ADM102

SAP Web AS Administration II

Date _____

Training Center _____

Instructors _____

_____

Education Website _____

Instructor Handbook

Course Version: 2003 Q3
Course Duration: 3 Day(s)
Material Number: 50062291
Owner: Thomas Menzel (D033108)

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

Copyright

Copyright © 2003 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

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

Trademarks

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

Disclaimer

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

About this Handbook

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

Typographic Conventions

The following typographic conventions are used in this guide.

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.

© 2003 SAP AG. All rights reserved.

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.

© 2003 SAP AG. All rights reserved.

Contents

Course Overview vii

Course Goals vii
Course Objectives viii

Unit 1: Technology Components in the Internet Environment .. 1

Internet Scenarios with SAP Systems 4
SAP Internet Transaction Server 11
Internet Communication Manager 29
Internet Communication Framework 45
SAP J2EE Engine: Administration 57
SAP Web Dispatcher 76

Unit 2: Globalization 99

Aspects of Globalization 101
Multiple Languages in an SAP System 117
Code Pages and Unicode: Fundamentals 139

Unit 3: Technical Aspects of External Communication 163

Communication with SAPconnect 165
Communication Using SMTP 175
Telephone Integration with SAPphone 205

Unit 4: extended Computer Aided Test Tool (eCATT) 221

Fundamentals of eCATT 223
Configuring eCATT 233
Executing eCATT Test Scripts 251

Unit 5: Advanced User Administration Topics 273

Introduction to CUA 275
Setting Up a CUA 282
User Administration with the CUA 303
Introduction to Directory Services 314
Technical Connection of Directory Services 326
Data Exchange with Directory Services 351

Unit 6: Outlook: SAP Exchange Infrastructure (SAP XI) 381

SAP XI: Introduction 382
SAP XI: Structure and Administration 389

© 2003 SAP AG. All rights reserved.

Glossary **409**

Index **417**

© 2003 SAP AG. All rights reserved.

Course Overview

This course provides an insight into different areas of the administration of SAP systems based on an SAP Web Application Server. While the ADM100 training course deals with the daily tasks of a system administrator, this training course examines more advanced topics.

Target Audience

This course is intended for the following audiences:

- SAP system administrators
- Technology consultants

Course Prerequisites

Required Knowledge

- Completion of SAPTEC or comparable level of knowledge
- Completion of ADM100 or comparable level of knowledge

Recommended Knowledge

- Experience administering SAP systems

Course Duration Details

Unit 1:
Technology Components in the Internet Environment

Internet Scenarios with SAP Systems	30 minutes
SAP Internet Transaction Server	35 minutes
Internet Communication Manager	45 minutes
Internet Communication Framework	35 minutes
SAP J2EE Engine: Administration	60 minutes
SAP Web Dispatcher	40 minutes

Unit 2: Globalization

Aspects of Globalization	40 minutes
Multiple Languages in an SAP System	70 minutes
Code Pages and Unicode: Fundamentals	40 minutes

Unit 3: Technical Aspects of External Communication

Communication with SAPconnect	35 minutes
Communication Using SMTP	60 minutes
Telephone Integration with SAPphone	30 minutes

Unit 4: extended Computer Aided Test Tool (eCATT)

© 2003 SAP AG. All rights reserved.

Fundamentals of eCATT	20 minutes
Configuring eCATT	40 minutes
Executing eCATT Test Scripts	30 minutes
Unit 5: Advanced User Administration Topics	
Introduction to CUA	25 minutes
Setting Up a CUA	85 minutes
User Administration with the CUA	75 minutes
Introduction to Directory Services	20 minutes
Technical Connection of Directory Services	75 minutes
Data Exchange with Directory Services	60 minutes
Unit 6: Outlook: SAP Exchange Infrastructure (SAP XI)	
SAP XI: Introduction	15 minutes
SAP XI: Structure and Administration	20 minutes

Course Goals

This course will prepare you to:

- Perform administrative tasks that affect the SAP Web AS technical infrastructure and configuration

Course Objectives

After completing this course, you will be able to:

- List and evaluate technology components of SAP systems in the Internet environment
- List the requirements for an SAP system used worldwide
- Explain the procedures for telephone integration and for sending mail from SAP systems
- Describe the eCATT procedure for automated tests
- Set up Central User Administration
- Connect an SAP Web Application Server to a directory service
- Explain the purpose of the SAP Exchange Infrastructure

SAP Software Component Information

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

Caution: As this instructor guide can only be updated at release changes, a separate area has been set up for **Instructor Guides** on the http://intranet.sap.com/epf-adm page. The current trainer

© 2003 SAP AG. All rights reserved.

notes are collected in a document there. This training material - and the URLs specified - correspond to the status in April 2003. For current information, see the extended instructor guide at the above page.

You must be well prepared to teach this challenging course. The challenge arises, among other things, from the fact that many participants will probably have many years' experience of administering SAP systems over various releases and will therefore be familiar with at least the fundamentals of some of the areas to be discussed. These participants may be interested in what has changed over the various releases in the relevant areas dealt with in the course.

The course also places demands on the up-to-dateness of the instructor's knowledge. In this way, it could become necessary over time to know that certain terms are no longer to be used, although they still appear in the course material (such as the replacement of mySAP Technology with SAP NetWeaver or of mySAP.com with mySAP Business Suite). In this situation, the instructor should provide the participants with the required delta knowledge, as appropriate, and should be prepared for participants' questions.

Attending the ADM100 course is useful as part of your preparation for teaching this course. You should be familiar with all of the topics dealt with in that course. You should also attend the ADM102 course 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-XI, BC-SEC-USR, BC-SEC-DIR, BC-MAS, BC-TWB-TST-ECA, and CA-ARC areas, and the areas about Java technology and the SAP J2EE Engine. The Quick Link */netweaver* in the intranet also serves as a starting point for finding additional information.

Planned **Schedule**:

- As the first exercise for the "Globalization" unit is to run overnight (due to the long runtime), it is important that you both complete the "Technology Components in the Internet Environment" unit and begin the "Globalization" unit on the first day of the course.
- On the second day, you should deal with the other lessons of the Globalization unit and units 3 and 4. If appropriate, you can even go through the first lesson in unit 5.
- The other central user administration topics and connection to directory services form the content of the third day of the course, which finishes with a brief look at SAP XI.

Finally, some notes on the training system:

 © 2003 SAP AG. All rights reserved.

- The course is taught on a copy of system TFC that is recreated every week. You may find artifact entries from the master system in various places in the training system.
- There are two SAP systems with the names DEV and QAS on each server (such as twdf0500). For instructor demonstrations that run on both DEV and QAS, you can display the different systems using different colors in SAP GUI (by choosing *Customizing of Local Layout → New Visual Design → Color Settings*).
- The course is taught principally in client 100. There is a reference user with the name ADM102 and the password 102ADM. Client 200 of the partner system (QAS for the participants that usually work on the DEV system and DEV for the participants that usually work on the QAS system) is also used for setting up the central user administration. The course participants use the existing ADM102-## users (where ## is their group number) with the initial password "init". These users are copies of the ADM102 reference user and provide all relevant authorizations for the course.
- There is **no** generally available preparation system for this course (this would not make sense as, for example, additional software is used that can only be installed once for each server). As an instructor, you can reserve a system by specifying a cost center in SAPNet under the alias */ism* or by sending an e-mail to *SAP Server Request*.

Caution: Accessing training servers outside the allocated times is not permitted.

© 2003 SAP AG. All rights reserved.

Unit 1

1

Technology Components in the Internet Environment

If you have already held the ADM100 course in Collection 22, the SAP ITS and ICM lessons will be familiar to you (and perhaps also to participants who have attended that course). In Collection 33, these two topics are no longer contained in ADM100, meaning that the curriculum is coherent.

However, this unit contains far more than the "old" ADM100: on one hand, small extensions have been made to the SAP ITS and ICM lessons, and on the other, lessons on the Internet Communication Framework (ICF), SAP J2EE Engine, and SAP Web Dispatcher have been added. We recommend that you prepare thoroughly (attend ADM102 and work through all demos and exercises).

This unit introduces various technology components, which you must be able to classify and differentiate from each other. This is at least as important as familiarization with the tools and transactions – otherwise the participants will lose the thread, with nothing but clicks and functions. In addition, the focus of this unit is definitely on administration (and not development).

Unit Overview

In this unit, you learn about a number of central technology components that are important if SAP systems are used for intranet or Internet applications. In ADM102, the focus is on administration of the components that are introduced, not on development.

- The SAP Internet Transaction Server (ITS) is used for a range of Internet Application Components (IACs) and with the SAP GUI for HTML.
- The Internet Communication Manager (ICM) is the process that turns the classic SAP application server into an SAP Web Application Server.

© 2003 SAP AG. All rights reserved.

- The Internet Communication Framework (ICF) provides an environment for handling HTTP requests in the work process using Web applications such as BSPs.
- With the SAP J2EE Engine, SAP Web AS provides a complete runtime environment for J2EE applications, in addition to ABAP.
- SAP Web dispatcher distributes HTTP requests to suitable application servers.

Unit Objectives

After completing this unit, you will be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the implementation areas of SAP ITS, ICM, and the SAP J2EE Engine
- Describe the architecture of the SAP ITS
- Start and stop the SAP ITS
- Perform simple administrative tasks on an SAP ITS
- Describe the implementation area of the ICM
- Configure and monitor the ICM
- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service
- Outline the structure of the SAP J2EE Engine
- Perform simple administrative activities for the SAP J2EE Engine
- Outline the function of the SAP Web dispatcher
- Explain how you can use the SAP Web dispatcher to distribute workload across the different instances of an SAP system

Unit Contents

Lesson: Internet Scenarios with SAP Systems 4
Lesson: SAP Internet Transaction Server 11
Exercise 1: Calling and Administering the SAP ITS 19
Lesson: Internet Communication Manager 29
Exercise 2: Administration of the ICM 37
Lesson: Internet Communication Framework 45
Exercise 3: Administrative Work with the ICF 51
Lesson: SAP J2EE Engine: Administration 57
Exercise 4: SAP J2EE Engine: Administration 71
Lesson: SAP Web Dispatcher 76

© 2003 SAP AG. All rights reserved.

Exercise 5: Administration of the SAP Web Dispatcher 85

© 2003 SAP AG. All rights reserved.

Lesson: Internet Scenarios with SAP Systems

3 Lesson Duration: 30 minutes

Lesson Overview

SAP provides a number of ways in which applications can be created for intranet or Internet users. This lesson introduces the technologies on which these applications are based and explains the differences between them.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the implementation areas of SAP ITS, ICM, and the SAP J2EE Engine

---

The SAP ITS and ICM components should be familiar in principle from the predecessor course SAPTEC. In ADM102, the focus is clearly on the **administration** of these components and the SAP J2EE Engine. This lesson deals with positioning the components and explaining what distinguishes them from each other. You should therefore not yet go into detail about the individual components.

- "Thanks to SAP Web AS, I no longer need SAP ITS as of 6.10"
- "SAP Web AS makes the Business Connector redundant."

Incorrect statements such as these unfortunately survive obstinately – both among SAP customers and employees. Therefore, a clear positioning is required at the start of the course.

Short summary: SAP ITS is required for existing "old" Web applications and the SAP GUI for HTML. Upgrading, for example, from SAP R/3 4.6C to SAP R/3 Enterprise does not automatically change the programming model of existing applications. The ICM is the new central process from SAP Web AS 6.10. New applications can be created on this technological foundation. This can be done by the customer – at SAP, this technology is used, for example, to a great extent for SAP BW 3.0. As of SAP Web AS 6.20, in addition to the ABAP runtime environment, SAP provides a Java runtime environment that meets the J2EE standard. For example, SAP CRM and the SAP Enterprise Portal are based on this foundation.

Some customers are unsure which approach is the correct one for them. It is difficult to answer this question across the board. The following aspects are certainly important:

© 2003 SAP AG. All rights reserved.

- Do I want to make changes or extensions to the standard SAP system? In this case, the technology is specified by SAP (however, it can change, depending on the release).
- What skills do the developers involved have? A team of long-term ABAP developers will probably prefer BSPs, while young university graduates may already have good Java knowledge and want to implement this.

Business Example

Your company wants to allow its customers browser-based access to data in the SAP system (for example, in the context of Web-based purchasing). As a member of the system administration team, it is your task to compare and evaluate different methods of realizing this.

SAP Internet Transaction Server (SAP ITS)

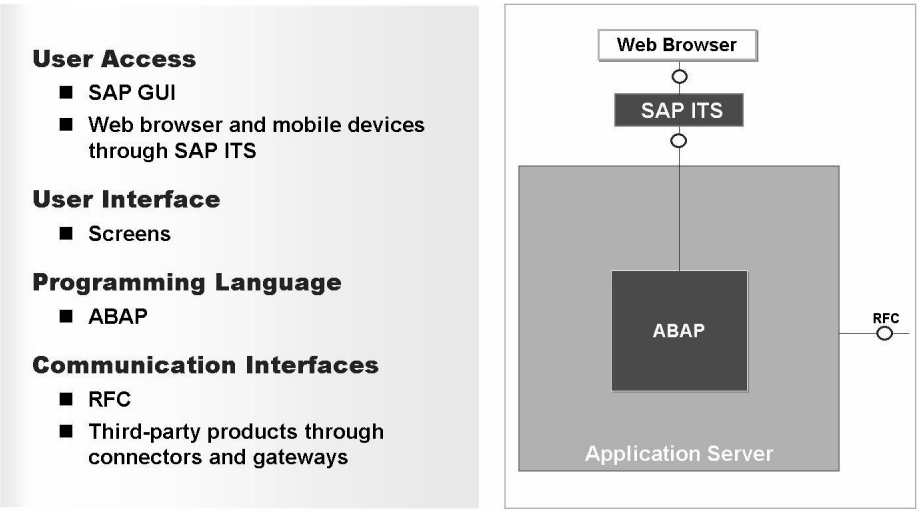


Figure 1: Web Enabling Using the SAP ITS

SAP delivered the first version of the SAP Internet Transaction Server (SAP ITS) with SAP R/3 3.1G in 1996. It is a software component that acts as a gateway between a Web server and an SAP system. SAP ITS switches between Internet protocols and formats (such as HTTP, HTTPS, and HTML) and those of the SAP system (such as DIAG, RFC, and screens).

The SAP ITS is constantly being developed. As it is backward compatible, you should always use the highest possible version released by SAP (for example, you can use SAP ITS 6.20 with an SAP R/3 4.6C installation).

© 2003 SAP AG. All rights reserved.

Web applications that were developed specifically for SAP ITS are called Internet Application Components (IACs). These include, for example, Employee Self-Service (ESS) and the SAP Online Store. The SAP GUI for HTML also uses the SAP ITS.

SAP ITS is therefore certainly required for **existing Web applications and the SAP GUI for HTML**, regardless of the SAP Basis or SAP Web Application Server release of the SAP system in question.

Internet Communication Manager (ICM)

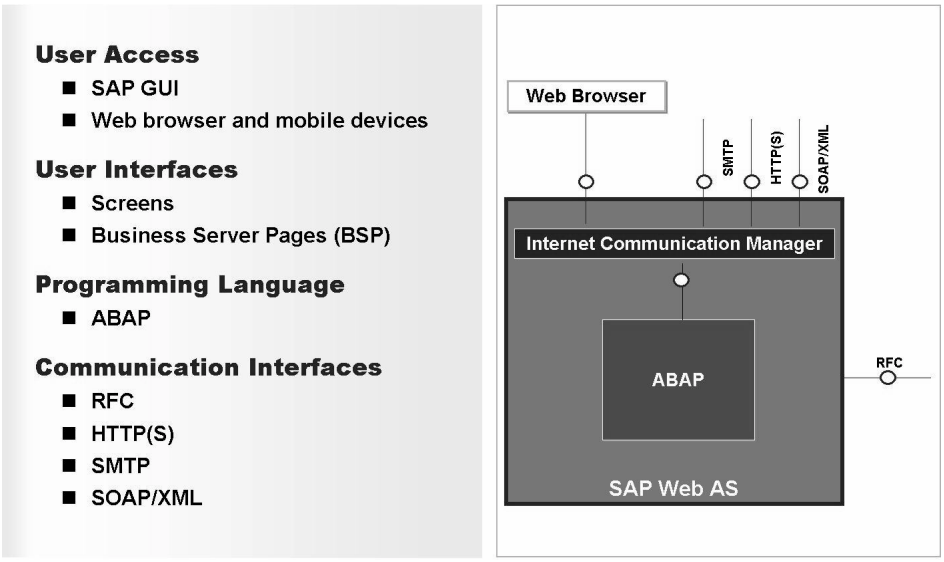


Figure 2: Openness Using the ICM

Based on the highly-scalable infrastructure, new technologies are used as of SAP Web AS 6.10 to process HTTP requests (and other protocols) directly from the Internet or to send HTTP client requests to the Internet. To do this, the SAP kernel was extended with a new process, the Internet Communication Manager (ICM).

SAP Web AS provides the technical platform for **new applications** at SAP or customers and partners. In this way, for example, the central software component of the mySAP Business Intelligence solution, SAP BW 3.x, makes extensive use of the new infrastructure. One programming model for applications of this type is to use Business Server Pages (BSPs).

© 2003 SAP AG. All rights reserved.

You must not equate the ICM with BSPs. BSPs do not work without the ICM, but the ICM is also the technical foundation for other scenarios (such as connecting mobile devices or e-mail software) that do not use BSPs. Although SAP BW 3.x also uses the ICF and therefore the ICM process, it is with its own implementation (without BSPs).

Some product managers have also coined and used the expressed ABSP (for ABAP BSP). To avoid confusion, you should not use this expression unless asked about it.

SAP J2EE Engine

User Access
- SAP GUI
- Web browser and mobile devices

User Interfaces
- Screens
- Business Server Pages (BSP)
- Java Server Pages (JSP)

Programming Language
- ABAP
- Java

Communication Interfaces
- RFC
- HTTP(S)
- SMTP
- SOAP/XML

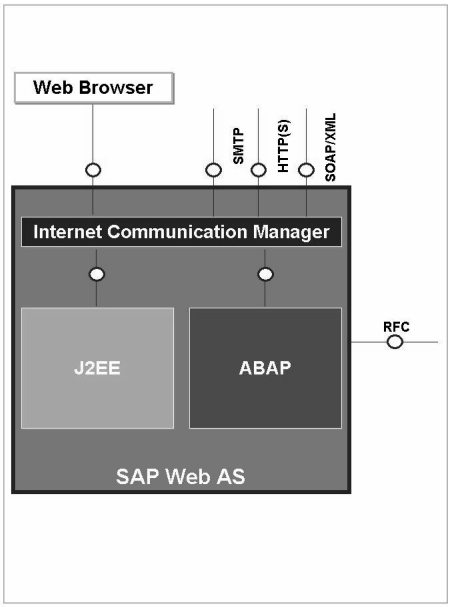


Figure 3: Integrated SAP J2EE Engine

With the SAP J2EE Engine, SAP has a complete J2EE-compatible application server in its product range. This server can be run on a stand-alone basis, or in the framework of SAP Web AS 6.20 (or above).

Developers, therefore, have a mature development and runtime environment for **new applications based on Java**. SAP CRM 3.1 and SAP Enterprise Portal 5.0 are examples of SAP software components that use the SAP J2EE Engine.

Note that the J2EE standard not only describes the (browser-based) user dialog, but also specifies a complete application server.

© 2003 SAP AG. All rights reserved.

If you are using this lesson in the ADM102 course, the entire unit relates to "Technology Components in the Internet Environment", and many Java projects do in fact concentrate on user dialogs. You should nevertheless emphasize that J2EE was created for business applications, and that this obviously involves much more than attractive user interfaces.

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

Consolidation of the material discussed in the lesson

Discussion Questions

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

Discuss with the participants what distinguishes the SAP ITS, ICM, and SAP J2EE Engine from each other. When can the different scenarios be used? Which products do customers use?

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the implementation areas of SAP ITS, ICM, and the SAP J2EE Engine

Related Information

- Quick Link */ui* (for User Interface) on the SAP Service Marketplace
- Quick Link */netweaver* on the SAP Service Marketplace

© 2003 SAP AG. All rights reserved.

Lesson: SAP Internet Transaction Server

7

Lesson Duration: 35 minutes

Lesson Overview

After a short introduction to the architecture of the SAP ITS, this lesson will show you the various options for administration.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the architecture of the SAP ITS
- Start and stop the SAP ITS
- Perform simple administrative tasks on an SAP ITS

The customer should know the basic structure of an SAP ITS from the SAPTEC course. You should therefore focus on the administrative aspects here. For internal information (including information about improvements with SAP ITS 6.20), see the intranet under abap.wdf.sap.corp.de:1080/its.

If you want to use a live demonstration: A number of Internet shops are currently operated at SAP using SAP ITS technology. Under www.sap.com, navigate to *Company* → *SAP Shop* (or go directly to www.sap.com/shop) and choose the *SAP Merchandise Shop* there. Point out the "typical" ITS URL there (containing */scripts/wgate/...*).

In Germany, you could use the fan shop of Bayern Munich FC: shop.fcbayern.de.

Business Example

Your company is making the browser-based SAP GUI for HTML available to some of its employees. As a member of the system administration team, it is your task to ensure the availability of the Internet Transaction Server, through which requests from the SAP GUI for HTML go to the SAP system.

© 2003 SAP AG. All rights reserved.

Architecture of the SAP ITS

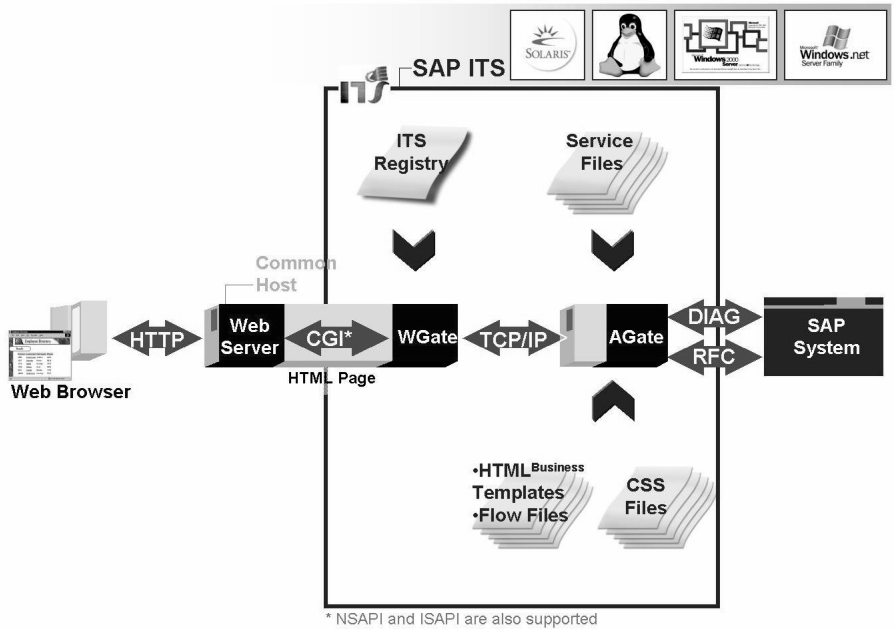


Figure 4: Architecture of the SAP ITS

A Web server (from third-party vendors) is required to operate the SAP ITS. The SAP ITS itself consists of two components, the **WGate** (Web Gateway) and the **AGate** (Application Gateway).

An HTTP(S) request is processed in the following steps:

1. The request is sent from the user's Web browser to the Web server using the HTTP(S) protocol.

2. The Web server recognizes from the structure of the requested URL that it is a request for the SAP ITS, and forwards the request (by CGI, ISAPI, or NSAPI protocol) to the WGate that is running on the same host.

3. The WGate transfers the request to an assigned AGate.

   Note: Up to SAP ITS 6.10, the file *wgate.conf* was used to configure the WGate. As of SAP ITS 6.20, this file simply refers to a section of the ITS Registry, an XML-compliant configuration file.

4. Service files (simple ASCII files) on the AGate determine which function is started in which component system.

5. The requested transaction or the requested function module is executed in the SAP system.

© 2003 SAP AG. All rights reserved.

6. The AGate converts the output into HTML, either using templates (called HTML Business Templates) or dynamically at runtime, as is the case with SAP GUI for HTML. Formatting information, such as font, font size, or colors, can be stored separately in cascading style sheet (CSS) files.

7. The data, which is now correctly formatted, is sent to the user's Web browser through the WGate and the Web server. If appropriate, additional MIME objects (such as images, audio, or video files) are then downloaded by a Web server.

The SAP ITS is **very scalable**, which allows many installation options. As a basic principle, the Web server and WGate always run on the same host. The WGate and AGate are available for various platforms. For a current list, see the Quick Link */sap-its*.

