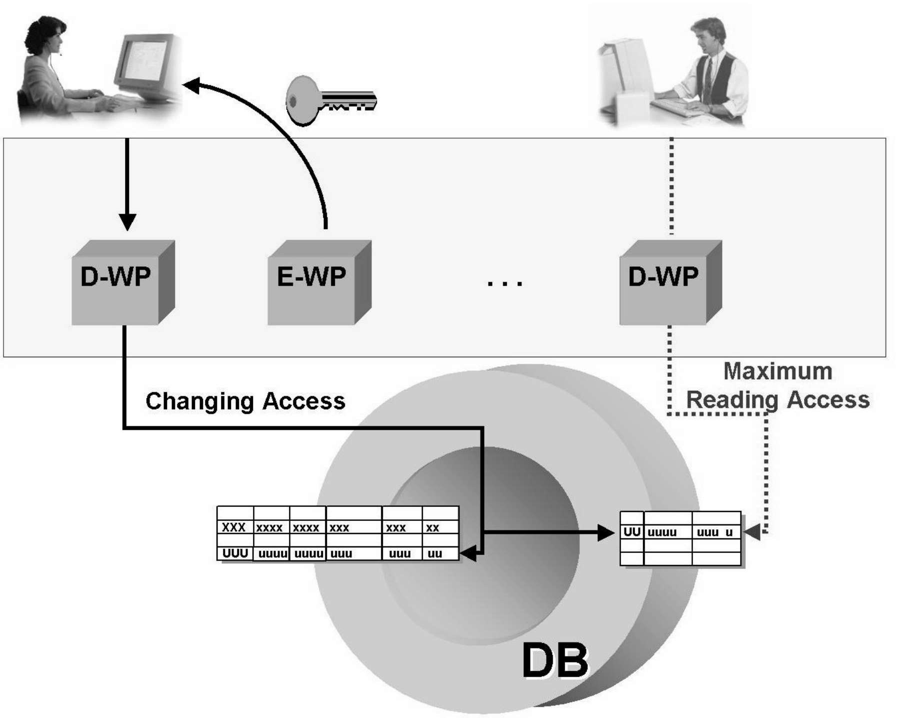


Figure 30: Fundamentals of enqueue processing in SAP systems

You could at this point demonstrate enqueue processing in the SAP system (for example, by calling transaction SU01 for an existing user in change mode and then showing that another user cannot access the same data in change mode, but only in display mode (and you cannot switch from display mode to change mode once you are in the transaction)).

## Requesting and Releasing Locks in the Enqueue Work Process

The enqueue work process administers the logical locks on SAP transactions using a lock table in the main memory of the SAP Web Application Server on which the enqueue work process is running. If the dialog work process that is handling the user request and the enqueue work process are not running on the same SAP Web Application Server, then these two work processes communicate by means of the message server (see graphic: "Requesting a lock entry").

**Note:** The SAP Web Application Server whose main memory contains the lock table is also known as the enqueue server.

In order for the system to execute lock requests, the lock object must be defined in the ABAP Dictionary. The lock object contains tables whose entries are to be locked. A lock mode can be defined for a lock object. The different types of lock mode are

- Exclusive locks (lock mode "E"; only assigned if no other locks exist for the data records required; no additional locks are then permitted for these entries)
- Shared locks (lock mode "S"; further shared locks – but no exclusive locks – can be requested for this object)

When a lock is requested, the system checks whether the requested lock conflicts with existing entries in the lock table. If the lock table already contains corresponding entries, the lock request is refused. The application program can then inform the user that the requested operation cannot currently be executed.

When a lock object is successfully activated in the ABAP Dictionary, the system generates an ENQUEUE function module and a DEQUEUE function module with the names `ENQUEUE_<lockobjectname>` and `DEQUEUE_<lockobjectname>` respectively.

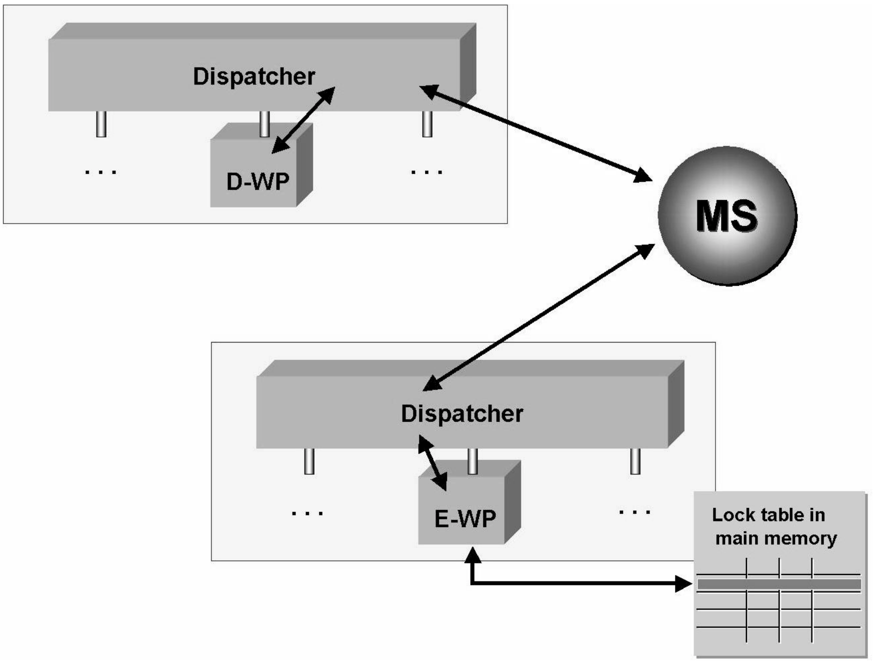


**Figure 31: Requesting a lock entry**

Locks set by an application program are either released by the application program itself or by the update program once the database has been changed. Locks that have been passed on to an update work process in this way are also written to a file at operating system level and can therefore be restored if the enqueue server goes down. Transaction SM12 (*Tools → Administration→ Monitor → Lock Entries*) displays locks held in the update process in blue and those held by the dialog work process in black.

There are basically two ways of deleting locks held by users:

- Ending the user session in the user overview (transaction SM04, *Tools → Administration → Monitor → System Monitoring → User Overview*)
- Manually deleting the lock entries in SM12

The first method (ending the user session) also results in the original lock owner leaving the transaction called and thereby releasing all locks held; the second method (manually deleting using SM12) merely deletes the lock entry from the lock table (theoretically enabling several users to change the same data records simultaneously).

**Caution:** Before deleting locks using transaction SM04 , system administrators must first check whether the user who owns the lock is still logged on to the system. You should only delete lock entries with transaction SM12 if the lock owner is no longer logged on to the system but still owns the lock (for example, if the connection between SAP GUI and the SAP system has been broken because the user has turned off his or her front-end computer without logging off from the system).

At this point you could show the participants the existing lock entries in transaction SM12 (these locks could, for example, have resulted from user data being changed in SU01). You can also use this example to illustrate the difference between shared locks and exclusive locks and show how one lock object can lock entries in several tables. It may also be helpful to draw the participants' attention to the consequences of deleting a lock entry manually. If you also demonstrate deleting a user session in SM04 , you must draw the participants' attention to the fact that – if they do this themselves – they must **never** delete the sessions of users participating in other courses who are also logged on to the same training system. You should also check that they have not done so in the system log (SM21) after the participants have completed the exercises.

# 93 Exercise 7: Enqueue Processing

Exercise Duration: 15 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Lock administration in the SAP system is demonstrated using examples.

## Business Example

You want to know how dialog processing functions in the SAP system.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:**

1. The course participants have a valid user in the SAP system that is authorized to execute transaction SU01 to make changes and to display the locks in transaction SM12 .

## Task Activities

Learn about how lock administration works in SAP systems.

1. Call transaction SU01 for user maintenance (*Tools → Administration → User Maintenance → Users*), enter your own user name, and access your user data in change mode. Open another session and attempt to access the same data in display mode and in change mode.
2. Check which tables have locks set using the list of lock entries (SM12, *Tools → Administration → Monitor → Lock Entries*).
3. Delete the session that is in change mode for your user data using the user overview SM04 (*Tools → Administration → Monitor → System Monitoring → User Overview*). Do the lock entries still exist?

# Solution 7: Enqueue Processing

## Task Activities

Learn about how lock administration works in SAP systems.

1. Call transaction SU01 for user maintenance (*Tools → Administration → User Maintenance → Users*), enter your own user name, and access your user data in change mode. Open another session and attempt to access the same data in display mode and in change mode.

   a) Call transaction SU01 for user maintenance (*Tools → Administration → User Maintenance → Users*), enter your own user name and choose *Change*. Your user data is now exclusively locked for you at SAP system level. Open another session, start transaction SU01 again, enter your user name again and choose *Change* again. The system displays the message that no other simultaneous session in the system can access your user data to change it (not even if the other session also belongs to your user). However, if you choose *Display*, then your user data is displayed.

2. Check which tables have locks set using the list of lock entries (SM12, *Tools → Administration → Monitor → Lock Entries*).

   a) You can display lock entries using transaction SM12 (see exercise for menu path). Choose *Enter*. You see a lock entry for your user for a data record of table *USR04* (user master authorizations).

3. Delete the session that is in change mode for your user data using the user overview SM04 (*Tools → Administration → Monitor → System Monitoring → User Overview*). Do the lock entries still exist?

   a) To delete your change-user session, call transaction SM04. Select your user with the mouse and choose the *Sessions* pushbutton. Select the *Maintain Users Own Data* line and choose *End session*. If you now select your current lock entries using transaction SM12 , the lock on table *USR04* has disappeared. If an action terminates in an SAP system (whether deliberately or not), the locks held by it are deleted.

## Lesson Summary

You should now be able to:

- Recognize the significance of a lock mechanism for the consistency of business data
- Outline the flow for a lock request

### Related Information

You can find questions and answers on the topic of enqueue and locks in the online documentation under *mySAP Technology Components* → *SAP Web Application Server* → *Client/Server Technology* → *The SAP Lock Concept (BC-CST-EQ)*

# Lesson: Update Processing

96

Lesson Duration: 40 minutes

## Lesson Overview

The principle of asynchronous updates is of fundamental importance to data updates within dialog transactions. This lesson explains in detail how asynchronous updates work.

## Lesson Objectives

After completing this lesson, you will be able to:

- Outline the principle of asynchronous updates
- Explain how updates are processed in the system

You need to ensure that the participants understand that asynchronous updates are a result of work process multiplexing. This lesson is of fundamental importance in understanding the technical process for dialog transactions; you should therefore spend an appropriate amount of time on it.

As preparation for the demonstration in the section "Terminations During Data Updates" and for the exercises, you could run report "VBTST300" before the course starts. Use option **I** (insert) instead of **U** (update), which may be set as default; this will ensure that there is a data record in the table that you can then overwrite, or attempt to save in the system again during the demonstration. While you are doing this, you should be aware that several courses may be using the same SAP system simultaneously, so you may at this stage already receive an express document regarding a termination during an update; this will occur if the insert has already been carried out by another instructor with the same user name.

## Business Example

You want to learn about the principle of asynchronous updates, as most SAP and application programs for update data in this way.

## The Principle of Asynchronous Updates

Alongside dialog work processes, at least one update work process is configured on every SAP system. Update work processes carry out updates, that is, they change the entries in database tables.

To ensure data consistency, the data in an SAP transaction must be updated either completely or not at all. If a runtime error occurs during part of the update, all critical database changes made by the update need to be recalled (“rollback”).

**Caution:** Since, to ensure data consistency, all changes made by an SAP transaction need to recallable until the final confirmation, all these changes must be bundled into a single database transaction. This ensures that rollback requirements are met.

If, during a dialog work process, data temporarily stored by the ABAP command `CALL FUNCTION .... IN UPDATE TASK` is passed to an update work process by the ABAP statement `COMMIT WORK` for further processing, then the dialog work process does not wait for the update request to be completed: the update is an asynchronous (not simultaneous). The asynchronous update process is illustrated in the graphic “The principle of asynchronous updates”.

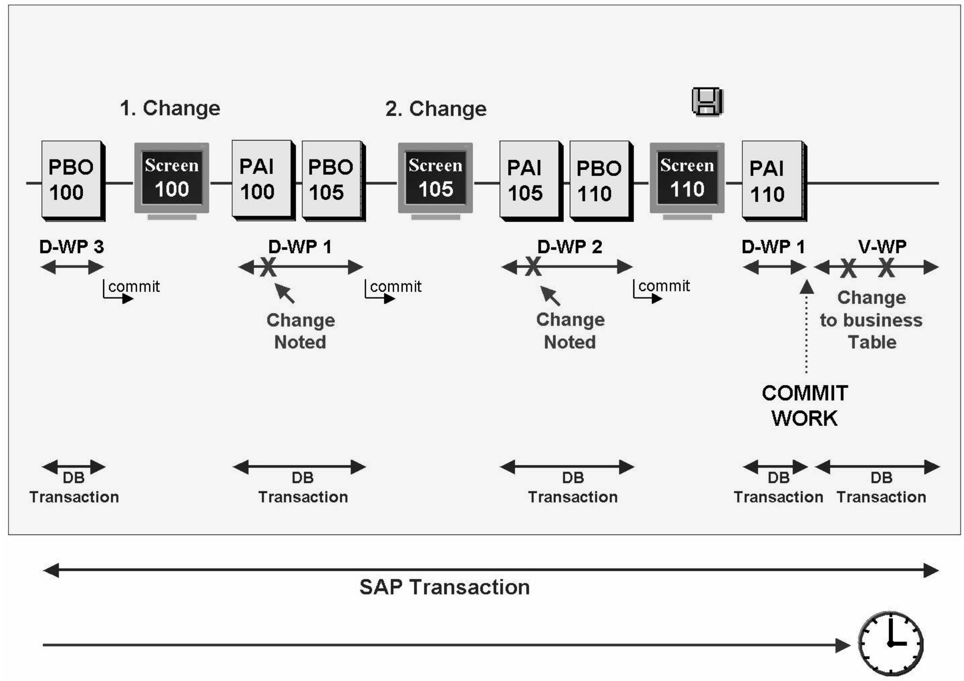


Figure 32: The principle of asynchronous updates

Asynchronous updates solve the problems caused by the different interpretations of transaction at database level and at SAP level. Bundling all updates for one SAP transaction into a single database transaction ensures that the data that belongs to this SAP transaction can be rolled back completely.

**Note:** The application developer decides whether and how to use asynchronous updates while programming the transaction.

## The Update Process

This section describes the process and the interaction of the various work process (dialog, enqueue, update) in detail; you could also involve the participants in illustrating the process on an overhead projector.

If a user wants to change a data record in an SAP transaction, he/she first calls the corresponding transaction (dialog), makes the appropriate entries on the screens, then finally initiates the update process by saving the data. This process triggers the following steps:

1. The program locks the data record for other users. The program does this by addressing the enqueue work process (via the message server if appropriate). The enqueue work process makes the relevant entry in the lock table or (if another user has already locked the data) informs the user that the data record cannot currently be changed.

2. If the enqueue work process succeeded in writing the lock entry to the lock table, then it passes the lock key it created to the user, the program reads the record to be changed from the database and the user can change the record on the screen image of the SAP transaction.

3. In the active dialog work process, the program calls a function module using `CALL FUNCTION ... IN UPDATE TASK` and writes the change request to database update tables. These are also called VB* tables, because their names begin with “VB”. They act as temporary memory and store the data to be changed until it can be collected and written to the target tables in the database (in a single database transaction).

4. At the end of the transaction (for example, when the user saves the data – possibly after completing other dialog steps), the program initiates the close of the transaction with the statement `COMMIT WORK`. The work process that is handling the active dialog step triggers an update work process.

5. On the basis on the information transferred from the dialog work process, the update work process reads the log records that belong to this SAP transaction from the VB* tables.

6. The update work process passes the changes marked and collected in the VB* tables to the database as a change request and evaluates the database response. If the changes were successfully written to the target tables, the update work process triggers a database commit after the last change to the database and deletes the entries from the VB* tables. If an error occurs, the update work process triggers a database rollback, leaves the log records in the VB* tables and marks them as defective.

7. The lock entries in the lock table are reset.

The VB* tables are VBLOG (temporary memory for data records, filled by the dialog work process), VBHDR (update headers), VBMOD (update modules), VBDATA (update data), and VBERROR (any error information); you can display the table structure using the ABAP Dictionary (transaction SE11).

The following graphic illustrates the asynchronous update process (without the processes for lock requests and releases).

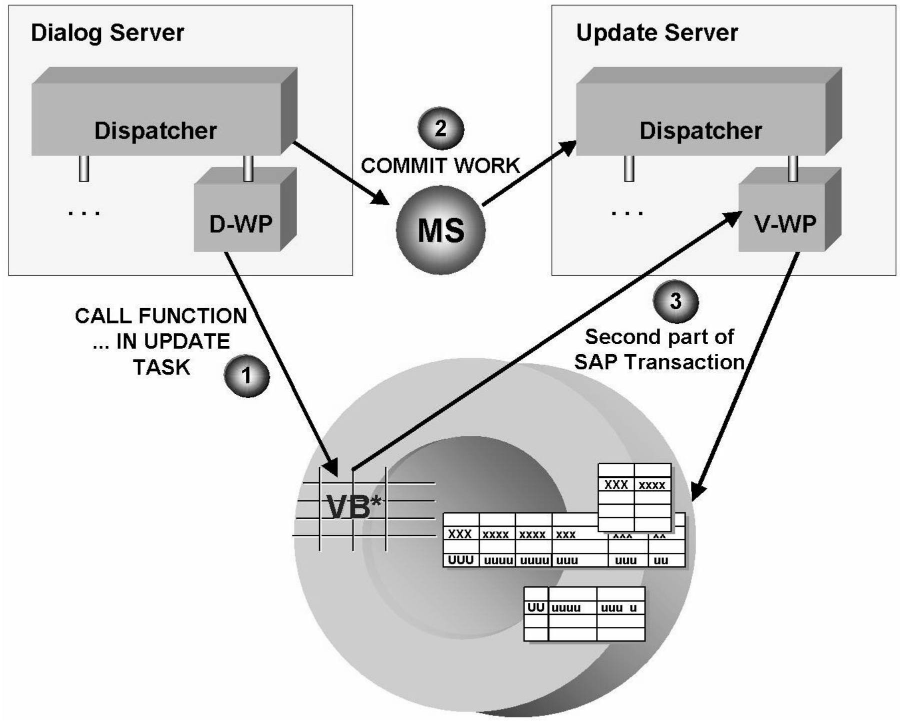


Figure 33: The asynchronous update process

You could at this point demonstrate asynchronous updates on the system. To do this, open two sessions and arrange them next to each other on the screen: in one session, call the process overview, transaction SM50 (if your system has several instances, make sure that you are logged on to the instance on which the update work process is configured). In the other session, in transaction SA38, call report VBTST300 (update test program) with parameters **U** (update), No. of COMMIT WORKS = **1**, No. of func. per update = **5000** (leave the other parameters unchanged). Once you have pressed the *Execute* pushbutton, go to the session with SM50 and keep refreshing it to show that a dialog work process is active to start with. Once the dialog part is finished, the message "update test finished" is displayed in the SA38 session, and the update part of the transaction starts in SM50.

**Hint:** Developers can differentiate between primary, time-critical (V1) update modules and secondary, non-critical (V2) update modules during programming. This differentiation means that critical database changes can be processed before less critical changes:

- V1 modules refer to time-critical changes. They are relevant to objects that have a controlling function in the SAP system, such as a change to the material stock or an order creation.
- V2 modules refer to less time-critical changes, for example, purely statistical updates such as creating a change document.

The V1 modules for an SAP transaction are processed sequentially in a single update work process. If your SAP system has a work process for V2 updates, then V2 modules will only be updated there. Once it has successfully completed processing, the V1 update work process releases the relevant locks again. This means that the "normal" update work processes are available again more quickly for time-critical (V1) updates, and that the relevant lock entries are deleted sooner. If you have not configured any V2 update work processes, then the V1 work process handles all updates.

## Terminations During Data Updates

If an error occurs during an update, then processing of the active update component terminates. Users can be automatically notified by express document when an update terminates.

If a dialog work process terminates when writing data to the VB* tables, the tables will contain data that will not be updated. The system can delete these entries automatically the next time you start the system. The application tables remain unchanged.

An asynchronous update may terminate for a variety of reasons. If, for example, several attempts are made to enter the same data record (using insert) in a table, this triggers the exception condition “Duplicate Key” in the coding because an entry already exists in the table under this key. Therefore, the corresponding data record cannot be written to the database table more than once.

At this point you could show the participants how the SAP system reacts when an update terminates (if you have made the preparations described at the beginning of this lesson). To do this, call report VBTST300 in transaction SA38 with parameters **U** (update), No. of COMMIT WORKS = **1**, No. of func. per update = **1** (leave the other parameters unchanged). This updates the database and should run without any problems. Then you can call the same program with the same parameters (except with **I** instead of **U**). The program will now try to insert an entry in the database that already exists there. As a result, at the next user interaction after you run the program (you may need to wait a second!), the system sends an express document with the title “Update was terminated”. This is a message from the update work process. Once you have received the message, you can display the update record that caused the error in SM13; here you also have the option of going to the short dump. You may find that several defective update records are displayed in SM13, since several courses may be using the same system.

You can use this demonstration to create defective update records that the participants can then display in the exercises.

When an update terminates, the system sends an “express document” to the user who triggered the update. Any additional steps must be carried out by the system administrator. Transaction SM13 (update requests) provides system administrators with analysis tools to handle terminated updates. Once the error that caused the termination has been corrected (for example, hardware damage repaired), the end user should restart the update.

# 103 Exercise 8: Update Processing

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Call important functions of transaction SM13 (Update Requests)

## Business Example

Using transaction SM13 to evaluate the update terminations that have occurred in the system is a routine task for system administration.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:**

1. The participants must have authorization to carry out transaction SM13 (update terminations) and ST22 (ABAP short dumps).

## Task Update records to Be Processed

Display the update records to be processed in your SAP system client.

1. Call transaction SM13 (update requests) and select all update records for your client. If you come across a defective update record, display the short dump for it.

# Solution 8: Update Processing

## Task Update records to Be Processed

Display the update records to be processed in your SAP system client.

1. Call transaction SM13 (update requests) and select all update records for your client. If you come across a defective update record, display the short dump for it.

    a) Call transaction SM13 (*Tools → Administration → Monitor → Update*) and, on the initial screen, select all the updates to be executed today in your client. Choose *Execute (F8)*. A list of all records still to be updated for the time period selected is displayed. If one record (or more) has the status "error", then select it. On the next screen, select the corresponding update module. This displays a new screen, on which you select *ABAP short dump* (button with list icon).

## Lesson Summary

You should now be able to:

- Outline the principle of asynchronous updates
- Explain how updates are processed in the system

# Lesson: Spool Processing

106

Lesson Duration: 25 minutes

## Lesson Overview

Document output to paper or in electronic form is one of the most frequently used functions in any data processing system. This lesson describes the basic flow for creating output and is illustrated by a simple example.

## Lesson Objectives

After completing this lesson, you will be able to:

- Describe the print process flow
- Use various concepts fundamental to the spool environment correctly

This lesson introduces the basic process for printing in an SAP system. Participants learn to differentiate between **remote** and **local** access methods. You also need to know about the popular access method **F** (front-end printing).

## Business Example

System administration enables output to paper and other media.

## Fundamentals of Creating Print Data in SAP Systems

SAP systems provide a wide variety of options for representing business and other data. This data, created and formatted in a system dialog step, can then be sent to printers and other output interfaces (fax, e-mail, and so on). A printer must first be set up in the system before it can be addressed.

You can select a printer that has already been set up by choosing the print icon (Ctrl + P), then using the F4 help. A standard printer is usually set as default in your user profile.

Once a printer has been set up, the SAP system has all the information it needs to be able to create a spool request.

A **spool request** contains information on the data to be output, its formatting, and the printer model used. The spool request generated is stored in the TemSe (**Temporary Sequential file** table).

**Hint:** Spool requests can be created by dialog work processes or by background work processes. Spool work processes do **not create** spool requests.

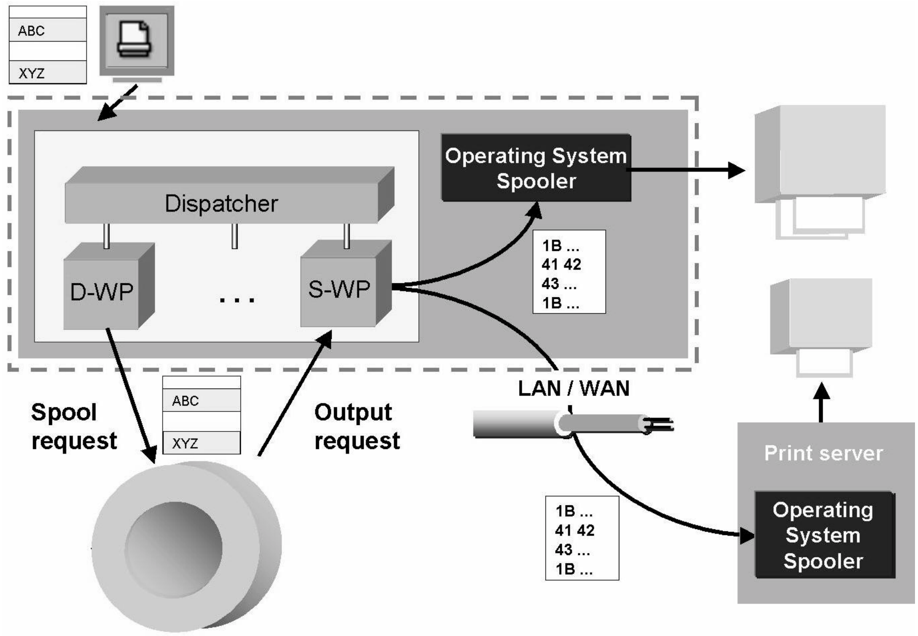


**Figure 34: Spool processing in an SAP system**

As shown in the above graphic, the spool work process prepares the data in the spool request and generates an **output request**, which contains all details in a format that the selected printer model can "understand". This data can either be passed on to an appropriate operating system spool process "locally", that is, on the same computer, or "remotely", that is, over a network connection.

**Hint:** In an SAP system, the connection between a spool work process and the operating system spool process is known as the "access method". There are more access methods than the two mentioned above, "remote" and "local". These are the two most commonly used access methods for connecting printers to SAP systems. In this context, "remote" or "local" do not refer to the physical location of the output, but to the place where the spool work process is "connected" to the operation system spool process.

For print processing, the best performance is achieved by sending the data to be printed to the operating system as soon as possible. You do this using the local access method. The operating system then performs all remaining tasks, such as queuing and data transfer to the selected printer.

**Note:** One minor but indispensable requirement for printing from SAP systems is that each selectable printer allows printing at operating system level.

## Demonstration: Printing from an SAP System

### Purpose

In this demonstration, you can see how spool requests and output requests are created.

### System Data

| | |
|---|---|
| **System:** | as specified by your instructor |
| **Client:** | as specified by your instructor |
| **User ID:** | as specified by your instructor |
| **Password:** | as specified by your instructor |
| **Set up instructions:** | The prerequisite for this demonstration is that, in the user profile *System* → *User Profile* → *Own Data* → *Defaults*, |

the setting “Output immediately” is **not** selected.

1. In a transaction that has a list output (for example, SM50), choose the print pushbutton (or Ctrl + P). Do not change any of the entries in the dialog box that appears and choose *Continue*. You have now created a spool request.
2. On this screen, choose *System* → *Own Spool Requests*. Information on the spool request you have just created is displayed. To view the output requests for a spool request, select the spool request and click on the *Output requests* pushbutton (or press F5). The message you receive for your spool request is “List contains no data”. Go back to the previous screen and choose *Print directly* for your spool request. This creates an output request that you can now display. Try to find out which device will carry out the printing, and trace what happens to your print-out (from the system’s point of view).

# 109 Exercise 9: Spool Processing

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Repeat the instructor's system demonstration and define the difference between spool and output requests.

## Business Example

System administration needs a list of the work processes set up on each instance.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task Printing a Simple List

Use the *Print* pushbutton to create a spool request for a simple list. Then create an output request for it.

1. Create a spool request for a simple list. To do this, call transaction SM50 and choose *Print*. Confirm the subsequent dialog box by choosing *Continue*. Note the message in the status bar.

2. Display your own spool requests. To do this, choose *System → Own Spool Requests*. Check the status and the number of pages of your spool request. Choose *Print directly* to create an output request for the spool request you selected. Now choose *Output requests* to display the output requests for your spool request. The status text is "Waiting for output formatter". Since there is no printer connected to the training system, you will not be able to print.

# Solution 9: Spool Processing

## Task Printing a Simple List

Use the *Print* pushbutton to create a spool request for a simple list. Then create an output request for it.

1. Create a spool request for a simple list. To do this, call transaction SM50 and choose *Print*. Confirm the subsequent dialog box by choosing *Continue*. Note the message in the status bar.

   a) Enter the transaction code SM50 in the command field, if appropriate enter /NSM50. Then choose *Print (Ctrl + P)*. Do not make any changes in the dialog box that appears, simply choose *Continue*.

2. Display your own spool requests. To do this, choose *System → Own Spool Requests*. Check the status and the number of pages of your spool request. Choose *Print directly* to create an output request for the spool request you selected. Now choose *Output requests* to display the output requests for your spool request. The status text is “Waiting for output formatter”. Since there is no printer connected to the training system, you will not be able to print.

   a) For the procedure, see the exercise.

## Lesson Summary

You should now be able to:

- Describe the print process flow
- Use various concepts fundamental to the spool environment correctly

### Related Information

- See the SAP Library and more advanced courses such as **ADM100**.

# Lesson: Background Processing

112 Lesson Duration: 40 minutes

## Lesson Overview

User requests can either be processed interactively by dialog work processes, or they can be processed by background work processes. This lesson gives you an overview of how background processing works in SAP systems. You will also schedule your own background job and monitor its execution.

## Lesson Objectives

After completing this lesson, you will be able to:

- Explain the difference between dialog processing and background processing
- Name areas where background processing is used
- Schedule simple jobs using the Job Wizard and monitor their execution and results

In this lesson the participants should gain an overview of the essential features of background processing, and an understanding of the differences between background processing and dialog processing. The participants should familiarize themselves with transactions SM36 (Job Wizard) and SM37.

## Business Example

As a system administrator or end user, it is your task to schedule a background job and to monitor its execution.

## The Concept of Background Processing

You may find it useful to base this section on a question and answer session. You could, for example, ask what background processing could be used for (and maybe collect answers on an overhead transparency) and discuss the participants' answers.

There are generally fewer work processes configured on an SAP instance than there are users logged on to it. Work processes need to handle user requests as quickly as possible so that requests from a number of users can all be processed with relatively short response times. This means that dialog work processes that are responsible for interactive requests should not be handling resource-intensive requests. If dialog work processes had to handle long-running dialog steps, they would not be available to other users for the duration of these steps. This would result in the remaining work processes having to serve considerably more users during that time, which would in turn result in significantly worse response times.

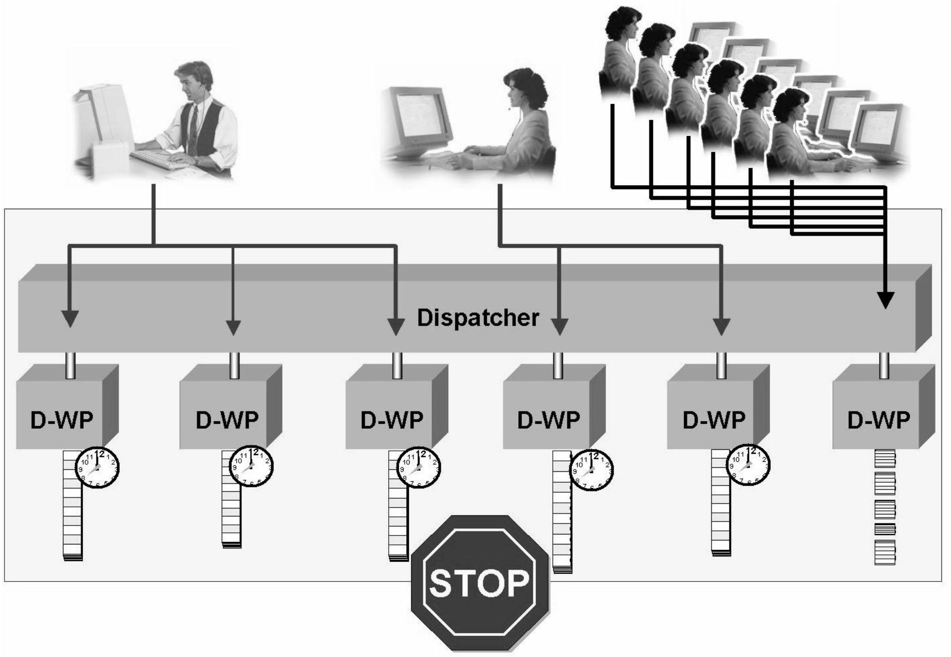


**Figure 35: The problem caused by long-running ABAP programs**

The profile parameter `rdisp/max_wprun_time` exists to avoid situations such as the one illustrated by the graphic. It defines the maximum time for which a dialog work process can take to process a dialog step (without a "commit" in the interim). The standard setting for this parameter is 600 seconds; if this is exceeded, the dialog step terminates and the transaction ends with an error message. Setting this parameter allows the administrator to ensure that users do not start long-running transactions in the dialog, but in background work processes that are specifically designed for long-running activities.

**Note:** To find out the current value of the parameter `rdisp/max_wprun_time` , you can, for example use transaction RZ11 (Maintain Profile Parameters) by entering the name of the parameter. A parameter value of 0 means that a time restriction for dialog steps has not been set. You can find additional information on this parameter in SAP Note 25528.

Along with long-running requests, background work processes typically also handle **periodic tasks** because after you have scheduled them once, you can automate periodic tasks to run at regular intervals. You can also use background work processes to automate data transfers from non-SAP systems to SAP systems.

The prerequisite for a user request to be executed in the background is that the request does not require any further user interaction. If, while processing a request, the SAP system does require further entries to be made, then these entries must be available when processing starts. This is accomplished using variants.

The results of a request processed in the background are – unlike the results of interactive dialog processing – not automatically displayed on the user's screen, but are stored in the TemSe (abbreviation of temporary sequential file; data file used to temporarily store data such as spool and background processing data).

You could demonstrate the use of variants in background processing by, for example, executing report RSPFPAR, since this report needs additional user input during processing. You can execute the report twice in the system (limiting the parameters to "rdisp*" each time is sufficient):

- Execute the report in the dialog, and draw the participants' attention to parameter `rdisp/max_wprun_time` and its value in the output list if appropriate
- Create a variant (using *Goto →Variants*) in transaction SA38, then press *Execute with variant*, and explain the difference between the two methods

You can use the variant you create here in the next section when scheduling jobs.

## Defining and Scheduling Jobs

Background processing is scheduled in the form of jobs. Each job consists of a name and one or more steps. A step can be an ABAP report, an external program, or another operating system call. You can also assign priorities to jobs (from A (highest) to C (lowest)).

The processing of a job is not generally triggered immediately (immediate start). You normally define a specific start time when scheduling a job. You can also start jobs periodically (for example, in the SAP system, certain system control jobs have defined recurrence cycles). You also have the option of linking the start of background processing to the occurrence of a specific event. Events can be triggered within the SAP system by ABAP programs or at operating system level (by the "sapevt" program).

