SAPTEC

Fundamentals of the SAP Web Application Server

SAP NetWeaver

Date _____

Training Center _____

Instructors _____

_____

Education Website _____

Instructor Handbook

Course Version: 2005 Q2
Course Duration: 3 Day(s)
Material Number: 50072413
Owner: Christoph Maerker (D038226)

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

Copyright

Trademarks

Disclaimer

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. 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
	Procedures
	Indicates that the item is displayed in the instructor's presentation.

Contents

Course Overview

This course offers an introduction to working with the functions of the *SAP Web Application Server*. The *SAP Web Application Server* is a 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 SAP system technology, such as system administrators, technology consultants and developers.

Course Prerequisites

Required Knowledge

- A basic understanding of data processing

Course Duration Details

Unit 1:
SAP Solutions

mySAP Business Suite and mySAP ERP	60 Minutes
Definition of SAP NetWeaver	30 Minutes
The SAP Release Strategy	30 Minutes
Exercise 1: SAP Release Strategy (optional)	10 Minutes

Unit 2: Navigation

Navigation in General	40 Minutes
Exercise 2: Logon and Screen Design	10 Minutes
Advanced Navigation in the SAP GUI	60 Minutes
Exercise 3: Selecting Functions, Searching and Filtering Data	30 Minutes
Appendix - Personalizing the User Interface	0 Minutes
Exercise 4: Appendix - Personalizing the User Interface	10 Minutes

Unit 3: The System Kernel

Principal Architecture of the SAP Web Application Server	75 Minutes
Exercise 5: Structure of an Instance	10 Minutes
Dialog Processing in the SAP System	30 Minutes
Exercise 6: Dialog Processing	5 Minutes
Communication with the Database	30 Minutes
Appendix - The SAP Transaction	0 Minutes
Appendix - Lock Management in SAP Systems	0 Minutes
Exercise 7: Appendix - Lock Management	15 Minutes
Appendix - Update Processing	0 Minutes
Exercise 8: Appendix - Update Procedure	10 Minutes
Unit 4: Software Development in SAP Systems	
Data Structure of an SAP System and Transports between SAP Systems (ABAP Stack)	50 Minutes
Accessing and Editing ABAP Repository Objects	75 Minutes
Exercise 9: Accessing and Editing ABAP Repository Objects	30 Minutes
Introduction to the SAP NetWeaver Java Development Infrastructure	50 Minutes
Unit 5: Communication and Integration Technologies	
Cross-System Business Processes	30 Minutes
Remote Function Calls and BAPIs	30 Minutes
Exercise 10: Remote Function Calls and BAPIs	10 Minutes
Web Services	30 Minutes
SAP Business Workflow	30 Minutes
Exercise 11: Leave Request as Workflow	10 Minutes
Unit 6: Tools for SAP System Administration	
Daily Tasks in System Management	60 Minutes
Exercise 12: Daily Tasks in System Management	25 Minutes
SAP Service Marketplace	30 Minutes
Exercise 13: SAP Service Marketplace (Optional)	10 Minutes
SAP Developer Network	30 Minutes
Exercise 14: SAP Developer Network (Optional)	10 Minutes
Unit 7: SAP NetWeaver and Enterprise Services Architecture	
SAP NetWeaver – An Overview	90 Minutes
From SAP R/3 to mySAP ERP and the Enterprise Services Architecture	30 Minutes

Course Goals

This course will prepare you to:

- Describe the architecture of the *SAP Web Application Server* in detail.
- Describe the software development processes at SAP
- 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 SAP systems
- 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 (or its successor site). Up-to-date tips for instructors are collected in a document (SAPTEC_52_IG.doc) in this area. The documents you have in front of you now represent the status as at the end of December 2004. 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 systems 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 book 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. The instructor should also know the basics of the architecture of the SAP Web AS Java.
- 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. An example of this is SAP Basis ↔ mySAP Technology ↔ SAP NetWeaver. 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, SAPNW, ADM100, ADM102, ADM200, BC400, 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. Furthermore, the *Java Technology in the SAP Web Application Server* area should be looked at.

- This course is taught using a copy of the SAP ECC Master System 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.
- An optional, second instance is no longer available for SAP ECC. The course takes place on 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.

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. In particular, the significance of mySAP ERP as the successor of the classical R/3 should be emphasized. 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

The last unit of the course provides a detailed introduction to SAP NetWeaver, once the participants know the structure and functions of SAP systems.

At first, a lesson deals with the distinction between solutions and components. Here, it should be made very clear, that a mySAP solution technically consists of several components (the SAP systems) and that every SAP system generally has its own database. After that, it is mentioned that all mySAP solutions are based on the same technology and integration platform: SAP NetWeaver. At this point, do not deal with the individual components of SAP NetWeaver in detail - this happens in the last chapter of this course - but point out that SAP Web AS is a central part of almost every SAP system. That is, participants will find the functions discussed in this course in their company in SAP CRM as well as SAP ECC and SAP XI.

In the last lesson of the chapter, it is attempted for the first time to document the release strategy of SAP in a customer course. We strongly recommend that the instructor closely monitors updates or changes to the release strategy in SAPNet and mentions them in the course if necessary.

Unit Overview

This unit introduces you to the structure of the **mySAP Business Suite** family of solutions. Furthermore, you are given a first overview of the integration and technology patform *SAP NetWeaver*. Finally, we will deal with SAP's release strategy.

Hint: Chapter 7 provides a detailed introduction to the component of *SAP NetWeaver*, once you are more familiar with the architecture of SAP systems.

Unit Objectives

After completing this unit, you will be able to:

- Name some of the options for using *mySAP Business Suite* and *mySAP ERP* in your company.
- State the basic concept of *SAP NetWeaver*
- Describe the central role of *SAP Web Application Server* in the context of *SAP NetWeaver*
- Describe the significance of the Ramp Up program
- Describe the 5-1-2 year support cycle
- Find out when SAP software releases will become available

Unit Contents

Lesson: mySAP Business Suite and mySAP ERP

Lesson Duration: 60 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 options for using *mySAP Business Suite* and *mySAP ERP* in your company.

This introductory lesson aims to give the participants an overview of the benefits available to companies using the mySAP Business Suite and mySAP ERP. 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

As your company is about to implement SAP software, you wish to get an overview of the SAP solutions. You urgently require this information to prepare for the planned meetings with the SAP consultants.

The SAP 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.

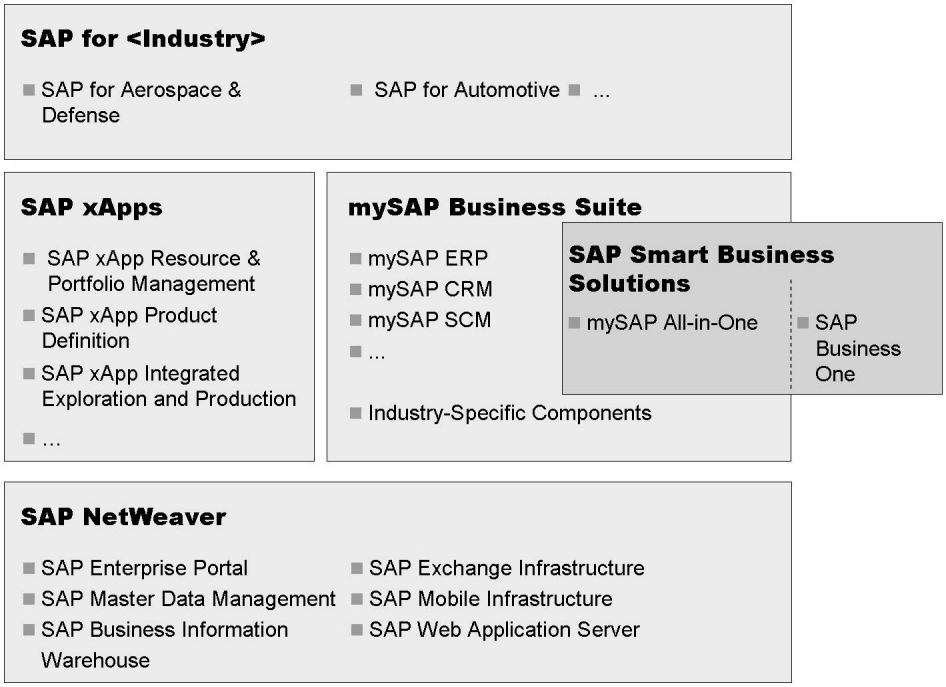

Figure 1: SAP product strategy

The basic elements of SAP solutions are listed below.

- *SAP NetWeaver*
- *mySAP Business Suite*
- *SAP Smart Business Solutions*
- *SAP xApps* (SAP Collaborative 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 on SAP Smart Business Solutions, see http://www.sap.com/smb.

SAP xApps

SAP xApps (Collaborative 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.

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 can therefore improve the effectiveness of your business relationships.
The various SAP solutions are combined into the *mySAP Business Suite* and individually developed further. For more information about all SAP solutions, see http://www.sap.com/solutions. The *mySAP ERP* is based on an SAP ECC 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. Components refer to, for example, an *SAP ECC* system, an *SAP SCM* system or also 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).

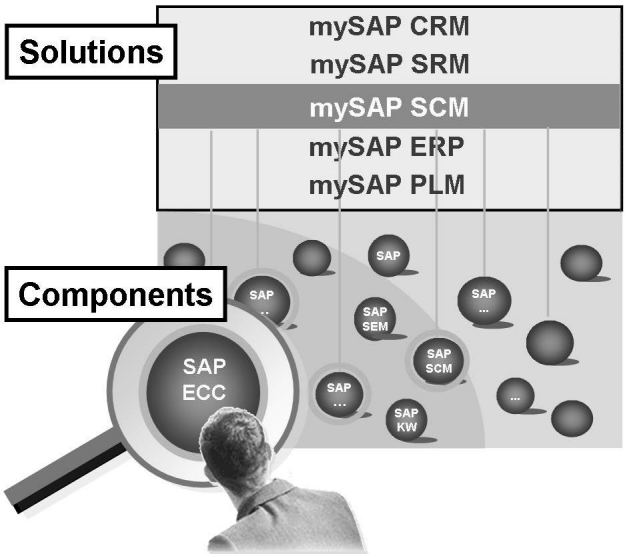

Figure 2: Solutions and components

Possibilities with mySAP Business Suite

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

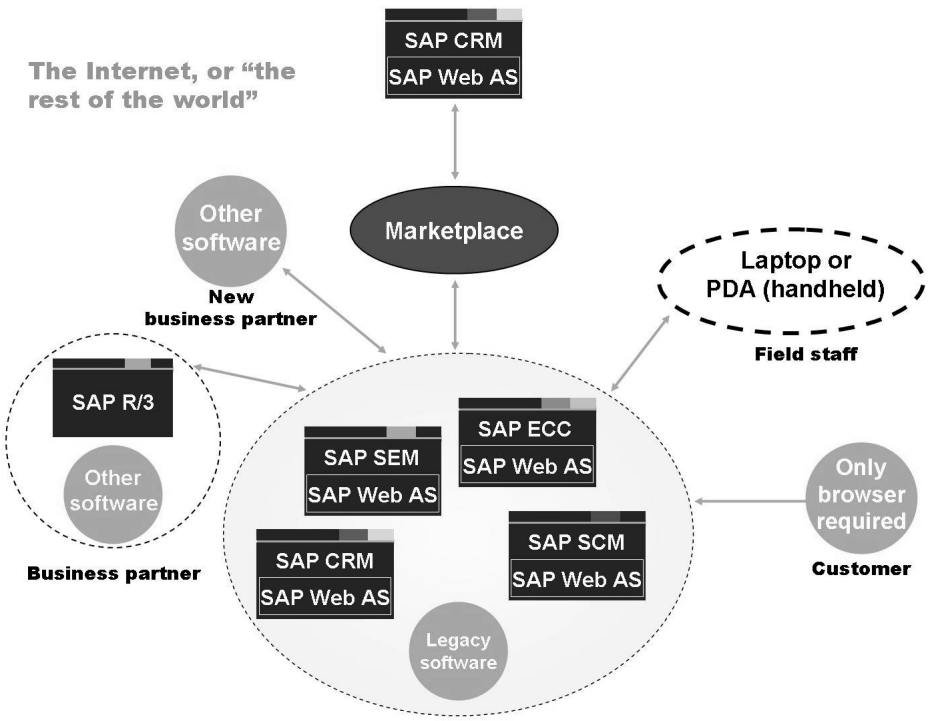

Figure 3: Possible Business Scenarios with mySAP Business Suite

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 Collaborative 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 options for using *mySAP Business Suite* and *mySAP ERP* in your company.

Related Information

For more information, see the following SAP Internet sites. All of these sites are freely available, 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.

- Global SAP Homepage, http://www.sap.com
- SAP Help Portal, http://help.sap.com
- SAP Service Marketplace, http://service.sap.com
- SAP Info, http://www.sap.info
- SAP Insider Online, http://www.sapinsideronline.com

Lesson: Definition of SAP NetWeaver

Lesson Duration: 30 Minutes

Lesson Overview

This lesson provides a brief overview of the technology and integration platform *SAP NetWeaver*. 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 of *SAP Web Application Server* in the context of *SAP NetWeaver*

This lesson should only provide a brief overview of SAP NetWeaver, a detailed consideration of the individual components is given later (as part of the SAPTEC course), when the participants have a good idea about the architecture of SAP systems. The instructor should teach the following: SAP distributes several mySAP solutions, all of which are based on the technology of SAP NetWeaver. The SAP Web AS as an Application Platform of the SAP NetWeaver supplies the runtime environment for all SAP systems. This means that participants do not merely learn about the functions of the “old” SAP Basis, but learn about contents that are of fundamental significance for their work in any system based on SAP NetWeaver (for example, SAP CRM 4.0, SAP SCM 4.0, SAP ECC 5.0, SAP EP 6.0) .

Hint: xApps and SAP MDM require the architecture of the SAP NetWeaver: user interface is the SAP Enterprise Portal, process integration requires the SAP XI, ...

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 a comprehensive technology platform, which can be integrated smoothly into existing systems. Because of the minimum amount of expenses of internal company integration, the operating costs of your entire IT landscape are

reduced. With *SAP NetWeaver*, you can flexibly implement, execute and improve business processes. In this way, you can pursue growth strategies and, at the same time, react quickly to changes.

SAP NetWeaver is the technical basis of the solutions from *mySAP Business Suite* and the Composite Applications, the *SAP xApps*. Furthermore, *SAP NetWeaver* is the core of the Enterprise Services Architecture, SAP's basic concept for Web Service solutions.

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

- **Integration of people, information and processes**
  - Across technological and organizational boundaries
  - .NET and WebSphere inter-operability
- **The technological foundation for SAP customers and partners**
  - Solutions with lower TCO (Total Cost of Ownership)
  - The Enterprise Services Architecture (ESA) increases the flexibility and expandability of business processes

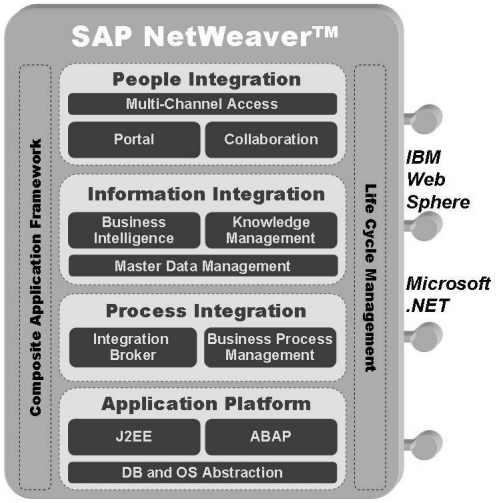

Figure 4: SAP NetWeaver

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 mySAP solutions will be based on *SAP NetWeaver* in the future. At the same time, *SAP NetWeaver* supports the execution of cross-application software, so-called *xApps*. *xApps* are the new class of integrative standard software. *xApps* enhance 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*.

If you want, you can now show the */pam* quick link in the SAP Service Marketplace (and other *SAP solutions*). On the matrix displayed, you can show the individual mySAP solutions with the associated components and explain that all mySAP solutions listed there use the SAP NetWeaver function.

Integration Levels of SAP NetWeaver

SAP NetWeaver provides core functions for the infrastructure of your business solutions in four integration levels.

- **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 the central storage of master data.

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 an 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.

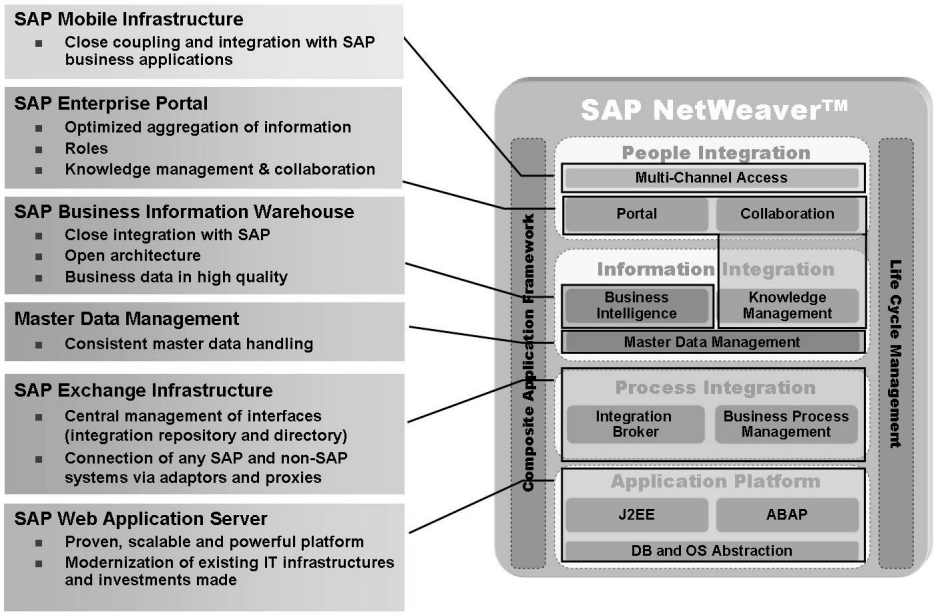

Figure 5: Components of SAP NetWeaver:

In the remaining part of this lesson, we will go into the *SAP Web Application Server* as an *Application Platform* of *SAP NetWeaver* in more detail.

SAP Web Application Server (SAP Web AS)

Nearly every SAP system is based on the *SAP Web AS* as a runtime environment (see following illustration)

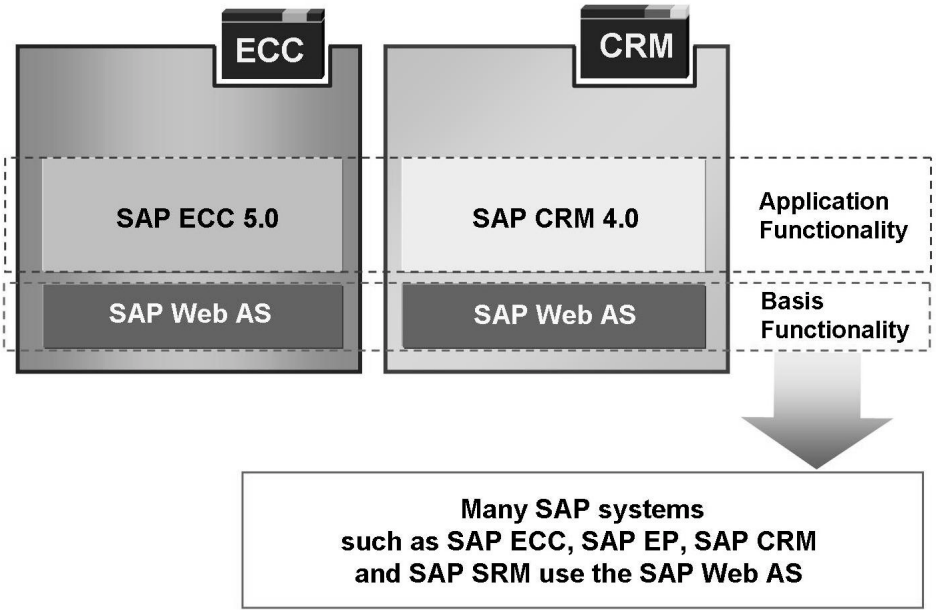

Figure 6: The SAP Web AS as a fixed component of SAP systems.

At this point, you could (and should) tell the participants that the SAP Enterprise Portal and the SAP Exchange Infrastructure are based on an SAP Web AS.

If the participants ask at this point what this architecture has to do with the SAP R/3 Enterprise they are already familiar with, point out that, the SAP R/3 Enterprise is also based on the SAP Web AS and the SAP ECC 5.0 technically consists of the same software components as SAP R/3 Enterprise (aside from the other release number).

Particularly in the administration environment, there may be questions about whether the term “SAP Web AS” on this slide can be understood as the kernel. Make clear that the kernel is a component of the SAP Web AS, but can in no way be seen in the same way! The kernel describes the runtime environment (dispatcher, work processes, message server, ...), whereas the SAP Web AS also describes the software components *SAP_BASIS*, *SAP_ABA* and (new in SAP Web AS 6.40) *SAP_BW*, i.e. the function, for example, of the transaction SU01 or SM59.

The *SAP Web Application Server* has, in addition to the traditional runtime environment for ABAP programs, a runtime environment for J2EE-based Java programs, the *SAP Web AS Java*. The *SAP Web AS* is – together with the database – the *Application Platform* of *SAP NetWeaver*.

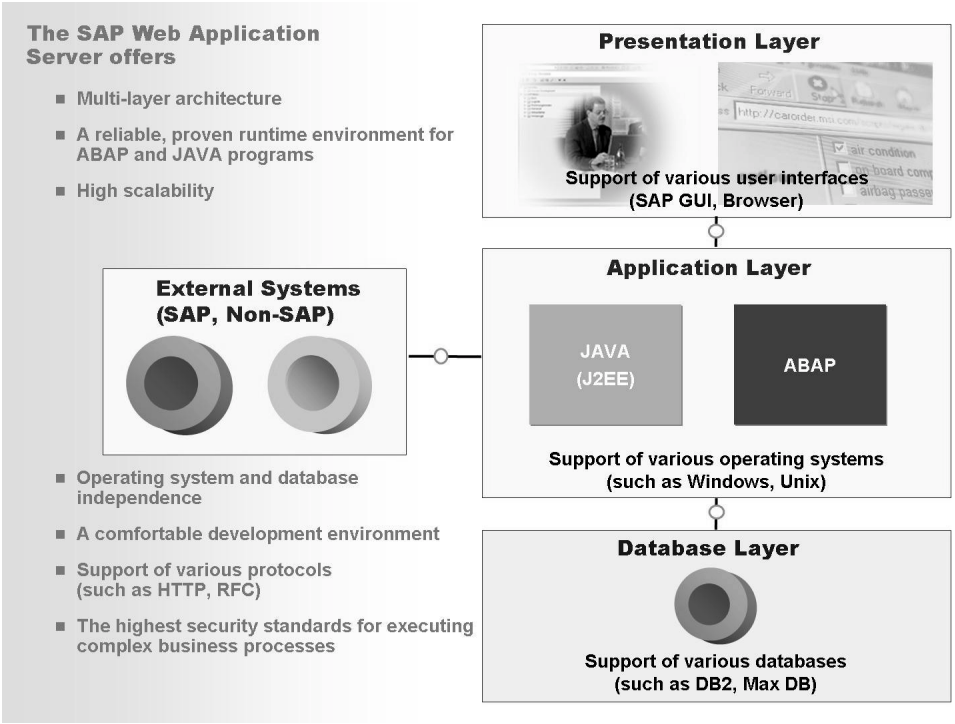

Figure 7: SAP Web Application Server (SAP Web AS)

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 ten 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*?

Note: For further information, here is an excerpt from a marketing brochure on SAP NetWeaver:

The technology platform of *SAP NetWeaver* is the comprehensive integration and application platform, which supports you in reducing your Total Cost of Ownership (TCO). *SAP NetWeaver* brings people, information and business processes together - beyond all organizational and technological limits. *SAP NetWeaver* integrates information and applications from very different sources. The SAP platform, together with the standard market technologies – *Microsoft .NET* and *IBM WebSphere* – is compatible and enhanceable and supports Java 2 Platform, Enterprise Edition (J2EE). *SAP NetWeaver* is the technical basis for *mySAP Business Suite* and *SAP xApps*. The technology ensures the highest degree of reliability, security and scalability so that your central business processes run smoothly. The platform based on Web Services provides an extensive, fully integrated function portfolio. Thanks to the preconfigured contents, the need for customer-specific integration expenses sink - and SAP makes a valuable contribution to reducing your TCO. *SAP NetWeaver* smoothes the way for you in the future: because its technology creates the prerequisites for innovative, cross-function business processes that promote your company's growth.

Lesson Summary

You should now be able to:

- State the basic concept of *SAP NetWeaver*
- Describe the central role of *SAP Web Application Server* in the context of *SAP NetWeaver*

Related Information

- You can find more information under http://www.sap.com/netweaver

Lesson: The SAP Release Strategy

15 Lesson Duration: 30 Minutes

Lesson Overview

This lesson explains what phases your SAP solution runs through over time until the next upgrade to a higher release as well as what maintenance duration is guaranteed for SAP products.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the significance of the Ramp Up program
- Describe the 5-1-2 year support cycle
- Find out when SAP software releases will become available

It should be clear to the participants how the SAP product they use will change over the course of time, whether through patches or through upgrades. SAP supports these processes in many ways, not least through the SAP Solution Manager.

Business Example

You want up-to-date information about the availability of releases of SAP solutions, in order to be able to make a good decision with regard to the purchase or upgrade of an SAP solution.

The SAP Software Lifecycle

If you decide to purchase a software solution from SAP (you can access an overview of the available solutions at http://www.sap.com/solutions), then you face various questions.

General questions concerning SAP software

- From when can my company implement certain SAP software?

  → generally after the end of the Ramp-Up phase
- How is this software maintained?

  → Using Support Packages and Upgrades
- How is this software maintained by SAP?

  → In accordance with the SAP maintenance strategy

The Ramp-Up

The *SAP Service Marketplace* offers you, with the Product Availability Matrix (PAM - http://service.sap.com/pam) an overview of the currently available SAP solutions and also provides a view of the planned availability of future releases.

If a software solution becomes available, it first of all runs through the so-called Ramp-Up. You can find a detailed description of the individual Ramp-Up phases at http://service.sap.com/rampup.

What does "Ramp-Up" mean?

- Ramp-Up is the name for the market introduction process of new products or new releases of SAP software
- The software can be used for a productive implementation
- During the Ramp-Up, the software is only available for a limited circle of customers, customers "apply" at SAP to participate in a Ramp-Up.
- The Ramp-Up serves to take customer wishes into consideration before the respective SAP software is generally available ("unrestricted shipment").
- Participants in the Ramp-Up are given direct contact to the SAP development and are provided with wide support in implementing the SAP software

Please consider the following points:

- SAP recommends using the *SAP Solution Manager* for support during the implementation and operation of SAP solutions.

The *SAP Solution Manager* is an SAP tool that can support the implementation of an SAP solution with various functions.

Note: During the Ramp-Up, SAP also offers you (if your company is taking part in the Ramp-Up) special training options. You can find more information on the so-called **Ramp-Up Knowledge Transfer** at http://service.sap.com/rkt.

How is SAP Software Maintained?

Your maintenance contract includes Support Packages, which are provided by SAP via the *SAP Service Marketplace*. Support Packages solve (component-dependent) problems that can result when using the SAP software. The Support Packages are provided by SAP at more or less regular intervals; you can find a list of the availability of different Support Package types in the *SAP Service Marketplace* under http://service.sap.com/ocs-schedules. In addition, since 2004, SAP has offered a packaging of Support Packages (if it is not necessary for you to import Support Packages as soon as possible after their release), the so-called Support

Package Stacks. This collection of Support Packages appear several times throughout the year: You can find details on this at http://service.sap.com/sp-stacks → *SP Stack Calendar*.

Note: Please note that neither the individual Support Packages nor the SP Stacks are cumulative. This means that, when you import Support Packages, then you import a gapless series of Support Packages and, if you use SP Stacks, you also import these without a gap.

One exception here are the Support Packages that the Support Package Manager (transaction SPAM in SAP systems) patches. These patches are cumulative and therefore do not have to be imported without a gap.

How is SAP software maintained?

- Using Support Packages (and Kernel Patches, etc.)
- By newer software releases

You should think about a follow-up release of the software at the latest, when your SAP software is approaching the end of its software maintenance period. During a release change of your SAP software, you can generally skip one or more of the software releases (of your solution) that have appeared in the meantime. In this way, for example, an upgrade of *SAP R/3 4.6C* to *mySAP ERP 2004* is technically possible, even when you carry out a product change during this upgrade (from *SAP R/3* to *mySAP ERP*).

You can find more details about the SAP release strategy in the *SAP Service Marketplace* under http://service.sap.com/releasestrategy. There you can find statements on planned and current SAP developments.

The SAP Maintenance Strategy and Planning Security for You

Since 2004, SAP has offered a new maintenance strategy for all products that are based on *SAP NetWeaver*. The so-called “5-1-2” maintenance strategy offers maintenance for five years for the standard maintenance fee, followed by one year extended maintenance for a maintenance fee of two percent more and, optionally, another two years for a maintenance fee which is four percent more than the standard maintenance. The following graphic shows this new maintenance strategy using the *SAP R/3* and *mySAP ERP* solutions as an example.

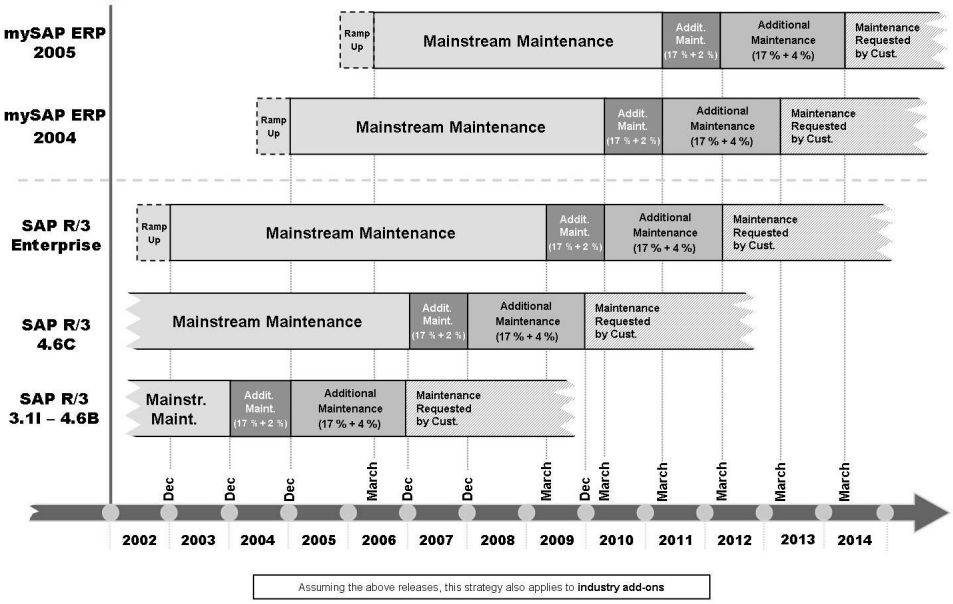

Figure 8: The SAP software maintenance strategy for SAP R/3 and mySAP ERP

As you can see, the product *SAP R/3* with release *SAP R/3 Enterprise* is maintained by SAP until the year 2012, i. e. *SAP R/3* will have a product life of 20 years!

In this way, SAP offers a very high degree of planning security for the operation of its business software.

Exercise 1: SAP Release Strategy (optional)

Exercise Duration: 10 Minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Name some information sources in the *SAP Service Marketplace*.

Business Example

You want to find out the duration of standard maintenance of the product *mySAP ERP 2004* and the time when a Support Package Stack is next available.

Task:

You use the *SAP Service Marketplace*, to find out the end of the standard maintenance for the product *mySAP ERP 2004*, and to find out when the next Support Package Stack for the application in your *mySAP ERP 2004* system will be released.

1. Use the address http://service.sap.com/pam, to find out the duration of the standard maintenance for the product *mySAP ERP 2004*. Please note that the Product Availability Matrix (PAM) is opened in a separate browser window.
2. Use the address http://service.sap.com/sp-stacks → *SP Stack Information* → *SP Stack Schedule*, to find out the availability of the next SP Stacks for the application of your *mySAP ERP 2004* system.

Solution 1: SAP Release Strategy (optional)

Task:

You use the *SAP Service Marketplace*, to find out the end of the standard maintenance for the product *mySAP ERP 2004*, and to find out when the next Support Package Stack for the application in your *mySAP ERP 2004* system will be released.

1. Use the address http://service.sap.com/pam, to find out the duration of the standard maintenance for the product *mySAP ERP 2004*. Please note that the Product Availability Matrix (PAM) is opened in a separate browser window.

   a) On the specified page, select the path *SAP Application Components* → *SAP ERP* → *SAP ERP 2004*, to find information about the end of the standard maintenance. The end of the standard maintenance for *mySAP ERP 2004* is set at 03/31/2010.

2. Use the address http://service.sap.com/sp-stacks → *SP Stack Information* → *SP Stack Schedule*, to find out the availability of the next SP Stacks for the application of your *mySAP ERP 2004* system.

   a) Under the specified path, you can find the information that, for example, SP Stack 8 is planned for week 26 of the year 2005.

Lesson Summary

You should now be able to:

- Describe the significance of the Ramp Up program
- Describe the 5-1-2 year support cycle
- Find out when SAP software releases will become available

Related Information

- http://www.sap.com/solutions
- http://service.sap.com/solutionmanager
- http://service.sap.com/pam
- http://service.sap.com/rampup
- http://service.sap.com/ocs-schedules
- http://service.sap.com/sp-stacks
- http://service.sap.com/releasestrategy

Unit Summary

You should now be able to:

- Name some of the options for using *mySAP Business Suite* and *mySAP ERP* in your company.
- State the basic concept of *SAP NetWeaver*
- Describe the central role of *SAP Web Application Server* in the context of *SAP NetWeaver*
- Describe the significance of the Ramp Up program
- Describe the 5-1-2 year support cycle
- Find out when SAP software releases will become available

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)

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.

3. SAP guarantees certain maintenance durations for *SAP NetWeaver*-based products. The standard maintenance can be extended because of the increased maintenance fees. How many years will an SAP product be maintained in total, if you are willing to pay a maintenance fee increased by 2% for a part of the maintenance time?

Answers

1. The following solutions are provided by SAP:

   Answer: B, C, D

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

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

   Answer: A, D

   The *SAP Web Application Server* is a further development of the previous *SAP Application Server*. The *SAP Web AS* can run applications written in Java using the *SAP Web AS Java* 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.

3. SAP guarantees certain maintenance durations for *SAP NetWeaver*-based products. The standard maintenance can be extended because of the increased maintenance fees. How many years will an SAP product be maintained in total, if you are willing to pay a maintenance fee increased by 2% for a part of the maintenance time?

   Answer: The software product is maintained 6 (5 + 1) years in this case.

Unit 2

25

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.

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
- List some variants of the *SAP GUI*
- Name and use elements of a *SAP GUI* screen
- Log off from the system successfully
- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Carry out complex, selective searches
- Sort and filter listed data
- Execute a where-used list
- Describe the standard menus **System** and **Help**.
- Use the various options for personalizing the SAP system

Unit Contents

Lesson: Navigation in General

Lesson Duration: 40 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
- List some variants of the *SAP GUI*
- Name and use elements of a *SAP GUI* screen
- Log off from the system successfully

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 an SAP GUI screen as well as log off from the system correctly. Course ADM100 covers how to configure SAP logon.

Business Example

You want to find out which options are generally available to your users to log on to the SAP system. What does the interface look like and what is the significance of the different screen elements of the *SAP GUI*?

Different SAP GUI Variants

The presentation interface, *SAP 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 or C++, runs on Windows-based platforms, and supports all transactions in an SAP system. The flow of data between the presentation layer and the application layer 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.

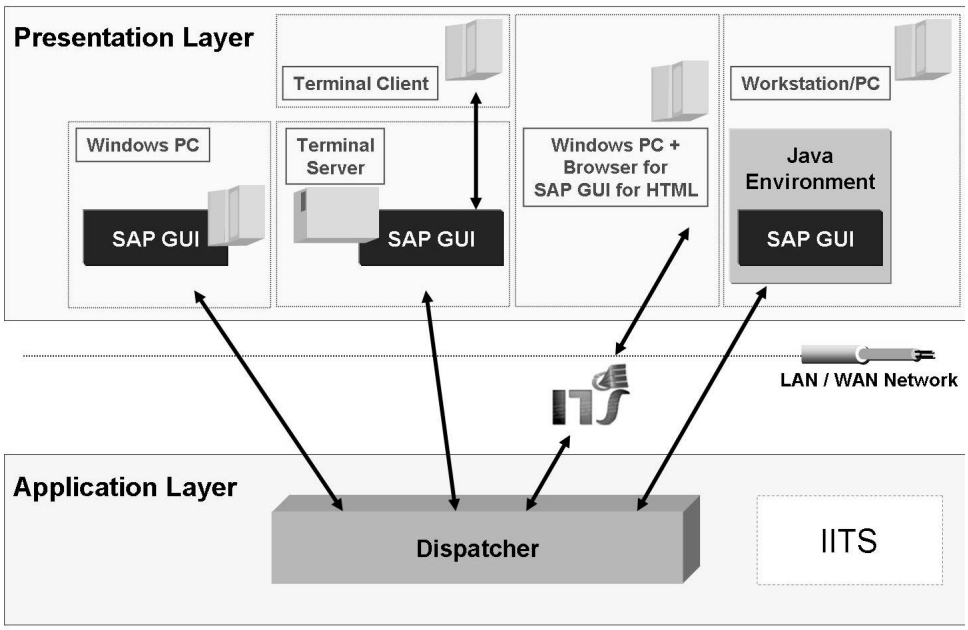

Figure 9: Alternative types of SAP GUI

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

Logging on to the System

All SAP solutions are accessible using a general front end program, the *SAP GUI (SAP Graphical User Interface)*. For the sake of simplicity, further descriptions refer to the *SAP GUI* for the Windows environment.

Hint: You can also access certain SAP system via specially designed front end programs. For example, the *SAP Business Information Warehouse (SAP BW)* has the *Business Explorer (BEx)* as a BW-specific front end.

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*; in practice, you never need to do this.

This is a possible command line for logging on to system I20: sapgui /M/iwdf4020/G/SPACE/S/sapmsI20 (load balancing) and sapgui /H/iwdf4020/S/sapdp00 (server selection)

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.

Hint: You can also store links and logon entries on the front end PC. The links have the form of an icon allow the targeted access to transactions in the SAP system.

During logon, the SAP Logon program also enables **logon load distribution** using the resources available for the system selected.

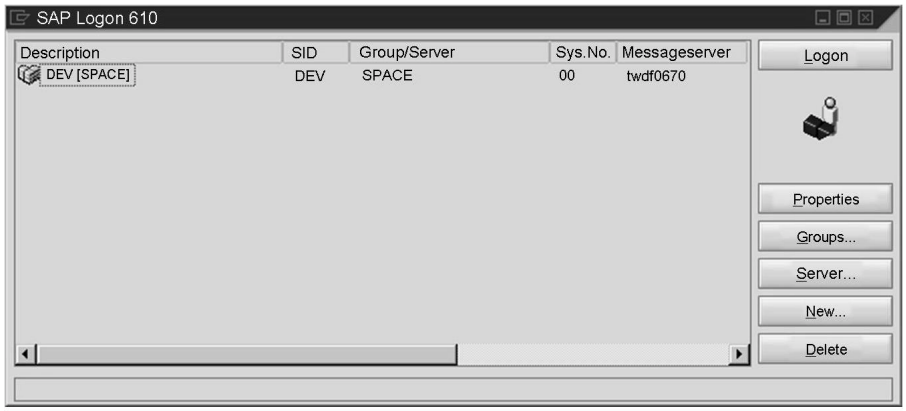

Figure 10: The SAP Logon program

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 on 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.

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. Parameter `rdisp/max_alt_modes` can be set to values from two to nine; the standard setting is six windows. The parameter cannot be set user-specifically but applies for all users of the system or the logon instance.

For SAP Web AS 6.20 you could increase the number to 16(!); from Web AS 6.40 nine sessions are the upper limit. Six windows, the old upper limit, is now the default setting.

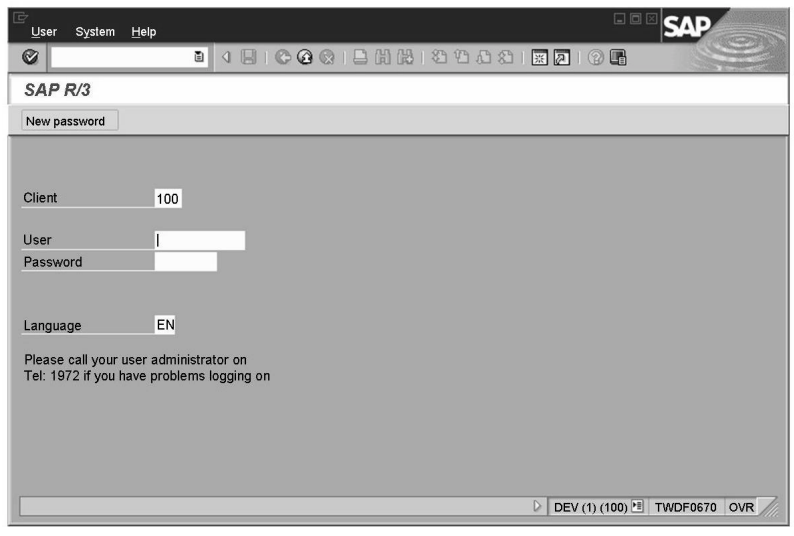

Figure 11: The logon screen for an SAP system

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:

Multiple logons are logged in table USR41_MLD.

- 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

Changing the Password

Use the initial password assigned to you when you log on to the system for the first time. During the first login, you are prompted to save a changed password. For security reasons it makes sense to change the password occasionally.

32

Changing the Initial Password

1. Enter your new password twice, whereby you activate both input fields in sequence by means of mouse-click or tab button.

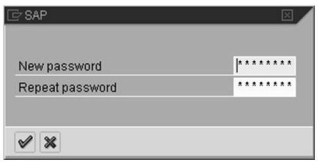

Figure 12: Dialog Box for Changing the Password

2. Finally, choose ✔ to save your new password.

Changing the Changed Password Again

Prerequisites

Hint: Users can change their password no more than once a day on the initial screen. In contrast, user administrators with authorization for transaction SU01 can change a user's password as often as they want. Actually, the administrator can only assign new initial passwords.

Procedure

1. On the logon screen of the SAP system, enter your data and your current password.
2. On the top left, click the *New Password* pushbutton.
3. Change your password as described above and save your data.

Different system parameters with the prefix login/* influence the password response in the SAP system. Parameter login/password_expiration_time, for example, specifies after which number of days the user has to change his password.

Screen Structure

You can, for example, use transaction GUIT to demonstrate the different screen elements. On the individual screens of this transaction, examples of input fields, pushbuttons, radio buttons and checkboxes are illustrated. Screen 700 shows frequently used icons in the SAP system. System table ICON contains all icons and their technical names. The behavior of the navigational arrows in transaction GUIT is also instructive: The green arrow (*Back*) is not active whereas the yellow and red arrows still work. You can also close transaction GUIT by entering command */n*. The instructor can also use other transactions to demonstrate the screen elements.

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.

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.

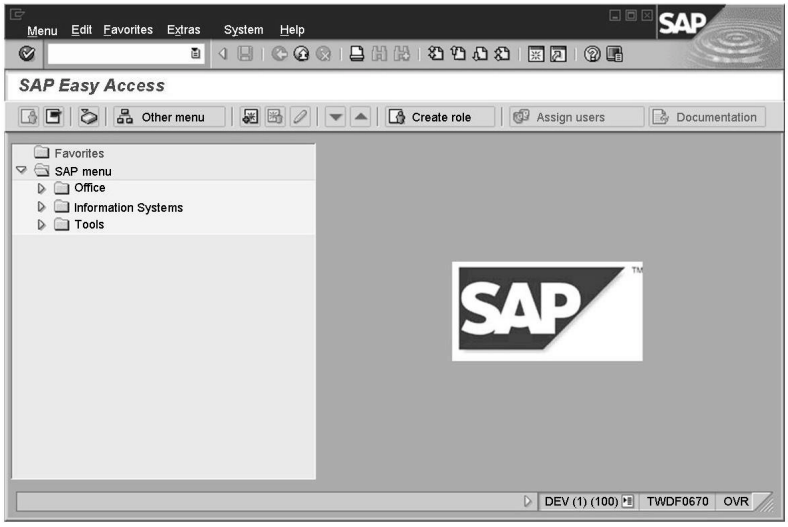

Figure 13: SAP Easy Access

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 *Tools* → *Settings* in the SAP Easy Access screen.

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

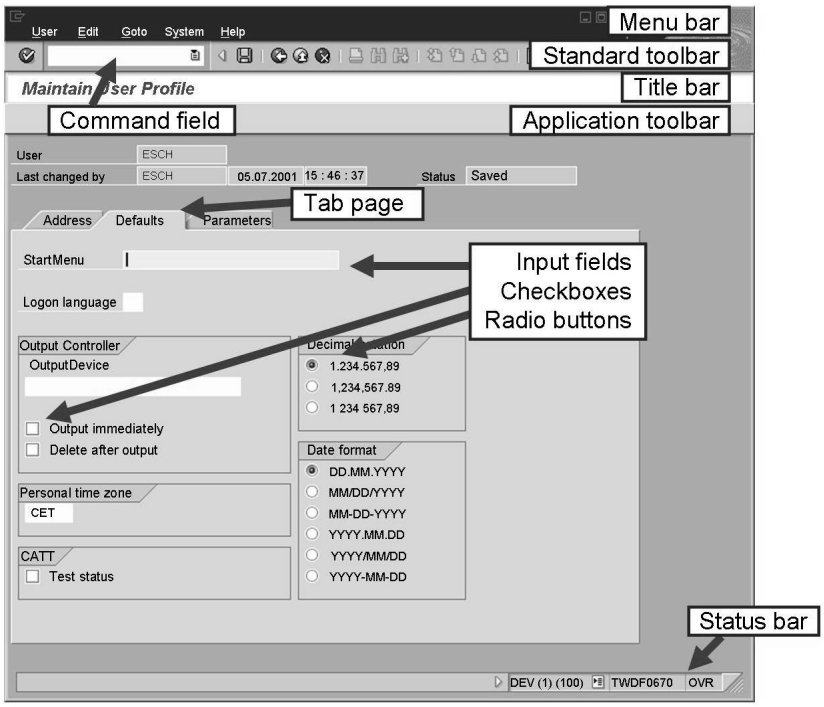

Figure 14: Maintaining your User Profile

An SAP screen can contain the following simple screen elements:

- Command field
- Menu bar, system function bar, title bar, application toolbar, status bar
- Check boxes
- Radio buttons
- Tab

You can start applications directly by entering their transaction code in the **command field**, which is usually hidden. Find transaction code. 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*.