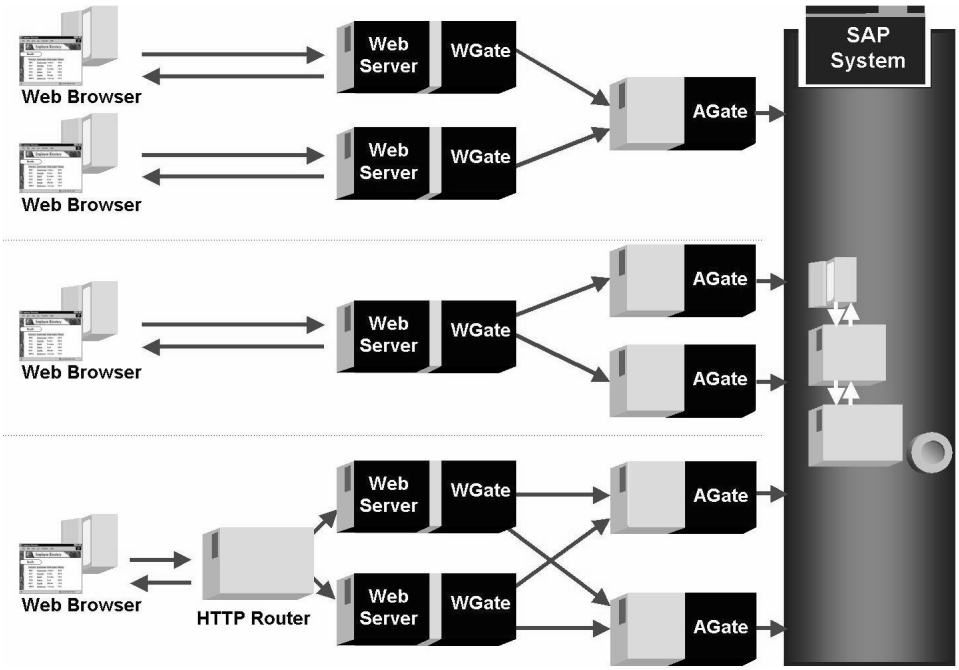


Figure 5: Scalability of the SAP ITS

- External products (HTTP routers, Web switches) can distribute incoming requests over multiple Web servers.
- The WGate and AGate can run together on one host (single host) or on separate hosts (dual host).
- You can run multiple (virtual) SAP ITS instances on a single host.
- A WGate can control multiple AGates.
- An AGate can be addressed by multiple WGates.

© 2003 SAP AG. All rights reserved.

- You can set the dimensions (number of workthreads and memory size) of each AGate for the expected user load.
- An AGate can log on to the SAP system using load distribution (logon groups).
- The AGate can reduce the volume of data transferred to the Web browser (compression with *gzip*).

There really are a large number of options. If customers would like details about these, show the first three entries under the following path in the online documentation for SAP Web AS 6.20: *mySAP Technology Components → ITS/SAP@Web Studio → ITS Tuning and Troubleshooting Guide*. Each of these contains helpful graphics.

The last point (compression) reduce the network load, but requires additional processing (CPU workload). For more information, see the documentation for the service parameters *~http_use_compression* and *~http_compress_level*.

Many customers wrongly regard the SAP ITS as a performance bottleneck. An SAP ITS "out of the box" will certainly not fulfill all requirements. However, you should deliver the message that there are now a very large number of scalability options.

Note: Your customers may also ask questions about security. The SAP ITS also has a great deal to offer here (HTTPS protocol, SNC interface, digital certificates, and so on). This is a good opportunity to advertise course ADM960!

We strongly recommend that you work with a current SAP ITS. The SAP ITS is backward compatible, meaning that, for example, you can use SAP ITS 6.20 with an SAP R/3 4.6C installation without any problems.

© 2003 SAP AG. All rights reserved.

Administration at Operating System Level

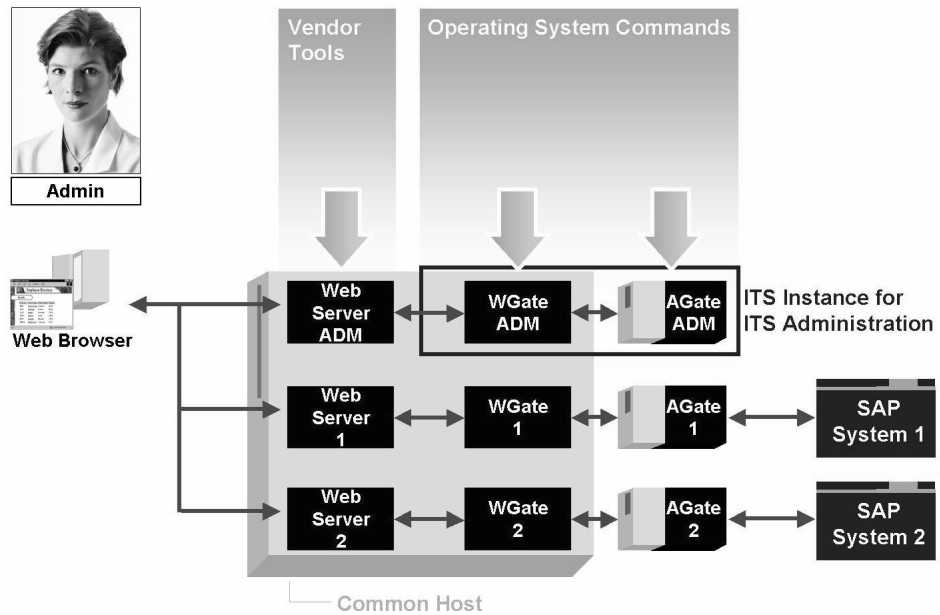


Figure 6: Administration Variants

As stated, an SAP ITS cannot be run without a Web server. SAP supports various products on a range of platforms. For information about administration of the Web server, contact your product's vendor.

As an administrator, how can you **start and stop** an SAP ITS? We will assume here that the required services for the Web server instances (under Microsoft Windows, for example, the required Windows Services) are running.

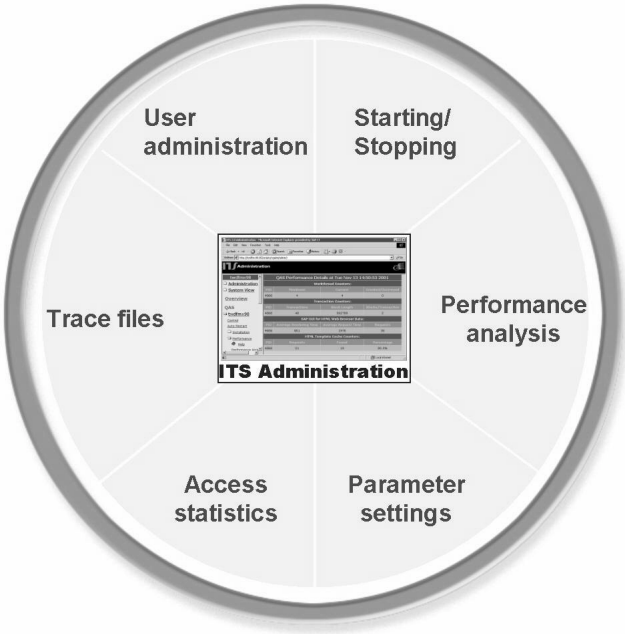
Every SAP ITS instance can be administered from **operating system level**. You can use the programs in the *ADMIN* directory (below the SAP ITS directory tree) to do this.

To start and stop an SAP ITS instance, call the program ITSVControl.exe at operating system level. The option */?* displays possible parameters. Important parameters:

- */c* <*command*>: You specify whether you want to determine the status of an SAP ITS instance (<*command*>=*status*), start it (<*command*>=*start*), or stop it (<*command*>=*stop*)
- */r* <*range*>: Whether the command refers to the WGate (<*range*>=*w[gate]*), the AGate (<*range*>=*a[gate]*), or both components (default; <*range*>=*)
- */v* <*vits*>: Here, <*vits*> stands for the name of an SAP ITS instance; specifying <*vits*>=* means that all installed instances are affected

© 2003 SAP AG. All rights reserved.

Introduction to the ITS Administration Tool



URL for the ITS Administration Tool:
http(s)://<server>:<port>/scripts/wgate/admin/!

Figure 7: Functions of the ITS Administration Tool

You can work more easily with the **ITS Administration tool**, which allows administration of SAP ITS instances using a Web browser. During the installation, you have the option of creating a separate SAP ITS instance for the ITS Administration tool (recommended approach). You can call tools using a special URL (see the figure above). Initially, the user *itsadmin* exists with the password *init*. After logging on, you can create additional users and assign authorizations to them (for example, to restrict them to display functions only).

You can find the functions for starting and stopping in the *Control* area for each SAP ITS instance.

Since SAP ITS 6.10, you can also monitor AGates of a virtual SAP ITS on remote hosts, although there are functional restrictions when compared to local SAP ITS instances.

The following list contains a number of selected functions of the ITS Administration tool:

- User administration
- Configuration of all SAP ITS parameters
- Starting/stopping
- Evaluating various log and trace files

© 2003 SAP AG. All rights reserved.

- Performance monitoring and tuning

You can also use a special agent (*SAPCCMSR*) to monitor an SAP ITS using CCMS analysis monitors (transaction RZ20). For more information, see the *System Monitoring and Alert Management*) area in the SAP Service Marketplace under the Quick Link */systemmanagement*, and SAP Note 418285.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

13

Exercise 1: Calling and Administering the SAP ITS

Exercise Duration: 15 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Call the SAP GUI for HTML
- Start and stop the SAP ITS at operating system level
- Make settings using the ITS Administration tool

Business Example

Your company uses a number of Employee Self-Service (ESS) functions in connection with an SAP R/3 Enterprise system. As an administrator, you are responsible for monitoring the SAP ITS through which the ESS applications run.

Task 1 SAP GUI for HTML

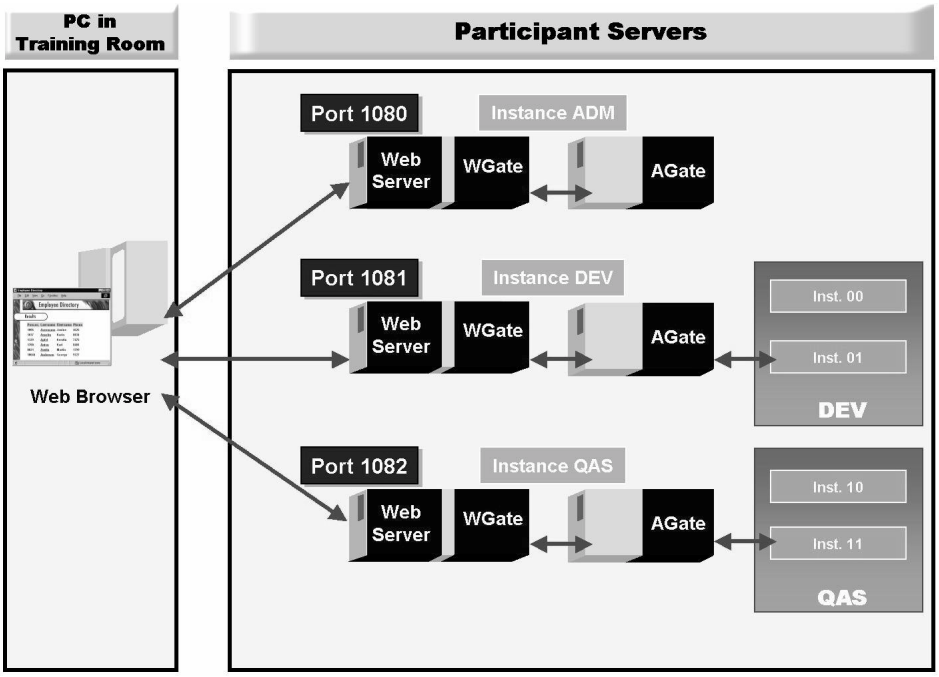


Figure 8: Complete Scenario of the Training Landscape

Log on to the SAP GUI for HTML and determine the SAP ITS release.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. Log on to your SAP system with the user **ADM102-##** using the SAP GUI for HTML.
2. Which release of the SAP ITS is used on the training system?
3. What happens when you log off the SAP GUI for HTML?

Result

You can call the SAP GUI for HTML and know which release of SAP ITS is in use.

Task 2 Optional: Administration at Operating System Level

Restart an SAP ITS instance with operating system tools.

1. Use the program ITSVControl to determine the status of all SAP ITS instances on your server.
2. Stop the **AGate of your** SAP ITS instance.
3. Restart the AGate of your SAP ITS instance.

Result

You can administer SAP ITS instances at operating system level. You require this knowledge, for example, if the instance with the ITS Administration tool is not running.

Task 3 ITS Administration Tool

Call the ITS Administration tool and use some of the functions.

1. Log on to the ITS Administration tool with the default user **itsadmin** and the password **init**.
2. Create a user, **`<sid>admin`**, for the ITS Administration tool that can only administer its own instance (and display the others).

   Then start the ITS Administration tool with the user you have just created.
3. Increase the maximum number of workthreads for your instance to eight.
4. **Optional:** Use the online documentation to determine the meaning of the service parameter *~exitURL*.

   Change the parameter for your instance to a Web address of your choice and observe the effect of this change in the SAP GUI for HTML.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Result

You are familiar with the interface of the ITS Administration tool and can make individual settings.

© 2003 SAP AG. All rights reserved.

Solution 1: Calling and Administering the SAP ITS

Task 1 SAP GUI for HTML

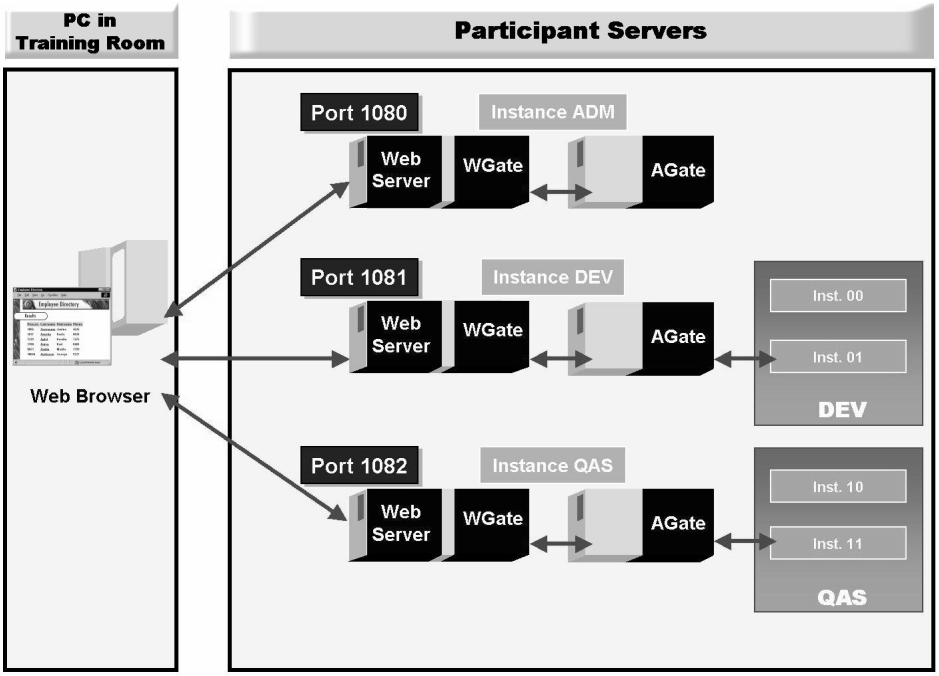


Figure 9: Complete Scenario of the Training Landscape

Log on to the SAP GUI for HTML and determine the SAP ITS release.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. Log on to your SAP system with the user **ADM102-##** using the SAP GUI for HTML.

   a) Start your local Web browser (in the training room) and enter the following URL: **`http://<server with domain>:<SID ITS port>/scripts/wgate/webgui/!`** (Example for the QAS group and the server twdf0042: **`http://twdf0042.wdf.sap.corp:1082/scripts/wgate/webgui/!`**).

   Note: The ITS port may not be available from your training room. In this case, use the Web browser available on your server.

   b) On the logon page, fill out the fields for *Client*, *SAP user* (*Login*), *Password*, and *Language*, and choose the *Logon* button.

   You can now call transactions using the command field or the *SAP Easy Access* menu (which you are familiar with from the SAP GUI for Windows).

   Hint: You can also partly or completely define the fields on the logon page using SAP ITS parameters. In addition to logging on with a user and password, the SAP ITS allows the use of digital certificates.

2. Which release of the SAP ITS is used on the training system?

   a) The SAP ITS release is usually contained as a comment in all HTML pages generated by the SAP ITS. To display this, choose the following function in Internet Explorer on any page generated by the SAP ITS (such as the output of the SAP GUI for HTML): *View* → *Source* or right-click and select *View Source*.

3. What happens when you log off the SAP GUI for HTML?

   a) In the SAP GUI for HTML, choose *System* → *Log off* and confirm the confirmation prompt.
   b) With SAP GUI for Windows, the window in use would now close. With SAP GUI for HTML, the browser window remains open. However, it displays the error message, "You are not authorized to view this page."

Result

You can call the SAP GUI for HTML and know which release of SAP ITS is in use.

Continued on next page

© 2003 SAP AG. All rights reserved.

Task 2 Optional: Administration at Operating System Level

Restart an SAP ITS instance with operating system tools.

1. Use the program ITSVControl to determine the status of all SAP ITS instances on your server.
   a) If you have not already done so, log on to the Terminal Server Client (also known as the RDP Client) at operating system level of your server using the user **`<sid>adm`**.
   b) Open Windows Explorer and navigate to the directory **G:\Program Files\ITS\2.0\ADMIN**. Right-click on the *ADMIN* directory and choose *CMD Prompt Here*.
   c) You can obtain the desired information by running the command **`ITSVControl /c status /v *`**.
2. Stop the **AGate of your** SAP ITS instance.
   a) In the command prompt (same path as previously), run the command **`ITSVControl /c stop /r agate /v <SID>`**. Replace **`<SID>`** with your system name (**`DEV`** or **`QAS`**).

      If you want, you can now try to log on to the SAP GUI for HTML again.
3. Restart the AGate of your SAP ITS instance.
   a) Run the command **`ITSVControl /c start /r agate /v <SID>`**.

Result

You can administer SAP ITS instances at operating system level. You require this knowledge, for example, if the instance with the ITS Administration tool is not running.

Task 3 ITS Administration Tool

Call the ITS Administration tool and use some of the functions.

Continued on next page

 © 2003 SAP AG. All rights reserved.

1. Log on to the ITS Administration tool with the default user **itsadmin** and the password **init**.

   a) Start your local Web browser (in the training room) and enter the following URL: **`http://<server with domain>:<Admin ITS port>/scripts/wgate/admin/!`** (Example for the server twdf0042: **`http://twdf0042.wdf.sap.corp:1080/scripts/wgate/admin/!`**).

   Note: The ITS port may not be available from your training room. In this case, use the Web browser available on your server.

   b) Log on with the user **`itsadmin`** and the password **`init`**.

2. Create a user, **`<sid>admin`**, for the ITS Administration tool that can only administer its own instance (and display the others).

   Then start the ITS Administration tool with the user you have just created.

   a) Under *Administration* → *User Management*, choose the *Add* button.
   b) Under *Login*, enter **`<sid>admin`** and under *Password*, enter any password. Under *User Privileges*, check *Administrator* only for your instance, and for the others leave the setting at *View Only*. Then choose *Add*.
   c) In the ITS Administration tool, there is no option for logging off. You can either close the Web browser window completely and then reopen it, or choose the URL **`http://<Server with domain>:<Admin ITS Port>/scripts/wgate/admin/!`** in the *Address* field.
   d) Now log on as the user **`<sid>admin`** with the password you assigned to the user.

Continued on next page

 © 2003 SAP AG. All rights reserved.

3. Increase the maximum number of workthreads for your instance to eight.

   a) In the ITS Administration Tool, navigate to the following setting: <*SID*> → <*Host*> → *Configuration* → *Performance* → *MaxWorkThreads*.

   b) Enter the value `8`. When you save, you receive a message that this change takes effect only after a restart.

   Hint: The number of available workthreads determines how many requests the AGate can process simultaneously.

   The value cannot be arbitrarily increased, as the main memory and CPU resources will be exhausted at some point. The settings in the training system represent a minimal configuration of the SAP ITS; production installations must be dimensioned for the expected workload.

   c) To restart, choose <*SID*> → *Control*. Restarting the AGate is sufficient. To do this, choose *Restart* for the entry *ITS Manager* - <*SID*>.

   Note: The Web server instance on which the WGate is installed is hidden behind *W3SVC/4*.

4. **Optional:** Use the online documentation to determine the meaning of the service parameter *~exitURL*.

Continued on next page

© 2003 SAP AG. All rights reserved.

Change the parameter for your instance to a Web address of your choice and observe the effect of this change in the SAP GUI for HTML.

a) Call the online documentation for SAP Web AS 6.20, for example through the SAP GUI for Windows (if this is configured), or the SAP Help Portal: help.sap.com.

b) You can search for the parameter *~exitURL* or choose it from the list under *mySAP Technology Components* → *ITS/SAP@Web Studio* → *ITS Service Parameters* → *Service Parameter List*.

Result

Result: *~exitURL* points to the Web address that is called when an ITS Service is closed.

c) To change the value, call the following path in the ITS Administration Tool: <*SID*> → <*Host*> → *Configuration* → *Global Services*→ *All Settings*. In this list of the global parameters for your instance, enter any Web address, such as help.sap.com, in the *~exitURL* field.

Then choose *Save* (at the end of the list).

d) Changes to service parameters do not require a restart of the SAP ITS. Now repeat the "Logging On to SAP GUI for HTML" section of the first exercise. When you log off, no error message appears – instead, your URL is called.

Result

You are familiar with the interface of the ITS Administration tool and can make individual settings.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Describe the architecture of the SAP ITS
- Start and stop the SAP ITS
- Perform simple administrative tasks on an SAP ITS

Related Information

- Online documentation for SAP Web AS 6.20: *mySAP Technology Components → ITS/SAP@Web Studio*
- Quick Link */sap-its* on the SAP Service Marketplace
- SAP Note 325616: *ITS System requirements*
- SAP Note 197746: *Maintenance Strategy: Internet Transaction Server (ITS)*
- SAP Note 321426: *SAP Internet Transaction Server 4.6D: New Features*
- SAP Note 419297: *SAP Internet Transaction Server 6.10: New Features*
- SAP Note 531617: *SAP Internet Transaction Server 6.20: New Features*
- SAP course NET170: *SAP Internet Transaction Server: Administration*

© 2003 SAP AG. All rights reserved.

Lesson: Internet Communication Manager

23 Lesson Duration: 45 minutes

Lesson Overview

In this lesson, you will learn about the Internet Communication Manager (ICM) process and ways to administer it.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the implementation area of the ICM
- Configure and monitor the ICM

The ICM is the process that turns the "classic" SAP application server into SAP **Web** Application Server.

ADM102 focuses on the administration of the ICM, which could conceivably be neglected. Creating applications is more complicated, but is not the topic of this presentation. The course NET200 covers creating applications based on BSPs.

If you want to use a live demonstration: In Germany, you can use the Job listings for Otto distribution under www.otto.com/job. On this site, choose *View* → *Source* to show the "typical" BSP URLs (containing */sap/bc/bsp/sap/z...*).

Business Example

In the context of your company's planned e-business activities, you want to implement applications based on Business Server Pages (BSPs). As a member of the system administration team, it is your task to configure the SAP systems in accordance with your requirements. To do this, you want to obtain an overview of the ICM process.

© 2003 SAP AG. All rights reserved.

Architecture of the ICM Process

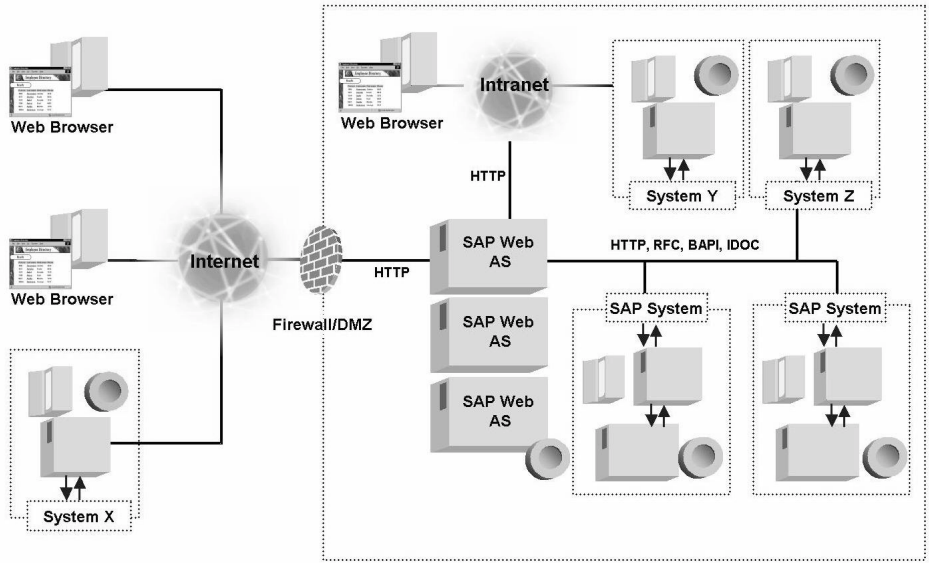


Figure 10: System Landscape with SAP Web AS (Example)

The figure above shows an example of a **system landscape** in which Web browsers from the Internet and intranet are connected with an SAP Web AS (in this case, distributed across a number of servers). Important features are:

- Support for standard Web protocols such as HTTP, HTTP, WebDAV, SOAP, and SMTP
- Output of standard Web formats such as HTML, XML, and XLST
- Complete integration into the SAP environment (development environment, user administration, authorization concept, system monitoring, and communication protocols)

SAP Web AS can function both as a **Web server** (server role) and as a **Web client** (client role). The server role, in which the SAP Web AS can accept and process HTTP requests from any Web client (such as a Web browser) and send back an HTTP response, is what we will discuss in this lesson.