You define and schedule jobs in an SAP system in transaction SM36 (Define Background Job); you can also call this transaction using *System* → *Services* → *Jobs* → *Define Job*. On the initial screen of this transaction, you have the option of using the Job Wizard to guide you through the steps involved in creating jobs.

The following graphic illustrates the technical process for a scheduled job.

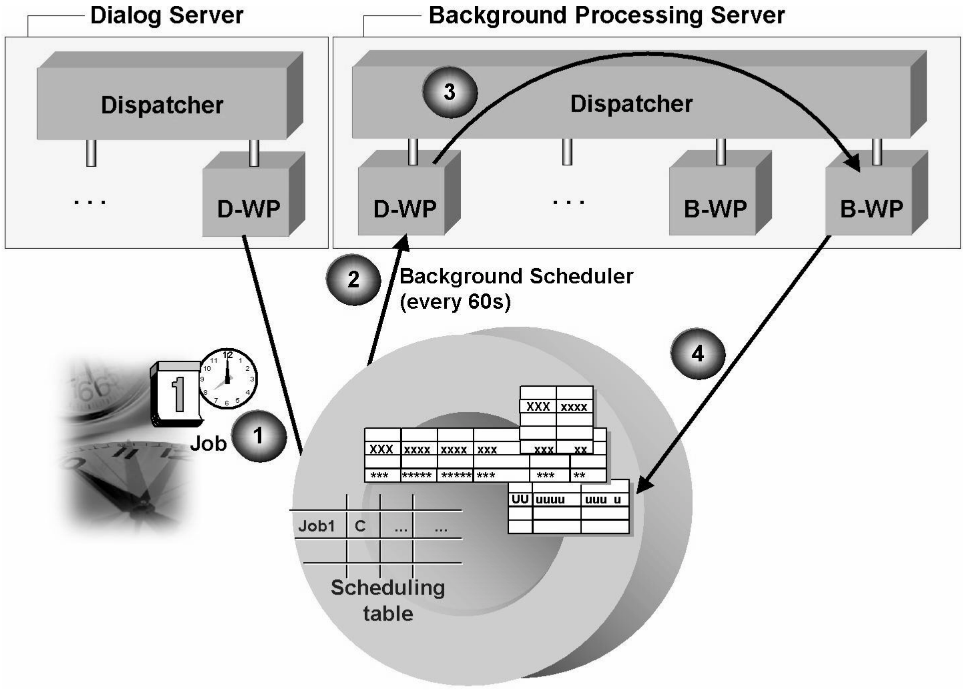


Figure 36: Scheduling and executing background jobs

Once the job has been defined (through interaction with the system, using a dialog work process), it exists as an entry in the scheduling table at database level. The background scheduler (ABAP program that the dispatcher starts in a dialog work process) periodically searches this table for jobs whose start time is imminent or already past.

**Hint:** The background scheduler is classified as a time-controlled job scheduler. This classification is to differentiate it from the event-controlled job scheduler.

You should no longer use the obsolete term *Batch Scheduler*, but rather refer to the *Background Scheduler* or the (time-controlled) *Job Scheduler*. Even if relevant system parameter still refer to *Batch Processing*, which actually only means stack processing, terms for work processes and schedulers have been changed to use background. You should therefore also use the terms that the participants will find in the parameter documentation and in the SAP Library.

The background scheduler's runtime is controlled by the parameter `rdisp/btctime`, which has a default setting of 60 seconds. If a job is scheduled to be executed and a background work process is available, the background scheduler passes the job to be processed to the available work process. This in turn executes the job.

**Caution:** The parameter value `rdisp/btctime = 0` disables the background scheduler, so jobs scheduled for a specific time cannot be executed.

## Monitoring Jobs

You can use transaction SM37 (Simple Job Selection) to monitor jobs; you can call this transaction from the initial screen of SM36 using *Job selection*, or from any screen using *System* → *Services* → *Jobs* → *Job Overview*.

On the initial screen of SM37, you can specify selection criteria (such as job name, user, job status, and job start time) for the jobs to be displayed. The next screen of this transaction displays a list of all the jobs that conform to your selection criteria. You can use this list to display various job details, for example:

- Display the job log using the *Job log* pushbutton
- Display the output lists for ABAP reports (if any exist) using the spool list

You may find it helpful to illustrate the above material by creating a simple job using the Job Wizard (for example, executing report RSPFPAR with the variant you created in the last section as the only step). If you schedule this job for a specific time (current time plus a minute or two), then you can monitor the job in SM37 and point out any delays in execution (caused by the background scheduler). You can also show the participants the spool list created by the report.

# 117 Exercise 10: Background Processing

Exercise Duration: 25 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Outline the difference between dialog processing and background processing and schedule a simple report in the background.

## Business Example

As system administrator or end user, you need to schedule the execution of a report in the background.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:** None

## Task Scheduling and Monitoring Jobs

Differentiate between executing a report in the dialog and executing it in the background; schedule the execution of a simple job.

1. Create a variant for report RSPFPAR in transaction SA38 (*System → Services → Reporting*). To do this, enter the report name on the initial screen of the transaction, then choose *Goto → Variants (F7)*. On the next selection screen, enter a name for the variant (for example, **`TEST##`**, where ## is your group number) and choose *Create*. On the next selection screen, restrict the profile parameters to **`rdisp*`**, enter a meaningful short description for the variant under *Attributes*, and save the variant.

2. Execute report RSPFPAR once in dialog without a variant and once with the variant you created (or another variant specified by your instructor). What is the difference between the two alternatives?

3. Use transaction SA38 to execute report RSPFPAR in the background. Choose the variant you created (or a variant specified by your instructor) and choose *Execute immed.*. What happens?

*Continued on next page*

4. Schedule a background job named “RSPFPAR_##” (where ## is your group number) using the Job Wizard in transaction SM36 (*System → Services → Jobs → Define Job*). This job should consist of a single step in which report RSPFPAR is executed with the variant that you created (or were given by the instructor). The job should have low priority and start 1-2 minutes in the future.

5. Examine the background job in detail in transaction SM37 (*System → Services →Jobs → Job Overview,* or the *Job selection* pushbutton in transaction SM36 ) by making an appropriate job selection on the initial screen. Refresh the display a few times in the job overview. Once your job has been executed, you can display the spool list that your job has created.

# Solution 10: Background Processing

## Task Scheduling and Monitoring Jobs

Differentiate between executing a report in the dialog and executing it in the background; schedule the execution of a simple job.

1. Create a variant for report RSPFPAR in transaction SA38 (*System → Services → Reporting*). To do this, enter the report name on the initial screen of the transaction, then choose *Goto → Variants (F7)*. On the next selection screen, enter a name for the variant (for example, **`TEST##`**, where ## is your group number) and choose *Create*. On the next selection screen, restrict the profile parameters to **`rdisp*`**, enter a meaningful short description for the variant under *Attributes*, and save the variant.

   a) For the procedure, see the exercise.

2. Execute report RSPFPAR once in dialog without a variant and once with the variant you created (or another variant specified by your instructor). What is the difference between the two alternatives?

   a) When you execute the report in the dialog (by choosing *Execute (F8)*), a selection screen is displayed that has no default entries in the input fields. When you *Execute with variant*, the fields on the selection screen are already filled with values in accordance with the variant you defined. This enables you to equip the report with all the user entries it needs before you run it. As a result, the report with the variant can be executed as a background job.

3. Use transaction SA38 to execute report RSPFPAR in the background. Choose the variant you created (or a variant specified by your instructor) and choose *Execute immed.*. What happens?

   a) For the procedure, see the exercise. A message appears in the status bar indicating that report RSPFPAR was started as a background job. The results of the report are not displayed on the screen.

4. Schedule a background job named "RSPFPAR_##" (where ## is your group number) using the Job Wizard in transaction SM36 (*System → Services → Jobs → Define Job*). This job should consist of a single step in

*Continued on next page*

which report RSPFPAR is executed with the variant that you created (or were given by the instructor). The job should have low priority and start 1-2 minutes in the future.

a) For the procedure, see the exercise. The Job Wizard guides you through the definition of the job. When defining a job step, choose "ABAP program step". Ensure that you use the system time when specifying the start time. You can display the system time (in a separate session) using *System → Status*.

5. Examine the background job in detail in transaction SM37 (*System → Services →Jobs → Job Overview*, or the *Job selection* pushbutton in transaction SM36 ) by making an appropriate job selection on the initial screen. Refresh the display a few times in the job overview. Once your job has been executed, you can display the spool list that your job has created.

a) For the procedure, see the exercise. Choose *Execute (F8)* on the initial selection screen of transaction SM37 . If you run the job overview within the 1-2 minutes specified, then you will see (possibly along with the processed job from the previous exercise) the job you have just created using the Job Wizard, with status "Released". Keep choosing *Refresh (F8)* at intervals of a few seconds and watch your job being processed. To display the spool list created by your job, select your job and then choose the *Spool* pushbutton. Your job has created a spool list. Select your spool list and then choose the pushbutton with the glasses (this icon generally represents display in SAP systems). The spool list displays an overview of all profile parameters in accordance with the selection you made when defining the variant.

## Lesson Summary

You should now be able to:

- Explain the difference between dialog processing and background processing
- Name areas where background processing is used
- Schedule simple jobs using the Job Wizard and monitor their execution and results

### Related Information

In this lesson, you learned about the basics of background processing. You can find out more about this topic by attending more advanced training courses or in the online help under *Getting Started* → *Background Processing* or under *SAP Web Application Server* → *Computing Center Management System* → *Computing Center Management System* → *Background Processing*.

## Unit Summary

You should now be able to:

- Outline simple client/server configurations
- Describe the processing flow for user requests in SAP systems
- List various types of SAP GUI
- Outline how SAP Web Application Server and SAP GUI communicate, and how SAP Web Application Server and the database communicate
- Name the most important processes on an SAP Web Application Server
- Define the term instance and recognize the characteristics of a central instance
- Outline the processing flow for a dialog step in the SAP system
- Describe the concept of work process multiplexing
- Define the term transaction
- Outline differences between a database transaction and an SAP transaction
- Recognize the significance of a lock mechanism for the consistency of business data
- Outline the flow for a lock request
- Outline the principle of asynchronous updates
- Explain how updates are processed in the system
- Describe the print process flow
- Use various concepts fundamental to the spool environment correctly
- Explain the difference between dialog processing and background processing
- Name areas where background processing is used
- Schedule simple jobs using the Job Wizard and monitor their execution and results

123

# Test Your Knowledge

1. What are the advantages of a three-tier client/server configuration as compared to a single-tier or two-tier configuration?

   *Choose the correct answer(s).*

   □ A Simpler scalability
   □ B Simpler administration
   □ C Load balancing possible
   □ D None of the above

2. Which process at the application server level receives the user request?

   *Choose the correct answer(s).*

   □ A work process
   □ B dispatcher
   □ C The buffer
   □ D SAP GUI

3. What types of SAP GUI are there?

   *Choose the correct answer(s).*

   □ A SAP GUI for the Windows environment
   □ B SAP GUI for the Java environment
   □ C SAP GUI for Linux
   □ D SAP GUI for HTML

4. The SAP Web Application Server database interface enables you to:

   *Choose the correct answer(s).*

   □ A Enhance performance using the local buffers
   □ B Access online databases on the Internet
   □ C Use database-independent Open SQL in ABAP programs
   □ D Access file content on the SAP Web Application Server

5. Which work process types can you find on an SAP Web Application Server?

   *Choose the correct answer(s).*

   □ A Dialog work process
   □ B Message server work process
   □ C Update work process
   □ D ICM work process
   □ E Background work process
   □ F enqueue work process
   □ G Spool work process

6. The term "central instance" describes the instance

   *Choose the correct answer(s).*

   □ A that you are logged on to while working in the system
   □ B On which all application processes required for operating an SAP system are configured
   □ C On which the gateway process is configured
   □ D On which the majority of work processes are configured
   □ E On which the majority of dialog work processes are configured

7. What tasks does the task handler carry out?

   *Choose the correct answer(s).*

   □ A Communicates with SAP GUI
   □ B Coordinates activities within the work process
   □ C Accesses the database
   □ D Processes the ABAP coding on which a transaction is based

8. Is the assignment of users to dialog work processes fixed?

   *Choose the correct answer(s).*

   □ A Yes, it is fixed for the entire time that the user is logged on to the SAP system
   □ B Yes, it exists for the entire duration of a transaction consisting of several screens
   □ C No. Each dialog step of a transaction, consisting of several screens, can theoretically be processed by a different work process
   □ D None of the above

9. Database transactions are based on the ACID principle. So are SAP transactions. Why does the database transaction concept nevertheless not cover all the needs of the SAP system?

   *Choose the correct answer(s).*

   □ A An SAP transaction can bundle several database transactions. It is only as a result of all of these bundled database transactions that the data is consistent in business terms.

   □ B Due to work process multiplexing, each work process starts a separate database transaction within each dialog step.

   □ C This is because SAP developers have not done their job properly.

   □ D Using the SAP transaction as a structural level above the database transaction significantly improves the performance of your SAP system.

10. Where is the lock table?

    *Choose the correct answer(s).*

    □ A In the database

    □ B In the main memory of the application server on which the enqueue work process is configured

    □ C At operating system level of the database host

    □ D The system administrator can define the location of the lock table using the profile parameter `renq/store_location`

11. It makes sense, in an SAP system, to configure several enqueue work processes on different SAP Web Application Servers.

    *Determine whether this statement is true or false.*

    □ True

    □ False

12. Why are asynchronous updates used for dialog transactions?

    *Choose the correct answer(s).*

    □ A Only asynchronous updates can access the SAP system buffers

    □ B Because an SAP transaction can consist of several database transactions and this approach means that the requirements for the rollback are met

    □ C Because only update work processes can access the database data

    □ D None of the above

13. During asynchronous updates, which process records the data to be changed in VB* tables?

    *Choose the correct answer(s).*

    □ A Update work process

    □ B dispatcher

    □ C Recording work process

    □ D Dialog work process

14. Answer the following questions with reference to the two access methods you have learned about, "remote" and "local". Other answers may be correct for other access methods. Which work process types are primarily responsible for **creating** spool requests?

    *Choose the correct answer(s).*

    □ A Dialog

    □ B Update

    □ C Spool

    □ D Background

    □ E Enqueue

15. Which processes and hardware elements are involved (and in what order) in handling the print process using the "remote" access method? The flow is only outlined here.

    *Choose the correct answer(s).*

    □ A *S-WP → Operating system → Network → Printer*

    □ B *SAP output formatting program → Network → SAP GUI → Front-end CPU → Printer*

    □ C *D-WP → TemSe → S-WP → Network → Operating system spool → Printer*

    □ D *D-WP → S-WP → Database → Network → TemSe → Operating system spool → Printer*

    □ E *D-WP → S-WP → TemSe → Network → Printer*

16. What is background processing used for?

    *Choose the correct answer(s).*

    □ A Periodic tasks

    □ B Tasks that require user interaction

    □ C Long-running tasks

    □ D Printing lists

127

## Answers

1. What are the advantages of a three-tier client/server configuration as compared to a single-tier or two-tier configuration?

   **Answer:** A, C

   Implementing an additional hardware layer for application processes makes it easier to adapt an SAP system if the number of users changes (scalability), and to assign user groups to specific application servers (software-oriented view), (load balancing). The additional hardware layer does not, however, reduce the administrative workload.

2. Which process at the application server level receives the user request?

   **Answer:** B

   The dispatcher receives the user request on the application server 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 buffer enables quicker processing of user requests.

3. What types of SAP GUI are there?

   **Answer:** A, B, D

   There is a SAP GUI for the Windows environment, based on the Windows platform, there is a platform-independent SAP GUI for the Java environment, and a SAP GUI for HTML, consisting of a browser and an Internet Transaction Server. There is no such thing as SAP GUI for Linux.

4. The SAP Web Application Server database interface enables you to:

   **Answer:** A, C

   When the ABAP Interpreter sends a query to the database, the database interface reads the data from the SAP Web Application Server's local buffer, if possible. If the data required is not present in the buffer, then the interface turns the Open SQL statement from the ABAP coding into database-specific native SQL coding. The database interface is not used to access the content of online databases on the Internet or files on the SAP Web Application Server.

5. Which work process types can you find on an SAP Web Application Server?

   **Answer:** A, C, E, F, G

   All of the above processes can in theory be configured on an SAP Web Application Server. However, not all of the above processes are work processes. The message server and ICM process are not work processes.

6. The term "central instance" describes the instance

   **Answer:** B

   The central instance is the instance on which all application processes required for operating an SAP system are configured. Specifically, the central instance therefore also contains processes that are only present once per system (such as the message server). Every instance has a gateway process, the existence of a gateway process does not therefore denote the central instance. The number of dialog work processes, or the total number of work processes configured on the central instance is not relevant in this context.

7. What tasks does the task handler carry out?

   **Answer:** B

   The task handler coordinates activities within a work process. The work process and the user's SAP GUI do not communicate directly, but through the dispatcher. The database is accessed through the database interface and the ABAP coding is executed by the ABAP processor.

8. Is the assignment of users to dialog work processes fixed?

   **Answer:** C

   The dispatcher reassigns the user to a dialog work process for each dialog step in a transaction. The assignment of users to dialog work processes is therefore not fixed. (The only exception to this rule is the debugging mode, in which one dialog process is assigned to one specific user.)

9. Database transactions are based on the ACID principle. So are SAP transactions. Why does the database transaction concept nevertheless not cover all the needs of the SAP system?

   **Answer:** A, B

   Every dialog work process starts a new database transaction within each dialog step. An SAP transaction can bundle several database transactions into units that makes sense in business terms – this is not related to performance issues.

10. Where is the lock table?

    **Answer:** B

    The lock table is located in the main memory of the SAP Web Application Server on which the enqueue work process is configured.

11. It makes sense, in an SAP system, to configure several enqueue work processes on different SAP Web Application Servers.

    **Answer:** False

    It does **not** make sense to configure enqueue work processes on different SAP Web Application Servers, since there can be only one lock table per SAP system and this lock table is located in the main memory of the SAP Web Application Server on which the enqueue work process is configured.

12. Why are asynchronous updates used for dialog transactions?

    **Answer:** B

    Asynchronous updates solve the problems caused by the different interpretations of transaction at database level and at SAP level. Bundling all updates for one SAP transaction into a single database transaction ensures that the data that belongs to this SAP transaction can be rolled back completely. Every work process is continuously connected to the database, therefore every work process can access the database dataset. Access to the SAP system buffers does not depend on the type of update used.

13. During asynchronous updates, which process records the data to be changed in VB* tables?

    **Answer:** D

    The dialog process records the data to be changed in the VB* tables, while the update work process handles the final update of the target tables (the tables that are actually relevant to business). There is no such thing as a recording work process.

14. Answer the following questions with reference to the two access methods you have learned about, "remote" and "local". Other answers may be correct for other access methods. Which work process types are primarily responsible for **creating** spool requests?

    **Answer:** A, D

    Most printing requests are started in the dialog or in background processing.

15. Which processes and hardware elements are involved (and in what order) in handling the print process using the "remote" access method? The flow is only outlined here.

    **Answer:** C

    Depending on the settings of the relevant system parameters, the TemSe might be located in different "places": normally it is either in the database or at file-system level on the central instance.

16. What is background processing used for?

    **Answer:** A, C

    Background processing is typically used for long-running tasks or tasks that recur regularly. The prerequisite is that these tasks do not require user interaction. The spool work process is responsible for printing lists.

# Unit 4

131

## The Workbench

This unit can only be introductory in nature (partly because of the many more advanced courses in the ABAP area). The participants need to understand that – due to the data structure in SAP systems – they must not carry out development activities in production systems. You do not need to go into too much detail in the lessons on working with the ABAP Dictionary or accessing and editing Repository objects. The participants merely need to get a feel for using the relevant ABAP Workbench tools. This overview is generally all that future system administrators need to know, and the developers among the participants usually acquire the in-depth knowledge they need by attending more advanced courses on the ABAP Workbench.

The content of this unit primarily refers to "classic" ABAP programming. The creation of Business Server Pages, which is possible as of SAP Web AS 6.10, is not covered in this unit, but is briefly discussed in the unit "Communication and Integration Technologies". Object-oriented programming is not discussed in the course documents. You can, however, bring up the subject, if you are familiar with the relevant ABAP Workbench tools, such as the Class Builder (in the lesson "Accessing and Editing Repository Objects").

Ideally, you should have finished the lesson "Working with the ABAP Dictionary" by the end of the second day.

### Unit Overview

This lesson introduces you to the ABAP Workbench as the central development environment in SAP systems; it also discusses some ABAP Workbench tools. In addition, you gain an overview of the data structure of an SAP system and, in this context, understand the significance of transports between various SAP systems.

## Unit Objectives

After completing this unit, you will be able to:

- Describe the data structure of SAP systems
- Name the various data types
- Describe the three-system landscape as recommended by SAP
- Explain the importance of transports
- Describe the procedure for transports
- Name important tools in the ABAP Workbench
- Access Repository objects
- Write a simple ABAP program
- Explain the significance of the ABAP Dictionary
- Outline the two-level domain concept
- Explain the significance of foreign key dependencies for ensuring data consistency
- Describe the entire development cycle for an object

## Unit Contents

# Lesson: Data Structure of SAP Systems

133

Lesson Duration: 30 minutes

## Lesson Overview

This lesson introduces the data structure of SAP systems. The lesson explains this data structure and and its impact on the way you develop your own objects and adapt SAP objects.

## Lesson Objectives

After completing this lesson, you will be able to:

- Describe the data structure of SAP systems
- Name the various data types

This lesson covers the data structure of SAP systems. It introduces the data types “client-specific”, “cross-client Customizing” and “Repository”. Examples of client-specific data are user data and application data such as invoices, material master records, and so on. Client-specific Customizing includes company codes, plants and storage locations. Cross-client Customizing includes the public holiday calendar, for example, and settings for the transport domain and system change option. Repository objects are ABAP Workbench development objects such as program objects, function group objects, Dictionary objects, and so on.

You need to point out the effects of changing Customizing and Repository data, and the resulting need for a three-system landscape. You also need to introduce and compare the possibilities of enhancing the SAP standard system using in-house developments, customer exits, and modifications.

## Business Example

As a developer in an SAP system, you have extensive change authorizations. Therefore, an understanding of the data structure and the development and transport process between SAP systems is fundamental to your work.

## Data Structure of SAP Systems

SAP systems have a specific data structure. Alongside the business settings (Customizing) that are only relevant to specific clients of an SAP system, every SAP system also always contains cross-application settings and objects.

You must differentiate between three data components in SAP systems:

- A **client** is a unit within an SAP system that is self-contained in terms of business, organization and data. A client is characterized by the fact that it has its own business data environment, its own master and transaction data, and its own user data. This data in a client is known as **client-specific data**. Client-dependent data types are closely interdependent. Business application data is checked against the Customizing settings for the client when this data is entered, and rejected if this data is inconsistent.
- When an SAP system is implemented, **Customizing** is set up for each client. You use Customizing settings to define the customer's organizational structures, such as distribution channels and company codes, in the system and to set parameters for SAP transactions so that they reflect customer-specific processes. In addition to the client-specific Customizing settings, there are also settings that only need to be made once to be valid for all clients in that SAP system. These cross-client Customizing settings include the public holiday calendar, for example, and settings for the system change option.
- The **Repository**, the central store for all ABAP Workbench development objects, is also cross-client. It contains all Dictionary objects (tables, data elements, domains, and so on), and also all ABAP programs, menus, and screens. Because of the Repository's cross-client nature, any Repository objects developed or changed in any one client are used in exactly the same form in every other client in that system.

  Repository objects are grouped together to form packages. Packages are containers for semantically related development objects (programs, tables, and so on) and replace development classes. Objects that are developed and transported together are bundled into one package. The Repository contains packages from all software components.

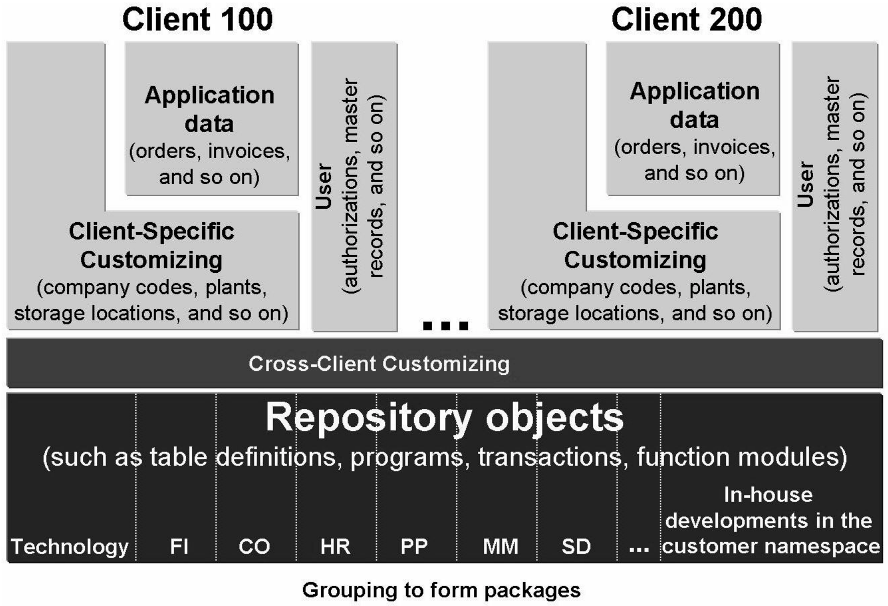

Client 100
Application data
(orders, invoices, and so on)
Client-Specific Customizing
(company codes, plants, storage locations, and so on)
User
(authorizations, master records, and so on)
Client 200
Application data
(orders, invoices, and so on)
Client-Specific Customizing
(company codes, plants, storage locations, and so on)
User
(authorizations, master records, and so on)
...
Cross-Client Customizing
Repository objects
(such as table definitions, programs, transactions, function modules)
Technology
FI
CO
HR
PP
MM
SD
...
In-house developments in the customer namespace
Grouping to form packages

Figure 37: Data structure of SAP systems

You can also create your own overhead transparency with the various data components to show the data structure of an SAP system. You need to include the individual elements as in the graphic, and draw more than one client if necessary. You can then keep referring back to the transparency during the lesson as required.

## Changing and Adapting the Data Structure

You can make changes or enhancements both in Customizing or to Repository objects.

### Working in Customizing

The SAP system software is standard business software that needs to be adapted to company-specific requirements when it is implemented. The process of adapting the software is known as Customizing. It includes both client-specific and cross-client settings. You may need to repeat Customizing – on a much smaller scale – when you upgrade your SAP system.

### Working in the Repository

Changes to Repository objects are not absolutely necessary for the use of an SAP system. However, you may need to change individual objects (for example, because coding repairs are suggested in an SAP Note). Changes or extensions to the Repository are possible in a variety of ways:

- Expanding the Repository through **in-house developments**. In the SAP system, you can create your own Repository objects, such as tables, programs, transactions, and so on. All customer development normally takes place in the customer namespace, that is, all the objects created by the customer have names from a specified namespace; these names generally start with the letters Y or Z. This is the procedure for ABAP programs, tables, and so on. SAP has also provided its customers with an additional method of globally unique assignment of individual namespaces for several years. Such a customer namespace could be, for example, "/<companyname>/".
- Changing the Repository through **customer enhancements**. In this type of change, you add customer-specific objects to the Repository. There are specific places in the coding, known as "customer exits", where you can supplement the SAP standard system with your own objects. There is also another, more up-to-date method for customer-specific enhancements to the SAP standard system: using "Business Add-Ins" (BAdIs).
- **Modifications** to the SAP standard code: changes to SAP objects such as tables, and table definitions, are known as modifications. The Repository delivered by SAP is not only extended, but changed. When you next upgrade your system or when importing Support Packages, you will therefore need to check these modifications against the new Repository. This modification check takes time, although using the "Modification Assistant" speeds up the process substantially.

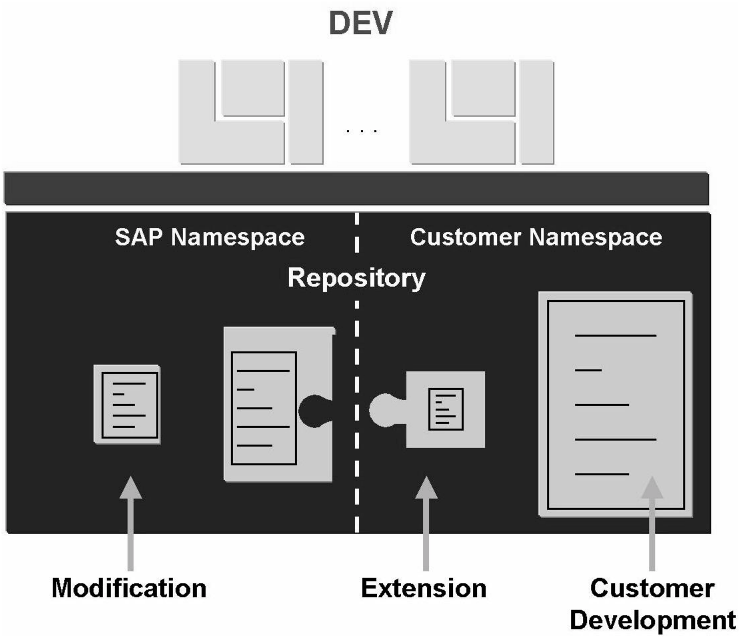


Figure 38: Changing the Repository

## Consequences and Effects of the Data Structure

Since the Repository objects are not client-specific, we strongly recommend that you do not use the same system for development and production. The risk of data inconsistency is too high. Therefore, SAP recommends that when you change the Repository and carry out cross-client Customizing, you use at least two and preferably three systems:

- One system that is used exclusively for development
- One system that is used for testing and quality assurance activities
- One system that is used exclusively for production operation

## Facilitated Discussion

With the participants, discuss the data components that make up an SAP system:

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

Ask questions on the points raised above.

- The Repository that is valid throughout the system and includes all Workbench objects
- Cross-client Customizing, also valid throughout the system with all cross-client Customizing settings
- The client structure, in which each client is an independent business entity

## Lesson Summary

You should now be able to:

- Describe the data structure of SAP systems
- Name the various data types

### Related Information

- For more information about this topic, attend the training course **BC425** - *Enhancements and Modifications*.

# Lesson: System Landscape and Transports

138 Lesson Duration: 30 minutes

## Lesson Overview

The three-system landscape recommended by SAP means that transports are important. Both developers and project managers need to have a good understanding of the structure and process for transports.

## Lesson Objectives

After completing this lesson, you will be able to:

- Describe the three-system landscape as recommended by SAP
- Explain the importance of transports
- Describe the procedure for transports

In this lesson you need to explain the structure of a three-system landscape and the reasons why this is the recommended landscape. You need to demonstrate the process for creating change requests.

## Business Example

You need to transport objects you have developed yourself, SAP objects, and Customizing settings to other SAP systems.

## The Three-System Landscape

The data structure of an SAP system has an impact on the way you work, particularly when changing data.

Since the Repository objects are not client-specific, you must not use the same system for development and production. The risk of data inconsistency or data loss is too high. Security considerations are also a powerful argument against development work being carried out in a production system.

To ensure system consistency, we recommend that you set up a system landscape consisting of three systems. Each of these three systems contains a working client and other clients as required. These three working clients should all have the same name to facilitate the consistency of Customizing settings. You can set up several SAP systems on one license, although you may only use one of these systems as a production system.

A three-system landscape facilitates the following recommended process:

- You develop your own programs and perform the required Customizing in the development system.
- All Customizing settings as well as changes (developments, corrections or modifications if required) to the Repository are transferred to the quality assurance system (or "test" system for short) to be checked there without influencing production.
- All objects and settings imported into the test system can be tested and then transferred into one or more production systems.

A three-system landscape also allows you to test upgrades, and helps to minimize downtime during upgrades of the production system.

The systems in a three-system landscape must have unique, three-character descriptions, for example, DEV, QAS and PRD. These abbreviations, which also occur in other courses, are used internationally in the SAP environment, and stand for:

- Development
- Quality assurance
- Production

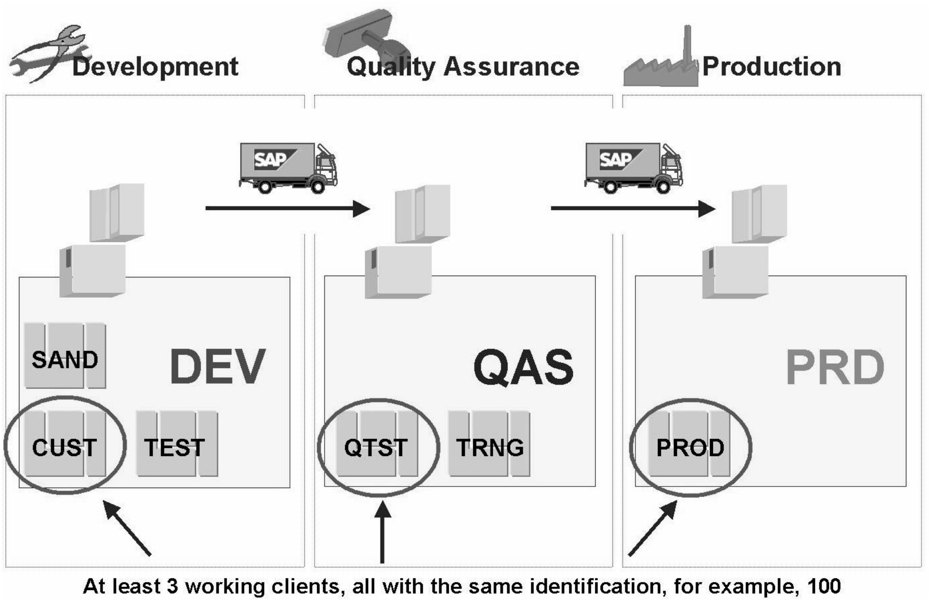

Development
Quality Assurance
Production
SAP
SAP
SAND
DEV
CUST
TEST
QAS
QTST
TRNG
PRD
PROD
At least 3 working clients, all with the same identification, for example, 100

**Figure 39: The three-system landscape**

## Transports

In a multi-system landscape, transports are used to transfer objects developed by the customer, SAP programs, and SAP tables from one system to another. To move programs in a multi-system landscape, you need change requests.

The Transport Organizer (transactions SE09 and SE10, menu path *Tools → ABAP Workbench → Overview → Transport Organizer*) logs changes to Repository objects and transports them as Workbench requests. The same approach is used to log and transport Customizing settings as Customizing requests.

Objects are transported using the following procedure: at the start of a development project, the development leader creates a change request. In doing so, he or she assigns the team members to the change request. The Transport Organizer SE09 assigns a number to the change request (<sid>K9<nnnnn>, so for example, DEVK900050). A request contains objects that are logically related and can only be constructively executed together. A request therefore enables the transport and administration of complete, constructive finished developments.

The Transport Organizer automatically creates a subtask for each development team member assigned to the change request. If an employee assigns a Repository object to the change request, the Repository object is logged in the task of that employee. At the end of the project, the task contains all of the Repository objects that the team member has worked on.

When the development project is completed from the point of view of the employee, he or she releases his or her task. This transfers the objects in the task to the change request. Once all team members have released their tasks, the development leader can release the change request. A change request therefore combines Repository objects that have been created or changed during a development project.