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.

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.

The **title bar** is the function that you are currently in.

The **application toolbar** shows the pushbuttons available in the application that you are currently in.

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.

Within a field group **check boxes** allow to select several options at the same time.

With **radio buttons** you can only select one option.

A **tab** organizes several subscreens to improve clarity.

Input fields and **pushbuttons** are additional screen elements.

Logging off from the System

You should log off the system when you have finished your work in the SAP system. You have several options for doing this:

36

Log off via the Menu Bar

1. In the menu bar, select *System* → *Log off*.

Hint: Alternatively, click the yellow arrow in the SAP Easy Access menu or enter the **/nend** command in the command field. If several sessions are open, the yellow arrow only closes the active window. The dialog field for logging off only appears in the last window.

Caution: The **/nex** command immediately terminates your logon without querying the dialog windows first. All sessions of your terminal session are closed and entries that have not been saved are lost!

2. A Log off Dialog Box appears and tells you that you will lose data that has not been saved when you log off.

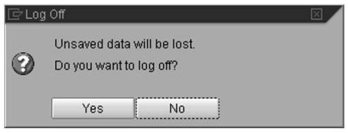

Figure 15: Log off Dialog Box

3. To log off, choose *Yes*.

Hint: From the perspective of the *Windows* operating system, an SAP session is a normal application. Hence, you can also close SAP sessions with the means of the operating system, such as by using the key combination *Alt+F4*. When you close the last window of your session, the dialog box is also displayed.

Exercise 2: Logon and Screen Design

Exercise Duration: 10 Minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Logon to the SAP system and navigate

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 2: 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 9 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. Alternatively, you can permanently display the transaction codes in the menu. Choose *Menu bar → Extras → Settings* and activate the field *Display Technical Names*. A fourth option would be to display the details by choosing *Menu Bar → Extras → Technical Details*. For this, the corresponding menu entry would first have to be selected with the mouse.

Lesson Summary

You should now be able to:

- Log on to the system successfully
- List some variants of the *SAP GUI*
- Name and use elements of a *SAP GUI* screen
- Log off from the system successfully

Related Information

- For more information, refer to the online documentation in *Help* → *SAP Library* → *Getting Started*.

Lesson: Advanced Navigation in the SAP GUI

Lesson Duration: 60 Minutes

Lesson Overview

In this lesson you will learn about various ways of calling functions in SAP systems. In addition, different selection options and help functions are introduced. Finally, you will learn about how to work with list outputs.

Lesson Objectives

After completing this lesson, you will be able to:

- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Carry out complex, selective searches
- Sort and filter listed data
- Execute a where-used list
- 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 to 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 functions can be started in the SAP system and what help the system provides you with.

User Menu and SAP Menu

Since release R/3 4.6, the SAP Access Menu starts automatically after login and is thereby the standard entry into the system. The left-hand navigation bar contains a clear tree structure, consisting of menu entries and favorites' entries. You can adjust the favorites' area in particular to your needs. The entries of the SAP menus and the user menus can only be changed by system administrators with certain authorizations.

SAP and area menus can be processed using SE43. The user menus are bound to authorization roles. The latter are processed using PFCG.

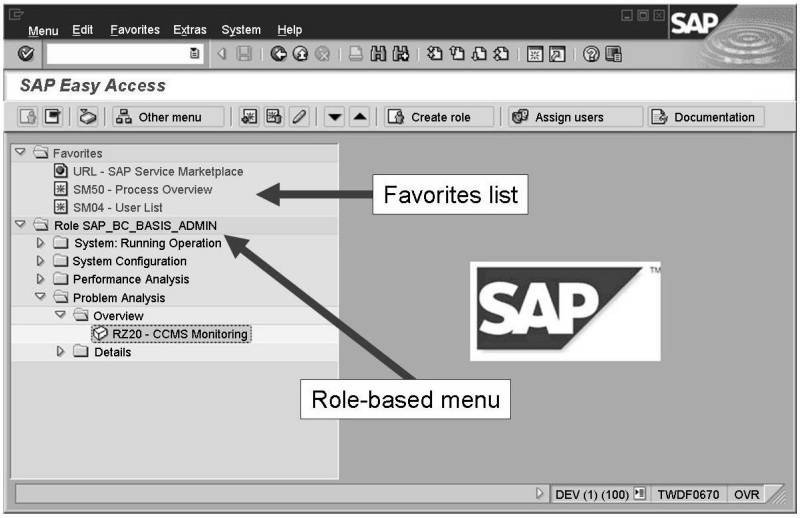

Figure 16: Favorites list and user menu

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. This means that a user can always call authorized functions using a transaction code, independent of the menu structure.

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.

The user and SAP menus can be structured in a pretty complex way. It is then difficult to remember the exact path for the transaction you are looking for. As long as the transaction has not been copied into the favorite area, you can find it more easily with special search transactions. The transactions SEARCH_SAP_MENU and SEARCH_USER_MENU search the corresponding menus for the predefined text pattern. The search result is prepared in a list view from which you can take the navigation path. The transactions found cannot be started directly with a double-click.

A further search option for SAP menu entries is the transaction SDMO, which also allows you to search for a text pattern. A (technically out-of-date) dynamic menu is generated, from which the transactions found can be started with a double-click.

Favorites Management

In addition to the area menu or SAP menu, the user is also provided with functions in the favorites' area. Often used transactions, web links or files can be stored in the favorites' area.

The users' favorites' entries are in the system tables SMEN_BUFFC and SMEN_BUFFI. Administrators can access these table to make changes and thereby change or delete the favorites of large user groups.

The favorites list thus 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.

Possible instructor demo: Demonstrate the download of your own favorites menu. Save the data, for example, under C:\temp\fav.txt on your front-end PC. After that, delete the complete favorites list. Now upload the favorites from the file you have just saved into the system again.

Hint: Complete favorites menus, incl. the folder structures, can be easily copied by uploading or downloading from one SAP system to another. The downloaded favorites menus are stored temporarily on the front-end PC.

The upload and download of favorites requires the corresponding authorizations for the authorization object S_GUI. It could be that the user does not have authorization in the productive customer system to upload.

To improve the structuring, sort the favorites in own folders. You can edit favorites in the *SAP Easy Access* screen using the menu entry *Favorites*.

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*. Here, the menu path is also written with the text of the favorites entry to a limited extent. If you have highlighted a menu entry using the mouse, you can copy the selected node in the favorites' area by right-hand mouse click.

Alternatively, you can also drag the elements from the menu area using the mouse and store them in the favorites' area (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.

Hint: To change the name of a transaction in Favorites, highlight the corresponding favorites entry and select the menu path *Favorites* → *Change*. You can change and save the text in the dialog box that opens.

You can delete favorites' entries that you no longer need. Here, only the Favorite is deleted, not the original transaction! If you want to delete the entire Favorites menu, highlight the entry *Favorites* and click with the right-hand mouse button. Choose *Delete all favorites*.

Calling Up 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

Figure 17: Various navigation options

You can reach the menus in the menu bar simply using the key combination **`Alt+E`** (the underlined letter of the selected menu item) or **`Alt`** and cursor navigation using the arrow keys on the keyboard.

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** 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 *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).

Help Options

The SAP system provides you with various help options, which will be introduced in the following section.

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. There you will find 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.

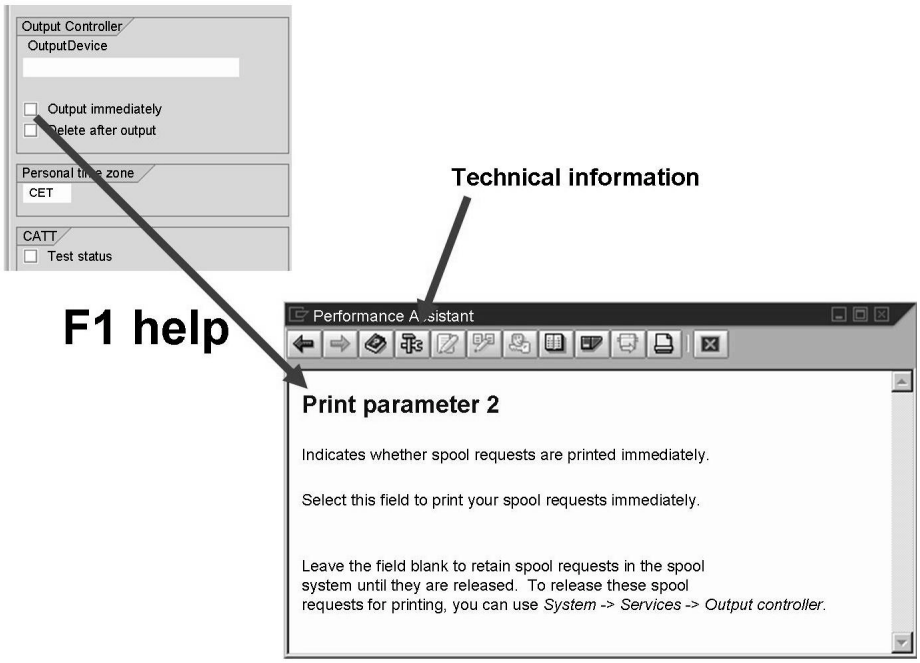

Figure 18: The F1 help

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 set defaults for frequently used input fields; 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.

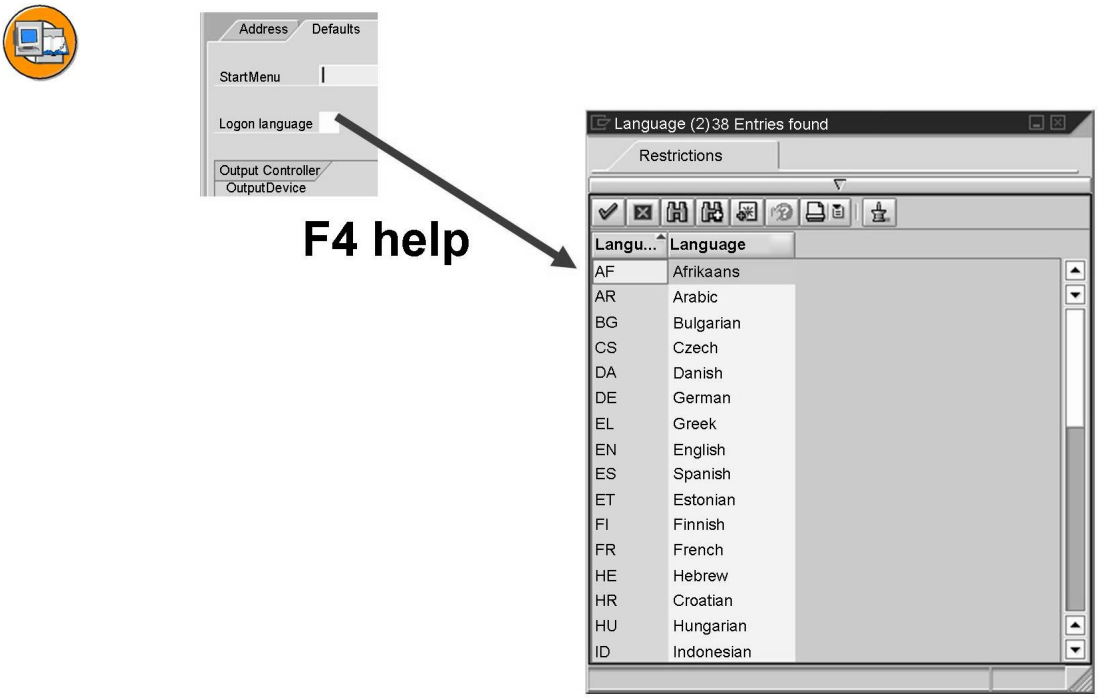

Figure 19: The F4 help

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.

Hint: Users can also create personal values lists. An S&D employee who is only responsible for Munich, for example, only wants to see the Munich clients and defines a corresponding personal value list. However, despite this, the employee still has the option at all times of switching to the total view of all customers or of updating or completely deleting the values list.

Demonstrate how you create a personal list of values using SU3 as an example for the possible entries of the logon language, *Fixed values* tab page. Enter DE and EN in your personal values list. Demonstrate the effect when you call up the possible entries again after creating the personal list.

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.

Figure 20: 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.

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*.

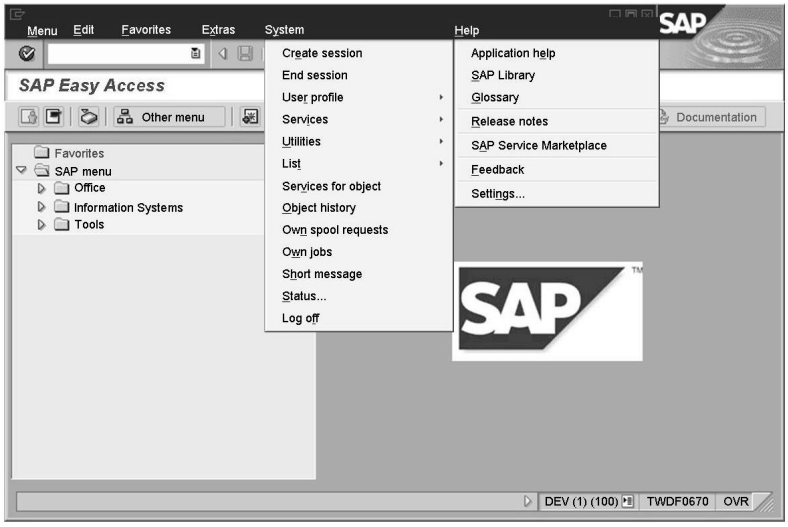

Figure 21: The System and Help menus

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**.

Appendix: Selection Screens

The following three sections are seen as an appendix. Depending on the time situation and what the participants already know, the instructor can explain these points more briefly or in more detail. Even when these subjects are only to be explained briefly, you must emphasize that every SAP user has to have a command of these.

When preparing data in your system, only certain data should be displayed selectively. **Various selection options** are available for this.

- **Search for individual values**

  An exact search for the specified individual values is made.

Figure 22: Selection according to individual values (appendix)

- **Search for value areas**

  A closed interval can also be specified; the interval limits here are included in the search.

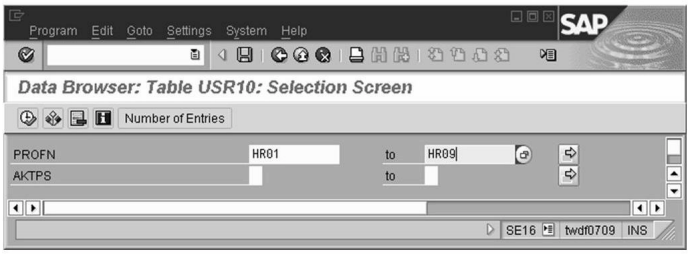

Figure 23: Selection of intervals (appendix)

- **Search using selection options**

  Here, search operators are specified meaning that searches for intervals that are upwardly or downwardly open are possible. Selection options can usually be called up via the menu entry *Edit → Selection options*, alternatively via the *F2* key. A special icon is often used to call up the selection options, as then the access to the menu bar is not possible.

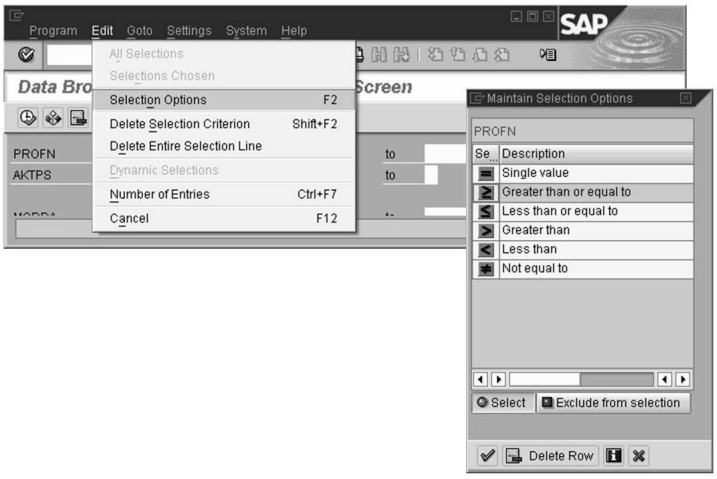

Figure 24: Setting selection options (appendix)

- **Excluding individual values**

  Works like the search for individual values, bit with the difference that the specified individual values can no longer exist in the data records.

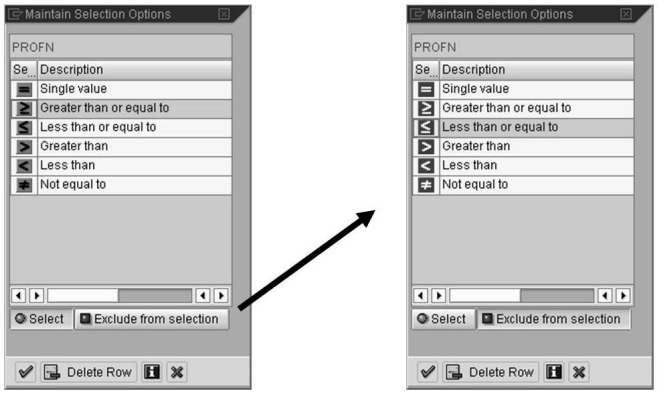

Figure 25: Calling up the exclusion selection option (appendix)

- **Exclusion from value ranges**

  The specified value ranges can not exist in the data records. As with the search for intervals, the interval limits are also affected by the exclusion selection.

- **Exclusion via the corresponding selection options**. The exclusion of open intervals is possible here.

- **Multiple selection**

  All search and exclusion options can also be used as part of a multiple selection at the same time. Therefore, you cannot only look for several individual values or intervals at the same time, but can freely combine the various searches and exclusions. You should try to first find the data you want with a pure search selection, because the logical structure of a positive search can always be more easily reproduced than the one for exclusions. If the positive serach is not sufficient, the search criteria can be enhanced by exclusions. In principle, a positive search can always be mapped by a complementary exclusion selection.

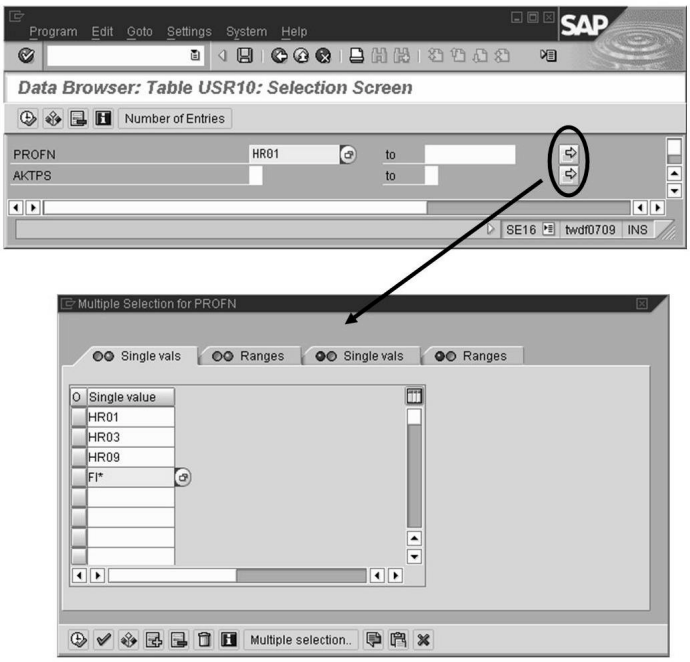

Figure 26: Calling up a multiple selection (appendix)

Hint: A good search strategy decides on the data found. Avoid logical errors, as otherwise you will only find a limited number of data records and, in extreme cases, none will be found. Selection options that are too broad are also bad, beacuse then too many data records are found and this is inefficient and makes the evaluation difficult. In the manual search, a hit list that is too large will oversee relevant entries. Therefore, invest enough time in formulating your search strategy. With complex selection screens, it is advisable to save own search entries as own variants.

The trainer can show how entry variants are created in a transaction of his/her choice.

If you have made the selection criteria too extensive and the search has already been running for several minutes, you can interrupt the already started transaction as long as you can see the hour glass. To do so, click the SAP symbol in the upper left-hand corner of your mode and choose the entry *Stop transaction*. The selection is stopped and you go back to the SAP initial screen.

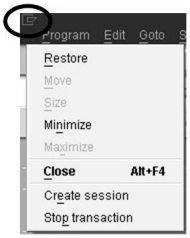

Figure 27: Stop running selection (appendix)

Appendix: List Editing

The instructor should be able to call up various list displays to display the navigation and editing options. For example, you can call up diverse system tables in using transaction SE16, such as T100, TSTC(T), SMEN_BUFFC, USR10, etc., in order to display their contents in a list output. You can also display a list of system parameters using transaction RSPFPAR. Of course, data from the application transactions can also be used, as long as the instructor is familiar enough with the transaction.

After successfully calling up a selection, the data records found are displayed in a list display. Various navigation and editing options are available.

- Searching for definitions or values using the search function; you are also provided with a search repeat
- Page by page scrolling in the list
- Navigation in the list using the scrollbar

Hint: If you are searching for only one certain data record, but have several dozen data records in the list display, then you have made the selection criteria too broad. In this case it is better to formulate a more precise search and to reselect instead of wasting too much time searching in the list.

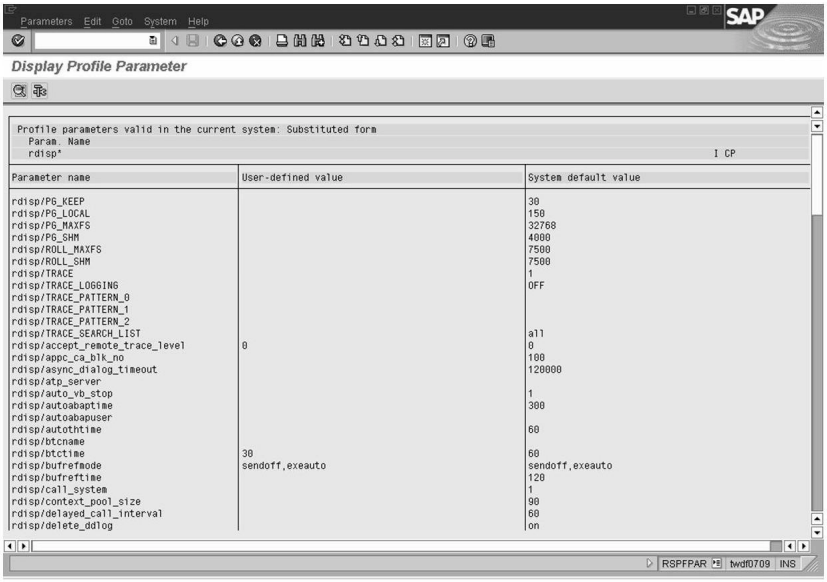

Figure 28: List display and navigation options (appendix)

- Sort according to individual columns

  The data records of a column can be sorted in ascending or descending order. Here, you highlight the header line of the column and press the keys *Sort in Ascending/Descending Order*.

- Filtering

  The filtering is a subsequent selection in the list display via the entries of selected columns. To do this, you highlight the column you want and press the key; then you define additional selection options.

- Editing the list layout

Every user can activate the SAP List Viewer in SE16. To do this, press *User parameters* Then you select the ALV grid display and save the setting.

The *SAP List Viewer* (ALV grid display) is a tool, with which the layout of the table can be individually designed. In this way, for example, you can display additional columns and calculate totals and subtotals. You can even save a layout variant.

- Printout of the List

  Start the printout using the pushbutton in the standard toolbar.

- Downloading the List

The list download can also be started using the %pc command which has to be entered in the command field.

As a rule, the application programmer stores a download pushbutton in the application toolbar, as long as downloading the list is planned. The download can also be started at all times via the menu bar entry *List → Export → Local file*. Alternatively, you can also use the menu bar entry *System → List → Save → Local file*.

Hint: Not all lists in the SAP system use the complete function of the *SAP List Viewer*. Other lists, by comparison, provide special functions, which go far beyond the functional scope of the*SAP List Viewer*.

Appendix: The Where-Used List

A good demonstration is the where-used list for an authorization object. In the Repository Information System, transaction SE84, the authorization objects are displayed under *Other objects*. You make your selection on the authorization object S_USER_PRO. You can keep the where-used list on the individual record view. Depending on the selection, the programs or transactions are displayed, in which an authorization check on the object is programmed.

The where-used list is a basic navigation technique, which every SAP user should know, even if it is not needed every day. As long as this function has been implemented in the application transaction, you can find out the use of a program element, for example, a data field in tables or structures.

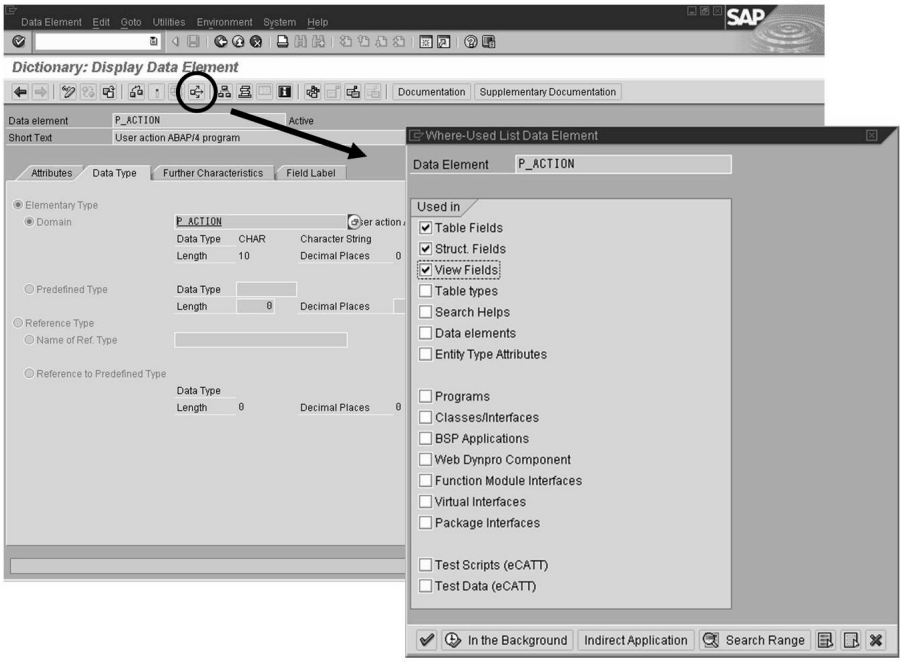

Figure 29: Calling up a where-used list for a data element (appendix)

55

Exercise 3: Selecting Functions, Searching and Filtering Data

Exercise Duration: 30 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 you have been given (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. Select from the SAP standard menu *Tools → Administration → Monitor → System Monitoring → User Overview*.
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.
3. Enter transaction SM04 as the start transaction, so that it is always called up according to the logon.

Continued on next page

Task 2: Creating a Favorites Menu

Save often-used transactions in a favorites menu.

1. Expand the SAP menu to the user overview: *Tools → Administration → Monitor → System Monitoring → User Overview*. Highlight the transaction entry with the mouse. Then select *Favorites → Add*in the menu bar. The transaction is saved in the favorites' area.
2. Add the following transactions to the favorites menu without expanding previously in the SAP menu: RSPFPAR, SEARCH_SAP_MENU, SEARCH_USER_MENU
3. Can you distinguish between favorites' entries by the method used to insert them?
4. Create a folder with the title *Search Options* in which you can store both search transactions.
5. Find out how often and where in the SAP menu the transaction for the user overview is stored.

Transaction	Path in the SAP menu
SM04	
SM04	
SM04	

Task 3: Using F1 and F4 Help

You can gain additional information on ABAP objects using the F1 and F4 helps.

1. Find out which system table the transaction codes of your SAP system are stored in! To do this, use the dialog input field to enter a start transaction in the *SAP Easy Access* menu settings. (*Menu bar → Extras → Set start transaction*)
2. You already know transaction SM04. Are there other transactions whose technical name also begins with “SMO”? Research this in the *TSTC* table with the help of the *Data Browser*, transaction SE16.

Task 4: Searching for Data Records

You should search for a dataset directly. For example, you want to list certain authorization profiles delivered by SAP.

1. In the *USR10* table, search for all profiles delivered by SAP.

Continued on next page

Task 5: Filtering Data (Optional)

You want to subsequently restrict or filter a large number of hits in the list display.

1. In the *USR10* table, display all of the composite profiles delivered by SAP that have been changed since 1999.
2. Filter the data in such a way that only the profiles from the year 2000 are shown.

Task 6: Where-Used List for a Data Element (Optional)

You want to research which tables and structure fields a certain data element can be found in.

1. In which tables and structures is the data element P_Action found? Begin your search in the Repository Information System, transaction SE84.

Solution 3: Selecting Functions, Searching and Filtering Data

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. Select from the SAP standard menu *Tools → Administration → Monitor → System Monitoring → User Overview*.

   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

3. Enter transaction SM04 as the start transaction, so that it is always called up according to the logon.

   a) Call up the following functions in the menu bar: *Extras → Set start transaction*. In the following dialog box, enter the technical name of the start transaction you want. Then save your entry.

Task 2: Creating a Favorites Menu

Save often-used transactions in a favorites menu.

1. Expand the SAP menu to the user overview: *Tools → Administration → Monitor → System Monitoring → User Overview*. Highlight the transaction entry with the mouse. Then select *Favorites → Add*in the menu bar. The transaction is saved in the favorites' area.

   a) See exercise text.

   

   Hint: Alternatively, you can use Drag&Drop or the context menu to copy transactions into the favorites' area. The context menu is opened by a right-hand mouse click.

2. Add the following transactions to the favorites menu without expanding previously in the SAP menu: RSPFPAR, SEARCH_SAP_MENU, SEARCH_USER_MENU

   a) Then select *Favorites → Insert transaction*in the menu bar. Enter the desired transaction in the dialog box and save the data. This process must be executed for every transaction individually.

Continued on next page

3. Can you distinguish between favorites' entries by the method used to insert them?

   a) Yes, when inserting with previous selection in the menu, a part of the path information of the menu access is stored as a favorites text.

4. Create a folder with the title *Search Options* in which you can store both search transactions.

   a) Choose *Favorites → Insert file*. Name the file *Search Options*. Then, drag both search transactions into the file using the mouse.

5. Find out how often and where in the SAP menu the transaction for the user overview is stored.

Transaction	Path in the SAP menu
SM04	
SM04	
SM04	

   a) Start the search transaction SEARCH_SAP_MENU. In the dialog box, enter SM04 and start the search. Transaction SM04 is stored a total of three times in the SAP menu. In the same way, you can now search in the user menu.

Transaction	Path in the SAP menu
SM04	*Tools → Administration → Monitor → System Monitoring → User Overview*
SM04	*Tools → Administration → Monitor → Performance → Exceptions/User → Active User → Local User*
SM04	*Tools → CCMS → Control/Monitoring → Performance → Exceptions/User → Active User → Local User*

Continued on next page

Task 3: Using F1 and F4 Help

You can gain additional information on ABAP objects using the F1 and F4 helps.

1. Find out which system table the transaction codes of your SAP system are stored in! To do this, use the dialog input field to enter a start transaction in the *SAP Easy Access* menu settings. (*Menu bar → Extras → Set start transaction*)

   a) You can get this information by calling up the F1 help in the dialog box to enter the start transaction. Then branch to the technical information. The table is called *TSTC*.

2. You already know transaction SM04. Are there other transactions whose technical name also begins with "SMO"? Research this in the *TSTC* table with the help of the *Data Browser*, transaction SE16.

   a) Start transaction SE16. Enter *TSTC* as the table name. Start the display of contents using function key *F7*. In the input field *TCODE* type in "SM0*" and call up the F4 help. You will find two other hits apart from SM04. SM01 and SM02

Task 4: Searching for Data Records

You should search for a dataset directly. For example, you want to list certain authorization profiles delivered by SAP.

1. In the *USR10* table, search for all profiles delivered by SAP.

   a) Start transaction SE16 and display the *USR10* table. The (last) processor is shown in the *MODBE* field. Enter SAP* and DDIC there as multiple selection on single values. Display the number of entries by activating the press button *Number of Entries*. You will find several hundred entries.

Hint: The display of the number of hits is mostly limited to a certain value, for example, a maximum of 250 or 500 hits. If you want to display more entries, increase the value in the *Maximum no. of hits* field.

Continued on next page

Task 5: Filtering Data (Optional)

You want to subsequently restrict or filter a large number of hits in the list display.

1. In the *USR10* table, display all of the composite profiles delivered by SAP that have been changed since 1999.

   a) Start transaction SE16. Call up the user parameters from the application function list; alternatively, you can also use the *F6* function key. On the *Data Browser* tab page, activate the *ALV grid display*. Save this setting. Now call up the *USR10* table. Carry out a multiple selection on the changers DDIC and SAP*. In the *Type* field, enter "C" for "Composite Profile" and display all associated data records.

2. Filter the data in such a way that only the profiles from the year 2000 are shown.

   a) Highlight the header line of the *MODDA* column. Then set a filter over this column, use the  key in the application function list. In the date field, enter the date 1.1.2000 as the lower limit. This alone would only search for profiles from 1.1.2000. Therefore, set a selection operator. Call up the selection options using *F2* key. Select the operator *greater than or equal*. Now you see only the profiles, which have been changed since 2000. Typically, you see at least both profiles SAP_ALL and SAP_NEW.

Task 6: Where-Used List for a Data Element (Optional)

You want to research which tables and structure fields a certain data element can be found in.

1. In which tables and structures is the data element P_Action found? Begin your search in the Repository Information System, transaction SE84.

   a) Start transaction SE84. Use a mouse click to open the *ABAP Dictionary* file. Double-click the entry *Data elements*. A selection screen appears on the right-hand side of the screen. In the *Data element* field, enter **`P_Action`** and start the search using the  key or function key *F8*. In the hit list, select the single entry and activate the *Display* icon. Start the where-used list using *Utilities* → *Where-Used List*. Make your selection using table fields and structure fields. You will get exactly one hit. The data element P_Action is used in the AUTHB structure.

Lesson Summary

You should now be able to:

- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Carry out complex, selective searches
- Sort and filter listed data
- Execute a where-used list
- Describe the standard menus **System** and **Help**.

Related Information

- Online documentation: You can find extended help for *SAP Easy Access* under *Help → Application help*.

Lesson: Appendix - Personalizing the User Interface

62 Lesson Duration: 0 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. Generally speaking, everything you can set up without user logon is a change of the local layout and thus a personalization in the SAP GUI on the front end PC. All other activities (such as creating a favorites list, entering the start transaction) that become possible after logging on, are user specific.

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.

In the standard toolbar, use the button *Customizing of local layout*:

Using the *Customizing of local layout* pushbutton, you can 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). 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.

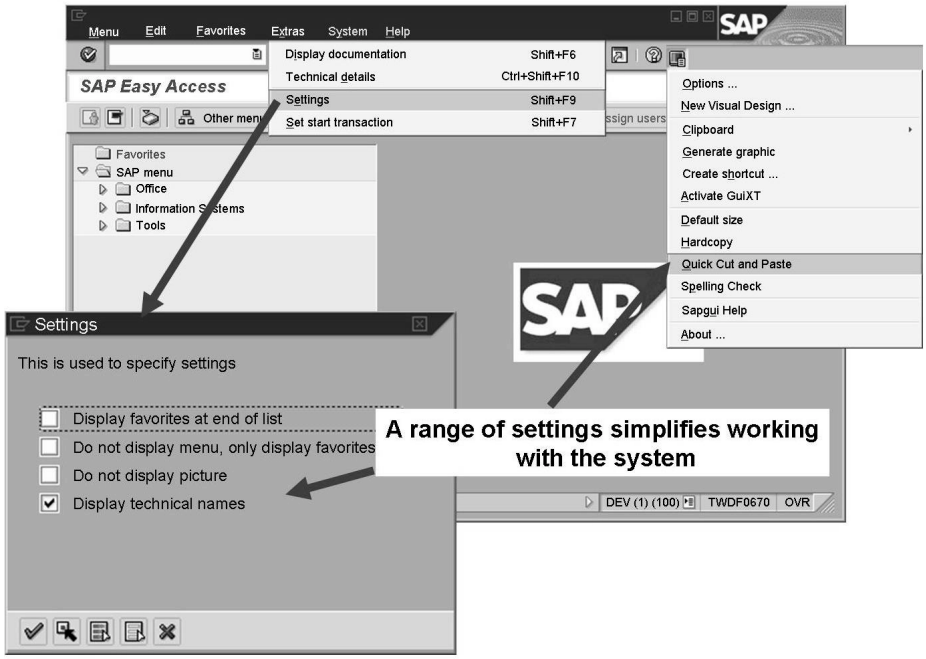

Figure 30: Two of the many personalization options

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 **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. There, you can, for example, always display the transaction code of the transaction that is currently being executed.

65

Exercise 4: Appendix - Personalizing the User Interface

Exercise Duration: 10 Minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Using basic personalization options.

Business Example

You want to personalize your interface, that is, you want to 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* function. 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: Appendix - 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* function. Now test this function on some fields in your *User Profile* (SU3). You can then use the same method to deactivate this function.

   a) Press the layout menu pushbutton and choose *Quick Cut and Paste*. 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
- List some variants of the *SAP GUI*
- Name and use elements of a *SAP GUI* screen
- Log off from the system successfully
- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Carry out complex, selective searches
- Sort and filter listed data
- Execute a where-used list
- Describe the standard menus **System** and **Help**.
- Use the various options for personalizing the SAP system

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. 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

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. 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 screen of your choice on the *SAP Easy Access* screen.

Unit 3

71

The System Kernel

This is a pivotal unit in this course. The terms of the ABAP and Java-based SAP Web Application Server are explained. Furthermore, it describes in detail how SAP systems process user requests. The process types of the SAP Web Application Server ABAP+Java are discussed and important concepts are explained. Topics such as printing and background processing are covered in course ADM100. 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.

Caution: Central topics such as SAP transaction, asynchronous update or lock mechanism are covered in the appendix of this unit. On the one hand, the appendix lessons are to be seen as optional. However, as subsequent courses in the administrative curriculum do not cover these topics in anymore detail, the instructor should briefly cover the content, if there is sufficient time or if the participants explicitly request this. However, if there is insufficient time, the participants can also gain this knowledge from autonomous study.

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
- Name the most important processes on an *SAP Web Application Server*
- Define the term instance and recognize the characteristics of a central instance
- Describe the processing flow for user requests in SAP systems

- Outline the processing flow for a dialog step in the SAP system
- Describe the concept of work process multiplexing
- Describe how the SAP system communicates with the database
- List the advantages of open SQL statements
- 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

Unit Contents

Lesson: Principal Architecture of the SAP Web Application Server


73

Lesson Duration: 75 Minutes

Lesson Overview

This lesson introduces the principal architecture of SAP systems. Based on simple client/server configurations, the individual processes of the *SAP Web Application Server* are introduced and the term of the SAP instance is defined. Following that, you will get an overview of how user requests are processed in SAP systems. This lesson concludes with an overview of the different installation options of an *SAP Web AS*.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline simple client/server configurations
- Name the most important processes on an *SAP Web Application Server*
- Define the term instance and recognize the characteristics of a central instance
- Describe the processing flow for user requests in SAP systems

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, in the *Additional Material* area). First you could, for example, set up a three-tier client/server architecture with several (ABAP) instances on different hosts. You can then assign an ABAP 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). This scenario can then be transferred to the Java stack and the central service instance can be explained.

Business Example

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

Introduction

SAP systems are used for mapping business processes or business applications. These applications should be implemented independent of the hardware environment used (operating system, database) to the greatest extent possible. For this, the *SAP Web Application Server* provides two runtime environments: the ABAP runtime environment and (as of *SAP Web AS 6.20*) also a Java runtime environment.

ABAP (Advanced Business Application Programming) is a programming language developed by SAP. Many business applications of an SAP system are written in ABAP. ABAP has been optimized for developing highly-scalable business applications. Customers can use the ABAP Workbench for developing completely new applications as well as enhancing and modifying SAP standard applications. In doing so, the entire, mighty infrastructure of the *SAP Web AS* can be used, which also supports the creation of the most complex applications by large groups of developers. *The SAP Web Application Server ABAP* provides the runtime environment of the programs written in ABAP.

Since *SAP Web Application Server 6.20* SAP not only provides an ABAP runtime environment for ABAP but also a runtime environment for **Java** programs. The integrated Java Web application server is a Java 2 Enterprise Edition (J2EE) application server called **SAP Web AS Java**.

The Java programming language was first introduced by Sun Microsystems Inc.™ in 1995. Java is an object-oriented and platform-independent programming language that has spread across many areas. The Java concept enables the development of a large range of different application types - from classical applications to applets used in websites to client/server applications.

Java 2 Platform Enterprise Edition (J2EE) is a manufacturer's standard for a whole range of software components that are primarily created in the Java programming language. Sun uses the J2EE compatibility test to ensure that the specifications of Java 2 Enterprise Edition are observed.

Hint: The purpose of this specification is to have a generally accepted framework for using modular components to develop distributed, multi-level applications. With the J2EE specification, Sun wants to ensure that conform Web applications run on all J2EE compatible servers. According to the J2EE specification the application logic is packaged in components, **Enterprise Java Beans** (EJB). They represent Java program components. A container implicitly provides the components with the services of the runtime environment.

The SAP Web Application Server 6.40 contains a 1.3 compliant J2EE Server.

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 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.

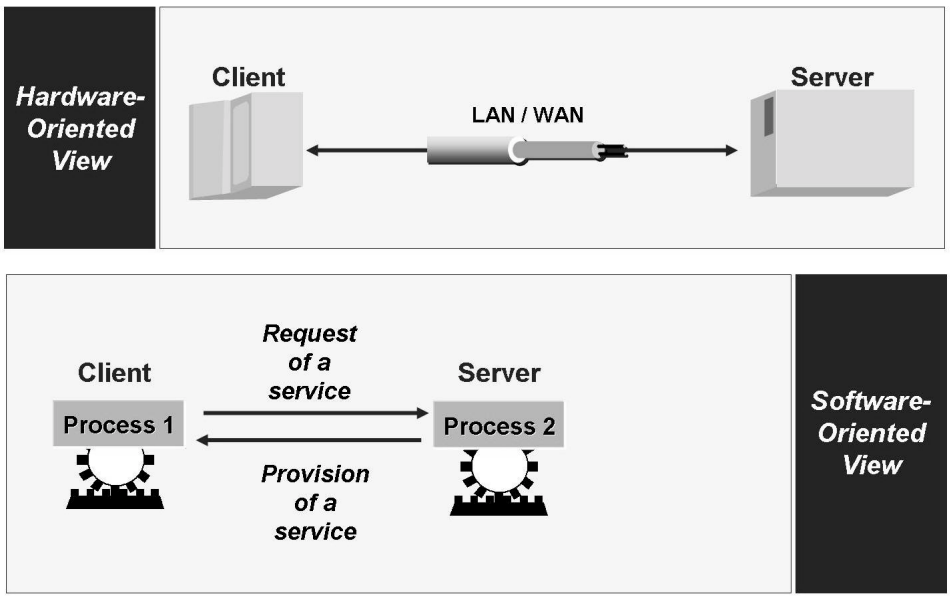

Figure 31: Hardware-oriented View and Software-oriented View

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

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 (in the ABAP environment).

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 in more detail below.

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 *Windows* PCs.
- In a three-tier configuration, each layer runs on its own host. Many different application servers can simultaneously work with the data of a database server.

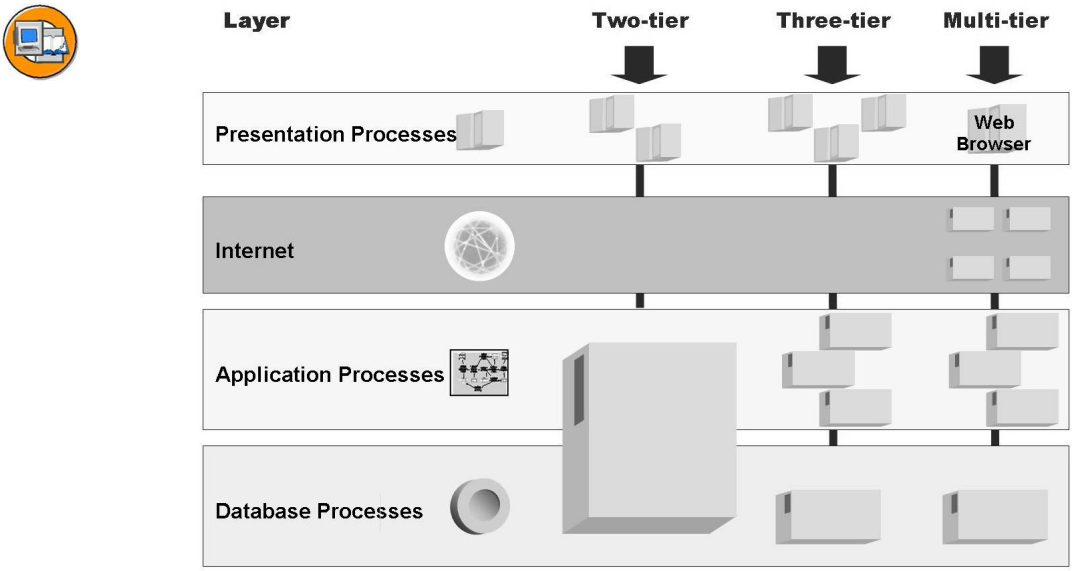

Figure 32: Simple Client/Server Configurations

Single-tier configurations are generally used for tests and demonstrations (for example, an SAP system on a laptop). If many users want to 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 levels (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 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. An additional level could be a web server, for example.

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 in the ABAP stack 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 ABAP 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.

According to the ABAP programming model, multi-layer architectures can be implemented for enterprise applications, whereby the application logic is executed on the (ABAP) application server.