Within a work process, the Internet Communication Framework (ICF) provides the environment for handling HTTP requests. The ICF is the bridge between the C kernel of the SAP system and the application program created in ABAP.

As of SAP Web AS 6.10, work processes can directly generate Web-compatible content in a way that can be forwarded to a browser using the ICM. One way of creating content of this type is to use applications with BSPs that were developed in the SAP system using a tool of transaction SE80, the Web Application Builder.

© 2003 SAP AG. All rights reserved.

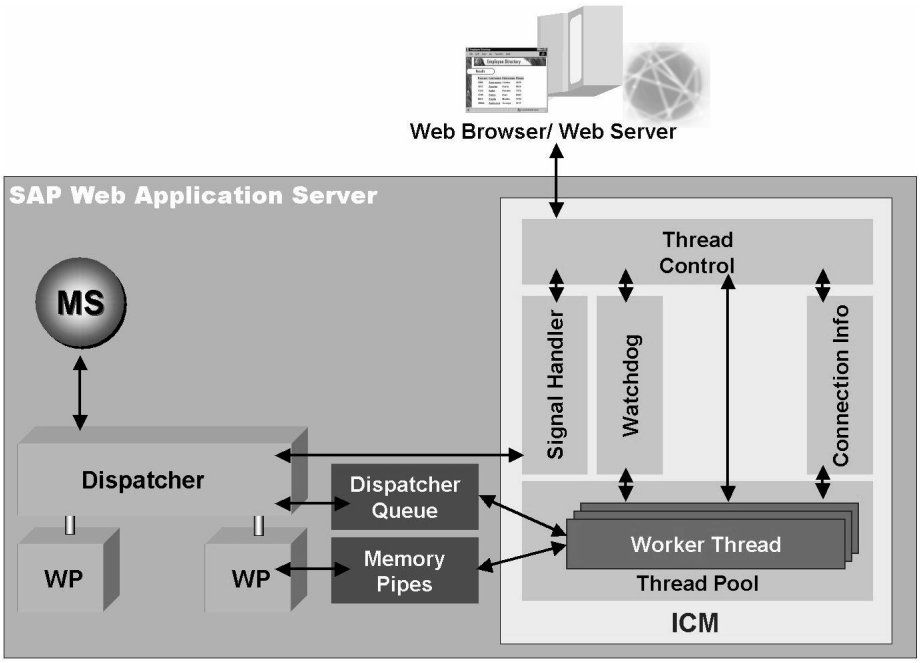


Figure 11: Internal Structure of the ICM Process

From a technical point of view, the ICM is a **separate** process (*icman* at operating system level) that is started and monitored by the dispatcher. Its task is to ensure that the SAP system can communicate with the outside world (using HTTP, HTTPS, and SMTP). In the server role, it can process requests from the Internet that arrive with URLs with the server/port combination for which the ICM is listening. The ICM then calls the appropriate local handler, depending on the URL. The ICM process uses **threads** to process the created workload in parallel. The components of the ICM are:

- **Thread control**: This thread receives the incoming TCP/IP requests and creates (or wakes) a worker thread from the thread pool to process the request.
- **Worker thread**: This thread handles requests and responses for a connection. A worker thread contains an I/O handler for the network input and output, and various plug-ins for the different supported protocols.
- **Watchdog**: Normally, a worker thread waits for the response (regardless of whether it is the client or the server); if a timeout occurs, the watchdog takes over the task of waiting for the response. The worker thread can then be used for other requests.
- **Signal Handler**: The signal handler processes signals that are sent by the operating system or another process (such as the dispatcher).

© 2003 SAP AG. All rights reserved.

- **Connection info**: This table contains information about every existing network connection.
- **Memory pipes**: These memory-based communication objects allow data transfer between the ICM and the work processes.

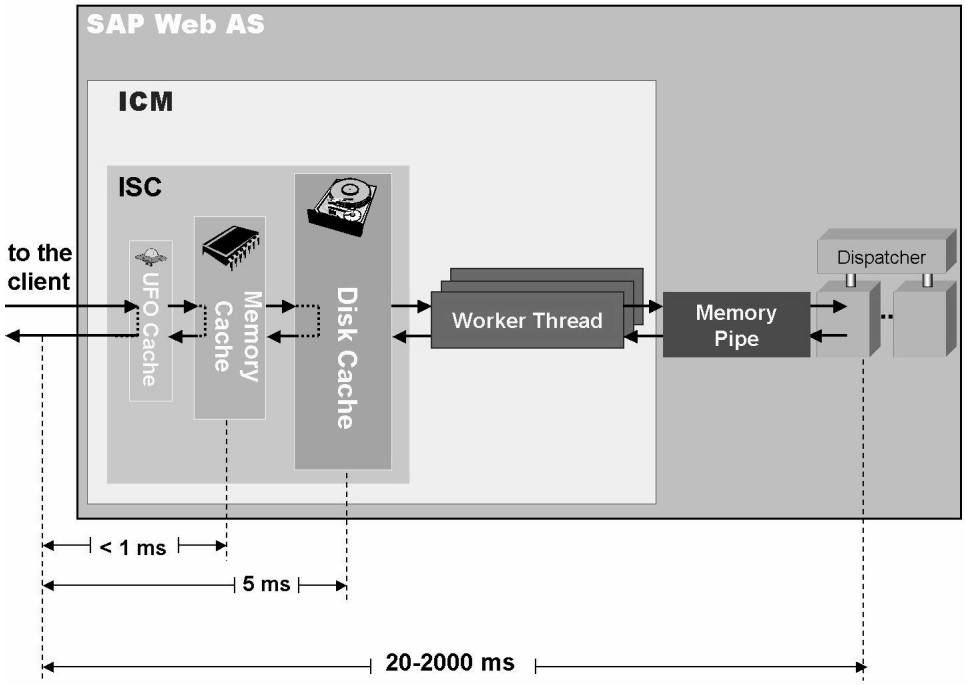


Figure 12: Internet Server Cache (ISC)

A part of the ICM that is important for performance is the Internet Server Cache (ISC), which stores HTTP(S) objects before they are sent to the Web browser. The next request can then be met directly from the ISC, which can significantly speed up accesses.

Some features of the ISC:

- **Two-level hierarchy**: When storing objects, the advantages of both the high speed of main memory (memory cache) and the storage capacity of hard disks (disk cache) are used.
- **Dynamic caching**: Traditional products are based on HTTP proxies and usually offer caching only of static content, such as images. The ISC can also cache dynamic content such as JSPs or BSPs.
- **Active caching**: The application has full control over ensuring that the objects in the cache are up to date.
- **UFO caching**: Invalid requests (Unfound Objects)) that lead to error situations in the application server or the database are directly rejected, so that the system is protected against invalid or malignant requests.

© 2003 SAP AG. All rights reserved.

- **Browser-dependent caching**: The developer of a BSP can define whether his or her application is dependent on browser type. If this indicator is set, the ISC uses the data in the cache only for requests from the same browser type.

The ISC is configured using the profile parameter *icm/HTTP/server_cache** and can be monitored and invalidated from the SAP system.

Only for instructors of the ADM102 course: Do not yet demonstrate anything about the ISC, as there are (probably) not yet any objects in it. There is a demonstration in the ICF section later in this unit (of the ADM102 course) that stores a large number of objects in the ISC.

Start Procedure and Monitoring

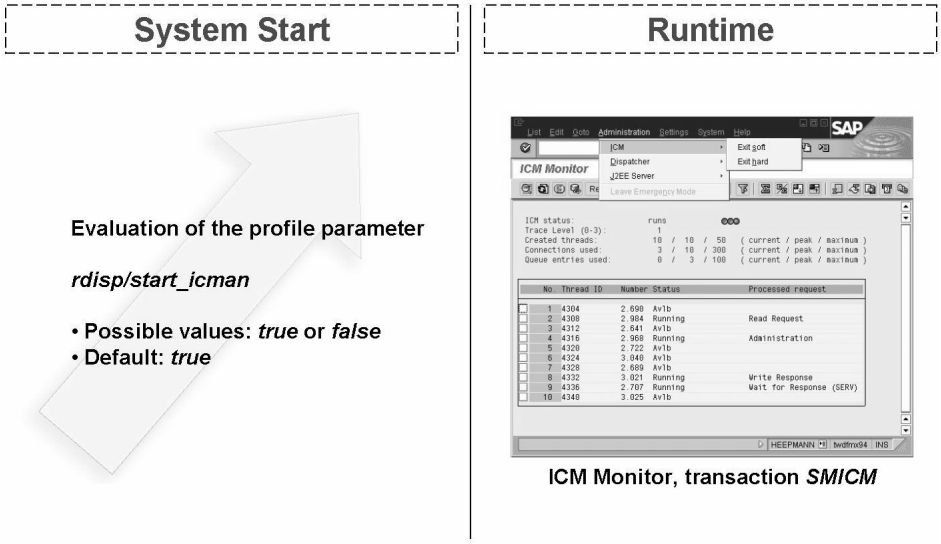


Figure 13: Starting the ICM

The profile parameter *rdisp/start_icman* controls whether an ICM process is also started when an application server is started. If no value is specified, the default setting *true* applies. You configure the ICM using profile parameters (most of which begin with *icm/*. The settings for *icm/server_port_<xx>* are of particular importance. These settings define the port used for each protocol.

In the SAP system, you can quickly obtain an overview of which application servers are running with an ICM using the server overview (transaction SM51).

© 2003 SAP AG. All rights reserved.

For more detailed information (such as the thread ID), see the **ICM monitor** (transaction SMICM). From this transaction, you can choose the menu path *Administration* → *ICM* to terminate the ICM with a soft termination (Signal 2) or a hard termination (Signal 9). The dispatcher will then start a new ICM process. By choosing *Administration* → *Dispatcher* → *Force ICM Restart*, you cause the dispatcher to restart the ICM if it has been terminated due to an error.

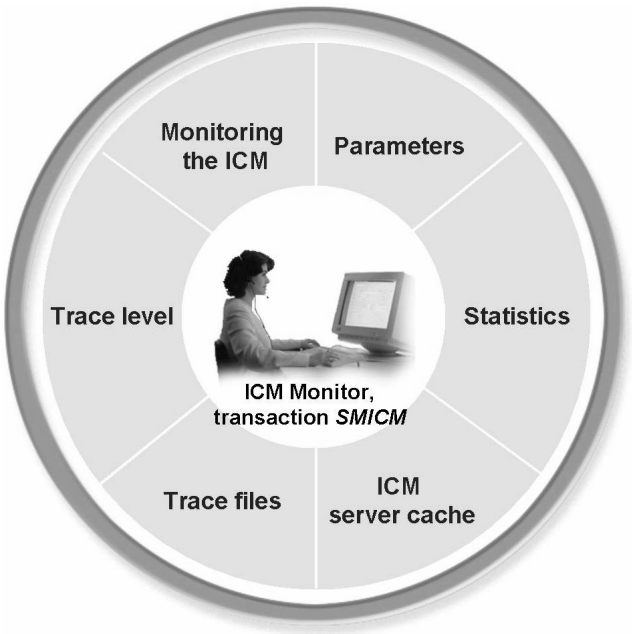


Figure 14: Monitoring the ICM Using the ICM Monitor

The most important tool for an administrator in the ICM environment is the ICM monitor (transaction SMICM). Some administrative activities (all available from transaction SMICM) are:

- Monitoring and restarting the ICM
- Setting the trace level (*Goto* → *Trace level* → *Set*) values from 0 to 3
- Evaluating the trace files (*Goto* → *Trace file* → *Display file*); the system reads the file *dev_icm* from the *work* directory of the current instance
- Overview of the profile parameters (*Goto* → *Parameters* → *Display*). The ICM is configured using profile parameters. The displayed values apply for the instance to which you are currently logged on. For documentation for the parameters, see the transaction RZ11 and the SAP Library.
- Display the statistics (*Goto* → *Statistics* → *Display*): You can use these statistics to find out how many requests the ICM has processed since it was started (or since the statistics were reset). The system also displays information about processing duration.

© 2003 SAP AG. All rights reserved.

- Monitoring (*Goto* → *HTTP Server Cache* → *Display*) and resetting the ICM server cache: The ICM server cache stores HTTP objects before they are sent to the client. The next time that this object is requested, the content can be sent directly from the cache to the client.

You can determine some of the listed data at operating system level using the *icmon* program. The call *icmon -h* displays the possible parameters for this small program, which can also, among other things, generate requests to simulate normal system workload.

The *icmon* program is (in the same way as *dpmon*) primarily a tool for our developers and is therefore neither documented nor officially supported by SAP.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

31

Exercise 2: Administration of the ICM

Exercise Duration: 10 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Monitor the ICM process

Business Example

Your company uses SAP BW 3.1 in the context of mySAP Business Intelligence. Browser-based functions such as Web reporting, interactive charts, and the Business Explorer Browser (BEx Browser) are to be made available at users' workplaces. As an administrator, you are responsible for monitoring the ICM processes that implement the connection of the Web browser to the SAP system.

Task 1 Checking the ICM Settings

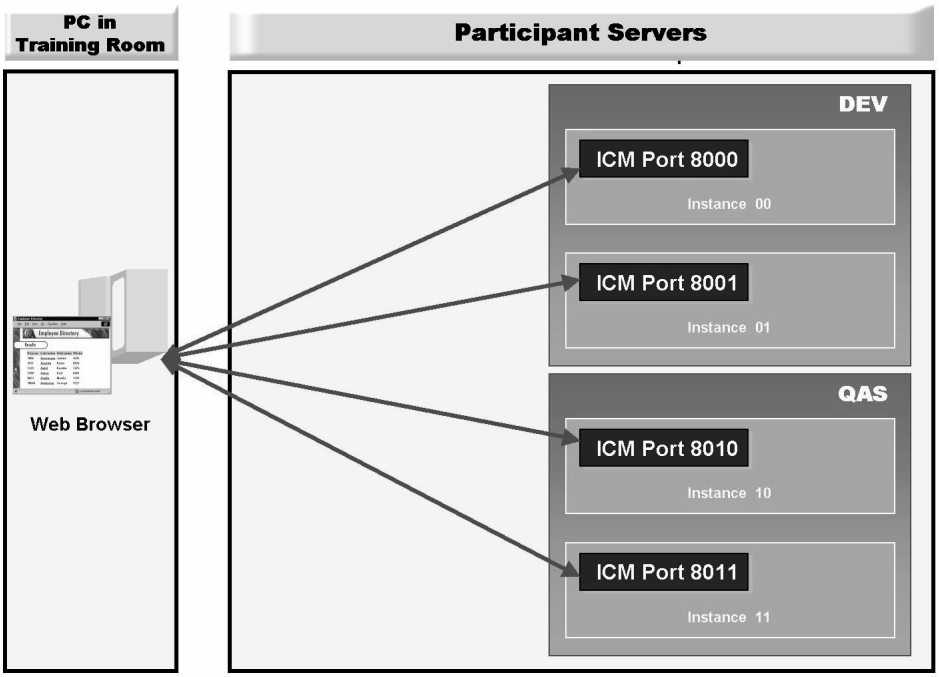


Figure 15: Complete Scenario of the Training Landscape

Number, port, and release of the ICM processes in the training environment

1. How many ICM processes are running in your SAP system?

Continued on next page

© 2003 SAP AG. All rights reserved.

2. Determine the port through which requests in the HTTP protocol are processed for the application server to which you are currently logged on.
3. Which release of the ICM is used on the training system?

Result

You know the port and release for the ICM process on the training system.

Task 2 Simple HTTP Request

Start a request in the Web browser and monitoring with the ICM monitor.

1. In your local Web browser (in the training room), call the following URL: **`http://<server with domain>:<ICM port>/sap/public/info`** (Example for the QAS group and the twdf0042 server: **`http://twdf0042.wdf.sap.corp:8010/sap/public/info`**).

   The system should display an XML document containing system data.

   Note: The ICM port may not be available from your training room. In this case, use the Web browser available on your server.

2. Open the ICM monitor and find out how many requests from your Web browser (perhaps due to a *Refresh* on the page above) have been processed by the worker threads.

   Hint: The data in the ICM monitor is instance-specific.

Result

You can monitor the activity of the ICM.

Task 3 Optional: Workload Test with the icmon Tool

Monitor the ICM work threads using generated workload.

1. At operating system level of your server, start the **icmon pf=<instance profile name>** command, where you specify an instance profile of your SAP system for **instance profile name**.

   Use the command **g** to generate workload as follows:

Continued on next page

 © 2003 SAP AG. All rights reserved.

Host	Host on which your SAP system is running, such as **`twdf0042`**
Path	**`/sap/public/ping`** (Default setting)
Port	An ICM Port valid for your system, such as **`8011`**
Optional Attributes	**`no`** (Default setting)
Number of requests	**`5000`**
Number of threads	**`2`**

2. Observe in the ICM monitor how the requests generated by *icmon* are processed by the worker threads.

Result

You can use the *icmon* tool to monitor the ICM and to start workload simulations.

© 2003 SAP AG. All rights reserved.

Solution 2: Administration of the ICM

Task 1 Checking the ICM Settings

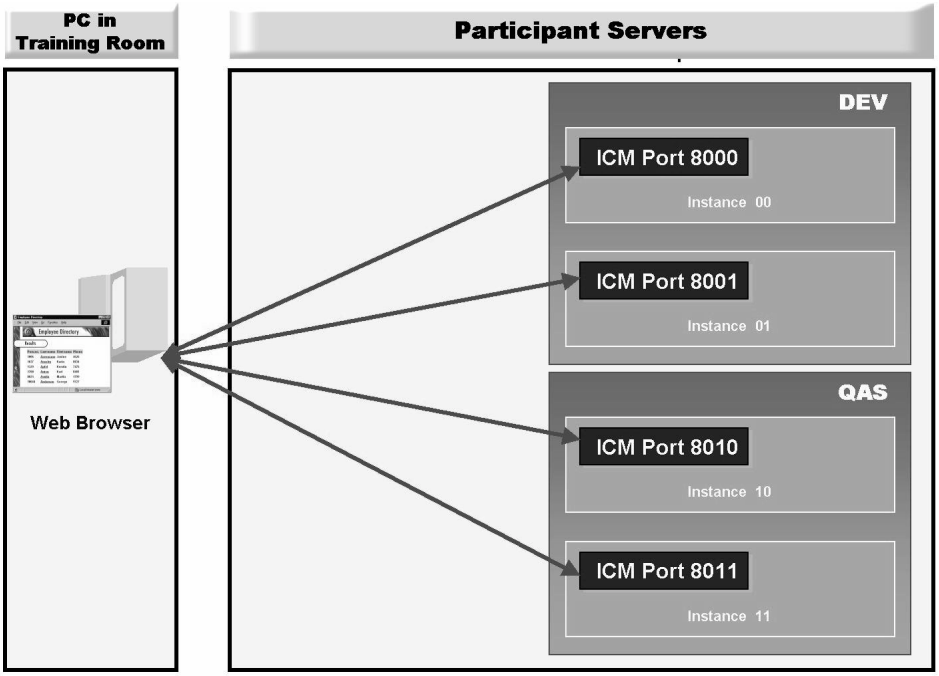


Figure 16: Complete Scenario of the Training Landscape

Number, port, and release of the ICM processes in the training environment

1. How many ICM processes are running in your SAP system?
   a) In the server overview (transaction SM51), count the application servers for which the ICM process is listed. There should be an ICM process configured for each of your instances.

Continued on next page

© 2003 SAP AG. All rights reserved.

2. Determine the port through which requests in the HTTP protocol are processed for the application server to which you are currently logged on.

   a) Check the value of the profile parameter *icm/server_port_0*, for example:

      - In transaction SMICM by choosing the *Services* button or by choosing *Goto* → *Services*
      - By calling the report **RSPFPAR**
      - In transaction RZ10
      - In the ICM monitor (transaction SMICM) by choosing *Goto* → *Parameters* → *Display*

Hint: The determined port is instance-specific. On the training systems, the parameter *icm/server_port_0* has the value *PROT=HTTP,PORT=80$$*. The variable *$$* is replaced by the instance number (unique for each server) when the ICM is started, ensuring that unique ports are definitely produced.

3. Which release of the ICM is used on the training system?

   a) You can determine the ICM release in the ICM monitor by choosing the *Release Notes* button or by choosing *Goto* → *Release Notes*.

      The information that you are looking for is at the start of the list. At the end of the list, all problems that are solved with the current patch level are listed (with associated SAP Notes).

Result

You know the port and release for the ICM process on the training system.

Task 2 Simple HTTP Request

Start a request in the Web browser and monitoring with the ICM monitor.

1. In your local Web browser (in the training room), call the following URL: **`http://<server with domain>:<ICM port>/sap/public/info`** (Example for the QAS group and the twdf0042 server: **`http://twdf0042.wdf.sap.corp:8010/sap/public/info`**).

Continued on next page

© 2003 SAP AG. All rights reserved.

The system should display an XML document containing system data.

Note: The ICM port may not be available from your training room. In this case, use the Web browser available on your server.

a) Enter the specified URL in your local Web browser and choose *Enter*.

Hint: All services under */sap/public* use a predefined user; therefore, no logon is required for this request.

As of SAP Web AS 6.20, services must be explicitly activated. This has already been done in the training system for the services specified in the exercises.

2. Open the ICM monitor and find out how many requests from your Web browser (perhaps due to a *Refresh* on the page above) have been processed by the worker threads.

Hint: The data in the ICM monitor is instance-specific.

a) In the ICM monitor (transaction SMICM), choose the *Refresh* button after you have sent a few requests to the ICM.

Result

You can monitor the activity of the ICM.

Task 3 Optional: Workload Test with the icmon Tool

Monitor the ICM work threads using generated workload.

1. At operating system level of your server, start the **icmon pf=<instance profile name>** command, where you specify an instance profile of your SAP system for **instance profile name**.

   Use the command **g** to generate workload as follows:

Host	Host on which your SAP system is running, such as `twdf0042`
Path	`/sap/public/ping` (Default setting)
Port	An ICM Port valid for your system, such as `8011`

Continued on next page

© 2003 SAP AG. All rights reserved.

Optional Attributes	**no** (Default setting)
Number of requests	**5000**
Number of threads	**2**

a) If you have not already done so, log on to the Terminal Server Client (also known as the RDP Client) at operating system level using the user **<sid>adm**.

b) Open Windows Explorer and navigate to the directory **G:\usr\sap\<SID>\SYS\profile**. Right-click on the *profile* directory and choose *CMD Prompt Here*.

c) Start the *icmon* program and specify an instance profile: **icmon pf=<instance profile name>** (Example for the QAS group and the server twdf0042: **icmon pf=QAS_D11_twdf0042**).

You can use the command **m** to obtain an overview of all available functions.

d) Use the command **g** and the parameters specified in the exercise to generate workload.

2. Observe in the ICM monitor how the requests generated by *icmon* are processed by the worker threads.

a) In transaction SMICM, choose the *Refresh* function to observe the activity of the worker threads.

Remember that the display in transaction SMICM is not system-wide, but only applies to your instance.

Result

You can use the *icmon* tool to monitor the ICM and to start workload simulations.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Describe the implementation area of the ICM
- Configure and monitor the ICM

Related Information

- Online documentation for SAP Web AS 6.20: *mySAP Technology Components → SAP Web Application Server → Web Applications (BC-MAS) → Web Application Server → Architecture of the SAP Web Application Server*
- SAP Note 508300: *ICM Patch Collection (Kernel 6.20)*
- SAP Note 421359: *ICM: Binding ports <1024 on Unix*

© 2003 SAP AG. All rights reserved.

Lesson: Internet Communication Framework

39 Lesson Duration: 35 minutes

Lesson Overview

The Internet Communication Framework (ICF) provides an environment for handling HTTP requests in the work process of an SAP system. This lesson introduces the ICF and provides more information about some administrative issues.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service

Simply put, the ICF form the bridge between the base technology (which ICF service can be called using which URL) and the development environment (which BSP application is used).

Do not confuse the customers with too many details about the classes and interfaces. You can direct programmers to the NET* curriculum (and in particular the course NET200), in which transaction SICF is also discussed – and from a developer's point of view. In ADM102, the focus is on enabling the participants to understand the ICF and presenting a number of selected administrative possibilities.