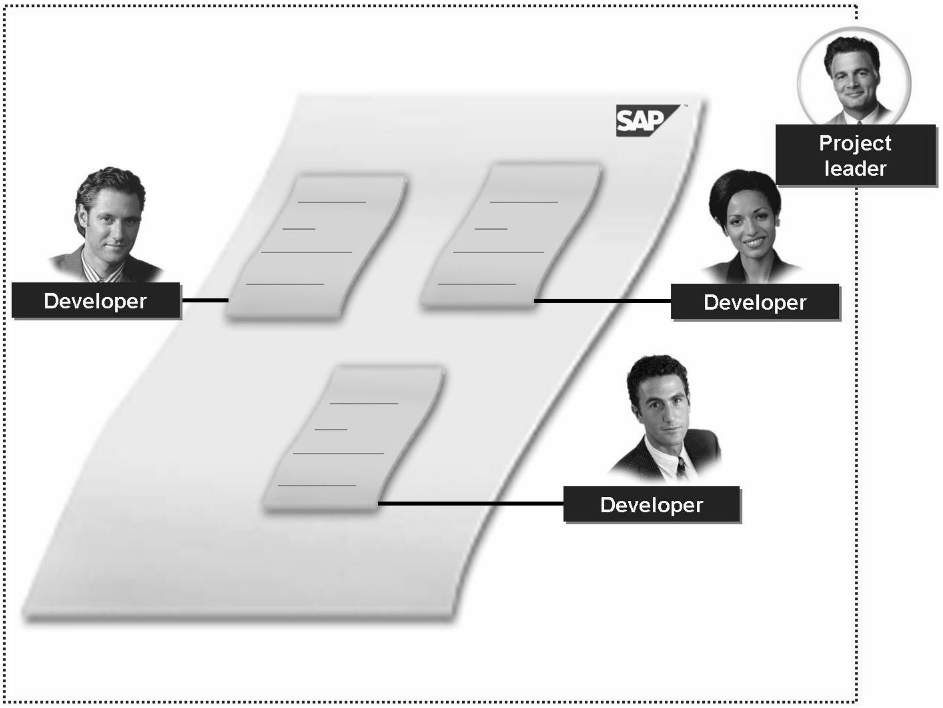


Figure 40: Change request and tasks assigned

At this point you could show the following system demonstration:

1. Call the Transport Organizer using *Tools* → *ABAP Workbench* → *Overview* → *Transport Organizer* or SE09 (or SE10).
2. Create a change request in the Transport Organizer (request type: Workbench request) and enter a description.
3. Show the participants the request and its subordinate task.

**Hint:** As of SAP R/3 4.6 C, the Workbench Organizer (SE09) and the Customizing Organizer (SE10) have been grouped together into a single transaction, the Transport Organizer. You can use either transaction code to call the Transport Organizer.

# 143 Exercise 11: Creating a Transport Request

Exercise Duration: 5 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Use the Transport Organizer.

## Business Example

You need to create a change request for a development project and assign team members to this request.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:**

1. The participants must be authorized to create a change request in transaction SE09.

## Task Working with Change Requests

Creating a change request:

1. Use the Transport Organizer to create a change request (request type: Workbench request).
2. Display the request you have created. How many tasks are assigned to this request?

# Solution 11: Creating a Transport Request

## Task Working with Change Requests

Creating a change request:

1. Use the Transport Organizer to create a change request (request type: Workbench request).

   a) Call the Transport Organizer using *Tools → ABAP Workbench → Overview → Transport Organizer* or transaction code SE09 (or SE10):

      On the initial screen of the Transport Organizer, choose *Create* and create a change request (request type: Workbench request). Confirm your entry. Enter a meaningful short description on the next screen.

2. Display the request you have created. How many tasks are assigned to this request?

   a) You can display the change request created for your user name by choosing *Display* on the initial screen of the Transport Organizer. One task is assigned to your change request (it is still “Unclassified”).

## Lesson Summary

You should now be able to:

- Describe the three-system landscape as recommended by SAP
- Explain the importance of transports
- Describe the procedure for transports

### Related Information

- For more, very detailed information on this topic, attend the training course **ADM325** - *Software Logistics*.

# Lesson: Accessing and Editing Repository Objects

146 Lesson Duration: 55 minutes

## Lesson Overview

The ABAP Workbench is the SAP system's integrated graphical development environment. It supports, among other things, the development, testing, and administration of applications written in ABAP. This lesson introduces various ABAP Workbench tools and the connections between them. At the end of this lesson, you will write your own short ABAP program.

## Lesson Objectives

After completing this lesson, you will be able to:

- Name important tools in the ABAP Workbench
- Access Repository objects
- Write a simple ABAP program

How you choose to structure this lesson depends to a great extent on your own knowledge. If you know a lot about the individual Repository tools mentioned in this lesson, do not hesitate to talk about them in detail and even give a brief demonstration of how they work. The primary objective of this lesson is not to teach the participants ABAP, but rather to give them a first impression of what tools are available for editing Repository objects and what exactly each of these tools do. Use the Object Navigator as a central starting point for editing Repository objects.

**Caution:** In the exercise for this lesson, the participants create Repository objects. These objects have been given generic names, which include, among other things, variables such as <CLNT>. It is frequently the case that participants working, for example, in client 800 call their program Z<800>... instead of Z800... . The angle brackets may mean that it is possible to save the object, but not to activate it. You should therefore explain that <...> simply means a variable and that the participants should enter the relevant **values without angle brackets**.

## Business Example

As a developer, you need to familiarize yourself with the most important ABAP Workbench tools and create your first ABAP program.

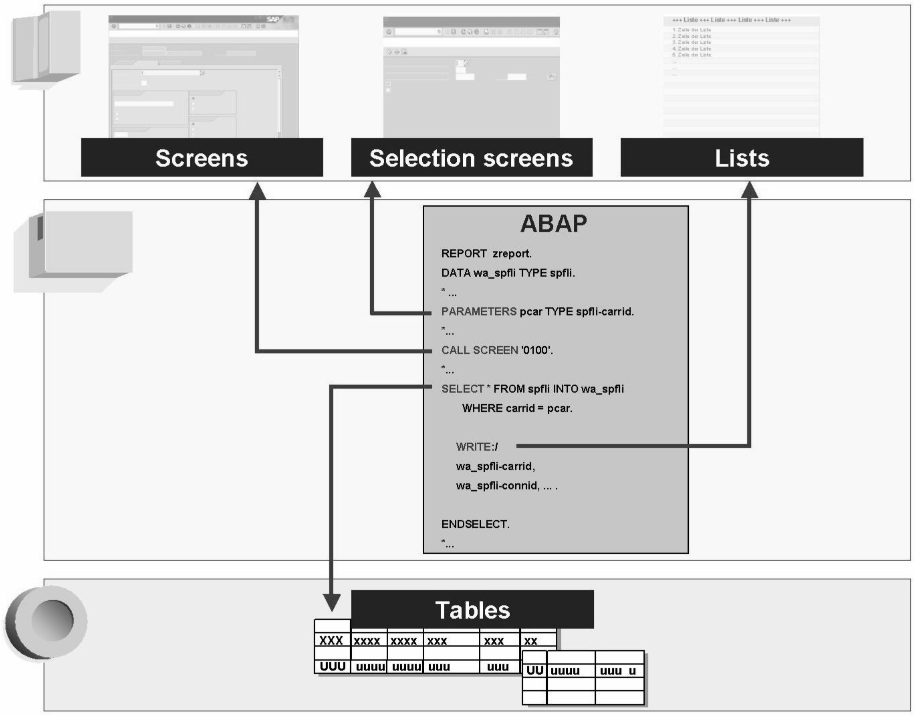
## The ABAP Language

ABAP (Advanced Business Application Programming) is a programming language developed by SAP. The majority of the business applications of an SAP system are written in ABAP. An ABAP program consists of individual statements. Every statement begins with a keyword and ends with a period.

The example program

```
REPORT first_report.
WRITE 'My first ABAP report.'
```

contains two statements, one on each line. The key words are `REPORT` and `WRITE`. The program displays a list. In this case, the list contains the line "My first ABAP report".



```
REPORT  zreport.
DATA wa_spfli TYPE spfli.
* ...
PARAMETERS pcar TYPE spfli-carrid.
*...
CALL SCREEN '0100'.
*...
SELECT * FROM spfli INTO wa_spfli
      WHERE carrid = pcar.

     WRITE:/
     wa_spfli-carrid,
     wa_spfli-connid, ... .

ENDSELECT.
*...
```

**Figure 41: The ABAP language**

The above graphic shows an excerpt from an ABAP program. You can use special commands or keywords in ABAP programs to create selection screens (keyword `PARAMETERS`), to print lists (keyword `WRITE`), or to access table content (for example, using the keyword `SELECT`). The ABAP statement `CALL SCREEN` calls a screen (consisting of a screen image and its flow logic) defined in the Screen Painter.

ABAP generally uses Open SQL commands to access the database. Open SQL consists of a set of ABAP statements that execute operations on the central database of the SAP system. These operations return the same results or error messages, regardless of the type of database used. This means that the programs developed are independent of the type of database used.

Some characteristics of the ABAP programming language are:

- Multilingual capability (text elements such as list headers, input field texts, and so on, are stored separately)
- Simple, effective development of graphical user interfaces (using the Screen Painter)
- Object-oriented programming (“ABAP Objects”)
- Platform independence (using Open SQL and the database interface)
- Efficient access to data structures (tables, data elements, and so on)

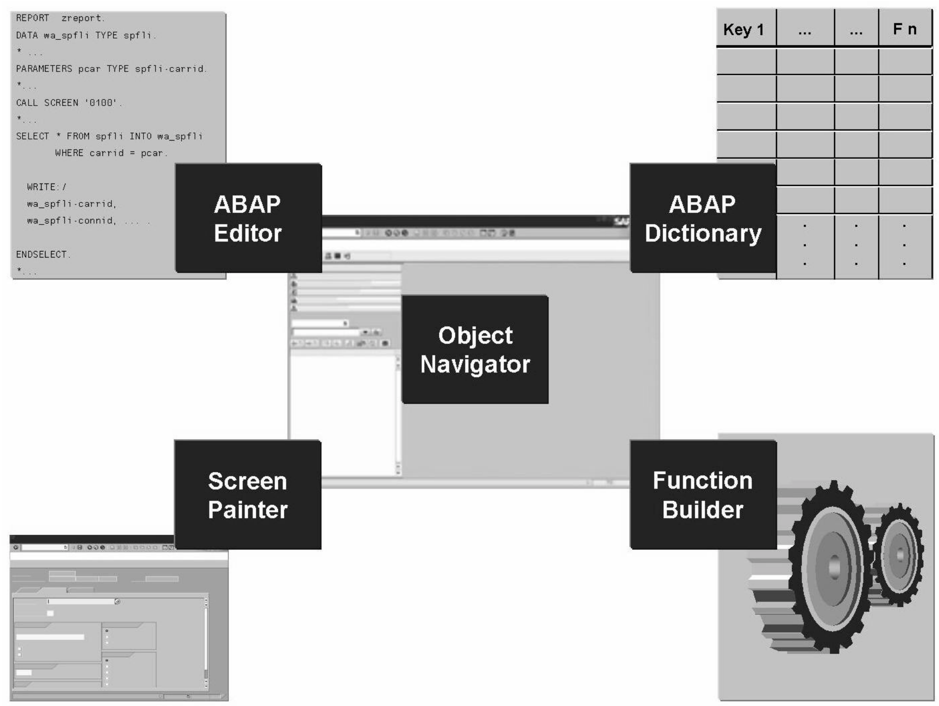
## The ABAP Workbench and its Tools

This section simply serves as a general introduction to the ABAP Workbench. The participants should get to know the Object Navigator as the central point of access to the individual tools. If you are familiar with the Class Builder, then you could discuss this tool briefly too.

You use the ABAP Workbench to write application programs. The Workbench is a graphical programming environment. The Workbench enables you to call programming tools, using pushbuttons, for example, or the context menu (right mouse click) or forward navigation (double-click on an object name). An ABAP application is, for example, either a transaction or a report.

You can find the ABAP Workbench tools in SAP Easy Access under *Tools* → *ABAP Workbench* → *Development*. From there, you can access a range of tools, including

- The ABAP Editor (transaction SE38), to write ABAP programs
- The ABAP Dictionary (transaction SE11), to define and describe tables, data elements, lock objects, and so on
- The Screen Painter (transaction SE51, in subdirectory *User Interface*), to create interactive user interfaces
- The Function Builder (transaction SE37), to create and manage function modules (these are encapsulated sections of ABAP code with a defined input/output interface



```
REPORT  zreport.
DATA wa_spfli TYPE spfli.
* ...
PARAMETERS pcar TYPE spfli-carrid.
*...
CALL SCREEN '0100'.
*...
SELECT * FROM spfli INTO wa_spfli
       WHERE carrid = pcar.

  WRITE:/
  wa_spfli-carrid,
  wa_spfli-connid, ... .

ENDSELECT.
*...
```

**Figure 42: Some ABAP Workbench tools**

The individual Workbench tools combine to form an integrated system. If, for example, you are working with program objects in the ABAP Editor, then the Editor will also recognize objects created using other tools. Double-clicking on an object automatically starts the Workbench tool used to create that object, allowing you to edit the object.

When working in the Workbench, you will come across development objects and packages:

- Development objects are objects that you can edit using the ABAP Workbench, such as reports, transactions or screens
- A package contains logically related development objects, for example, all objects for a specific application

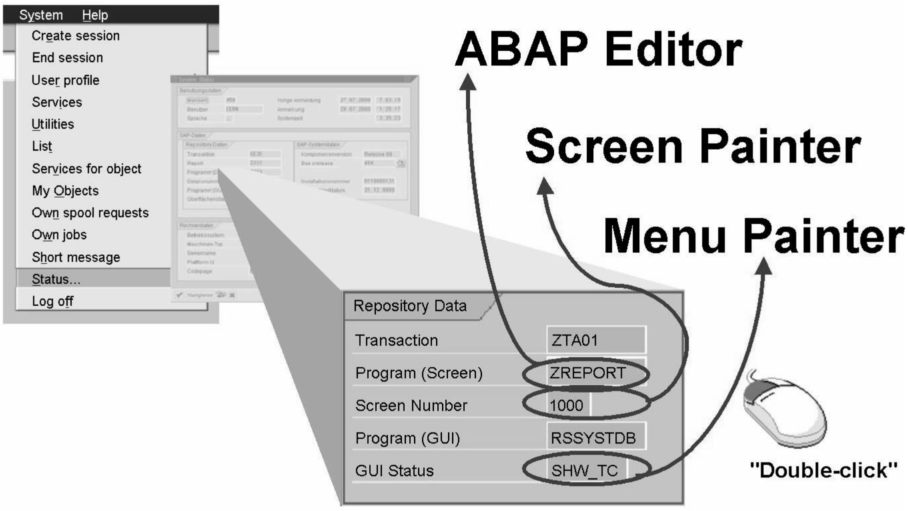
To help you organize your development processes in the integrated ABAP Workbench environment, SAP provides the Object Navigator (transaction SE80, menu path *Tools* → *ABAP Workbench* → *Overview* → *Object Navigator*). This enables simple, uniform access to Repository objects. Instead of working with tools and packages, you can work with objects in the Object Navigator, and the Workbench will call the right tool for each object.

At this point you could demonstrate to the participants how to use the central Object Navigator access for selected Workbench tools. For example, in transaction SE80, you can display program RSPFPAR. Double-clicking on the "RSPFPAR" directory takes you to the ABAP Editor, with the relevant source code, and double-clicking on the entry "1000" in the "Screens" subdirectory calls the Screen Painter. Depending on your own knowledge, you could also demonstrate other programs or how to call other Workbench tools.

## Accessing Source Code

SAP delivers the source code for all ABAP programs. You can view the code and use it, for example, as a template for your own programs.

In any application, you can choose *System* → *Status*, and double-click to navigate to the relevant ABAP Workbench tool. The Workbench displays the selected object in the appropriate tool (given that you have authorization to do this).



Repository Data

| | |
|---|---|
| Transaction | ZTA01 |
| Program (Screen) | ZREPORT |
| Screen Number | 1000 |
| Program (GUI) | RSSYSTDB |
| GUI Status | SHW_TC |

**Figure 43: Navigating into the source code**

A good place to start navigating in the source code would be the initial screen of transaction SE93 (Maintain Transaction), for example. Once you have demonstrated the interface, the pushbuttons and a few selected menu paths, you can use *System* → *Status* to go to the coding behind this transaction (by double-clicking on the field after *Program (Screen)*), to the Screen Painter (by double-clicking on the field after *Screen Number*), or to the Menu Painter (by double-clicking on the field after *GUI Status*).

## Creating ABAP Reports Using the ABAP Editor

You can use the ABAP Editor (transaction SE38) to create and edit programs. ABAP programs are not stored in the SAP system as ASCII files, but as entries in database tables.

When you want to create a new program, you need to enter both a program title and attributes for the program. These attributes include program type (such as "Executable program"), status (for example, "Test program"), and application component. When saving your program, you also need to assign it to a package (formerly called "development class)".

Once you have completed these activities, you can write your program text in the ABAP Editor.

The Editor provides a range of functions, including a syntax check and an option for ABAP keyword capitalization. You can also display syntax help for an ABAP keyword by positioning the cursor on the keyword and pressing F1. SAP recommends that you only develop ABAP programs using the ABAP Editor.

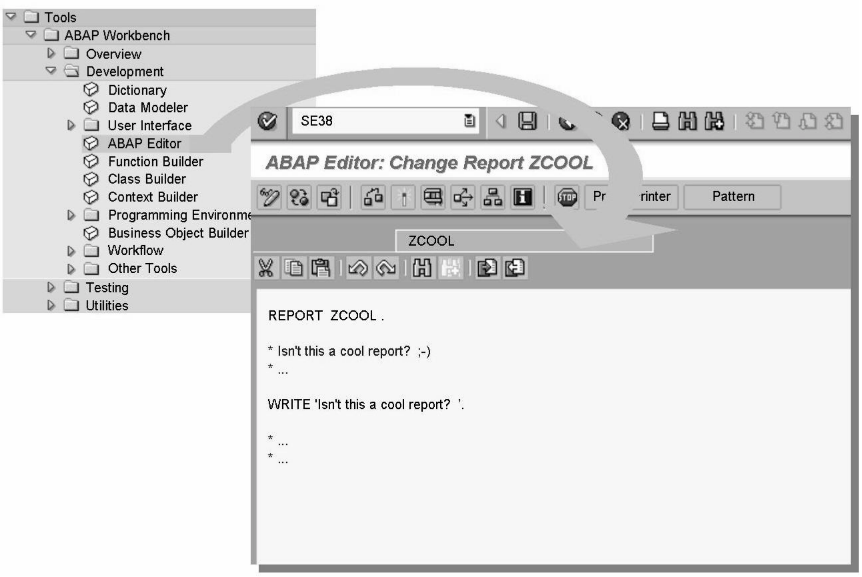
From the Editor, you can navigate to other tools in the development environment such as the ABAP Dictionary, the Screen Painter or the Menu Painter, by double-clicking on Repository objects in the coding.

If you create or change a program (or a development object in general) and then save it, an inactive version is always saved first in the Repository. This makes it possible to continue developing without changing the active system.

To make a Repository object available throughout the system, you then need to "activate" it. This creates an active version of the program which is then used if, for example, a user wants to execute your program.

You can execute your program in the ABAP Editor using *Direct processing (F8)*.

**Hint:** You can find an extensive collection of example programs for testing in transaction ABAPDOCU.



```
REPORT  ZCOOL .

* Isn't this a cool report?  ;-)
* ...

WRITE 'Isn't this a cool report?  '.

* ...
* ...
```

**Figure 44: The ABAP Editor**

Use either the Object Navigator or transaction SE38 itself to create a short ABAP program called “Z<CLNT>DEMO”. <CLNT> in this case is the number of the client you are working in. During the demonstration, explain each step as you go along (entering a title and attributes, selecting a package or saving as a local object, assigning to a transport request if appropriate, and so on). Then write a short ABAP program of your choice (`write: 'Hello World'.` is perfectly adequate, you can however write a more demanding program if you would prefer to). You could also demonstrate the F1 help on the keyword `write` or the Pretty Printer. Save your program. You could also include a minor syntax error in your program (for example, omit the final period in the coding), then correct the error after the syntax check and run the check again. Activate your program and execute it using the pushbutton *Direct processing (F8)*.

# 153 Exercise 12: Accessing and Editing Repository Objects

Exercise Duration: 15 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Write your first ABAP program.

## Business Example

As a developer, you are creating an ABAP program for a development project.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:**

1. The participants must be authorized to run the ABAP Editor and to create programs there.

## Task Writing an ABAP Program

In the ABAP Editor (transaction SE38) or the Object Navigator (transaction SE80), write an ABAP program that displays a line of text.

1. Name your program “Z<CLNT>PROGRAM_##”, where <CLNT> is the client you are logged on to, and ## is your group number.

   Set the following attributes for your program: *Type* = “Executable program”, *Status* = “Test program”, *Application* = “Cross-Application”. Then press *Save*.

2. On the selection screen that appears, either choose *Local object* or enter the name of a package specified by your instructor. If you do not save your program as a “Local object”, you also need to specify a transport request. If you have not yet created your own request, then do so now by choosing *Create* (and entering a meaningful title and yourself as the only team member)

*Continued on next page*

3. Confirming your entries (if you are using transaction SE38) or choosing the program name in the menu tree you have created in the Object Navigator takes you to the ABAP Editor, where you can write your program text (you may need to use *Display* ↔ *Change*). After pressing “Return” after the closing period (.) of the line `REPORT Z<CLNT>PROGRAM_## .` , you are on the next line, where you can write, for example:

```
write: 'This is my first ABAP program!'.
```

4. Then choose *Save*, *Check*, *Activate* and *Direct processing (F8)* in that order. Your ABAP program should now run without errors.

# Solution 12: Accessing and Editing Repository Objects

## Task Writing an ABAP Program

In the ABAP Editor (transaction SE38) or the Object Navigator (transaction SE80), write an ABAP program that displays a line of text.

1. Name your program “Z<CLNT>PROGRAM_##”, where <CLNT> is the client you are logged on to, and ## is your group number.

   Set the following attributes for your program: *Type* = “Executable program”, *Status* = “Test program”, *Application* = “Cross-Application”. Then press *Save*.

   a) For the procedure, see the exercise. You can call the ABAP Editor either using transaction SE38 (*Tools → ABAP Workbench → Development → ABAP Editor*), or from the Object Navigator (transaction SE80) by selecting “Program” from the selection list on the left side of the screen, entering your program name Z<CLNT>PROGRAM_##, choosing *Display*, and creating the program with the attributes specified in the exercise (without TOP include).

2. On the selection screen that appears, either choose *Local object* or enter the name of a package specified by your instructor. If you do not save your program as a “Local object”, you also need to specify a transport request. If you have not yet created your own request, then do so now by choosing *Create* (and entering a meaningful title and yourself as the only team member)

   a) For the procedure, see the exercise.

3. Confirming your entries (if you are using transaction SE38) or choosing the program name in the menu tree you have created in the Object Navigator takes you to the ABAP Editor, where you can write your program text (you may need to use *Display ↔ Change*). After pressing “Return” after the closing period (.) of the line `REPORT Z<CLNT>PROGRAM_##    .` , you are on the next line, where you can write, for example:

   ```
   write: ’This is my first ABAP program!’.
   ```

   a) For the procedure, see the exercise.

4. Then choose *Save*, *Check*, *Activate* and *Direct processing (F8)* in that order. Your ABAP program should now run without errors.

   a) For the procedure, see the exercise. Congratulations on creating your first ABAP program!

## Lesson Summary

You should now be able to:

- Name important tools in the ABAP Workbench
- Access Repository objects
- Write a simple ABAP program

### Related Information

You can find additional information on the ABAP Workbench and example programs in transaction ABAPDOCU, in the SAP online documentation in the areas *mySAP Technology Components →SAP Web Application Server → ABAP Workbench (BC-DWB)* and *mySAP Technology Components → SAP Web Application Server → ABAP Programming and Runtime Environment → ABAP Programming (BC-ABA)*, and in more advanced training courses.

# Lesson: Working with the ABAP Dictionary

156 Lesson Duration: 75 minutes

## Lesson Overview

The ABAP Dictionary is a central component of the ABAP Workbench. It contains both business and technical definitions and descriptions of SAP data. Many ABAP Workbench tools (such as the ABAP processor, the screen processor, and the Screen Painter) constantly access information in the ABAP Dictionary.

This lessons explains the significance of the ABAP Dictionary. You find out about how business process models can be mapped in the ABAP Dictionary. The concept of domains, and the role played by foreign keys in ensuring data consistency are also explained.

## Lesson Objectives

After completing this lesson, you will be able to:

- Explain the significance of the ABAP Dictionary
- Outline the two-level domain concept
- Explain the significance of foreign key dependencies for ensuring data consistency

This lesson explains the significance of the ABAP Dictionary. You need to explain to the participants that it is only structures that are defined in the ABAP Dictionary – the table entries themselves are stored in database tables. You also need to outline the domain concept in SAP systems. You may find it helpful to demonstrate how to create a table and set up a foreign key dependency between this table and one that already exists. If you are going to do this, you need to create a table as part of your preparation for this lesson (for example, you could create a table of football clubs in the national league, if, during the system demonstration, you are going to create a table of their current rankings in the league); you also need to fill this table with data.

**Caution:** Foreign key relationships are only used on maintenance screens that can, for example, be created using the table maintenance generator. (Procedure: Call the desired table in transaction SE11, then choose *Utilities* → *Table maintenance generator*. Enter "&NC&" in the *Authorization group* field, and the table name in the *Function group* field.) The fact that foreign key relationships are only used on (maintenance) screen is also documented in the online help. For more information, see the online documentation for the ABAP Dictionary (in transaction SE11, choose *Help* →

*Application Help*; the system displays the documentation under the path *mySAP Technology Components* → *SAP Web Application Server* → *ABAP Workbench*→ *BC-ABAP Dictionary*), then choose *Tables* → *Table Fields*. Under *Other Assignment Options* → *Check Table*, it says: "This input check appears on all the screens on which the field is used." Up to SAP R/3 4.6C, the foreign key relationships were also used when adding data records using transactions SE11 and SE16. However, as of SAP Web AS 6.10, they are (correctly) only used on the maintenance screen.

## Business Example

While implementing business processes, you need to maintain tables and the relationships between them in the ABAP Dictionary.

## What Is the ABAP Dictionary?

At this point you could display the initial screen of transaction SE11 (ABAP Dictionary: Initial Screen), and, depending on your own knowledge, explain some of the entries (apart from database table, as that is covered later in the lesson).

**Hint:** The *Type Group* entry allows you to store user-defined data types or constants in the ABAP Dictionary so that they can be used by any program. The ABAP Dictionary initial screen for SAP R/3 4.6 did **not** include type groups, although previous releases had done so; in other words, this represents a return to a "previous status".

The ABAP Dictionary enables all data definitions used in the SAP system to be described and managed centrally. It is an integrated and active dictionary, that is, the ABAP Dictionary is completely integrated in the SAP development environment. The Dictionary information is created only once, but is available throughout the system at all times. The ABAP Dictionary automatically provides all the information that has been created or modified, thus ensuring that runtime objects are up-to-date, and that data is consistent and secure.

The tasks of the ABAP Dictionary can be subdivided into:

- Database object definitions (tables, views, and so on)
- Type definitions (structures, table types, and so on)
- Services definitions (F1 help, F4 help, lock objects, and so on)

Tables, views, lock objects, and domains are important object types in the ABAP Dictionary:

- The definition of tables in the ABAP Dictionary is database-independent. This table definition then serves as the basis for the creation of a table with the same structure in the underlying database.
- Views are logical views of one or more tables. View structures are defined in the ABAP Dictionary. This structure is then the basis for the creation of a view on the database.
- Lock objects coordinate attempts by several users to access the same dataset. Function modules are generated from the lock object definition in the ABAP Dictionary; you can then use these function modules in application programs.
- You can use domains to group fields that have similar technical or business purposes. A domain defines the value range for all table fields and structure components that refer to that domain.

The documentation (F1 help) and the input help (F4 help, also called the "possible entries" help) for a field on an input screen are also provided by the ABAP Dictionary.

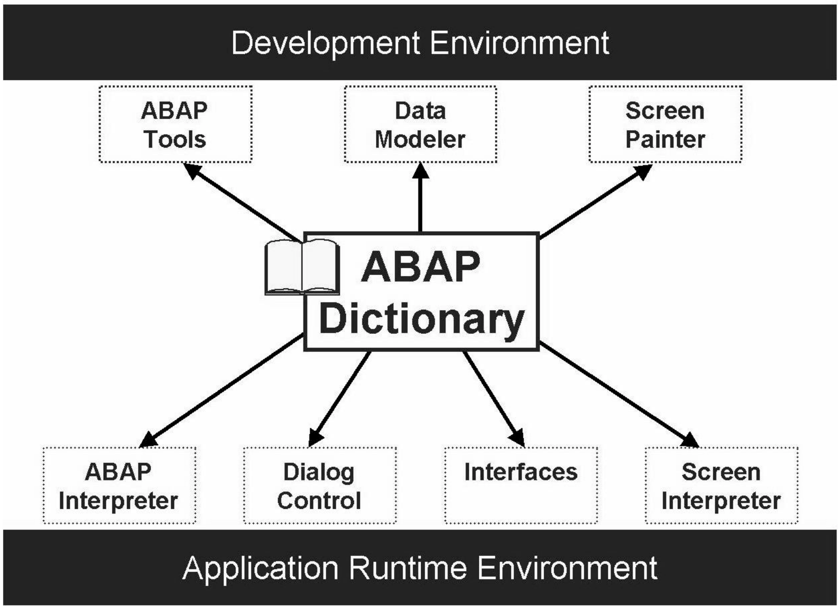


**Figure 45: Significance of the ABAP Dictionary**

The link of the ABAP Dictionary to the program flow is based on the interpretative method of the SAP Web AS runtime environment. Instead of working with the original of an ABAP program, the ABAP processor interprets a runtime object generated from the program text prior to its first execution. Runtime objects are automatically generated again before execution if a time stamp comparison reveals that they are no longer consistent with the current status of the ABAP Dictionary.

The ABAP Dictionary also allows you to manage, in the SAP system, database tables relevant to the SAP system. You do not need detailed, product-specific database knowledge for application development. The ABAP Dictionary converts the definitions at database level.

The interplay between the ABAP Dictionary on one side and the development environment or runtime environment on the other is outlined in the graphic “Significance of the ABAP Dictionary”.

**Hint:** Every database system also contains its own dictionary. This is **not** the dictionary referred to in this lesson.

## Modeling in the ABAP Dictionary

A person or a group can only cope with a limited level of complexity. In this context, you need to reduce the real world to those things that are relevant in a business environment. All “nonessential” aspects are omitted.

Models permit you to reduce the complexity of a system to its essential components. They differ in terms of their purpose and what is “essential” in this context.

The SAP application model documents the business-oriented relationships and processes in SAP applications. The structures of business objects and the business processes they refer to are described in detail for business purposes. Complex operations are shown graphically and are easier to follow. These models are mapped on the database using the ABAP Dictionary.

**Note:** In the ABAP Workbench, you can display data models as text or graphics and modify them in the Data Modeler (transaction SD11). The user can define sections of the data model as required.

The graphic “Modeling in the ABAP Dictionary” shows a section of the “flight data model” used for training purposes at SAP. The relationship between four tables is depicted:

- SCARR: contains the abbreviations for the carriers used in the model
- SPFLI: contains available flight schedules: what flight routes exist between which two points?

- SFLIGHT: contains information on individual flights on known routes, and their flight numbers
- SBOOK: contains all flight bookings, sorted by carrier, route, flight number, and customer number of the person making the booking

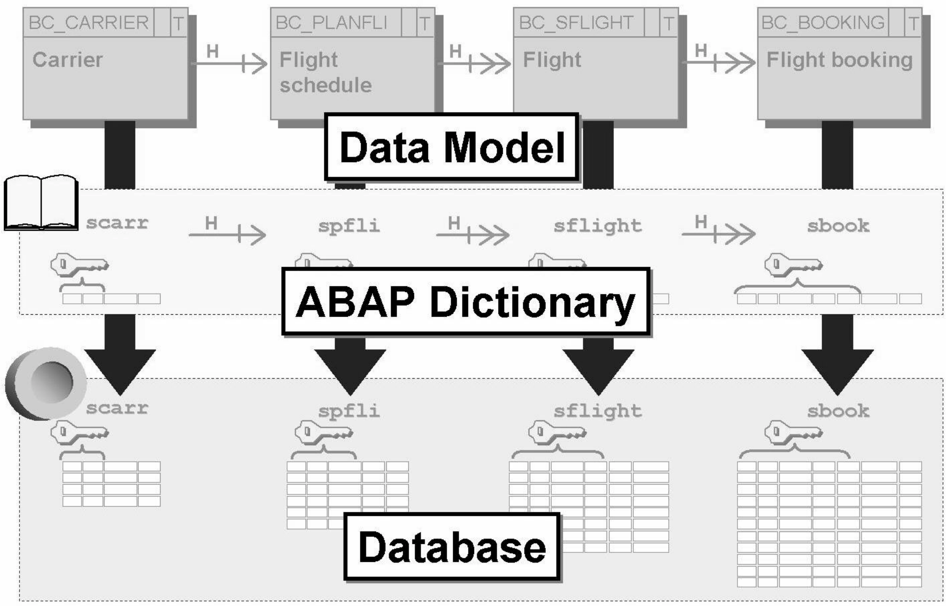


Figure 46: Modeling in the ABAP Dictionary

The upper section of the graphic is an excerpt from the data model. This model shows that the four tables are mutually dependent. For example, you can only make bookings for flights that exist, or entries in a flight schedule if the carrier you enter also exists.

Various tables based on the model are defined in the ABAP Dictionary (and thereby also created on the database). Dependencies between tables are defined in the ABAP Dictionary using foreign key dependencies (see below). The Dictionary only contains information on the table structure, it does not contain the individual entries on flights booked, carriers, and so on. These entries are stored in the relevant database tables.

## Table Definition and the Two-Level Domain Concept

This part of the lesson clarifies the difference between data element and domain.

**Hint:** In transaction SE11, the appearance of the initial screen for editing table definitions has changed by comparison with SAP R/3 4.6C. For example, the tab pages *Delivery and Maintenance* and *Entry help/check* are new. The *Delivery and Maintenance* tab page now contains the *Delivery Class* field, which used to be on the *Attributes* tab page.

You can define tables database-independently in the ABAP Dictionary. When you activate the table, a physical table definition is created on the database on the basis of the table definition stored in the ABAP Dictionary. The table definition in the ABAP Dictionary is converted into a definition for the database used.

A table is a two-dimensional matrix consisting of columns (fields) and rows (entries). It has a name and attributes, such as the table type. Every table in the ABAP Dictionary has a primary key. This is a combination of columns that uniquely identifies every row in the table. Primary key values can therefore not be repeated in a table.

A field (that is, a column in a table) has a name and attributes, for example, it may be a primary key field. A field is not an independent object; it depends on the table and can only be maintained within that table. You can use domains and data elements to define table fields:

- The domain is used to technically define the table field. Field length and type, output attributes and possible values restriction using fixed values, for example, are defined in the domain.
- Data elements are used to describe the semantic attributes of a field in the context of the table. These attributes are only significant within the table, but not generally (as technical attributes are). In the data element, you can, for example, define a short description of the table field that is displayed on the screen when you call the F1 help. You can also specify in the data element the text that is displayed on input fields that refer to the data element (field label, for example, “Destination Airport”).

The two-level domain concept (consisting of the data element level and the domain level) allows technical field attributes to be defined and maintained at the domain level. A domain can pass its field attributes on to any number of fields, and you only need to explicitly change the domain itself, but not the individual fields, when modifying the field attributes thus described. Basing fields on the same domain ensures that field values can be compared safely and without conversion.

Tables, data elements and domains are managed centrally in the ABAP Dictionary.