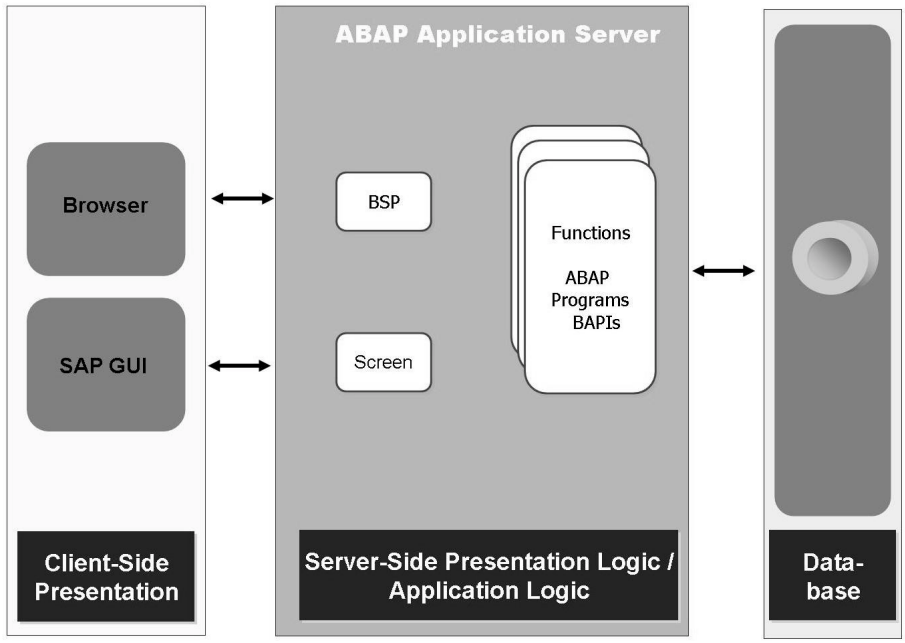

Figure 33: Architecture of ABAP Applications

The J2EE application server processes the queries of the clients, executes the application logic and returns the results to the clients. The application data is accessed at database level. The application logic is, for example, implemented in the form of ABAP programs or function modules, the server-side presentation logic consists of dynpros/screens (dynamic programs, consisting of a screen and its flow logic) or business server pages.

Hint: For developing, designing and implementing Web applications the *SAP Web Application Server* offers a page-based programming model with server-side scripting analogous to the well-known server-page technology, Business Server Pages (BSP). Server-side scripting allows direct access to all elements on the application server (for example, function modules, database tables, ABAP objects,...). Here, the presentation is separated from the business logic. This enables the use of alternative front-end technologies.

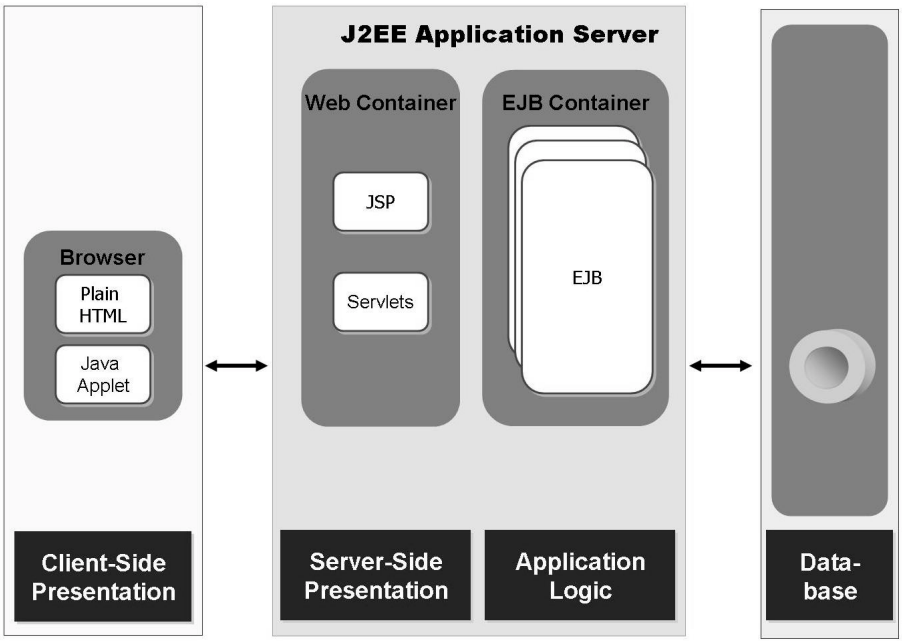

Figure 34: Architecture of Java Applications

According to the J2EE specifications, three-level client/server architectures for business applications can be implemented, the core of which is the J2EE Application Server (see graphic “Architecture of J2EE applications”).

At this stage, point out that for J2EE applications the server-side presentation is strictly separated from the application logic. This is a difference to “classical” ABAP programming, where the table in which the input values are written (application logic) can be specified at the same time as definition of the input fields of the screens (presentation logic).

The J2EE application server processes the queries of the clients, executes the application logic and returns the results to the clients. The application data is accessed at database level. The application logic is implemented in the form of Enterprise Java Beans. These are executed in a **EJB Container**, which is part of the J2EE Application Server. The communication between the client and the J2EE server is based on Web standards such as HTTP, HTML and XML. The J2EE-Server uses Java Server Pages (JSP) and Java Servlets, to generate HTML pages and XML data. Java Server Pages and Java Servlets are executed in a **Web Container**, which in turn is part of the J2EE Application Server. Within the HTML pages that are sent to the Web browser, Java applets can be integrated; these applets are executed in the browser.

Java Server Pages (JSP) is a technology developed by Sun, which is basically used for the simple dynamic generation of HTML and XML outputs of a Web server. The technology makes it possible to integrate Java code and special JSP actions into static content. That has the advantage that for this technology, the logic can be implemented independent of the design. Java Server Pages are converted to Java source code by means of a special JSP compiler. The Java compiler then converts this source code, which corresponds to a Java servlet, to byte code.

SAP Web Application Server Processes

The SAP runtime system consists of a number of parallel processes that work together. The following graphic illustrates the most important processes. Here, you can distinguish between the runtime environment for ABAP and the runtime environment for Java.

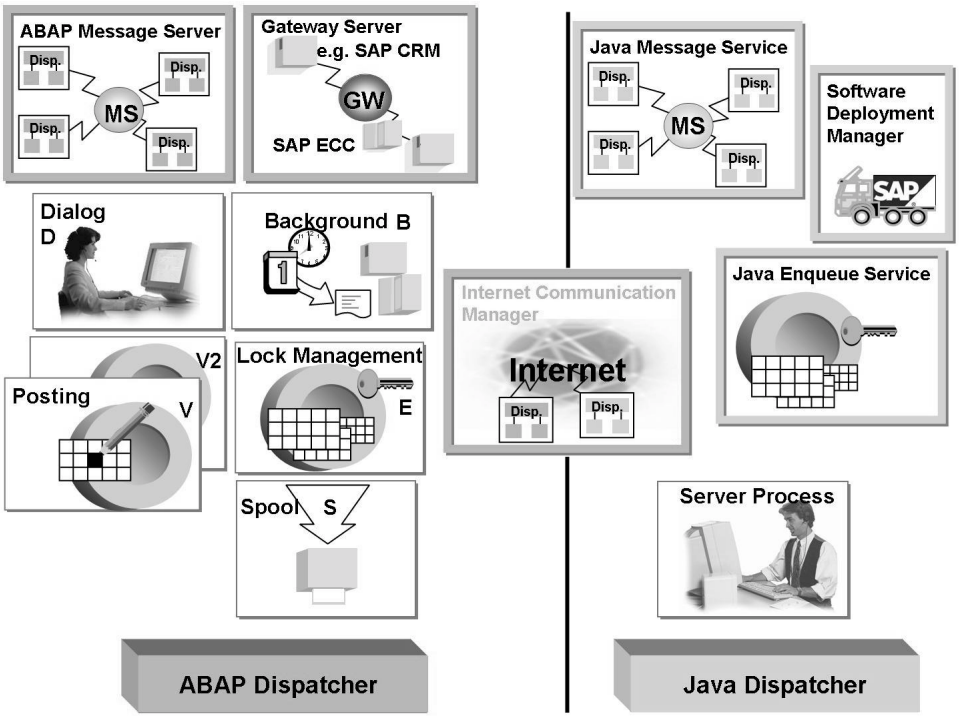

Figure 35: SAP Web Application Server Processes

In the **ABAP environment**, these processes on every application server include the dispatcher as well as a number of work processes depending on the hardware resources:

- The ABAP dispatcher distributes the user requests to ABAP work processes.
- 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.

This spool work process is also required if there is no printing in the system, for example, to store lists generated in the context of background process in TemSe.

- 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 two background work processes for each SAP system. You can configure more than one background work process for each dispatcher.

For the “normal” operation of the SAP system, a single background process would suffice. However, for an upgrade or during an import of ABAP transport requests, two background work processes are required.

- The enqueue work process administers the lock table in the shared memory. The lock table contains the logical database locks of the ABAP stack of the SAP system. Only one enqueue work process is needed for each system.

If more than one enqueue work process has to be configured, this should be configured on the same instance as the first enqueue work process as both work processes have to access the same lock table.

To sum it up: The dispatcher of an ABAP instance manages different types of work processes: dialog, update, background, lock and spool work processes. These work processes take on appropriate tasks when executing the business procedures in the SAP system.

Hint: The type and number of the work processes to be started for each instance is configured using the SAP system profile parameter. The following table summarizes the relevant profile parameters.

Type	Use	Profile parameter
DIA	Responsible for dialog requests	rdisp/wp_no_dia
UPD	Responsible for update	rdisp/wp_no_vb
UPD2	Exclusively responsible for less time-critical update (optional)	rdisp/wp_no_vb2
BGD	Executes background jobs	rdisp/wp_no_btc
SPO	Processes output requests	rdisp/wp_no_spo
ENQ	Responsible for locks	rdisp/wp_no_enq

Point out that this is only a selection of relevant dispatcher parameters. For lock mechanisms, for example, there are dedicated enque/* parameters.

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

- The message server (MS) handles the communication between the distributed dispatchers within the ABAP stack of 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) 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. In doing so, it recognizes whether the request is a call of a business server page or a Java application and forwards this request to the ABAP runtime environment or the Java runtime environment. 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 application server (software-based view).

The following processes exist in the **Java environment**:

- The Java dispatcher distributes new incoming Java requests to the Java server processes.
- The Java server process executes the Java applications. The server processes are implemented in the form of a Multi Threaded Server and can (in contrast to the ABAP work processes) process several queries in parallel.

It is for this reason that generally there are a lot less Java server processes configured per Java dispatcher than work processes are configured for each ABAP dispatcher. Due to port reasons a maximum of 16 Java server processes can be configured for each instance.

- The Java message service manages a list of Java dispatchers and server processes. It is responsible for the communication within the Java runtime environment.
- The Java enqueue service manages logical locks that are set by the executed Java application program in a server process.

Just like in the ABAP environment, every developer is responsible for requesting locks from the enqueue service and releasing them in the Java environment.

- The SAP Java Connector (JCo, not illustrated in the graphic) is responsible for the communication between the ABAP stack and the Java stack; it enables calls of the ABAP stack from the Java stack as well as vice versa.

Note: You can use the transaction SM50 (work process overview) to display an overview of the ABAP 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.

You can get an overview of started Java processes (Java dispatcher and Java server processes as well as SDM) in the same way, via the system information of the Java runtime environment (http://<Hostname>:<Java port of the instance>, for example http://twdfxxxx.wdf.sap.corp:50000→ *System Information*)

At this point you can demonstrate the current configuration of the local ABAP instance using transaction SM50. You could also, if appropriate, give a brief explanation of the (optional) V2 update work process. In the same way you can also demonstrate the configuration of the Java instance (optional). You require an appropriately authorized user in the Java stack to display the system

information. More precisely: On the client to which the SAPJSF user created during the installation points, there must be a user, which has been assigned the *SAP_J2EE_ADMIN* role in transaction SU01 (This role only exists for an SAP system that has an ABAP stack and a Java stack). For the SAPTEC course, the user *javauser* with password *java* has been set up for the demonstration.

The Instance

The section begins with the definition of the term instance for the SAP Web AS ABAP. Clearly define the term central instance. Make sure participants have understood this term before extending the instance to the SAPWeb AS ABAP+Java environment.

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 Application Server are often used as synonyms.

The Instance in the ABAP Environment

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 distinguished from all other instances of the SAP system and is also called **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.

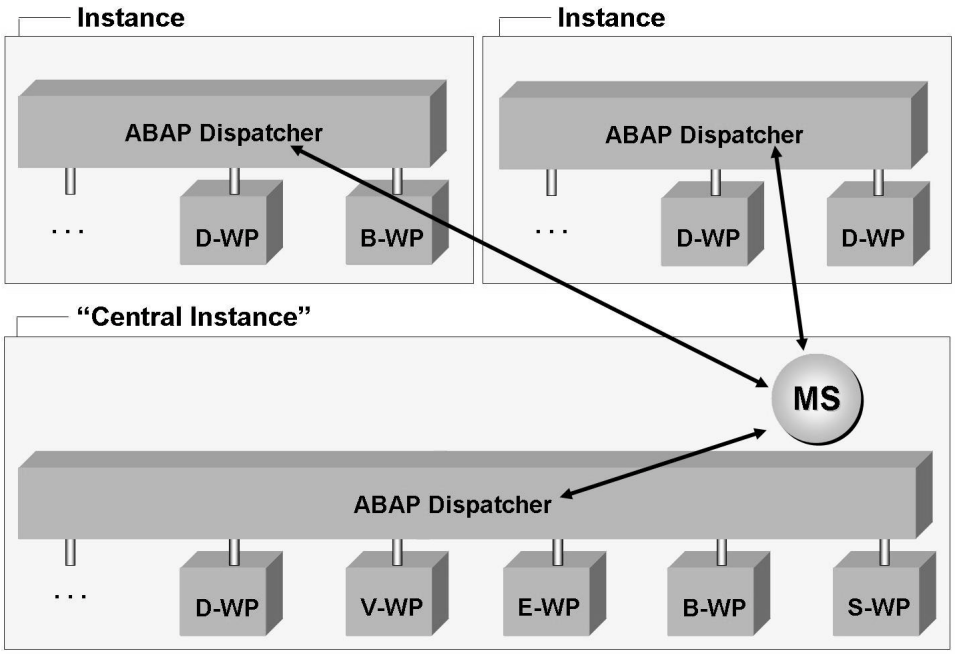

Figure 36: The Instance (using the ABAP Runtime Environment)

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

Presentation servers use the (ABAP) 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 find an overview of all instances configured in your SAP system in the *SAP Easy Access* menu under *Tools → Administration → Monitor → System Monitoring → SAP Server* (Transaction SM51). From here, pushbutton also provide links to the user overview (transaction SM04) and to the work process overview (transaction SM50) for the respective *SAP Web Application Server*.

The Instance in the ABAP+Java Environment

After the instance in the ABAP environment has been explained in the preceding chapter, this chapter now describes the integrated version of the *SAP Web Application Server* with ABAP and Java runtime environment. In this case, the term instance refers to ABAP processes as well as Java processes. The instance includes the ABAP dispatcher and its work processes and the *Internet Communication Manager* as well as the Java dispatcher with its Java server processes.

Note: The integrated version of the *SAP Web Application Server* is also called “Add-In Installation”.

The following graphics illustrate the architecture of such a system.

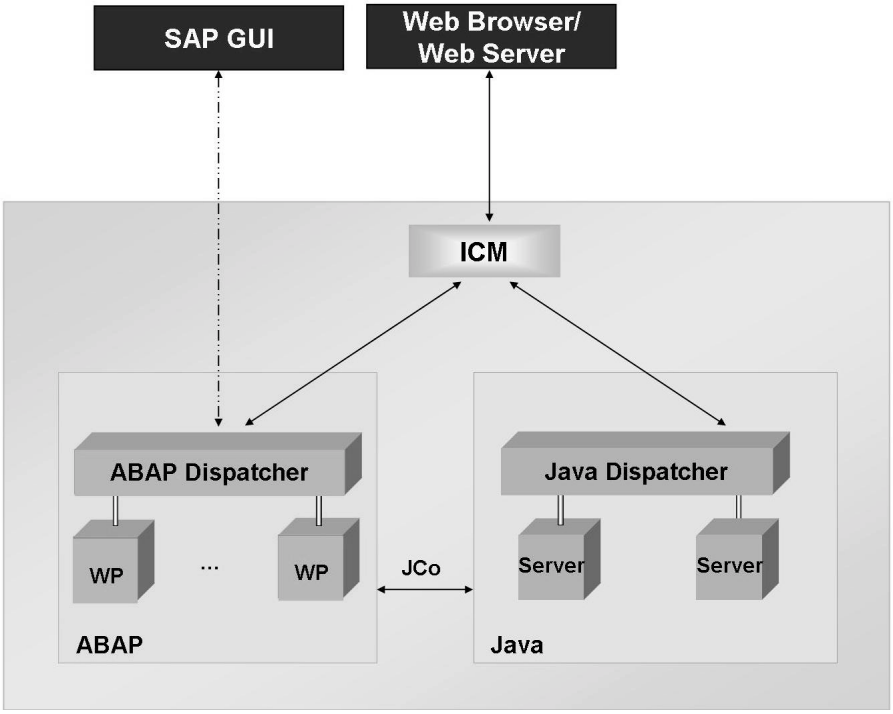

Figure 37: The Instance in SAP Web AS ABAP+Java

The following graphic shows the architecture of an SAP system with several instances as well as ABAP and Java runtime environment.

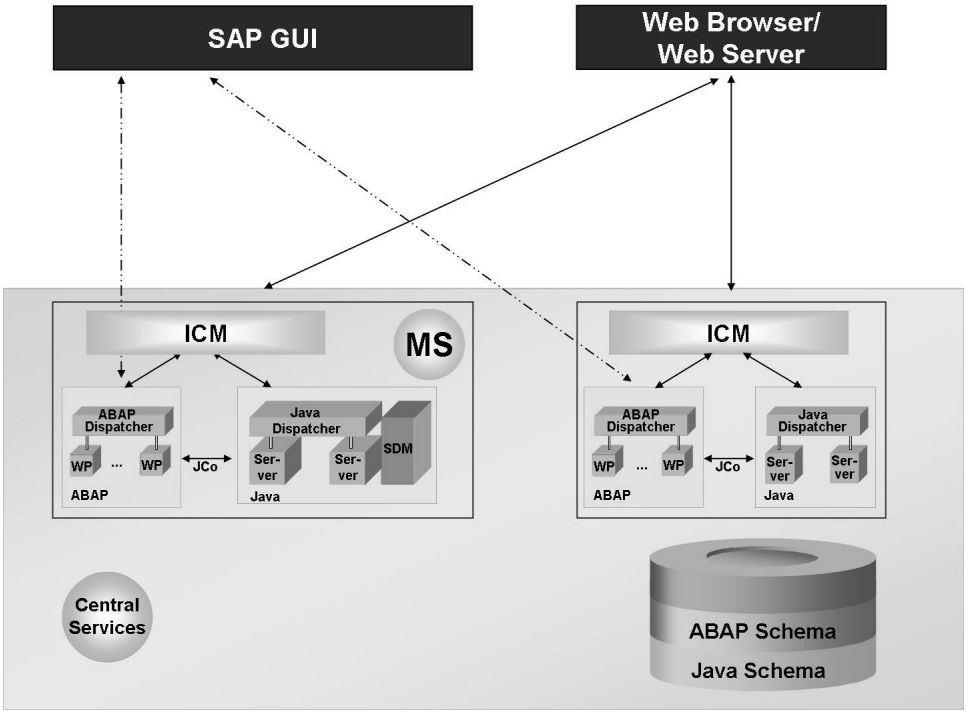

Figure 38: Structure of an SAP Web Application Server with Several Instances

In an *SAP Web AS ABAP+Java* with several instances, each of the integrated instances of the *SAP Web AS* can contain the (ABAP) dispatcher and its work processes as well as the Java dispatcher with its server processes. One of the instances is generally installed as a (ABAP) central instance, so it also contains the enqueue work process. Alternatively, the system can be set up without the central instance but with the stand-alone enqueue server. Furthermore, a common database contains a schema for each of the applications and data for ABAP and Java.

Stress that the database of the SAP Web AS has two different schemas: The ABAP schema, which can be accessed by the ABAP work processes and the Java schema, which can be accessed by the Java processes. The ABAP and Java runtime environment communicate by means of the AP Java Connectors (JCo) between work processes and Java server processes.

The **Software Deployment Manager** (SDM) is the standard tool with which the J2EE components are installed on *SAP Web AS Java*. It is part of the Java runtime environment and is always installed on the Java part of the central instance.

The **central services** run on one computer and form a separate instance. They consist of the message service as well as the enqueue service and represent the basis of communication and synchronization for the Java runtime environment:

As background information for you: As a preparation, look at the file structure of the training system (e.g. by means of transaction AL11)

- The SAP Web AS Java is, for example, stored in directory `G:\usr\sap\<SID>`.
- The instances (ABAP+Java) are each stored in a separate directory:
  - Central instance: `z.B. G:\usr\sap\<SID>\DVEBMGS00`
  - Dialog instance: `z.B. G:\usr\sap\<SID>\D01`
- The Java part of an instance is relative to these instance directories in directory `~/j2ee`
- The Java dispatcher and the server processes are in directory `~/j2ee/cluster` in directories `dispatcher` and `server<n>`.

Note: The term cluster has grown historically and is not correctly used in the place of the directory name. In SAP Web AS 6.40, a Java cluster refers to all Java components of an SAP system.

- The central services (previously SAP central services) are in the directory, for example: `G:\usr\sap\<SID>\SCS07`.

Types of the SAP Web AS

Depending on the requirements on the *SAP Web Application Server* there are different possible features, depending on which installation routine is used (see the following graphic).

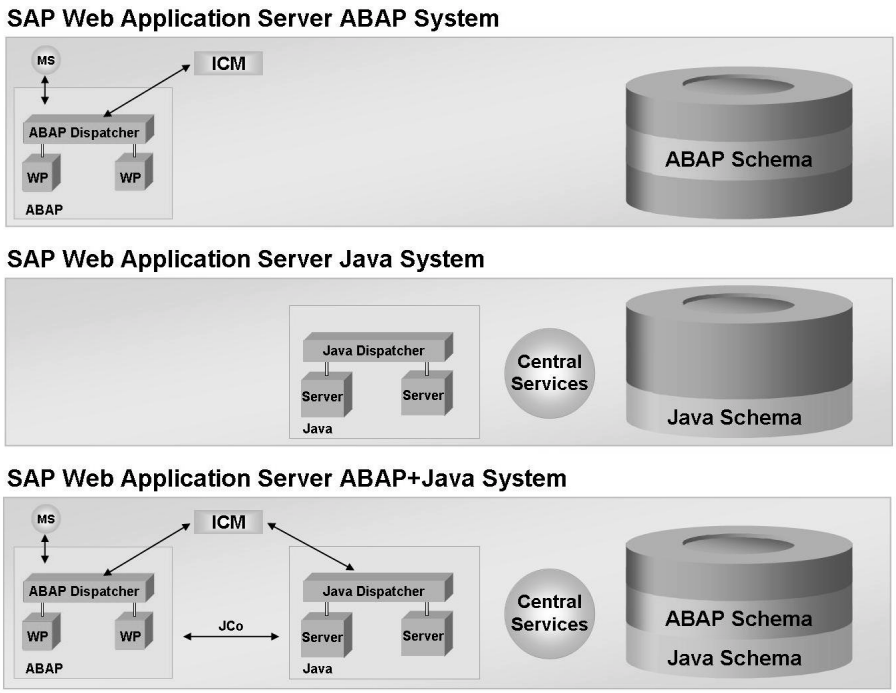

Figure 39: Possible Types of the SAP Web AS

Quickly point out the different installation options of the SAP Web AS. Stress that the option chosen generally depends on the SAP system to be installed. Therefore, when an enterprise portal is installed, an SAP Web AS Java is installed, whereas when an exchange infrastructure is installed, you need to install an ABAP stack as well as a Java stack. In contrast, when you install an SAP Business Information Warehouse the ABAP stack is mandatory while the Java stack is optional.

The graphic only shows the minimal configuration (that is, one instance).

- **SAP Web Application Server ABAP System:** Complete infrastructure in which ABAP-based applications can be developed and used.
- **SAP Web Application Server Java System:** Complete infrastructure for developing and using J2EE-based applications.
- **SAP Web Application Server ABAP + Java System (Add-In Installation):**Complete infrastructure in which ABAP-based and J2EE-based applications can be developed and used. This installation emphasizes the seamless Java ABAP integration.

One of the main characteristics of the *SAP Web AS* is that ABAP tables, programs and application data are stored in the ABAP schema of the database while Java data is stored in the Java schema. Here, the ABAP runtime environment can

access the ABAP schema of the database, and the Java runtime environment can access the Java schema. The SAP Java Connector is used for integrating the ABAP environment and the Java environment. (JCo).

Processing User Requests in the SAP System

Clearly point out to participants who already know about the classical ABAP runtime environment that classical transactions, such as the displaying of customers by means of transaction FD03 still happen exclusively via the ABAP runtime. The existence of a Java runtime environment does **not** mean that precisely this function is now implemented by means of the SAP GUI as well as in a Java-based way. It always depends on the chosen function, which runtime environment (ABAP or Java) and which user interface (SAP GUI or BSP / JSP) have to be used in order to be able to access the desired function.

Users can logon to the SAP system either via the *SAP GUI* or via a Web client (browser). When logging on via the *SAP GUI*, user requests are processed by the ABAP runtime environment, when logging on via a browser, processing can be done by the ABAP or the JAVA runtime environment, depending on the request.

Processing Requests by the SAP GUI

If necessary, you can also talk about the necessity of buffer synchronization and the applied procedure when you introduce the buffers and their significance.

The participants should learn about the general procedure for a user request in the SAP system. In doing, only the case of an internal user (with SAP GUI) is discussed here. The integration of Internet users via the ICM is not the subject of this section.

Users logon via the (ABAP) message server (load balancing) or they logon directly on the ABAP dispatcher, the work processes execute the user entries. Another task of the message server is to transfer requests in the ABAP world.

The processing of a user request in the SAP system, as outlined in the graphic, involves different processes on all three layers (presentation, application and database layer):

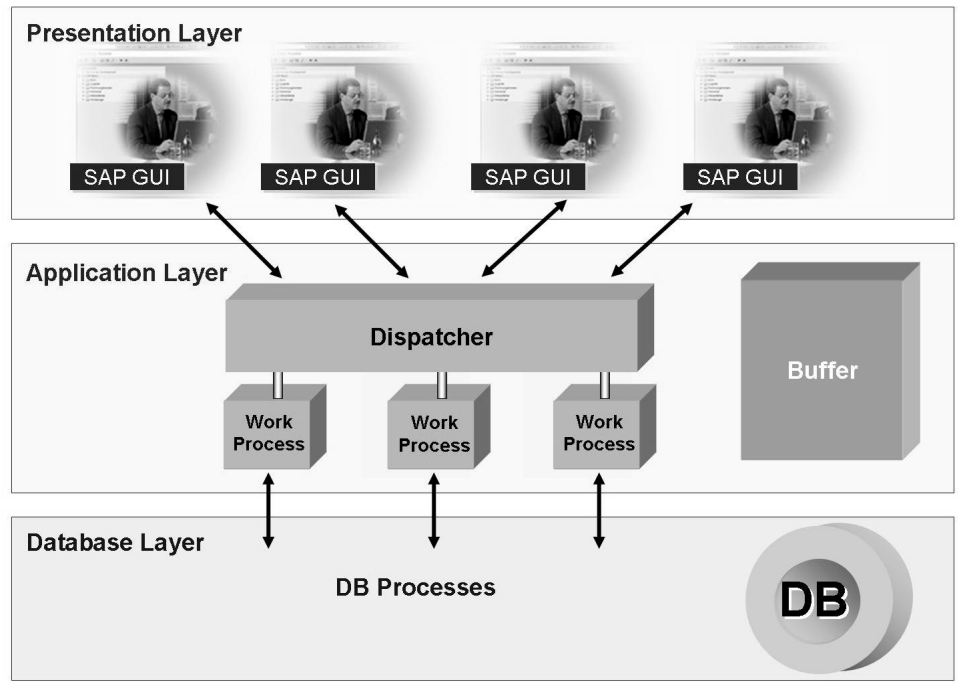

Figure 40: Processing a User Request

Explain that this graphic is a (detailed) excerpt from the above graphic on the structure of an SAP Web AS with several instances.

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 **(ABAP-) Dispatcher** is the central process of the *SAP Web Application Server ABAP*. It manages the resources for the applications written in ABAP in coordination with the respective operating system. The main tasks of the ABAP dispatcher include the distribution of the transaction load to the work processes, the integration of the presentation layer and the organization of communication transactions.

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

The ABAP 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. To process

user requests it might be necessary to read data from the ABAP schema of the **database** or to write to it. For this, every work process is connected directly to the ABAP schema of 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.

During initialization of the SAP system, the dispatcher executes the following actions (amongst others): Read system profile parameters, start work processes, logon to the message server.

Processing Web Requests

Web requests are accepted by an *Internet Communication Manager (ICM)*. These HTTP(S) requests can either be processed in the ABAP work process (such as BSP applications) or they can be requests for *SAP Web AS Java*. The *ICM* can use the URL to decide to where it forwards the request (if it cannot reply to the request from its cache).

You can describe the processing of Web request step-by-step using the above graphic “The Instance in SAP Web AS ABAP+Java”.

If the request is for the Java runtime environment, then it is sent to the Java dispatcher that forwards it to a Java server process. If necessary, the Java server process addresses the Java schema of the database and uses the Java dispatcher process to return its reply to the *Internet Communication Manager*. In the end, the *ICM* sends the reply of the SAP system back to the requesting user.

In contrast to ABAP, where there are different types of work processes (dialog, update, background...) for different types of tasks, the Java server process offers several services simultaneously (for example, *HTTP provider service*, *JDBC connector service*, *locking adapter service*). Depending on the type of user request, the Java dispatcher of an instance chooses a Java server process on which "suitable" services are configured.

In case of a request to the ABAP runtime environment the *ICM* forwards the request to the ABAP dispatcher, which treats this request like a classical request of a *SAP GUI* (see previous subsection). The response of the ABAP runtime environment is in turn, returned to the requesting user via the *ICM*.

If the participants ask at this point whether several computer/port combinations have to be released for external access in case of several instances, you can refer to the SAP Web dispatcher. The distribution of the HTTP(S) requests across all instances of an SAP system happens via the SAP Web dispatcher. It is a central entry point to the SAP system from the Internet. For every incoming request it has to decide to which instance to forward it. This lesson does not cover the architecture and configuration of the SAP Web dispatcher.

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 DEV)
Client: The client assigned to you (such as 100)
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 1: 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. Which instance are you working in? (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?

Task 2: Research System Parameters

Find out what parameter *rdisp/wp_no_dia* means. What is its default value?

1. Start transaction RZ11 and display the parameter. Call up the documentation using the *documentation* pushbutton or function key *F6*.

Continued on next page

Task 3: Research on the Internet Communication Manager

Find out about the configuration of the *Internet Communication Manager* on your instance.

1. Check whether the *Internet Communication Manager* of your SAP instance has been started. The *Internet Communication Manager* is started by means of parameter *rdisp/start_icman*.
2. Check, which protocols have been configured for your *Internet Communication Manager*.

Protocol	Port

Solution 5: Structure of an Instance

Task 1: 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. Which instance are you working in? (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.

Task 2: Research System Parameters

Find out what parameter *rdisp/wp_no_dia* means. What is its default value?

1. Start transaction RZ11 and display the parameter. Call up the documentation using the *documentation* pushbutton or function key *F6*.

   a) Call transaction RZ11 and, in the *Parameter name* field, enter the profile parameter (**`rdisp/wp_no_dia`**) you are looking for. Choose *Display*. The default value specified by the system is 2. However, your system administrator might have set another value in your SAP system (see *Current value* field). Now choose *Documentation*. You get a description of the meaning of the profile parameter (number of configured dialog work processes of the respective instance).

Continued on next page

Task 3: Research on the Internet Communication Manager

Find out about the configuration of the *Internet Communication Manager* on your instance.

1. Check whether the *Internet Communication Manager* of your SAP instance has been started. The *Internet Communication Manager* is started by means of parameter *rdisp/start_icman*.

   a) Start transaction RSPFPAR and select the parameter you are looking for. The process is started if the value is *true*.

2. Check, which protocols have been configured for your *Internet Communication Manager*.

Protocol	Port

   a) Start transaction SMICM. If the ICM is started, display the configured protocols using the *Services* button in the application toolbar or choose *Goto* → *Parameters* → *Display* Protocols HTTP, HTTPS and SMTP are displayed with the configured TCP ports.

Lesson Summary

You should now be able to:

- Outline simple client/server configurations
- Name the most important processes on an *SAP Web Application Server*
- Define the term instance and recognize the characteristics of a central instance
- Describe the processing flow for user requests in SAP systems

Related Information

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

Lesson: Dialog Processing in the SAP System

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. Requests in this queue are distributed by the dispatcher to available dialog work processes on a “first in, first out” (FIFO) 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:

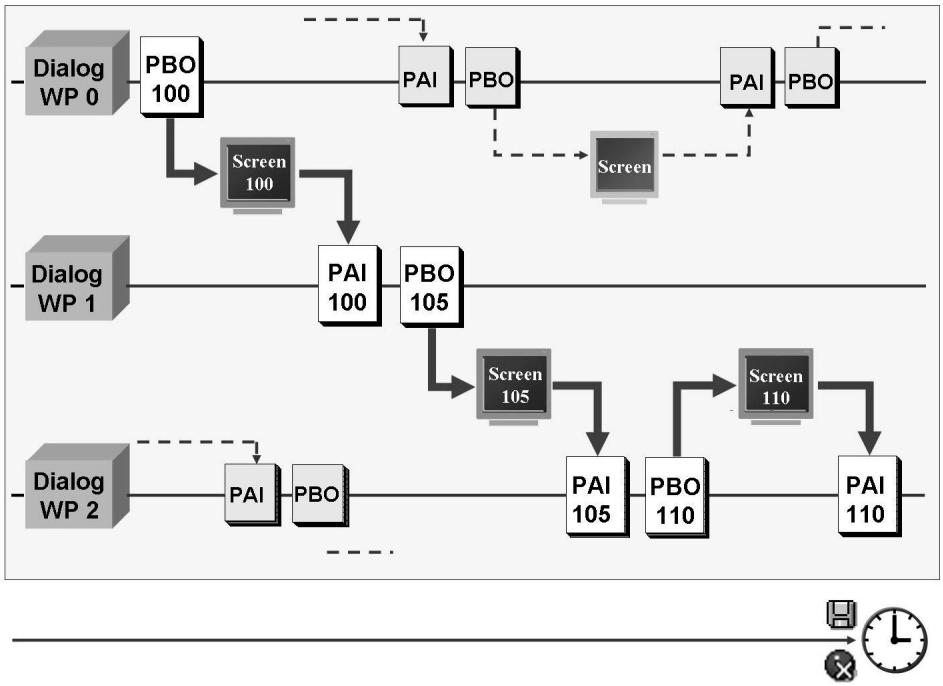

Figure 41: Work Process Multiplexing

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 kernel file, *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 dynamic programs), 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 Interpreter. The *screen* processor tells the ABAP processor which subprogram needs to be executed, depending on the processing status of the screen flow logic.

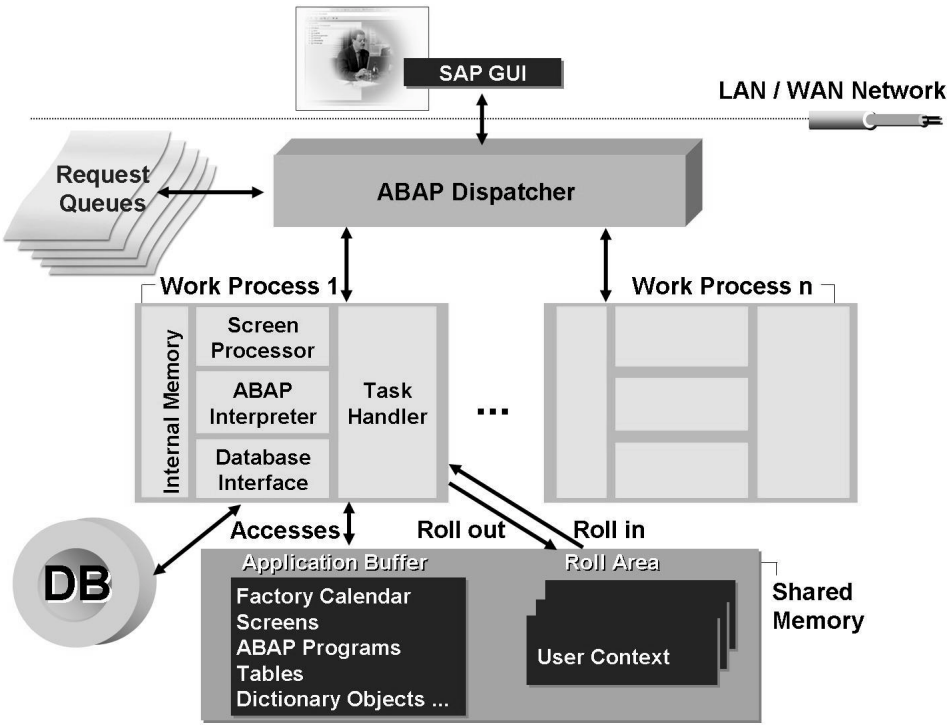

Figure 42: Processing flow for dialog steps

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).

99

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

Answer the following question using transaction SM50 (menu path: *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

Answer the following question using transaction SM50 (menu path: *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 of the entire SAP system, select transaction SM66 (here you have to make the correct settings using pushbuttons *process selection* and *settings*).

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: Communication with the Database

Lesson Duration: 30 Minutes

Lesson Overview

SAP systems can be used with a variety of operating systems and databases made by different manufacturers. The application data in the SAP system is accessed in the ABAP stack via platform-independent ABAP programs. But the data is stored database-specifically. Hence, the database queries from the ABAP programs have to be translated into the respective dialect of the database. This is taken care of by the database interface.

If in doubt, point out that this lesson explicitly covers the ABAP stack (and not the Java stack) of the SAP system. If there are any questions here, the Java stack of the SAP Web AS also provides options for platform independent development these days. Keywords: *Open SQL for Java*.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe how the SAP system communicates with the database
- List the advantages of open SQL statements

Try to go into as little detail as possible when it comes to individual database types. This lesson should be kept as general as possible.

Business Example

You want to find out how to access the database using Open SQL statements.

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 that also contains 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.

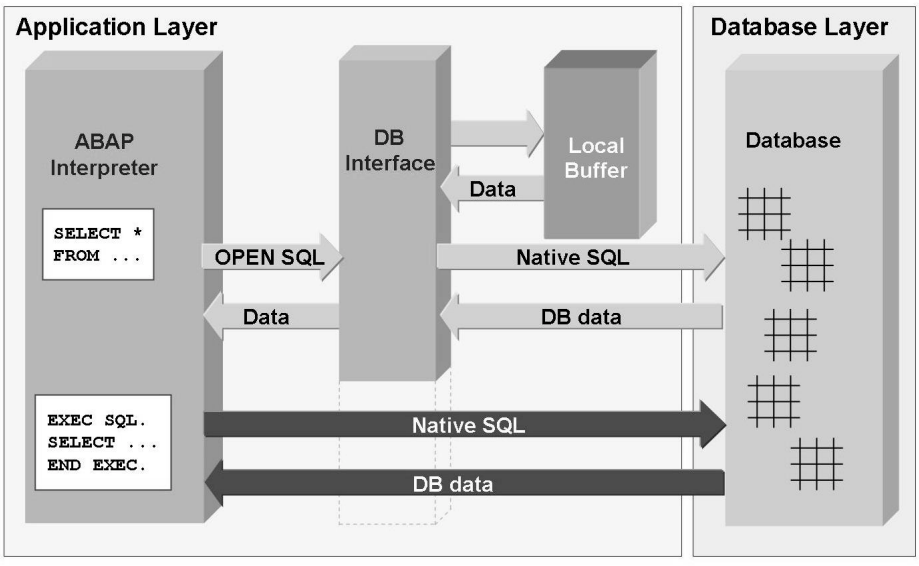

Figure 43: Database Query Flow

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 there is more than one instance then there is also 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 and also read frequently, 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.

Do you use Native SQL in your application programs? What are the consequences, for example, during a database migration?

Lesson Summary

You should now be able to:

- Describe how the SAP system communicates with the database
- List the advantages of open SQL statements

Lesson: Appendix - The SAP Transaction

105

Lesson Duration: 0 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

Atomic means that a transaction is either fully successful or does not have any effects at all. If a transaction-oriented system goes down, you need to ensure that inconsistent, partial results are not stored.

Consistent means that 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.

Isolated means that 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** because after the final confirmation they are stored 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 ("database commit") to the database system. During a database transaction (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 updates, 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.

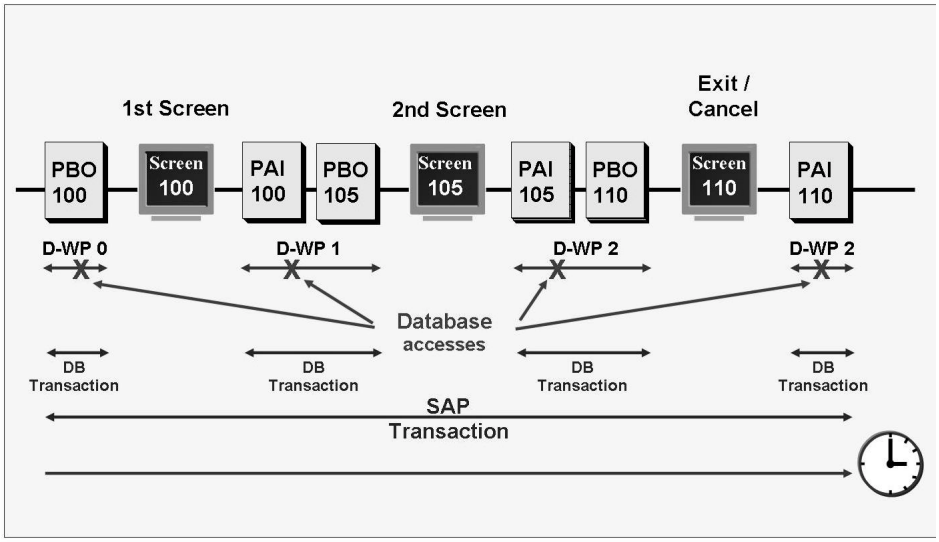

Figure 44: Relationship between database transactions and SAP transactions

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: Appendix - Lock Management in SAP Systems

109 Lesson Duration: 0 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 management**. 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.

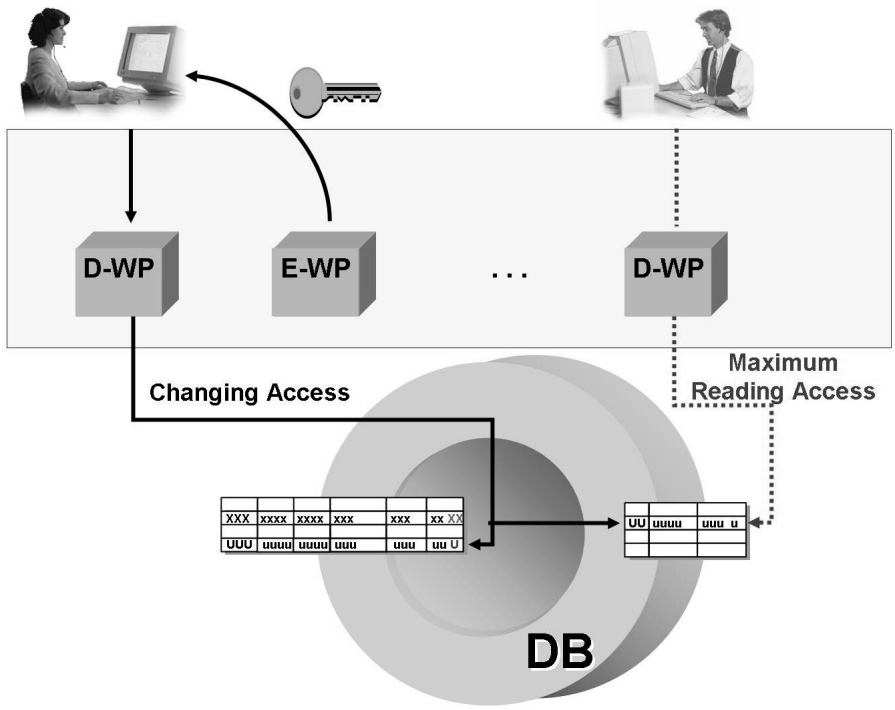

Figure 45: Fundamentals of enqueue processing in SAP systems

At this point you could show the lock management in the SAP system. For example, access transaction SU01 for an existing user and then show that a different user cannot access the same data in change mode but in display mode. (It is not possible to change from display mode to change mode either.)

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**. Here, we distinguish between exclusive locks and shared locks.

- Write locks (lock mode “X”); are only assigned if no other locks exist for the data records required; no additional locks are then permitted for these entries.
- Extended write locks (lock mode “E”); are only assigned if there are no locks on the data record yet. Only the lock owner can cumulatively assign further write locks.
- Shared locks (lock mode “S”); further shared locks – but no write locks – can be requested for this object.
- Optimistic lock (lock mode “O”)- available since *SAP Web Application Server 6.40*. The first lock types are pessimistic locks. The optimistic locks are intented for the case that data is displayed in change mode without being changed as well. If any changes are made, the conflict with possible changes that have been made at the same time is only determined and resolved during saving.