We recommend that your own preparation for this lesson should include the online documentation for the SAP Web AS 6.20, path *mySAP Technology Components → SAP Web Application Server → Web Applications (BC-MAS)→ Web Application Server→ Internet Communication Framework*.

Business Example

Your company wants to use BSP-based applications to connect your SAP systems to the Internet. As a member of the system administration team, it is your task to create links between called URLs and services and programs of the SAP system.

© 2003 SAP AG. All rights reserved.

Classifying the ICF

The Internet Communication Framework (ICF) provides the environment for handling HTTP requests in work processes of an SAP system (in its role as a Web server and a Web client). The ICF is the bridge between the kernel of the SAP system and the application program written in ABAP. The ICF consists of ABAP classes and interfaces, the objects and methods of which can be accessed in a BSP application.

The following provides more information about using the SAP system as a Web server (HTTP(S) server). For information about the Web client role of the SAP system, see the online documentation.

The application logic that is to be called by an HTTP request from the intranet or Internet is implemented by the HTTP request handler in each case. An HTTP request handler is a program (or, more precisely, an ABAP class) that is identified using a URL, and which receives HTTP requests that use this URL. The task of the HTTP request handler is to receive the data that is sent by a request (for example, coded into the URL as "query string" information), to perform a number of handler-specific processes, and to generate a response to this HTTP request.

If an HTTP request is received by the ICM that is to be processed in a work process, the task handler takes control. It then starts the ICF controller. From now on, we are in the ABAP world and in the ICF.

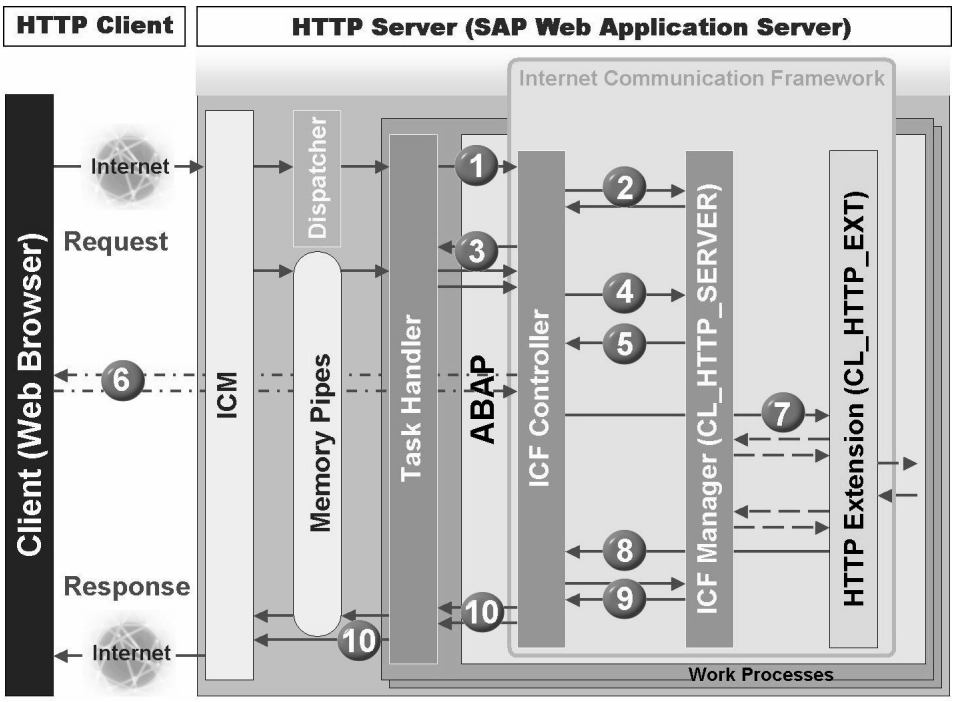


Figure 17: Interaction Model of an SAP System in the Server Role

© 2003 SAP AG. All rights reserved.

The system processes the following steps when the ICF controller is started:

1. Call the function module *HTTP_DISPATCH_REQUEST*.
2. Generate an object of the class *CL_HTTP_SERVER*, which is a server control block.
3. Read the HTTP request.
4. Fill the server control block with the HTTP request data.
5. Select the desired HTTP request handler using the URL.
6. Log on to the client.
7. Call the HTTP request handler (this can process the request data, call applications, access the response object, and so on).
8. Control returns to the ICF controller, which may transfer it to additional handlers.
9. A new HTTP stream is now generated in the structure of the object in the same way as the data is read (serialization of the response).
10. The result is written to a memory pipe and is returned to the client through the ICM.

The red arrows in the graphic represent the control flow; the blue arrows represent the data flow. You do not need to explain every step in detail. The participants should gain an idea of how to classify the ICF.

Maintaining the ICF Services

From a technical point of view, there is an ABAP class behind an HTTP request handler. This class implements the interface IF_HTTP_EXTENSION and the method HANDLE_REQUEST (). SAP delivers classes of this type; customers can, of course, also create their own classes with the Class Builder (transaction SE24, integrated into the Object Builder, and transaction SE80).

Linking a particular URL with an HTTP request handler is the task of ICF services. An ICF service therefore creates a connection between a URL to which an HTTP request is sent and development objects that process this request.

An SAP system (with SAP Web AS its technical foundation) already contains various services when it is delivered. The exact scope depends of course on the system type (SAP R/3, SAP BW, and so on) and the release. You can obtain an overview of all available services using the central maintenance transaction for ICF services, transaction SICF. All available

© 2003 SAP AG. All rights reserved.

services are displayed in a hierarchical structure in transaction SICF. The complete path for a service (such as */sap/public/info*) ultimately determines (together with the protocol, server name, and port) the URL under which the service can be called.

As of SAP Web AS 6.20, all ICF services are initially inactive. Activated ICF services represent a security risk in that they can be directly accessed using the HTTP protocol from the intranet or the Internet (depending on your network configuration). You should therefore ensure that access is restricted using appropriate measures, such as only allowing the ICF service to be accessed using one using with appropriate authorizations.

Note about the training system for the ADM102 course: As stated, all services are initially inactive as of SAP Web AS 6.20. Some services, such as the */sap/public/...* branch have already been activated in the master system for the ADM102 course – it is only for this reason that it was not necessary to activate any services during the "Internet Communication Manager" lesson (in the context of the ADM102 course).

Demonstrate the "Activating ICF Services" exercise at this point. If you want to demonstrate a different example from the one in the participant exercises, we recommend that you use the service */sap/bc/bsp/sap/it00*. It offers various test cases from and for developers of BSP applications.

Another tip: The BSP application *IT00* allows you to demonstrate the ICM server cache elegantly. To do this, choose in the Web browser point 7.1 "Accessing MIME objects (images) " and/or 7.2 "Accessing SAP Icons in MIME Repository" (this takes some time the first time it is called). After this, you can demonstrate that the images are in the ICM server cache: SMICM, menu path *Goto* → *HTTP Server Cache* → *Display*. Ensure that you have the correct instance! Double clicking a *gif* entry even displays the image.

Some bright participants may ask which client is actually used for the logon in the Web browser. In the training system, the setting for *login/system_client* points to the 100 clients. However, a client (or other logon details) can be stored for each service (or for entire trees, if appropriate even everything below *default_host*) in transaction SICF.

© 2003 SAP AG. All rights reserved.

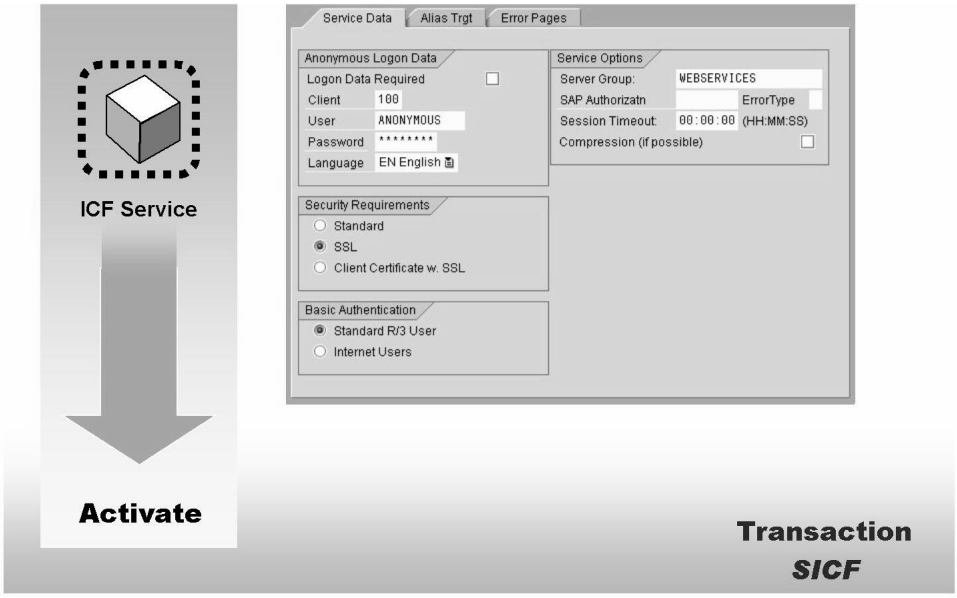


Figure 18: Properties of an ICF Service

You can maintain the following properties of ICF services in transaction SICF:

- Anonymous logon data: Allows you to store client, (service) user, password, and language

If the "Logon Data Required" flag is set, the ICF logon data overrides the data from any other logon procedures (such as Single Sign-On, User/Password in the URL, or the data specified in the "Basic Authentication" area).

- Security requirements: Whether the service requires an SSL connection or a client certificate with SSL instead of a normal logon
- Basic authentication: Whether the user name entered should be interpreted as a standard user (from transaction SU01) or as an alias name (extra field in transaction SU01)
- Server group: Logon group (from transaction SMLG) if you are using the SAP Web dispatcher
- SAP authorization: Required authorization (for authorization object *S_ICF*, field *ICF_VALUE*)
- Error type: Which message is sent if authorization errors occur
- Session timeout: After this length of time, an application is terminated (if you enter the value 0, the profile parameter *rdisp/plugin_auto_logout* applies))

© 2003 SAP AG. All rights reserved.

- Compression: The SAP system compresses the response using *gzip*, as long as the caller can perform the decompression (currently only available for Microsoft Windows and Linux)
- HTTP Handler List: List of HTTP handlers in execution sequence
- Error handling: What should happen if logon and application errors occur (response page or redirect)

Instead of original services, you can also use aliases that point to an existing service. Transaction SICF still differentiates between internal and external aliases. Internal aliases appear in the "normal" tree of all services, while external aliases allow a service to be called using any descriptive name (which can also contain the / character). An incoming URL is first checked against the external aliases, and then against the internal aliases.

Demonstrate the "Creating a new ICF Service" exercise at this point. For your information: The author of this lesson has already activated the services */sap/public/ping* and */sap/public/info* in the training master system for the ADM102 course. In the standard delivered system as SAP Web AS 6.20, these services are initially inactive (as are all ICF services initially).

If you want, you can also create an external alias (simply choose the →*External Aliases* pushbutton in transaction SICF, otherwise the procedure is exactly as described in the exercise). The external alias should have its own name, such as **`/SAPWebAS/myInfo`** (created as **one** alias).

© 2003 SAP AG. All rights reserved.

45

Exercise 3: Administrative Work with the ICF

Exercise Duration: 15 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Activate and call ICF services
- Create an ICF service

Business Example

Your company uses SAP BW 3.1 in the context of mySAP Business Intelligence. Your development department has extended the standard SAP system with a number of BSP-based applications. You are to ensure that these applications can be called.

Task 1 Activating ICF Services

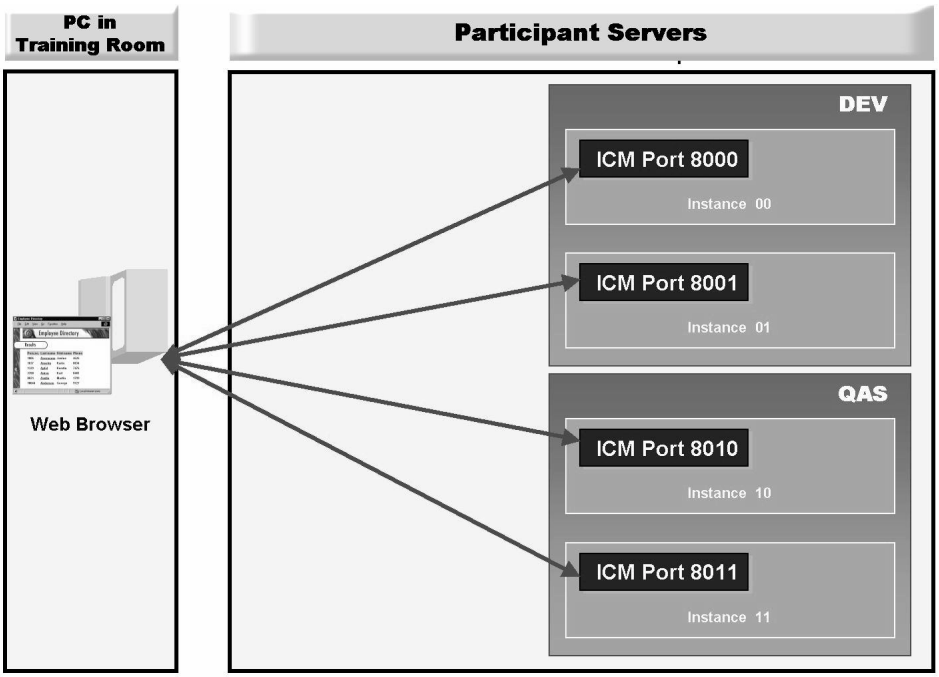


Figure 19: Complete Scenario of the Training Landscape

Activate and call services that are delivered by SAP.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. In your local Web browser (in the training room), call the following URL: **`http://<server with domain>:<ICM Port>/sap/bc/bsp/sap/tutorial_1`** (Example for the QAS group and the server twdf0042: **`http://twdf0042.wdf.sap.corp:8010/sap/bc/bsp/sap/tutorial_1`**).

   Why do you receive an error message?
2. Activate the ICF service */sap/bc/bsp*, including all subservices.
3. Call the URL **`http://<server with domain>:<ICM Port>/sap/bc/bsp/sap/tutorial_1`**.

Result

All services under */sap/bc/bsp* can now be called from the intranet and, if the network settings allow it, from the Internet.

Task 2 Creating a New ICF Services

Create a new service as an alias to an existing service.

1. Create an ICF service, */ADM102/myInfo*, that is an alias that points to */sap/public/info*.

   Remember to activate your new service.
2. Test the new service, */ADM102/myInfo*.
3. **Optional:** Determine all ICF services that point to the service */sap/public/info*.

Result

Your SAP system provides a new service, */ADM102/myInfo*.

© 2003 SAP AG. All rights reserved.

Solution 3: Administrative Work with the ICF

Task 1 Activating ICF Services

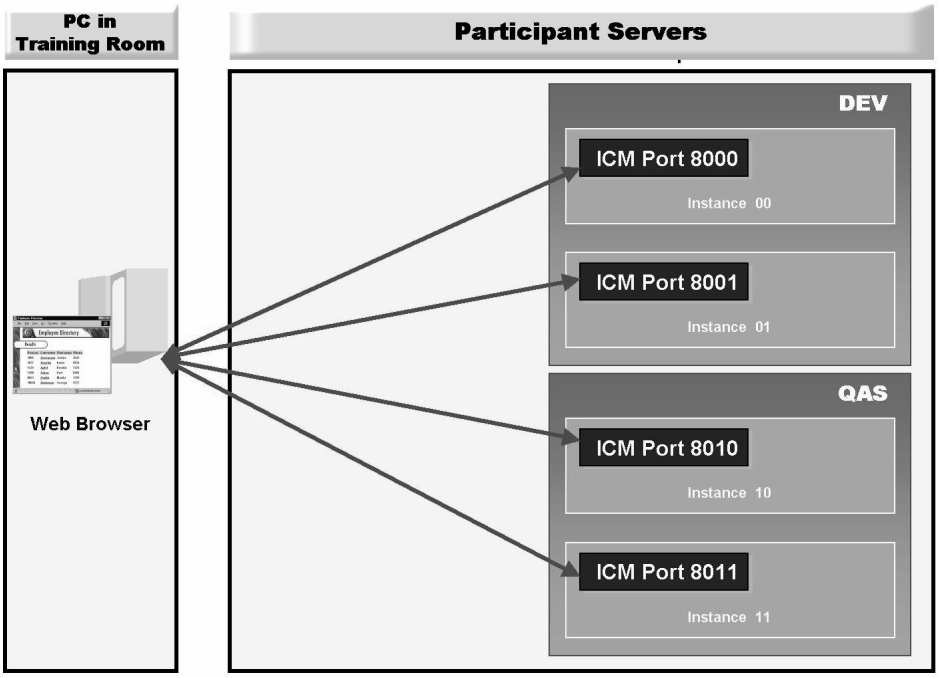


Figure 20: Complete Scenario of the Training Landscape

Activate and call services that are delivered by SAP.

1. In your local Web browser (in the training room), call the following URL: **`http://<server with domain>:<ICM Port>/sap/bc/bsp/sap/tutorial_1`** (Example for the QAS group and the server twdf0042: **`http://twdf0042.wdf.sap.corp:8010/sap/bc/bsp/sap/tutorial_1`**).

   Why do you receive an error message?

   a) Enter the specified URL in your local Web browser and choose *Enter*.
   b) You receive the error message, "You are not authorized to view this page" because the service you have called is inactive (as are all services initially delivered by SAP).

Continued on next page

 © 2003 SAP AG. All rights reserved.

2. Activate the ICF service */sap/bc/bsp*, including all subservices.
   a) In the service maintenance transaction SICF, navigate to the node */sap/bc/bsp*.
   b) Select this node and right-click to choose *Activate service* (alternatively, you can choose *Service/Virtual Host → Activate*).

      Choose the *Yes* button (tree icon) to activate the node */sap/bc/bsp* with all of its subservices.
3. Call the URL **`http://<server with domain>:<ICM Port>/sap/bc/bsp/sap/tutorial_1`**.
   a) After you log on with user **`ADM102-##`**, you receive a "Hello World" output from the SAP Web AS.

      As it may be necessary to generate objects, the first call can take some time. The system will respond quickly to subsequent requests to this service.

      Note: tutorial_1 is only the first tutorial of a complete set that provides an introduction to creating applications with BSPs. For all of the tutorials, see the documentation for the SAP Web AS 6.20 at *mySAP Technology Components → SAP Web Application Server → Web Applications (BC-MAS)→ Creating Web Applications with BSPs*.

Result

All services under */sap/bc/bsp* can now be called from the intranet and, if the network settings allow it, from the Internet.

Task 2 Creating a New ICF Services

Create a new service as an alias to an existing service.

1. Create an ICF service, */ADM102/myInfo*, that is an alias that points to */sap/public/info*.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Remember to activate your new service.

a) In the hierarchy display of transaction SICF, right-click on the topmost node, *default_host*. Choose *New Sub-Element* (or *Service/Virtual Host → Create Service*).

   Confirm the message about the namespace concept.

b) In the *Create a Service Element* window, enter **`ADM102`**, and as the *Type*, enter **`Independent Service`**.

c) In the next window, maintain only a description text. Leave all other fields unchanged and save your entries. To complete your entry, choose the *Cancel* button.

d) In the service tree, select your new node, */ADM102*, and right-click to choose *New Sub-Element*. Now create the element **`myInfo`** as a *reference to an existing service*.

e) In the *Create a Service Element* window, enter a description text.

   Navigate to the *Alias Target* tab page for the node *default_host/sap/public/info* and double-click this line.

   Save the data and close the window.

f) Select the node */ADM102/myInfo* in the service tree to activate it and right-click to choose *Activate Link*. Confirm the query by choosing *Yes*.

2. Test the new service, */ADM102/myInfo*.

   a) In the Web browser (locally in the training room), call the URL **`http://<server with domain>:<ICM port>/ADM102/myInfo`**. The browser should display an XML document containing system information.

3. **Optional:** Determine all ICF services that point to the service */sap/public/info*.

   a) In transaction SICF, navigate to the node */sap/public/info*. Choose *Service/Virtual Host → Aliased By*.

      The system displays a list that contains all services that point to */sap/public/info*. Your entry, */ADM102/myInfo*, also appears.

Result

Your SAP system provides a new service, */ADM102/myInfo*.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service

Related Information

- Online documentation for SAP Web AS 6.20: *mySAP Technology Components → SAP Web Application Server → Web Applications (BC-MAS) → Web Application Server → Internet Communication Framework*
- SAP Note 517484: *Inactive Services in the Internet Communication Framework*
- SAP course NET200: SAP Web AS: Developing Web Applications

© 2003 SAP AG. All rights reserved.

Lesson: SAP J2EE Engine: Administration

51 Lesson Duration: 60 minutes

Lesson Overview

In this lesson, you will obtain an overview of the administration of the J2EE application server (SAP J2EE Engine) that is contained in the standard SAP Web Application Server as of SAP Web AS 6.20. You will learn about basic Java terms, starting and stopping the SAP J2EE Engine, and configuration tools for the SAP J2EE Engine.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the structure of the SAP J2EE Engine
- Perform simple administrative activities for the SAP J2EE Engine

In this lesson, the participants should get to know basic terms from the Java area. The architecture and administration of SAP's J2EE server (SAP J2EE Engine) are a significant part of this lesson.

For more information about the J2EE Engine (including frequently asked questions), see the SAP Service Marketplace under the Quick Link */j2ee* (or under the Quick Link */netweaver*).

Business Example

Your company uses SAP Web AS 6.20 for J2EE applications. As a system administrator, it is your task to ensure that the SAP J2EE Engine operates correctly.

Introduction

SAP delivers a runtime environment for Java programs with SAP Web AS 6.20. The integrated Java Web application server is a Java 2 Enterprise Edition (J2EE) application server called **SAP J2EE Engine**.

J2EE is a standard defined by Sun Microsystems for developing distributed applications based on Java. The point of this specification from Sun is to ensure that Java developments can be run on Java application servers from different vendors. There are more than 30 Java Web application servers available on the market. The only differences between them are the version of the J2EE specification, how completely they support the J2EE specification, and whether they have passed the Sun J2EE compatibility test. The best-known J2EE servers are:

 © 2003 SAP AG. All rights reserved.

- SAP J2EE Engine
- Weblogic (BEA)
- Websphere (IBM)
- Sun ONE Application Server(Netscape/Sun)
- JRun (Allaire)
- Oracle 9 iAS (Oracle)
- Inprise (Inprise/Borland)

SAP Web Application Server 6.20 contains a 1.2 compliant J2EE Server.

Java is a simple programming language similar to C and C++. The Java compiler generates an architecture-neutral temporary format, known as the bytecode. A Virtual Machine interprets the bytecode. Java Virtual Machines (JVM) are available for different hardware and operating systems. Java programs are platform-independent and do not run directly in the operating system but rather in the JVM, and are therefore considered to be safe. Automatic memory management tools called garbage collectors ensure that Java developers no longer have to deal with memory management. Java is multi-thread compatible and can therefore process multiple incoming requests in parallel. J2EE provides a service, Remote Method Invocation (RMI), that can be used to call methods on remote servers (similar to RFC). Services and Application Programming Interfaces (APIs) are combined in three editions: J2SE (Java 2 Standard Edition) as a standard for developing Java applications for desktop devices; J2EE (Java 2 Enterprise Edition) as a standard for developing Java applications for servers; and J2ME (Java 2 Micro Edition) as a standard for developing Java applications for small devices such as PDAs.

Architecture of J2EE Applications

SAP implements distributed applications using the three-layer model (presentation layer, application layer, and database layer). With J2EE applications, it is possible to work with a four-layer model. In this situation, the client presentation layer is represented by the browser. On the J2EE server, application logic and the server presentation layer are separated. The database layer is represented by a database in which the application data is stored.

© 2003 SAP AG. All rights reserved.