**Hint:** If you want to check where in the SAP system a particular data definition (data element, domain, table, or similar) is used, then you can look in the "Where-used list" for that data definition.

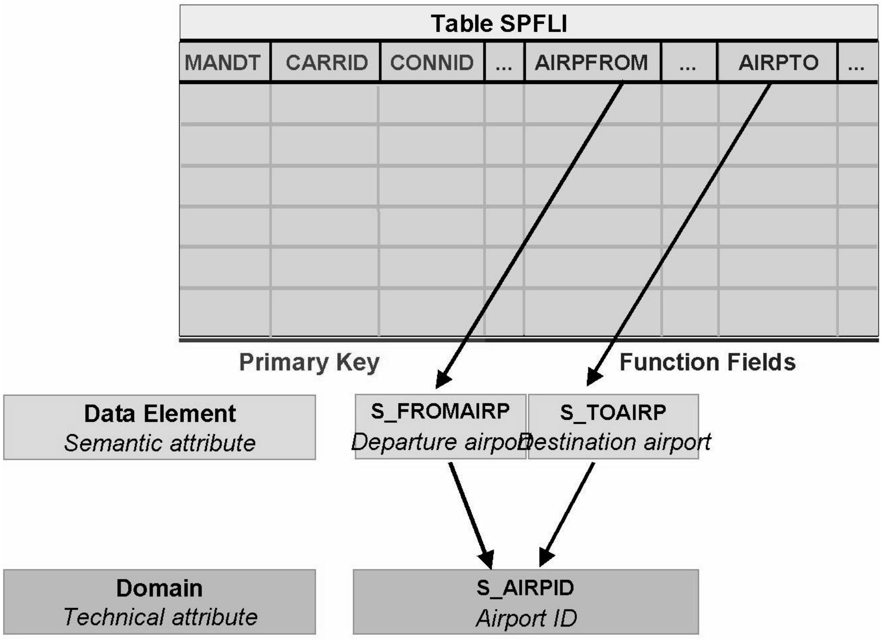


**Figure 47: Table definition and domain concept**

The graphic uses table SPFLI from the flight data model as an example. Flights (for example, Lufthansa flight XY from Frankfurt to Tokyo) are maintained centrally in this table. The table contains fields for the departure airport (*AIRPFROM*) and the destination airport (*AIRPTO*). Because departure and destination airports are different things in a business context, two data elements, *S_FROMAIRP* and *S_TOAIRP*, have been defined. However, because both columns contain the names of airports, both data elements refer to the same domain, *S_AIRPID*, that has the technical type CHAR, with the length 3.

At this point you should show the participants how to create a (transparent) table step by step. If you already created a check table as part of your preparation (see the instructor note at the beginning of this lesson), then you could now create a table whose entries can later be checked against the check table. During your demonstration, you do not need to describe each step in detail. It is simply intended to give the participants a feel for the steps involved in creating a table. If you like, you could, when defining the table, use data elements that do not yet exist in the system and that you therefore need to create. You should also explain what happens when you activate the table. You could also, if appropriate, run a where-used list from a domain.

162

## Creating a Table in the ABAP Dictionary

1. On the initial screen of the ABAP Dictionary (transaction SE11), select the object class “Database table”, enter a table name, and choose *Create*.
2. Enter an explanatory short text in the *Short text* field. You can later use this short text in the Information System to search for your table.
3. Enter the delivery class for the table on the *Delivery and Maintenance* tab page. On the same tab page, select an entry from the list field for the *Data Browser/Table View Maint.* field. For example, select “Display/Maintenance Allowed”, to enable users with the correct authorizations to use the Data Browser (transaction SE16) to change the table data.
4. Specify the table fields on the *Fields* tab page. To do this, carry out the following steps for each field:

    1. Enter a name for the table field in the *Field* column. The field name may only contain alphanumeric characters and must start with a letter.

    2. If the field belongs to the table’s primary key, select the *Key* column.

    3. Enter the name of a data element in the *Data element* field. The field will now take its “data type, length, decimal places”, and “short text” from this data element. If no suitable data element exists, you can access data element maintenance by entering a name and selecting it.

    4. You can use the *Built-in type* pushbutton to specify the data type, field length, decimal places, and short text yourself. Choose this button again if you want to switch back to entering data elements for other fields.

        **Note:** You cannot define foreign keys or specify fixed values for fields without data elements (that is, fields whose type you have defined directly); there is no F1 help for this kind of field.

        If you have fields of the type “CURR” (currency amounts) or “QUAN” (quantities), you need to enter reference fields and reference tables on the “Currency/Quantity Fields” tab page.

5. Maintain the technical settings for the table. You can display the relevant maintenance screen using *Technical settings*.

*Continued on next page*

6. Maintain (if necessary) the table's foreign key relationships to other tables. You can display the relevant maintenance screen by placing the cursor on the field you want to check and choosing *Foreign Keys*. The significance of foreign keys is explained in more detail in the next section.
7. Save the table. A dialog box appears, in which you must assign the table to a package (formerly: development class).

   **Note:** For tables in delivery classes G and E, you also need to maintain the customer namespace (table key area) for the table entries. SAP cannot change table entries in the customer namespace during an upgrade, or import new entries into this namespace. To get from the *Delivery and Maintenance* tab page to the relevant maintenance transaction, choose *Maintain Customer Namespace*.

8. Choose *Activate*.

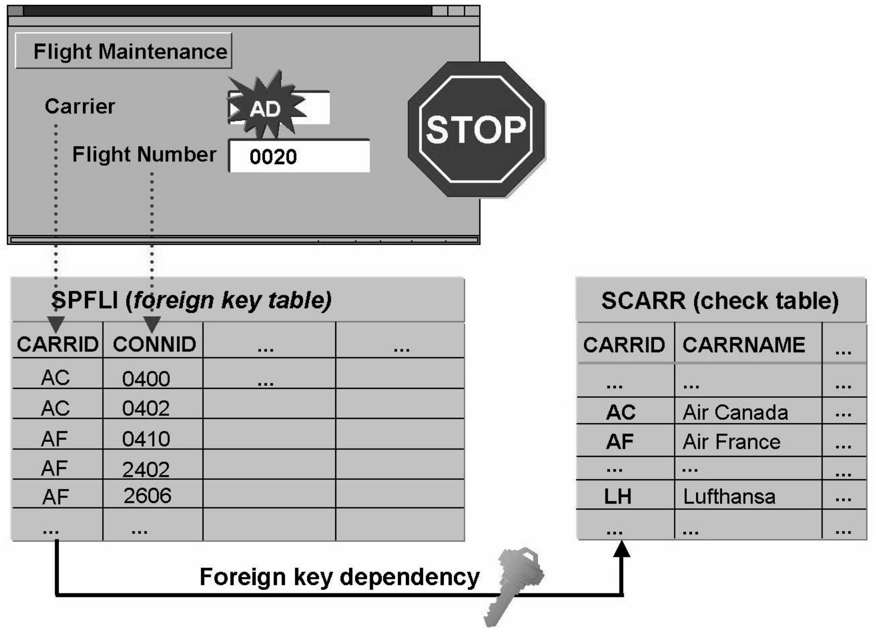
## Data Consistency Using Foreign Keys

The ABAP Dictionary enables you to define relationships between tables. These relationships are called foreign keys and must be defined explicitly at field level. In technical terms, a combination of table fields can only be a foreign key if this same combination of fields is the primary key for another table.

Foreign keys are primarily used to ensure data consistency. If new data is entered in a table, then you can use foreign keys to check whether the new data record is consistent with the data already in the system. The table in which the new data is entered is known as the foreign key table. The consistency is checked against the check tables.

**Note:** There are various technical prerequisites for creating foreign key dependencies between tables. These prerequisites significantly exceed the scope of this lesson and are not covered here.

The graphic below shows a new flight being entered for carrier "AD", using a dialog transaction. Flights are stored in the table SPFLI . Various foreign key dependencies to other tables have already been defined in this table. In this example, the foreign key dependency checks whether the carrier entered has been defined in the central carrier table SCARR. Carrier "AD" does not exist, so you cannot define any flights for this carrier. The entry is automatically rejected on the screen.



Flight Maintenance

Carrier: AD

Flight Number: 0020

SPFLI (*foreign key table*)

| CARRID | CONNID | ... | ... |
|---|---|---|---|
| AC | 0400 | ... | |
| AC | 0402 | | |
| AF | 0410 | | |
| AF | 2402 | | |
| AF | 2606 | | |
| ... | ... | | |

SCARR (check table)

| CARRID | CARRNAME | ... |
|---|---|---|
| ... | ... | ... |
| **AC** | Air Canada | ... |
| **AF** | Air France | ... |
| ... | ... | ... |
| **LH** | Lufthansa | ... |
| ... | ... | ... |

**Figure 48: Data consistency using foreign key dependencies**

**Hint:** When you create a foreign key dependency in the ABAP Dictionary, you also get input help (F4 help) for screen fields assigned to this table field.

You could at this point create a foreign key dependency between the table created for the participants and the check table you created as part of your preparation. You can then fill the table with data. You should draw the participants' attention to the F4 help created through the foreign key dependency.

# 165 Exercise 13: Working with the ABAP Dictionary

Exercise Duration: 20 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Learn to use the ABAP Dictionary

## Business Example

You need to create a new table in your SAP system and fill it with data. You would also like to display the table content.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:**

1. The participants must be authorized to run transactions SE11(ABAP Dictionary) and SE16 (Data Browser), and to create their own table.

## Task 1

Displaying table definitions

1. Find out which fields make up the primary key of table TADIR. To do this, call transaction SE11 (ABAP Dictionary, *Tools* → *ABAP Workbench* → *Development* → *ABAP Dictionary*), choose the *Database table* radio button, enter the table name, and choose *Display*.
2. What is the name of the field that has the short text *Package*?
3. To which data element and to which domain is this field assigned?
4. In which data elements is the domain *DEVCLASS* used? (Hint: select the domain name and then run a where-used list.)

## Task 2

Displaying table content

*Continued on next page*

1. Find out how many programs are in your SAP system Repository. To do this, call transaction SE16 (Data Browser, *Tools → ABAP Workbench → Overview → Data Browser*), enter the table name TADIR on the screen and choose *Enter*. To display the number of programs, on the following selection screen, enter the value `PROG` (for programs) in the *OBJECT* field and choose *Number of Entries*.
2. Find out which of these programs begin with "RSP". To do this, further limit your search on the selection screen by entering the value "RSP*" in the *OBJ_NAME* field and choosing *Execute*.

## Task 3

Create a table and fill it with data!

1. In the ABAP Dictionary (transaction SE11), create a database table of your customers with the name "Z<CLNT>CUSTOMER_##", where <CLNT> is your client and ## is your group number. Enter a meaningful short text. On the *Delivery and Maintenance* tab page, choose delivery class "A" and "Display/Maintenance Allowed". On the *Fields* tab page, define two key fields: *CLIENT* using the data element *MANDT* and *CUSTNUMBER* using the data element *KUNNR*. Define an additional field, *FIRST_CONTACT* using the data element *DATUM*. Save your table as a local object or enter a package name specified by your instructor. If you do not save your table as a local object, you also need to specify a transport request. If you have not yet created your own transport request, choose *Create*, thereby creating your own request (with an obvious title). Otherwise, select an existing transport request under *Own requests* and confirm your entry. Under *Technical Settings*, choose the *USER* data class and the size category"0", do not allow buffering for the table. Save your table again and activate it.
2. Use *Utilities → Table contents → Create entries* to enter your own data in the table and then use *Utilities → Table contents → Display* to display these entries.

# Solution 13: Working with the ABAP Dictionary

## Task 1

Displaying table definitions

1. Find out which fields make up the primary key of table TADIR. To do this, call transaction SE11 (ABAP Dictionary, *Tools → ABAP Workbench → Development → ABAP Dictionary*), choose the *Database table* radio button, enter the table name, and choose *Display*.

   a) For the procedure, see the exercise; the primary key consists of the fields *PGMID*, *OBJECT* and *OBJ_NAME*.

2. What is the name of the field that has the short text *Package*?

   a) You can use *Find* on the standard toolbar to search for the name *Package*. The field name for package is *DEVCLASS*.

3. To which data element and to which domain is this field assigned?

   a) The *DEVCLASS* field is assigned to the data element *DEVCLASS*. Double-clicking on the name of the data element displays the definition of the data element *DEVCLASS*. You can see that this data element refers to the domain *DEVCLASS*. Although the names of the data element and the domain are identical in this case, data elements and domains serve different purposes in the SAP system.

4. In which data elements is the domain *DEVCLASS* used? (Hint: select the domain name and then run a where-used list.)

   a) In the definition of the data element *DEVCLASS* from the previous exercise, choose the *DEVCLASS* domain name to display the definition of the domain. Here you can use the *Where-used list* pushbutton to run a where-used list in which you are only looking for uses in data elements. The result is a list of data elements that refer to the *DEVCLASS* domain.

## Task 2

Displaying table content

1. Find out how many programs are in your SAP system Repository. To do this, call transaction SE16 (Data Browser, *Tools → ABAP Workbench → Overview → Data Browser*), enter the table name TADIR on the

*Continued on next page*

screen and choose *Enter*. To display the number of programs, on the following selection screen, enter the value `PROG` (for programs) in the *OBJECT* field and choose *Number of Entries*.

a) For the procedure, see the exercise.

2. Find out which of these programs begin with "RSP". To do this, further limit your search on the selection screen by entering the value "RSP*" in the *OBJ_NAME* field and choosing *Execute*.

    a) For the procedure, see the exercise.

## Task 3

Create a table and fill it with data!

1. In the ABAP Dictionary (transaction SE11), create a database table of your customers with the name "Z<CLNT>CUSTOMER_##", where <CLNT> is your client and ## is your group number. Enter a meaningful short text. On the *Delivery and Maintenance* tab page, choose delivery class "A" and "Display/Maintenance Allowed". On the *Fields* tab page, define two key fields: *CLIENT* using the data element *MANDT* and *CUSTNUMBER* using the data element *KUNNR*. Define an additional field, *FIRST_CONTACT* using the data element *DATUM*. Save your table as a local object or enter a package name specified by your instructor. If you do not save your table as a local object, you also need to specify a transport request. If you have not yet created your own transport request, choose *Create*, thereby creating your own request (with an obvious title). Otherwise, select an existing transport request under *Own requests* and confirm your entry. Under *Technical Settings*, choose the *USER* data class and the size category"0", do not allow buffering for the table. Save your table again and activate it.

    a) For the procedure, see the exercise (compare also the procedure described in the lesson).

2. Use *Utilities → Table contents → Create entries* to enter your own data in the table and then use *Utilities → Table contents → Display* to display these entries.

    a) For the procedure, see the exercise.

## Lesson Summary

You should now be able to:

- Explain the significance of the ABAP Dictionary
- Outline the two-level domain concept
- Explain the significance of foreign key dependencies for ensuring data consistency

### Related Information

You can find additional information on the ABAP Dictionary in more advanced training courses and in the SAP Library under *SAP Web Application Server* → *ABAP Workbench* → *ABAP Dictionary* and in the training course **BC430** - *ABAP Dictionary*.

# Lesson: Transporting and Verifying Changes

170 Lesson Duration: 25 minutes

## Lesson Overview

This lesson covers the final steps in the process of exporting objects from the source system and importing them into other systems in the system landscape. The entire software development cycle is discussed.

## Lesson Objectives

After completing this lesson, you will be able to:

- Describe the entire development cycle for an object

This lesson covers the steps involved in finalizing a development project. This includes releasing both the tasks and the change request as a whole in the Transport Organizer. The entire software development cycle is discussed.

## Business Example

You need to release changes to your own objects and to SAP objects and to import them into other SAP systems.

## Actions at Close of Development

Once the development phase is finished in the development system, every developer releases his or her tasks. If developers cannot release their tasks themselves, then the development leader can do this for them. Once the development project is complete, the development leader releases the change request as a whole.

Change requests may be transportable or local. The Transport Organizer classifies the request automatically on the basis of the objects contained in the change request. Data export from the source system is only triggered if the change request is transportable.

Once a request is released, the Repository objects are copied from the source database to a directory at operating system level. The “Release” and “Export” steps are noted in the transport log for the change request. You can display this log even after the transport has taken place.

Import into the target system is not normally automatic, but is triggered by the transport administrator in the Transport Management System (TMS). The Repository objects are then copied from the central transport directory at operating system level into the target system database. The import logs can then be checked.

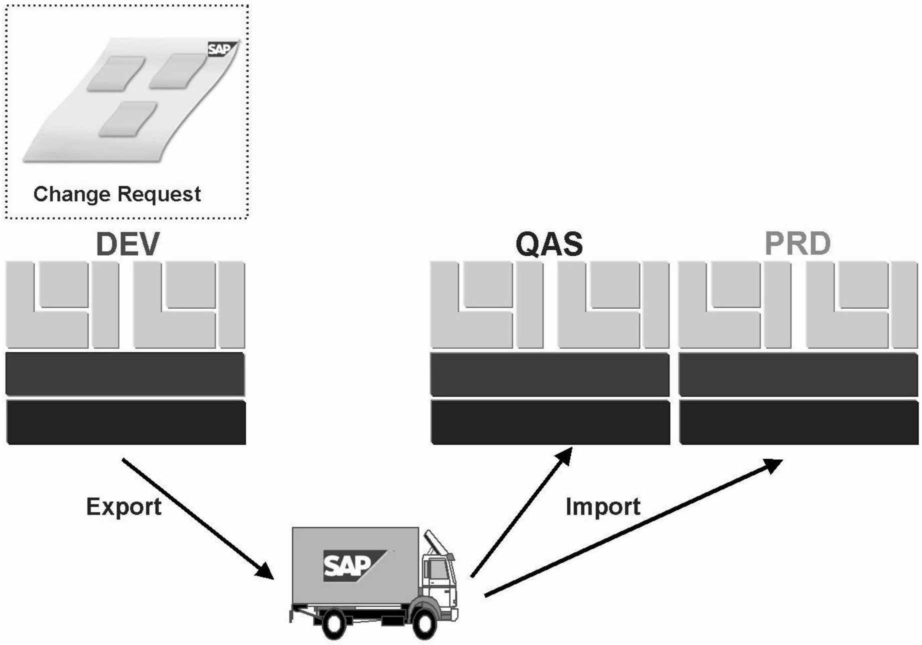


Figure 49: Export and import into other systems

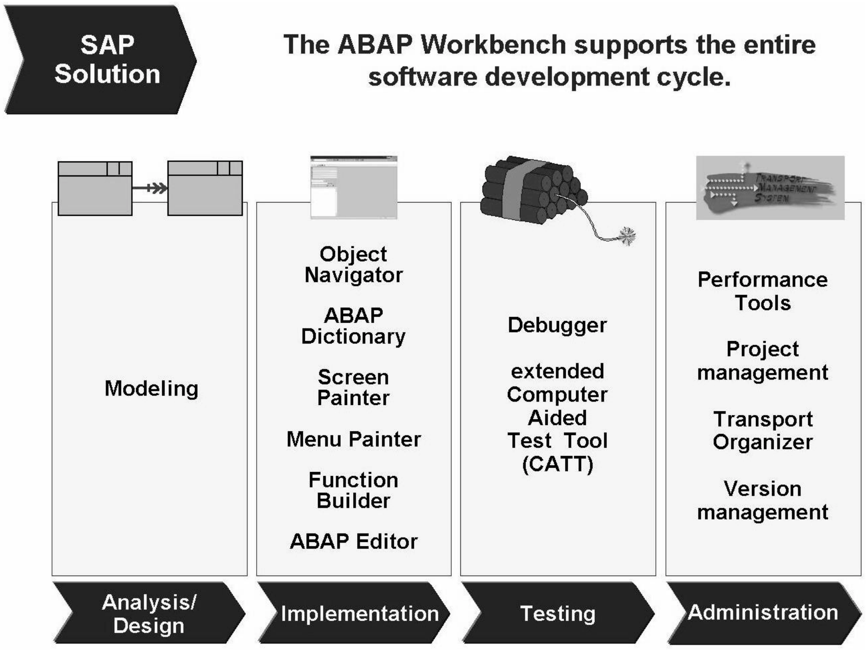
At this point you could show the following system demonstration:

Call the Transport Organizer using *Tools* → *ABAP Workbench* → *Overview* → *Transport Organizer* or SE09 or SE10, point out the transport request and expand it down to task level. Then release the task and the request using *Release directly*. Maintain the documentation for this request in the next window. You can display the various processing steps for the release and the export in the status bar.

If you are familiar with the Transport Management Systems (and the system landscape is correctly configured), you can also show the import queue for the next system (such as CL5 or QAS) in transaction STMS (Transport Management System), and if appropriate, even the import into the next system.

## The Software Development Cycle

You can map and carry out the entire software development cycle in the SAP system.



SAP Solution

The ABAP Workbench supports the entire software development cycle.

**Figure 50: The software development cycle**

SAP's ABAP Development Workbench is a programming environment for developing enterprise wide client/server solutions for business purposes. SAP supports the entire software development cycle with tools for modeling, programming using ABAP, defining data and table structures, and for designing graphical user interfaces. An extensive array of tools for testing, tuning, and maintaining software, as well as supporting development in large teams, are also available.

In the concept phase of a project, you enter the results of your analyses into the SAP data model. Then the models are converted into tables and programs. Next you develop the individual program components, such as user interface, ABAP code, and so on, in a sequence of your choice in the separate development tools designed for each functionality. You only need combine the different elements into a single application when you want to run it. The development cycle concludes with program tests and transport into the production system.

As a supplement to the development tools, SAP also provides a library of business and utility software components that you can easily incorporate into your own programs.

# 173 Exercise 14: Releasing a Transport Request

Exercise Duration: 5 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Finalize a development project

## Business Example

You need to finalize a development project.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task Releasing a Request

Release a change request and the tasks assigned to it.

1. Release your change request (including the tasks assigned to it) using the Transport Organizer.

# Solution 14: Releasing a Transport Request

## Task Releasing a Request

Release a change request and the tasks assigned to it.

1. Release your change request (including the tasks assigned to it) using the Transport Organizer.

   a) Call the Transport Organizer using *Tools* → *ABAP Workbench* → *Overview* → *Transport Organizer* or transaction code SE09 (or SE10):

      On the initial screen of the Transport Organizer, select your change request, which has status "Modifiable" and release it in the next window. To do this, first select the task and then choose *Release directly*.

      Then select your change request and choose *Release directly*. Before the request can be released, you need to document your work in a separate window.

## Lesson Summary

You should now be able to:

- Describe the entire development cycle for an object

## Unit Summary

You should now be able to:

- Describe the data structure of SAP systems
- Name the various data types
- Describe the three-system landscape as recommended by SAP
- Explain the importance of transports
- Describe the procedure for transports
- Name important tools in the ABAP Workbench
- Access Repository objects
- Write a simple ABAP program
- Explain the significance of the ABAP Dictionary
- Outline the two-level domain concept
- Explain the significance of foreign key dependencies for ensuring data consistency
- Describe the entire development cycle for an object

177

# Test Your Knowledge

1. What data is contained in a client?

   *Choose the correct answer(s).*

   □ A Cross-client Customizing, user data, and application data
   □ B Customizing and Repository objects
   □ C Client-specific Customizing, user data, and application data
   □ D Only application data entered in this client

2. The data structure of an SAP system consists of:

   *Choose the correct answer(s).*

   □ A Application data and Repository
   □ B Client-specific data, cross-client Customizing, and the Repository
   □ C Client-specific data and the Repository
   □ D Client-specific data and cross-client Customizing

3. Why does SAP recommend a three-system landscape?

   *Choose the correct answer(s).*

   □ A Because of the special data structure in SAP systems
   □ B To enable preparation and testing of upgrades
   □ C To sell more licenses
   □ D Because you can only create one client per system

4. Every developer has to create and release his or her own change requests.

   *Determine whether this statement is true or false.*

   □ True
   □ False

5. What is the ABAP Workbench?

   *Choose the correct answer(s).*

   □ A A performance assessment tool for ABAP programs
   □ B A sort of to do list for ABAP developers; contains all tasks still to be processed
   □ C An integrated graphical development environment for ABAP programs
   □ D A directory of all valid ABAP commands

6. How can you access an SAP transaction's underlying ABAP code (assuming that you have the authorization to do so)?

   *Choose the correct answer(s).*

   □ A *System → Status* , and double click "Program (Screen)" under "Repository Data"

   □ B By using *Program → Display source code*, available in every SAP transaction

   □ C By calling the ABAP Editor and entering the corresponding ABAP program name

   □ D By pressing F2 when you are in a transaction

7. What do you use domains for?

   *Choose the correct answer(s).*

   □ A For defining the technical attributes of a field

   □ B For adding foreign key dependencies

   □ C For defining the semantic attributes of a field

   □ D For entering data in a table

8. What is a foreign key ?

   *Choose the correct answer(s).*

   □ A The definition of technical attributes of fields in a table

   □ B A key that developers need to be able to edit Repository objects

   □ C The definition of relationships between tables to ensure their consistency

   □ D The key required for installing an SAP system on a host computer on which a non-SAP system is already installed

9. When a development project is complete, the tasks and change requests must be released so that they can be exported. Who releases which objects?

   *Choose the correct answer(s).*

   □ A Developers release their tasks

   □ B The development leader releases the request as a whole

   □ C The request is released automatically

10. The developer carries out the import into the target systems.

    *Determine whether this statement is true or false.*

    □ True

    □ False

11. You can carry out the entire software development cycle in the SAP system using the ABAP Workbench tools.

    *Determine whether this statement is true or false.*

    □ True

    □ False

180

# Answers

1. What data is contained in a client?

   **Answer:** C

   A client contains client-specific Customizing, user data, and application data.

2. The data structure of an SAP system consists of:

   **Answer:** B

   An SAP system consists of clients, cross-client Customizing, and the Repository.

3. Why does SAP recommend a three-system landscape?

   **Answer:** A, B

   Recommending the three-system landscape does not sell more licenses, since you can set up several SAP systems on one license, although you can only use one of these systems as the production system. You can always create several clients in an SAP system.

4. Every developer has to create and release his or her own change requests.

   **Answer:** False

   The development leader creates and releases change requests.

5. What is the ABAP Workbench?

   **Answer:** C

   The ABAP Workbench is the SAP system's integrated graphical development environment. It supports, among other things, the development and administration of applications written in ABAP, SAP's programming language. The Workbench also contains transactions for assessing the performance of ABAP commands and for managing development projects. These are, however, simply components of the Workbench.

6. How can you access an SAP transaction's underlying ABAP code (assuming that you have the authorization to do so)?

   **Answer:** A, C

   You can access the source code using *System* → *Status* and choosing *Program (Screen)*, or using the ABAP Editor. The menu path *System* → *Status* is available on every screen.

7. What do you use domains for?

   **Answer:** A

   You use domains for defining the technical attributes of fields in a table. The semantic attributes of a field are defined in the data element.

8. What is a foreign key ?

   **Answer:** C

   A foreign key defines a relationship between two tables. A foreign key assigns the fields of one table (known as the foreign key table) to the primary key fields of another table (known as the check table). The technical attributes of table fields are defined in a domain. You need an object key to edit Repository objects.

9. When a development project is complete, the tasks and change requests must be released so that they can be exported. Who releases which objects?

   **Answer:** A, B

   Every developer releases his or her own subtasks. Then the development leader releases the request as a whole. Requests cannot be released automatically.

10. The developer carries out the import into the target systems.

    **Answer:** False

    The import into the target system is started by the transport administrator in the Transport Management System (TMS).

11. You can carry out the entire software development cycle in the SAP system using the ABAP Workbench tools.

    **Answer:** True

    Transactions, tools, and interfaces for all phases of the software development process are available in the SAP system.

# Unit 5

183

## Communication and Integration Technologies

The participants are confronted with a very large number of new terms in this unit, and it is your job as instructor to sort these terms and, above all, to prioritize them. RFC is still the primary technology used for SAP system connections, although new options for linking systems are increasing in significance. XML data exchange between companies by means of HTTP, cross-system Workflow scenarios, and custom-built Web applications are among the topics that are attracting ever more attention from customers.

### Unit Overview

There is a vast array of methods for connecting SAP systems with other systems, and optimizing processes within a system. Many of the available technologies are briefly introduced in this unit, and you should get an idea of the various uses for each.

### Unit Objectives

After completing this unit, you will be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept
- Name various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs
- Describe the SAP Business Workflow concept
- Explain the flow of a workflow process
- Describe additional application areas for the SAP Business Workflow concept

- Describe the areas in which the SAP Internet Transaction Server (SAP ITS), the SAP Web Application Server (SAP Web AS), and the SAP Business Connector (SAP BC) are used
- Describe how the SAP ITS works
- Describe how the SAP Web AS works
- Describe how the SAP BC works
- Name various data transfer procedures

## Unit Contents

# Lesson: Cross-System Business Processes

185 Lesson Duration: 20 minutes

## Lesson Overview

This lesson explains the fundamentals of the Application Link Enabling (ALE) concept.

## Lesson Objectives

After completing this lesson, you will be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept

This lesson aims to give the participants an understanding of the significance of cross-system business processes and the need to link them using ALE.

## Business Example

A company wants to implement an Internet sales scenario in the context of the mySAP CRM solution.

## The Significance of Cross-System Business Processes

Let's start by defining cross-system business processes, using common situations as examples.

For example, it may be the case that within a company, the human resources system is separate from the rest of the business software system. Obviously, the systems cannot be completely separate, since the accounting system needs the employees' wage data. In this situation, you need cross-system business processes to exchange the relevant data.

Cross-system business processes are used, for example, if two companies collaborate closely and send joint orders to a vendor. The companies' business IT systems need to communicate with each other to consolidate the quantities to be ordered. In this case, the business process does not just cross system boundaries, but also company boundaries.

An additional example is the transfer of a limited quantity of specific data, for example, the electronic transfer of account statement data from a bank to a company.

Recent developments suggest that the significance of cross-system business processes will continue to increase rapidly.

## Application Link Enabling (ALE)

Application Link Enabling is a means of creating and operating distributed applications. The basic concept of Application Link Enabling is to ensure operation of a distributed, yet integrated system landscape. This involves business-controlled message exchange using consistent data across loosely linked application systems. The applications are integrated through synchronous and asynchronous communication, not through a central database.

Systems that use ALE to exchange data can be located at the same company, or they may belong to different companies. One of the characteristics of ALE is that different systems are linked in business terms through secure and consistent data transfer.

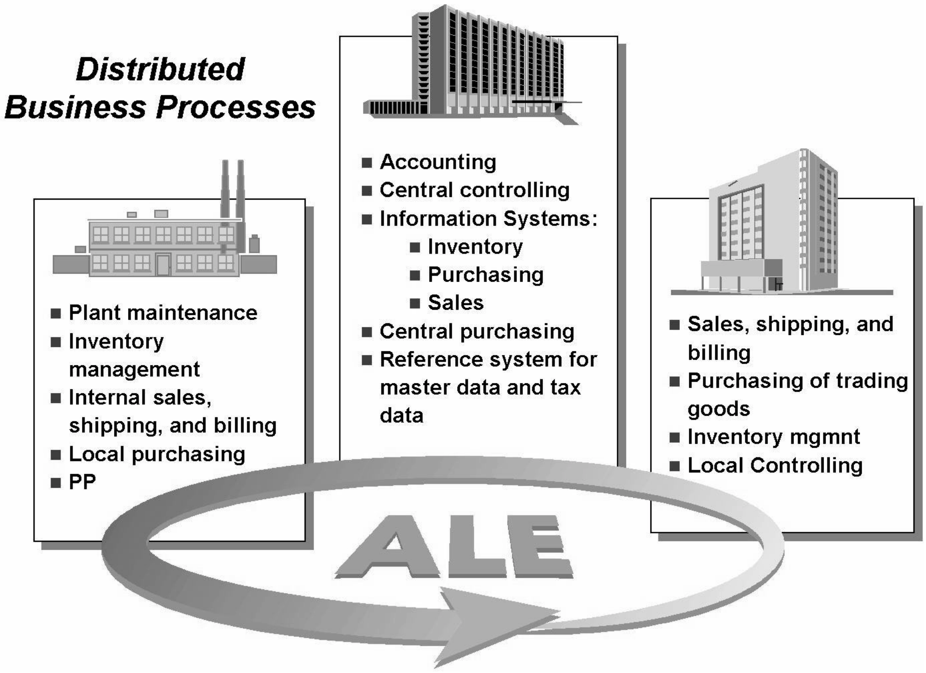


**Figure 51: Business process distribution using ALE**

You could also describe ALE as being composed of the elements: who exchanges which data when, with whom, and by what means.

Implementing ALE therefore requires that you clarify the following points in detail:

1. Identify the business process and the objects involved

2. Identify the information to be transmitted

3. Specify the format for the data to be transferred

4. Decide on the transfer technology to be used
5. Decide on the transfer type
6. Specify the destination of the data transfer

The following table contains examples for implementing ALE:

| Process | Internet Sales with mySAP CRM |
|---|---|
| Identify the information to be transmitted | Order data from the SAP CRM System, which is to be passed to an ERP backend |
| Format of the data | IDoc format |
| Transfer technology | by RFC |
| Transfer type | asynchronously, every 60 seconds |
| Objective | Provide goods and/or services for sale in the Internet |

The data is often identified within the SAP system using a business object (BO) and its Business Application Programming Interfaces (BAPIs). A BAPI is a method of a business object, for example, the material master record. A permissible method could be creating or changing the material master data. BAPIs normally enable you to edit all data belonging to the object.

The IDoc format describes the structure of “intermediate documents”. There are various kinds of IDoc formats for different types of data to be exchanged. Alternatively, you can use ALE to transfer data in an agreed XML format.

You can select your preferred data transfer technology within the constraints imposed by the system. For example, you can transfer data by Remote Function Call (RFC) or using HTTP or HTTPS.

There are two basic transfer types: synchronous and asynchronous. Synchronous transfer means that the data is transferred at the time of creation or change. You can start asynchronous transfers at intervals of your choice.

There are very few restrictions on the systems that can be linked. The systems involved must have the technical capability to receive the communications (RFC-enabled, HTTP-enabled) and interpret the format transferred (IDoc, XML). SAP systems of different releases can be linked using ALE.

## Facilitated Discussion

The aim of this discussion is to make customers aware of cross-system business processes in their own companies.

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

Which cross-system business processes do you (the participants) use?

## Lesson Summary

You should now be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept

### Related Information

- More advanced training courses on ALE are **BIT300**, **BIT320** and **BIT350**.
- You can find additional information on the topic of ALE on the SAP Service Marketplace using the alias */ibf* (and then under *Integration Scenarios*).

# Lesson: Remote Function Calls and BAPIs

189 Lesson Duration: 45 minutes

## Lesson Overview

This lesson provides an overview of the interface technologies available to you in the SAP system, while focusing on the significance of RFCs and BAPIs.

## Lesson Objectives

After completing this lesson, you will be able to:

- Name various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs

This lesson gives the participants an overview of the various interface technologies available. You need a thorough understanding of RFCs (that is, of transactions SE37 and SM59), and of the BOR and BAPIs.

## Business Example

You need to integrate existing applications with SAP applications. The interfaces available in the standard system are of particular interest here.

## Overview of Interfaces

SAP systems have interfaces at different communication levels. These 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 ALE environment. All “higher” interfaces, that is, those that access business objects or processes, use the same technology, the Remote Function Call (RFC).