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 respective name:

```
ENQUEUE_<Sperrobjektname>
DEQUEUE_<Sperrobjektname>
```

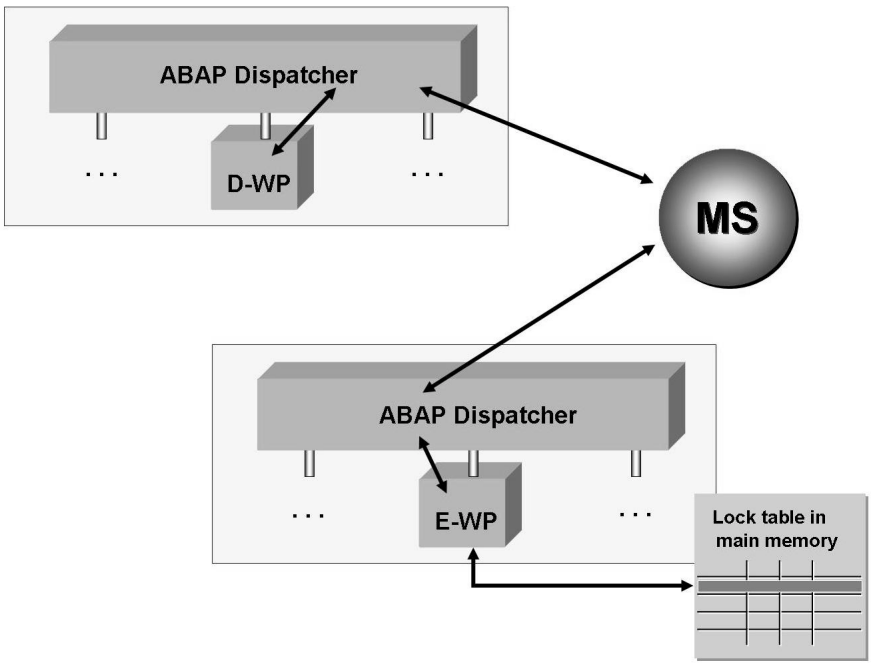

Figure 46: Requesting a Lock Entry

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 or *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.

External Stand-Alone Enqueue Server

The company HP has been offering the HP Somersault product, which also offers redundantly set-up, external lock servers, for some time. HP Somersault only runs on HP-UX, and the SAP systems must either run on HP-UX or Windows Advanced Server 2003. Refer to the Hewlett Packard homepage for more details.

With the *SAP Web Application Server 6.40* SAP also offers the SAP Stand-Alone Enqueue Server. Customers wishing to design the SAP lock mechanism for high availability can move the administration of the SAP locks to external servers. For reasons of high availability, the external lock servers have been designed redundantly. The SAP Stand-Alone Enqueue Server is available for all operating systems supported by SAP. Read SAP note 524816 for more details.


115

Exercise 7: Appendix - Lock Management

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 or 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: Appendix - Lock Management

Task: Activities

Learn about how lock administration works in SAP systems.

1. Call transaction SU01 or 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: Appendix - Update Processing

118 Lesson Duration: 0 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 be retractable 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".

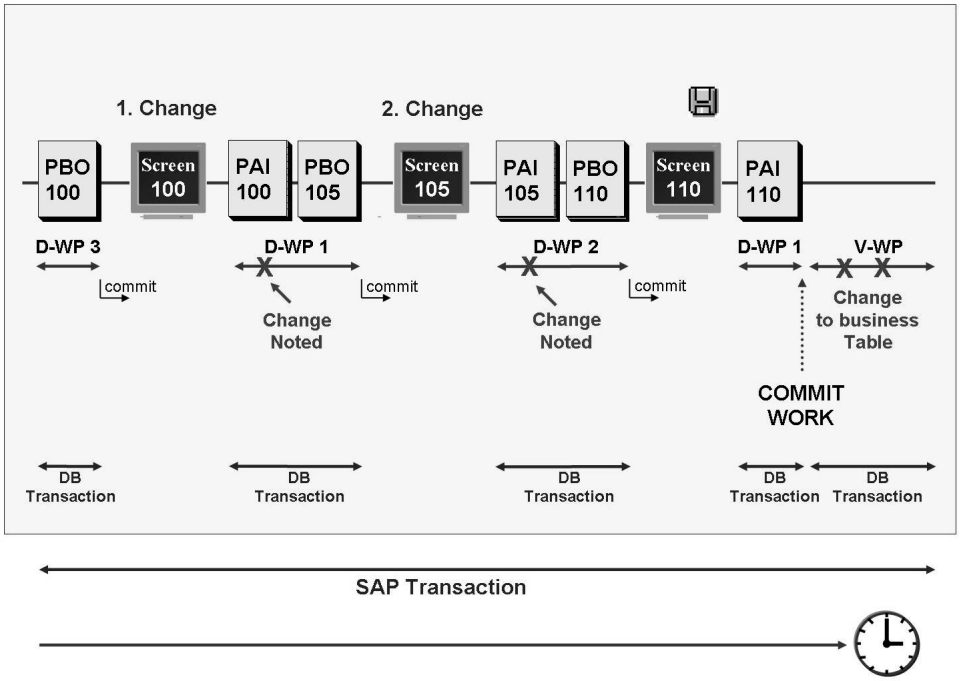

Figure 47: The principle of asynchronous updates

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 (using 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).

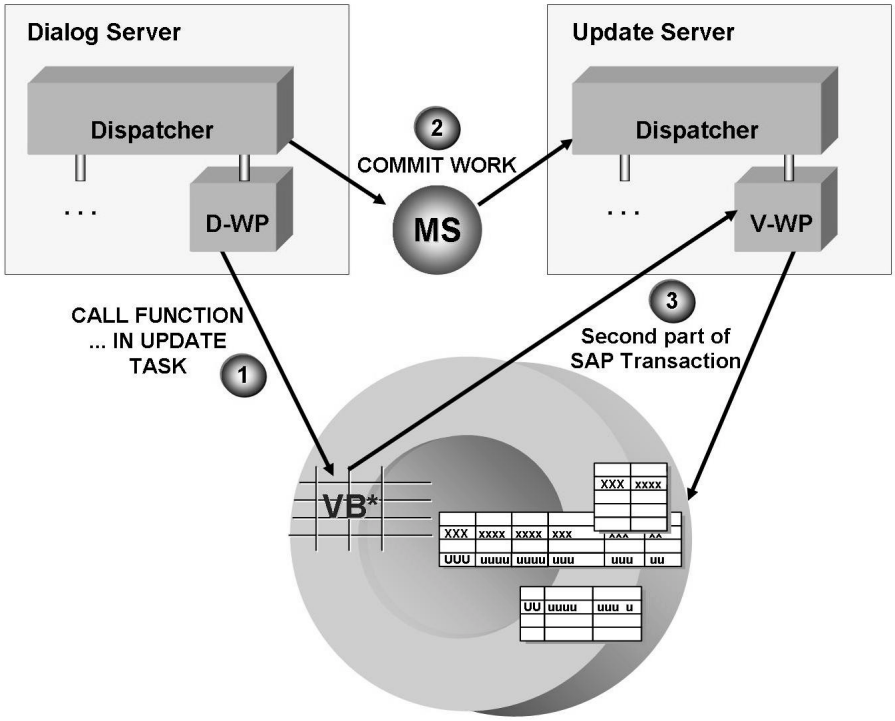

Figure 48: The asynchronous update process

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 mail 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.

Exercise 8: Appendix - Update Procedure

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: Appendix - Update Procedure

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 (menu path *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

Unit Summary

You should now be able to:

- Outline simple client/server configurations
- Name the most important processes on an *SAP Web Application Server*
- Define the term instance and recognize the characteristics of a central instance
- Describe the processing flow for user requests in SAP systems
- Outline the processing flow for a dialog step in the SAP system
- Describe the concept of work process multiplexing
- Describe how the SAP system communicates with the database
- List the advantages of open SQL statements
- 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

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 Buffer
   - □ D SAP GUI

3. 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

4. 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

5. 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

6. 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

7. 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

8. 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.

9. 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`

10. 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

11. 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

12. 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

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. 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.

4. 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.

5. 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.

6. 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.)

7. 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*.

8. 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 in a way that makes sense from a business point of view. The concept is independent of performance questions.

9. 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.

10. 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.

11. 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.

12. 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.

Unit 4

133

Software Development in SAP Systems

This unit can only be introductory in nature (partly because of the many more advanced courses in the ABAP and Java area).

First, explain the basic differences between the development strategy in the ABAP environment (central developments using the ABAP Workbench) and the development philosophy in the Java environment (decentral development, every developer has his/her own local development and runtime environment). Point out that the development process is different for ABAP and Java applications - due to the different concepts.

Following that, discuss the ABAP stack of the system first. The participants need to understand that – due to the data structure in SAP systems – they must not carry out development activities in production systems. Hence, repository objects written in development systems or Customizing have to be transported to other systems (this is the motivation for the term transport request).

Caution: To demonstrate the development process in the ABAP environment to participants, first use transaction SE80 to create a package in the customer name space (such as **`ZSAPTEC<client>`** and assign it to transport layer **`Z<SID>`** before the beginning of the course. Following that, create a transportable Workbench request in transaction SE09 and assign the participant groups of the course to this request. In the context of the execersises, the participants can thus assign their own developments to this order and, following that, you can demonstrate the release of the change request.

You don’t have to go into too much detail in the lessons on 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.

Unfortunately, the lesson on developing Java applications has to remain theoretical as it is not possible to provide a local development and runtime environment or central Java development infrastructure in the context of the course. The aim of this lesson is that participants - especially in decentral development as intended by Java - recognize the necessity of a central transport infrastructure, without being confused with the details of the structure of individual components (DTR, CMS, CBS, SLD). The quick link */jdi* in the SAP Service Marketplace also helps you to prepare for this lesson, which is especially challenging for instructors.

Unit Overview

Once you have installed and configured a software environment, you have to continuously adapt it to changing requirements. Requirements change, for example, when your company organization changes, an additional function is required or when there are quality improvements to SAP components, which SAP delivers at regular intervals.

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

This is followed by a brief introduction to the development environment for the Java stack of your SAP system. Here, you see that developers tend to expect a local development environment in the Java environment, while there is a central development environment in the ABAP environment (in the form of the *ABAP Workbench*). The different development philosophies for ABAP and Java can also affect the respective transport infrastructure.

Unit Objectives

After completing this unit, you will be able to:

- Describe the data structure of the ABAP Stack in an SAP system
- 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 ABAP Repository objects
- Write a simple ABAP program
- Explain the significance of the *ABAP Dictionary*
- Outline the two-level domain concept
- Name the individual components of the SAP NetWeaver Java Development Infrastructure

- Outline the concepts of a multi-system landscape for the Java development process

Unit Contents

Lesson: Data Structure of an SAP System and Transports between SAP Systems (ABAP Stack)

135

Lesson Duration: 50 Minutes

Lesson Overview

This lesson introduces first of all the data structure of the ABAP Stack in SAP systems. The lesson explains this data structure and and its extensive impact on the way you develop your own objects and adapt SAP objects. The data structure requires a multi-system landscape. The development objects are synchronized here among the different system of the landscape with the help of change requests using so-called transports.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the data structure of the ABAP Stack in an SAP system
- Describe the three-system landscape as recommended by SAP
- Explain the importance of transports
- Describe the procedure for transports

This lesson covers the data structure of the ABAP Stack in SAP systems. It introduces the data types “client-specific data”, “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. These cross-client Customizing settings include the public holiday calendar, for example, and settings for online documentation. Repository objects are ABAP Workbench development objects such as program objects, function group objects, Dictionary objects, and so on.

Subsequently, 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 customer developments, customer exits, and modifications.

At the end of this lesson, the steps involved in finalizing a development project are demonstrated. This includes releasing both the tasks and the change request as a whole in the Transport Organizer. In addition, the software development cycle is briefly discussed.

Caution: In preparation before beginning the course, you should already have created a package in the customer name space using transaction SE80, to which all newly created development objects are to be assigned

in the following. Similarly, in preparation before the course, you can either already have created a transportable workbench request that contains all participants as employees or you enter this change request in the course of this lesson in front of the participants.

Business Example

As an ABAP 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

The ABAP Stack in SAP systems is characterized by 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.

Hint: In this lesson, the term *SAP system* always refers to the ABAP Stack of an SAP system.

In SAP systems, the following terms must be differentiated with regard to the data structure:

- Client-specific data
- Cross-client Customizing
- Repository

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. For example, 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, 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 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 linked development objects (programs, tables, and so on). Packages are characterized by the properties nesting, interfaces, visibility and use accesses. The ABAP Repository contains packages from all software components.

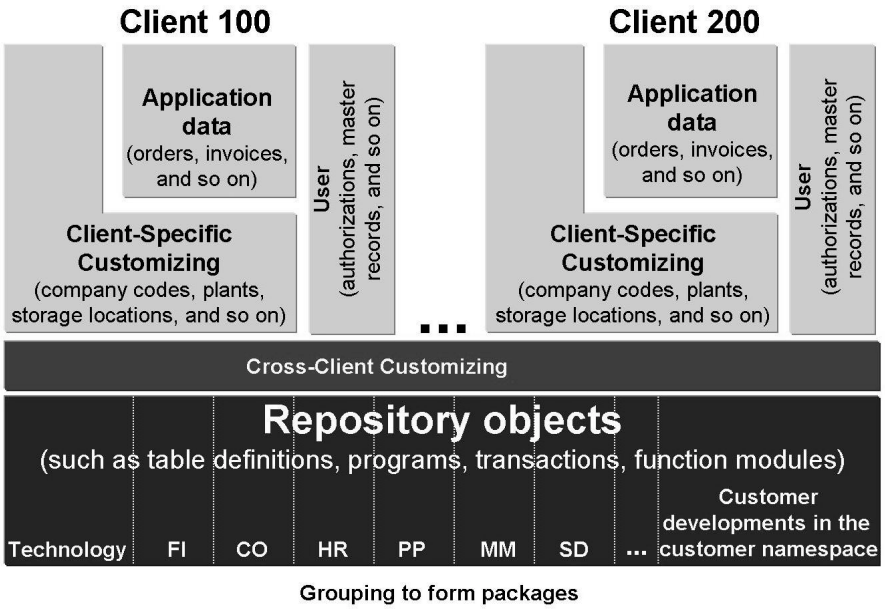

Figure 49: The Data Structure of the ABAP Stack in an SAP system

You can also create your own overhead transparency with the various data components to show the data structure of the ABAP Stack in an SAP system. To do this, you can 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.

Draw attention to the fact that the table structure (i.e. the definition of a table) is in the Repository (i.e. is cross-client), whereas the contents of this table can belong to the application data (i.e. can be client-specific).

Changing and Adapting the Data Structure

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

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 have a special introductory guide for this in the SAP system: the SAP Reference IMG (to be accessed using transaction SPRO). You may need to repeat Customizing on a much smaller scale when you upgrade your SAP system.

Changes to Repository objects are not absolutely necessary for the use of an SAP system. If necessary, however, you can change individual objects. Changes or extensions to the Repository are possible in a variety of ways:

- Customer developments
- Customer enhancements
- Modifications

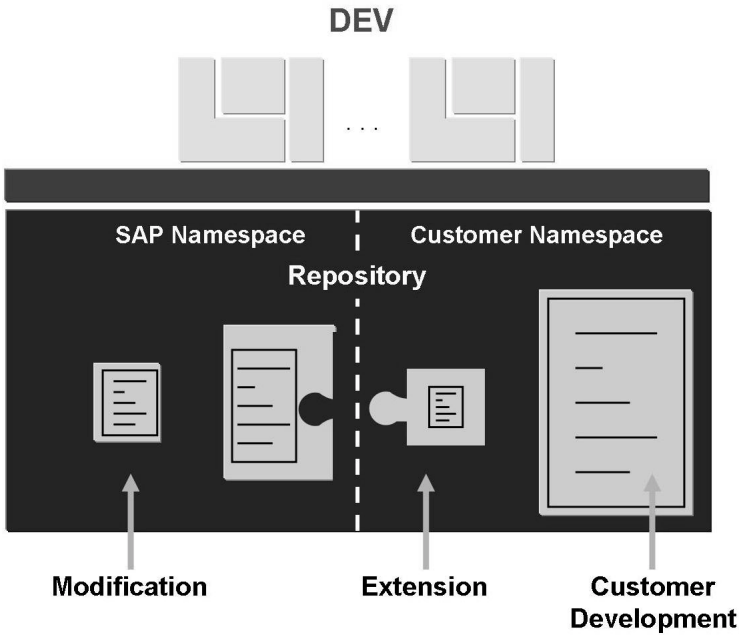

Figure 50: Adjustments to the ABAP Repository

Expanding the Repository through **Customer 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 applies to 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 **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 modern method for the customer-specific enhancement of the standard: the use of so-called **Business Add-Ins** (BAdIs).

If the participants ask about the enhancements in the system, in the Repository Information System (transaction SE84), you can show them the *Enhancements* area, if you like.

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 adjustment takes time, although using the **Modification Assistant** speeds up the process substantially.

The Three-System Landscape

Make it clear that the necessity of a multi-system landscape is a logical consequence of the data structure of the ABAP Stack (as it became clear in the last section that the customer can make changes to cross-client data).

There are consequences from the data structure of the ABAP Stack in an SAP system described above - especially when changing data. Since the ABAP 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. Also taking into consideration security factors (and perhaps performance aspects, if the developers have to debug programs), developers must not work on a productive system.

To ensure system consistency, SAP recommends 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.

Optionally, you can now discuss the advantages and disadvantages between two and three system landscapes with the participants. Explain that, in any case, a stable test environment is required, which the developers cannot influence by permanently changing of the Repository. When using a two-system landscape (in which the test system naturally takes place in the development system), a stop in development would be necessary in order to extensively test the development. If this is not practical, then a three system landscape is definitely required.

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

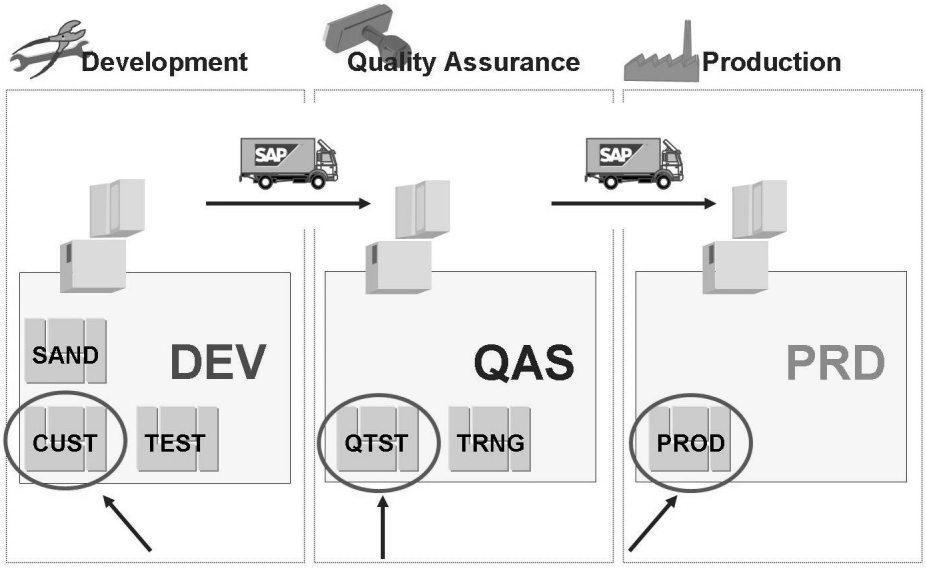

Figure 51: Three System Landscape in the ABAP Environment

Transports in the ABAP Environment

After explaining the necessity of a multi-system landscape in the previous step, the question now arises as to how a transfer of development objects can take place between the DEV and QAS systems. This section provides the answer to this question.

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**.

Work on Repository objects are transported and logged by the Transport Organizer (transactions SE09 or SE10, menu path *Tools → ABAP Workbench → Overview → Transport Organizer*) as Workbench requests. The same approach is used to log and transport Customizing settings as Customizing requests.

The following procedure is selected for the transport of objects: When a development project starts, the person responsible for the development project 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 (in the <SID>K9<nnnnn> format, so for example, DEVK900050). A request should contain 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.

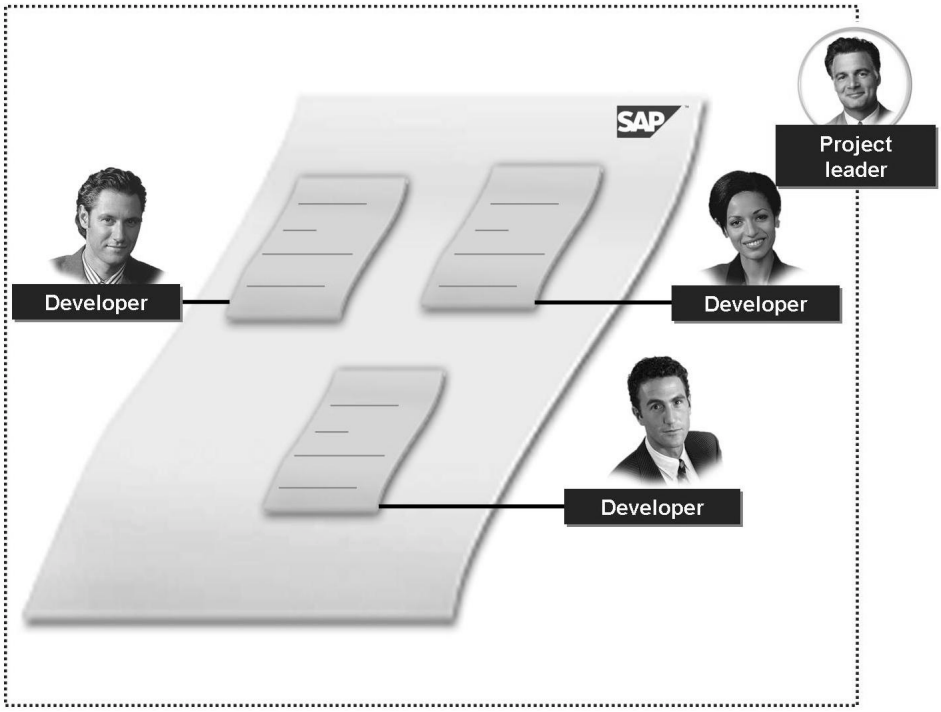

Figure 52: Change Request and Assigned Tasks

The Transport Organizer automatically creates a task for this change request for every employee who is 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. The task collects all of the Repository objects that this team member has worked on as part of the change request.

At this point you could show the following system demonstration:

1. Call up the Transport Organizer via *Tools → ABAP Workbench → Overview → Transport Organizer* (or transaction SE09).
2. Create a change request in the Transport Organizer (request type: Workbench request) and enter a description. Assign all participant groups to this request. Point out the difference between Customizing and Workbench requests.
3. Show the participants the request and its subordinate task(s).

At the same time, create another Workbench request and assign only yourself to it as a user. This Workbench request shall be released at the end of this lesson (without including the development objects).

Actions at Close of Development

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.

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 protocol for the change request. You can display this log even after the transport has taken place.

Import into the target system is in general not 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.

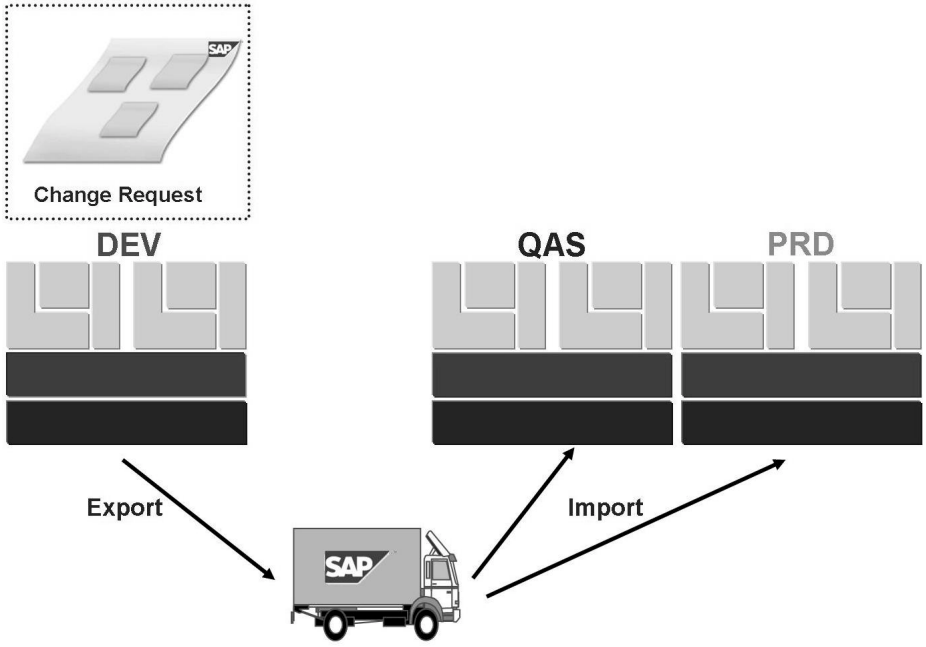

Figure 53: Export and Import into Other Systems

At this point you could show the following system demonstration: However, point out to the participants that the release of an “empty” transport request (as you are now demonstrating) does not make sense in practice and this only serves to explain the procedure at close of development. As the creating Repository objects has not been discussed at this point, only an “empty” request can be released here.

Call up the Transport Organizer via *Tools → ABAP Workbench → Overview → Transport Organizer* or transaction SE09, and show the change request created above that only has you as a user and expand this up to the task. Then release the task and the request using *Release directly*. (For this, it could be that you have to change the type using *Request/Task → Change Type → Development/Correction* (as you have not yet developed anything into this task as a developer, the type first of all is still *unclassified*. As soon as you would have developed something, the type of task would also have changed here) If necessary, you have to 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.

Showing the import queue for follow-up systems or even starting the import in a follow-up system (with a correctly set up TMS) goes far beyond the aim of this lesson and should therefore not be shown.

The Software Development Cycle

You can map and carry out the entire ABAP software development cycle in the SAP system.

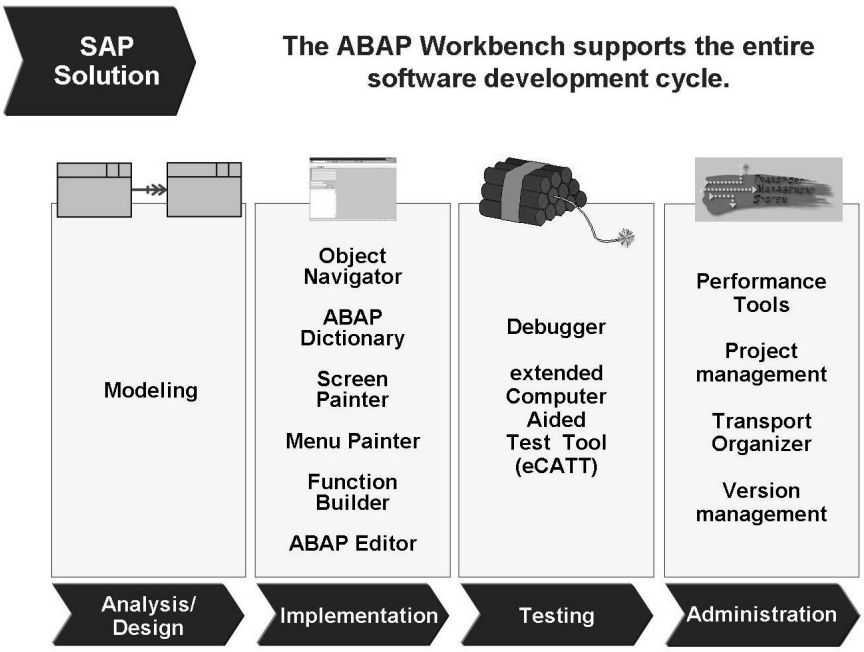

Figure 54: The Software Development Cycle

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.

Facilitated Discussion

Discuss with the participants the necessity of a multi-landscape system.

Discussion Questions

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

Why can you not develop and work productively on one and the same SAP system (there is a client encapsulation concept)? It should be clear here that the client encapsulation concept only applies to client-specific data and not to cross-client data (cross-client Customizing and Repository changes). If a developer writes and executes an erroneous program in the development client, this could have immediate effect on the productively used client.

Lesson Summary

You should now be able to:

- Describe the data structure of the ABAP Stack in an SAP system
- 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*.
- You can find further information about enhancements and how to deal with modifications in the training course BC425 – *Enhancements and Modifications*.

Lesson: Accessing and Editing ABAP Repository Objects

144 Lesson Duration: 75 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.

Let's briefly introduce the ABAP programming language first. You will see how you can navigate into the source code shipped by SAP. You will also write your own short ABAP program.

Following that, the meaning of the *ABAP Dictionary* is explained. Here, a table is used to talk about the domain concept.

Lesson Objectives

After completing this lesson, you will be able to:

- Name important tools in the *ABAP Workbench*
- Access ABAP Repository objects
- Write a simple ABAP program
- Explain the significance of the *ABAP Dictionary*
- Outline the two-level domain concept

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 in the ABAP environment.

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**.

The prerequisite for this lesson (in particular the exercises for this lesson) is that you have already created a package and a workbench request to which you have assigned all participant groups as employees.

This lesson also 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. For illustration purposes, it might be useful to create a table as a showcase.

Business Example

As an ABAP developer, you need to familiarize yourself with the most important ABAP Workbench tools and create your first ABAP program. Display the table definitions and contents as well.

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 erster_report.

WRITE 'Mein erster ABAP Report!'.
```

contains two statements, one on each line. The keywords are `REPORT` and `WRITE`. The program displays a list. In this case, the list contains the line “My first ABAP report!”.

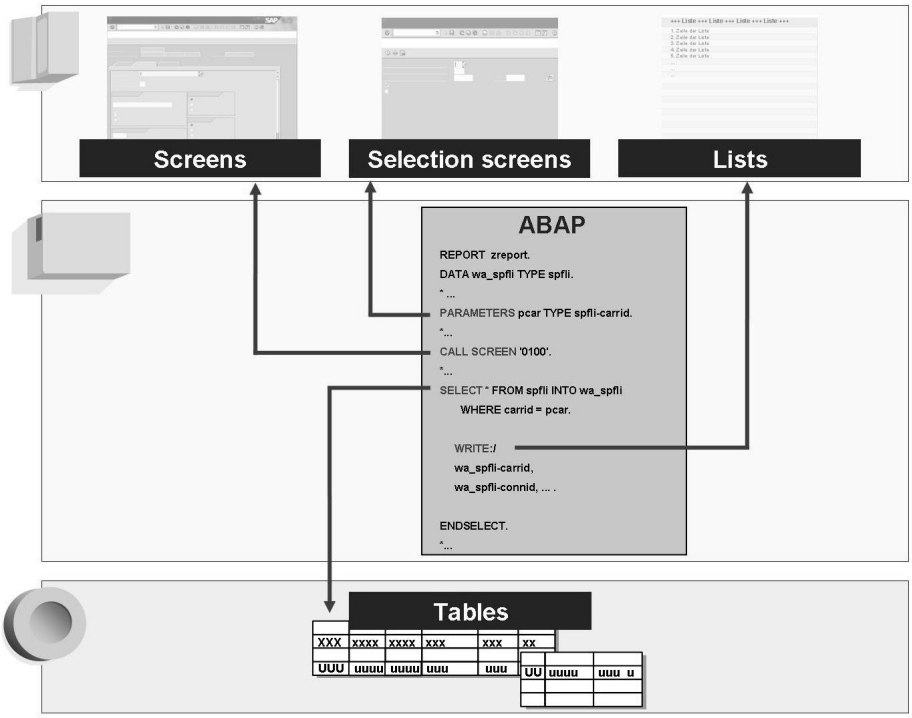

Figure 55: The ABAP Language

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 (transaction SE24), 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 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

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

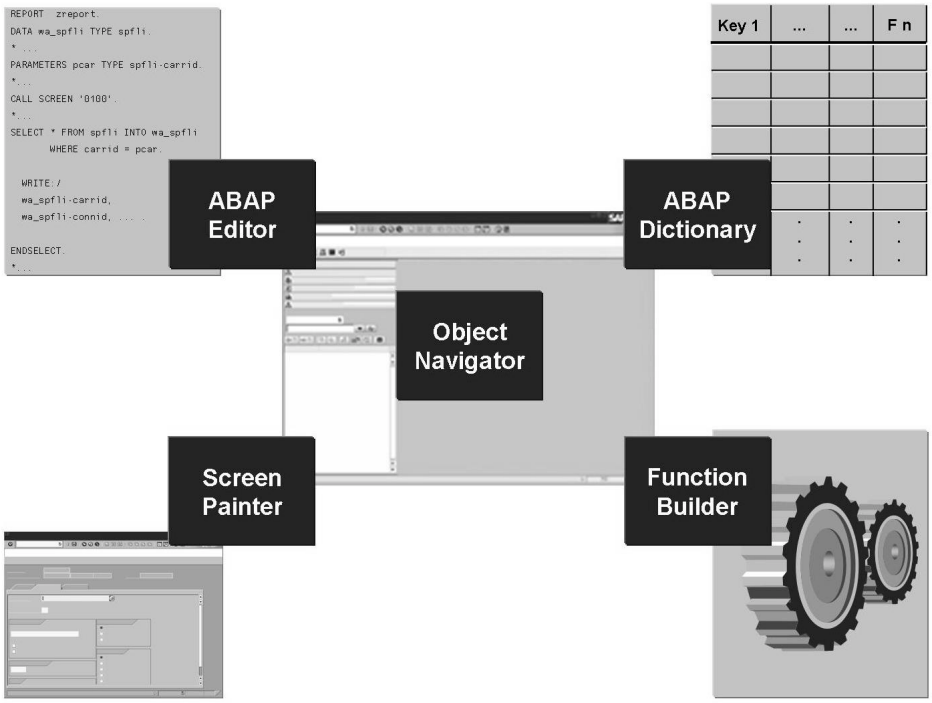

Figure 56: Some ABAP Workbench tools

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. By selecting an object and subsequently double-clicking, the tool is started from the workbench in which the object was created. Following that, you can edit this 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

Hint: In Releases prior to *SAP Web AS 6.10* packages are still called development classes, which is why SAP systems still have the data element and domain *DEVCLASS* (for defining the data element and domain, see below)

SAP provides the *Object Navigator* (transaction SE80) to help you organize your development processes in your integrated environment *ABAP Workbench*. 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 ABAP 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).

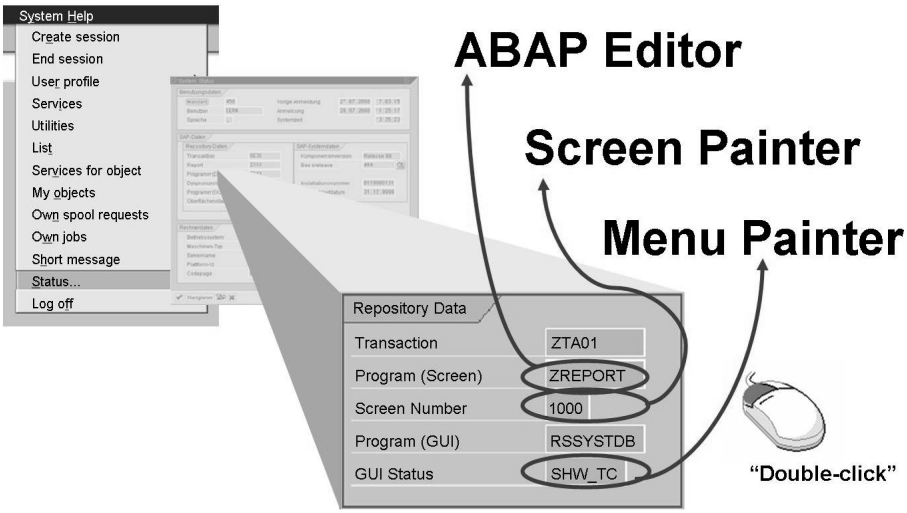

Figure 57: Navigating into the source code

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

After talking about the ABAP language and general navigation into the source code, own Repository objects are created in the following section. Here, the ABAP editor is introduced for creating ABAP program, the Object Navigator for creating Business Server Pages and transaction SE11 (ABAP dictionary) for the definition of dictionary structures. Alternatively, you can also demonstrate how an ABAP program is created in the Object Navigator.

You can use the *ABAP Editor* (transaction SE38 or link in the *Object Navigator*, transaction SE80) to create and edit programs. ABAP programs are not stored in the SAP system as external 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.

If necessary, tell your participants that the package concept has replaced the formerly used development classes. This is not just pure renaming, as packages can be nested and have use explanations as well as defined interfaces to the outside.

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 that 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 .

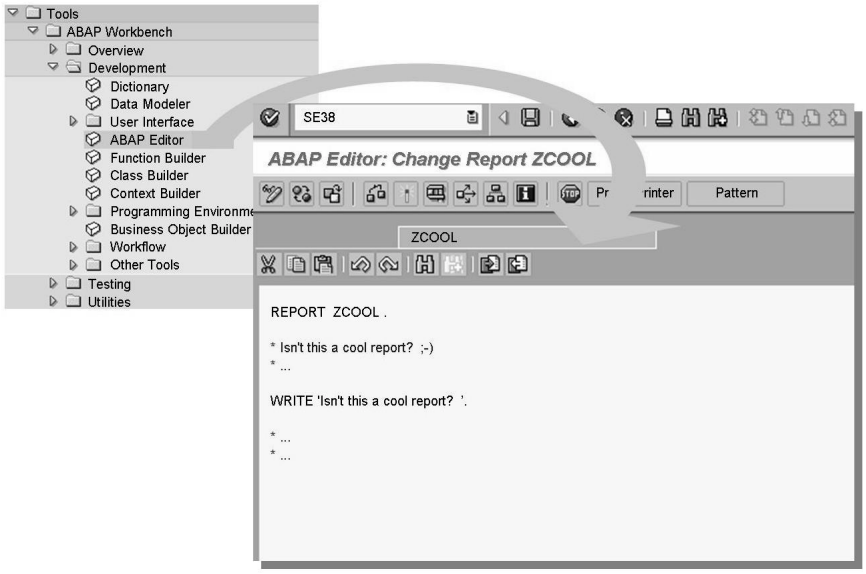

Figure 58: Editing ABAP Reports Using the ABAP Editor

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. Assign it to the package and the change request you have already created in preparation for this lesson. 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)*.

Creating Business Server Pages using the Object Navigator

The *Internet Communication Manager* 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.

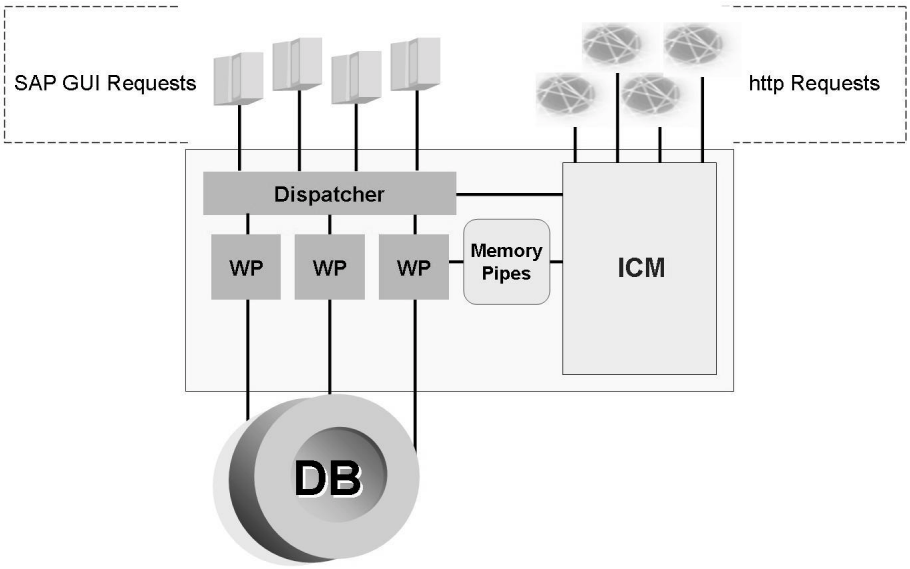

Figure 59: The SAP Web Application Server

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 extension 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.

Hint: You should confuse Business Server Pages with Java Server Pages (JSP), which are executed in the Java runtime environment of the SAP Web AS.

Create a Business Server Page now. To do so, choose the BSP application entry in the Repository Browser of the Object Navigator (transaction SE80) and enter the name of you BSP in the field below (e.g. Z<CLNT>DEMO_BSP). Choose Return, confirm the subsequent popup and enter a meaningful description. Assign the BSP to the package and request created before (you have to confirm the request twice, as an entry in the Internet Communication Framework, ICF, is created in addition to the BSP).

Right mouse click on the object name and choose *Create → Page* (choose *page with flow logic*) and enter the name of the page (for example **`mypage.htm`**) as well as a short description. On the next screen you can either use the already existing source code (simple save the object and activate all objects offered) or replace the previously entered text (such as “hello world”) with your own text.

Next, execute the page by choosing *Test/execute (F8)*. To do so, log on to the SAP system from the browser using your user ID.

Hint: In order to execute the BSP successfully, the corresponding service must have been activated in transaction *SICF* and the fully qualified name of the application server (such as **`twdfxxxx.wdf.sap.corp`**) must have been entered in the *application server* field in transaction SE80 under *Utilities → Settings...*.

Optionally, you can use the appropriate pushbutton to go from the Object Navigator to the Repository Information System, which we you can find information on the content of the ABAP Repository. Here, the Repository Information System serves as the central entry point. For example, you can use the path *Repository Information System → ABAP Dictionary → Database Tables* to display all tables starting with T.

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.

What Is the ABAP Dictionary?

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

At this point you should 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). You can, for example, show participants an SAP table with its structure (defined in transaction SE11) and its application data) filled from the application transactions, can be viewed, for example, by means of transaction SE16 or from SE11).

Stress that developers create their dictionary objects in transaction SE11 and that these objects are created in the database when they are activated; that is, developers do not have to bother with the specifics of the database used. Lock objects in the SAP system are created in the same way. The lock mechanism of the database is not used as the database only locks at database transaction level and not at the level of the SAP transactions....

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* (transaction SE11) 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 input help) for a field on an input screen are also provided by the *ABAP Dictionary*.

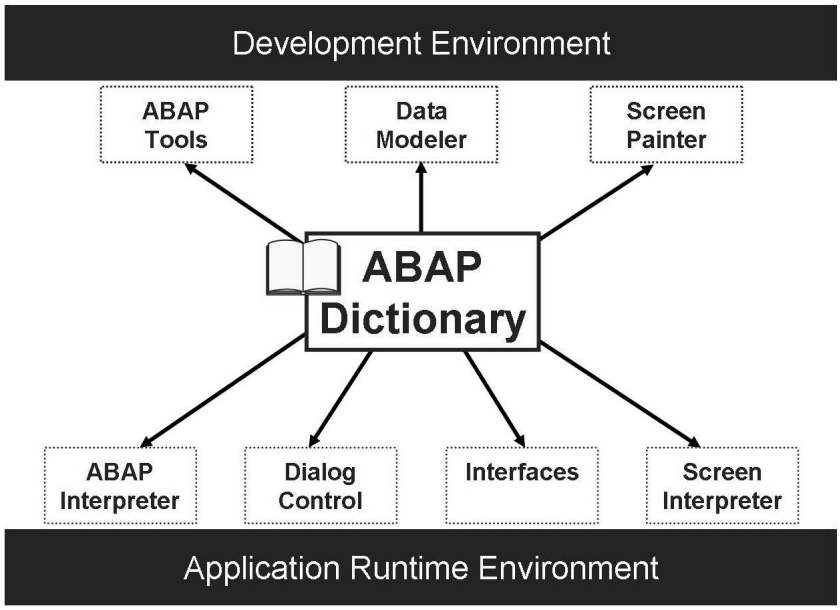

Figure 60: Significance of the ABAP Dictionary

The integration of the *ABAP Dictionary* into 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 interaction 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.

Appendix: Table Definition and the Two-Level Domain Concept

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

You can define tables database-independently in the ABAP Dictionary. When you activate the table, a physical table definition is created in 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 in transaction SE11 for that data definition.

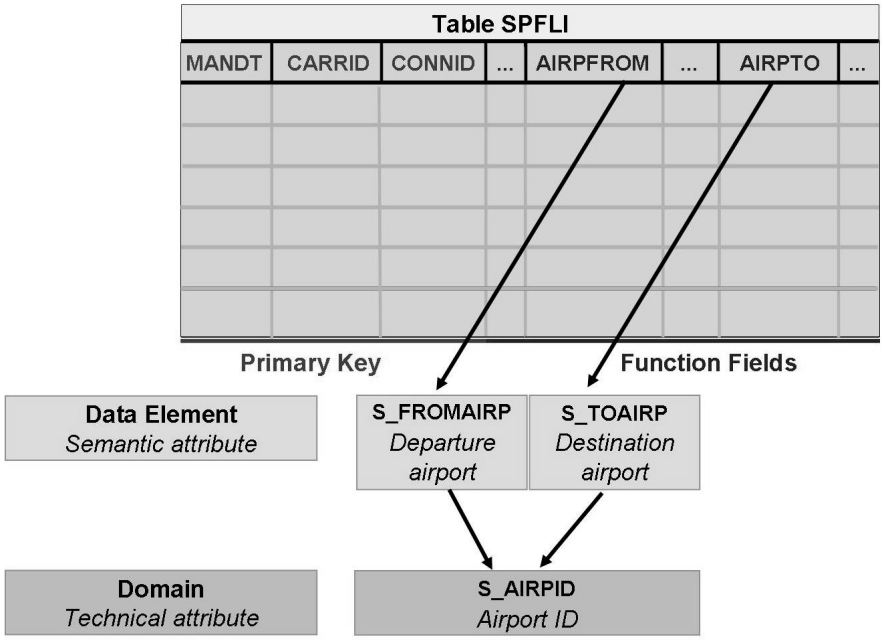

Figure 61: Appendix: Table definition and domain concept

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, use transaction SE11 to display table SPFLI. Explain the difference between the name of the field and the corresponding data element. Double-click to go to data element S_FROMAIRP and show that it is assigned to domain S_AIRPID. Double-click again to go to domain S_AIRPID and from there, display the where-used list of the domain in data elements. You see which data elements refer to this domain.

Then, display the content of table SPFLI by means of transaction SE16.

You can, if you have the time and participants are interested, give participants a step-by-step demonstration of the creation of a (transparent) table. 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.

After completing the exercises of this lesson, you can show participants how you, as the development project leader, release the change request in transaction SE09. For this, you should ensure that all groups have released their tasks beforehand. If some groups have saved the exercises as a local object, you can delete the exercises of these participants from the change request before demonstrating how the request is released.

Appendix: Modeling in the ABAP Dictionary

The following step goes beyond the actual objective of this lesson, namely to introduce the most important tools of the ABAP Workbench, and has thus been classified as an appendix that you can discuss if the participants are interested and you have sufficient time available. Maybe you would like to use the slide in this section to point out that database tables are defined in the SAP system (more precisely: in the ABAP Dictionary) and that the database creates a suitable table once they have been activated successfully, without developers having to deal with the specifics of the respective database.

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

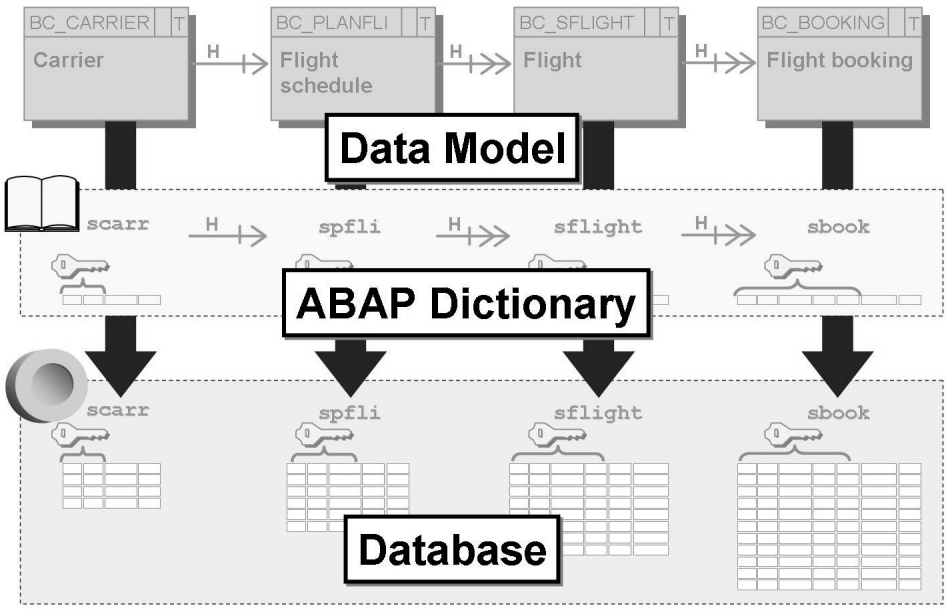

Figure 62: Appendix: Modeling in the ABAP Dictionary

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.

157

Exercise 9: Accessing and Editing ABAP Repository Objects

Exercise Duration: 30 Minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Display table structures and contents in the ABAP stack
- Write a short ABAP program
- Release a task in the *Transport Organizer*

Business Example

As an ABAP developer, you look at table contents and definitions and create an own ABAP program well as (optionally) a Business Server Page (BSP).

Caution: For this task it is necessary that you have already created a transportable change request of the *Workbench Request* type in the Transport organizer and assigned all participant groups of this course to this request as employees. Furthermore, you should have created a package in transaction SE80 before starting this exercise. Participants then assign their own developments to this package and to this change request and then release their tasks. After the completion of the exercises you can show participants how to release the actual change request.

System Data

System: The training system assigned to you
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. Participants need to be authorized to access the *ABAP Editor* and the transaction for the *ABAP Dictionary* (SE11) and to create programs there and to display table contents and definitions. Apart from that, they have to be authorized to release their own task in the *Transport Organizer* (transaction SE09).

Task 1: Working with Change Requests

Checking your own change requests and tasks

1. Display all change requests to which you are assigned. How many tasks are assigned to these requests?

Task 2: Optional: Displaying Table Definitions

Analyze the definition of table TADIR.

This exercise is very instructional, but it can be marked as an optional exercise as the respective section of the corresponding lesson has been defined as an appendix.

1. Find out which fields make up the primary key of table TADIR.
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 3: Displaying Table Contents

Display the content of the TADIR tables.

1. Find out how many programs are in your SAP system Repository.
2. Find out which of these programs begin with “RSP”.

Task 4: Writing a Short ABAP Program

From the *ABAP Editor* write an ABAP program that outputs a text line.

1. Start the *ABAP Editor* (transaction SE38). 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”. Choose *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. For this, choose the request you displayed in task 1.

Continued on next page

3. This takes you to the *ABAP Editor* where you can write your program text (if necessary, after choosing *Display* ↔ *Change*). Choose *Return* after the final period in the line

   ```
   REPORT Z<CLNT>PROGRAMM_##.
   ```

   then go to the next line and enter, 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.

Result

Congratulations on creating your first ABAP program!

Task 5: Optional: Creating a Simple Business Server Page (BSP)

Display simple output from your SAP system in your browser.

1. 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

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 there specify 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*.

2. Enter the name of the page **`Hello_##.htm`**. ## again corresponds to your group number.
3. Optional: On the *Layout* tab page, you can change the source code, for example by replacing the text `Hello World!` (within the quotation marks) with this text `Hallo`. Make sure that you text is still in quotation marks.
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.

Continued on next page

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!

Task 6: Releasing a Request

Then release your task for the change request.

1. Use the *Transport Organizer* to release your task for the change request.

Solution 9: Accessing and Editing ABAP Repository Objects

Task 1: Working with Change Requests

Checking your own change requests and tasks

1. Display all change requests to which you are assigned. How many tasks are assigned to these requests?

   a) Call the *Transport Organizer* using *Tools → ABAP Workbench → Overview → Transport Organizer* or use the transaction code SE09: On the initial screen of the *Transport Organizer* ensure that your name is entered in the *user* input field and that the *Workbench Request* field is selected.

   b) You can use the *Display* pushbutton to display the change requests to which you are assigned as the developer. If you instructor has created one change request for all course participants, you should see an entry for the change request for which there are several tasks (one for each group of participants).

Task 2: Optional: Displaying Table Definitions

Analyze the definition of table TADIR.

This exercise is very instructional, but it can be marked as an optional exercise as the respective section of the corresponding lesson has been defined as an appendix.

1. Find out which fields make up the primary key of table TADIR.

   a) 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*. 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*.

Continued on next page

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 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, double-click on 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 3: Displaying Table Contents

Display the content of the TADIR tables.

1. Find out how many programs are in your SAP system Repository.

   a) 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”.

   a) Further limit your search on the selection screen by entering the value **`RSP*`** in the *OBJ_NAME* field and choosing *Execute*.

Task 4: Writing a Short ABAP Program

From the *ABAP Editor* write an ABAP program that outputs a text line.

1. Start the *ABAP Editor* (transaction SE38). Name your program **Z<CLNT>PROGRAM_##**, where <CLNT> is the client you are logged on to, and ## is your group number.

Continued on next page

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

a) Call *ABAP Editor* using transaction SE38 (menu path *Tools* → *ABAP Workbench* → *Development* → *ABAP Editor*).

b) Enter your program name **Z<CLNT>PROGRAMM_##** and choose *Create*. Create the program with a meaningful title and the attributes specified in the task. Choose *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. For this, choose the request you displayed in task 1.

   a) For the procedure, see the exercise.

3. This takes you to the *ABAP Editor* where you can write your program text (if necessary, after choosing *Display* ↔ *Change*). Choose *Return* after the final period in the line