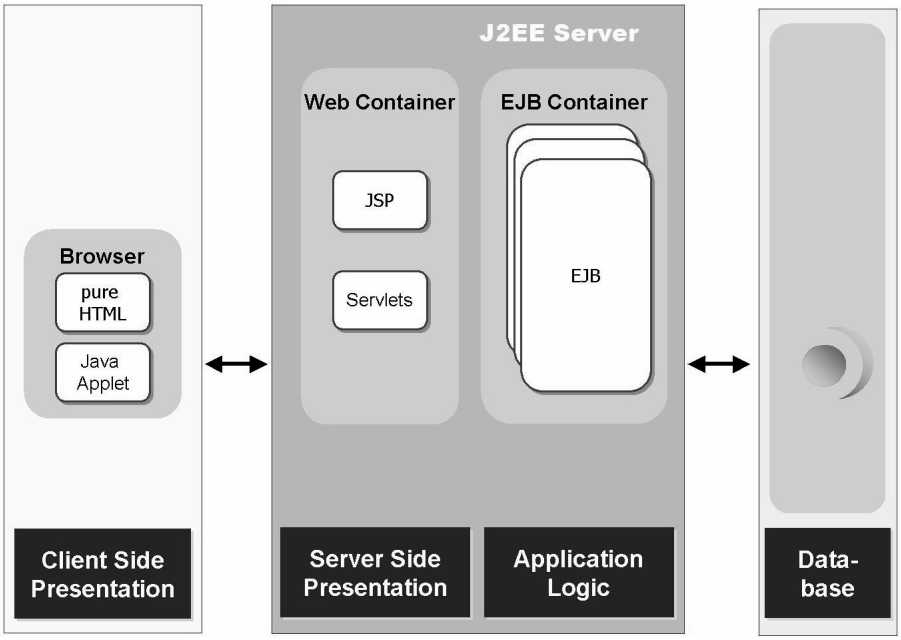


Figure 21: Architecture of J2EE Applications

SAP previously mapped the application logic (such as business processes) in the SAP system itself. If you now use a J2EE server, you can also implement the application logic on the J2EE server in Enterprise Java Beans (EJB). Enterprise Java Beans runs on the J2EE server in the EJB container. To create HTML pages or XML data, the Web container, in which Java Server Pages (JSPs) and servlets are executed, is used. Java Server Pages are HTML coding extended with Java coding. Java Server Pages are converted to servlets and then executed. Servlets are comparable with Common Gateway Interface (CGI) programs that, for example, search a text file and output the result in an HTML page. Servlets also consist of a mixture of Java and HTML coding.

 © 2003 SAP AG. All rights reserved.

Structure of the SAP J2EE Engine

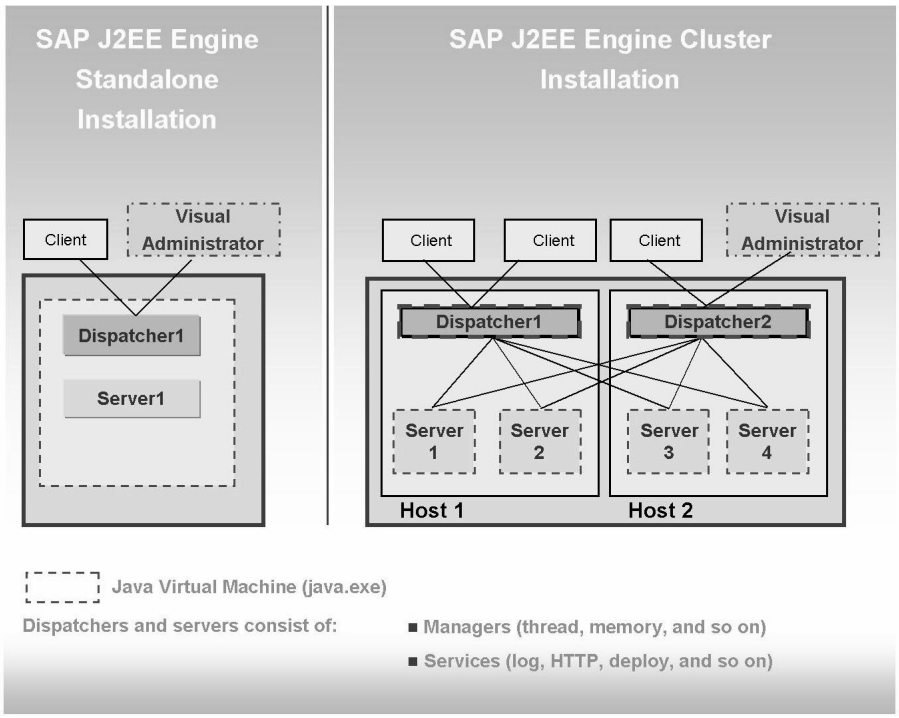


Figure 22: Structure of the SAP J2EE Engine

The SAP J2EE Engine consists of the **dispatcher node** and **server node** elements.

The main function of the **dispatcher node** is to receive requests from the client and to distribute these across the server nodes using a load balancing procedure. The dispatcher provides services for communication between dispatchers and servers, and for communication between dispatchers and clients. With load balancing, the SAP J2EE Engine distributes the client requests proportionally to the servers in a cluster. In this procedure, the server with the least workload is identified and requests are forwarded to it. The data for the utilization of the servers is collected.

The SAP J2EE Engine 6.20 load balancing system checks the loads of all servers that are running together in a cluster, and searches for the server with the least load. If a request is received from a client for a server resource (a specific service), the load balancing system selects the appropriate server within the cluster on which the service is running. This is mainly used for services that are available on multiple dispatchers and servers. These services use a function that is provided by a special interface called load balance context. The utilization of a server node is based on the common utilization status of the mangers that are running

© 2003 SAP AG. All rights reserved.

on the server (ClassLoader Manager, Cluster Manager, Log Manager, Memory Manager, Service Manager, Swap Manager, Thread Manager, and TimeOut Manager). The managers evaluate their own utilization and the load balancing system evaluates the utilization of the entire cluster. This information is updated and then newly calculated for a particular length of time using the round robin algorithm.

The **server node** covers the J2EE functions. The main function of the server node is to process client requests and to send them back to the dispatcher. Every server node has a connection to all other servers and to all dispatchers. There is no direct connection from the servers to the client.

The statement *every server node has a connection to all other servers and to all dispatchers* applies only to SAP Web Application Server 6.20. With SAP Web Application Server 6.30, every dispatcher has a connection to every server, but the servers do not communicate with each other.

Both dispatchers and servers consist of managers and services. Managers simplify the administration of the Java Virtual Machine, using thread management or memory management. Services contain the basic functions of the SAP J2EE Engine. Some services run only on the dispatcher or the server. You can use the services to administer dispatchers and servers, for example, for load balancing, to set up logging, or to make security settings.

First, the runtime environment is started with the managers, then the services (interfaces and libraries are loaded), and finally the application. The system is regarded as started once all managers and all core services are completely available. Components of the higher levels (such as services) use defined interfaces (API) to access the functions of the components of the lower levels (such as managers).

The J2EE server can be installed either as a stand-alone server or as a cluster. In the case of a stand-alone installation, there is exactly one dispatcher and one server that run together in a Java Virtual Machine. In the cluster version, there can be more than one of each node (dispatchers and servers). The total number of nodes in the cluster is limited to 64.

© 2003 SAP AG. All rights reserved.

Every cluster element in a cluster runs in a separate Java Virtual Machine. In this way, it is possible to distribute cluster elements across physically separate hosts.

Note: As a runtime environment, you require not only the Java Runtime Environment (JRE), but also functions that are provided with the Software Development Kit (SDK). The SDK also contains the JRE. For more information, see the document OS/DB/JDK releases for SAP Web AS 6.20, which is available under the alias */platforms* on the SAP Service Marketplace (*Availability of SAP components in detail → SAP Web AS/Kernel/Basis*).

The Java Runtime Environments (JREs) for dispatchers and servers are delivered on the installation CD. However, we recommend that customers always use the newest available SDK, which can be downloaded from the Internet by following the installation guide. The SDK contains additional functions that, for example, the installation tool SAPInst requires.

In the case of a SAP J2EE Engine standalone installation, there is exactly one dispatcher node and one server node, that run together in a JVM. This configuration is not scalable, but on the other hand, communication performance between the nodes is better, and only memory for **one** JVM is occupied.

© 2003 SAP AG. All rights reserved.

SAP Web AS 6.20 with SAP J2EE Engine

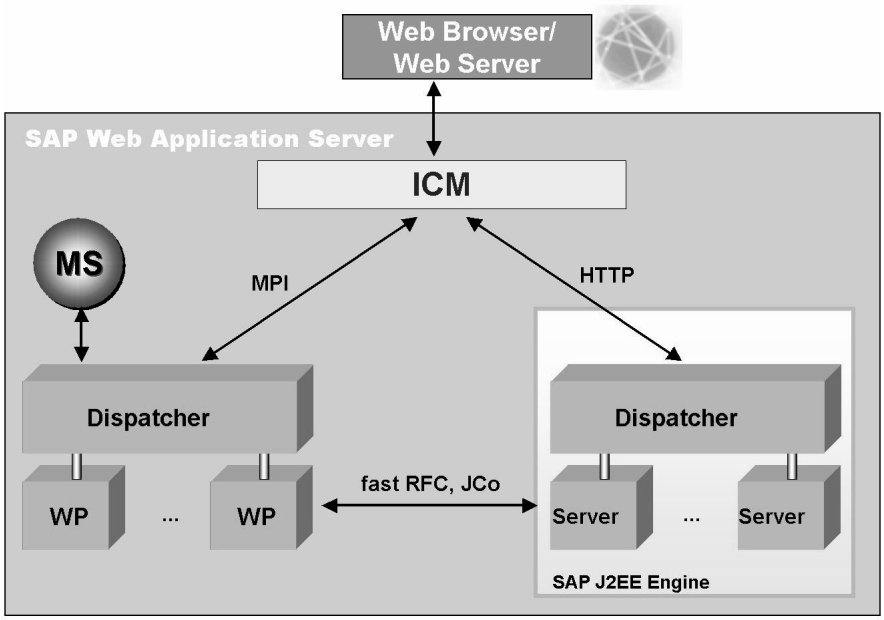


Figure 23: Architecture of SAP Web AS 6.20 with SAP J2EE Engine

The SAP J2EE Engine is an integral part of SAP Web AS 6.20 and by default is available after installation. All requests from the browser go to the Internet Communication Manager (ICM). The ICM decides whether to forward the request to the SAP J2EE Engine or to the SAP Web dispatcher.

All requests to the SAP J2EE Engine are forwarded directly from ICM to the HTTP service of the SAP J2EE Engine using HTTP. Internally, the dispatcher uses a load balancing procedure to forward requests in the SAP J2EE Engine to the appropriate server, where they are processed. The SAP J2EE Engine and SAP Web dispatcher can exchange data using the Java Connector (JCo), which is based on RFC. The data is exchanged directly between the ABAP work process and the Java process. The ABAP dispatcher decides which ABAP work process processes the request. **Fast RFC** was developed to ensure that a high level of performance can be maintained for the data exchange between the SAP J2EE Engine and the dispatcher and work processes based on the JCo.

The SAP Web dispatcher is also responsible for starting the SAP J2EE Engine in addition to the ICM and the SAP work processes.

© 2003 SAP AG. All rights reserved.

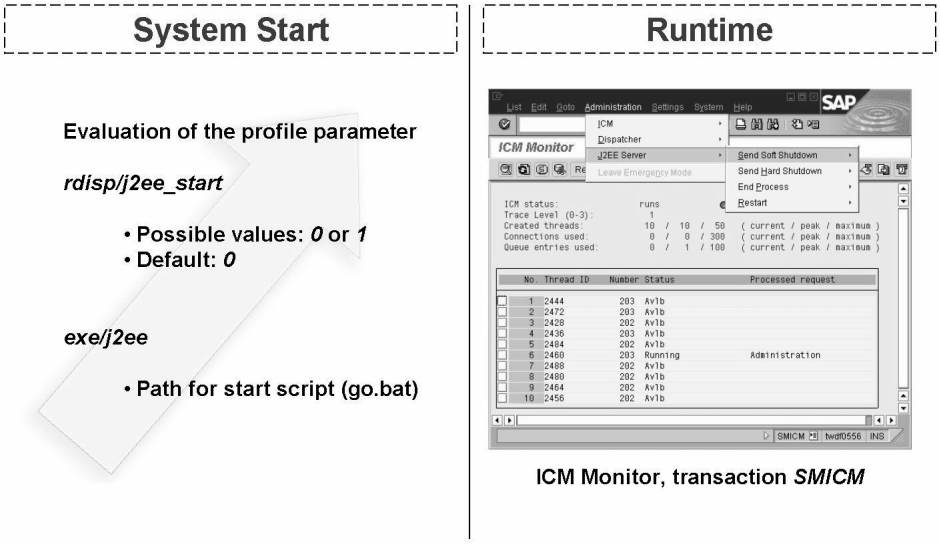


Figure 24: Starting the SAP J2EE Engine

The SAP J2EE Engine is only started automatically by the SAP Web dispatcher if the parameter *rdisp/j2ee_start* is set to 1 (default value: 0). You can use transaction RZ11 to change the value of this parameter dynamically. The parameter *exe/j2ee* specifies the directory in which the start scripts for the SAP J2EE Engine are stored (such as *\usr\sap\<SID>\DVEBMGS00\j2ee\cluster\dispatcher\go.bat*).

The J2EE server is started by the SAP system in the ICM monitor (transaction SMICM or *Tools* → *Administration* → *Monitor* → *System Monitoring* → *Internet Communication Manager*). From the initial screen of this transaction, you can use *Administration* → *J2EE Server* to send either a soft or a hard shutdown to the J2EE Engine (in each case, with or without a restart). The dispatcher does not actively terminate the connection, but rather the J2EE Engine must terminate itself. It is also possible to end the process and to set the restart flag for the SAP J2EE Engine.

You can demonstrate transaction SMICM for the participants at this point. To do this, ensure that you are logged on to the central instance. You can go access the administration functions for the J2EE Server under the menu path *Administration* → *J2EE Server*.

Other J2EE parameters:

- The parameter *icm/HTTP/j2ee_<xx>* is used to define the communication between the ICM and the J2EE Server. The following syntax applies: *icm/HTTP/j2ee_<xx> = host=<host>, Port=<port>, CONN=<number of connections>*

© 2003 SAP AG. All rights reserved.

- The parameter *rdisp/j2ee_error* is used to automatically deactivate starting a J2EE server if a certain number of attempts to start the server fail. You can reactivate the starting of the server by dynamically changing the value of the parameter to 1.
- The parameter *rdisp/j2ee_timeout* defines the timeframe within which the SAP J2EE Engines log on to the dispatcher. If this does not happen, the system assumes that the J2EE Server encountered a problem when starting and it is restarted until the setting for the parameter *rdisp/j2ee_error* takes effect.
- *icm/dpj2ee*: If the SAP J2EE Engine is not started with the SAP Dispatcher, but rather with the standalone dispatcher "dpj2ee", the ICM must be informed using this parameter, and the value of the parameter must therefore be set to *True*.

After you have configured the SAP J2EE Engine, that is, for example, you have entered the port, you can call a Java-based Web application with the browser. Of course, the application must be available on the SAP J2EE Engine. The mechanism with which the application is stored, or rather installed, on the SAP J2EE Engine, is called **deployment**.

At this point, it is useful to send a request to the SAP J2EE Engine to demonstrate that you can call Web applications using the browser. Start the browser and enter the URL: **`http://twdfxxxx.wdf.sap.corp:50000`** (for the DEV system; the port for the QAS system is 51000). This calls the documentation for the SAP J2EE Engine. This documentation is identical to the online documentation. There is an *Examples* menu entry there. You can now select and deploy one of the examples (such as PC Shop). To do this, select *PC Shop* and choose the *RUN* pushbutton. The PC Shop application is now deployed. You can find entries for the PC shop in the file system under G:\usr\sap\<SID>\D*\j2ee\cluster\server\services\servlet_jsp\work\jspTemp\shop\root\shop.jsp. You can now also call this using the browser with the following URL: http://twdfxxxx.wdf.sap.corp:50000/shop/index.jsp.

© 2003 SAP AG. All rights reserved.

Solving Problems with the SAP J2EE Engine

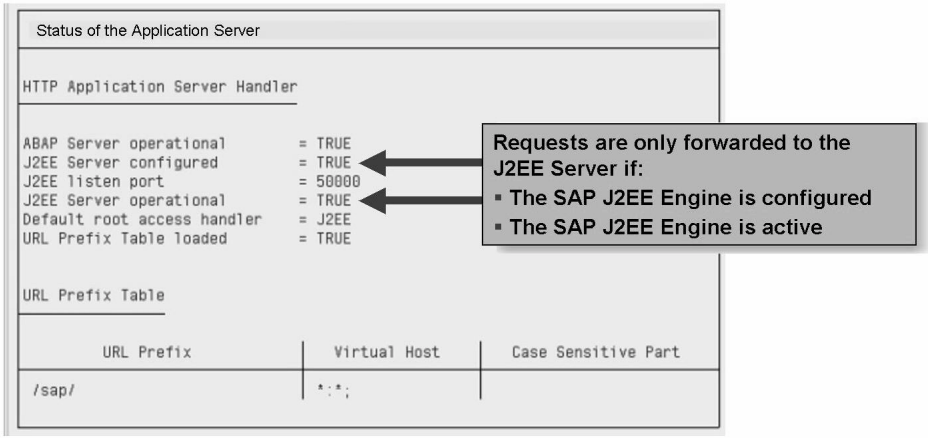


Figure 25: Identifying Problems with SMICM

You can use transaction SMICM to check the status of the J2EE Engine by choosing *Goto* → *HTTP Application Server* → *Display Data*. The following data is displayed if the SAP J2EE Engine is available:

- ABAP Server operational = TRUE
- J2EE Server configured = TRUE
- J2EE listen port = 5<instance number>00
- J2EE Server operational = TRUE
- Default root access handler = J2EE
- URL Prefix Table loaded = TRUE

If the value *false* is returned for the "J2EE Server configured" entry, then check the parameters *rdisp/j2ee_start* and *icm/HTTP/j2ee_XX*. You can use the "J2EE Server operational = TRUE/FALSE" entry to identify whether the SAP J2EE Engine is running. If the value FALSE is returned, this means that the SAP J2EE Engine is sending inactive to the SAP Web dispatcher. To correct errors, you must either check the SAP J2EE Engine itself or the developer trace file *dev_disp*. The *dev_disp* file should contain the following entries:

- DpJ2eeStart: j2ee state = STARTED
- DpJ2eeLogin: j2ee state = CONNECTED
- DpJ2eeMsgProcess: j2ee state = ACTIVE

At this point you can demonstrate for the participants how you can identify whether the SAP J2EE Engine is available. To do this, call transaction SMICM and choose *Goto* → *HTTP Application Server* → *Display Data*. Now stop the SAP J2EE Engine by choosing the following menu

© 2003 SAP AG. All rights reserved.

path, for example *Administration* → *J2EE Server* → *Send Hard Shutdown* → *With Restart* or *Send Soft Shutdown* → *with Restart*. There are now entries in the developer trace dev_disp that you can display by calling transaction AL11 and double clicking the DIR_HOME entry. If you have stopped the SAP J2EE Engine in transaction SMICM by choosing *End Process*, there is no entry in the *dev_disp* file for the stop process. If you end the SAP J2EE Engine with the *Without Restart* flag, you can also start the SAP J2EE Engine manually (*Administration* → *J2EE Server* → *Restart* → *Yes*).

Configuring the SAP J2EE Engine

You can change important SAP J2EE Engine settings for managers (memory, threads) and services by Telnet, with properties files, or using the Visual Administrator.

Methods of Configuring the J2EE Server:

- **Telnet or server console**
- **Properties files**
- **Visual Administrator**

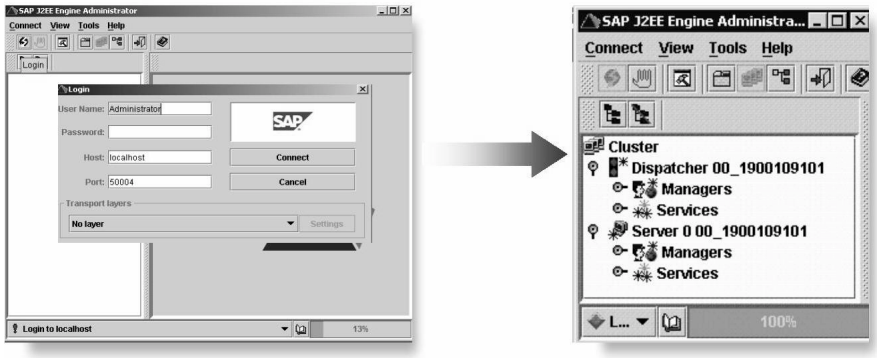


Figure 26: Configuring the SAP J2EE Engine

You can configure the SAP J2EE Engine in different ways, for example, by using properties files, directly by Telnet to the server using the server console, or with the Visual Administrator. Properties files are the lowest level of configuration, as it is files in the operating system that are actually changed. The server does not need to be available. You can use the Visual Administrator to configure and administer the running clusters and servers. You can start the administration console through a path (such as *Start* → *Programs* → *SAP J2EE Engine 6.20*→ *Instance* <*xx*> → *Administrator*) or from the file system level (for example, *\usr\sap*<*SID*>*\D*\j2ee\admin\go.bat*). You can then change the parameters for managers and servers.

© 2003 SAP AG. All rights reserved.

At this point, you can - depending on your experience with the SAP J2EE Engine - optionally give the following demonstration: Call the Visual Administrator at operating system level by choosing *Start* → *Programs* → *SAP J2EE Engine 6.20*→ *Instance* <*xx*> → *Administrator* and then choose the menu path *Connect* → *Login*. On the first screen, enter the following values:

- User Name: Administrator (= default after the installation)
- Password: leave this blank (= default after the installation)
- Host: Specify the host name (in this case, localhost, for example)
- Port: 5<instance number>04

From the list, select *No layer* for the transport layer (Java RMI/P4 is used). The two other entries have the following meanings: *HTTP Tunneling* allows communication through proxies and firewalls and *SSL* transfers the data through the Internet in an encrypted format. Now choose the *connect* pushbutton. You can now demonstrate and explain the managers described below. The Security Service, with which the connection is created to the SAP system in the case of SAP Web Application Server 6.20, is important.

- The Cluster Manager manages the communication between the nodes in the cluster. The JoinPort parameter specifies the port of the dispatcher or server that listens for connections. The OpenPort parameter specifies the port that is used for cluster element connections.
- The IpVerification Manager manages a list (*managers/ipverify/hosts.txt*) of host names with which you can allow or deny access to nodes.
- The Log Manager is used for recording events.
- The Memory Manager stores information about the amount of memory used. Most important parameters: 64 MB < *MaxMemory* < *Physical Memory*
- The Thread Manager starts and manages threads in the Java Virtual Machine. Important parameters: *MinThreadCount* < *MaxThreadCount* < *OS maximum*
- You use the HTTP service to set the port that is used for external communication.
- The DBPool service creates database connections and saves them so that they can be used again.
- The EISConnector service allows communication between the SAP J2EE Engine and the Enterprise Information Systems (EIS).
- The Monitor service provides data about the current status of the system.

© 2003 SAP AG. All rights reserved.

- The Log service defines eight types of log messages, from Emergency (0) to Debug (7)
- The Deploy service is responsible for distributing the J2EE applications to the appropriate containers.
- The Security service is responsible for managing the users and groups. This service is also used to define the RFC connection for the communication between the SAP Dispatcher and the SAP J2EE Engine. The connection data is maintained on the *R/3 Properties Manager* tab page.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

59

Exercise 4: SAP J2EE Engine: Administration

Exercise Duration: 15 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Identify from the SAP system whether the SAP J2EE Engine is active
- User SAP tools to administer the SAP J2EE Engine

Business Example

You are using the SAP J2EE Engine and want to manage it using SAP tools.

You can optionally add the additional task of logging on to the Visual Administrator to this exercise and asking the participants to call the services and managers that you have just presented again, and to familiarize themselves with the tool.

System Data

System: This depends on the course setup in use. An SAP system with SAP Web AS 6.20 is required.
Client: This depends on the course setup in use. For course ADM102: client 100
User ID: This depends on the course setup in use. A course user that has authorization to execute transaction SMICM is required. For the ADM102 course, the ADM102-## users (where ## is the group number) have the required authorizations.
Password: This depends on the user used. The ADM102-## users have the initial password init.
Set up instructions:

1. The SAP J2EE Engine must be installed and configured.
2. The user should also have authorization to execute transaction RZ11.

Task

Check in the SAP system whether the SAP J2EE Engine is active and view the log file that is written when the SAP J2EE Engine is started or stopped. Deploy a Web application and call it using the browser.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. Check whether the SAP J2EE Engine is active on the central instance of your system.
2. Restart the SAP J2EE Engine using a hard or a soft shutdown. Verify that the restart was successfully performed.
3. Start a browser and enter the following URL: http://twd-fxxxx.wdf.sap.corp:5XX00, where twdfxxxx refers to the host on which the central instance of your training system is running and XX is the instance number of the central instance of your SAP system (XX=00 for DEV groups, XX=10 for QAS groups). Under *Examples*, choose the *PC Shop* example and deploy the application. Then start the application again using the browser, for example by entering http://twdfxxxx.wdf.sap.corp:5XX00/shop. (XX = instance number).

© 2003 SAP AG. All rights reserved.

Solution 4: SAP J2EE Engine: Administration

Task

Check in the SAP system whether the SAP J2EE Engine is active and view the log file that is written when the SAP J2EE Engine is started or stopped. Deploy a Web application and call it using the browser.

1. Check whether the SAP J2EE Engine is active on the central instance of your system.

   a) Call transaction SMICM and choose *Goto* → *HTTP Application Server* → *Display Data*. If the results "J2EE Server configured = TRUE" and "J2EE Server operational = TRUE" are returned, the SAP J2EE Engine is available.