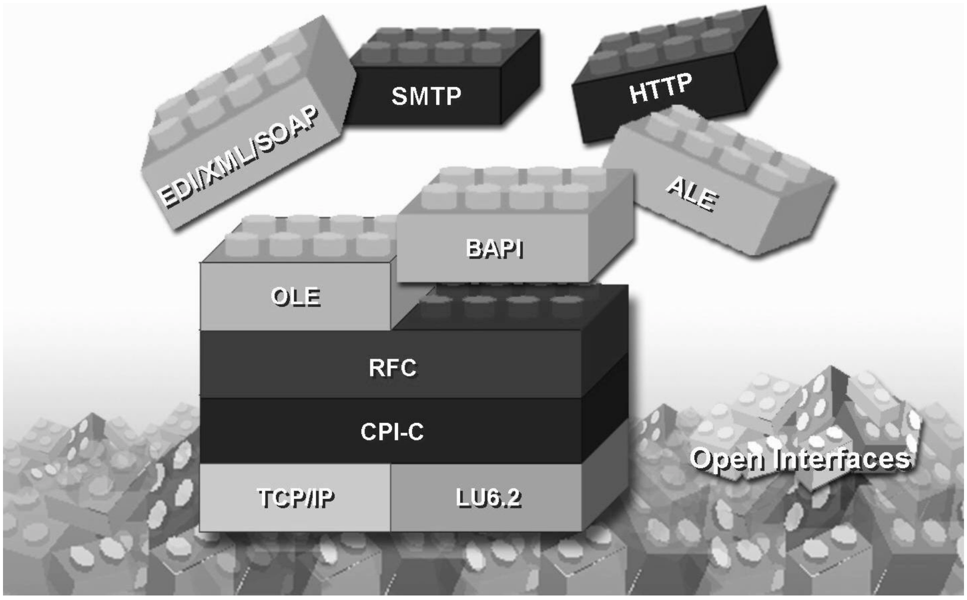


**Figure 52: Interfaces technologies used in SAP systems**

The following list identifies the technologies named in the above graphic.

- TCP/IP: Transmission Control Protocol/Internet Protocol
- CPI-C: Common Program Interface Communication
- LU 6.2: Logical Unit Type 6.2
- HTTP: HyperText Transfer Protocol
- SMTP: Simple Mail Transfer Protocol
- RFC: Remote Function Call
- OLE: Object Linking and Embedding
- ALE: Application Link Enabling
- EDI: Electronic Data Interchange
- BAPI: Business Application Programming Interface
- XML: Extensible Markup Language
- SOAP: Simple Object Access Protocol

## Remote Function Call

The Remote Function Call interface is an SAP interface protocol based on CPI-C and TCP/IP. It simplifies the programming of communication processes between different systems. **RFCs enable you to call and execute predefined functions in a remote system – or within the same system.** RFCs manage the communication process, parameter transfer and error handling.

RFC describes an interface, not the programming language in which the function runs. You can also use RFCs to call functions in non-SAP systems. The procedure for RFC communication between two SAP systems is that the calling system uses an RFC definition in the system called to access a specific function.

This function is normally a remote-enabled function module. You can also, depending on the release, use RFC to call functions in SAP R/2 systems.

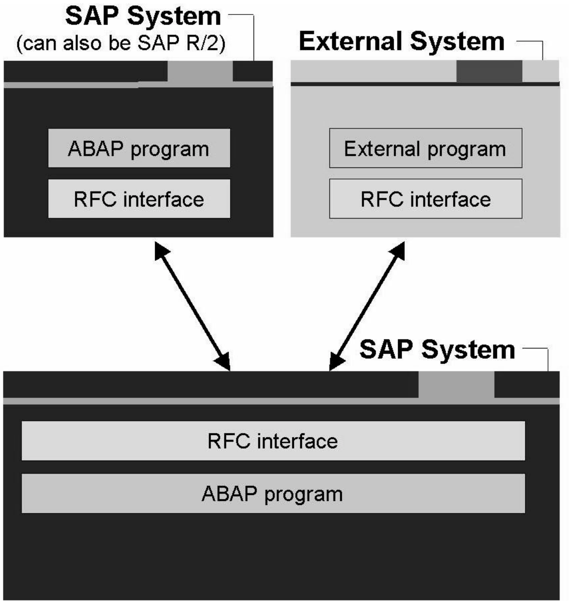


**Figure 53: RFC connection possibilities**

If you want to start external programs remotely, you need an RFC interface outside the SAP system. This could be, for example, a simple Dynamic Link Library (DLL). Every RFC interface is bidirectional, so external programs can also use RFC to access functions in SAP systems.

**Note:** All function modules (including those that are remote-enabled) are created, together with their import and export parameters, using the Function Builder. You can call the Function Builder using *Tools* → *ABAP Workbench* → *Development* → *Function Builder* or using transaction code SE37.

To call an RFC module from an SAP system, you need to know the import and export parameters (defined in the Function Builder), and there must be a technical connection between the two systems. This connection is called an “RFC connection” or an “RFC destination”.

You can manage your RFC connections using *Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations* or using transaction SM59.

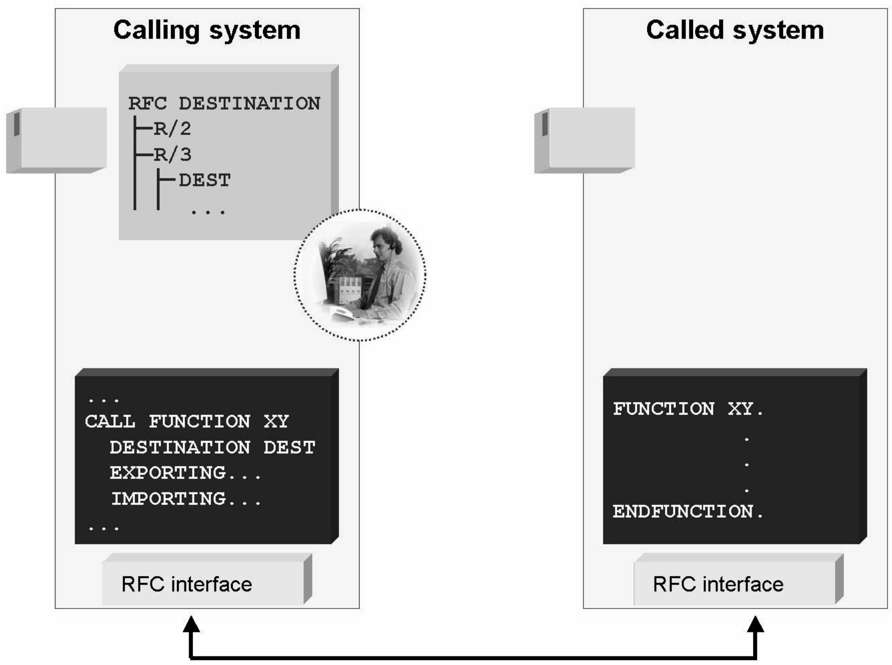


**Figure 54: Remote Function Call in detail**

In the above graphic you can see, on the left side, the calling system, in which an RFC destination named "DEST" has been created. An RFC destination in transaction SM59 should not be confused with an "SAP system", since an RFC connection can only point to one specific client in an SAP system. These are therefore also referred to as connections between "logical systems"; this term is used, above all, in the ALE environment.

This also means that you can have at least as many RFC connections between two systems as there are clients in the target system. Since you can specify a logon user for the destination in each RFC connection, you can therefore also access clients in the target system several times, for example, with a different logon user each time. If you need a bidirectional RFC connection between two systems, that is, that the system called can also execute RFC modules in the calling system, then you need to set up an equivalent second RFC connection in the system called.

**Hint:** When you are defining RFC destinations, RFC connections are

- Addressed to **one** client, when they are pointing at an SAP system
- Accessible from all clients in the system

In ABAP, you use RFCs to call a function module in another system as follows:

```
CALL FUNCTION <Name>
  DESTINATION <Target>
  EXPORTING ...
  IMPORTING ...
```

The function to be executed in the target system is named. The name of the target must refer to one of the RFC connections available. When you are creating an RFC connection, you can specify logon data for the target system; if you do not do this, you need to enter logon parameters when you start the RFC. `Exporting` and `Importing` are used to pass parameters to the target function and to receive the returned parameters. The function called in the target system is executed using the user ID entered for the connection.

**Note:** It is also possible to create RFC connections with which the user calling this connection is used in the target system. That is, the same connection can be used by different users in the target system. This procedure is also called "Trusted RFC". It is, of course, a prerequisite that identical users are created in the source and target systems. Trusted RFC is explained in the course **ADM960** - *Security in SAP System Environments*.

The RFC has become the most important interface in the SAP environment. Some special RFC modules, which follow certain conventions, are also known as BAPIs (Business Application Programming Interfaces).

## BOR and BAPIs

A Business Application Programming Interface (BAPI) is a standardized programming interface that facilitates internal and external access to business processes and data in SAP systems. BAPIs are defined in the Business Object Repository as methods of SAP business objects and enable an object-oriented view of business data in an SAP system. Functions that can be called using BAPIs are normally implemented and stored in the ABAP Workbench Function Builder as RFC-enabled function modules. You can obtain an overview of the available BAPIs in the BOR, for example by choosing the pushbutton with the same name from the Business Object Builder (*Tools → ABAP Workbench → Development → Business Object Builder*), transaction SWO1. You can access the BOR directly using transaction code BAPI.

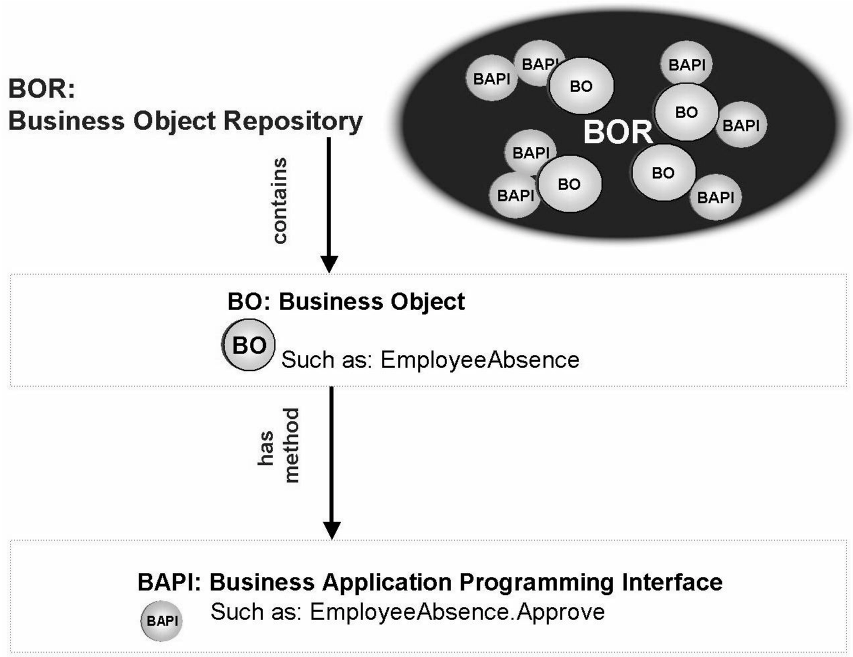

BOR:
Business Object Repository
BOR
BAPI
BO
contains
BO: Business Object
BO
Such as: EmployeeAbsence
has method
BAPI: Business Application Programming Interface
BAPI
Such as: EmployeeAbsence.Approve

**Figure 55: BOR and BAPIs**

BAPIs, which represent methods for business objects in an SAP system, are used in a variety of contexts.

**Possible Uses for BAPIs:**

- To link business processes across system boundaries (for example, with ALE)
- Used by SAP to integrate various solutions in the framework of mySAP Business Suite
- To connect an SAP system to the Internet
- Used in conjunction with SAP Business Workflow
- To connect to external programs

You must make sure beforehand that you are familiar with the execution of the following demonstration. If possible, avoid executing this function using the dummy RFC connection "NONE"; instead, with the participants, create a connection in transaction SM59. You could also prepare this beforehand.

**Hint:** This demonstration consists of two parts that the participants will then do themselves in the exercises. The first part concerns itself with a simple presentation of the BOR using a documentation call for a BAPI as an example. The second part is rather more complex, as it introduces BAPI usage with the Function Builder.

1. Follow the path mentioned above for the BOR, or call transaction BAPI directly.
2. Choose the *Alphabetical* tab page.
3. Search for the business object "USER".
4. From the BAPIs available for this business object, select the "Change" BAPI.
5. Display the documentation for this BAPI.
6. Now switch to the **USER.Display** BAPI. Go to the *Tools* tab page and choose the *Function Builder*. Then choose *Display*.

   You can now see the source code for the BAPI in the **Function Builder**.
7. Press *Test/Execute* (or the F8 key).

   Specify `NONE` as the RFC target system entry; this refers to your own system. Enter your own user name. Choose *Execute* (or the F8 key). As a result, your user data, requested by the RFC from the BAPI, is displayed.
8. Once you have confirmed this screen, the result screen for the query is displayed. You can view the return code and the response time for your query. This response time includes the time you took to view the user data display.

**Note:** BAPIs are created and tested in exactly the same way as other function modules, using the Function Builder, transaction SE37, and are then defined as BAPIs in the BOR.

195

# Exercise 15: Remote Function Calls and BAPIs

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Use BAPIs

## Business Example

You need data from another system.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task Using a BAPI

Use a BAPI to display the address data for your user in the system.

1. Start the overview transaction for the Business Object Repository.

2. Find the method *USER.Change*. Display the documentation for this method.

3. Call the Function Builder for the *USER.Display* method. You can now see the source code for the BAPI in the **Function Builder**.

4. Execute this BAPI for your user with the RFC destination **`NONE`**.

5. Once you have confirmed the dialog box, the result screen for the query is displayed. You can view the return code and the response time for your query.

### Result

In this example, the export parameter for the calling side was the user "in the other system", the import parameter was the receipt of the dialog box. From the point of view of the function **called**, the import parameter was the user name and the export parameter was the dialog box.

# Solution 15: Remote Function Calls and BAPIs

## Task Using a BAPI

Use a BAPI to display the address data for your user in the system.

1. Start the overview transaction for the Business Object Repository.

   a) Use *Tools → ABAP Workbench → Development → Business Object Builder* and choose *Business Object Repository*, or call transaction BAPI directly. If you are following the Business Object Builder menu path, then select “BAPI” in the dialog box.

2. Find the method *USER.Change*. Display the documentation for this method.

   a) Use the *Alphabetical* tab page and look for the *User* business object.

      From the BAPIs available for this business object, select the *USER.Change* BAPI. Use the appropriate tab page to display the documentation for this BAPI on the right side of the screen.

3. Call the Function Builder for the *USER.Display* method. You can now see the source code for the BAPI in the **Function Builder**.

   a) Switch to the *USER.Display* BAPI. Choose the *Tools* tab page, followed by the *Function Builder*. Then choose *Display*.

4. Execute this BAPI for your user with the RFC destination **`NONE`**.

   a) Choose *Test/Execute* (or the F8 key).

      Specify **`NONE`** as the RFC target system entry; this refers to your own system. Enter your user name on the query screen. Choose *Execute* (or the F8 key). As a result, your user data, requested by the RFC from the BAPI, is displayed.

*Continued on next page*

5. Once you have confirmed the dialog box, the result screen for the query is displayed. You can view the return code and the response time for your query.

   a) The return code for your query is displayed as the value for the export parameter *Return*.

## Result

In this example, the export parameter for the calling side was the user “in the other system”, the import parameter was the receipt of the dialog box. From the point of view of the function **called**, the import parameter was the user name and the export parameter was the dialog box.

## Lesson Summary

You should now be able to:

- Name various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs

### Related Information

- You can find all SAP interfaces on the Internet at: http://ifr.sap.com
- For information about other interfaces, see the Interface Adviser: http://service.sap.com/interfaceadviser

# Lesson: SAP Business Workflow

199

Lesson Duration: 20 minutes

## Lesson Overview

This lesson provides an overview of the concept and capabilities of the SAP Business Workflow (referred to simply as workflow).

## Lesson Objectives

After completing this lesson, you will be able to:

- Describe the SAP Business Workflow concept
- Explain the flow of a workflow process
- Describe additional application areas for the SAP Business Workflow concept

The example of employee absence processing is used to illustrate the workflow topic. If you have in-depth knowledge of workflow, you can demonstrate the standard example with the FORMABSENC business object. You can find further information on this subject in the SAP Business Workflow training courses.

## Business Example

The leave request process is a good example of how workflow can be used.

## Workflow Basics

Workflow in SAP systems (or even between SAP systems) aims to increase the speed and transparency of business processes. A workflow model breaks a process down to its individual steps, which are then assigned to various people, or rather, to their roles within the company. The automated sequence of steps ensures that tasks are rapidly assigned to the appropriate employees. This lesson describes two views of the same workflow. One is the view of the people participating in the workflow, the other is a more technical view designed to clarify the process in the system.

## A Workflow and its Participants

The process “an employee requests leave of absence” is used as a typical example.

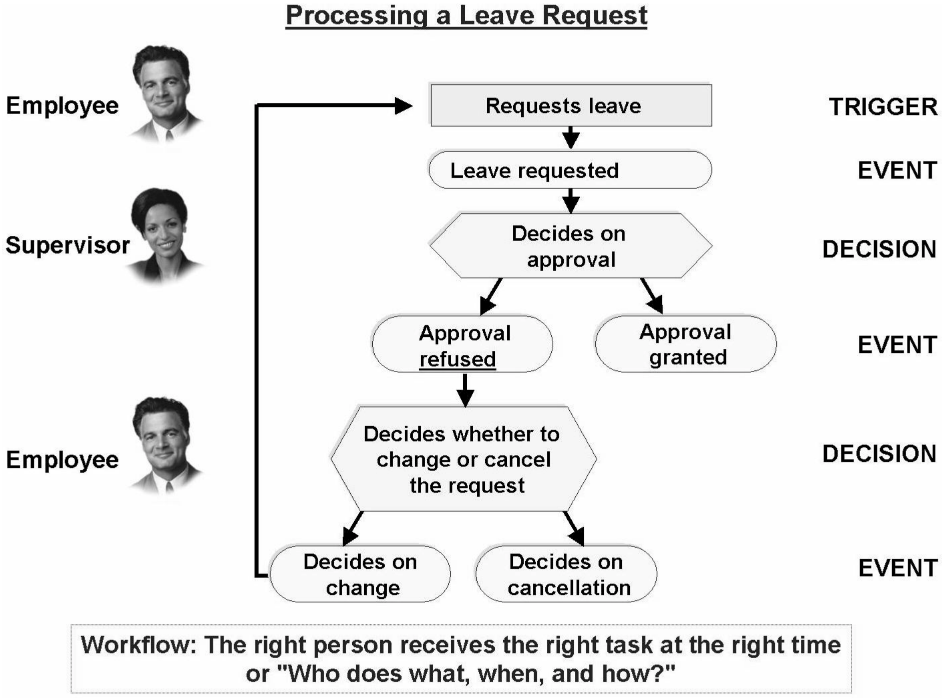


**Figure 56: Example of an SAP Business Workflow process**

Two people are involved: the person who creates the request (requester) and his or her assigned supervisor (approver). The requester fills out the form and saves his or her entries. The save action triggers a workflow event, for example, “leave requested”. This event is received by an appropriately configured workflow and passed to an “approver” in accordance with predefined rules. The supervisor (or “approver”) receives a corresponding workflow item in his or her “Office Inbox” (*Office → Work Center → Inbox*, transaction SBWP). A workflow event therefore creates a link between an activity in the SAP system and the people involved. When the approver calls up the workflow item, the approver is automatically referred to the function “Approve/reject request”. There are now two scenarios to consider:

1. The request is approved
2. The request is rejected

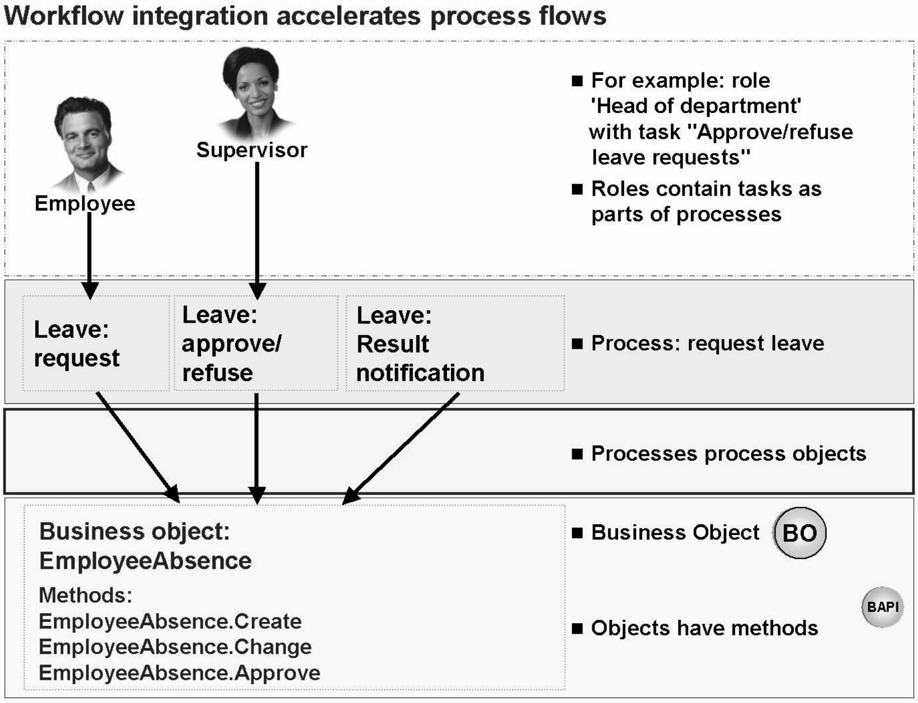
If the request is approved (which triggers a workflow event), the requester is informed and the workflow is complete. If the request is rejected, the requester is also informed and has in turn two options:

1. Accept the rejection
2. Change the leave request

If the requester accepts the rejection, the workflow is also complete; if the requester changes the leave request, another workflow item is sent to the approver's Office Inbox.

## A Workflow and its Technology

A workflow creates a link between the people who participate in a process and the program steps that belong to this process.



Workflow integration accelerates process flows

- For example: role 'Head of department' with task "Approve/refuse leave requests"
- Roles contain tasks as parts of processes
- Process: request leave
- Processes process objects
- Business Object
- Objects have methods

Business object: EmployeeAbsence

Methods:
EmployeeAbsence.Create
EmployeeAbsence.Change
EmployeeAbsence.Approve

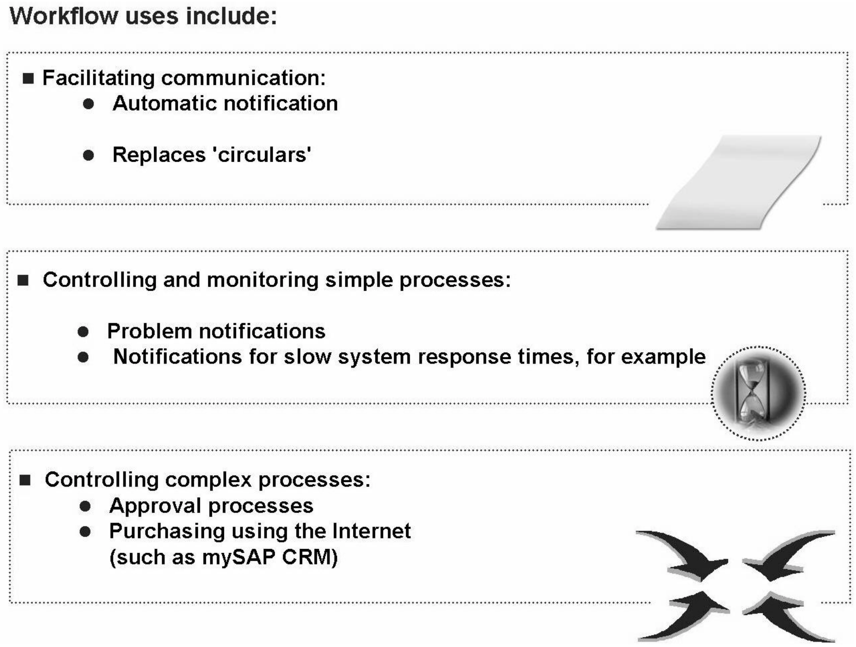
**Figure 57: The workflow environment**

Different authorizations enable employees in a company to carry out different tasks in one or more systems. Some of these tasks trigger events that are in turn assigned to steps in a workflow model. In other words, when an employee uses his or her authorizations to carry out a task that triggers a workflow event assigned to it, then, for example, a specific method of the relevant business object is called in the system.

An employee's authorizations typically grant access to individual steps in a more comprehensive process. A process works on a particular business object, for example, that is accessed using the methods assigned to it. These methods are defined as BAPIs in the system.

## Workflow Application Areas

Since an unlimited number of workflow steps can be assigned to a workflow event, and complex workflow sequences can be assigned to a process, the workflow function is used in a wide variety of business areas in SAP systems. For example, complex process flows in the mySAP CRM or mySAP SCM solutions would hardly be possible without workflow integration.



Workflow uses include:

- Facilitating communication:
  - Automatic notification
  - Replaces 'circulars'
- Controlling and monitoring simple processes:
  - Problem notifications
  - Notifications for slow system response times, for example
- Controlling complex processes:
  - Approval processes
  - Purchasing using the Internet (such as mySAP CRM)

Figure 58: Workflow application areas

Workflow is also particularly well suited to automating the distribution of information on work in progress (for example, information on the status of a purchase order) to all those concerned. Workflow can additionally use generated XML messages to trigger cross-system activities in remote systems. Defining appropriate events and assigning them to business object methods enables you to use workflow in almost any area.

## Facilitated Discussion

Exploring the possibilities of the SAP Business Workflow

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

What other specific uses for workflow are the participants aware of, or can they think of any other situations in which workflow could be used?

## Lesson Summary

You should now be able to:

- Describe the SAP Business Workflow concept
- Explain the flow of a workflow process
- Describe additional application areas for the SAP Business Workflow concept

### Related Information

For more information, see the SAP Library under *SAP Web Application Server → Business Management → WebFlow Engine (BC-BMT-WFM)* and in more advanced courses about the SAP Business Workflow: **BIT600**, **BIT601**, **BIT603**, and **BIT610**.

# Lesson: Internet Technologies

204

Lesson Duration: 50 minutes

## Lesson Overview

In this unit, you learn about the basic possibilities for connecting your SAP systems to the Internet.

## Lesson Objectives

After completing this lesson, you will be able to:

- Describe the areas in which the SAP Internet Transaction Server (SAP ITS), the SAP Web Application Server (SAP Web AS), and the SAP Business Connector (SAP BC) are used
- Describe how the SAP ITS works
- Describe how the SAP Web AS works
- Describe how the SAP BC works

This lesson provides a large amount of information about possibilities for connecting SAP systems to the Internet. You must ensure, while you are presenting these options that you describe their functions and possibilities precisely and draw clear distinctions between them. After this lesson, the participants must have a clear idea of which technologies they should use for which tasks.

## Business Example

You are planning to introduce e-business activities and need to get an overview of the technologies available.

## Overview

Since SAP Web AS 6.10, the following possibilities exist, among others, for connecting an SAP system to the Internet.

- Connect using the SAP Internet Transaction Server (SAP ITS)
- Exchange data using the SAP Web Application Server's (SAP Web AS) Internet Communication Manager (ICM)
- Exchange business data in HTTP and XML using the SAP Business Connector (SAP BC)

You can also distinguish between these approaches by the way they are used:

The SAP ITS and the SAP Web AS enable user interaction in dialogs, while the SAP BC enables automated data exchange between different systems. You can also operate the SAP BC as a stand-alone software component with the SAP Exchange Infrastructure (SAP XI) 2.0. While the SAP XI is primarily aimed at business integration of different systems within a company, the SAP BC is well-suited to cross-enterprise data exchange. For this reason, the SAP BC is introduced in this lesson.

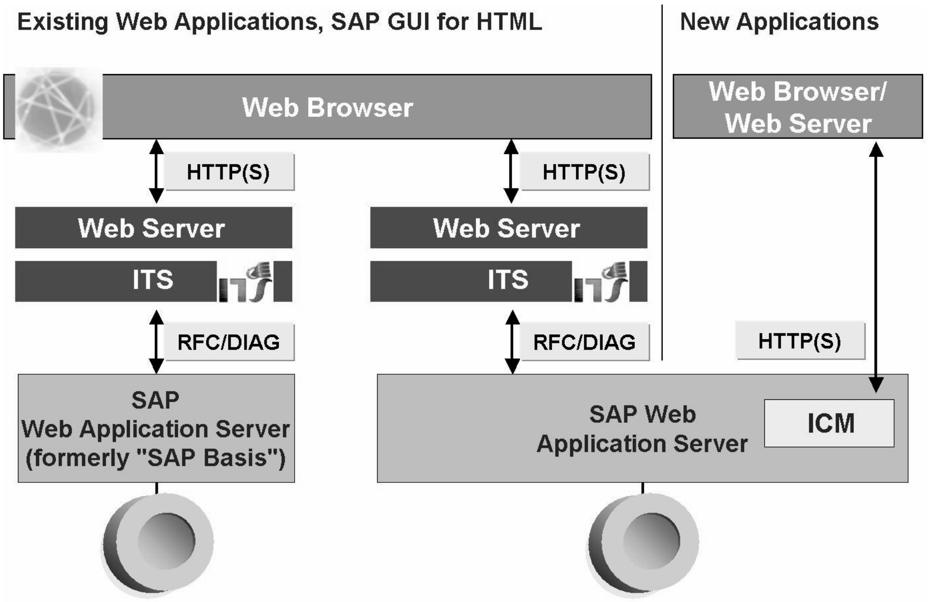

Existing Web Applications, SAP GUI for HTML
New Applications
Web Browser
Web Browser/ Web Server
HTTP(S)
HTTP(S)
Web Server
Web Server
ITS
ITS
RFC/DIAG
RFC/DIAG
HTTP(S)
SAP Web Application Server (formerly "SAP Basis")
SAP Web Application Server
ICM

**Figure 59: SAP ITS, SAP Web AS: HTTP communication**

As illustrated in the above graphic, the SAP ITS can also still be usefully implemented in a SAP Web Application Server environment, and has certainly not become obsolete.

## SAP Internet Transaction Server (SAP ITS)

**The SAP ITS provides the following options:**

- Automatic conversion of SAP screens into HTML format (SAP GUI for HTML)
- Use of screen-based IACs (Internet Application Components). This method of presenting business processes on the Web works with business transactions in adaptable HTML formats predefined by SAP
- Use of flow-file-based IACs. The Internet formatting is developed by the customer, and the business process flow is controlled using flow files

The **SAP ITS** enables SAP systems connected to it to communicate using the Hypertext Transfer Protocol (HTTP). The SAP ITS acts as a gateway to the SAP system. One of the SAP ITS's principal tasks in this context is converting the communication protocol from HTTP into DIAG or RFC and back.

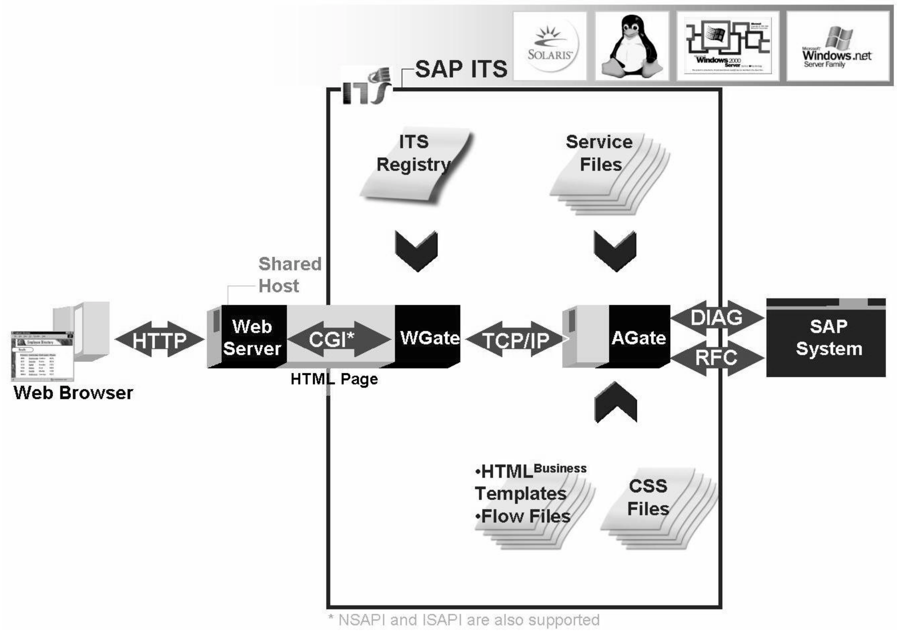


**Figure 60: How the SAP ITS works**

The Web server transmits requests sent by the browser to the SAP ITS **WGate** (Web Gateway). The WGate forwards the request to the second SAP ITS component, the AGate (Application Gateway). The **AGate** loads a service file, specified in the URL, from the AGate host's file system and uses the entries in the service file to identify the SAP system that it needs to address. The AGate then starts the logon process for that SAP system. The request is converted into DIAG or RFC protocol for the SAP system.

**Note:** The service files include information on the SAP system. The service file selected also determines which Web application is started. One example of a service file is the file webgui.srvc, which makes the SAP GUI for HTML available. A URL that would call the SAP GUI for the DEV system would be, for example:

http://igDEV.wdf.sap-ag.de:1080/scripts/wgate/webgui/!

The reply of the SAP system (SAP BW, SAP CRM, SAP R/3, and so on) uses DIAG or RFC and is directed to the ITS **AGate** (application gateway). On the SAP system side, the ITS behaves as SAP GUI for Windows or SAP GUI for Java.

The AGate uses HTML$^{Business}$ templates to generate an HTML page as the reply for the browser. The WGate then uses the CGI interface (Common Gateway Interface) or NSAPI/ISAPI to transmit this page to the relevant Web server. The Web server then forwards this page to the browser front end that sent the original request.

**Note:** The SAP ITS WGate must run on the same hardware as the Web server.

The SAP ITS consists of two software components, the AGate and the WGate, that can run on separate hardware in production operation.

**Summary of SAP ITS Tasks:**

- Log browser session on to SAP system
- Administer session information
- Convert HTTP to DIAG or RFC and back
- Control process flow using flow files (optional)

**Hint:** The use of flow files to represent Internet-based business processes is supplemented by the use of the SAP Web AS ICM. If you are planning development work related to Internet-based business processes, make sure that you are well informed about the possible uses of the SAP Web AS, for example, by attending the training courses **NET050** or **NET200**.

There are still a small number of restrictions for the SAP GUI for HTML (see SAP Note 314568). All elements of SAP screen images are now fully convertible, so you do not need to program additions to SAP GUI for HTML if you want to use your browser as a graphical front end.

You can download the current version of the SAP ITS at http://service.sap.com/patches. For information about SAP ITS, see http://service.sap.com/sap-its and the training course **NET170**.

## SAP Web Application Server (SAP Web AS)

A classic SAP system is implemented with a three-tier client/server architecture:

- Presentation layer
- Application layer
- Database layer

The system is scalable with regard to the presentation and application level. Good scalability is an important prerequisite for enabling a number of users to work simultaneously in the system. The **SAP Web Application Server** is a further development of classic SAP Basis technology. A new process has been added to the SAP kernel, the **Internet Communication Manager** (ICM). This enables **direct** processing of requests from the Internet, sent, for example, from a browser using HTTP protocol. This ensures the scalability of the SAP Web Application Server.

**Note:** The message server is responsible for load balancing for working with SAP GUI in an SAP system that consists of several application servers. The SAP Web Dispatcher performs this load balancing for incoming requests from the Internet. The SAP Web Dispatcher is a small program that was first delivered with the kernel of SAP Web AS 6.20.