```
REPORT Z<CLNT>PROGRAMM_##.
```

   then go to the next line and enter, 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. *Save* saves the program in the database. *Check* executed a syntax check. *Activate* generates a runtime object so that the generated program can now also be executed by people outside the ABAP Workbench.

Result

Congratulations on creating your first ABAP program!

Task 5: Optional: Creating a Simple Business Server Page (BSP)

Display simple output from your SAP system in your browser.

1. 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

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 there specify 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*.

a) 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 and choose *Save*. ..
5. Either assign the object to a package specified by your instructor, or save it as a “Local object”.

2. Enter the name of the page **`Hello_##.htm`**. ## 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 *page* characteristic: **Hello_##.htm**. ## again corresponds to your group number.
- Choose *Page with Flow Logic* as the page type.
- Enter a meaningful description and choose *Continue*.

3. Optional: On the *Layout* tab page, you can change the source code, for example by replacing the text `Hello World!` (within the quotation marks) with this text `Hallo`. Make sure that you text is still in quotation marks.

a) See exercise text.

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.

Continued on next page

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!

Task 6: Releasing a Request

Then release your task for the change request.

1. Use the *Transport Organizer* to release your task for the change request.

   a) Access the *Transport Organizer* again (either via men path *Tools* → *ABAP Workbench* → *Overview* → *Transport Organizer* or directly with the transaction code SE09.

   b) On the initial screen of the *Transport Organizer* choose your change request which is set to "modifiable" and choose *Display*. On the next screen, select the task and then choose *Release directly*.

Hint: You might have to create the documentation for this task first. For this, double-click your task. Then go to the *Documentation* tab page and, in change mode, enter a meaningful description for your task and then choose *Back*.

Lesson Summary

You should now be able to:

- Name important tools in the *ABAP Workbench*
- Access ABAP Repository objects
- Write a simple ABAP program
- Explain the significance of the *ABAP Dictionary*
- Outline the two-level domain concept

Related Information

More information on the ABAP Workbench is provided in the leading courses for ABAP-Workbench (starting with courses **BC400** - *ABAP Workbench Basics* and **BC430** – *ABAP Dictionary*) as well as in the SAP Library for SAP NetWeaver '04 in the area *SAP NetWeaver → Application Platform (SAP Web Application Server) → ABAP Technology → ABAP Workbench (BC-DWB).*

Lesson: Introduction to the SAP NetWeaver Java Development Infrastructure

166

Lesson Duration: 50 Minutes

Lesson Overview

Opposed to the ABAP environment, in which the central development is carried out (all developers are registered on the same SAP system), a decentralized development philosophy is pursued in the Java environment.

This lesson first of all provides an overview of the steps that Development has to make to execute a Java application. Following that, typical problems that arise in a team-oriented development in the Java environment are discussed. These problems should be avoided where possible with the *SAP NetWeaver* Java Development Infrastructure (JDI), by transferring tried and tested concepts from the ABAP world into the Java environment.

Lesson Objectives

After completing this lesson, you will be able to:

- Name the individual components of the SAP NetWeaver Java Development Infrastructure
- Outline the concepts of a multi-system landscape for the Java development process

---

This lesson serves to motivate the SAP NetWeaver Java Development Infrastructure, referred to as JDI in the following. The difference in the development philosophy between the ABAP world (central development) and the Java world (decentralized development) should be made clear. The participants should familiarize themselves with the typical problems which arise when creating larger Java applications in larger development teams and learn that the JDI avoids many of these problems or offers solutions to them.

When introducing the Java development process, you should make it clear to the participants that many of the concepts here which were known from ABAP have been transferred to the Java world. The presentation of the JDI components can be done relatively briefly, however you should spend some time on tasks of the individual components within the development process.

A development of a Java object itself cannot be shown in this lesson, as you would need a JDI which was specially configured for this course as well as access to the SAP NetWeaver Developer Studio (Studio). As working with the studio and its different perspectives is vastly different to working with the ABAP Workbench, this is not included in this lesson.

Hint: Only ABAP Repository objects can be displayed using transactions from the ABAP Workbench (such as SE11 or SE38). Similarly, it is only possible to show the contents of ABAP tables using SE16. There are not (yet) any analogous “transactions” for the Java schema of the database. Creating tables and programs is done in the Studio respectively, that is, on the developer PC. When importing into other systems, the objects are deployed in their databases so that, for example, the tables are also in these systems and can be filled with data by the applications.

Business Example

Your company uses SAP systems that are based on the *SAP NetWeaver* technology platform and plans to develop its own Java-based applications. As a member of the system administration, you want to have an overview of the development infrastructure provided by SAP for this purpose.

If any of the participants should ask which applications they should develop in Java and which in ABAP, you could perhaps differentiate as follows (without starting an argument between the ABAP and Java philosophies): for developments in the portal or XI environment, the Java-based development is more suitable (as, for example, a portal requires the Java runtime environment). Classic business applications, by comparison, presumably continue to be programmed in ABAP.

Introduction

With the *SAP Web Application Server*, SAP provides a homogeneous infrastructure for both Java-based and ABAP-based applications.

Because of the different development philosophy between ABAP and Java, it is relatively difficult to compare the development processes with each other on both sides.

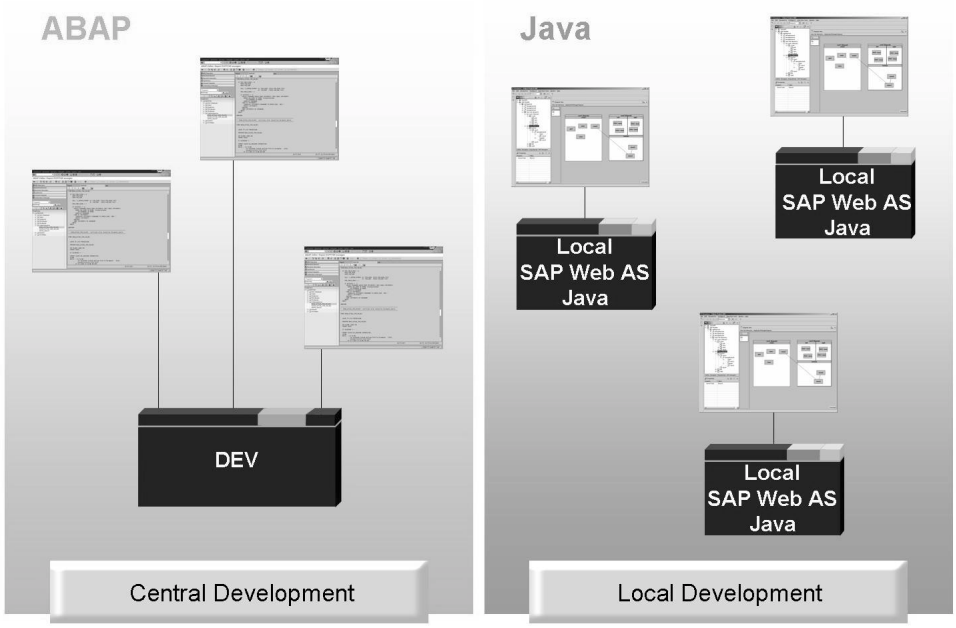

Figure 63: Different development philosophies

In the ABAP environment, the developers logon to the development system (for example, the development system for CRM applications). They work in this development system with the help of the *ABAP Workbench* as an integrated ABAP development environment. The developers only have the *SAP GUI* installed on their PCs. The runtime environment for the first test of the written applications (also in context with the programs of other developers) is the central development system.

In the Java environment by comparison, the developers are used to a decentralized development. Here, every developer not only has the development environment, but also a local runtime environment on his PC and can therefore develop and test locally independent of all other colleagues. Not until development completion (or at other specified points in time), are the applications of the individual developers combined in a central system.

After this general comparison, the following should deal with the development of Java applications in the SAP environment in more detail.

The Java Programming Language

Java is an object-oriented programming language, similar to C++ or C#.

Java was actually derived from C++ and is therefore a “simplified” C++ (for example pointers have not been included).

The entire Java code exists in methods of classes, all states are represented by attributes of classes. Java has a comprehensive library of routines, which simplify the cooperation with TCP/IP logs, such as FTP or HTTP. With the help of Method Invocation (RMI) it is also possible to call up objects on other computers.

Source codes written in Java are first of all translated by the Java compiler *javac* into an intermediary code that is independent of architecture, the so-called byte code. A class file is created from every source file (.java file). This byte code cannot yet be executed alone, but is interpreted and executed by a so-called virtual machine. This virtual machine must have been developed for every supported platform, if necessary, separately.

The Java runtime environment consists of three main components:

- Class Loader to load all classes required for executing the program
- Bytecode Verifier to check whether the loaded classes are compatible with specification of the virtual machine
- The Java virtual machine itself

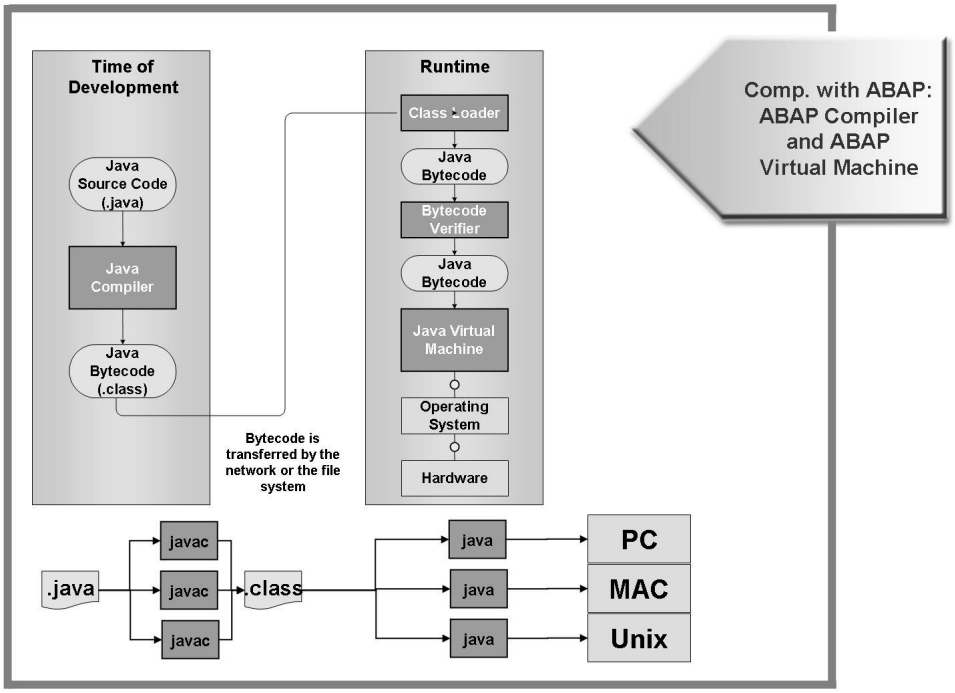

Figure 64: From the development to the execution of a Java application

If the participants are already familiar with the structure of ABAP developments, you can draw parallels to ABAP: ABAP also has a compiler and an interpreter. The intermediary code here is the ABAP load, the ABAP "Virtual Machine" (part of the work process) is an interpreter of ABAP loads.

If a Java program is to be executed, a runtime environment (Java 2 Runtime Environment, JRE) is required. The JRE consists of the Java Virtual Machine, the standard Java interfaces (classes that provide standard services, such as Remote Method Ivocation (RMI)) and other components, which are required for executing the Java applications and Applets. The Java Virtual Machine is responsible for the independence of the J2EE platform from the hardware and the operating system.

In the Java 2 Software Development Kit (J2SDK) in addition to the content of the JRE, there are some tools with which Java applications can be developed. These tools include the Java Compiler and the Java Debugger.

An integrated development environment (IDE) is generally used for developing own programs and this supports the development of J2EE applications. This builds on the SDK.

If you have a runtime environment installed on your (front end) computer, as an option, you can show the participants the associated directory structure with the files at operating system level at this point. In the directory for the JRE or the subdirectory /bin/hotspot for the Virtual Machine and in the subdirectory /bin, there are some files for the services provided.

Hint: The J2SDK includes the following standard packages

- Java Virtual Machine
- Standard interfaces (Core APIs) such as
  - java.lang with elementary classes like object and system
  - java.io for input and output
  - java.util for data structures
  - java.beans for Java components,
- Tool collections for graphic GUIs: java.awt and java.swing
- Development tools such as Java Compiler and Java Debugger
- The file format JAR for packaging entire libraries or applications as Java archives.

You can find more information on the delimitation between JRE and SDK in the Internet at http://java.sun.com/j2se/overview.html.

If a developer wants to write and (locally) test an own program, he generally requires a development environment and a runtime environment.


Hint: Background information for Java development on the SAP Web AS: The SAP Web AS has two different schemata in its database: The ABAP schema, which the ABAP work processes can access and the Java schema, which can access the Java processes. A communication between the ABAP and Java runtime environment is done with the help of the SAP Java Connectors (JCo) between the ABAP work processes and the Java server processes. In comparison to ABAP, the Java runtime environment does not (yet) know the concept of clients, which is why there are not (yet) any client-specific data in the Java schema. As opposed to ABAP, where there are different work process types for different tasks (dialog, spool, update, background, lock), the Java server process provides different services simultaneously (foe example, HTTP provider service, locking adapter service, JDBC connector service). The Java dispatcher of an instance selects a Java server process dependent on the type of user query, on which the "suitable" service is configured.

As part of the SAPTEC training course, the lesson "Principal Architecture of the SAP Web Application Server", which goes into the architecture of an instance and the processing of a user query in more detail, has already been discussed.

Excursus: The SAP NetWeaver Developer Studio

The *SAP NetWeaver Developer Studio* is SAP's own environment for developing multi-level Java-based applications. The development environment is based on the open source product *Eclipse*, whose open plug-in architecture provides a suitable platform, which can be enhanced by special functions.


Hint: A plug-in is an enhancement of the Eclipse platform by, for example, a new function. The integration of plug-ins via an extension point is predefined by Eclipse. When starting Eclipse, the runtime looks for plug-ins (plug-in.xml files in the <inst_dir>\eclipse\plugins directory). After that, the environment (main window, tool bars, menus,...) is prepared. This is why, for example, when structuring a tool bar, first of all those buttons are searched for that are there anyway and, finally, all plug-ins that are attached to the extension points of the tool bar.

The *SAP NetWeaver Developer Studio* provides a series of tools for all aspects of application development, some of which are mentioned as examples in the following list:

- Web Screen Tools can be used to develop Web interfaces
- J2EE Tools are tools for creating J2EE applications, such as those from Enterprise Java Beans
- Data access is supported by Java Dictionary Tools or other Persistence Tools

Persistence describes the possibility of saving the data you use in the long-term, i.e. beyond its use within a program. The basics for this are Open SQL for Java, logical locks and the Java Data Dictionary. The Java Data Dictionary is the central storage location for the platform-neutral definition of data types and database objects within the Java Stack.

UML is the abbreviation for Unified Modeling Language, a standardized design language in the form of graphic notation, used to present structures and processes on object-oriented program systems. UML provides a general standard for “technical drawings”, not only in informatics.

Web Dynpro Tools are tools for creating a new generation of user interfaces of browser-based business applications.

The particularity of these development tools lies in their seamless integration into the SAP infrastructure. All created development objects can thus be stored and managed in a central Repository, the Design Time Repository (DTR), in which an automatic build process is built using Component Build Service (CBS), is added to the Change Management and finally distributed via a defined software logistics process.

Motivation and Goals of the SAP NetWeaver Java Development Infrastructure (JDI)

As an introduction to this lesson, you can also use the facilitated discussion (to be found at the end of the lesson), in order to make participants aware of the potential problems of the (decentralized) development of J2EE applications.

With the Java 2 Software Development Kit (J2SDK), basically all of the tools needed for the development of smaller Java applications - in addition to the runtime environment, also programs such as the Java Compiler or a debugger - are provided. In addition to these, there are various local development environments (Integrated Development Environments, IDEs) for a comfortable development of applications, which build on the J2SDK.

When developing Java applications in larger development teams, however, new challenges arise. The following illustration shows the typical development process in the Java environment and shows some of the difficulties that have to be dealt with. These difficulties often have to do with the fact that the development takes place locally in the runtime environment of the individual developer and this development environment does not correspond in every detail to the central test environment.

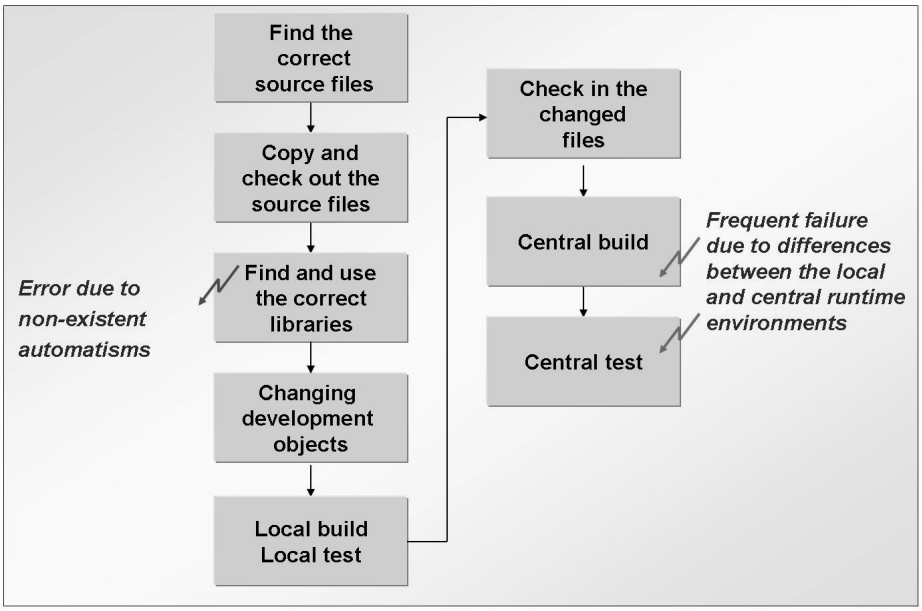

Figure 65: Typical problems in a team-oriented development in the Java environment

Java applications can exist in various versions (just like ABAP applications). Before a new application can be created or an existing application can be revised, the developer must ensure that he uses the correct version of the required source files or the used libraries. Furthermore, the use of different versions of the same objects within different development tasks typically requires a reconfiguration of the local development environment (time-consuming and susceptible to errors).

After the correct version of all necessary sources have been found and have been checked out of a central directory, the local development can begin. In the meantime, the developers from time to time carry out - based on their own, local runtime environment - local builds, in order to be able to test their applications in the local runtime environment. After the completion of this development, the changed and referenced sources are once again checked in to a central storage location. Errors can occur here, if the local environment of the developer differs from the environment of the central test system. For example, in the meantime, referenced objects could have been changed by other developers and then rechecked in, in the changed form. The test in the central test system can also

cause unforeseen errors due to the different runtime environments (of developers). As a result, the own development must be adjusted to suit the changed framework conditions.

It is important to make clear that, because of the local development and the local runtime environment in general, every developer has an own specification of the runtime environment that is different from other developers (for example, because he has deployed components in his runtime environment, which the other developers have not used or have used in other versions) and, in addition, this environment differs from the environment on the central test system.

If the participants know the development process in the ABAP environment, you can point out the differences to ABAP, where these problems generally do not arise because of the central development (all developers work in the same runtime environment).

Goals of the SAP NetWeaver Java Development Infrastructure (JDI)

In the ABAP environment, the problems described above do not arise due to the central development environment (all developers work in a central runtime environment and each use the current version of the respective development object). The goal of the SAP NetWeaver Java Development Infrastructure – in the following, referred to as the Java Development Infrastructure, JDI - is therefore to avoid the problems described above by transferring tried and tested concepts from the ABAP world into the Java world. Here, SAP builds on the known standards (J2EE or WebDAV and DeltaV as a Repository standard for the access and the versioning of any development objects). The local development environment is based on the open source development tool *Eclipse* .

If you want, you can make clear to the participants that SAP - in order to adapt J2EE to large business applications - had to enhance the programming model (key word here: Web Dynpro), had to provide new features (for example, Open SQL for Java or SQLJ), integrate open standards, which are not defined in the J2EE standard (such as Web Services) and had to provide a highly productive development environment – the SAP NetWeaver Developer Studio.

Hint: SQLJ (SQL Java) consists of a number of commands, which Java extends by static SQL commands. JDBC, by comparison, offers a dynamic interface between Java applications and the database. SQLJ and JDBC thus complement each other.

Structure of the JDI

The *SAP NetWeaver Developer Studio* provides access to the SAP NetWeaver Java Development Infrastructure (JDI). The JDI consists of local development environments (IDE) on the PCs of the developers and server-based services, which provide the development teams centrally with a consistent development environment and supports the software development during the entire lifecycle of a product.

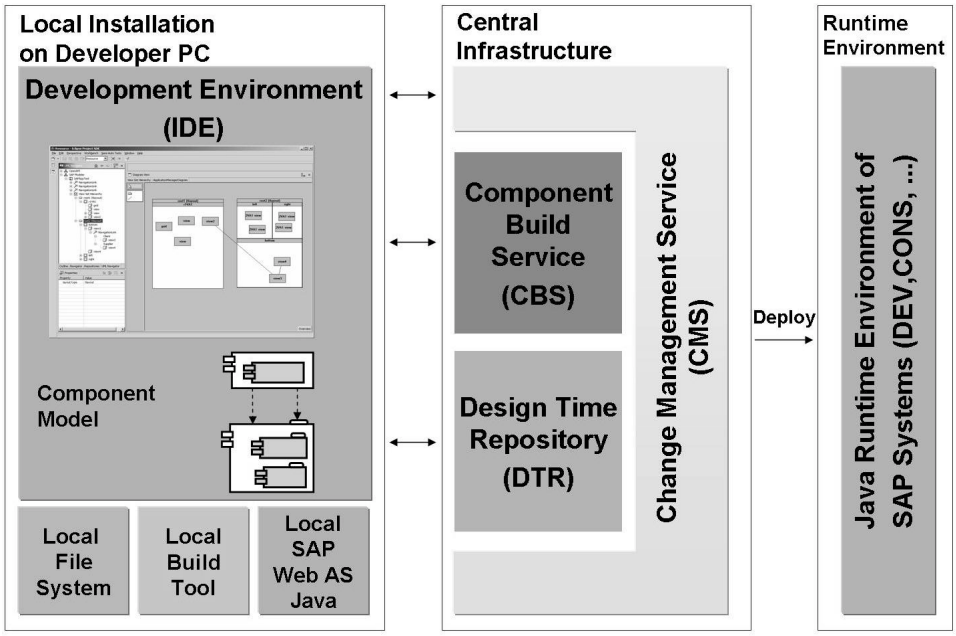

Figure 66: Overview of the SAP NetWeaver Java Development Infrastructure

Point out that the local development environment is installed and configured on every developer PC, but that the central infrastructure, if necessary, only exists one time per company, i.e. this central infrastructure is **not** generally provided separately for every SAP system. By comparison, the runtime environment (DEV, CONS, TEST, PROD) exists many times - like in the ABAP environment.

The development of Java applications in the *SAP NetWeaver Developer Studio* is based on a software component model. Using this, software projects can be structured systematically in clear and reusable units from the beginning. Components can use other components in a well-defined and controlled way, can encapsulate subordinate components and can publish their functions through a set of public interfaces, the public parts.

On the side of the central infrastructure, there are - among others, the *Design Time Repository*, the *Component Build Service*, and the *Name Server* as part of the System Landscape Directory.

Software development worldwide requires a conflict-free method for creating names for software objects. Without such a method, different software teams may use the **same** names by chance for software objects with different uses. If the software objects are then combined in the same runtime environment, this could even deactivate applications because of a simple name conflict. In order to avoid name conflicts, the SAP System Landscape Directory (SLD) provides a name reserving service (Name Server), which makes it possible to reserve globally unique names.

The SAP System Landscape Directory (SLD) simplifies the administration of the system landscape. The SLD server contains component information, system landscape descriptions and a name reservation, which is based on the Standard Common Information Model (CIM). For more information on the namespace concept, see the SAP Service Marketplace with the Quick Link */namespaces*.

The **Design Time Repository** (DTR) enables a versioning of the source code management and therefore the shared development of software in teams as well as the transport and replication of sources.

The Design Time Repository consists of two parts, the DTR client and the DTR server. The DTR client is part of the SAP NetWeaver Developer Studios. The DTR server manages the data versioning. All files are stored in a relational database. The data exchange between DTR client and server is done via the http protocol in accordance with WebDAV und DeltaV. DeltaV is described in more detail in IETF-RFC 3253.

The **Component Build Service** (CBS) is used for the central Build of the source texts. The operation for the developers is integrated into the *SAP NetWeaver Developer Studio*. For the Build process, the *CBS* communicates automatically with the *Design Time Repository*. For further processing, the *CBS* communicates with the system logic of the Change Management Service. It provides the following core functions

- Build on demand
- Central storage of Build results and Build tools
- Activation concept

The central Build of changes is done rapidly and upon demand by the developer. In comparison to the “Nightly Build”, this results in a considerably reduced time for removing errors. The CBS provides both the Build results and the necessary environment for the Build (for compiling necessary libraries, generators, Build scripts) in a central archive pool. This archive can be downloaded from the SAP NetWeaver Developer Studio into the local environment of the developer. In

addition, a difference is made between an inactive and an active status of the development objects. To ensure that changes are transferred from inactive to active status, these must be activated. Prerequisite for this is the successful central Build of the changed development components. The CBS itself is based on the standard Java Build tool *ant* and creates deployable units.

The **Change Management Service** (CMS) is used for the central administration of the Java development landscape and the transport across the entire software lifecycle. With this, you can configure the development landscapes and transport software changes. The functions of the *CMS* are closely interlinked with the *DTR*, the *CBS* and the *System Landscape Directory (SLD)*. As an administrator tool, the *CMS* includes these components and links the individual parts to one unit, with which divided development projects can also be realized.

The following tools, which are also part of JDI, are not listed in the illustration.

- Software Deployment Manager (SDM) for the manual and automatic deployment of archives in central runtime systems
- System Landscape Directory (SLD), that holds information on the system landscape (only the Name Server is mentioned in the illustration as part of the SLD)

Overview of the Development Process

The following illustration provides an overview of the process of a Java development using the SAP NetWeaver Java Development Infrastructure (JDI).

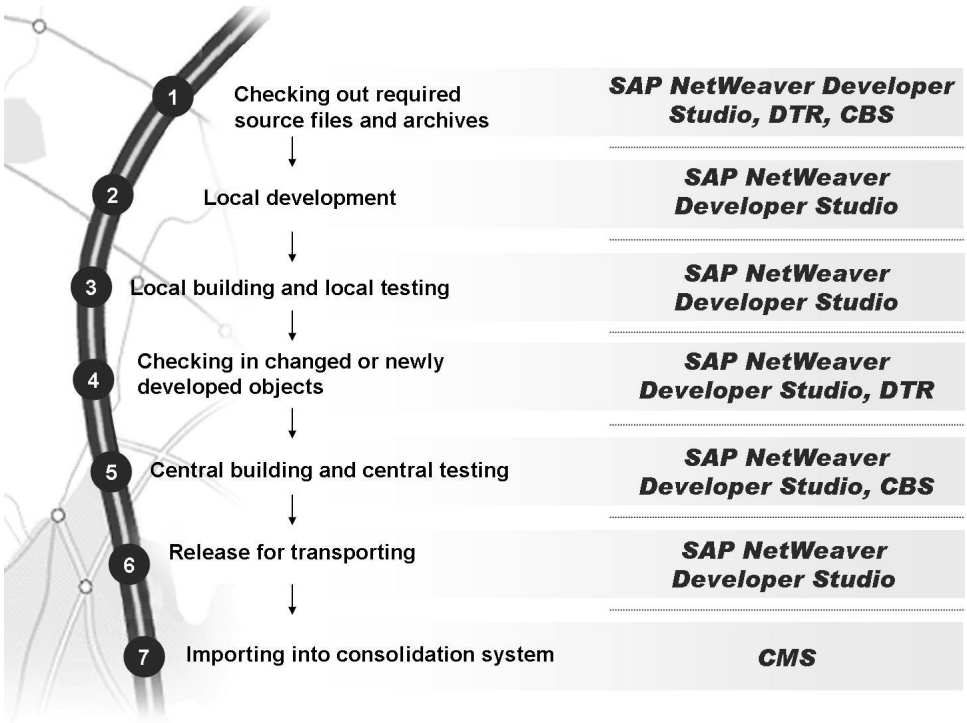

Figure 67: Java Development Process using the JDI

The developer always works in the *SAP NetWeaver Developer Studio* (in the following referred to as “Studio”), which provides him with the central services of the JDI (such as the view of the *Design Time Repository, DTR*)

At the beginning of the development, the required source files from the *DTR* and the required archives (for example, from other used software components) from the *Component Build Service (CBS)* are stored on the developer's PC. The local development can then begin. The source files are created or revised here. The sources can be built locally from time to time (to do this, the local build process is called up from the Studio). A local testing on the runtime environment of the *SAP Web AS Java*, which is located on the developer's PC, can also be carried out.

After completing the local development, the changed objects are checked in to the *DTR* (under a new version number). After that, the developer himself can trigger a central build on the central development system (DEV) from out of the Studio. After successful central building of the entire application, it can also be tested centrally (i.e. in the runtime environment of the central development system).

If all tests of the developer were successful, he can release his changes for import into follow-up systems. This corresponds to the release of the task and of the change request in the ABAP environment. The developer's work is now completed. Subsequently, the source files waiting on import can be imported centrally into the consolidation system by means of the *Change Management Service (CMS)*.

The Four-System Landscape

As the running developments of the software developers should not disturb the productive system operation in the local development systems and in the central development system, the productive environment and the development environment must be strictly separated from one another. Furthermore, for testing the developments, the most stable (i.e., not constantly changed by the import of new developments) runtime environment possible should be provided. SAP recommends a four-system landscape in connection with the development of Java applications. The different systems represent different development statuses of the software components to be developed.

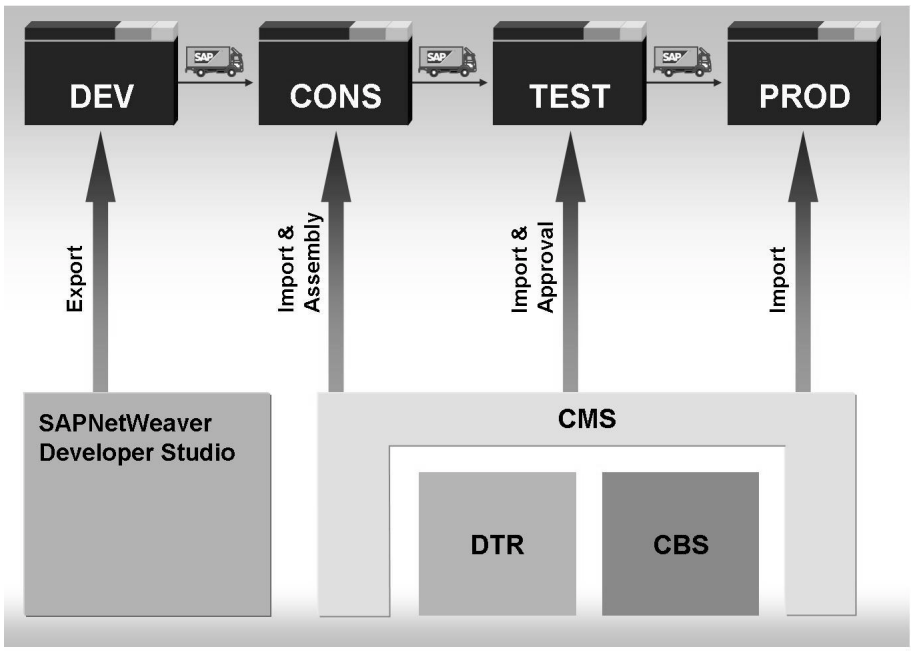

Figure 68: The Four System Landscape for Java Developments

At this point, go a little bit more into detail about the necessity of a multi-system landscape. If your participants are familiar with the ABAP area, you can also draw parallels to the ABAP environment in which a three-system landscape is recommended. Emphasize that the transports of CMS are triggered and controlled as part of the JDI.

The more specific reason for a four-system landscape (instead of a three-system landscape) lies in the different transport process: While, in the ABAP environment, transporting is done from the development system to the productive system on the level of transport requests, in the Java environment - beginning with the assembly

step - entire software components are transported respectively. The change of one individual program in the DEV system therefore results in the import of the entire associated software components into the TEST or PROD system.

The central development system (DEV) is used by the individual developers to test the local developments in a larger context, i.e. in the interaction of changes made by other developers. The consolidation system (CONS) and the test system (TEST) are used to consolidate a certain fixed status of a software component or for the final test (also in the sense of an integration test). Not until a test has been successful, is the development finally imported into the productive system (PROD).

The transport between the individual systems is triggered and controlled by *CMS*.

As a conclusion or summary to this lesson, you can once again compare the main differences between ABAP and Java runtime environments. The following table, which you can add to, may be helpful here:

	ABAP	**Java**
Development scenario	central	decentralized
Development environment	ABAP Workbench	SAP NetWeaver Developer Studio
Storage location of the source code	ABAP Repository of the database of the respective SAP systems	Design Time Repository (DTR) of the JDI
Transport into follow-up systems	Transport Management System (TMS)	Change Management Service (CMS) of the JDI

Facilitated Discussion

The participants should be made aware of the typical problems that can arise during the creation of larger Java applications (especially with larger groups of developers) and recognize that the JDI does not even let most of these problems arise in the first place or it provides solutions for those that do.

Discussion Questions

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

What problems can arise in larger development projects in the J2EE environment, for example, because of the decentralized development?

Hint: Typical problems are

- Manual configuration of the local development environment
- Complex, manual consolidation of the local development in the build process (regular failure because of changed classes, which are referenced in the own development), a central nightly build is often carried out, that is, there is no build-on-demand. This leads to longer correction times.
- Manual distribution of the software in the follow-up system and manual deployment in test systems.
- No support for the delivery and the upgrade of the versioned Java source code.

Lesson Summary

You should now be able to:

- Name the individual components of the SAP NetWeaver Java Development Infrastructure
- Outline the concepts of a multi-system landscape for the Java development process

Related Information

You will find an overview of the SAP NetWeaver Development Infrastructure in the online documentation on *SAP NetWeaver '04* in the area *SAP NetWeaver → Application Platform (SAP Web Application Server) → Java technology in the SAP Web Application Server → Architecture Handbook → SAP J2EE Engine System Architecture → SAP NetWeaver Java Development Infrastructure*.

The courses ADM200 – *SAP Web AS Java Administration* and ADM325 – *Software Logistics* each contain an extensive section on the change management and the software logistics in the Java environment.

Unit Summary

You should now be able to:

- Describe the data structure of the ABAP Stack in an SAP system
- 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 ABAP Repository objects
- Write a simple ABAP program
- Explain the significance of the *ABAP Dictionary*
- Outline the two-level domain concept
- Name the individual components of the SAP NetWeaver Java Development Infrastructure
- Outline the concepts of a multi-system landscape for the Java development process

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 the ABAP Stack in 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. At the end of an (ABAP) development project, 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

5. In the ABAP environment, 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

6. What is the purpose of a domain in the context of the ABAP Workbench?

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

7. Which of the following services are part of the SAP NetWeaver Java Development Infrastructure (JDI)?

Choose the correct answer(s).

- □ A Design Time Repository (DTR)
- □ B Object Development Tool (ODT)
- □ C Component Build Service (CBS)
- □ D Change Management Service (CMS)
- □ E Central Make Tool (CMT)

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 the ABAP Stack in an SAP system consists of

   Answer: B

   the ABAP Stack of 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. At the end of an (ABAP) development project, 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.

5. In the ABAP environment, you can carry out the entire software development cycle in the SAP system using the ABAP Workbench tools.

   Answer: True

   Transactions, tools, and interfaces are available in the SAP system for all phases of the (ABAP) software development process.

6. What is the purpose of a domain in the context of the ABAP Workbench?

   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.

7. Which of the following services are part of the SAP NetWeaver Java Development Infrastructure (JDI)?

Answer: A, C, D

The SAP NetWeaver Java Development Infrastructure (JDI) includes the *Design Time Repository* for versioned source code management, the *Component Build Service* for the central Build based on the component model and the *Change Management Service* for the transport of Java objects to follow-up systems. The central Make Tool and *Object Development Tool* are imaginary names, the development of J2EE applications is done with the help of the *Eclipse*-based *SAP NetWeaver Developer Studio*.

Unit 5

181

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 set priorities. 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
- List 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
- Make a Remote Function Call
- Explain Web Services
- Describe UDDI and WSDL
- Describe the *SAP Business Workflow* concept
- Explain the flow of a workflow process
- Submit a leave request within the *SAP Business Workflow*
- Describe additional application areas for the *SAP Business Workflow* concept

Unit Contents

Lesson: Cross-System Business Processes

182 Lesson Duration: 30 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.

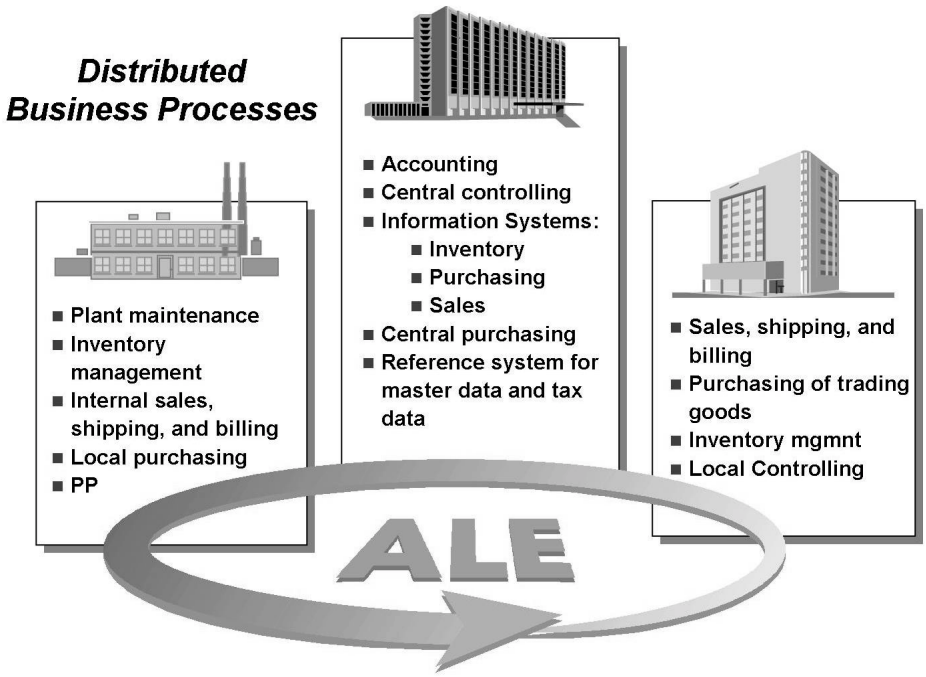

Figure 69: Business process distribution using ALE

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 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 quick link */ibf* (and then under *Integration Scenarios*).

Lesson: Remote Function Calls and BAPIs

Lesson Duration: 30 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:

- List 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
- Make a Remote Function Call

This lesson gives the participants an overview of the various interface technologies available. You need a thorough understanding of RFCs (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).

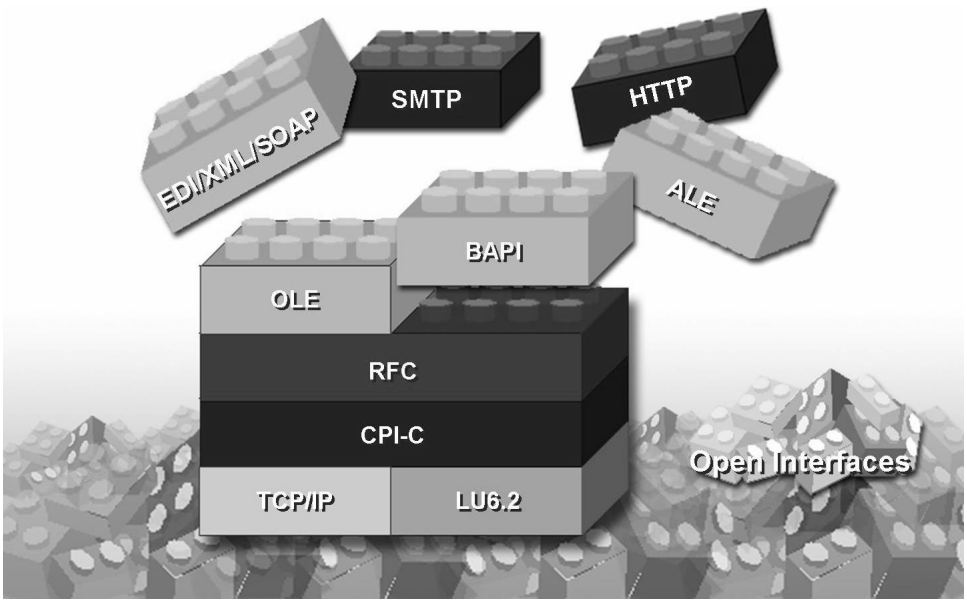

Figure 70: Interfaces technologies used in SAP systems

SAP systems use the following interface technologies that are listed in the above graphic:

- ALE: Application Link Enabling
- BAPI: Business Application Programming Interface
- CPI-C: Common Program Interface Communication
- EDI: Electronic Data Interchange
- HTTP: HyperText Transfer Protocol
- LU 6.2: Logical Unit Type 6.2
- RFC: Remote Function Call
- OLE: Object Linking and Embedding
- SMTP: Simple Mail Transfer Protocol
- SOAP: Simple Object Access Protocol
- TCP/IP: Transmission Control Protocol / Internet Protocol
- XML: Extensible Markup Language

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.

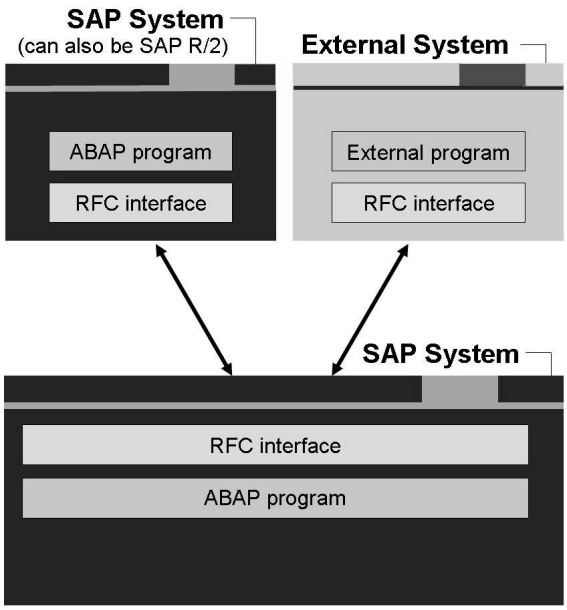

Figure 71: RFC connection possibilities

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.

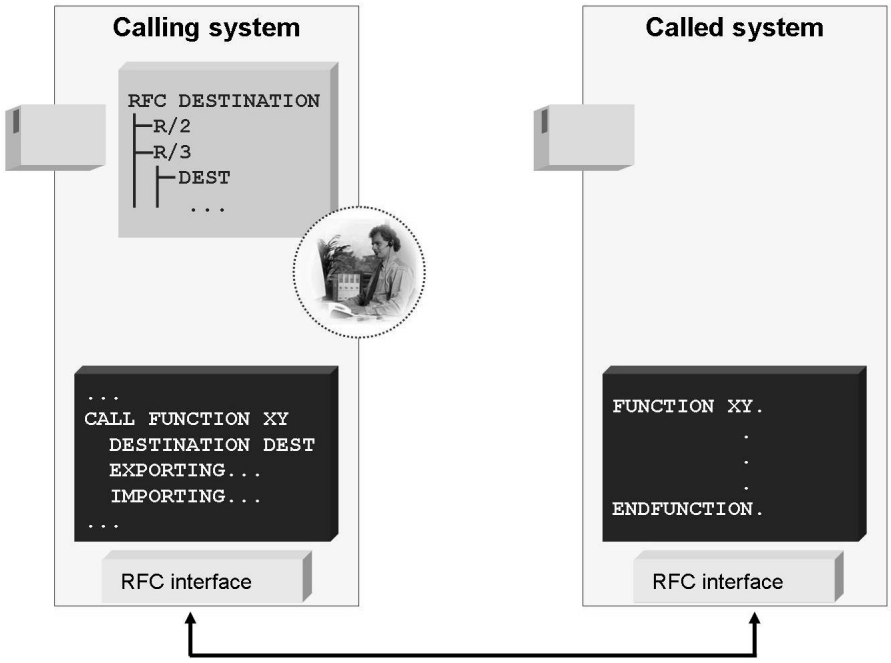

Figure 72: Remote Function Call in detail

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 <Ziel>
  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: You can also create RFC connections for which the user of the user making the call is used in the target system. That means different users can use the same connection 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 display an overview of available BAPIs in the BOR, for example, using the *Business Object Repository* pushbutton in the Business Object Builder (*Tools → ABAP Workbench → Development → Business Object Builder*), transaction SWO1. You can access the BOR directly using transaction code BAPI.

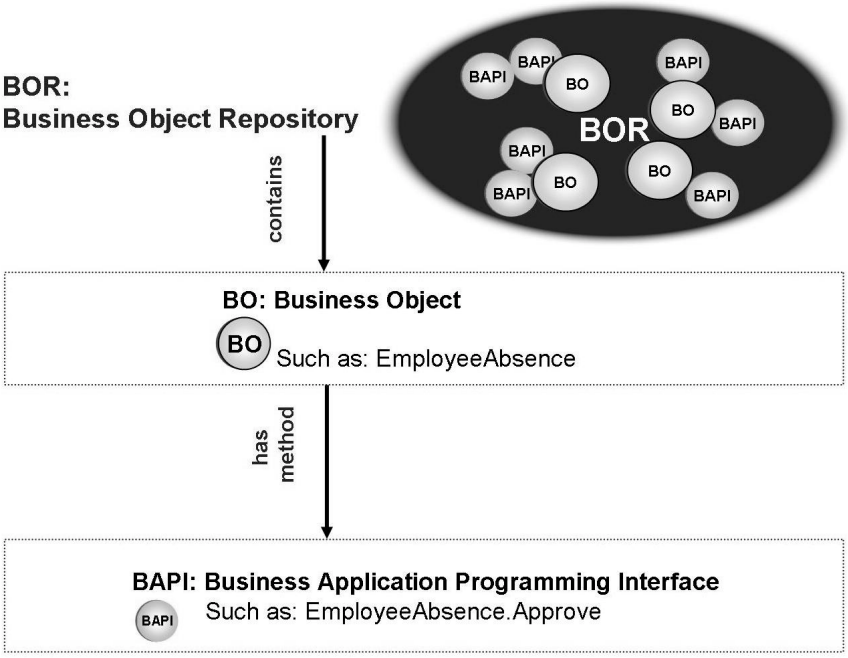

Figure 73: BOR and BAPIs

BAPIs, which represent methods for business objects in an SAP system, are used in a variety of contexts. Here are some possible uses for BAPIs:

- To link business processes across system boundaries (for example, when using 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.

193

Exercise 10: 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 10: 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) Choose *Tools → ABAP Workbench → Development → Business Object Builder* and choose *Business Object Repository* (Transaction BAPI). 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) Press *Test/Execute* (F8).

   Specify **`NONE`** as the RFC target system entry; this refers to your own system. Enter your user name on the query screen. Select *Execute*. As a result, your user data, requested by the RFC from the BAPI, is displayed.

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:

- List 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
- Make a Remote Function Call

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: Web Services

196 Lesson Duration: 30 Minutes

Lesson Overview

This lesson provides you with an introduction to the subject of Web Services.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain Web Services
- Describe UDDI and WSDL

Web Services are modular function programs that can be combined flexibly. Web Services, together with WSDL, can be described as software interfaces, and not so much as web-capable applications. However, Web Services can also be used from the Web Browser, but this on its own is not their main characteristic. In the instructor's demo the Web Service system information, for example, is called up in an ABAP program. SAP prefers the development of Enterprise Services that can be the total or combination of individual Web Services. Flexible, modular programming will complement monolithic software architectures to a great extent in the future, at least that is the opinion of consultancies based on current surveys.

Business Example

Your company wants to technically realize online services using Web Services.

Web Services - A Short Introduction

The *SAP Web Application Server* is also a development platform for Web Services. A Web Service is a service, which is made available via Internet protocols and, as a rule, it can be operated using an Internet browser. Web Services are the technical basis for making individual functions of an applications directly available. Here, the existing function of an application can be addressed via standardized access protocols and contents can also be exchanged in a structured form. In this way, cross-application functions, which are sold by SAP as complete units for the business process under the name of Packaged Composite Application (*SAP xApps*) can also be developed flexibly. The combination of several granular services, in

the sense of self-contained business scenarios, is called an Enterprise Service. In the *SAP Web Application Server* the following, basic standards for Web Services are implemented:

- eXtensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)

  SOAP describes a protocol, which you can use to call up Web Services in distributed system landscapes. SOAP uses HTTP as a transport protocol. A SOAP message has a header with the additional information and a body with the actual message.

- Web Service Description Language (WSDL)

  WSDL is a meta language, which is used to describe the function of a Web Service. Functions, parameters and return codes in particular are described in a machine-readable form. WSDL is standardized by the World Wide Web Consortium (W3C), see the following URL: http://www.w3c.org/TR/wsdl.html

- Universal Description, Discovery, and Integration (UDDI) is a directory service for dynamic Web Services. A directory of Web Services is provided via a SOAP interface. The information here is highlighted in white, yellow and green pages. You can find more information on UDDI at the following address: http://www.uddi.org. SAP itself also operates a UDDI server at http://uddi.sap.com, on which Web Services can be registered and searched for.

Web Services and the SAP Web Application Server

Web Services can already be developed in the SAP system from release *SAP Web Application Server 6.20*, however, the development tools have been considerably enhanced for release 6.40. Thus, an existing, remote-capable function module from the *Function Builder* can be transformed (transaction SE37) into a Web Service. A Web Service is a module that can be used flexibly in different applications. The creator publishes the Service in a publicly accessible UDDI directory. The customer can then search directly for Web Services in the UDDI directory.

The demo scenario does not go into the publishing of the Web Service in a UDDI directory for reasons of clarity.

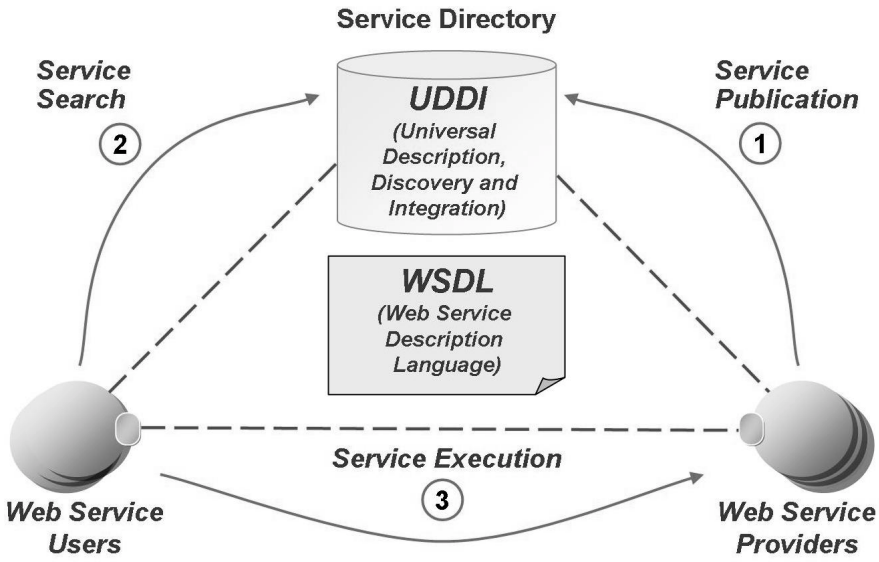

Figure 74: Outline of a Web Service scenario

Technically, the once defined Web Service can be called up in different ways, for example, from an ABAP program or from a Business Server Page.

The instructor can present the Web Service *Z_WS_SYSINFO* as a demonstration, which shows you the following system data: *SID*, *Host name of the DB server* and *operating system of the DB server*. This Web Service can be called up from a) the ABAP program *Z_SYSINFO* and b) from the Business Server Page *Z_BSP_SYSINFO*. The BSP is not public, please logon with the group user SAPTEC-##.

The following outlines how a Web Server is created from an RFC-capable ABAP function module.

The exact instruction for setting up the Web Service is not necessary in SAPTEC, as the Web Service has already been set up in the training system. However, the interested trainer should look at the courses SAPNW and BC416 or the online documentation for the necessary setup procedures.

All important details shall be mentioned here, so that the settings can also be checked in case of errors. For the function module *RFC_SYSTEM_INFO*, in transaction SE37 (display mode: *Utilities → Other Utilities→ Generate Web Service → From Function Module*), the Web Service *Z_WS_SYSINFO* has been created. The virtual interface is called *Z_WS_INTF_SYSINFO*. In transaction SICF, you can display the ICF node *<localhost>/sap/bc/srt/rfc/sap/Z_WS_systeminfo* generated for the Web Service, and if necessary, you can activate it manually. In transaction WSADMIN the WSDL file is generated (highlight Web Service and

choose the path *Web Service → WSDL* (whereby SAP Features can be included in the WSDL file); in some cases a renewed logon to the SAP system is necessary here). In the SE80, the Client Proxy Object *ZREF_CO_Z_WS_SYSINFO* (if necessary, under the local objects of the internal user; using the right-hand mouse button, click the path *Create → Enterprise Service/WebService → Proxy Object* as source *URL/HTTP Definition* and enter the URL of the WSDL page), the ABAP program *Z_SYSINFO* and the Business Server Page *Z_BSP_SYSINFO* are entered. In transaction LPCONFIG, the logical port *ZSYSINFO* for proxy class *ZREF_CO_Z_WS_SYSINFO* is defined and activated as a default port.

- The Service Provider, for example, generates the Web Service from a function module. Furthermore, a URL and the WSDL file are generated.
- The Service Requester creates a proxy object, which refers to the URL of the Web Service. Next, an ABAP class that matches the proxy object is generated and a logical port is assigned.
- The proxy object is written in and integrated into an executable program, for example, in ABAP, and called up there.

Facilitated Discussion

Show the significance of Web Services for the future IT landscape and for future business scenarios.

Discussion Questions

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

Do you intend to use Web Services soon? What implementation options do you see for Web Services in your company?

Lesson Summary

You should now be able to:

- Explain Web Services
- Describe UDDI and WSDL

Lesson: SAP Business Workflow

200 Lesson Duration: 30 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
- Submit a leave request within the *SAP Business Workflow*
- 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.

SAP Business Workflow Basics

Conveniently, transaction SWUI_DEMO can be used to start different demo workflows without preparations. Simply start the demo for the leave request. The participants will carry out an exercise for this.

Caution: When the course was created, transactionSWUI_DEMO still showed an error in the 6.40 system, the *Start* pushbutton could note be used and produced an error message. But the demo workflow can be started without problems using the alternative button *Start with Details*. On the next screen, simply press *Start Workflow* or *F8*.

The first call of the demo workflow takes a very long time because the program buffer has not been filled yet and the program load might still have to be compiled. Often, the executing work process even switches to PRIV mode. The instructor should call up the workflow in the preceding break in order to speed up the demonstration.

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.

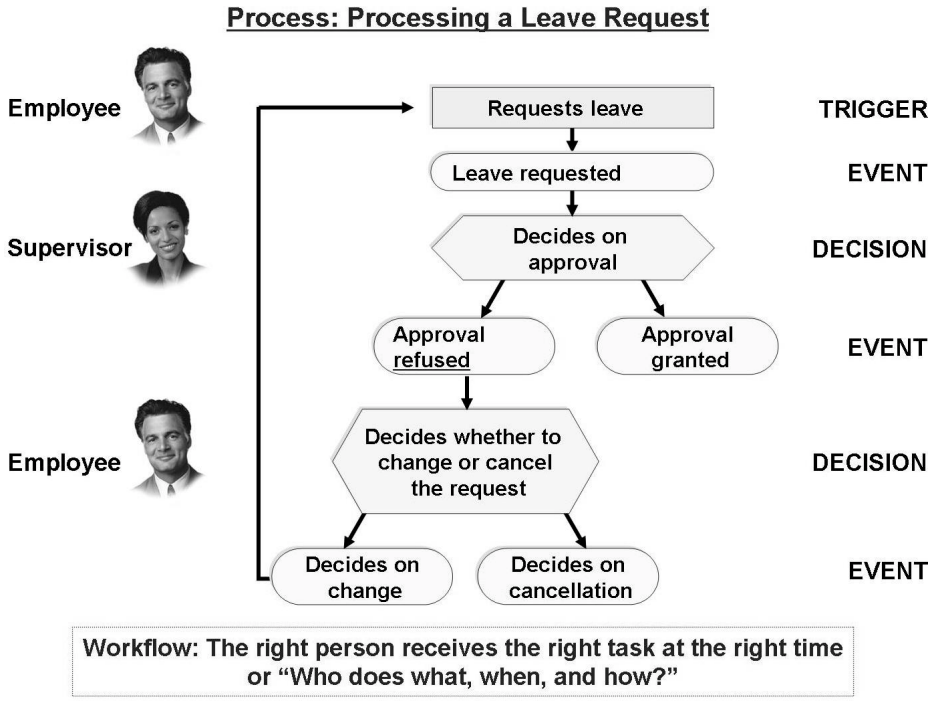

Figure 75: Example of an SAP Business Workflow process

Two people are involved: the person who creates the request (requester) and his or her assigned supervisor. 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 → Workplace → Inbox* or 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.

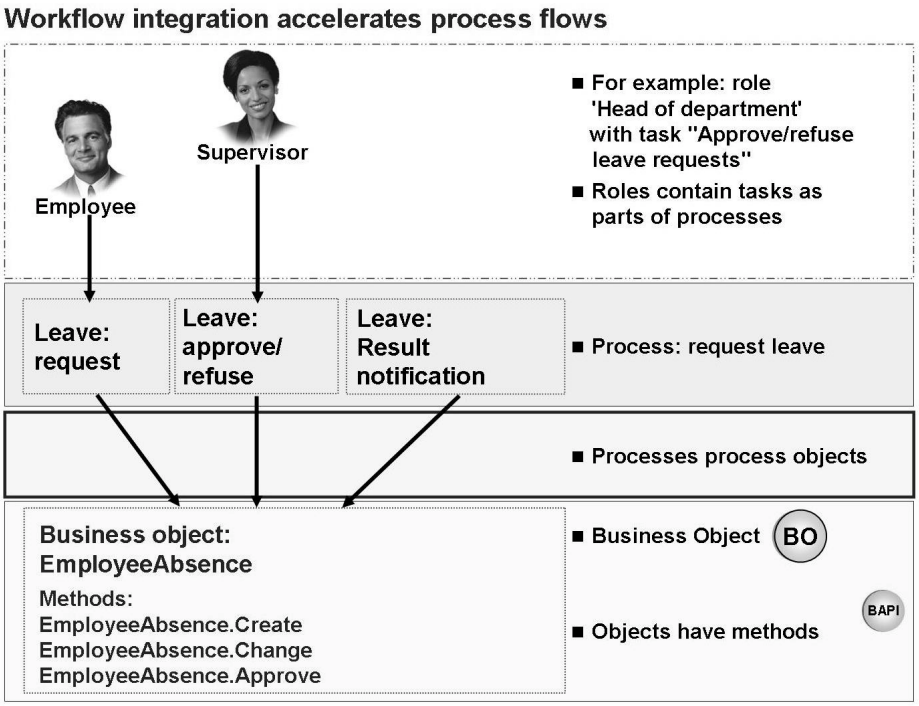

Figure 76: The workflow environment

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.

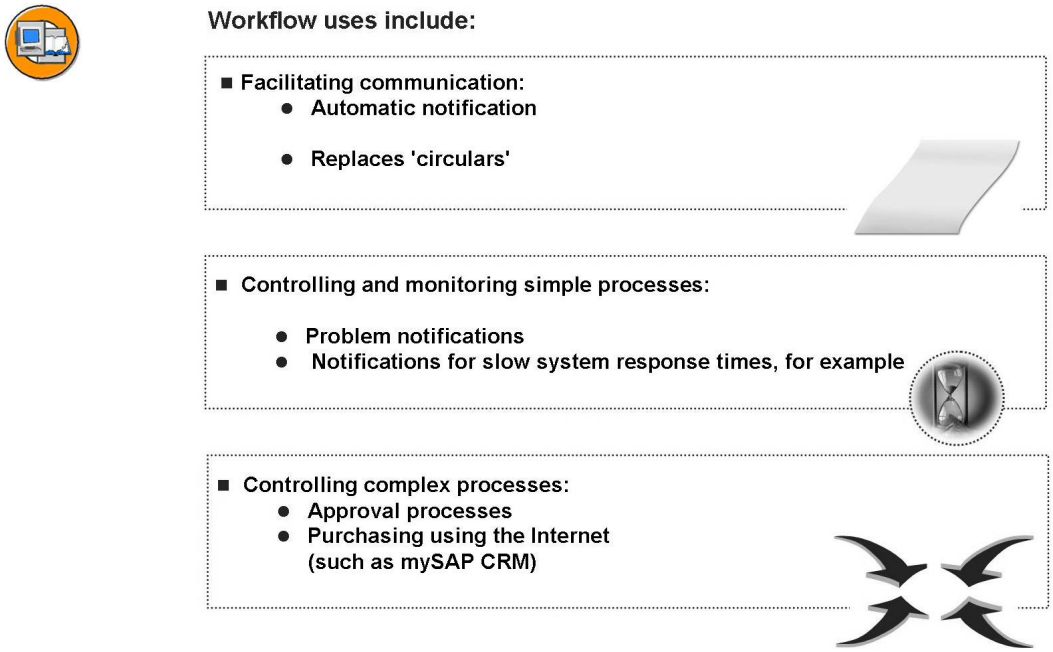

Figure 77: Workflow application areas

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?

Exercise 11: Leave Request as Workflow

Exercise Duration: 10 Minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Describe the *SAP Business Workflow* concept

Business Example

In your production SAP system, the procedures are to be implemented using the *SAP Business Workflow*, hence, you are interested in the concept and possible uses of the workflow.

System Data

System: The training system assigned to you.
Client: The training client assigned to you.
User ID: This exercise is carried out with course users SAPTEC-##, whereby ## refers to the respective group number.
Password: The initial password issued by you or the password that has already been changed by the users.
Set up instructions:

1. You use standard transaction SWUI_DEMO. Hence, no special preparations are required.

Task: Management of Leave Requests using SAP Business Workflow

Use *SAP Business Workflow* to generate and process a leave request.

When the course was created, transaction SWUI_DEMO still had an error in the 6.40 system. The *Start* button cannot be used, instead an error message is output: “Error: tipsStart is undefined” If the error still appears in the training system use the Start with Details button. On the following screen you simple press function key *F8*. Then you continue the exercise as described.

Before participants do the exercise, the instructor should have already called up the demo workflow during a break. This fills the program buffers and compiles loads that might be missing so that the calls are quicker for the participants.

1. Your user SAPTEC-## generates a leave request for today.
2. Check whether the leave request you just entered has arrived in the inbox of the *SAP Business Workplace*.
3. The supervisor approves the leave request.

Solution 11: Leave Request as Workflow

Task: Management of Leave Requests using SAP Business Workflow

Use *SAP Business Workflow* to generate and process a leave request.

When the course was created, transaction SWUI_DEMO still had an error in the 6.40 system. The *Start* button cannot be used, instead an error message is output: "Error: tipsStart is undefined" If the error still appears in the training system use the Start with Details button. On the following screen you simple press function key *F8*. Then you continue the exercise as described.

Before participants do the exercise, the instructor should have already called up the demo workflow during a break. This fills the program buffers and compiles loads that might be missing so that the calls are quicker for the participants.

1. Your user SAPTEC-## generates a leave request for today.

   a) Start transaction SWUI_DEMO. In the left navigation area, choose the *Demo for leave request process* entry. Choose *Start*. You see an input template. Please enter *Training* as the department and specify an absence of eight hours; the current date is already entered in the date field but you can still change the entry. Furthermore, enter *Special reasons* as the reason for the absence. Choose pushbutton *Save*.

2. Check whether the leave request you just entered has arrived in the inbox of the *SAP Business Workplace*.

   a) Start transaction SBWP. Open the *Inbox* folder. When you double-click the *Workflow* folder, the workflow list of open workflow items is displayed. Your leave request should be displayed there.

   

   Hint: In the context of the demo workflow you are both employee and supervisor in one person. In a real workflow scenario, only your supervisor would see the generated leave request.

Continued on next page

3. The supervisor approves the leave request.

   a) Start transaction SBWP. Open the *Inbox* folder. By double-clicking the *Workflow* folder a list of open workflow items is displayed. Choose your leave request and process it. You start processing using the *Execute* button. Choose *Approve*. Use the green arrow or the *F3* button to return to the previous screen. Choose *Close Work Item* to close the workflow.

Hint: In a real workflow the employee would be notified by express mail that his leave request has been approved. This express mail is not sent in the demo workflow.

Lesson Summary

You should now be able to:

- Describe the *SAP Business Workflow* concept
- Explain the flow of a workflow process
- Submit a leave request within the *SAP Business Workflow*
- 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 on the SAP Business Workflow: **BIT600**, **BIT601**, **BIT603**, and **BIT610**.

Unit Summary

You should now be able to:

- Name various cross-system business processes
- Explain the ideas behind the ALE concept
- List 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
- Make a Remote Function Call
- Explain Web Services
- Describe UDDI and WSDL
- Describe the *SAP Business Workflow* concept
- Explain the flow of a workflow process
- Submit a leave request within the *SAP Business Workflow*
- Describe additional application areas for the *SAP Business Workflow* concept

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 or 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 Microsoft 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

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 or 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.

Unit 6

213 # Tools for SAP System Administration

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 and the SAP Developer Network, 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:

- Name frequently-used administration functions in SAP systems
- The concept of the *SAP Solution Manager*
- Name the central functions of the *Computing Center Management System (CCMS)*
- Monitor performance attributes
- List some of the services provided on the *SAP Service Marketplace*
- Find and use SAP Notes in the *SAP Service Marketplace*
- List some of the services of the SAP Developer Network
- Find development news in the SAP Developer Network

Unit Contents

Lesson: Daily Tasks in System Management

214 Lesson Duration: 60 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
- The concept of the *SAP Solution Manager*
- Name the central functions of the *Computing Center Management System (CCMS)*
- Monitor performance attributes

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 depicted in the graphic “Important system administration transactions”.

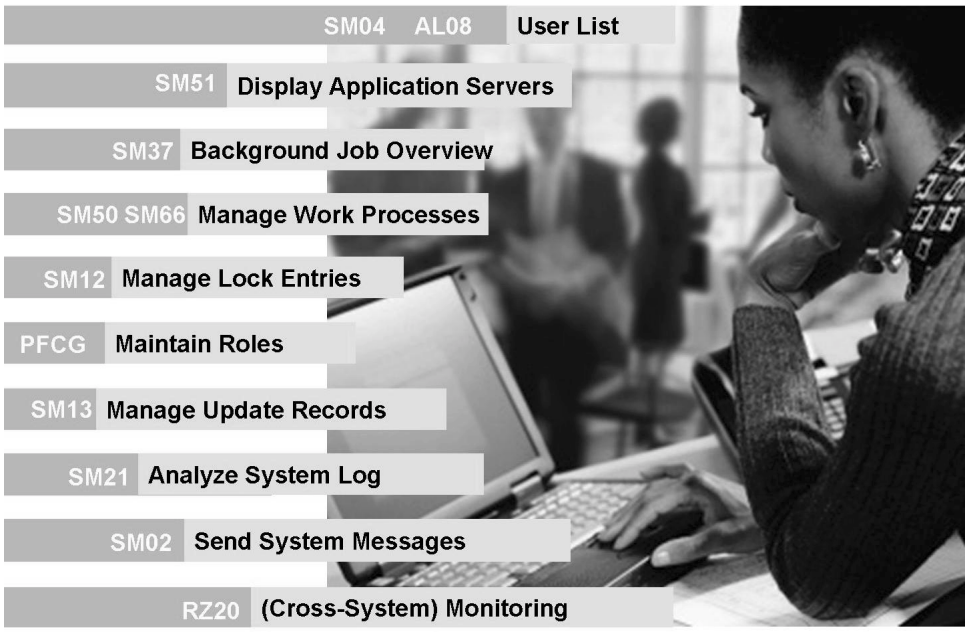

Figure 78: Important system administration transactions

Transaction **SM37** 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.

Transaction **SM51** displays all instances that are currently active in your SAP 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.

The user overview (transaction **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 (transaction **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.

The scope of the functions provided in transaction **SM50** and transaction **SM66** 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.

Transaction **SM12** 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.

Transaction **SM13** 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 successful initially. 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)*..

Transaction **SM21**, (the system log or SysLog) enables you to analyze 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.

Transaction **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.

Transaction **SU01**: You can use user maintenance to maintain individual user master records. The user administrator can use it to create new users or administer existing users, for example, assign new roles and authorization profiles. For the mass maintenance of users, you can use transaction **SU10**.

Transaction **PFCG**: The Profile Generator, available since *SAP R/3 3.1*, is used for the structured creation and assignment of authorizations via roles. Apart from the direct user assignment, the indirect user assignment is also supported by means of Organizational Management.

Transaction **RZ20** is discussed in detail in the next section of this lesson.

Possibilities of the Computing Center Management System (CCMS)

The *Computing Center Management System* (*CCMS*) provides a number of important administration functions:

At this point, you can introduce transaction HIER. The best thing to do is to navigate to component BC-CCM-MON. Point out that this nomenclature is used to classify documents in the online documentation as well as the SAP Note database.

- System administration (start, stop, reconfigure the system)
- Background processing and job scheduling
- Printer landscape configuration
- System tuning
- Database administration (backup and so on)
- 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.

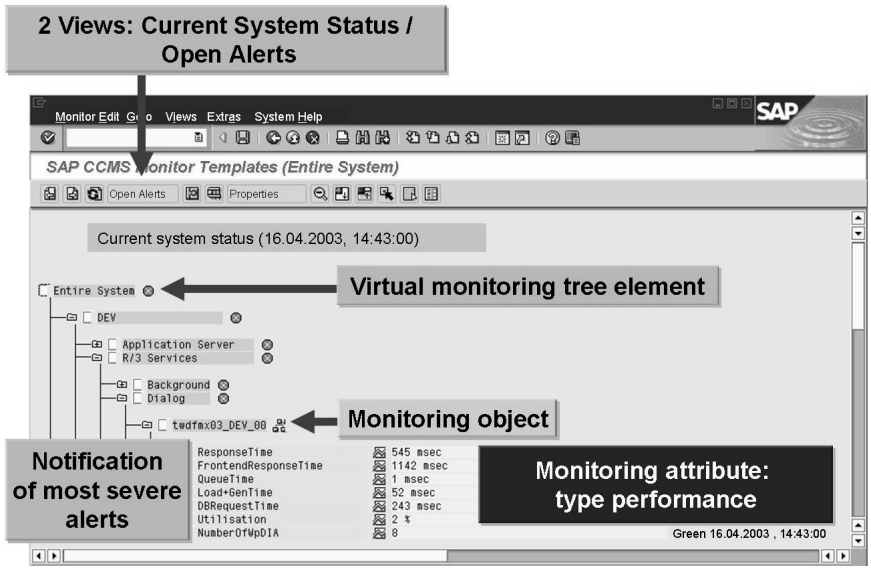

Figure 79: System monitoring with transaction RZ20

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 references to analysis options available elsewhere, such as the above-mentioned functions SM50, SM04 and so on. Open interfaces can be used to incorporate other system monitoring tools including non-SAP 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).

The Concept of the SAP Solution Manager

The *SAP Solution Manager* is a powerful tool.

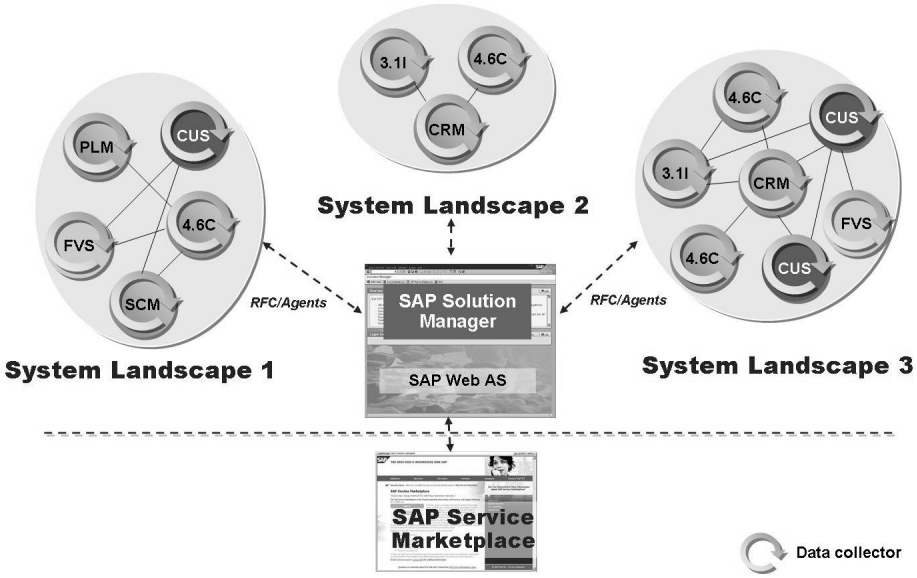

Figure 80: Monitoring multiple system landscapes using the SAP Solution Manager

The *SAP Solution Manager* can provide comprehensive support for 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 on the *SAP Solution Manager*, refer to the following address: http://service.sap.com/solutionmanager.

You can also subscribe to a newsletter on the *SAP Solution Manager* homepage, which will ensure that you are always up to date with news about the *SAP Solution Manager*.

The following courses provide basic information about working with the *SAP Solution Manager*:

The installation course is only offered by the Partner Academy.

- TSLM10 – *SAP Solution Manager Infrastructure and Installation*
- SMI210 – *SAP Solution Manager - Implementation Method Overview*
- SMI310 – *SAP Solution Manager - Implementation Tools in Detail*

Exercise 12: Daily Tasks in System Management

Exercise Duration: 25 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.

Task 5: User Maintenance

Assign role *SU53ONLY* to your own user.

1. Restart the user maintenance, transaction SU01. Open your own user master record SAPTEC-## in change mode. Go to the *Roles* tab and enter role *SU53ONLY*. Save the changes. Leave the transaction with function key *F3* and call up the user menu again. You should be able to see the new menu options of the new role.

   

   Hint: Under *SAP Easy Access* settings, switch on the display of the *first level*. That way, all assigned roles are listed separately. Without this setting, redundant menu entries are suppressed.

2. Alternatively, the role can be assigned via the profile generator. Start transaction PFCG and change the role *SU53ONLY*. Go to the *user* tab. In the *user ID* column, enter your own user. Save the result. Call up the user comparison using the function key of the same name and carry out a complete termination. The status of *user comparison* key should now be green.

Solution 12: 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”.

Continued on next page

Task 5: User Maintenance

Assign role *SU53ONLY* to your own user.

1. Restart the user maintenance, transaction SU01. Open your own user master record SAPTEC-## in change mode. Go to the *Roles* tab and enter role *SU53ONLY*. Save the changes. Leave the transaction with function key *F3* and call up the user menu again. You should be able to see the new menu options of the new role.

   

   Hint: Under *SAP Easy Access* settings, switch on the display of the *first level*. That way, all assigned roles are listed separately. Without this setting, redundant menu entries are suppressed.

   a) See exercise text.

2. Alternatively, the role can be assigned via the profile generator. Start transaction PFCG and change the role *SU53ONLY*. Go to the *user* tab. In the *user ID* column, enter your own user. Save the result. Call up the user comparison using the function key of the same name and carry out a complete termination. The status of *user comparison* key should now be green.

   a) See exercise text.

Lesson Summary

You should now be able to:

- Name frequently-used administration functions in SAP systems
- The concept of the *SAP Solution Manager*
- Name the central functions of the *Computing Center Management System (CCMS)*
- Monitor performance attributes

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

226 Lesson Duration: 30 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:

- List some of the services provided on the *SAP Service Marketplace*
- Find and use SAP Notes in 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 an **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). Caution: 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

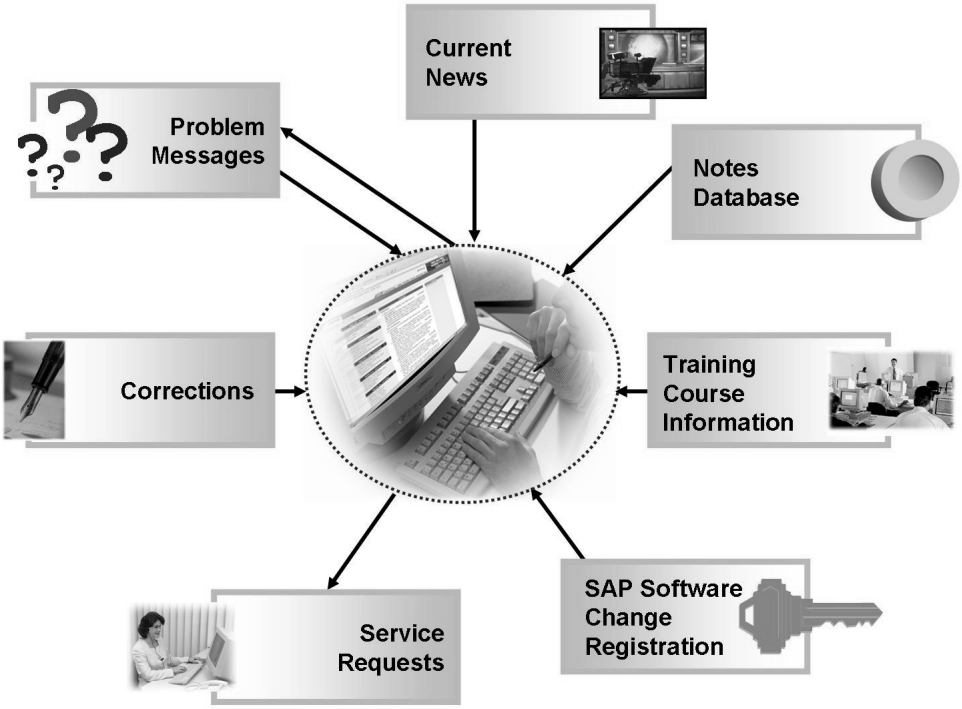

Figure 81: service.sap.com – the SAP Service Marketplace

Some *SAP Service Marketplace* functions are named below. The focus here is on services that are particularly relevant to *SAP Web Application Server*.

Administering users

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).

Developer key

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* ("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*.

Notes Database

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).

Note Assistant

All SAP systems as of the *SAP Web AS 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.

Corrections and messages

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.

Access to your SAP system and to other services

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*).

Software requirements

Are you looking for information on **software requirements** (operating system release or database release) for a specific SAP solution? You can find the answer using the Quick Link */platforms*.

Information on the latest developments

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**

Planing hardware investments

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.

Contacts

If you need to get in touch with an **SAP partner**, you can find their contact details in the Partner Directory at */partnerdir*.

Training offer

/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.

231

Exercise 13: SAP Service Marketplace (Optional)

Exercise Duration: 10 Minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Logon to the *SAP Service Marketplace* call up some function.

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: Using the SAP Service Marketplace

Participants must have own S users for logging on.

Log on to the *SAP Service Marketplace*:

1. Start a Web browser.
2. Go to the *SAP Service Marketplace*.
3. Start the Notes search by double-clicking the *SAP Search Notes* quick link in the *SAP Support Portal* area.
4. Logon to the *SAP Service Marketplace* with your own S user.

Continued on next page

There is no generic S user, which you can give to course participants at this point.

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 6.40* kernel?

Solution 13: SAP Service Marketplace (Optional)

Task 1: Using the SAP Service Marketplace

Participants must have own S users for logging on.

Log on to the *SAP Service Marketplace*:

1. Start a Web browser.
   a) You can normally access the Web browser (for example, *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. Start the Notes search by double-clicking the *SAP Search Notes* quick link in the *SAP Support Portal* area.
   a) Solve the task as described. A dialog box is displayed for the logon.
4. Logon to the *SAP Service Marketplace* with your own S user.

There is no generic S user, which you can give to course participants at this point.

   a) Enter the logon data of your S-user.

Continued on next page

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*.

2. With which databases can you use the *SAP Web Application Server 6.40* kernel?

   a) There are various ways of answering questions about platform availability. Two approaches are outlined below:

   - Enter the search criteria **`Kernel 6.x 6.40`** to restrict the Notes search (Quick Link */notes*). To reduce the number of hits, also enter the application component **`XX-SER-SWREL`**.
   - The other approach uses the Quick Link */platforms*. You can use the link *OS/DB combinations for downward compatible SAP Web AS 6.40 kernel* to navigate to the document on the *SAP Web AS 6.40* availability.

   You can find details on the operating systems released for the 6.40 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
727683	MaxDB

Lesson Summary

You should now be able to:

- List some of the services provided on the *SAP Service Marketplace*
- Find and use SAP Notes in 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.

Lesson: SAP Developer Network

Lesson Duration: 30 Minutes

Lesson Overview

This lesson introduces you to the SAP Developer Network and the offers that can be found there.

Lesson Objectives

After completing this lesson, you will be able to:

- List some of the services of the SAP Developer Network
- Find development news in the SAP Developer Network

The SAP Developer Network was created in the context of the SAP NetWeaver initiative. The content of SDN is mainly aimed at developers and consultants. The SAP DN is an important enhancement to the already existing online information offer of SAP.

Business Example

As a developer of SAP software, you are interested in information on current developments and trends. The SAP Developer Network is a source of information that is available on the Internet.

The Target Audience of the SAP Developer Network

The SAP Developer Network (SDN) in an online community of developers, consultants and IT architects and integrators. In short, it provides and excellent meeting point for SAP experts. Registered members of the SDN can access information from many technical areas. However, the focus is on topics from the *SAP NetWeaver* area.

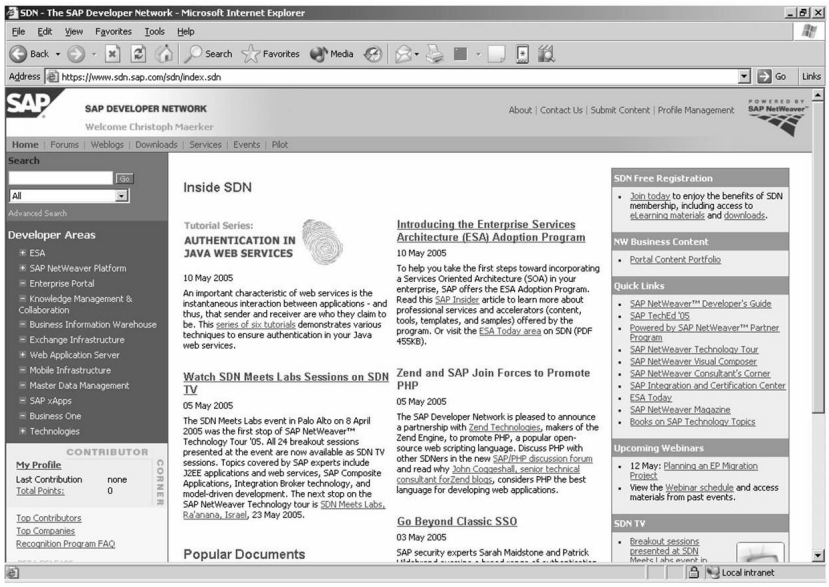

Figure 82: SDN Logon Screen

Services in the SAP Developer Network

SAP offers its customers a range of information options on the Internet. The following websites should be listed as examples:

- The online documentation at http://help.sap.com
- The *SAP Service Marketplace* at http://service.sap.com
- The information portals SAP Info (http://www.sap.info) and SAP InsiderOnline (http://www.sapinsideronline.com)
- The SAP Developer Network at http://sdn.sap.com

The SAP Developer Network is the latest addition to the SAP online information portals. Es ergänzt die bestehenden Informationsquellen, speziell um Themen aus dem Bereich des *SAP NetWeaver*. Why should you join SDN, what are the advantages and offers? The function of the SAP Developer Network can be summarized as follows:

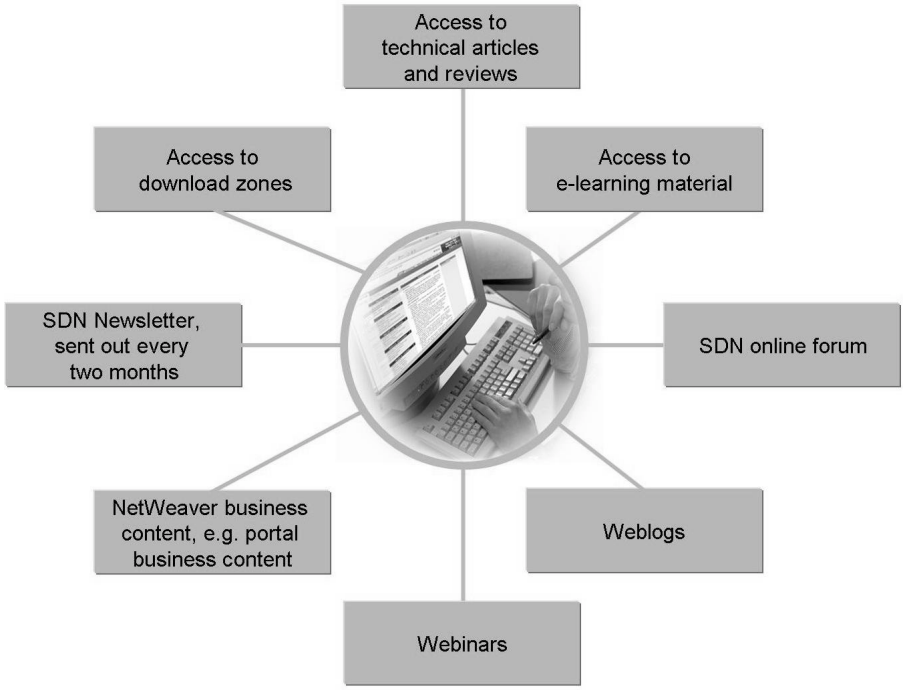

Figure 83: sdn.sap.com, the SAP Developer Network

- Access to technical articles and reviews
- Access to e-learning material
- SDN online forum
- Web logs
- Webinars
- SDN newsletter, distributed every two months
- NetWeaver Business Content, such as Enterprise Portal Business Content
- Access to download zones

The SAP Developer Network is thus a valuable addition to the existing online information sources of SAP.

Exercise 14: SAP Developer Network (Optional)

Exercise Duration: 10 Minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Log on to the SAP Developer Network and systematically search for contents

Business Example

You require current information on secure RFC connections.

System Data

System:	The SDN on the Internet, the URL is http://sdn.sap.com.
Client:	Not required.
User ID:	Please use your own SDN user.
Password:	The password is individual and you should know it.
Set up instructions:	

Task: Logon and Navigation in SDN

You want to find out about setting up secure RFC connections.

You can also search for developer notes without logging on as an SDN user.

1. Please log on to SDN with your own SDN user and search for documents on the subject of *Secure RFC*.

Result

You see a hit list with relevant documents, including the PDF file called *Securing RFC Connections*.

Solution 14: SAP Developer Network (Optional)

Task: Logon and Navigation in SDN

You want to find out about setting up secure RFC connections.

You can also search for developer notes without logging on as an SDN user.

1. Please log on to SDN with your own SDN user and search for documents on the subject of *Secure RFC*.

   a) In the web browser, open URL http://sdn.sap.com and log on with your SDN user. Enter the search text **`Secure RFC`** and start the search.

Result

You see a hit list with relevant documents, including the PDF file called *Securing RFC Connections*.

Lesson Summary

You should now be able to:

- List some of the services of the SAP Developer Network
- Find development news in the SAP Developer Network

Unit Summary

You should now be able to:

- Name frequently-used administration functions in SAP systems
- The concept of the *SAP Solution Manager*
- Name the central functions of the *Computing Center Management System (CCMS)*
- Monitor performance attributes
- List some of the services provided on the *SAP Service Marketplace*
- Find and use SAP Notes in the *SAP Service Marketplace*
- List some of the services of the SAP Developer Network
- Find development news in the SAP Developer Network

Test Your Knowledge

1. Which of the following transaction codes can you use to view and edit user logons?

   Choose the correct answer(s).

   - □ A SM04
   - □ B SM30_USERS_SSM
   - □ C USER
   - □ D AL08
   - □ E SM51

2. 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

3. 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

4. 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

5. 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

6. You can download developer keys from SDN.

   Determine whether this statement is true or false.

   - □ True
   - □ False

Answers

1. Which of the following transaction codes can you use to view and 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.

2. 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.

3. 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).

4. 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.

5. 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.

6. You can download developer keys from SDN.

Answer: False

You apply for developer keys using the *SAP Service Marketplace* not in SDN.

Unit 7

247

SAP NetWeaver and Enterprise Services Architecture

Compared to previous versions of this course, the last chapter of SAPTEC has been added. In the end, this reflects the increasing complexity of the software environment. The instructor provides a broad overview of the topics listed here without discussing the matter in too much detail. You can refer to course *SAPNW – SAP NetWeaver Overview* for details.

After participants have been given an overview of the architecture of SAP systems, user interfaces and the communication between different systems in the preceding chapters, the first lesson now intrduces them to the concept of SAP NetWeaver in more detail than in the first unit. The Application Platform area with SAP Web AS has been omitted deliberately here, as the SAP Web AS was already covered extensively in this course. This is followed by a preview of the future with the Enterprise Service Architecture.

Unit Overview

This final unit introduces the concept of *SAP NetWeaver* and its elements in more detail. Apart from that, there is a preview of the future in which the meaning of the Enterprise Service Architecture (ESA) is discussed.

Unit Objectives

After completing this unit, you will be able to:

- Name the integration aspects of *SAP NetWeaver*
- Explain the basic concepts of the individual integration levels of *SAP NetWeaver*
- Explain the evolution from *SAP R/3* to *mySAP ERP* and the Enterprise Services Architecture.
- Describe the significance of the Web Services within the Enterprise Services Architecture

Unit Contents

Lesson: SAP NetWeaver – An Overview

Lesson Duration: 90 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:

- Name the integration aspects of *SAP NetWeaver*
- Explain the basic concepts of the individual integration levels of *SAP NetWeaver*

This lesson briefly introduces all NetWeaver components (except for the actual Web Application Server). Exercises and system demonstrations are not planned here. Caution: The content covered brushes on topics for which multi-day seminars are required for merely acquiring the basic knowledge. Do not underestimate this. If necessary, always refer to course SAPNW or to the introductory seminars for the individual components, such as SAPBI, SAPEP, BIT100, BIT400, and so on. The instructor notes contain much background information for instructors, which is to help especially if there are questions. The contents of the notes should not be part of the presentation in any case.

Business Example