2. Restart the SAP J2EE Engine using a hard or a soft shutdown. Verify that the restart was successfully performed.

   a) Choose *Administration* → *J2EE Server* → *Send Hard Shutdown* → *With Restart* or *Send Soft Shutdown* → *With Restart* to stop the SAP J2EE Engine.

   b) You can open the developer trace file *dev_disp* by calling transaction AL11 and double-clicking the *DIR_HOME* entry. Search the file for entries that look like this:

      - DpIJ2eeShutdown: HARDSHUTDOWN j2ee state = SHUTDOWN
      - DpJ2eeStart: j2ee state = STARTED
      - DpJ2eeLogin: j2ee state = CONNECTED
      - J2EE server info
        - start = TRUE
        - state = ACTIVE

   Hint: If you have stopped the SAP J2EE Engine in transaction SMICM by choosing *End Process*, there is no entry in the *dev_disp* file for the stop process. If you end the SAP J2EE Engine with the *Without Restart* flag, you can also start the SAP J2EE Engine manually (*Administration* → *J2EE Server* → *Restart* → *Yes*).

3. Start a browser and enter the following URL: http://twd-fxxxx.wdf.sap.corp:5XX00, where twdfxxxx refers to the host on which the central instance of your training system is running and XX

Continued on next page

 © 2003 SAP AG. All rights reserved.

is the instance number of the central instance of your SAP system (XX=00 for DEV groups, XX=10 for QAS groups). Under *Examples*, choose the *PC Shop* example and deploy the application. Then start the application again using the browser, for example by entering http://twdfxxxx.wdf.sap.corp:5XX00/shop. (XX = instance number).

a) On your PC, start a browser and enter http://twd-fxxxx.wdf.sap.corp:5XX00 (XX = the instance number of your central instance, twdfxxxx is the name of the host on which this instance is running). The system displays the documentation for the SAP J2EE Engine. Go to the *Examples* menu entry and select *PC Shop* there. There is now a *Run* button on the right of the screen. Choose this button, and on the next screen, choose the *Deploy & Run* button.

b) Restart a browser and enter http://twd-fxxxx.wdf.sap.corp:5XX00/shop (XX = instance number). You have now called your PC Shop Web application directly using the browser.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the structure of the SAP J2EE Engine
- Perform simple administrative activities for the SAP J2EE Engine

Related Information

For more information about the administration and configuration of the J2EE Engine, see the the J2EE Engine section of the online documentation (*SAP Web Application Server → J2EE Engine*) and the SAP Service Marketplace, under the */j2ee* Quick Link.

© 2003 SAP AG. All rights reserved.

Lesson: SAP Web Dispatcher

63 Lesson Duration: 40 minutes

Lesson Overview

This lesson highlights what the SAP Web dispatcher is used for and how to operate it.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the function of the SAP Web dispatcher
- Explain how you can use the SAP Web dispatcher to distribute workload across the different instances of an SAP system

---

The SAP Web Dispatcher was first delivered with SAP Web AS 6.20 (although it can also be used with SAP Web AS 6.10 systems). Its configuration and operation are simple.

It is important here to explain what the SAP Web Dispatcher achieves. The customer receives here (without extra costs) a product that is designed for a throughput up to around 3000 hits per second (HTTP). The SAP Web Dispatcher handles SAP-specific load distribution and is sufficient for the vast majority of cases. In cases of extremely high throughput, a customer can, of course, purchase hardware Web switches from third-party vendors. This often offer additional features such as hardware SSL acceleration. SAP has made the specifications for the load distribution public; there is close cooperation with the vendor f5 Networks (www.f5.com) in this area.

Make it clear that the SAP Web Dispatcher is a small program (closely related to the ICM) – the fears of some customers that they have to create an additional server with a database in the DMZ are therefore entirely unfounded.

Business Example

Your company offers its customers browser-based access to data from SAP systems. The Web applications are technically implemented as BSPs and J2EE applications. As a very large number of your customers use this service, your SAP system, which is connected to the Internet, has multiple instances. It is your task as a member of the system administration team to implement load distribution across the different instances of this SAP system.

© 2003 SAP AG. All rights reserved.

Implementation Area of the SAP Web Dispatcher

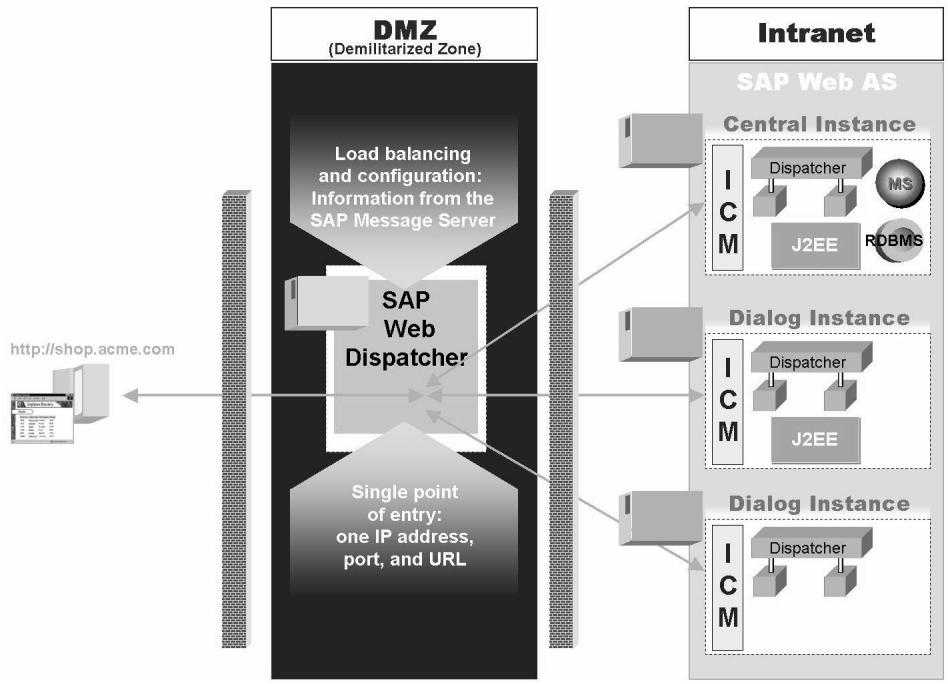


Figure 27: An overview of the SAP Web Dispatcher

You should always consider using the SAP Web dispatcher if you are operating Web applications that are technically implemented as applications for the ICM (for example, BSPs) or for the SAP J2EE Engine on an SAP system.

Some of the requirements that motivated the development of the SAP Web dispatcher:

- The Web applications are also to be used from the Internet. The company network is protected by a Demilitarized Zone (DMZ) and the critical business processes run on servers that are not known in the Internet. How can you avoid the need to place an SAP application server within the DMZ?
- The addressed SAP system consists of multiple application servers (instances) that are distributed across multiple hosts. However, the Web applications provided should run under a descriptive address; technical details such as server name and port number are to remain hidden to users.
- What form can a sensible load distribution take? It may be the case that not all of the application servers provide all services (ICM or SAP J2EE Engine).

 © 2003 SAP AG. All rights reserved.

Requirements such as these can be implemented using third-party products known as Web switches. Although there are advantages such as high throughput and implementation in close proximity to the hardware, these must be offset against the disadvantages of additional costs and the restricted SAP integration.

The SAP Web dispatcher, delivered as of SAP Web AS 6.20, acts like a software Web switch. It is a stand-alone program that you can run on a separate host without any additional software. In this way, the SAP Web dispatcher implements a central entry point for HTTP(S) requests to an SAP system, including load distribution across multiple instances.

Hint: The SAP Web dispatcher is only relevant in the Web environment. For access using an SAP GUI, the message server performs the load distribution.

Server Selection and Load Balancing

The SAP Web dispatcher ultimately forwards an HTTP(S) request to a specific application server. This section is intended to outline the criteria by which this is performed.

The two images really belong “in each other”; however, to avoid making the graphic too complicated, two graphics have been created. The second image is, as it were, a magnifying glass on the evaluation of the “SAP prefix”. For more detailed information about the background, see the online documentation. We recommend that your preparation for this lesson includes reading this documentation. A number of topics are deliberately omitted here.

© 2003 SAP AG. All rights reserved.

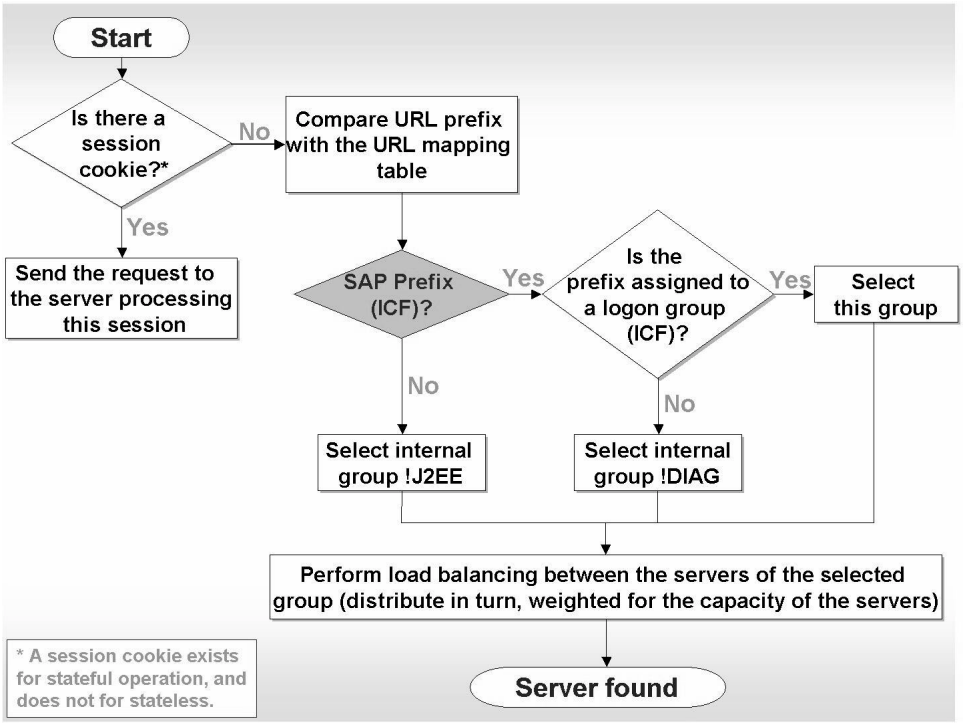


Figure 28: From the HTTP(S) Request to the Application Server

The SAP Web dispatcher first uses a cookie to identify whether the request concerns a stateful application. If this is the case, the decision is simple. The request is sent to the application server that is processing this session.

Otherwise, the SAP Web dispatcher must determine whether this is a request for the ICF ("abap") or the SAP J2EE Engine ("j2ee"). It is not possible to tell from a URL such as *http://www.myshop.com/cars/index.html* whether the request is for the ICF or the SAP J2EE Engine.

© 2003 SAP AG. All rights reserved.

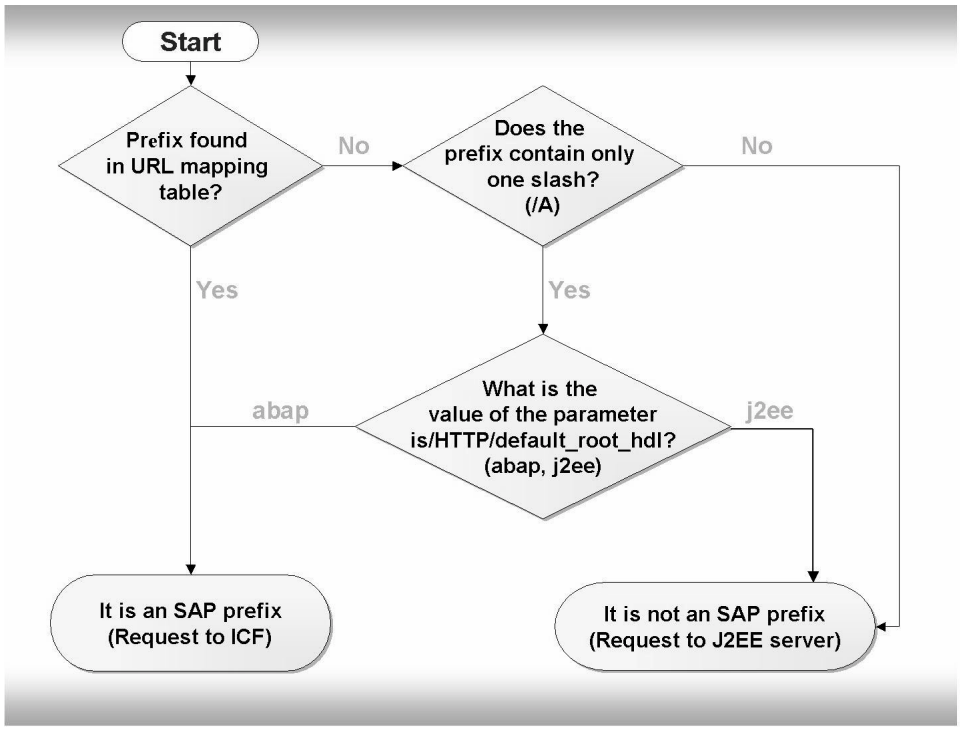


Figure 29: Analysis of the URL

If there is at least one SAP J2EE Engine active in the SAP system, the prefix (in the example, */cars*) is compared to the prefixes known in the ICF. If the prefix is known, the request is an ABAP request. If the URL contains only **one** "/" character after the host name, special handling is required. The value of the profile parameter *is/HTTP/default_root_hdl* determines the destination in this case.

You can display the list of URL prefixes simply by calling transaction SMICM and choosing the path*Goto→ HTTP Application Server → Display Data*.

In the case of a J2EE request, the request is sent to the (internally defined) !J2EE server group, which contains all application servers with SAP J2EE Engines. If the request is an ABAP request, the !DIAG group is selected. This group consists of all application servers with dialog work processes. A logon group is only used if this logon group (maintained in transaction SMLG) is explicitly specified in the ICF service.

© 2003 SAP AG. All rights reserved.

If you are teaching the ADM102 course, you can provide a reminder for the lesson about the Internet Communication Framework by double clicking any service in transaction SICFand pointing out the *Server Group* field on the *Service Data* tab page.

The following section describes the default behavior of the SAP Web Dispatcher for the load balancing procedure. You can use the profile parameter *wdisp/load_balancing_strategy* (described in the online documentation for SAP Web AS 6.30, but already valid for the current SAP Web Dispatcher 6.20) to control the exact behavior of the procedure.

The SAP Web dispatcher distributes the requests in turn within the selected server group, weighted according to the capacity of the individual servers. The capacity is calculated in the case of an ABAP request from the number of dialog work processes and, in the case of a J2EE request, from the number of server processes.

The SAP Web dispatcher collects the information about the server groups from the message server. The HTTP interface of the message server allow you to display the information with a Web browser. To do this, call the URL *http://<message server with domain>:<message server port>/msgserver/commands*.

This service is not official and therefore not documented. Demonstrate it nevertheless! Example URL for the DEV group and the server twdf0044: **`http://twdf0044.wdf.sap.corp:8100/msgserver/commands`**. Note that you should specify the **HTTP port of the message server** (profile parameter *ms/http_port*) here.

You can now call up various data in the list, in each case in HTML, XML, and text format. The SAP Web Dispatcher also collects this data from the message server – which explains why details about the message server appear in the profile for the SAP Web Dispatcher.

Operating the SAP Web Dispatcher

The internal structure of the SAP Web dispatcher is based on the ICM process. A profile file is also used in this case to determine the settings with which the SAP Web dispatcher is started. You can easily copy and use the executable file (sapwebdisp.exe) that SAP makes available for all supported operating systems to a separate host, together with the profile.

© 2003 SAP AG. All rights reserved.

For templates for the profile and parameter descriptions, see the online documentation. The SAP Web dispatcher essentially only needs to know the port at which it should receive HTTP(S) requests (parameter *icm/server_port_<x>*) and on which host (*rdisp/mshost*) and with which HTTP port (*ms/http_port*) it can access the message server.

You start the SAP Web dispatcher with the operating system command **sapwebdisp pf=<profile file>**, where you can set additional options such as trace file and trace level (see the online documentation).

Hint: In Microsoft Windows, you can set up the SAP Web dispatcher as a service with the command **`ntscmgr install sapwebdisp -b <program path>\sapwebdisp.exe -p "service pf=<profile file> <options>"`**.

Demonstrate the exercises "Configuring and Starting the SAP Web Dispatcher" and "Using the SAP Web Dispatcher" at this point.

You stop the SAP Web dispatcher using a **kill** command at operating system level. To do this, you require its process ID (PID), which you can identify from the output when it is started or from the trace file (by default *dev_sapwebdisp*). Under Unix, the command is **`kill -2 <PID>`**; under Microsoft Windows, it is **`sapntkill -INT <PID>`**.

Demonstrate the "Stopping the SAP Web Dispatcher" exercise at this point.

Hint: You can also use *icmon* to generate load for the SAP Web Dispatcher...

The following are topics that are connected to the SAP Web Dispatcher, but which have not been included in this lesson:

- End-to-end SSL and SSL termination
- URL filtering (with a configuration file similar to SAProuter)
- High availability (important, consider an Internet service)
- HTTP load distribution by the message server (possible, but not recommended)

Finally, another outlook: The SAP Web Dispatcher is, of course, being further developed. As it is fully backward compatible, customers can always use the newest available SAP Web Dispatcher. The developers are currently working on a Web interface for the ICM and the SAP Web

© 2003 SAP AG. All rights reserved.

Dispatcher. You can already monitor the ICM in transaction RZ20, and it may also be possible sometime to monitor SAP Web Dispatcher using this transaction...

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

69

Exercise 5: Administration of the SAP Web Dispatcher

Exercise Duration: 10 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Start and stop the SAP Web dispatcher
- Use and monitor the SAP Web dispatcher

Business Example

Your company is planning to go live with an extensive Web application based on SAP Web AS. Due to the expected load and as a fail-safe, you are planning to use multiple application servers on different hosts. You are responsible for ensuring that the applications can be called using a descriptive URL.

System Data

System: An SAP system with SAP Web AS 6.20 or higher is required.
Client: This depends on the course setup in use.
User ID: This depends on the course setup in use.
Password: This depends on the course setup in use.
Set up instructions: There must be a prepared profile file for the SAP Web Dispatcher at operating system level of the training system (in directory G:\Additional_Files\ADM102\SAP Web Dispatcher). The course participants must be able to access this file.

© 2003 SAP AG. All rights reserved.

Task 1 Configuring and Starting the SAP Web Dispatcher

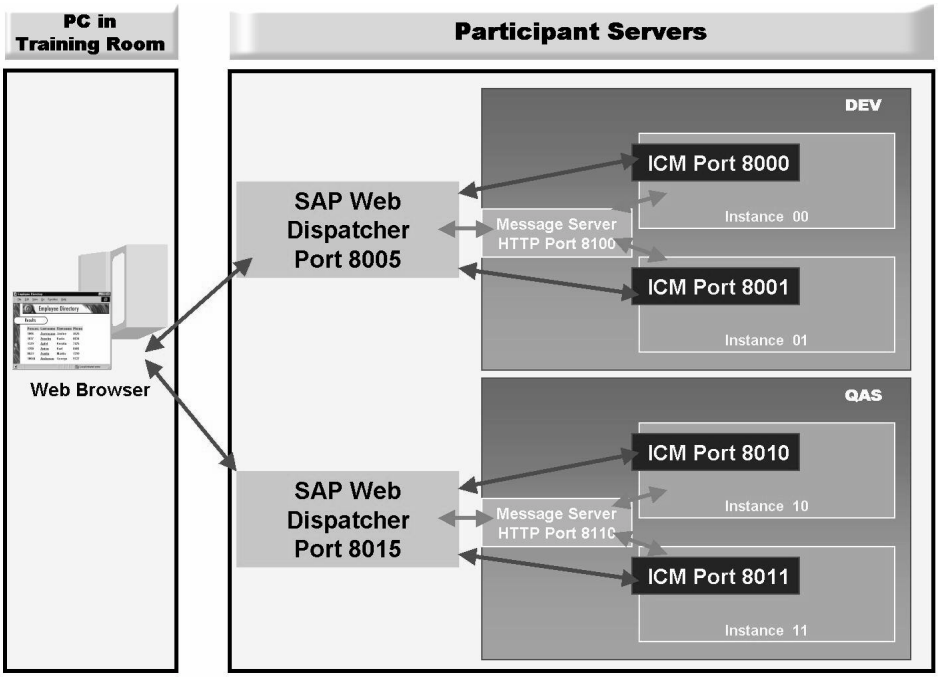


Figure 30: Complete Scenario of the Training Landscape

Determine the release of the SAP Web dispatcher and then start it with the prepared configuration file.

Note: In practice, the SAP Web dispatcher would run on a separate host (possibly with Internet access) under a descriptive name. On the training systems for the ADM102 course, the SAP Web dispatcher and the SAP system run with two instance on one host.

1. Which release of the SAP Web dispatcher is used on the training system?
2. There is a prepared profile file for each group on the servers of the training systems in the directory *G:\Additional_Files\ADM102\SAP Web Dispatcher*.

   Use this file to answer the following questions for your SAP system. At which port will your SAP Web dispatcher receive requests? To which SAP system are these requests forwarded?
3. Start the SAP Web dispatcher with the prepared profile file. The trace should be written to a file, *dev_sapwebdisp*<##>, where ## is your group number.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Result

The SAP Web dispatcher for your SAP system is now running.

Task 2 Using the SAP Web Dispatcher

Send requests that are distributed using the SAP Web dispatcher.

1. Call the ICF service */sap/public/info* using your SAP Web dispatcher.
2. Call the SAP J2EE service */shop* using your SAP Web dispatcher.

Result

You have learned about the SAP Web dispatcher's mode of operation.

Task 3 Stopping the SAP Web Dispatcher

Stop the SAP Web dispatcher using operating system tools.

1. Stop your SAP Web dispatcher.

Result

The SAP Web dispatcher has been stopped correctly.

© 2003 SAP AG. All rights reserved.

Solution 5: Administration of the SAP Web Dispatcher

Task 1 Configuring and Starting the SAP Web Dispatcher

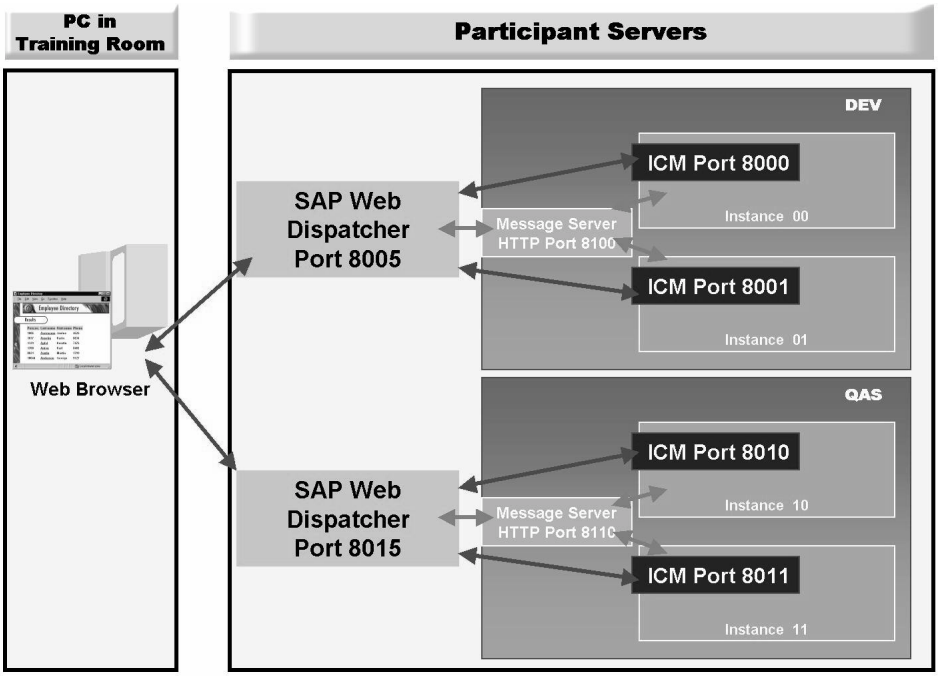


Figure 31: Complete Scenario of the Training Landscape

Determine the release of the SAP Web dispatcher and then start it with the prepared configuration file.

Note: In practice, the SAP Web dispatcher would run on a separate host (possibly with Internet access) under a descriptive name. On the training systems for the ADM102 course, the SAP Web dispatcher and the SAP system run with two instance on one host.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. Which release of the SAP Web dispatcher is used on the training system?

   a) If you have not already done so, log on to the Terminal Server Client (also known as the RDP Client) at operating system level using the user **`<sid>adm`**.
   b) Open the command prompt (under Microsoft Windows: *Start* → *Run...* and then enter **`cmd`**) and run the command **`sapwebdisp -V`**.

      The system displays output, in which the information that you require appears at the start (patch number). At the end of the list, all problems that are solved with the current release status are listed (with associated SAP Notes).

2. There is a prepared profile file for each group on the servers of the training systems in the directory *G:\Additional_Files\ADM102\SAP Web Dispatcher*.