### Details on the Internet Communication Manager

The ICM enables SAP systems to communicate outside the SAP environment using the HTTP, HTTPS and SMTP protocols. The ICM can process requests from the Internet that include its server/port combination in their URLs. If database data is required for processing the request, then a connection to a work process is created using memory pipes.

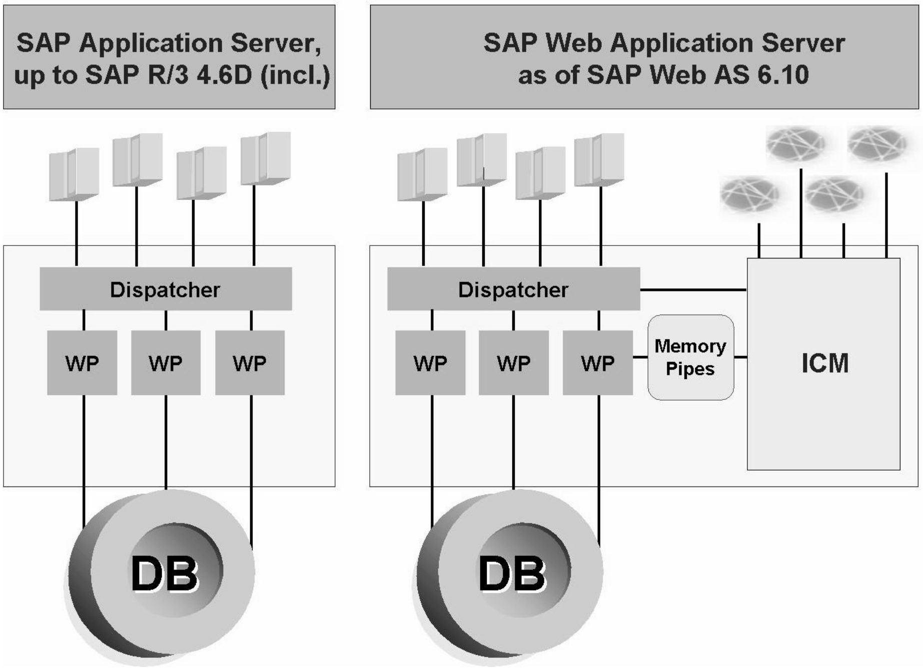

SAP Application Server, up to SAP R/3 4.6D (incl.)
SAP Web Application Server as of SAP Web AS 6.10
Dispatcher
WP
WP
WP
DB
Dispatcher
WP
WP
WP
Memory Pipes
ICM
DB

**Figure 61: The SAP Web Application Server**

Within a work process, the Internet Communication Framework (ICF) provides the environment for handling HTTP requests. The ICF acts as a bridge between the C kernel of the SAP system and the application program, written in ABAP.

As of SAP Web AS 6.10, work processes are able to directly create Web-enabled content that the ICM then transmits to the browser front end that sent the original request. You can develop this content, called **Business Server Page applications**, in the SAP system using a tool in transaction SE80, the **Web Application Builder**.

Business Server Pages (abbreviated to BSPs) enable you to map complex Internet business processes in the SAP Web AS. You can use the SAP system to create some of the graphic elements required for a corporate website (such as HTML pages, or Web themes) and to manage (and store) those and other elements (such as the MIME objects used). You can use both ABAP and JavaScript as scripting languages. The business logic is created in ABAP Objects. ABAP Objects is an object-oriented enhancement of the ABAP programming language. It enables you to use the principles of object-oriented programming in ABAP by using concepts such as encapsulation, inheritance, classes, and interfaces.

Often, SAP customers already have tools for creating attractive corporate websites. To enable you to continue using these tools, SAP systems support the WebDAV standard (DAV = Distributed Authoring and Versioning). In other words, you can appropriate pages in the SAP Web AS, although you need not, if you prefer to use other tools.

## The SAP Business Connector (SAP BC)

The **SAP Business Connector** (abbreviation: **SAP BC**) provides technology that enables simple integration of applications across system boundaries. The differences between using the SAP BC as opposed to RFCs or IDocs as transfer protocols/data formats are as follows:

- The open (non-proprietary) XML standard is used as the data format
- Data is exchanged using the Internet standard protocol HTTP
- Secure communication using an SSL (Secure Socket Layer) connection is possible

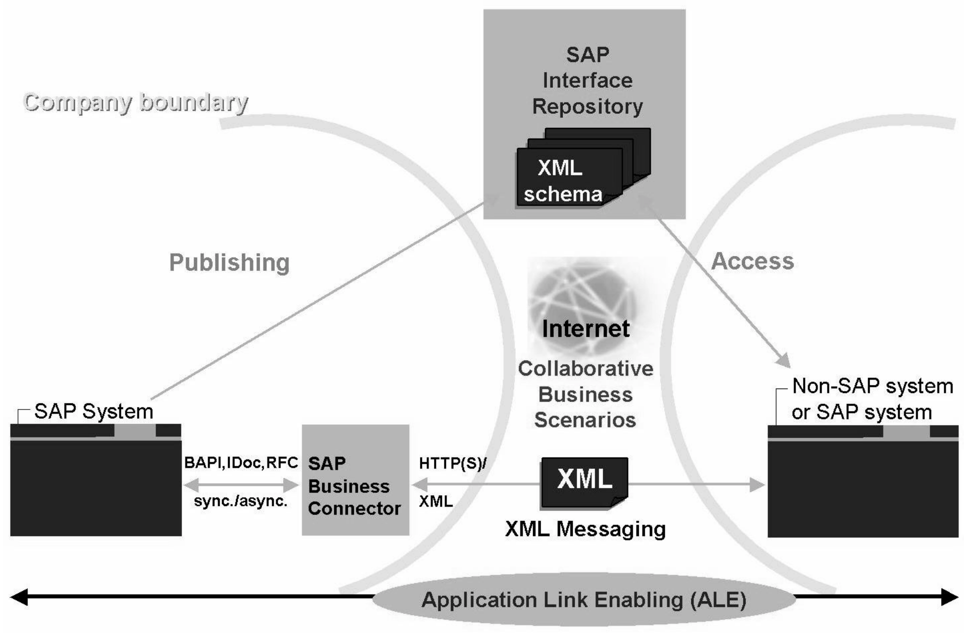


**Figure 62: The SAP Business Connector environment**

The SAP BC, like the SAP ITS, is not integrated into the SAP Web AS runtime environment; instead it is an independent software component with its own release cycle. This guarantees that the functions provided are as up-to-date as possible. At the same time, the use of standard SAP interfaces such as RFC, BAPI, and IDoc ensures complete compatibility with all SAP solutions, regardless of the release of the SAP system.

Since the SAP BC can communicate with all XML-enabled non-SAP systems (or rather, their XML converters) you can also use it in conjunction with ALE; in other words, you can map inter-enterprise, Internet-based business processes.

To use XML to exchange data, both business partners must agree on an XML schema that they both use and that both systems can interpret. The schema defines how each XML tag is to be interpreted in the specific business context. Since one company may well want to exchange data in XML format with a number of other companies, it is vital that you ensure that all participants in the process are using the same XML schema. One simple method of exchanging XML schemas is provided by the SAP Interface Repository (SAP IFR). You can access the Interface Repository on the Internet at http://ifr.sap.com. There you can call up all standard XML schemas archived by SAP; once you have downloaded one of these schemas, you can request notification should it be changed. This ensures that all participants always use the most up-to-date schemas.

You can download the basic version of the SAP BC free of charge at http://service.sap.com/connectors. This ensures that you can always use the most recent version of the SAP BC. In addition to the encryption provided by HTTPS, you can encrypt the XML messages using a licensed version of the SAP BC, or using software provided by one of SAP's partners.

**SAP Exchange Infrastructure**

SAP XI integrates the existing business software solutions in your company. As SAP XI provides a central integration platform, even applications from different vendors can easily use it to exchange data. The data exchange is performed, for example, in an XML protocol, which is "understood" from the applications involved. However, SAP XI also provides the option of converting data streams that do not exist in XML format into XML format. There are a large number of "mapping" mechanisms available for this conversion. Using SAP XI can significantly reduce the number of "point-to-point" interfaces required between your applications and help to simplify your IT landscape. SAP XI is therefore implemented to simplify your enterprise-internal data interfaces.

# 213 Exercise 16: Internet Technologies

Exercise Duration: 15 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Create a simple BSP application.

## Business Example

This could be the first step to Internet-enabling your company.

### System Data

**System:** The training system assigned to you (such as I30)
**Client:** The client assigned to you (such as 802)
**User ID:** The user name that you have given the participants (such as SAPTEC-##)
**Password:** The password that you have given the participants for the relevant user.
**Set up instructions:** None

## Task Creating a Simple Business Server Page (BSP)

Display simple output from your SAP system in your browser.

1.

The following setting may be required (although it is no longer required in the available training system at the time this material was produced):

In transaction SE80, choose *Utilities* → *Settings...* → *Business Server Pages* → *Application Server* and specify there the name of the application server of the training system of which you want to use the ICM. You can find the host server name from the status bar of an SAP GUI window. The complete address is then: <hostname>.wdf.sap.corp. You may also require the configured port number for the HTTP protocol; you can find this out in transaction SMICM by choosing *Services*.

Call transaction SE80 and create a BSP application named **Z<CLNT>_BSP_##**. <CLNT> is the client you are logged on to and ## is your group number. The application will therefore be called, for example, Z100_BSP_01.

*Continued on next page*

2. Enter the page path (= page name): Hello_##.htm for the page to be created. ## again corresponds to your group number.

3. On the *Layout* tab page, enter the following lines between the "body tags":

   ```
   <center>

   Hello World!

   </center>
   ```

4. Save and activate all objects you have created. Then carry out a test using the right mouse button. You need to enter your user ID and password for the training client. The client is predefined (see below) and the language is the original language of the object called.

**Note:** You may need to add the following string to the generated URL in your browser window: **`?sap-client=###`**.

### in this case corresponds to the number of the client in which you are working. You can set the default client [100] in transaction SICF, although this is not covered in more detail here.

## Result

Congratulations, you have created your first BSP!

# Solution 16: Internet Technologies

## Task Creating a Simple Business Server Page (BSP)

Display simple output from your SAP system in your browser.

1.

> The following setting may be required (although it is no longer required in the available training system at the time this material was produced):
>
> In transaction SE80, choose *Utilities* → *Settings...* → *Business Server Pages* → *Application Server* and specify there the name of the application server of the training system of which you want to use the ICM. You can find the host server name from the status bar of an SAP GUI window. The complete address is then: <hostname>.wdf.sap.corp. You may also require the configured port number for the HTTP protocol; you can find this out in transaction SMICM by choosing *Services*.

Call transaction SE80 and create a BSP application named **Z<CLNT>_BSP_##**. <CLNT> is the client you are logged on to and ## is your group number. The application will therefore be called, for example, Z100_BSP_01.

a) Then carry out the following actions in transaction SE80:

   1. Select the object type "BSP Application" from the possible entries help for the "object type" field.
   2. In the field beneath it, enter the name of your object following the **Z<CLNT>_BSP_##** pattern. Choose *Return*.
   3. Confirm that you want to create the object.
   4. Enter a short description of your choice for the object.
   5. Either assign the object to a package specified by your instructor, or save it as a "Local object".

*Continued on next page*

2. Enter the page path (= page name): Hello_##.htm for the page to be created. ## again corresponds to your group number.

   a) Proceed as follows:

      - Right mouse click on the object name and choose *Create → Page*. Enter the following name as the pages path characteristic: **Hello_##.htm**. ## again corresponds to your group number.
      - Enter a meaningful description.
      - Choose *Page with Flow Logic*

3. On the *Layout* tab page, enter the following lines between the "body tags":

   ```
   <center>
   Hello World!
   </center>
   ```

   a) Enter the specified commands in the Editor.

4. Save and activate all objects you have created. Then carry out a test using the right mouse button. You need to enter your user ID and password for the training client. The client is predefined (see below) and the language is the original language of the object called.

**Note:** You may need to add the following string to the generated URL in your browser window: **`?sap-client=###`**.

### in this case corresponds to the number of the client in which you are working. You can set the default client [100] in transaction SICF, although this is not covered in more detail here.

   a) Choose *Save*, followed by *Activate*. Select all objects when you are activating.

      Navigate to your page and choose the entry *Test* from the right mouse-click context menu. Enter your user ID and password for the training system if you are prompted to do so. Note that the description you entered (see above) appears in the title bar of the browser.

## Result

Congratulations, you have created your first BSP!

## Lesson Summary

You should now be able to:

- Describe the areas in which the SAP Internet Transaction Server (SAP ITS), the SAP Web Application Server (SAP Web AS), and the SAP Business Connector (SAP BC) are used
- Describe how the SAP ITS works
- Describe how the SAP Web AS works
- Describe how the SAP BC works

### Related Information

You can find additional information on connecting SAP systems to the Internet in more advanced training courses about system administration (such as **NET200** or **BIT530**), and at the following addresses:

- http://service.sap.com/sap-its/
- http://service.sap.com/connectors
- http://ifr.sap.com
- http://www.sap.com/netweaver

# Lesson: Data Migration

217 Lesson Duration: 20 minutes

## Lesson Overview

In this lesson, you will learn about the possibilities that are available to you for migrating data in SAP systems with the Data Transfer Workbench (SXDA) and the Legacy System Migration Workbench (LSMW).

## Lesson Objectives

After completing this lesson, you will be able to:

- Name various data transfer procedures

This lesson is a challenge for the instructor: initial data migration is a very important topic for many customers. At this introductory level you should not, however, be discussing the technical programming details of data migration. There is a separate training course on this subject, the 5-day "Data Transfer" course. In this lesson you should concentrate on outlining the basic methods and technologies used.

## Business Example

In SAP implementation projects, the issue of transferring legacy data into the new system often arises. This lesson introduces the data transfer tools provided in the SAP system.

## Principles of Data Transfer

**There are two fundamental data transfer scenarios:**

- Initial (one-time) data transfer (for example, when the system is implemented)
- Periodic data transfer (for example, in an ALE scenario)

Since periodic data transfer basically follows the same patterns and uses the same methods as initial data transfer, this lesson concentrates on initial data transfers.

## Data Transfer Tools

The SAP system provides two important tools for data transfer:

- The Data Transfer Workbench (DX Workbench, transaction SXDA)

- The Legacy System Migration Workbench (LSMW, transaction LSMW)

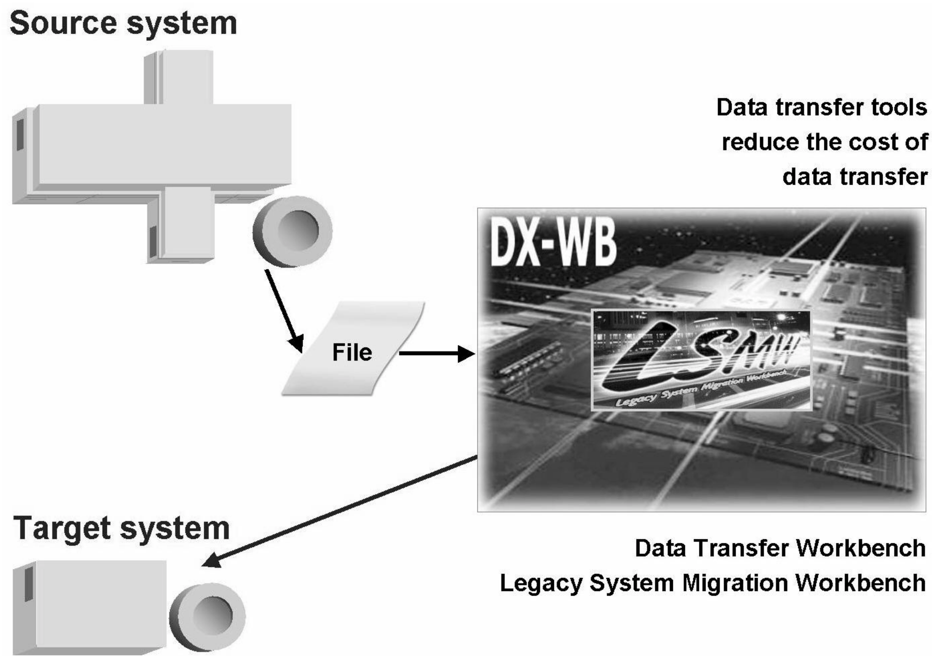

Source system
Data transfer tools reduce the cost of data transfer
DX-WB
File
LSMW
Legacy System Migration Workbench
Target system
Data Transfer Workbench
Legacy System Migration Workbench

**Figure 63: Data transfer tools**

The DX Workbench is available in every SAP system after installation and can be used to manage and carry out both periodic and initial data migration projects.

The LSMW is also available in the DX Workbench once you have imported an additional transport request into the system. The LSMW is primarily a conversion tool. It enables you convert external formats into SAP data formats (such as the IDoc format). You can also start data transfer in the LSMW.

The LSMW is available on the SAP Service Marketplace, using the Quick Link */lsmw*, then the download area on that page.

The three basic steps involved in data migration are:

- Extract data from source system
- Convert data or "map" the relevant data fields
- Import data into the target system

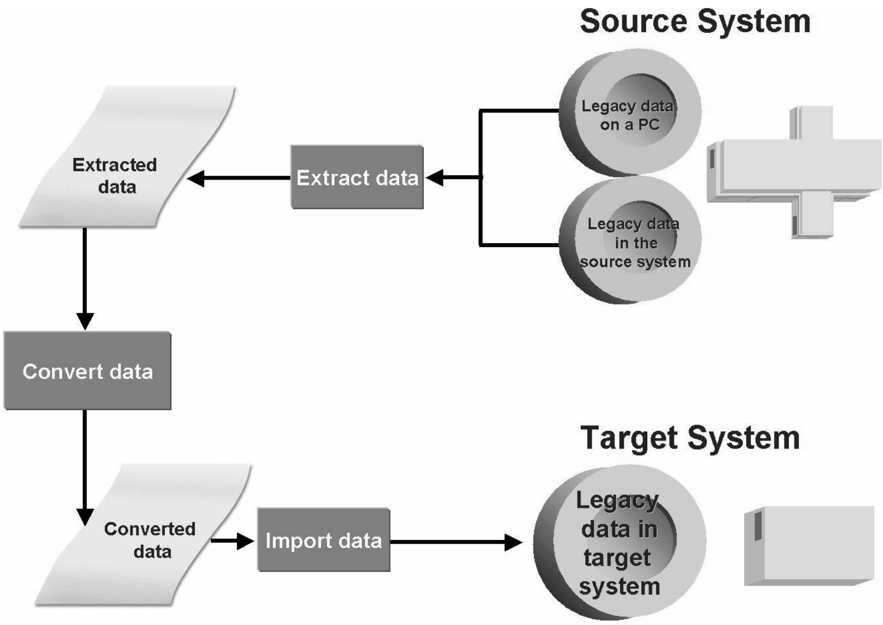


**Figure 64: The three steps of data migration**

You need to use tools in your source system for the first step of the process, extracting the data from the source system. The data extracted can basically be in any format.

To be able to convert the data correctly, you need to answer the following questions in detail:

- What data needs to be transferred?
- What table fields in the source system correspond to this data?
- What table fields in the SAP system correspond to this data?
- Does the source data cover all the entries required by the SAP system?

The question of whether all required fields in the SAP system can be filled with data from the legacy system helps you to decide whether you need to change field attributes from “mandatory” to “optional”, or whether you can set fixed values as default values for the fields required by the SAP system.

## Data Transfer Technology

Mature Technologies Are Available for Data Imports:

- “Classic” batch input
- Direct input (where available)
- Business object methods (BAPIs)
- IDoc technology (ALE)

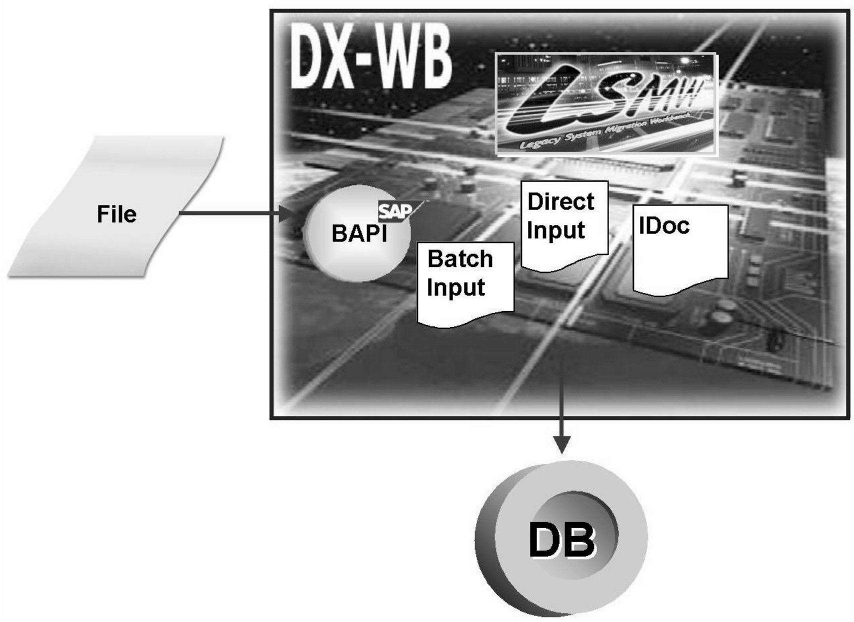


**Figure 65: Data transfer technology**

Two of these four technologies are discussed below:

- “Classic” batch input
- Using BAPIs to import data

## “Classic” Batch Input

The batch input is described here as “classic” to denote that this method has been in use for several years.

Batch Input: Transaction SM35. You can point out here that there is a recording function. You can run standardized processes more than once, meaning that you can also use the batch input procedure for special periodic data transfers. The prerequisite for this is that there is a (new) batch input folder at the relevant time. You may be able to realize this using scripts or programs at operating system level.

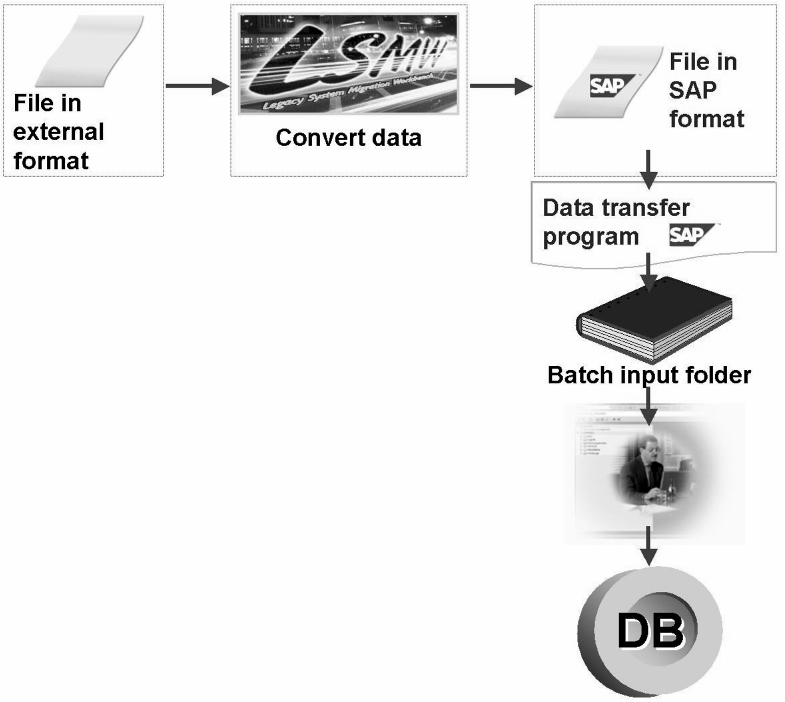


Figure 66: Data transfer with batch input

"Classic" batch input flow is as follows:

1. External data is converted into a format specified by the SAP system (batch input folder).
2. This data is then imported into standard SAP transactions using background processing.

During this process, the data consistency is checked in the same way as if the data had been entered in the transactions manually.

The source system data must first be converted into the simple format of the batch input folder using a data transfer program. You can create the data transfer program semi-automatically in the SAP system. Both the Data Transfer Workbench and the Legacy System Migration Workbench provide comprehensive functions for this purpose.

## Data Transfer Using BAPIs

You can also use business objects and their methods for initial data migration. Business Application Programming Interfaces (BAPIs) have been available since SAP R/3 3.1G. BAPIs are a user-friendly way of importing data into an SAP system. BAPIs have all the characteristics required to fulfill this task: security, speed, integrity, and ease of use.

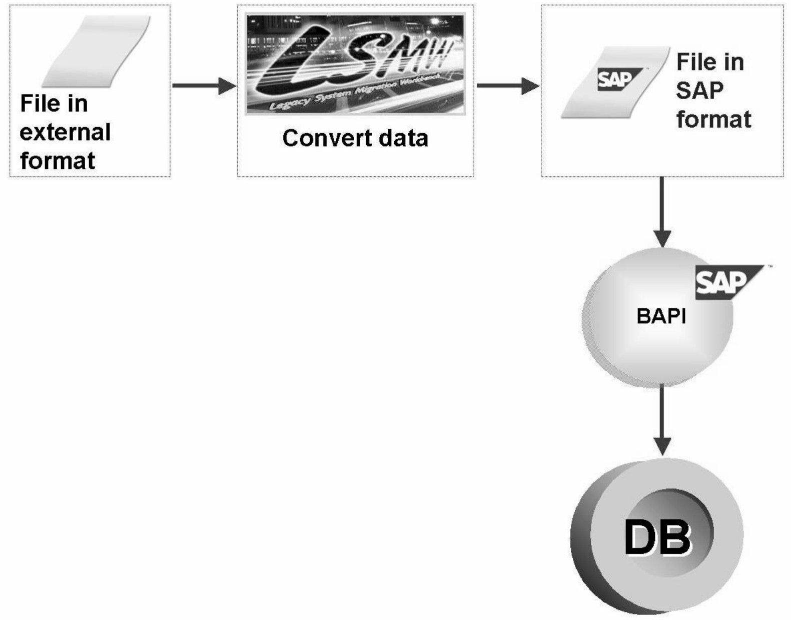


Figure 67: Data transfer using BAPIs

If you are using the BAPI interface, the data to be imported again needs to be in a defined format.

## Further Advantages of Data Transfer Tools:

- Use of standardized, tested procedures
- Reusability of data transfer projects
- Large selection of data conversion technologies
- Data conversion programs generated on the basis of conversion rules
- Included in maintenance contract, in other words, available to customers and partners free of charge
- Data consistency ensured
- Ease of use
- Little knowledge of ABAP required

**Hint:** Note that the tools discussed (SXDA and LSMW) do not write the data to the target system database themselves, but instead use interfaces that already exist. These tools provide user-friendly access to the various import options and comprehensive support for managing and programming a data transfer project.

## Facilitated Discussion

This optional discussion is intended to clarify the benefits of using the tools introduced in the lesson.

## Discussion Questions

Use the following questions to engage the participants in the discussion. Feel free to use your own additional questions.

What expectations do you (the participants) have of data migration tools? Do the tools provided by SAP meet these expectations?

## Lesson Summary

You should now be able to:

- Name various data transfer procedures

### Related Information

You can find additional information on the topic of data transfer

- in the more advanced training course about data transfer (**BC420**)
- At the URL http://service.sap.com/lsmw

## Unit Summary

You should now be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept
- Name various interface technologies used by SAP systems
- Describe the process for a Remote Function Call
- Explain the significance and use of business objects and their BAPIs
- Describe the SAP Business Workflow concept
- Explain the flow of a workflow process
- Describe additional application areas for the SAP Business Workflow concept
- Describe the areas in which the SAP Internet Transaction Server (SAP ITS), the SAP Web Application Server (SAP Web AS), and the SAP Business Connector (SAP BC) are used
- Describe how the SAP ITS works
- Describe how the SAP Web AS works
- Describe how the SAP BC works
- Name various data transfer procedures

225

# Test Your Knowledge

1. Application Link Enabling (ALE) allows you to:

   *Choose the correct answer(s).*

   □ A Exchange data only between SAP systems, as long as they have the same release status

   □ B Exchange data across system boundaries, but only for SAP applications

   □ C Exchange data between collaborating enterprises, using certain formats and technologies

   □ D The communication between different systems of your system landscape

   □ E Update your order data using the appropriate BAPI, only once every 24 hours

2. The following interfaces and communication options are supported by SAP systems:

   *Choose the correct answer(s).*

   □ A HTTP (HyperText Transfer Protocol)

   □ B SMTP (Simple Mail Transfer Protocol)

   □ C RFC (Remote Function Call)

   □ D BAPIs (Business Application Programming Interfaces)

   □ E XDTP (Extended Data Transfer Protocol)

   □ F STP (SAP Transfer Protocol)

3. You can use BAPIs to:

   *Choose the correct answer(s).*

   □ A Request data from an SAP system

   □ B Pass data to an SAP system

   □ C Transfer SAP screen images to third-party applications (such as MS Word)

   □ D Access business processes in SAP systems

4. You access BAPIs in SAP systems using an RFC interface.

   *Determine whether this statement is true or false.*

   □ True

   □ False

5. Business ____________________ Programming Interfaces are specialized _______________ modules. They are accessed using the ________ interface. They are created and managed using the ________________ Builder.

   *Fill in the blanks to complete the sentence.*

6. SAP Business Workflow ensures that:

   *Choose the correct answer(s).*

   □ A Appropriately configured business processes can be partially automated

   □ B Appropriately configured business processes are executed in consistent sequences

   □ C The right employee receives the right work at the right time

   □ D Your workflow-supported business processes are handled more efficiently

   □ E All your company processes that have been implemented in ABAP run without errors

7. You can also use SAP Business Workflow functions (for example, with XML) to trigger functions in other systems.

   *Determine whether this statement is true or false.*

   □ True

   □ False

8. The SAP ITS provides the following functions/options:

   *Choose the correct answer(s).*

   □ A Automatically converts protocols from HTTP to DIAG/RFC and back

   □ B Automatically generates SAP screens as HTML pages

   □ C Uses IACs to implement predefined Internet scenarios

   □ D Sends IDocs

   □ E The WGate communicates directly with dialog work processes

   □ F The AGate communicates with SAP system application servers

9. The following statements are true of the SAP Web Application Server:

   *Choose the correct answer(s).*

   □ A The SAP Web AS enables the creation of Business Server Pages

   □ B The ICM is a process outside the SAP Web AS, as is the SAP ITS

   □ C The ICM uses a Web server to communicate with Web browsers

   □ D You can use a URL to call Business Server Pages directly in your browser

   □ E You can only install the SAP Web AS in conjunction with the SAP ITS

10. The SAP BC enables ALE scenarios across company boundaries.

    *Determine whether this statement is true or false.*

    □ True

    □ False

11. SAP provides the following tools free of charge to help you organize your data transfer project:

    *Choose the correct answer(s).*

    □ A Data Transfer Workbench (DX Workbench)

    □ B Data Transfer Enabler (DTE)

    □ C Flat File Importer (FFI)

    □ D Legacy System Migration Workbench (LSMW)

    □ E Database Content Replicator (DCR)

    □ F ERP Data Transfer Wizard (EDTW)

12. The following methods and interfaces are available for importing data into SAP systems:

    *Choose the correct answer(s).*

    □ A BAPIs (Business Application Programming Interfaces)

    □ B IDocs (Intermediate Documents)

    □ C Batch input

    □ D DIIs (Data Import Interfaces)

    □ E STF (SAP table filler)

    □ F None of the above

13. Where can you download the Legacy System Migration Workbench?

229

# Answers

1. Application Link Enabling (ALE) allows you to:

   **Answer:** C, D

   ALE is a very powerful method of exchanging data between systems. These systems may be located within the same company, or they may be distributed between several companies. The data is transferred by RFC in a previously defined format. The transfer type may be synchronous or asynchronous.

2. The following interfaces and communication options are supported by SAP systems:

   **Answer:** A, B, C, D

   From an SAP system, you can communicate with other systems using, for example, HTTP, SMTP, RFC, or BAPIs. XDTP and STP do not exist.

3. You can use BAPIs to:

   **Answer:** A, B, D

   You can use BAPIs to access business processes in an SAP system and to request and transfer data between systems. GUI functions cannot be transferred to third-party products.

4. You access BAPIs in SAP systems using an RFC interface.

   **Answer:** True

   BAPIs are nothing other than special, remote-enabled function modules. They can therefore also be addressed using RFCs.

5. Business Application Programming Interfaces are specialized function modules. They are accessed using the RFC interface. They are created and managed using the Function Builder.

   **Answer:** Application, function, RFC, Function

   You can start the Business Object Repository using transaction code BAPI, and the Function Builder using transaction code SE37.

6. SAP Business Workflow ensures that:

   **Answer:** A, B, C, D

   Supporting business processes with workflow enables the work steps that belong to these processes to be handled on a partially automated basis. These steps are assigned in consistent sequences to the appropriate employee at the right point in time. This enhances processing efficiency. Using workflow obviously does not guarantee that the programs you write will always run without errors.

7. You can also use SAP Business Workflow functions (for example, with XML) to trigger functions in other systems.

   **Answer:** True

   You can use workflow to send XML messages that then trigger subsequent actions in other systems, as long as they are appropriately configured.

8. The SAP ITS provides the following functions/options:

   **Answer:** A, B, C, F

   The SAP ITS automatically converts protocols from HTTP to DIAG/RFC, and vice versa; it also generates HTML pages from SAP screens and makes predefined Internet scenarios available as IACs. The ITS does not send IDocs, nor does the WGate communicate directly with work processes. The application gateway carries out this part of the communication process.

9. The following statements are true of the SAP Web Application Server:

   **Answer:** A, D

   The SAP Web AS provides the ICM as a completely integrated process that enables you to call BSPs directly, using a URL. You do not need either an SAP ITS or a Web server to do this.

10. The SAP BC enables ALE scenarios across company boundaries.

    **Answer:** True

    Because the SAP BC can transfer data in XML format to other systems, it enables inter-enterprise ALE scenarios.

11. SAP provides the following tools free of charge to help you organize your data transfer project:

    **Answer:** A, D

    The Data Transfer Workbench and the Legacy System Migration Workbench.

12. The following methods and interfaces are available for importing data into SAP systems:

    **Answer:** A, B, C

    BAPIs, IDocs and batch input, along with direct input, are the primary procedures used to import data into an SAP system. The remaining answers are wrong.

13. Where can you download the Legacy System Migration Workbench?

    **Answer:** You can download the Legacy System Migration Workbench on the SAP Service Marketplace, using the Quick Link */lsmw* followed by the download area for that page.

# Unit 6

## 233 System Administration Tools and Aids

Future system administrators may, at this late stage of the course, let you know that they feel a little underrepresented by this one unit. You should point out that the “System Kernel” unit already dealt with a number of administration topics. The unit on interfaces is also of equal importance to administrators and developers. This unit should focus on the various aids and help options provided by the SAP Service Marketplace, rather than on the administration functions available in the system.

### Unit Overview

This unit introduces a range of tools designed for frequently-performed administrative tasks. You also gain an insight into the SAP Service Marketplace and what it has to offer, such as detailed information on a broad range of topics and access to the SAP Notes database.

### Unit Objectives

After completing this unit, you will be able to:

- Outline the basic principles of the authorization concept
- Create and copy users
- Describe the concept of Central User Management
- Explain how user management concepts utilize directory services with the LDAP protocol
- Name frequently-used administration functions in SAP systems
- Explain the concept of the SAP Solution Manager
- Name the central functions of the Computing Center Management System (CCMS)
- Use the SAP Service Marketplace
- Name some of the services provided on the SAP Service Marketplace

## Unit Contents

# Lesson: User Administration

234 Lesson Duration: 35 minutes

## Lesson Overview

This lesson explains how to create and maintain SAP users, and how to use existing roles to assign authorizations.