After a merger with Rubble AG & Co. KG, which specializes in smaller markets, globally active Flintstone Industries Ltd. has a very heterogeneous IT infrastructure. To map business processes not only SAP systems are used but also software components from other providers as well as business applications that have been developed internally. After the merger, Flintstone Industries wants to map the changed business processes to the existing IT landscape as efficiently as possible. In doing so, as much flexibility as possible should be ensured for the future. Furthermore, the very high costs of interface maintenance between the individual software components should also be drastically reduced at the same time. The master data of the formerly separate companies is to be managed centrally in future. Top management also demands uniform reporting across all areas. Finally, sales employees are to receive a central portal that always shows them the current status of the sales orders and always supplies employees with the latest information.

Current Challenges for Companies with IT Landscapes that have Grown over a Number of Years

These days, companies want to reduce costs, find new ways of increasing turnover and profitability and be able to flexibly adjust to all types of changes. In this context, the question of how to adjust or integrate existing applications and flexibly implement new applications plays a central role. Existing investments shall be used optimally, at the same time new business processes have to be supported quicker and in a more intelligent way.

Cisco Systems GmbH, with more than 60 acquisitions is an example of this. Cisco must be able to provide its customers with consistent information on the order status across all product lines and structures of business units. The information for this is kept in numerous old and new software applications. If business structures and customer requirements change, the effort and costs involved in maintaining such a service can become immensely high.

Legal requirements demanding a higher degree of traceability as well as changes to processes are a good example of the external “pressure to conform” affecting companies. The following could be mentioned here: Sarbanes-Oxley Act for US listed companies (proof of processes for creating the balance sheet) or requirements in the area of deregulation (for example, E.ON may not put other suppliers in a worse position than itself) or Basel 2 in financial accounting (proof of the own stability by banking institutions).

These days, realistic system landscapes often consist of many systems. The business processes that are to be mapped in these complex system landscapes contain process steps running on different systems. For all system transitions, sending systems are connected to receiving systems by means of interfaces (point to point connection). Different interfaces are often implemented using the technology that is considered to be ideal for the respective interface. Due to this, administrators not only have to take care of a complex system environment but must also know many different technologies.

In the following graphic, every interface is represented by a connecting line. This illustrates the complexity of an integration solution.

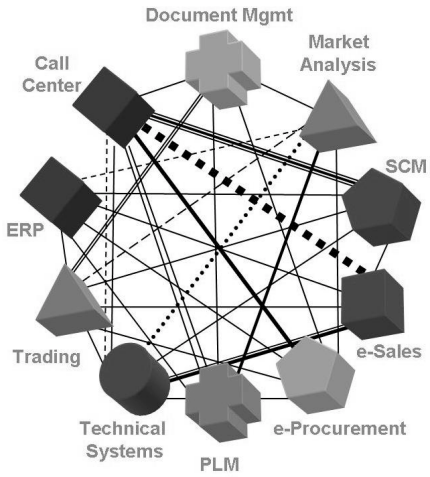

Figure 84: Challenges of Current IT landscapes

Studies on how long it takes to change existing business processes or introduce new processes still determine times ranging from months to years, depending on the process. People asked attribute about a third of this to the inflexibility of IT. With **SAP NetWeaver** it is much quicker, easier and more flexible to implement and adjust complex business scenarios. For example, using *SAP NetWeaver* reduces the complexity of system landscapes by

- being a **single** platform for integrating information and systems
- providing functions that make time-consuming, expensive integration projects unnecessary
- ensuring compatibility with .NET and WebSphere
- making it possible to increase the flexibility of business processes with the new Enterprise Services Architecture concept

There are certain standards that regulate how certain components work together. For example, there is a standard for the data exchange between BW systems or standard for the communication between portals. But there is currently no standard regulating how a BW system can be connected to a portal (more or less) at the push of a button. And that is the solution approach of SAP NetWeaver and that is the advantage of the SAP integration solution.

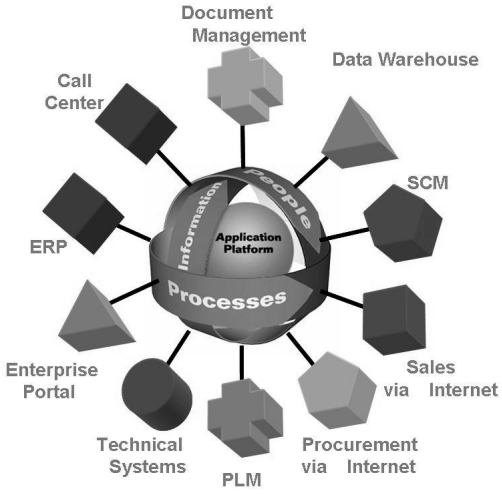

Figure 85: Integrated System Infrastructure

Caution: *SAP NetWeaver* consists of a number of modules that can also be regarded separately. The most important thing though is that the whole is more than the sum of its individual components. *SAP NetWeaver* enables you to flexibly develop business processes on tailored components without having to give up existing investments.

You should always point out the increased flexibility offered by the SAP NetWeaver Platform, if possible based on scenarios given by the participants.

SAP NetWeaver

SAP NetWeaver is the technical foundation on which almost all *mySAP* solutions are currently based. *SAP NetWeaver* is the functionally enhanced successor of the SAP application platform *mySAP Technology* and serves as the basis for the Enterprise Services Architecture, in order to meet requests for flexibility and integration between systems, interfaces, users and processes. It connects information, business processes and people across system and organizational boundaries. It is the central tool for reducing the total cost of ownership (TCO) of complex system infrastructures.

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

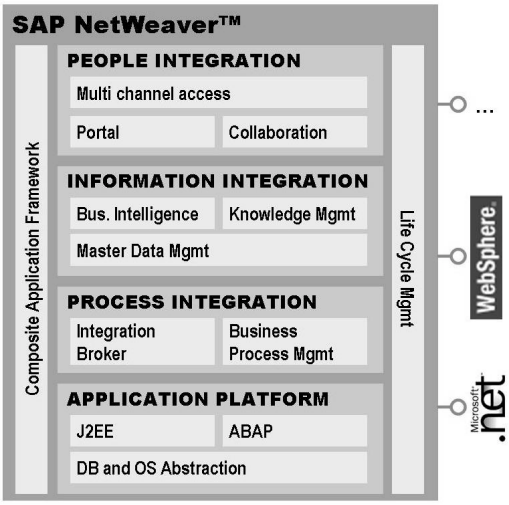

Figure 86: SAP NetWeaver - Overview of Components

On the following four **integration levels** *SAP NetWeaver* provides the core functions for the technical infrastructure of business solutions:

- People Integration
- Information Integration
- Process Integration
- Application Platform

SAP NetWeaver also supports cross-application software, so-called composite applications or xApps (xApps connect heterogeneous systems in continuous cross-function processes so that the underlying applications can be more or less ignored). Furthermore, different software interfaces ensure full interoperability of applications that are running on *Microsoft .NET* and *IBM WebSphere*.

The following additional slide is very handy for mapping the integration levels to software components. At the same time, it introduces most important courses on the respective topics.

Integration Layer	Who To whom it applies	What Central keywords	Where Main component	How Most important courses
People Integration	All users	Bringing people together with information and systems	EP	SAPEP, EP200, EP300
Information Integration	Managers, Key Users, Administrators, Developers	Preparation and consolidation of data/data views	MI MDM BW	SAPBI, BW310, BW305
Process Integration	Administrators, Developers	Modeling processes, distributing and grouping data (in accordance with modeling)	MDM XI	BIT400, BIT430, BIT450
Application Platform	Administrators, Developers	Work platform for all SAP NetWeaver components	Web AS	SAPTEC, ADM100, ADM200, BC400, JA300

SAPNW

Figure: SAP NetWeaver: Who, What and How...

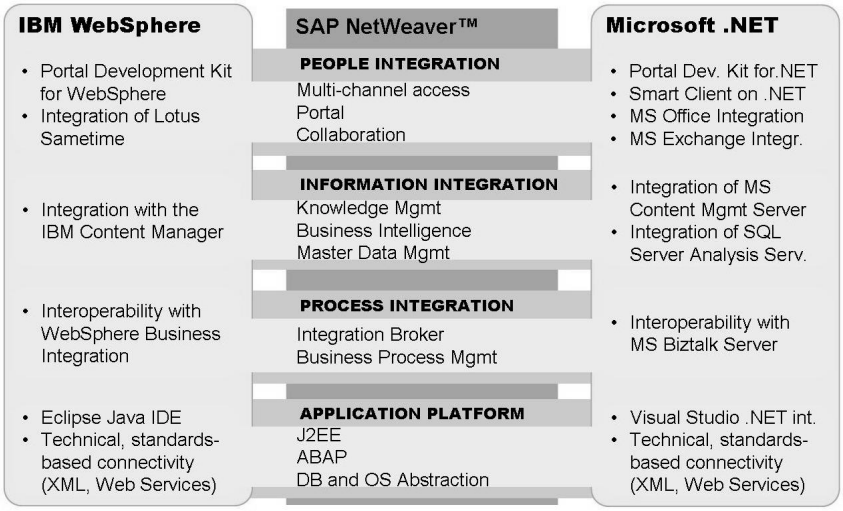

Figure 87: SAP NetWeaver - Interoperability

Examples of xApps are SAP xRPM (Resource and Program Management), SAP xPD (Product Definition), SAP xEM (Emissions Management) or SAP GTS (Global Trade Services). If requested, you can deal with xRPM in more detail. xRPM is used for multi-project management, that is the cross-administration of different projects in different project systems. SAP xApp Resource and Program Management integrates data from existing systems for project management,

human resources, financial accounting, cost accounting and time recording to provide management, project manager, resource manager and project members with an overview of the project portfolio including a simple drilldown function.

Due to the complete integration of the different levels, SAP xRPM is a really good example of what ESA and NetWeaver mean. SAP xRPM enables the effective monitoring and analysis of large projects. This includes risk recognition and rating. Customers can define their own risk factors and then analyze all current projects in the operational systems according to these factors. xRPM provides a type of resource management with which it is possible to specifically search for employees with certain expertise. These employees can then be requested immediately for a project and once the supervisor has given his approval (obtained via the workflow), both the HR data of the employees is maintained and appropriate costs are posted to the project. Of course, this assumes that the employees have been maintained accordingly in the xRPM system. This only has to be done once.

Information on SAP xRPM is available in the help portal or on a CD that can be purchased. You can purchase the CD from Tanja Foerster. It contains an offline demonstration in English. Refer also to the following link: http://www.sap.com/germany/solutions/xapps/xrpm/index.aspx

---

The following covers the individual integration levels in more detail, starting with people integration.

Integration Level People Integration: Using an Enterprise Portal

---

Possible introduction to the following section:

Modern companies typically use many different applications. Every employee has to log on to and log off from these applications several times a day. This is both time consuming and annoying. In addition, forgotten passwords make work more difficult, both for the employees themselves as well as for the IT department. Furthermore, there is often an “information overload”: Too many intranet pages, too many local servers with too many files. As a response to this, a central view of applications is often requested for management as well as employees. SAP Enterprise Portal provides the solution to this.

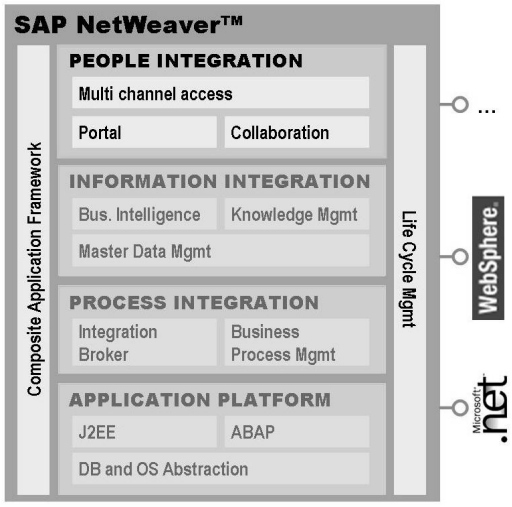

Figure 88: People Integration

An enterprise portal is the gateway to the end user. Based on the end user view, this section illustrates, how a user-friendly interface makes it possible to access relevant data from very different systems. Furthermore, it illustrates the meaning of role-based user and content management and shows that using business packages can significantly reduce the amount of resources required for implementing a portal.

An enterprise portal offers a central point of access to information, applications and services in your enterprise. All SAP and non-SAP systems, data warehouses, desktop documents as well as web contents and web services are brought together on one uniform interface. By using the single sign-on, users benefit from the convenient authentication and communication between IT systems. Knowledge management in the portal turns unstructured data into important information (for the enterprise), collaboration facilitates the collaboration of people across enterprise borders and personalization adapts the structure to individual user requirements.

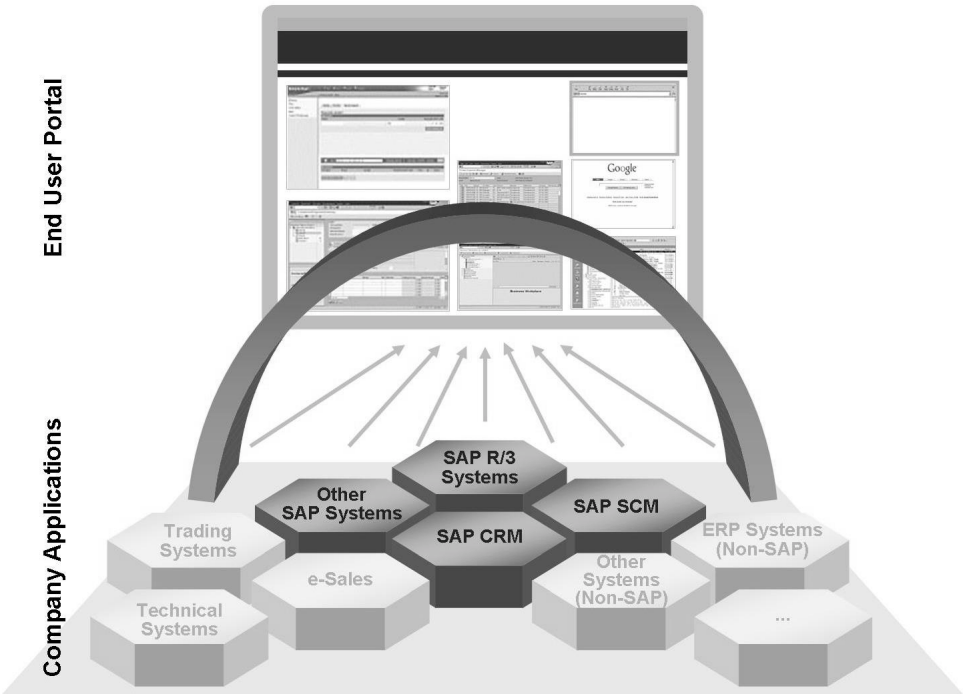

Figure 89: SAP Enterprise Portal

You can use the catchphrase that a portal should and can provide the right information and the right functions to the right processor at any time and from any location.

To do this, *SAP Enterprise Portal* provides an infrastructure you can use to retrieve and edit data from information sources in your company and the Internet by means of **iViews**. SAP provides *iView* templates but you can also create customer-specific *iViews*.

Furthermore, the *SAP Enterprise Portal* provides powerful **search functions** with which you can selectively and intuitively retrieve files and documents from different sources of information such as local database applications, websites or pools of enterprise documents.

Collaboration functions allow you to use common resources and common content. The individual tools and services include virtual project rooms and tools for real-time interaction, irrespective of the geographical distance. You can use discussion forums to exchange data and information with colleagues whose workplace is not in the same spot or maybe even in a different time zone.

The core functions of *SAP Enterprise Portal* are written in Java. Hence, you need a J2EE runtime environment such as the one provided by the *SAP Web Application Server Java*. *SAP Enterprise Portal* uses an open architecture. The standards it supports include SOAP, UDDI, XML. The portal has powerful security functions

including extensive support of directory services, digital certificates and SSL protocol (Secure Socket Layer). *SAP Enterprise Portal* is highly scalable and is thus also designed for large numbers of users. Mobile devices are also supported.

Roles in the SAP Enterprise Portal

Roles determine, which navigation paths (specified using the file structure) can be used to access which ***content*** (specified using the integrated *iViews* and pages). Roles can be assigned to individual users or groups of users.

Note: The term *content* covers all contents that are available to a user in his/her role-based portal view. The most important *content* objects are:

iView

Program that determines data from different sources of the enterprise or Internet, displays it in the *SAP Enterprise Portal* and, if applicable, makes it available for processing.

Page

Layout of one or more *iViews*.

Workset

Collection of tasks, services and information that can be used to create roles. Worksets consist of *iViews* and pages, arranged in a folder structure that determines the navigation paths.

Users use the portal either as their central workplace with access to different applications or the portal serves them as the user interface within a specific application. For example, this is the case in *SAP Master Data Management (SAP MDM)* as well as in xApp *SAP xRPM* for multi project management for the employee or manager self service work center.

These roles for **administering** users and content are generally separated. Administrators for users and roles define both objects according to the enterprise requirements in the system and assign roles to users. Content administrators define which content is available in the system, ensure that it is administered properly and decide which roles can work with which content.

You have to distinguish between the role maintenance in the portal and the **authorizations in connected systems**. The portal uses the procedure authorizations of the respective application and thus does not have to transfer authorization profiles to the enterprise portal. This increases security and reduces the amount of work required to administer authorizations. At the same time, it ensures that users can only access data and information according to their authorization profile.

In order to avoid duplicate work, it is possible to exchange roles between the portal and an SAP system. Here, only the menu of the role is stored on the portal side but not the corresponding authorization data from the SAP system, which remain in the back end.

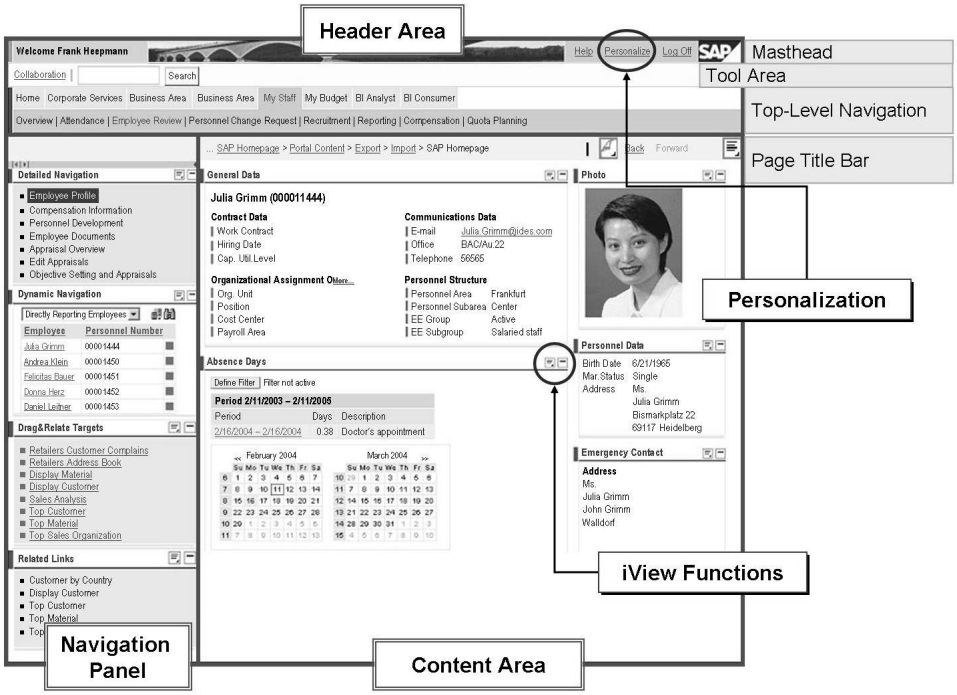

Figure 90: Look&Feel of the SAP Enterprise Portal

The portal view can be customized. All UI elements are in turn iViews and can be moved to where the user wants them. For example, you can move the detail view to the right part of the screen.

If a demo is requested: http://idesportals.wdf.sap.corp:1080/ There, you can access portals with predefined users in different releases and SP levels.

The following graphic provides an overview of the architecture of the SAP Enterprise Portal.

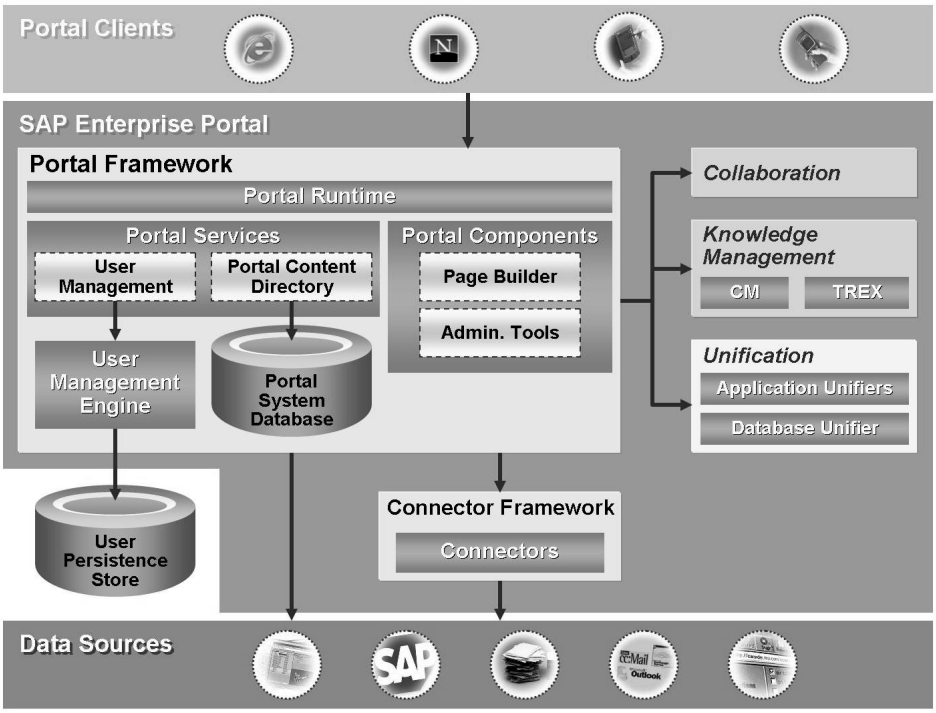

Figure: The Architecture of the SAP Enterprise Portal

Discuss the graphic using the following memos.

The Portal Runtime (PRT) is the core of the portal. From a technical point of view it is a Java application running on the J2EE engine. It forms the runtime environment for all portal components and services.

All portal cost components are HTML creating Java applications, such as the Page Builder, which constructs the portal pages and sends them to the portal client or the admin tools used to administer roles, users, business packages and so on.

Portal services act as an interface to other portal components or services for exchanging procedures or data. The Portal Content Directory (PCD) for example, serves as an interface to the Portal System Database (in which all portal“data” is stored). The User Management Service is another example. It serves as the interface between PRT and the User Management Engine (UME), a separate software component that stores the user data in the User Persistence Store.

You must distinguish the user management in the portal from that in other SAP applications such as SAP R/3, CRM and so on. Portal users can be “stored” in very different locations - in the portal database, on the LDAP servers or in SAP systems.

Business Packages

In addition to about 100 roles, SAP also provides standard content in ***Business Packages***. *Business Packages* contain predefined portal content, that can be used to call up *transactions* and *reports* from all sorts of systems. Furthermore, it also contains documents and information, based on the *roles* for users, managers and specialists.

Business Packages facilitate the work of your content manager and significantly reduce the implementation time of your enterprise portal as they enable portals to be created without additional development work. So while your competitors are still busy programming their enterprise portal platform, you are already enjoying the advantages of using the *SAP Enterprise Portal*.

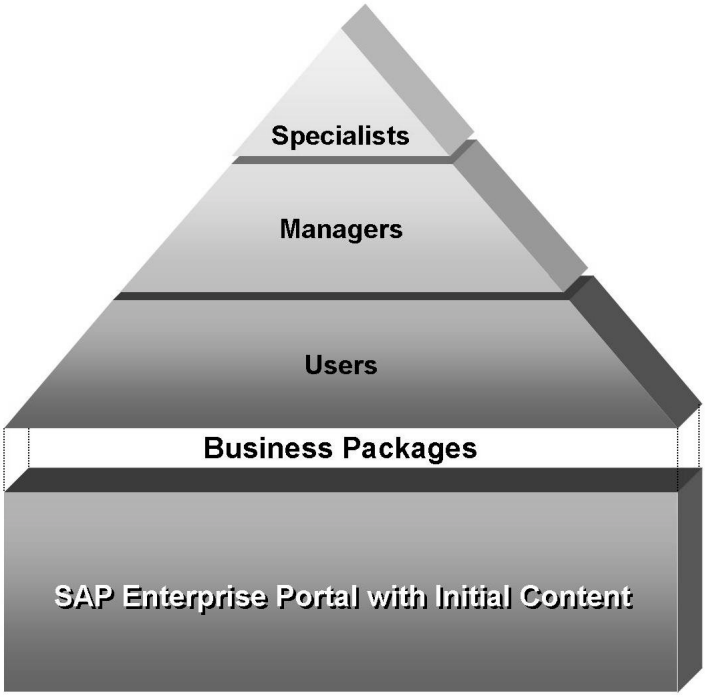

Figure 91: Target Groups of Business Packages

Business Packages are classified according to the following target groups: end users, specialists and managers.

Business packages for users

Contains the tasks that a portal user may carry out in the portal, irrespective of his other role in the enterprise. The tasks help beginners to familiarize themselves with the portal as quickly as possible and thus increase the acceptance of the enterprise portal. The content includes functions regarding e-mailing, task lists, calendar administration, travel expense settlement, administration of benefits, **employee self services**, e-learning and the search in the employee directory.

Business packages for managers

Decision makers can use the content of these packages for efficient analyses and for gaining decision-relevant information. Furthermore, they offer extensive tools for planning and administering the portal use and the budget. Line managers, team managers and project managers can thus reduce the time they have to spend on administrative tasks and focus on strategic tasks. The most wide-spread *business packages* for managers include the **Manager Self-Services** (MSS).

Business packages for specialists

These *business packages* are aimed at the experts from different departments in the enterprise such as sales, human resources, marketing, finance and production. They provide appropriate analysis tools, which enable them to act quickly based on the right information. But they can also provide operational tools, such as campaign management for marketing employees.

You can buy *Business Packages* from the *iView Studio* , a **central marketplace for portal content**. You find the *iView Studio* at the following address www.sdn.sap.com. As a registered user, you will find the individual packages there. You can display and download contents under the corresponding roles.

Mobile Infrastructure

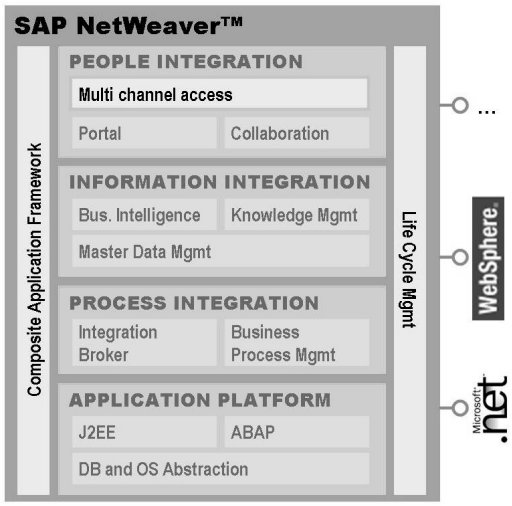

Figure 92: Mobile Infrastructure

SAP Mobile Infrastructure (SAP MI) is a technology solution of *SAP NetWeaver*, which is the basis for the "SAP Solutions for Mobile Business". They are usually an enhancement to an existing SAP application, as demonstrated by the following graphic. For example, there is an enhancement for SAP Human Resources, with which travel data and working times can be entered on the go. In the same way,

service employees can be directly informed of new orders via a mobile device and they can confirm data directly at the customers instead of writing down data on sheets of paper and entering them into the back end system later on.

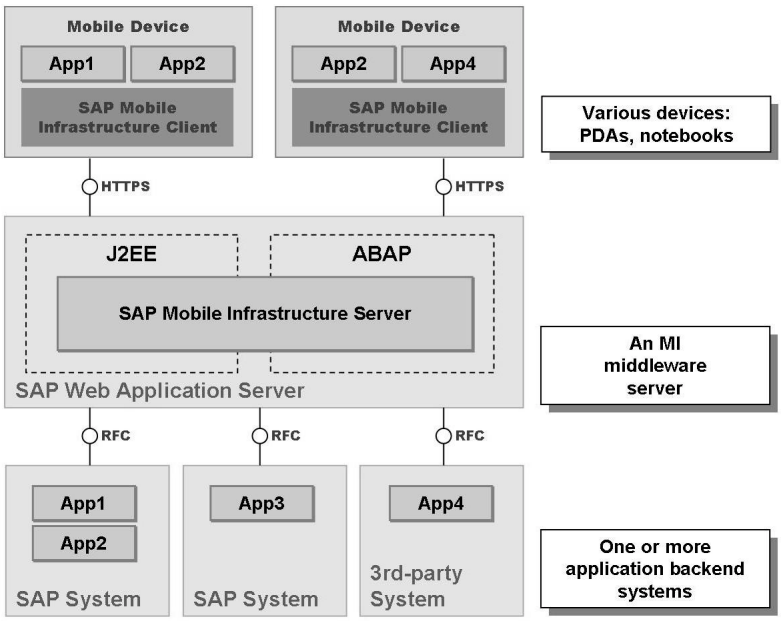

Figure 93: Architecture of the SAP Mobile Infrastructure

Mobile business increases flexibility as well as making it easier to provide information to field sales representatives. A sales employee can dynamically access customer and order data without having to download and save it from the backend before visiting a customer. A third point is getting rid of "piles of paper" where data has to be entered into the IT systems afterwards. This speeds up processes, reduce the amount of work required for field sales representatives and is less error-prone as the data is entered when it is created.

However, you can also use *SAP MI* to "mobilize" non-SAP based applications. *SAP MI* is locally installed on a mobile device and is equipped with a web server, database layer and a dedicated business logic. Hence, remotely working employees never have to wait for a network connection to complete time-critical business transactions but can work offline. To synchronize the data on the mobile device with the back end, *SAP MI* provides tools for synchronization and data replication. *SAP MI* is equipped with a Java Virtual Machine and provides an open programming model with which developers can develop mobile applications. This open system architecture facilitates platform independence of mobile devices as well as networks and supports mobile devices such as personal digital assistants (PDAs), laptops and smart phones.

The technology of CRM mobile clients (CRM Field Sales) has nothing to do with SAP MI. It only runs in SAP CRM, is .NET based and is focused on working with large amounts of data on laptops. Mobile Infrastructure in SAP NetWeaver in contrast, is aimed at tasks that require less data and can be stored on PDAs.

SAP MI wants to offer introductory courses for Mobile Intelligence from quarter 04 2004. One course is on the administration and the other is on developing customer-defined mobile solutions. See also SAPNet quick link nw-mi and rkt-mi. See the following link regarding current MI scenarios: http://www.sap.com/solutions/mobilebusiness/index.aspx

Integration Level Information Integration

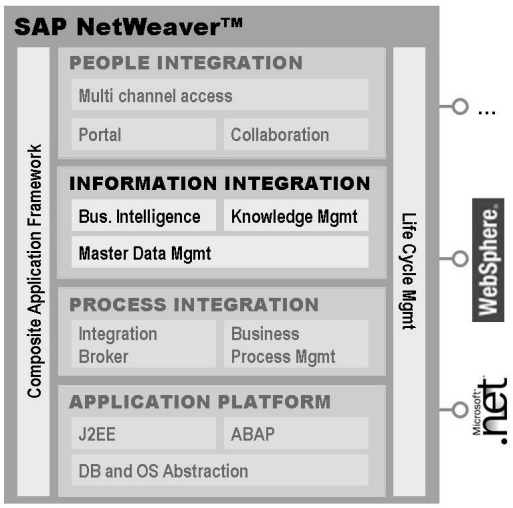

Figure 94: Information Integration

This section shall explain the basic idea of SAP Business Intelligence and illustrate the meaning of the business content provided by SAP.

For each solution supplied by SAP, there are options of reporting via the data of the respective application. Usually, the application provides a large number of standard reports for this purpose, but user-defined reporting is also possible. Customers can use the query interface or program their own reports in the *ABAP Workbench* and thus read the data of the production system as they want. This type of reporting can also still be used with *SAP NetWeaver*.

By now, more than 6000 customers successfully use the software solution *SAP Business Intelligence* (system component *SAP BW*) either in parallel to this or exclusively. The reasons for that include the increasing requirements for integrated solutions for the enterprise-wide analysis of data. In times of globalization and market expansion, it is important to have access to the relevant information from one's own enterprise at any time and to be able to analyze it flexibly, including in aggregated form and without placing a performance strain on the transactional system.

Make sure you make the techies amongst the participants well aware of the performance aspect: Increasingly complex queries via the business data in the OLTP system, increase the load on these systems, which were not set up for extensive data warehousing originally. In addition to this, the BW also offers functions that do not exist in the “old” form of reporting (such as complex hierarchies).

Release 6.40 implicitly contains the SAP BW component in SAP Web AS, that is, it is available after the installation of the SAP Web AS. In particular for smaller installations and simple reporting tasks, BW reporting could thus be executed directly on the OLTP system, that is, on the database that the OLTP uses. Depending on the database, the performance is better or worse. Oracle, for example, meets the different requirements (performance relevant settings) for OLTP and OLAP systems by using different user schemas.

In heterogeneous system infrastructures, the extraction and preparation of consolidated transaction and master data from SAP systems and source systems by other suppliers are particularly challenging. Apart from integrated data procurement, options for detailed data analysis and the multi-media display of the analysis results are required to meet the increasing quality requirements for enterprise information.

Collaboration of OLTP and OLAP

You have to look at the transaction-oriented OLTP (Online Transaction Processing) and the analysis-oriented OLAP environment (online analytical processing) as integrated. Huge amounts of information are created from the data for business processes, which are not easily used for practical analysis. Hence, the data is cleaned first (cleansing) and, due to its different origins, it is technically and semantically formatted (homogenized). The resulting analyses can, in turn, be used to generate knowledge that can help enterprise management to define its enterprise strategy and the business processes derived thereof. The following graphic illustrates this cycle:

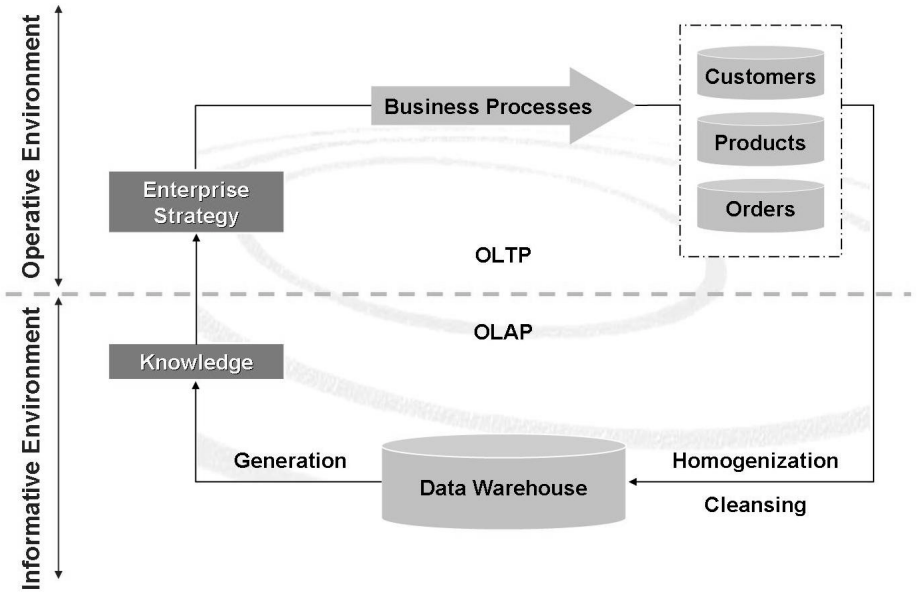

Figure 95: Delimitation: Operational/Informational Environment

OLAP stands for Online Analytical Processing. Typical OLAP applications are data warehouse systems such as the SAP BW. OLTP is the abbreviation of Online Transaction Processing. Typical OLTP applications are enterprise resource planning systems such as mySAP ERP. There are fundamental differences between OLTP environments and OLAP environments. This can be illustrated with the following terms:

- **Degree of detail**

  While the OLTP level stores data at a very high level of detail, the data in the data warehouse is aggregated for performant access.

- **History**

  By means of archiving data in the OLTP area, data is kept with a minimal history. In the data warehouse area, comprehensive historic data is required.

- **Changeability**

  Frequent changes of data distinguishes the operational area, whereas the data for the analyses is "frozen" from a certain point in time.

- **Integration**

  In contrast to the OLTP environment, the requirements on general, integrated information for analysis are very high.

- **Normalization**

  Due to the reduction of the data redundancy by normalizing tables, the number of tables is very high in operational use. Data retrieval and performance are the reasons for the rather limited normalization in the data warehouse.

  Normalization means that a characteristic for an object within a database is only stored once in a specific table. It is easier to keep the contents of databases with normalized tables consistent as only one entry has to be changed when changes are made.

  Example: All personal data for a customer are stored in a customer table and a customer is clearly identified by a customer number. If other tables, such as invoice tables, refer to the customer, then his customer data is not stored again, only the customer number. If the address data is to be output as well, the customer table also has to be read.

- **Optimized access**

  An OLAP environment is optimized for read access. For operational OLTP applications, functions such as change, insert and delete also have to be implemented in similarly performant ways.

The following graphic provides an overview of the architecture of the SAP BW in a heterogeneous system infrastructure.

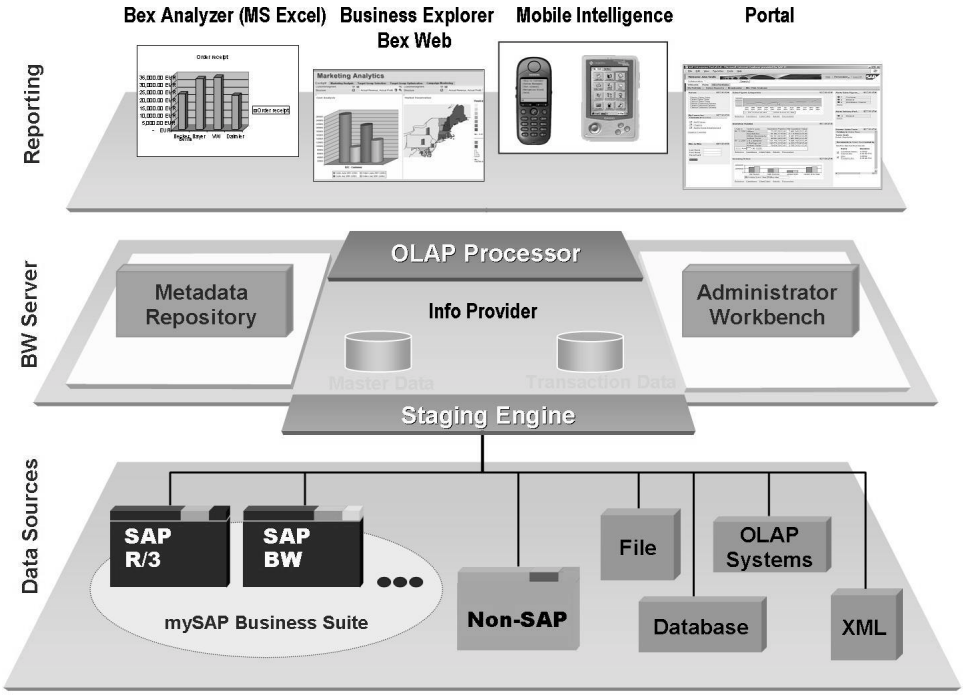

Figure 96: The 3 level architecture of the SAP BW

Discuss the graphic using the following memos.

The lowest level of the slide shows the source systems. SAP provides extraction mechanisms for production data from SAP systems. For non-SAP system, there are the BW BAPI interface, the interfaces for accessing relational database systems and multi-dimensional applications and an interface for processing XML files. You can also load data from flat files.

Metadata and application data (master and transaction data) is managed in the *Business Information Warehouse* server. When a report (**query**) is executed, the **OLAP processor** reads the data.

The third level shows the different reporting tools: *BEx Analyzer*, embedded in *Microsoft Excel*, and *BEx Web Analyzer*, embedded in *BEx Web*. You can use it to execute queries in the browser. In general, data can be transferred to a cell phone and, in particular, it is possible to run and display analyses on portal interfaces.

The structuring and visualization of all information is ensured by the fact that, based on the requirements and expectations on reporting, suitable data is provided in the *SAP BW*. This data can originate from SAP systems as well as non-SAP systems, a large portion of the users of *SAP BW* extracts and analyzes non-SAP data.

There are consulting statements claiming that 80 to 85 percent of the data in customer SAP BWs originates from external systems.

The *Business Information Warehouse* Server administers application and metadata. The OLAP processor analyzes the data when it is accessed via the reporting interface.

The physical model of the SAP BW is based on a schema in which, put simply, the master data (customer numbers, customer names, product names, status, country indicator, base units of measure...) is stored in **master data tables** and the transaction data (sales, stocks, quantities...) is stored in a **fact table** and in **dimension tables**.

In addition to all other dimension tables, there are always two special dimension tables for reference to times and units.

Dimension tables are linked to the central fact table by means of foreign key dependencies. There, the dimension attribute with the finest level of detail of the respective dimension table is a foreign key in the fact table. This way, all data records of the fact table can be identified uniquely. The link between the central fact table and the surrounding dimension tables that is thus created is also referred to a star schema because, in graphical depictions, the dimension tables are usually grouped around the central fact table in a star shape. The SAP BW star schema is an enhancement of this simpler, classical star schema. Part of the enhancement is that the dimension tables do not contain any master data information. This is contained in separate master data tables that are, in turn, linked to the key figures of the fact tables via SID tables (SID = surrogate ID = generated surrogate key). On the one hand, this is done for performance reasons (because the use of generated INT4 keys enables faster data access than access via long string alphanumeric keys). On the other hand, it facilitates the multi-language capability of the system as well as, for example, the historization of dimensions.

In order to get the actual value of a field, the SAP BW uses the keys in the *dimension tables* to read the corresponding master data tables. The system determines the transaction data from the *fact table*.

Both the *fact table* and the *master data tables* are based on **InfoObjects**, more precisely, the required *characteristics* and *key figures* that have to be created in the data warehouse.

Hint: In the BW, business evaluation objects (customers, sales and so on) are called *InfoObjects*. *InfoObjects* are subdivided into characteristics, key figures, units, time characteristics and so on. *InfoObjects* are thus the smallest modules in the SAP BW.

Fact tables are created in the context of defining ***BasisCubes***, a special form of *InfoCube*.

InfoCubes are the central objects, on which reports and analyses are based from a data perspective. SAP knows several forms of InfoCubes:

- **BasisCubes**

  InfoCubes that are physical data stores, that is, tables on the Data Warehouse Server

- **Virtual InfoCubes**

  InfoCubes that do not contain any data but are a layer on a physical dataset

From a reporting perspective, an InfoCube describes a self-contained data set of a business area on which queries can be defined. From a technical perspective the BasisCube consists of a set of relational databases that are composed multidimensionally, that is from a central fact table and several dimension tables that surround it. Apart from InfoCubes there are also other objects that can be used for reporting (such as ODS objects, operational data store objects, which are generally used to store consolidated data at document level). In contrast to the star-shaped arrangement of tables in an InfoCube, ODS objects are implemented in flat tables. All objects that can be used for reporting are called Info Providers.

From a reporting perspective, there is no difference between physical and virtual *InfoCubes*

How does data get into *InfoCubes*?

Physical data stores are supplied with data from the requested source systems by means of *extraction runs*. The following graphic illustrates all relevant terms in the context of extraction. The data is stored in an SAP system. A BW plug-in must be installed so that an SAP system can be used as a source system.

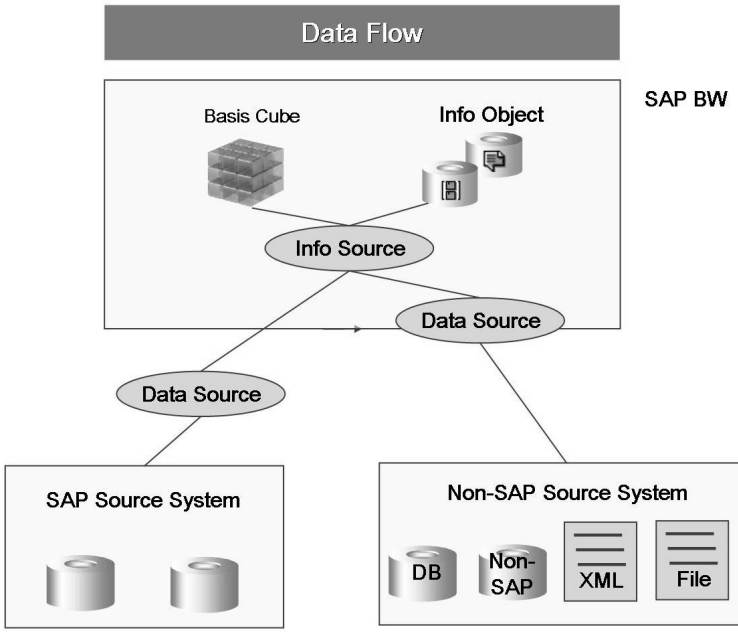

Figure: SAP BW: Data Flow regarding a Basis Cube

Source systems provide data that is replicated into the BW system via *DataSources*. In the BW system, you define data flow from the DataSource to an *InfoSource*, which is the actual source for the *BasisCube* or the *master data tables*.

You can define rules affecting the data transfer both during the data transfer from DataSource to the InfoSource as well as during the data transfer from the InfoSource to the tables. (Definition of conditions that have to be met so that data can be transferred, execution of currency conversions, enhancing master data fields, for example to delimitate different systems after corporate takeovers...)

The SAP BW also provides solutions for the data transfer from non-SAP systems:

- Data transfer with third party tools
- Data transfer with XML or SOAP documents
- Data transfer via DB connect
- Data transfer from flat files
- Data transfer using UD-Connect (as of SAP BW 3.5)
- Data transfer by means of the SAP Exchange Infrastructure (as of SAP BW 3.5)

UD-Connect makes it possible to access data from any relational database. As sources, DB connect only permits those databases that collaborate with SAP systems by default.

For more information on *data flows* and their definition, refer to the various BW courses, such as BW310, BW340, BW350.

Business Content

Business Content (BC) contains preconfigured, role and task related information models that can be adapted to individual enterprise requirements. For example, this could be the area of cost center accounting or travel expense processing. The structure of the Business Content is based on application components and consists of **roles, workbooks, queries, InfoCubes, InfoObjects, InfoSources, update rules** as well as *extractors* for *SAP R/3, SAP New Dimension Applications* and other selected applications.

Queries are defined queries in the reporting of the *SAP BW* (corresponding to ABAP reports and queries in an SAP system) and you can display the result of a query in different *Business Explorer (BEx)*components: as a **workbook** in *BEx Analyzer* (Microsoft Excel-based) or as a **Web application** in *BEx Web* (browser-based).

An **InfoSource** is a set of logically related **InfoObjects** that contains all available information on a business process (such as cost center accounting). You fill the InfoCubes in the *SAP BW* from the InfoSource according to certain **update rules**.