   Use this file to answer the following questions for your SAP system. At which port will your SAP Web dispatcher receive requests? To which SAP system are these requests forwarded?

   a) Navigate to the specified directory with Windows Explorer. Open "your" profile file, for example with Notepad.
   b) The following parameters answer the questions:

      - *icm/server_port_<x>*: Specifies the port at which the SAP Web dispatcher listens for each protocol (for example, *PROT=HTTP*)
      - *rdisp/mshost*: Name of the host on which the message server is addressed
      - *ms/http_port*: Port at which the message server is queried using HTTP

      The last two parameters specify the addressed SAP system uniquely.

Hint: The *SAPSYSTEM* parameter is used to create shared memory areas and must be differentiated from all other SAP instances on the host of the SAP Web dispatcher.

Continued on next page

© 2003 SAP AG. All rights reserved.

3. Start the SAP Web dispatcher with the prepared profile file. The trace should be written to a file, *dev_sapwebdisp*<##>, where ## is your group number.

   a) Run the command **`sapwebdisp pf=<profile file> -f dev_sapwebdisp<##>`** in the command prompt. To do this, you must either specify the profile file with a complete path name, or call the command from the directory *G:\Additional_Files\ADM102\SAP Web Dispatcher*.

   The SAP Web dispatcher outputs a message saying that it has successfully started.

Result

The SAP Web dispatcher for your SAP system is now running.

Task 2 Using the SAP Web Dispatcher

Send requests that are distributed using the SAP Web dispatcher.

1. Call the ICF service */sap/public/info* using your SAP Web dispatcher.

   a) In your local Web browser (in the training room), call the following URL: http://<server with domain>:<SAP Web Dispatcher Port>/sap/public/info (Example for the QAS group and the server twdf0042: http://twdf0042.wdf.sap.corp:8015/sap/public/info).

   b) If you choose *Refresh* a number of times in the Web browser, you can observe that both instances are used for processing the requests.

Hint: The distribution of the requests is based on the number of dialog work processes configured for each instance.

Continued on next page

© 2003 SAP AG. All rights reserved.

2. Call the SAP J2EE service */shop* using your SAP Web dispatcher.

   a) In your local Web browser (in the training room), call the following URL: http://<server with domain>:<SAP Web Dispatcher port>/shop (Example for the QAS group and the server twdf0042: http://twdf0042.wdf.sap.corp:8015/shop).

   Note: The SAP Web dispatcher analyzes the HTTP request to find an appropriate target server. As the SAP J2EE Engine is only configured for **one** instance in the training room, no distribution across multiple application servers is performed for these requests.

Result

You have learned about the SAP Web dispatcher's mode of operation.

Task 3 Stopping the SAP Web Dispatcher

Stop the SAP Web dispatcher using operating system tools.

1. Stop your SAP Web dispatcher.

   a) Determine the process ID (PID) of your SAP Web dispatcher at operating system level of your server.

   You can find this number, for example, from the output when starting the SAP Web dispatcher or from the trace file *dev_sapwebdisp*<##>.

   b) Open another command prompt (in Microsoft Windows, choose *Start* → *Run...* and enter **cmd**).

   c) Run the command **sapntkill -INT <PID>** with the process ID that you have just identified.

Result

The SAP Web dispatcher has been stopped correctly.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the function of the SAP Web dispatcher
- Explain how you can use the SAP Web dispatcher to distribute workload across the different instances of an SAP system

Related Information

- Online documentation for SAP Web AS 6.20: *mySAP Technology Components → SAP Web Application Server → Web Applications (BC-MAS)→ Web Application Server→ Architecture of the SAP Web Application Server→ SAP Web Dispatcher*
- SAP Note 538405: *Composite SAP Note about SAP Web Dispatcher*
- SAP Note 552286: *Troubleshooting for the SAP Web Dispatcher*
- SAP Note 499327: *Using the SAP Web Dispatcher with SAP Web AS 6.10*

© 2003 SAP AG. All rights reserved.

Unit Summary

You should now be able to:

- Describe the options that SAP provides for intranet and Internet scenarios
- Describe the implementation areas of SAP ITS, ICM, and the SAP J2EE Engine
- Describe the architecture of the SAP ITS
- Start and stop the SAP ITS
- Perform simple administrative tasks on an SAP ITS
- Describe the implementation area of the ICM
- Configure and monitor the ICM
- Explain the importance of the Internet Communication Framework (ICF) for handling HTTP requests in the SAP system
- Outline the interaction model
- Describe what constitutes an ICF service
- Outline the structure of the SAP J2EE Engine
- Perform simple administrative activities for the SAP J2EE Engine
- Outline the function of the SAP Web dispatcher
- Explain how you can use the SAP Web dispatcher to distribute workload across the different instances of an SAP system

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

77

Test Your Knowledge

1. Which of the following technology components can be used together with an SAP R/3 Enterprise system?

   Choose the correct answer(s).

   ☐ A SAP Internet Transaction Server (SAP ITS)
   ☐ B Internet Communication Manager (ICM)
   ☐ C SAP J2EE Engine

2. Which tools can you use to administer an SAP ITS?

   Choose the correct answer(s).

   ☐ A Commands at operating system level
   ☐ B Transaction SITS in the SAP system
   ☐ C ITS Administration tool
   ☐ D Microsoft Management Console with the SAP ITS Snap-In

3. Which statement is correct?

   Choose the correct answer(s).

   ☐ A The ICM is implemented as a thread and is available for a large number of operating systems.
   ☐ B You can use an instance profile parameter to configure how many ICMs are started for each dispatcher.
   ☐ C SAP recommends that you operate a separate ICM for each client in an SAP system.
   ☐ D None of the above answers is correct.

4. The ICM is relevant only for applications based on BSPs.

   Determine whether this statement is true or false.

   ☐ True
   ☐ False

5. In which trace/log file can you identify whether the SAP J2EE Engine was correctly started?

   Choose the correct answer(s).

   ☐ A dev_w0
   ☐ B dev_disp
   ☐ C dev_icm
   ☐ D sapstart.log

© 2003 SAP AG. All rights reserved.

6. What are the names of the significant components that make up the SAP J2EE Engine?

   Choose the correct answer(s).

   ☐ A Clusters
   ☐ B Dispatchers
   ☐ C Servers
   ☐ D Work Processes
   ☐ E ICM

7. Which statement(s) is/are correct?

   Choose the correct answer(s).

   ☐ A The SAP Web dispatcher stores system information in a small local database.
   ☐ B The SAP Web dispatcher knows the capacity of all application servers in the SAP system.
   ☐ C The SAP Web dispatcher makes a firewall superfluous.
   ☐ D The SAP Web dispatcher communicates with the message server.

© 2003 SAP AG. All rights reserved.

79

Answers

1. Which of the following technology components can be used together with an SAP R/3 Enterprise system?

   Answer: A, B, C

   All of the above. The SAP ITS can be used with any SAP system (as of SAP Basis 3.1G). The technical foundation of SAP R/3 Enterprise is SAP Web AS 6.20 - for which the ICM and SAP J2EE Engine can be configured.

2. Which tools can you use to administer an SAP ITS?

   Answer: A, C

   There are various programs at operating system level after an SAP ITS installation that you can use to administer an SAP ITS. The browser-based ITS Administration tool, with which you can also monitor remote SAP ITS instances, is easier to use.

3. Which statement is correct?

   Answer: D

   All of the answers provided are incorrect. The ICM is a process (which internally consists of multiple threads) that is available for all operating systems supported by SAP. You can use a profile parameter to configure either no ICM or one ICM for each SAP instance (and therefore for each dispatcher). Scaling is performed using the settings for the number of threads and memory allocation.

4. The ICM is relevant only for applications based on BSPs.

   Answer: False

   Executing BSP applications requires the ICM. However, there are also other applications (Web services such as SOAP/XML or Web reporting with the SAP BW), that use the ICM and ICF but do not use BSPs). Thanks to the ICM and ICF, the SAP Web AS can also perform the role of Web client and can process other protocols such as SMTP in addition to HTML.

© 2003 SAP AG. All rights reserved.

5. In which trace/log file can you identify whether the SAP J2EE Engine was correctly started?

   Answer: B

   The file in which the system logs whether the SAP J2EE Engine has been started is *dev_disp*. The dispatcher starts the SAP J2EE Engine. Therefore the information is stored in the *dev_disp* file. *sapstart.log* is the log file for the start process of an SAP system. *dev_w0* is the trace file for a dialog work process, and *dev_icm* is the trace file for the Internet Communication Manager (ICM) process.

6. What are the names of the significant components that make up the SAP J2EE Engine?

   Answer: B, C

   The SAP J2EE Engine consists of one or more dispatchers and servers. Work processes are only used in the "standard" SAP environment, and the ICM is the new process in SAP Web AS 6.10. A cluster is simply a term for whether you are working with multiple servers and dispatchers with the SAP J2EE Engine.

7. Which statement(s) is/are correct?

   Answer: B, D

   The SAP Web dispatcher communicates with the message server by HTTP(S) to obtain information about the available application servers (which includes their capacity). Although the SAP Web dispatcher can act as a URL filter, it does not replace the function of a firewall. Only requests that are addressed to the configured port of the SAP Web dispatcher are processed.

© 2003 SAP AG. All rights reserved.

Unit 2

81

Globalization

This unit highlights various aspects of globalization. A number of topics are dealt with only briefly (see the individual lessons). If participants ask questions, you can go into more detail, depending on your level of knowledge. However, you should not become over-involved in details.

The time scheduling for the three lessons in the unit is very important. As the exercise task for the second lesson runs for a very long time, you must ensure that the participants start the first part of the exercise before the end of the first day of the training course. The exercise task involves importing an additional language into the SAP system. Three steps are required to do this (see the exercises and solutions). The first step has a runtime of around two hours, the second and third steps each run for around three-quarters of an hour. The runtime depends on the hardware of the training server and especially on whether the DEV and QAS groups run the exercise at the same time. The times listed above assume that the groups will be running the exercise tasks at the same time. The exercise will therefore only run for around half this time on the instructor's server. The first step must therefore run overnight, and the second and third steps during the course of the rest of the unit the next morning. We therefore recommend that you run the exercise as close to the beginning of the second lesson as possible. You can discuss the lesson itself in more detail the next morning (while the second step of the exercise is running).

Unit Overview

There are a number of special considerations that arise in connection with using software for companies that have subsidiaries in different countries and whose employees speak different languages. In this unit, a number of selected issues connected to globalization are explored. For example, we will discuss using country versions, how time zones function, and the effects of multiple languages in one SAP system. We will also explore the utilization of code pages and Unicode in SAP Web AS 6.20.

© 2003 SAP AG. All rights reserved.

Unit Objectives

After completing this unit, you will be able to:

- Recognize some effects of a centralized or decentralized implementation, the possibility of using multiple time zones in one SAP system, and the need for country versions
- Outline the process required to install an additional language in the SAP system
- Import an additional language into the SAP system
- Define terms such as code page, MDMP, and Unicode, and explain their relationships
- List advantages and disadvantages of using Unicode on the SAP Web Application Server

Unit Contents

Lesson: Aspects of Globalization 101
Lesson: Multiple Languages in an SAP System 117
Exercise 6: Importing an Additional Language into the SAP System 133
Lesson: Code Pages and Unicode: Fundamentals 139

© 2003 SAP AG. All rights reserved.

Lesson: Aspects of Globalization

82

Lesson Duration: 40 minutes

Lesson Overview

This lesson examines various aspects of globalization, such as implementation strategies, country versions, and time zones. You will learn about centralized and decentralized implementation. Country versions take into account country-specific considerations in business processes. It may be necessary to use different time zones with cross-time-zone business processes.

Lesson Objectives

After completing this lesson, you will be able to:

- Recognize some effects of a centralized or decentralized implementation, the possibility of using multiple time zones in one SAP system, and the need for country versions

---

This lesson is primarily to explain the concepts behind the material, rather than demonstrations in the system. The participants do not need to be able to import and customize additional country versions, nor to be able to set up time zones. The participants should simply gain a basic understanding. The topics presented here also belong less to the tasks of an administrator. Rather, they are basic organizational issues to be considered when implementing an SAP system.

A useful demonstration in the system is, for example, to check whether non-standard country versions have imported as add-ons (transaction SAINT), how the default time zone has been set up (transaction STZAC, or transaction SPRO *Goto → SAP Reference IMG → General Settings → Time Zones → Maintain System Settings*), and how you can set up time zones in relation to users (transaction SU01).

Business Example

The company with which you are employed has subsidiaries worldwide. As a member of the system administration team, you want to learn whether it is useful to have these work in one central system, which country versions would be required, and whether these are contained in the standard SAP system. You also want to know what effects different time-zones have on cross-zone business processes and the work of end users.

© 2003 SAP AG. All rights reserved.

Centralized and Decentralized Implementation

In principle, a large number of different implementations are conceivable with regard to centralized and decentralized data storage and business processes. Some of these are described here. When considering these, we will look at "best practice" observations from real implementations, along with technical advantages, disadvantages, and restrictions.

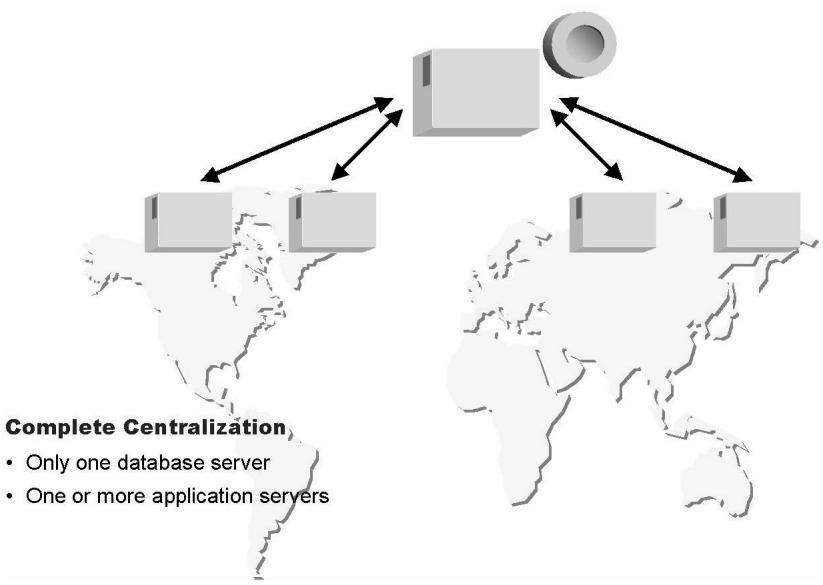


Figure 32: Complete Centralization

Complete centralization means mapping all business processes in a single client in a single system. Mapping using multiple clients can no longer be considered as centralization, as these function almost as independent systems. Advantages of complete centralization are:

- Central control
- Simpler and more integrated reporting
- Simpler maintenance
- Harmonization and standardization

Disadvantages are:

- Global complexity
- Higher risk
- Performance and size
- Differences in local business processes
- Problems with ownership

The following must also be considered:

 © 2003 SAP AG. All rights reserved.

- Global options and restrictions
- Which transactions are required, in detail
- Which end users work with the system
- Local acceptance for a global solution
- Difficulties that are to be expected with a global implementation

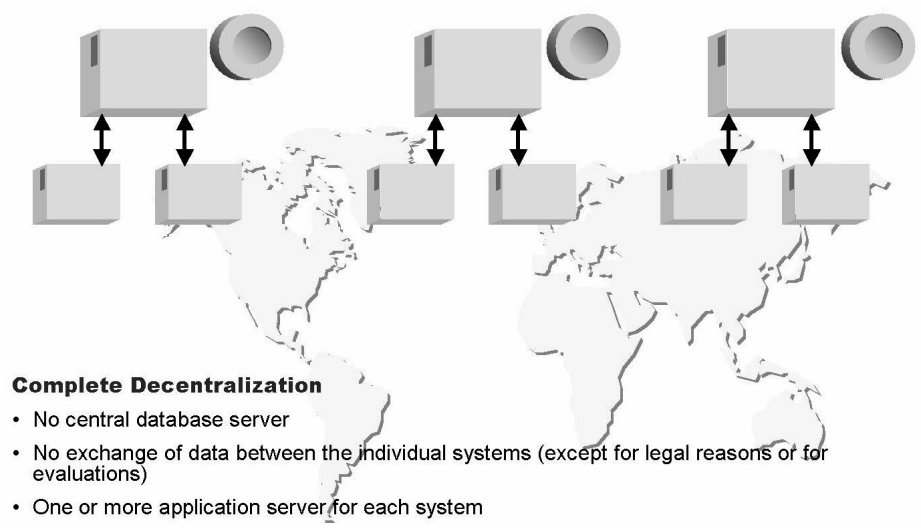


Figure 33: Complete Decentralization

With a completely decentralized implementation, the individual local systems are detached and independent of each other. Each individual system can use its own standards. Communication with other systems takes place only to a limited extent, using individual, defined interfaces. Advantages of an implementation of this type are:

- Local freedom with regard to functions, time planning, and approaches
- Local independence
- No expense for reconciliation

Disadvantages are:

- No synergies
- Redundancies, incompatible systems and interfaces
- Redundancies with regard to implementation and maintenance
- Restrictions in terms of SAP support

The following must also be considered:

- Global costs due to incompatibilities
- Global costs for implementation and maintenance

 © 2003 SAP AG. All rights reserved.

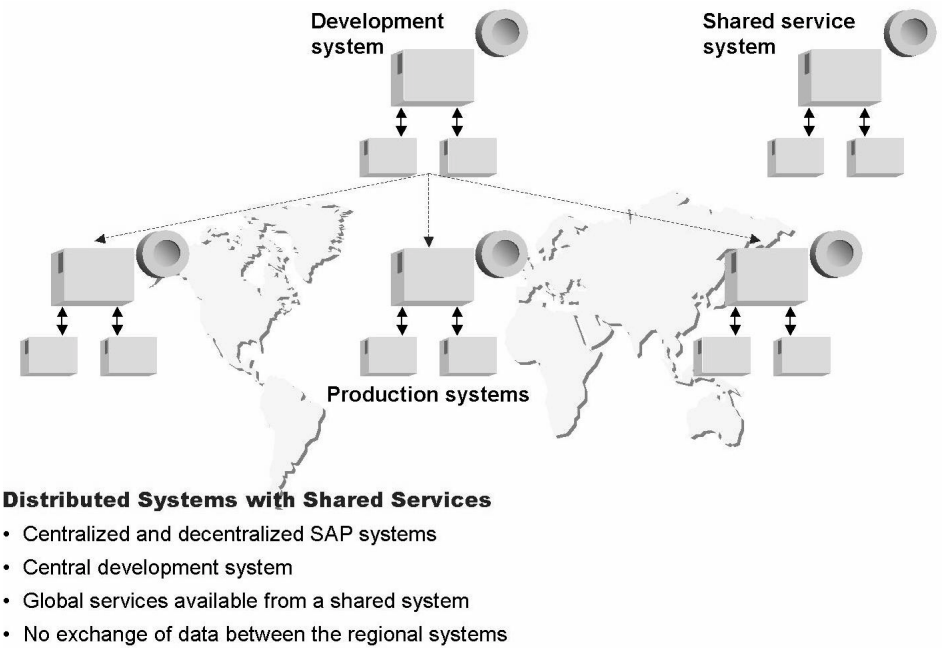


Figure 34: Distributed Systems with Shared Services

A decentralized approach with a common development system and shared services that are used globally can combine advantages of both purely centralized and purely decentralized implementations:

- Local independence with regard to business processes
- Avoidance of redundant functions
- Distribution and avoidance of implementation and maintenance costs

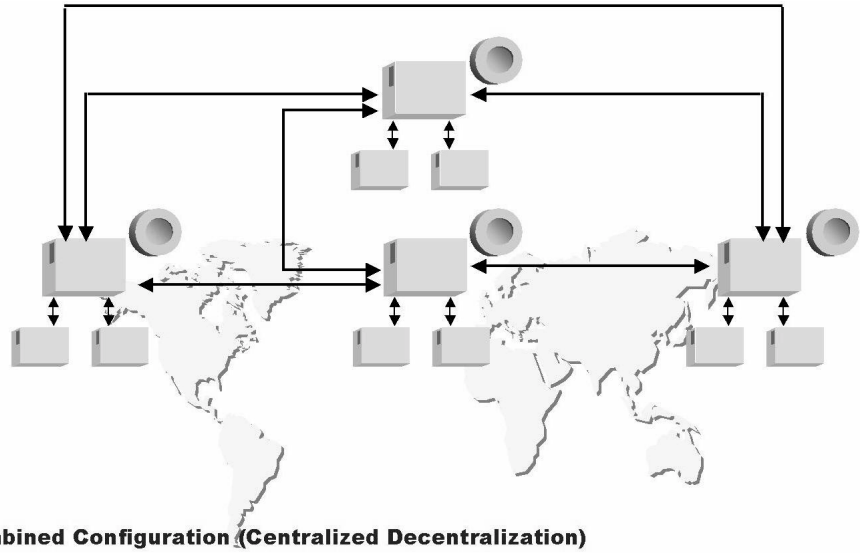
A number of disadvantages remain:

- No real integration of business processes
- Need for interfaces with associated problems

The following must also be considered:

- Harmonization and standardization
- Global costs for implementation and maintenance

© 2003 SAP AG. All rights reserved.



Combined Configuration (Centralized Decentralization)

- Centralized and decentralized SAP systems
- Regular exchange of data between the systems to support global processes

Figure 35: Combined Configuration: Centralized Decentralization

If multiple individual systems are connected in such a way that they are integrated and consolidated using business processes, this is described as a centralized decentralization. Advantages of such an approach are:

- Distribution and reporting
- Specific integration and complexity
- Standards and communication using business processes
- Possibilities for change

Disadvantages are:

- Restricted global control
- Complex network of business processes between the systems
- Less than optimal from a local point of view

The following must also be considered:

- Whether distributing the business processes is actually useful
- Distribution scenarios
- Local acceptance
- Possibilities for cooperation

In addition to the considerations related to business processes, other criteria must be taken into account. These could be of a geographical or technical nature, or relate to service and support.

© 2003 SAP AG. All rights reserved.

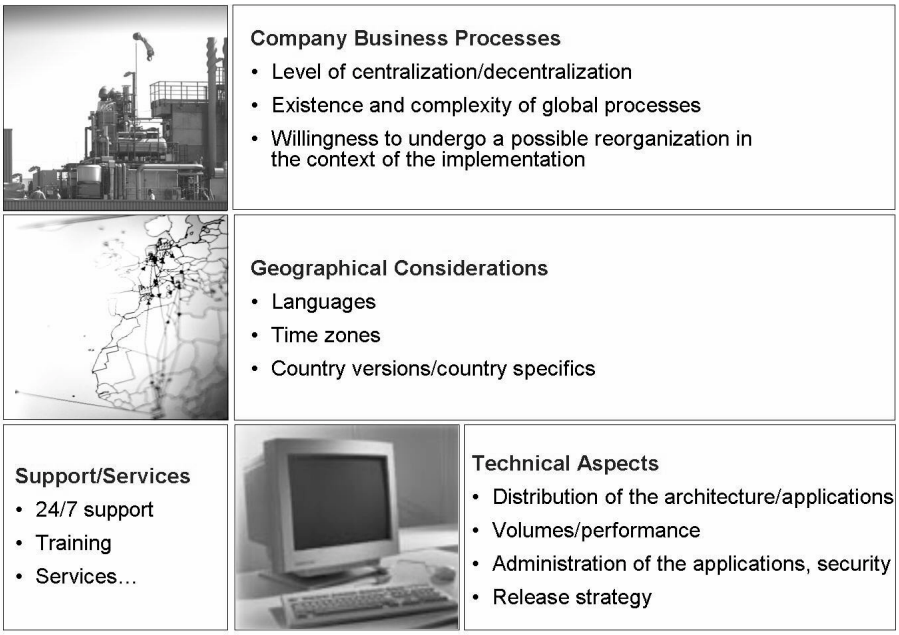


Figure 36: Criteria for a Strategy

There are different scenarios for implementation. The trend is currently moving in the direction of centralization and shared services. Restructuring from a centralized to a decentralized solution is relatively simple. Restructuring in the other direction requires significantly more expense. If you are attempting to create a centralized implementation, you should contact SAP for help; Global ASAP exists for this purpose. It offers a strategic Solution Roadmap with tools and project plans for the implementation. For more information, see the SAP Service Marketplace, http://service.sap.com/globalization and the Global Systems Symposium, http://www.sap.com/community/events/gss/index.asp, where there are presentations from customers who operate globally.

Country Versions

When implementing an SAP system, you not only need to consider that the local language for each country may need to be made available in the system, but also that there are country-specific differences in business processes and requirements in each country. In this way, for example, the details of the Financial Accounting (FI) application in SAP R/3 function in a different way in Germany from the way they function in Spain or Poland. You must take this aspect into account. This is known as a country version. Different country versions result in different business processes, different data storage, and different Customizing options. This is particularly the case due to legal differences, such as taxation, reconciliation, and reporting. Default settings and templates are also provided with a country version.