Alongside client-specific user management, this lesson also covers more advanced concepts, such as Central User Administration and accessing directory services using LDAP.

## Lesson Objectives

After completing this lesson, you will be able to:

- Outline the basic principles of the authorization concept
- Create and copy users
- Describe the concept of Central User Management
- Explain how user management concepts utilize directory services with the LDAP protocol

This lesson focuses on user management within a client. More advanced concepts, such as Central User Management and directory services, are only briefly discussed. Directory services are also known informally as “LDAP servers”. LDAP stands for Lightweight Directory Access Protocol and is a protocol used to access directory services.

## Business Example

You need to manage the users in your system and their authorizations.

## Principles of User Management

You must have a user master record to be able to log on to an SAP system. The user master record enables you to define which role is assigned to the user, that is, which activities are included in the user menu and which authorizations are assigned to the user.

User master records are client-specific. You must maintain separate user master records for each client in the SAP system.

You should explain the most important terms in the list.

**User Management Terms**

**User Name**

The string created in the system that enables a person to authenticate themselves in a client in the SAP system. System activities take place under this user name. Data from the user master record is used for the authorization check.

**User**

Abbreviated version of user name or SAP user. Sometimes also the person that has logged on to the SAP system.

**User Master Record**

User name and user data.

**User Data**

Data for a user name

**Authorization**

The authority to perform actions in the SAP system. You need to have different authorizations for different actions. Authorizations for activities refer, for example, to the following activities, for example: calling a transaction, displaying print requests, changing customer data

**Authorization Profile**

A collection of authorizations. SAP delivers a large number of single and composite profiles.

**Role**

The functional description of an SAP user with the associated authorization profile and user menu entries. Roles are centrally administered in the system. You can assign one role to multiple users; one user can also have multiple roles.

## The Authorization Check Concept

A role contains authorizations for the SAP system and menu entries for the user menu. Roles are maintained by the system administrator or by the relevant departments. If a role is assigned to a user, then this role is entered in the user master record for that user, who consequently receives the authorizations attached to the role. The transactions that this role gives the user authorization for are then listed in the user menu. You can assign multiple roles to one user.

When a user logs on to the SAP system, the authorizations from the roles stored in the user master record are entered in the user context. The user context is stored in the shared memory of the application server that the user is logged on to. Rolling in the user context enables the work process that is handling the user request to access the user's authorizations. These authorizations are used for authorization checks when the user starts transactions and works in them.

**Caution:** If roles are changed or if new roles are assigned to the user, then these changes usually only take effect when the user next logs on to the SAP system.

Authorization checks take place on two levels:

- The first-level check ensures that the user is authorized to call the transaction.
- The second-level check ensures that the user has the authorizations required for individual actions. This normally requires several authorization checks.

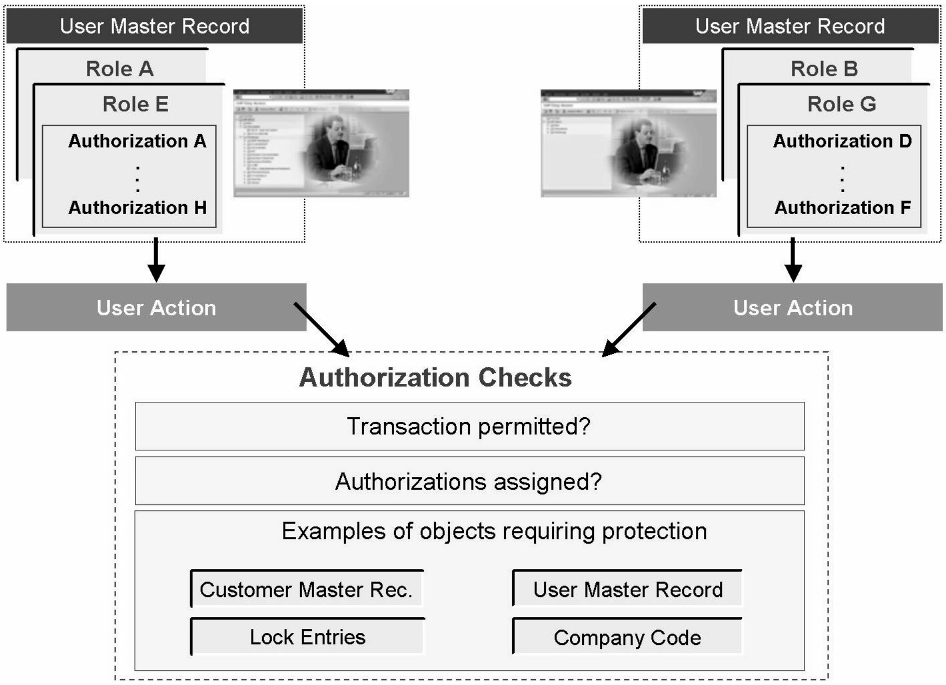


**Figure 68: SAP authorization checks**

## Authorization Check Process

When a user calls a transaction, the work process checks whether the user is authorized to start that transaction.

- If the user does not have authorization, the system rejects the request and displays the corresponding message.
- If the user is authorized to start the transaction, the initial screen of that transaction is displayed.

When a user enters data in a transaction or selects a function, the system again checks whether the user is authorized to make these entries or use the function. If the user has the relevant authorizations, then the user request is processed; if not, then the system rejects the user action.

## Creating and Maintaining User Master Records

You can access user maintenance using *Tools* → *Administration* → *User Maintenance* → *Users*. You can also call transaction SU01. You can create a new user master record by copying an existing one, or you can start from scratch.

Demonstrate how to create a user in the system. Draw the participants' attention to the mandatory fields that must be maintained for a user to be created.

To create a user master record from scratch, proceed as follows:

1. In the *User* field, enter the user name that the person will use to identify themselves in the SAP system. The user name has a maximum length of 12 characters. This user name is used to create the user master record. Do not fill in the *Alias* field, because this field is used to select user master records.

2. Choose *Create*.

3. You must maintain at least the following fields: the last name on the *Address* tab page, and, on the *Logon data* tab page, the initial password and the repeat password (must be identical).

4. Choose *Save*.

Selected tab pages in transaction SU01 are described in detail below:

**Logon data**: The *Wizard* pushbutton generates a password based on the security guidelines and enters it in the password fields. You do not need to enter the password again in the Repeat password field. For security reasons, you should avoid using the same initial password for different users, otherwise anyone could log on to the SAP system using this common initial password and a newly created user.

The *Deactivate* pushbutton deactivates the user’s password. The user can no longer use a password to log on to the SAP system. It is now only possible for the user to log on using Single Sign-On. If you want to reactivate the logon procedure with password, then you need to assign a new password to the user.

The *Deactivate* pushbutton does not lock the user master record itself, but rather only logon to the system with a password. Despite the deactivation of the password, the system continues to execute the background jobs released for this user. The user master record is locked, on the other hand, using transaction SU01, pushbutton *Lock/Unlock*. The system no longer executes background jobs for a user locked in this way.

Passwords can contain any characters, including numbers, spaces and punctuation. No distinction is drawn between upper and lower case. Passwords must comply with the following rules: they

- May not start with three identical characters
- Must consist of at least three characters
- May not start with a question mark, “?”, an exclamation mark, “!”, or a space
- May not be “PASS” or “SAP*”

When users log on for the first time, they must enter a new password. If a user later changes the password, then he or she may not change it back to any of his or her last five passwords.

User names that are intended for people to use to log on to the SAP system must be assigned the user type “Dialog” on the *Logon data* tab page. Only users of type “Dialog” can log on to the SAP system using SAP GUI.

The input field *User group for authorization check* on the *Logon data* tab page defines who is allowed to maintain this user. If no data is entered in this field, then all user administrators can maintain the user. Enter the group **SUPER** for users with critical authorizations (such as user administrators or system administrators). The effect of this entry is that only users who themselves belong to this group can change the data for this user.

This instructor note contains additional information that exceeds the normal scope of this lesson. It is only included to enable the instructor to answer questions from participants.

The field *Alias* on the initial screen of user maintenance is primarily used in this context to find internal technical user names when you only know the Internet user name. You cannot use an alias to log on to an SAP system using either SAP GUI or RFC.

If you need to create a user with an alias, assign the alias name in the input field on the *Logon data* tab page.

The alias name is then used for Internet transactions. If users log on to the Internet via the Internet Transaction Server, then they log on using the user name known to the original system. Users must enter the alias name with the corresponding password to identify themselves in Internet transactions (for example, when ordering items). If the user has forgotten his or her alias name, then he or she can set up another account for themselves on the Internet. A new user with the corresponding alias is then created in the SAP system. The 12-character user name is generated automatically based on an algorithm.

In some Internet scenarios you can use an SU01 user instead of an SU05 user if the *Alias* field and the *References* pushbutton have been maintained for the SU01 user. You can find further information on this topic in the SAP Library.

**Groups**: Along with the user groups already mentioned in conjunction with the *Logon data* tab page, there is also a *Groups* tab page. This tab page enables you to assign a user to one or more “general user groups”. These general user groups are primarily used to group users for mass maintenance (transaction SU10).

**Defaults**: you can set the logon language for the user on the *Defaults* tab page. The user then does not need to fill in the language field when logging on to the SAP system. You can, however, always enter a different language on the logon screen if you need to. The same tab page also enables you to make settings for printing from the SAP system. Set the name of the standard printer in the field *OutputDevice*. If you want to output the spool request immediately or delete the spool request when it has been output successfully, check the corresponding fields. These settings are predefined for users. However, users can override the values predefined here when printing, if required.

**Parameters**: you maintain user-specific values for fields in the SAP system on the *Parameters* tab page. This makes the system more user-friendly. User parameters enable you to set default values for fields in which the user usually or always enters the same value. The user can always overwrite these default values as required. The parameter IDs for the fields are displayed under Technical Information in the F1 help for each field.

**Roles** and **Profiles**: these tab pages enable you to control which entries the user menu contains and which authorizations are assigned to the user. You can see which roles have been assigned to the user on the *Roles* tab page. Here you can assign further roles to the user using the values help for the corresponding input field and confirm your entries by choosing "Enter". This normally creates an entry on the *Profiles* tab page that contains the name of the role in the text field. The profiles contain the user's authorizations. If an entry for a role is missing on the *Profiles* tab page, then the user will see the entries for that role in his or her user menu, although he or she will not be authorized to execute that role. SAP provides predefined templates for roles; the names of these roles all have the prefix "SAP_".

We recommend that you only assign users to roles that do not start with the prefix "SAP_". You can only use the reference user on the *Roles* tab page to assign additional authorizations. This does not create entries in the user menu.

For more information about the input fields, see the SAP Library under *mySAP Technology Components → SAP Web Application Server → Security (BC-SEC) → Users and Roles (BC-SEC-USR).*

The initial screen of transaction SU01 also contains the following pushbuttons:

**Copy user**: the *Copy* pushbutton allows you to select data that you want to copy from the user master record. Creating a user assigned to standard roles as a copy template makes it easier to create new users efficiently.

**Change password** and **Lock/Unlock** user: if you lock or unlock a user, this change takes effect when that user next logs on to the system; if the user is already logged on when you make the change, then he or she is not affected for the duration of that logon session. The SAP system automatically locks users when 12 consecutive logon attempts have failed. The system removes this automatic lock on the user name at midnight. You can manually unlock a user before the lock period expires. If you set a lock manually, then it is effective until you remove it again.

If you set a new password for a user, then it becomes effective immediately. If a user has forgotten his or her password, then he or she can use the new password you set as soon as you make the change. Users can change their passwords once a day at most. A user administrator, on the other hand, can change user passwords as often as required.

Users can maintain their own data using *System → User Profile → Own Data* (transaction SU3). They can maintain their own data on the *Defaults*, *Address* and *Parameters* tab pages. Users cannot, however, maintain the company address on the *Address* tab page.

## Assigning Users to Roles with SAP Easy Access

The SAP Easy Access menu contains an option for assigning one role to multiple users. SAP provides predefined templates for roles; the names of these roles all have the prefix "SAP_".

Choose the *Other menu* pushbutton on the SAP Easy Access screen. A dialog box containing all roles in the SAP system is displayed. Select a role to display it. If the role selected corresponds to an employee's field of activities, choose *Assign users*. A dialog box appears in which you can enter user IDs and user names as required. You are then asked whether an authorization profile should be generated or not. If you confirm that it should be, then the authorizations that correspond to the role are assigned to the user. You can check this on the *Profiles* tab page of the user master record.

If you are asked whether a profile should be generated, please contact your system administrator.

Choose *Documentation* to display additional information on the role selected.

## Central User Administration

If you have an SAP system group consisting of several SAP systems with several clients, and the same users often need to be created and maintained in each client, then you can use Central User Administration. Central User Administration enables you to maintain all users centrally in one client of one of the systems; this client is then known as the central system. The information is then automatically distributed to the remaining clients in the various SAP systems.

You use the same user maintenance transaction (SU01) to maintain users in Central User Administration.

All user master records are created in the central system. Central User Administration also contains settings that define which fields can only be maintained in the central system, and which fields can also be maintained in local clients. You can also make settings that ensure that data from the local clients is transferred back to the central system. The availability of these settings depends on the individual fields.

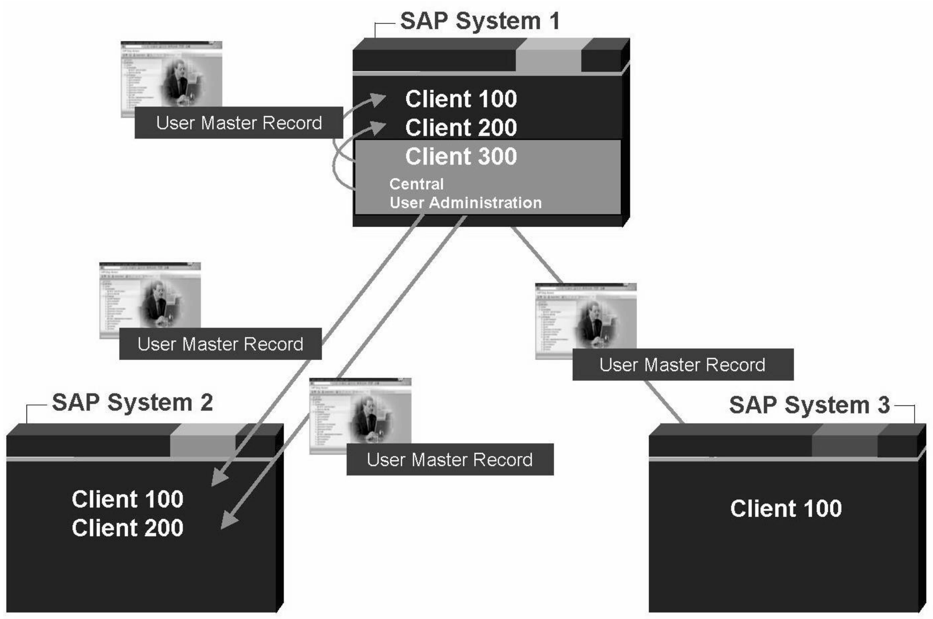


**Figure 69: Central User Administration**

## Accessing Directory Services using LDAP

Directory services allow various applications in an IT landscape to access shared information at a central location. The information is stored on a central directory server that the various systems of your IT landscape can access. The directory server acts as an "IT address book" for information that is normally used by several systems, such as personnel data (name, department, organization), user data and information on system resources and system services. You can use directory services to maintain information in SAP systems for directory-compatible applications (such as user administration or SAPoffice). The standardized Lightweight Directory Access Protocol (LDAP) is usually used as the access protocol.

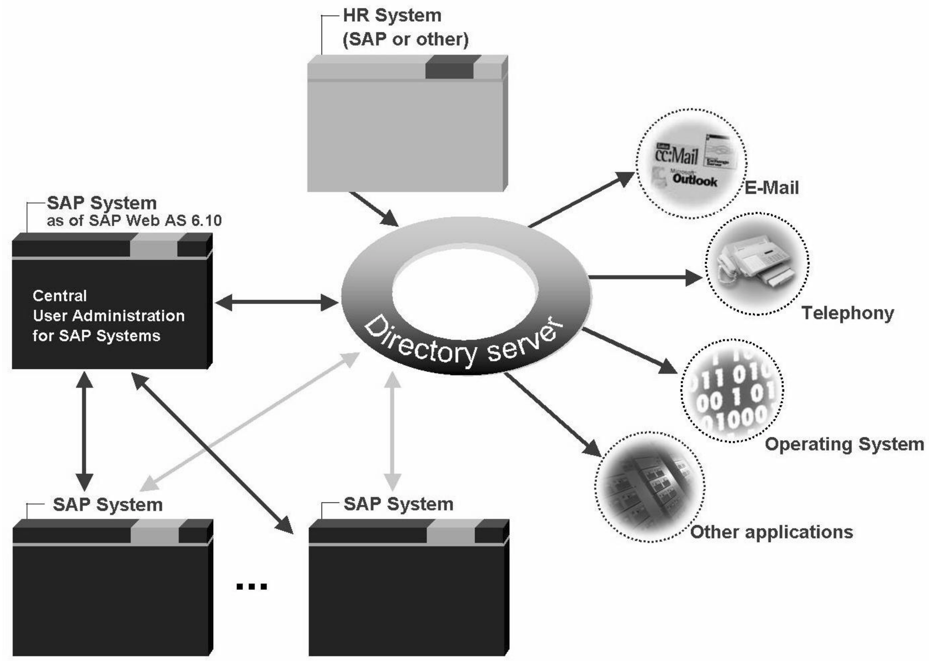


**Figure 70: Connection to Directory Services**

Directory services provide a central information and administration point and therefore simple shared information usage between various applications. You specify the synchronization direction for each field, that is, whether the SAP system overwrites the data in the directory, or the directory overwrites the data in the SAP system.

The RFC-LDAP interface has been available since SAP R/3 4.0. The LDAP Connector has been available since SAP R/3 4.6 to make this interface easier to use. Setting up LDAP connections has been greatly simplified again as of SAP Web AS 6.10. For more detailed information about LDAP, see the online documentation, under *mySAP Technology Components → SAP Web AS → Security → Directory Services (BC-SEC-DIR)*.

Data synchronization works with time stamps. These allow you to restrict the data volume. These time stamps enable you to specify that only data whose time stamp has changed need to be synchronized. You do not need to change the direction of synchronization to use the most up-to-date data.

**Note:** As of SAP Web AS 6.10, you can easily connect SAP systems to a directory service. Before SAP Web AS 6.10, while it was certainly still possible to create a connection, it was significantly more difficult to do so.

**Hint:** You can extend a Central User Administration using a connection to a directory service. **The two concepts** are certainly not mutually exclusive, but rather **work very well together**.

The SAP system can exchange data with directory services from various vendors. The SAP system may require attributes that are not in the standard schemata of the directories. SAP usually provides a schema extension for this purpose.

Since the schemata of directories and other sources of data are not identically structured, you need to define which SAP data fields correspond to which directory attributes. This does not need to be a 1:1 definition, since one field can be mapped to several attributes. The attributes assigned to the fields must also be present in the directory. If not, you need to enhance the schema in the directory.

Example: the SAP data field *FULLNAME* consists of the attributes "givenName" and "sn". The situation is reversed for telephone numbers. In the SAP system database, they consist of two fields, *Telephone Nr.* and *Telephone Ext.*, while the LDAP directory only contains the attribute *telephoneNumber*.

If you are using SAP systems as of SAP R/3 4.5B, you can connect these with the Central User Administration. Personnel data such as name, department, organization can be made available from HR systems (such as SAP) to other SAP systems through a directory server.

# 243 Exercise 17: User Administration

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Create user master records.

## Business Example

New employees require user names that enable them to authenticate themselves when logging on to an SAP system.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task 1 Creating User Master Records

Create a new user master record and copy an existing user master record.

1. **Creating a new user**: create a user with the name Admin-##, where ## corresponds to your group number.

   Call the transaction for user maintenance. Use Admin-## as the user name. Choose *Create*. Enter a last name and set an initial password. Make a note of the initial password. Save the user master record.

2. **Copying a user**: create a user with the name TAdmin-## (## again corresponds to your group number) by copying the user "TAdmin".

   Call the transaction for user maintenance. Enter the name "TAdmin" in the *User* field. TAdmin is the template user. Now choose *Copy*. In the dialog box that appears, enter the name TAdmin-## in the *To* field. Select all the check boxes apart from *Address*. Enter a last name and initial password. Make a note of the initial password. Save the user master record.

## Task 2 Testing Users

Test the users you have just created.

*Continued on next page*

1. **Logging on with user Admin-##:**

   Log on to your SAP system with the user you have created, “Admin-##”. Check the entries in the user menu. Choose transaction SU01.

2. **Logging on with TAdmin-##:**

   Log on to your SAP system with the user you have created, “TAdmin-##”. Check the entries in the user menu. Choose transaction SU01.

# Solution 17: User Administration

## Task 1 Creating User Master Records

Create a new user master record and copy an existing user master record.

1. **Creating a new user**: create a user with the name Admin-##, where ## corresponds to your group number.

   Call the transaction for user maintenance. Use Admin-## as the user name. Choose *Create*. Enter a last name and set an initial password. Make a note of the initial password. Save the user master record.

   a) Call transaction SU01 for user maintenance. You can also call this transaction using *Tools* → *Administration* → *User Maintenance* → *Users*. Enter "Admin-##" in the *User* field, where ## corresponds to your group number. Choose *Create*. Enter a name in the *Last name* field on the *Address* tab page. On the *Logon data* tab page, enter an identical password in the fields *Initial password* and *Repeat password*. Make a note of the initial password. Choose *Save*.

2. **Copying a user**: create a user with the name TAdmin-## (## again corresponds to your group number) by copying the user "TAdmin".

   Call the transaction for user maintenance. Enter the name "TAdmin" in the *User* field. TAdmin is the template user. Now choose *Copy*. In the dialog box that appears, enter the name TAdmin-## in the *To* field. Select all the check boxes apart from *Address*. Enter a last name and initial password. Make a note of the initial password. Save the user master record.

   a) Call transaction SU01 for user maintenance. Enter the name "TAdmin" in the *User* field. TAdmin is the template user. Now choose *Copy*. In the dialog box, enter the name "TAdmin-##" in the *To* field, where ## corresponds to your group number. Select all the check boxes apart from *Address*. On the *Logon data* tab page, enter an identical password in the fields *Initial password* and *Repeat password*. Make a note of the initial password. Enter a name in the *Last name* field on the *Address* tab page. Choose *Save*.

## Task 2 Testing Users

Test the users you have just created.

1. **Logging on with user Admin-##:**

*Continued on next page*

Log on to your SAP system with the user you have created, “Admin-##”. Check the entries in the user menu. Choose transaction SU01.

a) Log on to your SAP system with the user “Admin-##” and the password that you made a note of for this user. Enter a new password. You **cannot** access **any** user menu. You are **not authorized** to call transaction SU01.

2. **Logging on with TAdmin-##:**

Log on to your SAP system with the user you have created, “TAdmin-##”. Check the entries in the user menu. Choose transaction SU01.

a) Log on to your SAP system with the user “TAdmin-##” and the password that you made a note of for this user. Enter a new password. You can access your user menu. You are authorized to call transaction SU01.

## Lesson Summary

You should now be able to:

- Outline the basic principles of the authorization concept
- Create and copy users
- Describe the concept of Central User Management
- Explain how user management concepts utilize directory services with the LDAP protocol

### Related Information

You can find additional information on the topic of user management

- In the SAP Library
- In more advanced training courses, such as the course **CA940** (as of the end of 2003, **ADM940** - *SAP Authorization Concept* )

# Lesson: Daily Tasks in System Management

248 Lesson Duration: 25 minutes

## Lesson Overview

System administrators use certain system functions very often. This lesson gives you an overview of the most important of these functions. The SAP Solution Manager and the Computing Center Management System (CCMS), which offer a large number of important administration functions, play especially important roles here.

## Lesson Objectives

After completing this lesson, you will be able to:

- Name frequently-used administration functions in SAP systems
- Explain the concept of the SAP Solution Manager
- Name the central functions of the Computing Center Management System (CCMS)

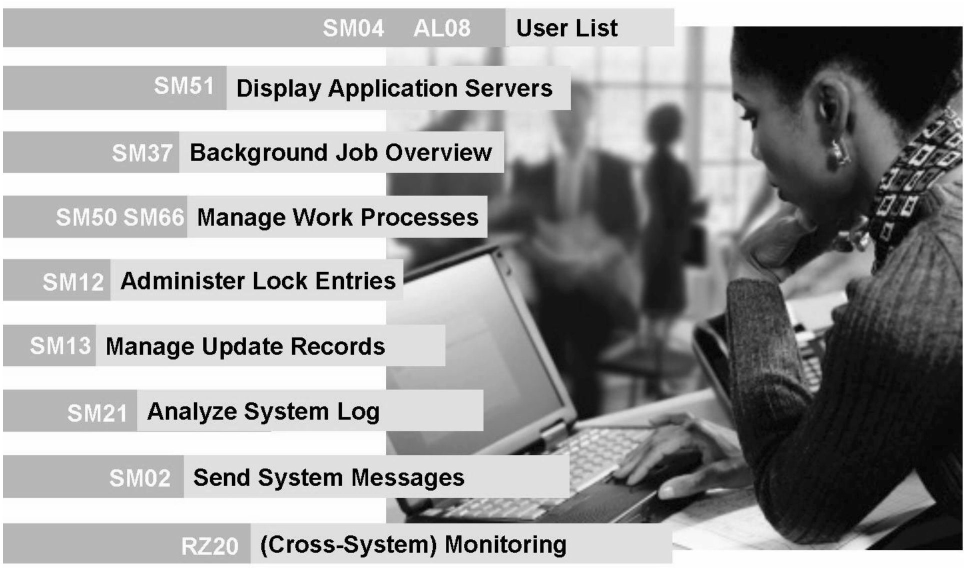
This lesson aims to give the participants an overview of the most important administration functions in SAP systems. The participants will already be familiar with a number of the transactions listed (see "Lesson Prerequisites"), which are therefore not covered in detail. This lesson focuses on the system log and transaction RZ20. The concept of the SAP Solution Manager is also introduced.

## Business Example

You need to monitor and control the system in a technical context.

## Overview of Central Administration Functions

SAP systems provide various functions for monitoring and configuring the technical status of the system. System administrators use a range of transactions on a daily basis; these transactions are named in the graphic "Important system administration transactions".



SM04 AL08 User List

SM51 Display Application Servers

SM37 Background Job Overview

SM50 SM66 Manage Work Processes

SM12 Administer Lock Entries

SM13 Manage Update Records

SM21 Analyze System Log

SM02 Send System Messages

RZ20 (Cross-System) Monitoring

**Figure 71: Important system administration transactions**

**SM37**: this transaction provides an overview of background jobs that are either planned or have already been executed in the system. You can select these jobs in a variety of ways, for example, using the name of an ABAP program that runs in one of the job steps. Note that you can use the *Change layout...* pushbutton to adapt the quantity of information displayed to your own requirements.

**SM51**: transaction SM51 displays all instances that are currently active in your system. You can not only access numerous other transactions from here, but also execute them for an application server you have selected from the display. *Goto* → *Server information* → *Queue Information* displays a simple and informative overview of the current status (particularly the available capacity) of request queues managed by this instance.

**SM04** and **AL08**: the user overview (SM04) displays all users logged on to the local instance. Various administration functions are available here, for example, logging off a user, or displaying an overview of main memory resources requested by individual sessions. The global user overview (AL08) on the other hand does not provide any editing functions, but simply displays a list of all users logged on to the system, sorted by instance.

**SM50** and **SM66**: the scope of the functions provided in these two transactions is very similar, the global work process overview (SM66) generally provides at least the same editing functions as the local work process overview (SM50). The local work process overview (SM50) provides comprehensive information on the work processes configured on an instance, such as their current status, the amount of CPU time required since they started and administration functions (restarting a work process, trace functions). Work processes should always be configured to restart automatically: value "Yes" in *Start* column. The F1 help provides further information on various fields in the overview.

**SM12**: this transaction enables you to view and manage entries in the lock table. The lock table is normally managed by the enqueue work process. However, in exceptional cases, the administrator may need to intervene. SM12 displays all entries in the lock table, with a flag in the appropriate column to denote shared locks. Use the *Details* pushbutton to display additional information, such as the transaction that set the lock. You should only delete locks manually in SM12 in exceptional circumstances and after thorough checks. This check should ensure that the lock to be deleted is really not used any longer. You can carry out these checks using transactions SM50 and SM04. Locks owned by update work processes are displayed in blue.

Up to SAP R/3 4.6C (inclusive), a shared lock is set on table *ADRC* when a user maintains his or her own profile (SU3 or also with SU01). This shared lock no longer exists as of SAP Web AS 6.10.

**SM13**: this transaction enables you to ensure that the update process runs smoothly. If problems occur during an update, they are logged in this transaction and you can analyze them here. If you use this transaction to view data to be updated (such as a salary raise), then this action is logged in the system log. In transaction SM13, you have various options for switching to update administration (SM14). SM13 also enables you to reprocess update records if the process was not initially successful. You should not use this option for update records of type "V1". For more information, see the SAP Library: *mySAP Technology Components → SAP Web Application Server → Client/Server Technology → Updates in the SAP System (BC-CST-UP).*

**SM21**: The system log (also called SysLog) enables you to evaluate system messages that are recorded in a log file. A distinction is made between messages, warnings, and errors. Entries are correspondingly color-coded. The SysLog allows you to identify the user that triggered the problem, the transaction they were using, and other details. Always use the SysLog when you need to trace error situations that occurred in the past. The size of the SysLog is set by the parameter `rslg/max_diskspace/local`. Each log entry takes up 192 bytes, and the default log size is 500160 bytes, which is a multiple of 192 that corresponds to 2605 entries. Once the log is full, the oldest entries are overwritten. Every instance has its own SysLog. If you are using Unix, you have the option of creating a central SysLog.

**SM02**: system messages enable you to pass on important system-related (or general) news to the people affected. System messages are displayed for each user the next time they interact with the system, on condition that the message's expiry date has not yet been reached. When creating system messages, you can not only specify an expiry date, but a client or server. This means that these messages will either be displayed for all users in that client or on that server, or for all users in the whole system.

**RZ20**: this transaction is discussed in detail in the next section of this lesson.

## The Concept of the SAP Solution Manager

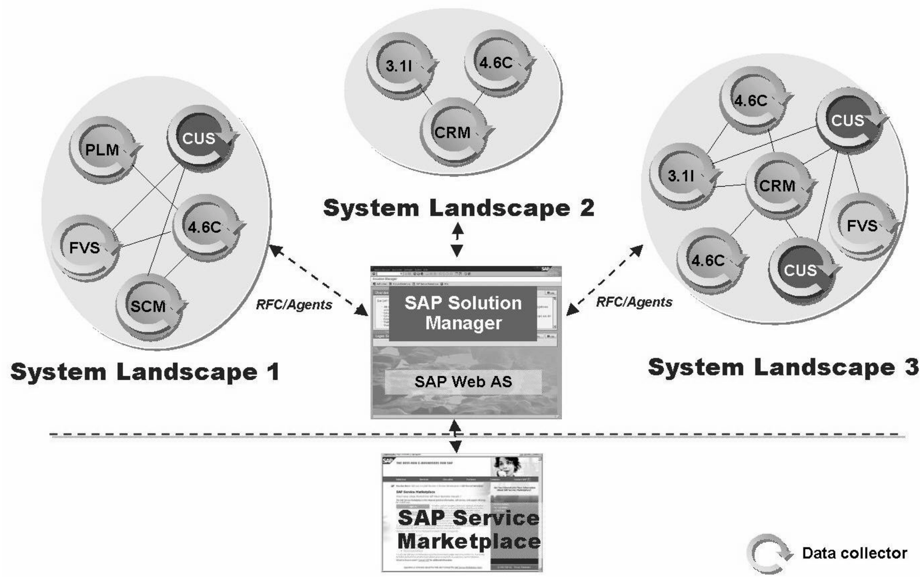


**Figure 72: Monitoring multiple system landscapes using the SAP Solution Manager**

The SAP Solution Manager is a powerful tool that can provide you with extensive support with the following tasks:

- Simplification of the implementation and integration of new SAP solutions in an existing system landscape
- Central documentation of the components of the system landscape
- Monitoring the running operation of all SAP components
- Monitoring cross-system business processes
- Constant optimization and checking of the performance of SAP systems using *Early Watch Alerts* and *Best Practices*
- Setting up a company-internal support infrastructure using SAP CRM resources (as of SAP Solution Manager 3.1)

- Accessing special functions of the SAP Service Marketplace

For more information about the SAP Solution Manager, see the information at the following address: http://service.sap.com/solutionmanager.

The following courses provide basic information about working with the SAP Solution Manager:

**TSLM10** - *SAP Solution Manager Infrastructure and Installation*

**SMI310** - *Solution Manager Implementation Tools*

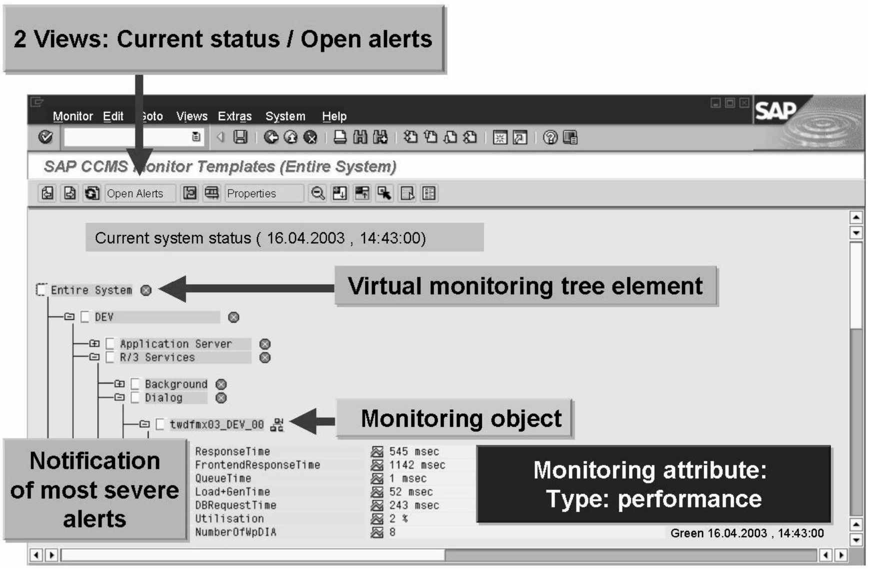
You can also subscribe to a newsletter on the SAP Solution Manager homepage mentioned above, which will ensure that you are always up to date with news about the SAP Solution Manager.

## Possibilities of the Computing Center Management System (CCMS)

The Computing Center Management System (**CCMS**) provides a number of important administration functions:

- System administration (start, stop, reconfigure the system)
- Background processing and job scheduling
- Printer landscape configuration
- System tuning
- Database administration (including backups)
- System profile management
- Dynamic load balancing
- System monitoring

System administrators use many of these functions on a daily basis. RZ20 (alert monitor) provides a central point of entry for system monitoring. The graphic “System monitoring with transaction RZ20” shows a typical alert monitor screen plus a few explanatory notes.