A **DataSource** in the *SAP BW* always describes a business unit of master data (such as material master data) as well as transaction data (such as sales data). From a source system perspective, every DataSource has meta-information, such as the field or field descriptions for the master and transaction data as well as programs that describe how the extraction is carried out. This information is source system specific, that is, a DataSource depends on the source system.

All these objects are summarized in **roles**. A role corresponds to certain functions in the enterprise (purchasing manager, head of finance...) and a need for information resulting therefrom.

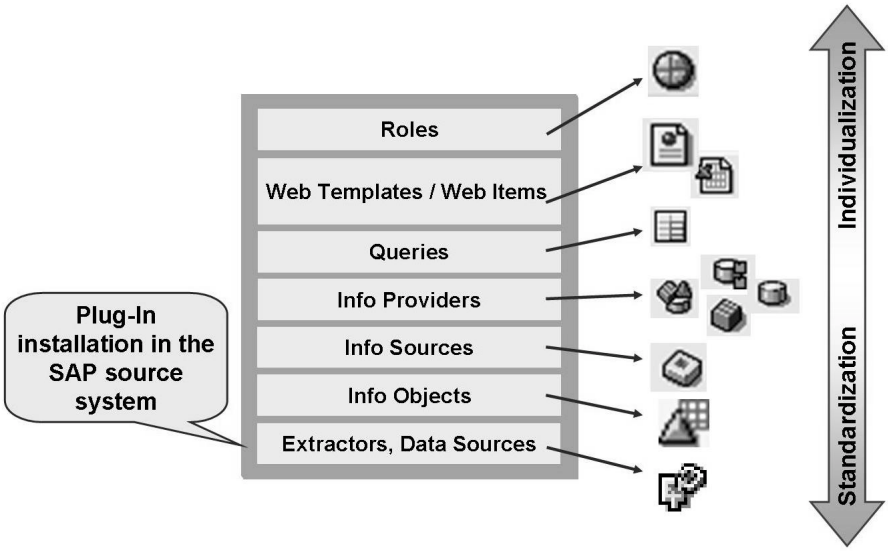

Figure 97: SAP BW: Business Content

The Business Content is shipped as an add-on to the *SAP BW*. The documentation provides detailed information on the business content of the individual applications.

The distinct advantage of the Business Content is that it is based on a data model and you can thus activate all required and related objects at the same time. As the Business Content also contains objects of the *data flow* you load the corresponding data from the SAP source systems very quickly and, if necessary, analyze is using predefined **queries**.

In general, the objects in the Business Content are shipped in an inactive version that has to be explicitly transferred, that is copied and activated.

SAP provides *extractors* for transferring data from an SAP source system to an SAP BW in almost all applications. They are usually preprogrammed and are shipped with the Business Content.

If you are asked what has to be considered if an already applied data model (for example, from the Business Content delivered by SAP) has to be changed retrospectively:

Analyses in the BW are based on InfoCubes. InfoCubes are based on a data model. Even though all care is taken during modeling, customers regularly notice at some stage that the data model is insufficient and that they have to enhance it. Enhancing a data model means that the data has to be loaded again. And then it can happen that the data is no longer available in the OLTP system in the way that it is required. For this, the SAP BW provides the function of the Enterprise Data Warehouse Layer. Customers can use it to store all records that are loaded into

the SAP BW at some stage in non-aggregated form for subsequent loading. But such a procedure produces significant datasets, that is, archiving is an issue for customers using this function.

Information on archiving the SAP BW:

For NetWeaver 04 data is written from the BW to files and has to be reloaded to the SAP BW for analysis. For NetWeaver 05, queries will not only be able to include online data but also data that is no longer stored in the BW.

Knowledge Management

Knowledge Management is to enable users to access all sorts of internal and external contents by using intelligent search functions. Furthermore, there are also publication and registration options available. But users can also just browse, classify or manage contents. The publication cycle of documents can be controlled by means of online communication or workflow functions. Access to information is controlled by means of authorization profiles. *Knowledge Management* also allows experts to be assigned to certain subject areas.

- **Content Management** support for the entire life cycle of documents, from the creation to the archiving of a document
- **Search & Classification** (search engine TREX) full text search across all types of documents and their automatic classification

There are various possible information sources: Desktop documents, websites, spreadsheets, database tables, design drawings from PLM (DVS), powerpoint presentations or text files.

Integrated master data management with SAP MDM

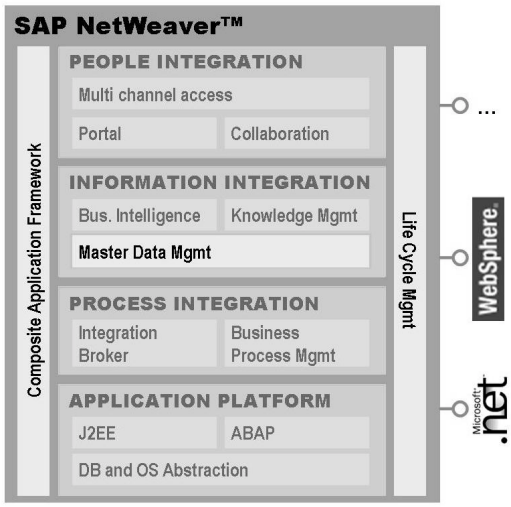

Figure 98: SAP Master Data Management

If an enterprise stores master data in different locations and systems at the same time, then this multiple storage can lead to redundancies and variances, which might significantly disrupt business processes. **SAP Master Data Management (SAP MDM)** enables you to create master data that is uniform across the enterprise and to distribute it to different dependent systems. *SAP MDM* uses the technological basis of the SAP Exchange Infrastructure (*SAP XI*) to distribute the data. It not only distributes the data but for new master data, it always checks whether identical master data objects are already available and generates comprehensive information on duplicates, which can in turn be transferred into the *SAP BW* or used in analyses. This reduces data management costs and significantly simplifies the correction of data errors.

Everyone who knows the central user management can use this as an analogous example from the Basis technology.

Shell is a good real live example for the benefits of MDM. At Shell, there are operational units (similar to SAP business units), which basically have contracts with suppliers like a company within the company. It was only by means of MDM that it was found out that Shell is one of the main customers of Coca Cola brand soft drinks. By means of the Content Integrator it was possible to find out that the different names for Cola (Coke, Cola, Coca Cola, and so on) and the different suppliers (local suppliers of Coca Cola), which appeared in the orders, were actually identical. The data was loaded into the BW and assigned correctly. This revealed the actual purchase quantity. Of course, that gave Shell a lot more leverage for negotiating discounts...

If there are questions regarding the software components: MDM does not only require XI and BW, but also Enterprise Portal as the user interface, even though some settings can still be made in the SAP GUI for Windows.

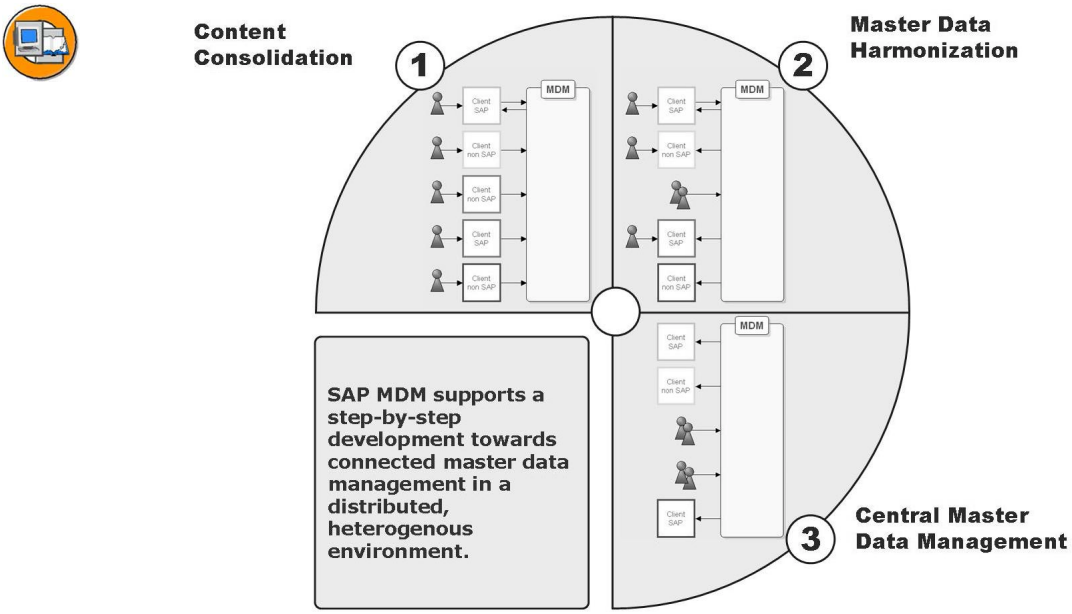

Figure 99: Integrated master data management with SAP MDM

MDM structure

A central MDM system gets master data from local systems. The communication between the systems happens via RFC connections. An SAP BW is required for reporting across centralized master data. Master data has to be matched so that reporting can run properly. For example, identical products have different material key figures in the local systems. The data is analyzed according to the matching rules. Depending on matches in certain data fields, points are assigned and entered in a matching table. Threshold values are then used to distinguish whether or not this refers to the same product. If an identical product is determined that was maintained differently in local systems, a synchronization takes place. If required, this synchronization can also happen manually.

MDM scenarios

With regard to locally maintained data, you have to distinguish between different scenarios. Scenario 1 means the master data is only maintained locally. In this case, only a matching takes place, no synchronization. Scenario 2 means the master data is maintained locally but harmonized/matched via the central system. In scenario 3 the master data is only maintained centrally and distributed to local systems.

Integration Level of Process Integration

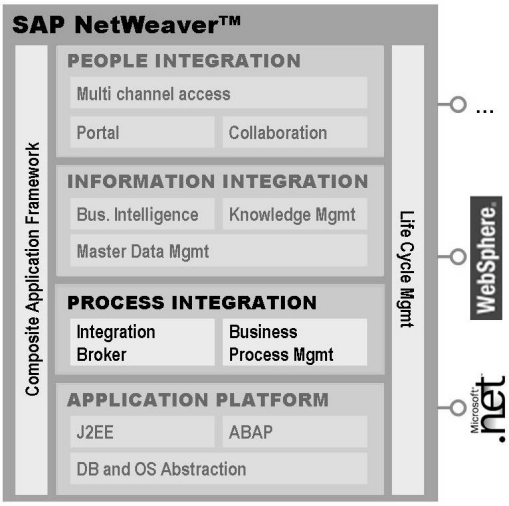

Figure 100: Process Integration

The aim of this lesson is to introduce the fundamental modules of SAP XI: *SLD*, *Integration Repository*, *Integration Directory*, *logical* and *technical routing* and so on.)

Participants should get an idea of the levels involved and understand the difference between **design time** (general definition of the objects) and **configuration time** (combining objects in concrete scenarios with concrete systems).

Caution: Note on the terminology

At the end of June 2004, the following terminology changes were decided:

Business Scenario -> integration scenario

Business process -> integration process

Scenario (in the integration directory) -> configuration scenario, plural: configuration scenarios

The course material already uses the new terminology.

The *SAP Exchange Infrastructure (SAP XI)* can be used to implement cross-system business processes. Within the overall architecture of *SAP NetWeaver*, *SAP XI* takes care of the process integration. In detail, it enables the connection of non-SAP and SAP systems from different suppliers in different versions on different platforms (Java, ABAP and so on). *SAP XI* is based on an open architecture, mainly uses open standards (especially from the realm of XML and Java) and offers services that are essential in a heterogeneous and complex system

infrastructure: runtime infrastructure for exchanging messages, configuration options for controlling business processes and the flow of messages as well as options for transforming message contents between sender and receiver. With *SAP XI 3.0* the scope of functions of *SAP XI* was enhanced mainly in the area of business-to-business communication and cross-system business process management.

In general, the *SAP Exchange Infrastructure* includes the following components:

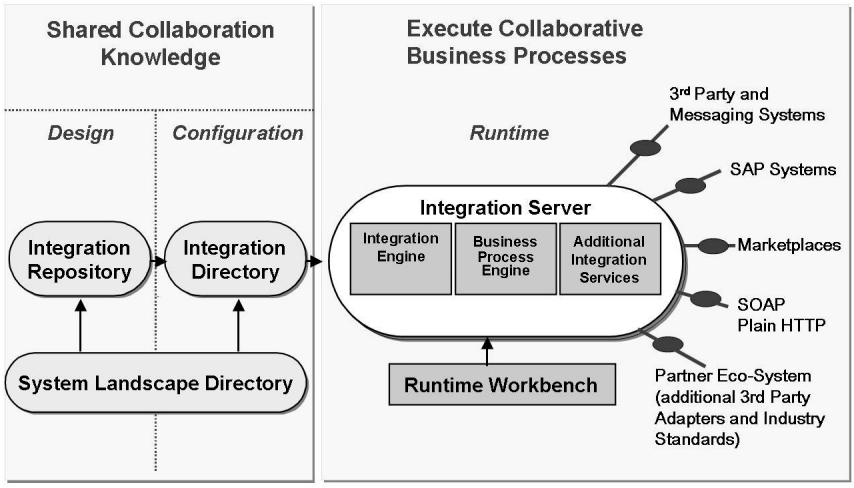

Figure 101: Components of the SAP Exchange Infrastructure

- In the *System Landscape Directory* , system landscapes are mapped, that is, the different systems are defined with which the business processes are handled.
- In the *Integration Repository* a description of all required interfaces is entered in the **design phase**, irrespective of the platform. Based on this description, a **mapping** has to be defined in the *Integration Repository* if the structure and format of the data to be exchanged are not suitable.

  So far, this does not affect the systems and processes.
- In a **configuration phase**, the data types, message types, interfaces and mappings that are defined in the *Integration Repository* are then selected and assigned to each other based on the system infrastructure and business processes in question (*Technical and Logical Routing*).

  The result is stored in the *Integration Directory*.
- The application specific contents are transferred from the sender to the receiver by means of **messages** in a freely-definable **XML schema**. The interface used (IDoc, file, database...) by the data structures determines the structure of the message.

  At runtime, the Integration Engine of the *Integration Server* analyzes the configuration of incoming messages in the *Integration Directory*. Based on the data available there, it determines the recipient(s) of the message, maps the incoming message to the interface structure of the recipient and forwards it. The *Integration Server* is thus the central communication and distribution machine for XML messages.

In the SLD environment there are some important terms that are not only relevant to SAP XI: product, product version, software component and software component version. In the SAP environment, a ***product*** in SLD corresponds to a technical SAP component. It consists of one or more ***software components*** and the customer can see, install and update it. A *product* is shipped in a **product version**, which in turn contains ***software component versions***. For example, the product SAP APO with version 3.1 includes the software components of the SAP Basis in the software components version 4.6D, SAPBW in version 2.1, SAP ABA in version 4.6D, SAP APO in version 3.1, and so on.

All operational systems of the system infrastructure are recorded in SLD. Apart from the respective host name and other technical attributes this also includes the information on which software components are installed on a server. Each *operational system* is assigned one or more *business system* names. This *business system* name is required for configuring the scenarios for sending and receiving systems (basically the “postal address” for mail). For SAP systems as the operational system, there can be one *business system* for each client. In general, the *business system* name for SAP systems is identical to the term *logical system* that you know from the ALE world.

As of SAP XI 3.0 it will be possible to already register a system in the System Landscape Directory when it is set up, that is, manual maintenance tasks are no longer necessary.

For a developer who processes objects in the *Integration Repository*, the *software component version* is decisive as the development happens in the context of the *software component version*. At the beginning of the development, a *namespace* for the objects, that belong together semantically (comparable to an ABAP or Java package) is set up in the *Integration Repository*. Objects are thus clearly identified by their names and *namespace*.

At first, a *namespace* is a freely selectable constant. *Namespaces* must be globally unique, that is, across software component versions. One option is to use the company name. When you create a *namespace*, parts of the name must never start with a number. You cannot change namespaces retrospectively without restrictions.

Connection of different systems to the SAP Exchange Infrastructure

If an involved system is an external system or if existing interfaces from SAP systems with Basis Release <= 4.6 are to be used in the context of *SAP XI* they are connected to the Integration Server via ***adaptors***. SAP provides a number of adaptors for this purpose. (IDoc, File, RFC, Database Adaptor, Mail, SOAP, RosettaNet Adaptor, ...)

SAP XI not only provides adaptors but also provides an adaptor framework with which customers can build user-defined adaptors. Besides that, 3rd party adaptors are part of the delivery. This includes, in particular, the Seeburger AG with specific EDI adaptors.

If customers would like to receive more information on EDI adaptors, they can contact sindhu.gangadharan@sap.com.

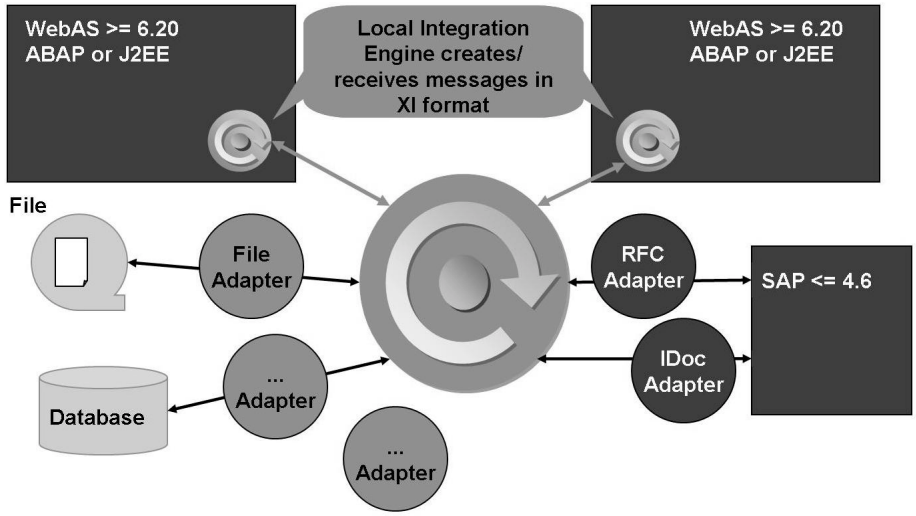

Figure 102: Connecting Different Systems to SAP XI

The sending system makes the data available in document format, for example, IDoc, and sends it to the adaptor via a protocol. The adaptor transforms the document to *SAP XI* format and sends it to the *Integration Server* using HTTP(S). In the configuration you specify which adaptor the receiver is to use to receive the message. The *Integration Server* sends the message to the corresponding adaptor, which converts the message to the receiver's protocol and sends the message to the receiver.

If a cross-system business process it to be implemented by means of *SAP XI* and the involved systems include *SAP Web AS* >= 6.20, then *SAP XI* provides a new programming model. This can be used to define the interface descriptions in the *Integration Repository* irrespective of the platform. These descriptions can be used to generate *proxies* in the different application systems for JAVA applications or SAP systems. In the application system *proxies* represent the interface from the *Integration Repository*.

From a technical perspective, proxy objects are classes and methods in a programming language (ABAP or Java) that can generate and process messages for a file format defined in the *Integration Repository*.

SAP provides executable proxies for all cross-platform *message interfaces*. However, customers must first generate these proxies before they can use them for their own interfaces.

An application of a business system calls a proxy to send a message. The generated proxy generates a message in SAP XI format from the interface parameters and the *proxy runtime* sends it to the *Integration Server*, which processes it further and then forwards it. In order to process messages in a receiving business system, the *proxy runtime* on the receiving system accepts the message and calls the ABAP class that implements the corresponding proxy.

The form of processing presumes a local SAP Integration Engine on the connected business systems, the implementation of which is part of the configuration of scenarios with SAP Exchange Infrastructure.

A new programming model is applied here: the central definition of interfaces, based on *message interfaces*.

You know the basic idea of creating an RFC-enable function module with its interface and the method source code for the planned functions in system A, so that this module can then be called externally. In the same example, if you use the SAP Exchange Infrastructure, you define the message format centrally in the *Integration Repository*. That means, you basically define the interface that is used. In system A you then only have to generate the method source code for processing the message data. In contrast, if you use adaptors the interfaces are still developed decentrally in the business system and the interfaces are then transported into the *Integration Repository*.

There are no current plans to make it possible to program proxies in languages other than ABAP and Java.

For more information, please refer to courses BIT400 and BIT450.

How the SAP Exchange Infrastructure works - the SAP XI Pipeline

In general, the SAP XI runtime consists of the following components:

- Integration Engine

  Executes the integration logic of the *Integration Server*. The purpose of the *Integration-Engine* is to correctly and consistently process the *Pipelines*, that is, to receive, process and forward XML messages.

- Integration Server

  Is basically the software component SAP Web AS with XI specific enhancements. As soon as the *Integration Server* receives a message in SAP XI format, the processing takes place in the pipeline of the *Integration Server* by means of the *Integration Engine*.

- Proxy runtime

  Is required to use proxies to exchange messages with the *Integration Server*. Proxies are based on a specific programming model.

- Adaptor engine

  Is required to use the RFC adaptor and external adaptors to exchange messages with the *Integration Server*.

- The actual adaptors

  This is about those adaptors that are connected to the adaptor engine or are operated independently of it, such as the IDoc adaptor or the plain HTTP adaptor.

- Monitoring

  Central monitoring supports the monitoring of the SAP XI runtime components, the message flow and the performance.

Note: The *pipeline* refers to all the steps that have to be executed while an XML message is processed, that is a defined sequence of services that a message has to go through. The run through the *pipeline* always happens in the same form as as soon as a message reaches the *Integration Server*.

First, the receiver of the message and its input interface are determined by means of the configuration data of receiver and interface determination. This step is called ***logical routing***.

Following that, the pipeline determines how the receiver can be physically reached using ***technical routing***. If required by the configuration (as in: the structure of the message to be transferred differs between the sender and the receiver), then the assigned **mapping program** is executed that maps the output interface to the input interface. At the end, the *Integration Server* sends the message to the target system.

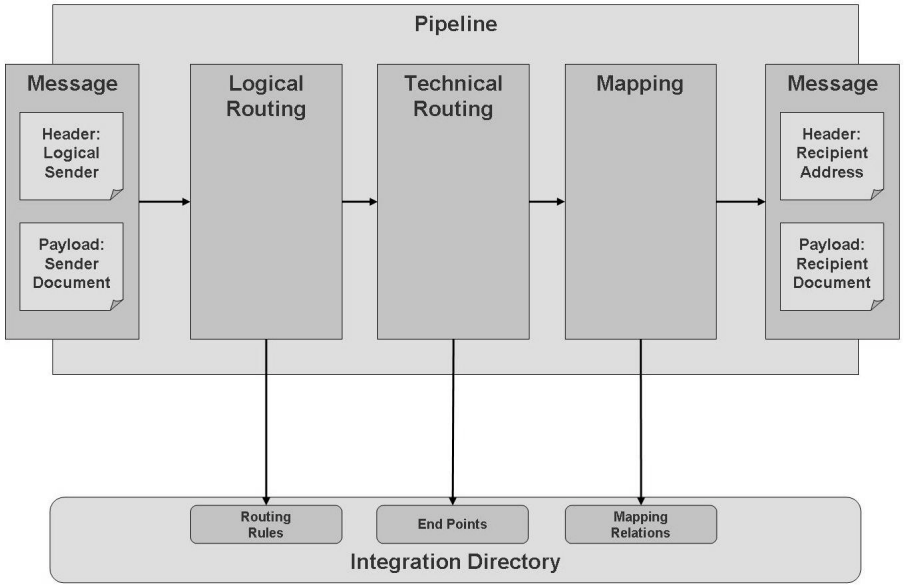

Figure: Pipeline of the Integration Server

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:

- Name the integration aspects of *SAP NetWeaver*
- Explain the basic concepts of the individual integration levels of *SAP NetWeaver*

Related Information

- Refer to http://www.sap.com/netweaver for more information.

Lesson: From SAP R/3 to mySAP ERP and the Enterprise Services Architecture

Lesson Duration: 30 Minutes

Lesson Overview

This lesson aims to highlight the main differences between the software solutions *SAP R/3* and *mySAP ERP* as well as the significance of the Enterprise Services Architecture (ESA).

Lesson Objectives

After completing this lesson, you will be able to:

- Explain the evolution from *SAP R/3* to *mySAP ERP* and the Enterprise Services Architecture.
- Describe the significance of the Web Services within the Enterprise Services Architecture

The instructor should be familiar with all links (and their content) on the Service Marketplace mentioned in this lesson.

Business Example

You want to understand the development of *SAP R/3* after *mySAP ERP* and the Enterprise Services Architecture, in order to be able to fully use the advantages that these offer to your company.

Enterprise Resource Planning to Date

With ERP software, that is, software for controlling and processing business-related company processes, SAP has set standards worldwide. The central idea behind the software, namely the **real-time processing** of different business processes in a company and their implementation in the successful solutions *SAP R/2* and *SAP R/3* has made SAP into a company that is active worldwide. *SAP R/3* was complemented by other solutions from SAP, for example, by *mySAP CRM, mySAP SCM* and *mySAP SRM*.

In the past years, SAP has strongly increased the value of its own application platform and has consolidated this platform with the *SAP NetWeaver*, combining herein the technical foundations for all SAP solutions.

With the *SAP NetWeaver*, all future SAP solutions (e.g. also the *mySAP ERP 2004* solution) are provided with the basic functions, including those from other components, for example, functions of the *SAP Business Information Warehouse*.

The development of *SAP R/2* after *SAP R/3* and other solutions from the past years were characterized by different factors.

Primary differences, but also common features of SAP R/2 and SAP R/3

- In common: real-time processing of business processes
- In common: use of ABAP as a programming language, optimized for implementation in business software
- In common: constantly increasing features (during the maintenance period) by functions newly created by SAP
- In common: adjustability of mapped processes to company-specific activities
- In common: all information in a central database
- **Difference**: host-based system on client-server-based system

Thus, the change from *SAP R/2* to *SAP R/3* (and other "new" software from SAP) primarily meant a change in the technical infrastructure and in the design of the user interface. However, applications were mainly developed "in the same way". What now?

Enterprise Resource Planning and the Enterprise Services Architecture in the Future

The following should provide you with an overview of the differences between *SAP R/3* und *mySAP ERP (2004)* with the central components *ECC 5.0 (ERP Central Component)*.

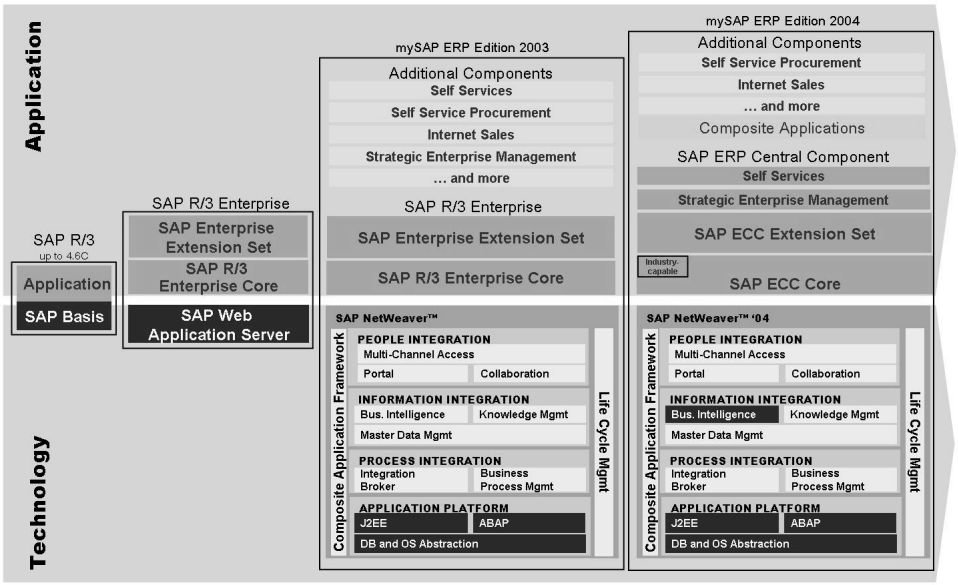

Figure 103: Functions of mySAP ERP 2004

With *mySAP ERP 2004*, SAP is taking the first steps towards business applications, which build on an Enterprise Services Architecture.

The tasks, which *SAP R/3* systems and other SAP software fulfil in companies worldwide, naturally also have to be fulfilled in the future. That means that the Enterprise Services Architecture will not make any basic changes to these processes. What are the core characteristics of an Enterprise Services Architecture?

In the usual **Client Server Architecture**, the business process data are in the system database, the application processes run on application servers and are made available via predefined interfaces. Business processes that do not belong to the classic SAP world can be integrated more easily via interfaces. The processes are very often integrated via “human integrators”. To do this, the employees of the company must know when they have to call up what systems for data maintenance in the company's business processes.

In the **Enterprise Services Architecture** , by comparison, there are role-based user interfaces, which act as central entry points for employees of the company to carry out their work using different applications in different systems. New process steps that are provided as **Enterprise Services** and whose data can be saved in totally different databases, can be integrated using general standards into the process world of the company with minimal effort. With the help of cross-system process definitions and process control through workflow, the amount of work for “human integrators” can be reduced.

The central idea of ESA is that data and application functions can be fused to reusable Enterprise Services. ESA's role model here is the lean manufacturing in the automobile industry. There, lean manufacturing means that subsystems of automobiles (brakes, drive axle, motor, steering mechanism) are standardized in such a way that they can be used and put together by different manufacturers. This means that the components of a car no longer have to originate exclusively from the respective manufacturer. Lean Manufacturing not only applies to the automobile industry, but it is very advanced there.

From an IT point of view, ESA should place a company in a similar position as an automobile manufacturer. The implemented complexity of applications corresponds to the thousand components in traditional automobile production, while the components of an ESA platform are the counterpart to the standardized components in automobile manufacture.

The IT industry is only in its beginnings here. Components, which deserve this name rudimentarily (as not completely standardized) exist mainly for basic technologies, for example, the relational databases or technologies for web servers and browsers. As an example of the only rudimentary “componentization”, you can mention that databases, despite all common factors, e.g. in SQL dialect or in the way in which procedures are composed, can still be very different.

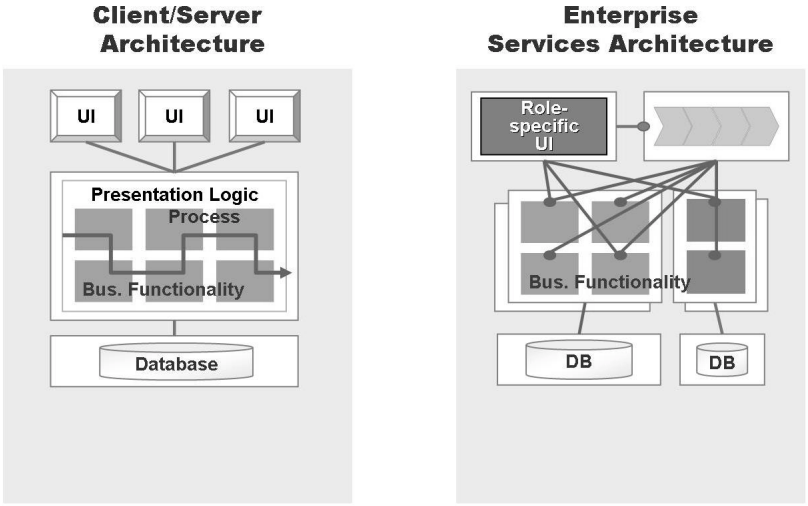

Figure 104: Client Server Architecture versus Enterprise Services Architecture

The individual steps within an Enterprise Service can be processed using Web Services. What is a **Web Service** as opposed to an **Enterprise Service**?

Enterprise Services describe the larger business logic. An Enterprise Service does not address detail functions, but a complete, industry-specific process that can consist of many small individual steps. All actions together form the

Enterprise Service, which thus provides a context-oriented business process logic. **Web Services** , by comparison, are small modular applications, which run within the framework of Internet technologies and which are generally called up as a detail function within applications or Enterprise Services. Standards were agreed to describe the call-up of Web Services. (Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP), Universal Description, Discovery, and Integration (UDDI)).

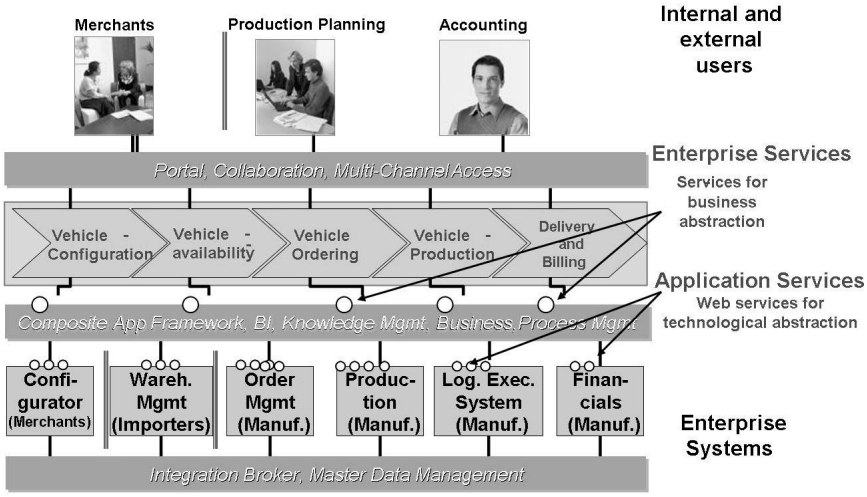

Figure 105: Enterprise Services Architecture and Web Services

Hint: You can find more information on Web Services at http://www.w3c.org/2002/ws.

ESA can be charactarized using the following key words:

Characteristics of the Enterprise Services Architecture (ESA)

- An ESA application is generally implemented across systems.
- An ESA application is created in ABAP or in Java.
- An ESA application generally has no “own” database.
- new functions are entered “outside” of existing systems (for ESA applications).

The availability of Enterprise Services provides many new options.

Enterprise Services enable ...

- ... the efficient creation of new applications without having to modify the underlying system.
- ... very high flexibility in the configuration of business processes, also “in operation”.
- ... the simplified creation of applications that use the functions of several systems.

The following overview shows the SAP planning with regard to other activities for Enterprise Services Architecture until 2007.

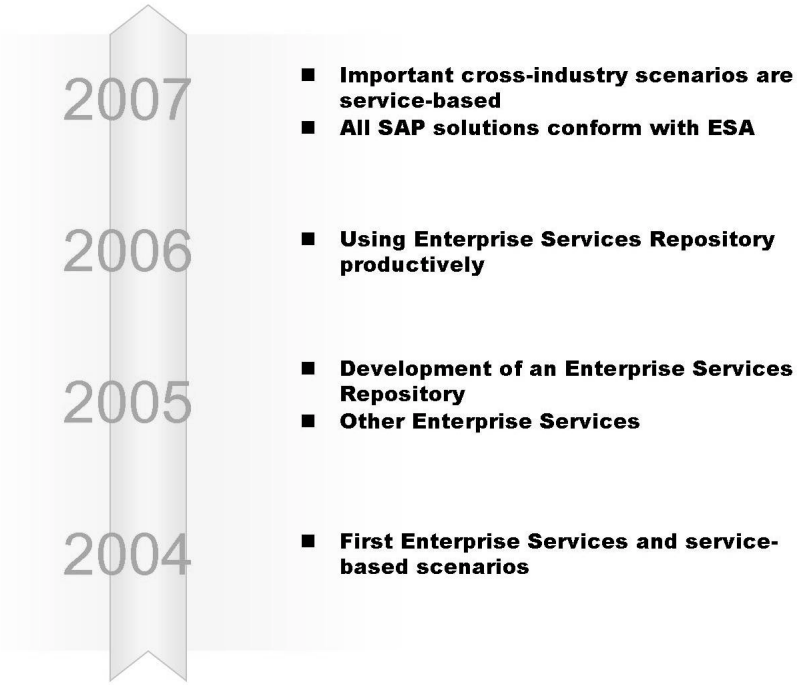

Figure 106: Enterprise Services Architecture until 2007

You can see that SAP will offer the functions of software products on a huge scale as Enterprise Services and will base other, new applications on them. At the same time, you will have the opportunity here to design and use new and flexible business processes across systems without having to intervene in your business systems. Of course, the Enterprise Services Architecture also provides you with entirely new possibilities, such as to link functions of SAP systems with the functions of Enterprise or Web Services of other providers via the Internet.

Facilitated Discussion

No facilitated discussion

Discussion Questions

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

No questions

Lesson Summary

You should now be able to:

- Explain the evolution from *SAP R/3* to *mySAP ERP* and the Enterprise Services Architecture.
- Describe the significance of the Web Services within the Enterprise Services Architecture

Related Information

- http://service.sap.com/erp
- http://service.sap.com/esa

Unit Summary

You should now be able to:

- Name the integration aspects of *SAP NetWeaver*
- Explain the basic concepts of the individual integration levels of *SAP NetWeaver*
- Explain the evolution from *SAP R/3* to *mySAP ERP* and the Enterprise Services Architecture.
- Describe the significance of the Web Services within the Enterprise Services Architecture

Test Your Knowledge

1. What is the follow-up product of *SAP R/3*?

Answers

1. What is the follow-up product of *SAP R/3*?

   Answer: The follow-up product of *SAP R/3* is *mySAP ERP*.

Course Summary

You should now be able to:

- List the solutions offered by SAP
- Navigate in SAP systems
- 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 interpreter
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.

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.

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.

Buffer
Area in the main memory of an instance that can be used for temporary storage of data frequently used by the applications.

Business Add-In
The location in a program defined by the developer at which software recipient layers (industries, partners, customers and so on) can insert coding 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 Management Service (CMS)
Part of the NetWeaver Development Infrastructure, which is responsible for the management of the software lifecycle. It includes the configuration of development landscapes, the development infrastructure and the transport of software changes.

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.
Software component that uses the services provided by a server (software-oriented view) or 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.

Component Build Service (CBS)
The *Component Build Service (CBS)* is part of the SAP NetWeaver Java Development Infrastructure and is used for the central Build of the source texts.

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.

Design Time Repository (DTR)
Central memory for all kinds of source files. The memory is presented logically as a hierarchical file and folder structure. Physically, the contents are stored in a database and are accessible via the open logs WebDAV and DeltaV.

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 the 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.

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

field group
A field group comprises a set of related screen elements, for example, all checkboxes in one selection.

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.

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).

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.

Java Virtual Machine
Interpreter for monitored execution of Java byte code on the respective hardware platform.

keyword
The first word in an ABAP statement. The keyword determines the meaning of the entire statement.

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.

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.

Name Server
part of the SAP System Landscape Directory (SLD). The Name Server enables the reservation of globally unique names for Java development objects.

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.

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.

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 "/" (for example: 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 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, which the user can use to exchange information with the computer. You can use the user interface to choose commands, start programs, display files and execute other options by pressing function keys or 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 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 the information in 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
An open integration and application platform for all SAP solutions and certain solutions from SAP partners. SAP NetWeaver is a web-based platform, which acts as the basis for the Enterprise Services Architecture (ESA) and which enables a cross-company integration and reconciliation of employees, information und business processes that does not depend on technology. Thanks to open standards, information and applications can be integrated, which originate from practically every source and which can be based on practically every technology. SAP NetWeaver is interoperable with most important technological standards, such as Java 2 Platform, Enterprise Edition (J2EE) und Microsoft .NET. . The platform provides maximum reliability, security and scalability, so that business processes that are decisive for success run without any problems. SAP NetWeaver contains functions for business intelligence, company portals, exchange infrastructures, master data management, mobile infrastructures and a Web Application Server.

SAP NetWeaver Development Infrastructure (JDI)
The SAP NetWeaver Development Infrastructure (JDI) extends the concept of an Integrated Development Environment (IDE, SAP NetWeaver Developer Studio) by server-based services, which provides development teams centrally with a consistent development environment and supports the software development during the entire lifecycle of a product.

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 Reference IMG
A complete guideline for introducing SAP systems, which contains all the necessary Customizing activities, sorted according to application components.

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 can be, for example, an *SAP R/3 Enterprise*, an *SAP BW*- or an *SAP CRM* system. SAP systems are the central components of SAP solutions within the framework of the *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 usually used synonymously.

SAP Web AS Java
Since SAP Web AS 6.20, SAP ships a J2EE application server called SAP Web AS Java. On this application server you can execute programs written in Java (more precisely: according to the J2EE standard).

screen
Also called “dynpro”, from DYNamic PROgram. A *screen* consists of a screen image and its underlying flow logic.

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*.

screen processor
Component that executes the *screen* flow logic of application programs.

Server
Software component that provides a service (software-oriented view) or 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

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.

The SAP NetWeaver Developer Studio

SAP's own development environment for the development of Java-based multi-level business applications. The new development environment builds on the open source product *Eclipse*.

the SAP System Landscape Directory (SLD)

SAP J2EE application, which contains all relevant information about software products and components that can be installed on a system landscape. Furthermore, it contains a description of the current system landscape.

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 up a transaction, enter the transaction code into the command field and 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 and an import process: The export process reads the objects and stores them in the 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 protocol of all actions during export and import.

User Master Record

The user master record contains the definition of a particular user in the client. Some of the fields are name, first name, initial password and phone number. The user master record is used to create the user context (see: user context) when a user logs on to the system.

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 following work process types exist: 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.

Index

T

U

V

W

X

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.