2 Views: Current status / Open alerts
Monitor Edit Goto Views Extras System Help
Open Alerts
Properties
Virtual monitoring tree element
Monitoring object
Notification of most severe alerts
Monitoring attribute: Type: performance

SAP CCMS Monitor Templates (Entire System)

Current system status ( 16.04.2003 , 14:43:00)

```
Entire System
  DEV
    Application Server
    R/3 Services
      Background
      Dialog
        twdfmx03_DEV_00
ResponseTime            545 msec
FrontendResponseTime    1142 msec
QueueTime               1 msec
Load+GenTime            52 msec
DBRequestTime           243 msec
Utilisation             2 %
NumberOfWpDIA           8
```

Green 16.04.2003 , 14:43:00

Figure 73: System monitoring with transaction RZ20

At this point you should use a simple default monitor to demonstrate how to display the average dialog response time on the instance you are using. You could, for example, use the “Dialog Overview” monitor from the “SAP CCMS Monitor Templates” collection for this demonstration. Since one of the exercises is based on this demonstration, you should carry it out slowly and clearly.

Transaction RZ20 provides a system monitoring infrastructure that enables you to monitor many system parameters centrally. It also provides links to analysis options in other transactions, for example, the functions SM50, SM04, and so on, listed above. Open interfaces allow you to integrate additional non-SAP system monitoring tools.

You can not only monitor several systems using an RFC connection, but you can also monitor systems with widely varying releases, right back to SAP R/3 3.1. You can also create your own views of the system(s) that only display the parameters you need for a specific user group. These individual views are also called “monitors”.

You can monitor the values displayed for the attributes you are monitoring using easy-to-set threshold values. If a monitored value exceeds or falls below the specified threshold, then it can be displayed as a warning (yellow) or as a problem (red). One of the particular features provided by the alert monitor is automated reactions to specific problem situations. If an automatic reaction has been configured for an attribute, and that attribute subsequently exceeds (or falls below) the threshold specified, then a predefined reaction is triggered. There is almost no limit to the degree of complexity of the triggered reaction. Anything that can be achieved through background processing or programming can be used as a reaction to an error situation. For example, if a severe problem occurs, this could trigger an SMS to the administrator's cell phone. The graphic "System monitoring with transaction RZ20" shows, among other things, a typical monitoring attribute, specifically the average dialog response time (over the last 15 minutes). Since it does not exceed the threshold specified for a warning, it is colored green.

One major feature of the alert monitor is the centralized representation of the system status. As you have learned in this lesson, there is a whole range of important transactions, but it is their integration in the central monitoring architecture that makes them significantly easier to use and more efficient, even across system boundaries.

The alert monitor is an extremely powerful and multifaceted tool for monitoring system status. It is configured using transaction RZ21 (Monitoring: Properties and Methods).

# 255 Exercise 18: Daily Tasks in System Management

Exercise Duration: 15 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Use individual administration functions

## Business Example

You learn more about individual selected administration functions.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task 1 User Overview

Find out how many users are working on the same instance as you.

1. Use the appropriate transaction to determine how many users are logged on to the same instance as you.

## Task 2 Lock Overview

Gain an overview of current locks in the system.

1. Call the transaction for changing your user profile (transaction SU3). Then, while you are still in that transaction, display an overview of current locks in the system in a parallel session.

## Task 3 Request Queues

Find out the maximum number of requests waiting in the dialog request queue on your instance.

1. In the request queue display, choose *Refresh* several times.

*Continued on next page*

## Task 4 Average Dialog Response Time

Find out the average dialog response time.

1. Use transaction RZ20 and a monitor template to establish the average dialog response time over the last 15 minutes.

# Solution 18: Daily Tasks in System Management

## Task 1 User Overview

Find out how many users are working on the same instance as you.

1. Use the appropriate transaction to determine how many users are logged on to the same instance as you.

   a) Transaction SM04 contains the answer to this question.

## Task 2 Lock Overview

Gain an overview of current locks in the system.

1. Call the transaction for changing your user profile (transaction SU3). Then, while you are still in that transaction, display an overview of current locks in the system in a parallel session.

   a) Transaction SM12 displays an overview of current locks in the enqueue work process.

## Task 3 Request Queues

Find out the maximum number of requests waiting in the dialog request queue on your instance.

1. In the request queue display, choose *Refresh* several times.

   a) You can find the information you need in transaction SM51 under *Goto* → *Server information* → *Queue Information*. Note that this display allows you to draw interesting conclusions about the system load.

## Task 4 Average Dialog Response Time

Find out the average dialog response time.

1. Use transaction RZ20 and a monitor template to establish the average dialog response time over the last 15 minutes.

   a) Start transaction RZ20 and then choose the "Dialog Overview" template from the "SAP CCMS Monitor Templates". Note the time specified under "Dialog Response Time".

## Lesson Summary

You should now be able to:

- Name frequently-used administration functions in SAP systems
- Explain the concept of the SAP Solution Manager
- Name the central functions of the Computing Center Management System (CCMS)

### Related Information

For more information about this topic, see the SAP Library under BC (System Services) and the training course ADM106 “Advanced SAP System Monitoring”.

# Lesson: SAP Service Marketplace

258

Lesson Duration: 20 minutes

## Lesson Overview

This lesson introduces you to the central point of entry for all SAP services, the SAP Service Marketplace. A few services that are particularly relevant to mySAP Business Suite are presented as examples.

## Lesson Objectives

After completing this lesson, you will be able to:

- Use the SAP Service Marketplace
- Name some of the services provided on the SAP Service Marketplace

Don't get bogged down in the details. The focal point is that the participants need to know that the SAP Service Marketplace is **the** portal for services provided by SAP (and SAP partners). Every customer should know the SAP Service Marketplace URL by heart. An SAP Service Marketplace user is a vital survival aid for every SAP customer (though you can have more than one).

## Business Example

The system administrator has established that there is a problem in an SAP system and is looking for a solution.

## SAP Service Marketplace Target Group

You can find information on all SAP solutions and on SAP as a company at http://www.sap.com. This website is open to the general public. The SAP Service Marketplace on the other hand, at http://service.sap.com, is directed specifically at SAP's **customers and partners**. The SAP Service Marketplace enables you to access various services, special information and additional offers.

## Logging on, Personalizing, and Navigating

To log on, you must be **registered as a SAP Service Marketplace user** (formerly known as OSS user or SAPNet user). There is normally a contact person in each company who creates users for employees as required and sets authorizations. Access to the SAP Service Marketplace is **free of charge** to customers (apart from the Internet access required).

Once you have logged on to the SAP Service Marketplace, you have a range of topics to choose from. You can personalize your homepage, that is, set up your own pages with topics that are of particular interest to you.

**Quick Links** enable you to quickly access specific areas of the SAP Service Marketplace. To call up a Quick Link, simply add it to the Web address, preceded by a "/" (for example: http://service.sap.com/ui). After logging on to the SAP Service Marketplace, you can access a selection of Quick Links.

Log on to the SAP Service Marketplace using a valid user (such as your own user). Warning: internal (D or I) users can see content in some sections that is not accessible to customers.

Demonstrate how to call up a Quick Link. Demonstrate also how to navigate with the Quick Link appended to the URL. The list under "Quick Links" (unfortunately) does not contain all available Quick Links! The "Search for Content Area" only searches by Quick Links and their descriptions (status: September 2001). What is now known as a "Quick Link on the SAP Service Marketplace" used to be called an "alias in SAPNet".

## Important Services Provided on the SAP Service Marketplace

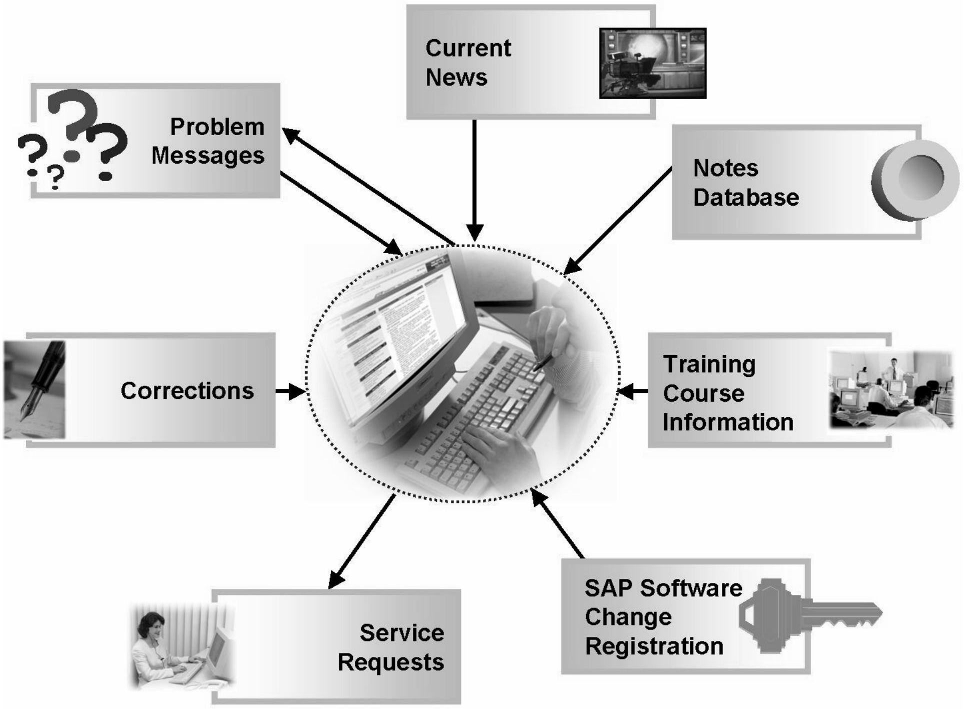


**Figure 74: service.sap.com – the SAP Service Marketplace**

Some SAP Service Marketplace functions are named below. The focus here is on services that are particularly relevant to SAP Web AS.

- As an SAP Service Marketplace user with the appropriate authorizations, you can request and administer other **SAP Service Marketplace users** at */user-admin*. SAP Service Marketplace users are also subject to an authorization concept. You can, for example, withhold authorization to create SAP messages (see below).
- A developer needs a **developer key** to be able to create or change objects in an SAP system. You can request this key using the Quick Link */sscr* (which stands for SAP Software Change Registration).

  Changes to SAP objects (such as ABAP programs) are called modifications and require an **object key**, which you can also request at */sscr*.
- You can access the comprehensive **Notes Database** using the Quick Link */notes*. You can access specific SAP Notes either by entering the Note number directly, or using search criteria (such as the name of a transaction or an error code).

- All SAP systems as of the SAP Web Application Server 6.10 include the **Note Assistant**. This is a tool (transaction SNOTE) that can automatically import corrections to ABAP code from SAP Notes into SAP systems. */noteassistant* provides information on how to implement the Note Assistant in SAP systems with earlier releases, as well as other information on the tool.
- SAP regularly provides **corrections** for known problems (for example, as Support Packages, kernel patches, or front-end patches). You can download them at */patches*. SAP recommends that you keep your correction status as up-to-date as possible.
- It may happen that an error occurs for which there is no relevant SAP Note or correction available. If this does happen, you can enter a **message** to SAP under */message*. This message is then processed by SAP Support.
- If an SAP employee needs **access to your SAP system** to work on a problem, then you can release the connection using */serviceconnection*.
- Other services, such as **Remote Consulting** (*/remoteconsulting*: SAP consulting services without consultant site visits) or **EarlyWatch** (*/earlywatch*: SAP experts proactively analyze your SAP systems to ensure optimum performance and availability), also require this type of access. You can find and order services provided by SAP in the **Service Catalog** (Quick Link */servicecat*).
- Are you looking for information on **software requirements** (operating system or database releases) for a specific SAP solution? You can find the answer using the Quick Link */platforms*.
- You can find information on the latest developments in the **SAP NetWeaver** area using */netweaver*.
- For information about **SAP Web AS**, use the quick link */webas*
- Quick link */enterprise* points to news about **SAP R/3 Enterprise**
- The **Quick Sizer** (*/quicksizer*) is a tool that helps you to plan your hardware investment. Your expected hardware requirements (such as disk space and main memory) are determined on the basis of your load profile. Standardized SAP benchmarks carried out by SAP's hardware partners at regular intervals on their servers help you to size your hardware correctly.
- If you need to get in touch with an **SAP partner**, you can find their contact details in the Partner Directory at */partnerdir*.
- */education* provides details of SAP's current **Education Services**. You can find out which courses in your country still have free places, or even book them online.

# 261 Exercise 19: SAP Service Marketplace

Exercise Duration: 10 minutes

## Exercise Objectives

After completing this exercise, you will be able to:

- Log on to the SAP Service Marketplace and call some functions.

## Business Example

Your system administrator needs some answers on specific issues.

### System Data

| | |
|---|---|
| **System:** | The training system assigned to you (such as I30) |
| **Client:** | The client assigned to you (such as 802) |
| **User ID:** | The user name that you have given the participants (such as SAPTEC-##) |
| **Password:** | The password that you have given the participants for the relevant user. |
| **Set up instructions:** | None |

## Task 1

Log on to the SAP Service Marketplace:

1. Start a Web browser.
2. Go to the SAP Service Marketplace.
3. Log on to the SAP Service Marketplace. If you do not have your own S-user, use one specified by your instructor.

If you do not know any general SAP Service Marketplace users, then, before you teach this lesson, check whether the user used by the participants for the online evaluation gives access to SAP Notes on the SAP Service Marketplace.

## Task 2

Display specific information:

1. Display SAP Note 81069. What is this SAP Note about?
2. With which databases can you use the SAP Web Application Server 6.20 kernel?

# Solution 19: SAP Service Marketplace

## Task 1

Log on to the SAP Service Marketplace:

1. Start a Web browser.

   a) You can normally access the Web browser (for example, the Microsoft Internet Explorer) as an icon on your desktop or using an entry in the *Start* menu.

2. Go to the SAP Service Marketplace.

   a) Enter the URL of the SAP Service Marketplace, http://service.sap.com, in the address field of your Web Browsers.

3. Log on to the SAP Service Marketplace. If you do not have your own S-user, use one specified by your instructor.

If you do not know any general SAP Service Marketplace users, then, before you teach this lesson, check whether the user used by the participants for the online evaluation gives access to SAP Notes on the SAP Service Marketplace.

   a) Choose the link to log on. Your instructor will specify a user, which is also used to access the online evaluation form.

## Task 2

Display specific information:

1. Display SAP Note 81069. What is this SAP Note about?

   a) You may find a link to SAP Notes on the homepage. Otherwise you can use the */notes* Quick Link:

      - By suffixing `notes` to the address line `https://...sap.../` in your browser or
      - By choosing the entry "Quick Links" in the header bar and selecting NOTES from the list

   b) Enter the Note number `81069` and start the search for the SAP Note.

   c) Display the whole SAP Note by selecting the title. SAP Note 81069 contains information on the Web browser settings required to access the SAP Service Marketplace.

*Continued on next page*

2. With which databases can you use the SAP Web Application Server 6.20 kernel?

   a) There are various ways of answering questions about platform availability. Two approaches are outlined below:

      - Enter the search criteria **`Kernel`** to restrict the Notes search (Quick Link */notes*). To further reduce the number of hits, enter the release **`620`** and the application area **`XX-SER-SWREL`**.
      - The other approach uses the Quick Link */platforms*. Navigate to the document on the availability of the SAP Web AS 6.20 using the link *Platform Availability for SAP Basis / SAP Kernel*.

      You can find details on the operating systems released for the 6.20 kernel in the following SAP Notes:

| SAP Note | Database |
|---|---|
| 407320 | SAP DB |
| 407322 | DB2 UDB |
| 407325 | DB2 on OS390 |
| 410783 | DB2 on AS400 |
| 407317 | Informix |
| 407328 | SQL Server |
| 407314 | Oracle |

## Lesson Summary

You should now be able to:

- Use the SAP Service Marketplace
- Name some of the services provided on the SAP Service Marketplace

### Related Information

You can find additional information on the SAP Service Marketplace using the SAP Service Marketplace URL: http://service.sap.com.

## Unit Summary

You should now be able to:

- Outline the basic principles of the authorization concept
- Create and copy users
- Describe the concept of Central User Management
- Explain how user management concepts utilize directory services with the LDAP protocol
- Name frequently-used administration functions in SAP systems
- Explain the concept of the SAP Solution Manager
- Name the central functions of the Computing Center Management System (CCMS)
- Use the SAP Service Marketplace
- Name some of the services provided on the SAP Service Marketplace

267

# Test Your Knowledge

1. Select user passwords that are valid in an SAP system:

   *Choose the correct answer(s).*

   □ A SAP*
   □ B PASS
   □ C 111
   □ D A§
   □ E mi45no
   □ F other4

2. If you are using a directory service with the LDAP protocol, then there is only one synchronization direction: the directory service overwrites data in the SAP system.

   *Determine whether this statement is true or false.*

   □ True
   □ False

3. Which of the following transaction codes can you use to view/edit user logons?

   *Choose the correct answer(s).*

   □ A SM04
   □ B SM30_USERS_SSM
   □ C USER
   □ D AL08
   □ E SM51

4. You can use transaction SM50 (Work Process Overview) to display the amount of CPU time used by individual work processes since the last time they were started.

   *Determine whether this statement is true or false.*

   □ True
   □ False

5. When the local SysLog file reaches its maximum size, it is closed and a new SysLog file is created. The file name, which is automatically generated, includes a time stamp.

   *Determine whether this statement is true or false.*

   □ True
   □ False

6. RZ20 offers the following options:

   *Choose the correct answer(s).*

   □ A Central monitoring of a large number of SAP systems
   □ B Simple definition of system-specific threshold values
   □ C Links to many standard administration functions
   □ D Monitoring of systems with releases prior to 4.0B
   □ E Definition of automatic reactions to specific system statuses
   □ F Simple restarts of entire SAP systems or individual instances

7. The SAP Service Marketplace is directed at which of the following?

   *Choose the correct answer(s).*

   □ A Everyone who is interested in SAP
   □ B SAP employees
   □ C SAP partners
   □ D SAP customers
   □ E SAP competitors

269

# Answers

1. Select user passwords that are valid in an SAP system:

   **Answer:** E, F

   The rules for passwords do not allow “SAP*” or “PASS”, two-character passwords or passwords beginning with three identical characters.

2. If you are using a directory service with the LDAP protocol, then there is only one synchronization direction: the directory service overwrites data in the SAP system.

   **Answer:** False

   You can set the direction of synchronization individually for each field: data in SAP system is overwritten or data in directory service is overwritten.

3. Which of the following transaction codes can you use to view/edit user logons?

   **Answer:** A, D

   You can use transactions SM04 and AL08 to edit or only view user sessions. The other options either do not exist or are not relevant to viewing user sessions.

4. You can use transaction SM50 (Work Process Overview) to display the amount of CPU time used by individual work processes since the last time they were started.

   **Answer:** True

   CPU time consumption is recorded for every work process and can be displayed in transaction SM50 using *CPU*. When a process is restarted, it is not only allocated a new process ID by the operating system, but the internal timer is also reset.

5. When the local SysLog file reaches its maximum size, it is closed and a new SysLog file is created. The file name, which is automatically generated, includes a time stamp.

   **Answer:** False

   When the local SysLog file reaches its maximum size (set by profile parameters), the oldest records are overwritten (cyclical overwriting).

6. RZ20 offers the following options:

   **Answer:** A, B, C, D, E

   The options listed are, with the exception of the last one, supported by transaction RZ20.

7. The SAP Service Marketplace is directed at which of the following?

   **Answer:** C, D

   The SAP Service Marketplace is directed at SAP customers and SAP partners.

# Course Summary

You should now be able to:

- List the solutions offered by SAP
- Navigate in the system
- Describe the system architecture using the appropriate terminology
- Describe the process for development work in the SAP environment
- Name interface technologies in the SAP environment
- Name central administrative functions of SAP systems

## Related Information

You can find additional information on the topics covered in this training course:

- In more advanced system administration courses
- In the SAP Library
- In the SAP Notes database
- On the SAP Service Marketplace
- On the SAP homepage

# Glossary

**ABAP**

Advanced Business Application Programming. SAP's own programming language for developing application programs.

**ABAP Dictionary**

Central redundancy-free information store in the SAP system for the logical structures of application development objects (such as tables, views, and data types) and for their representation in the structures of the relational database used.

**ABAP editor**

ABAP Workbench program editor. You can use the ABAP Editor to create, test and modify ABAP programs, function modules, screen flow logic and logical databases. You can enter source text and carry out syntax checks.

**ABAP processor**

Work process component that executes the processing logic of an ABAP program and communicates with the database interface.

**ABAP Workbench**

The SAP system's integrated graphical development environment. The ABAP Workbench supports the development, modification, testing and administration of client/server applications written in ABAP.

**access method**

The access method describes the connection between a spool work process and the operating system spool.

**Activation**

Process that makes a runtime object available. The effect of activation is to generate runtime objects, which are accessed by application programs and screen templates.

**ALE**

Application Link Enabling: a means of creating and operating distributed applications.

**background job**

A background job describes work steps that are executed without user interaction (in the background) in the system. Background jobs are handled by background work processes.

**BAPI**

A Business Application Programming Interface is a standardized programming interface that facilitates internal and external access to business processes and data in SAP systems.

**BOR**

The Business Object Repository gives you an overview of the business objects in an SAP system, and functions for managing them.

**Business Add-In**

The location in a program defined by the developer at which software recipient layers such as industries, partners and customers can insert addition code without modifying the original object.

**Business Server Page**

A Business Server Page (BSP) is a storage unit (HTML, XML) that may contain server-side scripting and that is developed in the SAP Web Application Server development environment, the Web Application Builder. A BSP is therefore part of a BSP application and contains the layout part, all event handlers, page attributes, and type definitions.

**central system**

A central SAP system consists of a single instance, which provides all necessary SAP services, and the accompanying database. Both the instance and the database will normally be hosted on the same computer.

**Change request**

An information source in the Transport Organizer that records and manages all alterations made to Repository objects and Customizing settings during a development project.

**client**

A client usually represents a company in an SAP system. This means that if an SAP system has several clients, then several companies can be represented and simultaneously active in that system. The client has a corresponding key field in the tables of the database for that SAP system. If you are logged on to a specific client, then you can only access data for that client. Clients therefore correspond to independent business entities.

**client (client/server)**

Software component that uses the services provided by a server (software-oriented view). Work station for which the server provides resources (hardware-oriented view).

**command field**

You can start applications directly by entering the transaction code in the command field.

**CPI-C**
Common Program Interface Communication describes the exchange of data between different programs. Data “packed” in CPI-C can be transferred using various technical protocols, such as TCP/IP or LU6.2.

**customizing**
Customizing is the overall procedure for setting up one or more SAP systems. This procedure is directed toward adapting the standard, industry-specific SAP system functions to a company’s particular business requirements. Customizing is obligatory both during the first installation and during an upgrade and is performed in the SAP system using the Implementation Guide (IMG).

**data element**
ABAP Dictionary object that describes the data type and semantic meaning of a table field or structure field.

**database interface**
Component of a work process that connects it to the database. The database interface translates Open SQL into database-specific SQL, thereby enabling communication with the database.

**database transaction**
Non-divisible sequence of database operations, that must be either executed completely, ending with a database commit, or not at all.

**development object**
Component of an ABAP application. Development objects include: programs such as transactions, reports, subroutines and function modules; program components such as event blocks, screens and menus; and objects used by several programs (for example, database fields, field definitions and program messages).

**DIAG**
Dynamic Information and Action Gateway: protocol for communication between SAP GUI and the application server (also known as the SAP GUI protocol).

**dispatcher**
Central process on an instance. It is responsible, among other things, for starting work processes and distributing transaction load across work processes.

**domain**
An ABAP Dictionary object that describes the technical attributes of a data element, such as data type, length and value range. You can group fields that have similar technical or business purposes under a single domain. All fields based on a domain are updated automatically when you change the domain. This guarantees the consistency of the fields.

**DX Workbench**
The Data Transfer Workbench provides a comprehensive range of functions for data transfer in SAP systems.

**EDI**
Electronic Data Interchange. The electronic exchange of structured data, such as business documents, between domestic and international companies using a variety of hardware, software and communication services. For this purpose, the data involved is formatted according to predefined standards. You can configure EDI using ALE.

**enqueue server**
The instance of an SAP system on which the enqueue work process administers the lock table in a distributed SAP system is also known as the enqueue server.

**F1 help**
You can use the F1 key to display an explanation of fields, menus, functions and system messages. The F1 help also displays technical information on the relevant field.

**F4 help**
The F4 help displays input help (possible values) for a field that is ready for input.

**favorite**
Favorites contain SAP system functions, or links to Internet content or files on the end user's front-end computer.

**field group**
A field group comprises a set of related screen elements, for example, all checkboxes in one selection.

**foreign key**
Definition of a relationship between two tables. A foreign key assigns the fields of one table (known as the foreign key table) to the primary key fields of another table (known as the check table). Foreign keys are used to check input on screens, and to define the relationships between tables in a view, matchcode object, or lock object.

**front end**
Workstation computer

**Function builder**
Tool for creating and managing function modules. You can use the Function Builder to create, change, test and document function modules.

**gateway**
Interface that converts one communication protocol into another communication protocol.

**HTML**

HyperText Markup Language (HTML): graphical document description language for creating HTML pages for the Internet. HTML documents have a uniform format and consist of pure ASCII text. They can either be stored statically in the Web server's file system, or be dynamically generated by special programs at runtime. The Web server sends the HTML documents to the Web browser, which interprets and displays them.

**HTTP**

World Wide Web (WWW) application protocol. The HyperText Transfer Protocol (HTTP) controls communication between the Web browser (the HTTP client) and the Web server (the HTTP server).

**IAC**

An Internet Application Component (IAC) provides a complete business solution for connecting mySAP Business Suite components to the Internet. Internet Application Components enable users to access business information in mySAP Business Suite components by starting SAP transactions, function modules or reports in the Web browser. You can make Internet Application Components available to users outside your company using the Internet or to users within your company through the intranet. You can use the Internet Application Components supplied by SAP either with or without modifying them, or you can use them as templates for developing your own.

**IDoc**

Intermediate document: SAP standard format for electronic data exchange between systems.

**instance**

Administrative unit that combines SAP system components providing one or more services. The services provided are started and stopped at the same time.

**Internet Communication Manager**

The Internet Communication Manager (ICM) is the component of the SAP architecture that enables an SAP system to communicate directly with the Internet.

**Job**

Chain of programs executed chronologically by particular control commands.

**keyword**

The first word in an ABAP statement. The keyword determines the meaning of the entire statement.

**LDAP**

Lightweight Directory Access Protocol. Protocol defined in IETF-RFC 1777, for accessing address directories.

**lock mode**

Status that determines whether a user has exclusive access to a data record, or whether access is shared with other users.

**lock object**

Object type in the ABAP Dictionary. Activating a lock object generates function modules for setting and releasing locks; you can then use these function modules in ABAP programs.

**lock table**

The lock table is a table in the enqueue server's main memory that contains entries for the current locks in the system. It also contains information for each lock on the owner, the lock mode, and the name and field of the table locked.

**LSMW**

The Legacy System Migration Workbench (LSMW) provides an easy-to-use environment for converting and importing data from outside an SAP system.

**LU6.2**

Logical Unit Type 6.2: SNA log for program-to-program communication. SNA (System Network Architecture) prescribes the logical structures, formats and logs for the transfer of data within a network.

**message server**

Independent program that maintains a list of all instances in an SAP system. The message server determines which instance a user logs on to and organizes communication between instances.

**mySAP Business Suite**

mySAP Business Suite is a complete package of open enterprise solutions that link all people involved, information, and processes, and therefore improve the effectiveness of your business relationships.

**Native SQL**

Database language that enables the use of database-specific SQL statements in an ABAP program.

**Object Navigator**

Navigation tool for managing development objects.

**OLE**

Object Linking and Embedding is supported by SAP systems. The information required by the OLE interface is transferred using RFC to OLE-enabled applications outside the SAP system.

**Open SQL**

SAP-specific range of SQL statements. Open SQL allows you to avoid conflicts between database tables and makes ABAP programs independent of the database system used.

**Output Request**
An output request contains the data from the spool request in a format specific to the printer model.

**package**
Container for semantically related development objects. A package consists of sub-packages and development objects (programs, tables, screens, function modules, classes, and so on) that are developed and transported together. Packages are characterized by the properties nesting, interfaces, visibility and use accesses. Packages are created and managed with the Package Builder. Packages replace development classes.

**PAI**
Process After Input. PAI is a processing block in the screen flow logic that is executed after the screen is displayed. This processing block calls modules in ABAP programs and determines the processing that is required after a user action on the screen.

**PBO**
Process Before Output. Block of code that is processed after a screen is called but before it is actually displayed.

**profile parameter**
A profile parameter describes a specific setting in the SAP system. You can differentiate between parameters that are valid for a specific instance and those that are valid for the entire system.

**Quick Link**
Navigation tool to facilitate rapid access to specific areas of the SAP Service Marketplace. To call up a Quick Link, simply add it to the Web address, preceded by a "/" (such as: http://service.sap.com/netweaver).

**Repository**
Central store for all ABAP Workbench development objects. The development objects stored in the SAP system Repository include: program objects, function group objects, Dictionary objects, Business Engineering objects and other objects.

**request queue**
Queue into which incoming requests are placed.

**RFC**
The Remote Function Call (RFC) is an SAP interface protocol based on CPI-C. It simplifies the programming of communication processes between systems.

**role**
A role is a collection of activities that a person performs to participate in one or more business scenarios in an organization. You access the transactions, reports, Web-based applications and other objects contained in roles through user menus.

**SAP BC**

The SAP Business Connector (SAP BC) enables the exchange of data between different systems, even across company boundaries. These systems communicate by means of an XML schema known to both.

**SAP Easy Access**

SAP Easy Access is the default initial screen in SAP systems. The left side of the screen contains a tree hierarchy of the menus available to you in the SAP system; you can use the right side of the screen to display a graphic, such as your company logo.

**SAP GUI**

SAP Graphical User Interface; medium that enables the user to exchange information with the computer. The user interface allows you to select commands, start programs, display files and execute other options by pressing function keys or pushbuttons, or by selecting menu options.

**SAP GUI for HTML**

SAP GUI that runs in the Web browser and generates HTML pages dynamically on the basis of SAP screens. Requires an Internet Transaction Server (SAP ITS).

**SAP GUI for Java**

Platform-independent SAP GUI, requires a Java environment.

**SAP GUI for Windows**

SAP GUI implementation in a Windows environment

**SAP ITS**

The SAP Internet Transaction Server (SAP ITS) acts as an interface between an SAP system and the Internet. It enables users to communicate directly with the SAP system by starting business transactions, function modules and reports in a Web browser. When a user starts an application, the request passes from the Web browser to the Web server to which it is connected. The Web server transmits the request to the ITS, which in turn sets up a connection to the SAP system. The ITS then controls the communication and data exchange between the SAP system and the Web server. The ITS consists of two primary components – the WGate (Web gateway) and the AGate (application gateway).

**SAP Logon**

When you call up the SAP Logon, it displays a list of SAP systems for which you can start the logon process. This list is taken from a file on the front end: *saplogon.ini*. This file is usually centrally preconfigured and provided for end users. During logon, the SAP Logon program also enables logon load balancing using the resources available for the system selected.

**SAP NetWeaver**

SAP NetWeaver is the technical foundation on which almost all SAP solutions are currently based.

**SAP Note**

An SAP Note is text information on a specific topic, problem or system message that you may come across when working in the system. All SAP Notes are stored at SAP in an online database, and customers can call them up from there. An example question could be: “Which versions of Database XY are approved by SAP for use with my SAP system? ”

**SAP Service Marketplace**

Central portal (URL http://service.sap.com) that provides services, information and other offers to SAP customers and partners.

**SAP system**

An SAP system is, for example, an SAP R/3 Enterprise system, an SAP BW system, or an SAP CRM system. SAP systems are the central components of SAP solutions within the framework of mySAP Business Suite.

**SAP transaction**

An SAP transaction describes a logically complete action in an SAP system. From the user’s point of view, a transaction represents a unit (for example, creating a list of a certain type of customer, changing a customer’s address, creating a flight reservation for a customer, or executing a program).

**SAP Web Application Server**

Server (software-oriented view) that provides a range of services for operating an SAP system. The terms instance and SAP Web Application Server are generally used as synonyms.

**screen**

Also called “dynpro”, from DYNamic PROgram. A screen consists of a screen image and its underlying flow logic.

**Screen interpreter**

Component that executes the screen flow logic of application programs.

**Screen painter**

Tool for creating screens for a dialog transaction. The graphical layout of the screen image and its underlying flow logic are defined in the Screen Painter.

**Server**

Software component that provides a service (software-oriented view). Central computer within a network that provides resources for the individual work-station computers (hardware-oriented view).

**shared memory**

Area of the main memory that all work processes for that instance can access.

**SMTP**

SMTP: Simple Mail Transfer Protocol. SMTP is the most commonly used protocol for transmitting e-mails on the Internet. The e-mail program passes the e-mail to an SMTP server, which then transfers it to the recipient's mail server. In SAP systems, the ICM has now taken on the role of the mail server.

**SOAP**

SOAP: Simple Object Access Protocol For an exact and current definition of the current SOAP standard, see http://www.w3.org

**Spool Request**

A spool request contains information on the data to be output, its formatting, and the printer model used.

**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.

**task handler**

The task handler coordinates processes and manages resources within a work process.

**TCP/IP**

The Transmission Control Protocol/Internet Protocol, developed in 1969, describes a procedure for transferring data between computers. It is the standard protocol for Internet data transfer.

**TemSe**

Abbreviation for temporary sequential file; file in which data, including spool requests and background processing job logs, is stored temporarily.

**The buffer**

Area in the main memory of an instance that can be used for temporary storage of data frequently used by the applications.

**transaction code**

A transaction code (also known as a TCode) is a sequence of characters that identifies a transaction in the SAP system. A transaction code may contain up to 20 characters and must always begin with a letter. Permitted characters are letters from A to Z, numbers from 0 to 9, and the underscore. To call a transaction, enter the transaction code in the command fieldand choose Enter.

**transport**

Transfer of SAP system components from one system to another. The components to be transported are specified in the object list of a transport request. Every transport consists of an export process and an import process: the export process reads objects from the source system and stores them in data files at operating system level. The import process reads objects from the data file and writes them to the database of the target system. The SAP system maintains a transport log of all actions during export and import.

**user context**

Data that is assigned to one specific user. When a user starts a transaction in the SAP system, the work process that handles the request needs the user context. The user context contains a user-specific area with user and authorization data, and a session context for every external SAP session.

**user master record**

The user master record contains the definition of a particular user in the client. Some examples of fields are first name, last name, initial password, telephone number, and so on. The user master record is used to build up a user context (see entry for user context) when the user logs on to the system.

**Variant**

Default value for input fields on a selection screen in a report, to enable the report to be executed in the background, for example.

**VB* tables**

Update tables; the data to be changed is stored here until the full set can be collected and written to the final database tables (within a single database transaction).

**view**

Virtual table that contains no data, but is an application-specific view of one or more tables in the ABAP Dictionary.

**WebDAV**

DAV = Distributed Authoring and Versioning. WebDAV is an XML-based enhancement of the HTTP protocol for asynchronous document management that is used as a standard for accessing documents using a Web browser.

**work process**

Process that handles requests in an SAP system. The work process types are: dialog (for executing dialog programs); update (for changing database entries); background (for executing background jobs); spool (for print formatting) and enqueue (for executing lock operations).

**workflow event**

A workflow event creates a link between an activity in the SAP system and the people involved.

**XML**

XML = Extensible Markup Language. XML is a subset of the Standard Generalized Markup Language (SGML) developed for use on the World Wide Web. Another subset of SGML is HTML, for example.

# Index

# Feedback

SAP AG has made every effort in the preparation of this course to ensure the accuracy and completeness of the materials. If you have any corrections or suggestions for improvement, please record them in the appropriate place in the course evaluation.