 © 2003 SAP AG. All rights reserved.

You must technically differentiate here between situations where the required country version already exists in the SAP standard and can be activated directly in Customizing, and situations in which the country version must be subsequently implemented in the system. A non-standard country version is usually imported as an add-on using transaction SAINT. In some cases, it is a question of transport requests that are simply imported using normal transport control. For some countries, there are no ready-made country versions. In this case, you must adjust the system as far as possible using Customizing, and then implement additional specifics using your own developments, user and customer exits, and BAdIs or modifications.

At this point, you can use transaction SAINT to demonstrate whether a country version has been imported into the system as an add-on. This should not be the case.

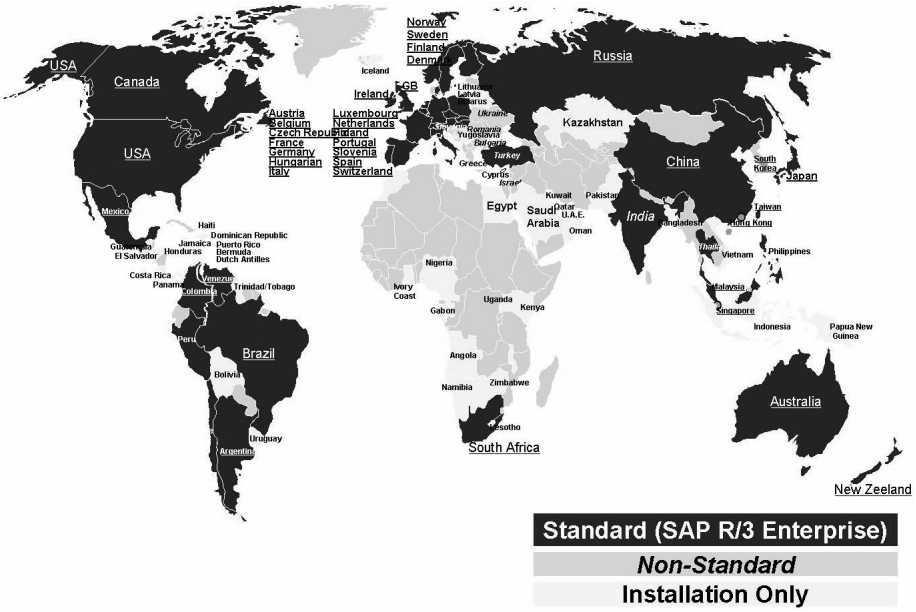


Figure 37: Availability of Country Versions

Standard (SAP R/3 Enterprise) means that this country version is contained in the standard SAP R/3 Enterprise system. There are fewer country versions in older releases. **Non-standard** refers to country versions that are available, but which must be imported as an add-on or as a transport request. **Installation only** refers to country-specific systems for which there is no ready-made country version. This graphic shows the current status at the time of printing. For more information about country versions that are currently available, see the SAP Service Marketplace at

 © 2003 SAP AG. All rights reserved.

http://service.sap.com/globalization. For more information about using individual country versions, see the online documentation, under *SAP Library → SAP R/3 Enterprise Application Components → Cross-Application Components → Country Versions*.

As soon as country versions are included in the standard SAP system, they are available immediately with the next release. They can be used in parallel and are compatible with each other and with other SAP solutions.

Non-standard country versions are provided by SAP or by one of SAP's partners, but this is not necessarily the case with every successive release. In this way, a country version could be available as an add-on with SAP R/3 4.0B, but not with SAP R/3 4.5B, and then available again with SAP R/3 4.6B. It could then be included in the standard system for SAP R/3 Enterprise (4.7). Of course, this has implications for the upgrade strategy of the system landscape. These country versions are also often not available at the time of a new SAP release, but only some time later. These considerations also apply when importing Support Packages. You must consider the fact that a subsequently imported country version makes modifications in the standard SAP system. The use of multiple non-standard country versions in a single SAP system should therefore be checked very carefully before implementation. It is also not possible to guarantee compatibility with other add-ons (such as industry solutions). You should ensure that relevant Conflict Resolution Transports (CRT) exist when importing Support Packages.

Time Zones

A company operating globally with cross-country and therefore cross-time-zone business processes requires a way of synchronizing time-dependent processes. SAP supports the use of different time-zones in some SAP solutions to enable this. It is not sufficient to use a uniform, system-wide timer in this situation. This would lead to a situation where, in extreme cases, a business process falls on a different day from the one that should technically be the case for the balance sheet. In this way, for example, a Japanese end user working for a Japanese company in Japan that creates a document at 6 a.m., but who happens to be working in an internal system with a German timer, would create this document on the wrong day, in this case, the previous day. This would be incorrect from a business point of view.

SAP systems support time zones in various areas. Time zone functions are fully integrated into the SAP kernel. This means that the system can convert date and time more quickly.

The local time of a particular user is dependent on the geographical area to which the user belongs. The time difference between local time and the absolute time, Universal Time Coordinated (UTC), is influenced

 © 2003 SAP AG. All rights reserved.

by the following factors: geographical area, political factors (time zones are assigned to a country, a region, or even a postal code), and possibly daylight saving time, which does not have a uniform duration.

The local time is calculated on the basis of Greenwich Mean Time (GMT), also known as Universal Time Coordinated (UTC). For example, if it is 18:00 in Greenwich, it is 10:00 in Los Angeles. Los Angeles is therefore eight hours behind Greenwich. At the same time, it is 3:00 a.m. in Japan. (That is, nine hours ahead of UTC.) Daylight saving time is also taken into account in some time zones. In this case, the system uses a daylight saving time rule (DST rule) to calculate the local time. In these time zones, the clocks are put an hour forward, to make full use of the longer days during the end of the spring months, summer, and the start of autumn, and to take the sunlight during the evening hours into account. The assignment of a particular town or region to a time zone depends on geographical and political factors. The time difference between two places on the globe can be up to 24 hours.

The time for a time zone is determined by the following components:

- Time zone: This component contains only the name of the time zone (such as CET or EST) and reference to the corresponding time zone and summer time rules.
- Time zone rule: This rule defines the time difference (+/- hours and minutes) to UTC on the basis of geographical and political factors.
- DST rule: This rule defines the time difference between daylight saving time and standard time (+1 hour). It does not, however, define the start and end dates of daylight saving time.
- Fixed summer time rules: In cases where daylight saving time is not defined by variable rules, this rule defines the start and end dates of daylight saving time for a particular year.
- Variable summer time rules: These rules define how the system calculates the start and end dates of daylight saving time. You can change these rules at any time. This means that you do not have to enter the start and end dates of daylight saving time every year.

© 2003 SAP AG. All rights reserved.

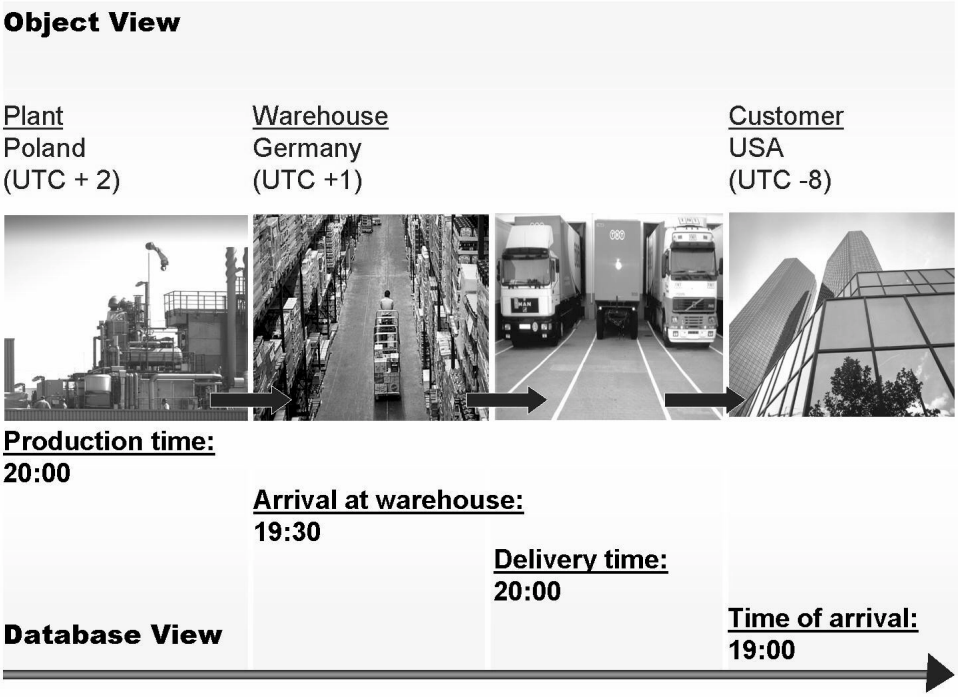


Figure 38: Business Process Across Multiple Time Zones

It is necessary to operate multiple time zones in one SAP system simultaneously. The default time zone is used for processes that cannot be assigned to an individual time zone. The time is calculated based on UTC (GMT). In general, it is possible to assign a time zone either to an end user or to an object.

Time zones are of particular importance in the following areas:

- Basis/SAP Web AS (SAP Web AS: uniqueness and serialization of system events)
- Logistics (planning and tracing, dispatch and transport, goods movement)
- Finances (determining posting date, resulting in posting period)
- CRM (determining the date in business processes, call centers, mobile clients, Internet sales)

We must decide various time zones for additional considerations:

- Operating system time: Time of all servers of the SAP system, including daylight saving time
- System time: Must be identical to operating system time; is set in the SAP Reference IMG
- User time: Time zone in which the user works; is set in the user master record

© 2003 SAP AG. All rights reserved.

- Business/reference object time: Objects can have a time zone as an attribute
- Default time: Is used if no other time zone is defined

Before time zones can be used in the SAP system, you must make system settings, maintain time zones, and assign these to users.

- Maintain system settings: The first time that you log on to an SAP system, the system automatically determines the time zone that it is in. As a precaution, you should check that the time zone setting in the system is correct: In Customizing (transaction SPRO,) choose *Goto* → *SAP Reference IMG* → *General Settings* → *Time Zones* → *Maintain System Settings*. Enter the system time zone and the default time zone for the user. As the system and user are usually in the same place, the system time zone and default time zone for the user are usually the same. To activate the time zone function, select the *Time zones active* field.
- Maintain time zones: As the default time zones are delivered with the SAP system, you do not need to maintain them manually. You should, however, check the settings. In Customizing (transaction SPRO), choose *Goto* →*SAP Reference IMG* → *General Settings* → *Time Zones* → *Maintain Time Zones*. As the individual parts of a time zone structure are based on each other, you can define time zones by completing parts of the table as follows. First define the variable summer time rule or the fixed summer time date (choose *Variable Summer Time Rules* or *Fixed Summer Time Rules*). Then define the time adjustments that are caused by daylight saving time (choose *Summer time rules*). Then define the time adjustments that are caused by the time zone (choose *Time zone rules*). Finally, define the time zone key (choose *Time zones*).
- Maintain geographical assignments: In Customizing (transaction SPRO), choose *Goto* → *SAP Reference IMG* → *General Settings* → *Time Zones* → *Maintain Geographical Assignments*. To assign countries, regions, or postal codes, choose *Time zones in country*, *Time zones in country/region*, or *Generic time zones to postal code*. Define the standard time zone for the country or the region. If there is more than one time zone in a country or a region, enter the preferred time zone.
- Define time zones in user defaults: In the main SAP menu, choose *Tools* → *Administration* → *User Maintenance* → *Users* (transaction SU01). Enter the name of the user and choose *Change*. Select the *Defaults* tab page, and enter the time zone for the user in the *Personal Time Zone* field. If there is no time zone defined for a user, the system uses the time zone for the company to which the user is assigned. To view the time zone for the company, select the *Address* tab page. Then choose *Display Full Company Address* (the arrow icon) under *Company*. The company's time zone is displayed on the detail screen for the company address.

© 2003 SAP AG. All rights reserved.

- Check time zone function for the user: In the main SAP menu, choose *System* → *Status*. If the time zones function and the user defaults are correctly configured, no time zone information appears in the *Usage data* section of the *System: Status* dialog window for users in the same time zone as the system; for users in a different time zone, the time zone of the user, the local time of the user, and the local date of the user are displayed (the local date is displayed only if the local date and system date are different).

At this point, you can show the content of table *TTZCU* or transaction STZAC, in which the default time zone must be set (since SAP R/3 3.1H). The *System time zone* must be the same as the *User's default time zone* and the *Time zones active* checkbox must be set.

You can also show the assignment of a time zone to a user master record in transaction SU01.

The settings for the time zones are made using the IMG (transaction SPRO) *Goto* → *SAP Reference IMG* → *General Settings* → *Time Zones*. If possible, you should also go into more detail about defining different time zones with daylight saving time at this point (transaction STZBC).

SAP Web AS (the SAP "Basis" system) always uses only the system time zone, and not any deviating time zone. This includes services such as

- Background processing
- Update objects for asynchronous updates
- System log messages
- Change verifications

This means, for example, that if a background job is to be defined (for example, using transaction SM36) or displayed (for example, using transaction SM37), the individual time zone of the calling user must be taken into account, that is, the user must select the relevant system time, and not his or her own local time.

An exception to this rule is the overview of spool requests (transaction SP01). The individual time zone of the calling user is taken into account here.

© 2003 SAP AG. All rights reserved.

You can demonstrate this at this point using transactions SM37 and SU01. To do this, you must create and use a user with a time zone different from the system time zone. In the training system for the ADM102 course, a user has already been defined in client *100* with the user name *Downunder* and the password *timezone*. An Australian time zone has been assigned to this user. You can use this user for demonstration purposes.

Essentially, it is possible to determine the time zone using the user master record or business objects.

Defining the time zone using the user master record: You can assign an individual time zone to every user master record in the SAP system from the SAP menu by choosing *Tools* → *Administration* → *User Maintenance* → *Users* (transaction SU01). Many business transactions in SAP R/3 use this setting.

For example, a user that works in Los Angeles has been assigned the PST (Pacific Time, -8 hours from UTC) time zone. He is working, however, in an SAP R/3 system that is in Germany and that has CET (Central European Time) as the system time zone. If this user posts to an account on Thursday evening at 8:00, not only would the default value for the date of his post be that Thursday, the action would also be posted with Thursday's date. This procedure is correct from a business point of view, although it is already Friday morning according to the system time. In this way, it is possible for posts to be made for a date in the past, or, in the case of an East-Asian time zone, even for a date in the future (from the point of view of the SAP system). This is a simple solution that is satisfactory for the user, as long as it is not a case of cross-time-zone business processes.

Defining the time zone using an object: The time zone can also be linked with objects, such as company code, plant, warehouse, or customer. This is useful when considering cross-time-zone business processes. This procedure is used in particular in mySAP Customer Relationship Management (mySAP CRM) and with SAP Advanced Planner and Optimizer (SAP APO) in the context of mySAP Supply Chain Management (mySAP SCM). Time stamps are stored in UTC time and displayed converted to the local time zone, as required. The SAP system uses the local date, local time, and the time zone of the object to calculate the time stamp. The system stores the time stamp and the time zone of the object. If the user then calls the object with the time details again, the system converts the UTC time information stored in the time stamp to the relevant local time. The system therefore stores the time information only in UTC and not the local time originally entered by the user.

© 2003 SAP AG. All rights reserved.

Conclusion

It is important to note in conclusion that there is no uniform arrangement with regard to time zones in the SAP system. Whether and how time zones are taken into account, whether user-related, object-related, or not at all, depends on the individual implementations in the individual transactions. Here is a summary of key points:

- The system time zone is (almost) always used for SAP Basis/SAP Web AS transactions.
- SAP R/3 usually works with the user-related time zone.
- Only the system time zone is used in SAP R/3 HR.
- The object-related time zone is often used in mySAP CRM, with SAP APO, and in mySAP SCM.
- The use and consideration of time zones is release-dependent.
- In general, time zones are taken into account “as implemented.”

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

Repeat the aspects of globalization from this lesson

Discussion Questions

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

- When should I choose a centralized/decentralized system configuration?
- What is a country version? How does it get into the SAP system?
- How do time zones work?

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Recognize some effects of a centralized or decentralized implementation, the possibility of using multiple time zones in one SAP system, and the need for country versions

Related Information

For more information about time zones, see the online documentation under *mySAP Technology Components* → *Cross-Application Functions* → *Time Zones (CA-GTF-TIM)* and the online documentation for individual products (such as SAP CRM).

© 2003 SAP AG. All rights reserved.

Lesson: Multiple Languages in an SAP System

96 Lesson Duration: 70 minutes

Lesson Overview

This lesson explains the ability of the SAP system to use multiple languages simultaneously. This includes the mechanism of language imports and language supplementation. Translations and address versions are also briefly discussed.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the process required to install an additional language in the SAP system
- Import an additional language into the SAP system

This lesson should allow the participants to import additional languages into an SAP system independently. To do this, it is necessary to first explain the concepts of the multilingual capability. Related areas that should be explained are: Supplemental mechanism for languages that are not completely translated, reimporting texts from Support Packages after a language import, translation possibilities with transaction SE63 (conceptually only), address versions.

The exercise task includes importing an additional language. As this runs for quite a number of minutes (hours), it is sensible to set the exercise task over the lunch break (too short, if the import is performed simultaneously in both systems on each host) or, better, over night. The specified times are almost doubled if the DEV and QAS groups perform the exercise tasks at the same time. In principle, however, this is certainly possible.

This lesson does not consider the fact that importing an additional language could also make the use of an additional code page (in the ADM102 course, this is explained in another lesson). We assume here that the additional language runs on the code page that is already installed.

Business Example

You are a system administrator in a company that operates worldwide. To support employees in a new subsidiary with a local language, you want to install an additional language in your SAP system.

© 2003 SAP AG. All rights reserved.

Multilingual Capability of the SAP System

Depending on release status, the SAP system supports up to 30 different languages. The language with which a dialog user works is specified during logon by making a selection on the logon screen, by a system default, or by a default setting in the user master record. So that it is possible to log on with the desired language, this language must first be imported into the SAP system. After a new installation of an SAP system, only German (DE) and English (EN) are initially installed. You can use transaction SMLT to import additional languages, if desired.

If you select a language other than German, English, and Japanese, you should note that this language is incompletely translated. The texts that are not translated in this language must be provided using a supplemental language. This supplemental language must be a completely translated language. This can also be recursively configured; so that you use a number of supplemental language, each of which supplements texts that are not translated in the previously configured languages.

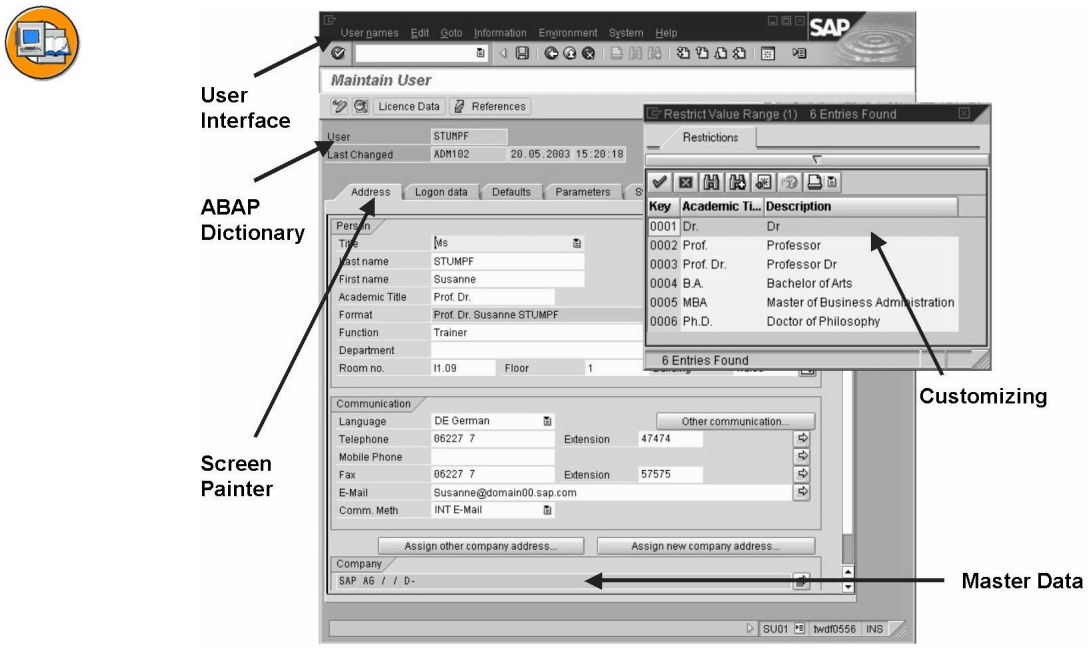


Figure 39: Language-Dependent Objects

There are a large number of different language-dependent objects. These objects are translated to a greater or lesser extent for different languages. Delivered objects for which no translation exists in a language are completed using the supplemental language. If the customer creates new language-dependent texts, they exist only for the language version that corresponds to the logon language. The texts are, of course, not maintained for other languages. This can lead to inconsistent data. This

© 2003 SAP AG. All rights reserved.

means, for example, that not only are the texts not displayed for users that logon with a different logon language, but these users may also receive an error message such as "Complete all required fields" for objects that have already existed for some time. However, individual objects behave differently in this context:

- The text is displayed in a different language
- Nothing is displayed
- The user receives a warning or error message
- The transaction can no longer be run

From "Only English" to "No English at All"

The extent to which individual languages need to be imported into the SAP system and the extent to which these are completely translated depends not only on economic, but also political and cultural criteria. User acceptance is an important aspect in these considerations.

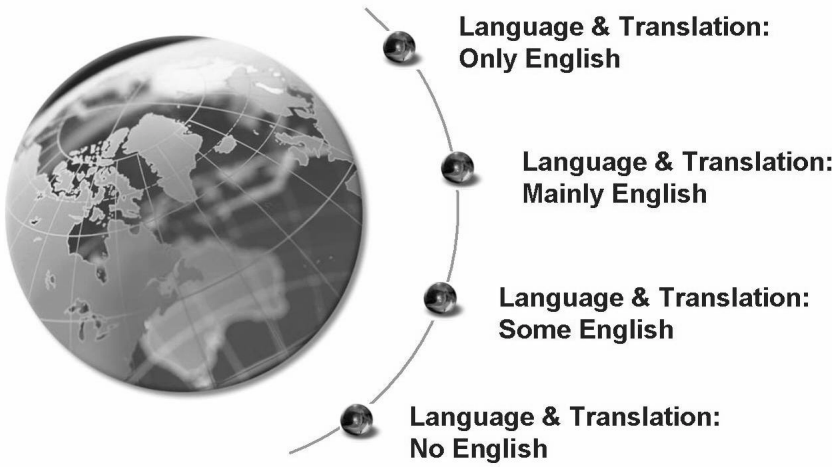


Figure 40: Using Different Languages

Language and translation: Only English

- In this case, the SAP system supports only English as a logon language and when printing. All evaluations are generated in English. All communication is performed in English. The problem of language-dependent objects does not arise, and no costs are accrued for translations. The question here is not only whether the users have any problems with the English language, but also whether English language evaluations, such as annual financial statements, are legally acceptable. This is therefore a solution for a relatively small number of countries that use English as an official language.

Language and translation: Mainly English

 © 2003 SAP AG. All rights reserved.

- In this case, it is assumed that the users have a command of English. They log on in English, and the menus and so on are also in English. However, evaluations are generated in the national language. If desired, language-dependent objects are translated. This requires a certain translation cost and a small amount of Customizing (for example, for units of measurement). The master data is available in the national language. Communication is also performed in the local national language.

Language and translation: Some English

- This solution can be regarded as "Best Business Practice." The master language of the system is English. However, the users do not require significant knowledge of English. They log on in their local language, and can therefore use language-dependent objects, such as menus, input screens, and so on. Communication and all legally-relevant evaluations are also performed in the national language. Texts that are not translated are displayed using the supplemental language English. Depending on the level of translation of the language in question, there may be no translations in some areas. Additional texts can be translated, if desired. However, this is not necessary in many internal areas. There are periodic language supplements for newly generated texts. The standard tools are used for these. Manual translations remain manageable.

Language and translation: No English

- In this case, the users have no English knowledge or English is unacceptable due to political or cultural considerations. In this case, not only must the corresponding national language be imported, but all other required texts that have not been delivered translated must be manually translated. It is not possible to use English as a supplemental language. This means continual translation activity in a multilingual system. This is necessary even if the system is only operated in one language, as new language-dependent texts are imported into the system with every Support Package and every upgrade, and it may be necessary to translate these.

The Translation Workbench

The translation workbench is a tool for translating language-dependent texts. Translation is supported by a proposal pool. The translation workbench can be used to translate additional texts in incompletely translated languages, or to provide your own language-dependent texts. You can call the translation workbench by choosing *Tools → ABAP Workbench → Utilities → Translation → Short and Long Texts* (transaction SE63). You can find the relevant editors for different language-dependent texts here.

 © 2003 SAP AG. All rights reserved.

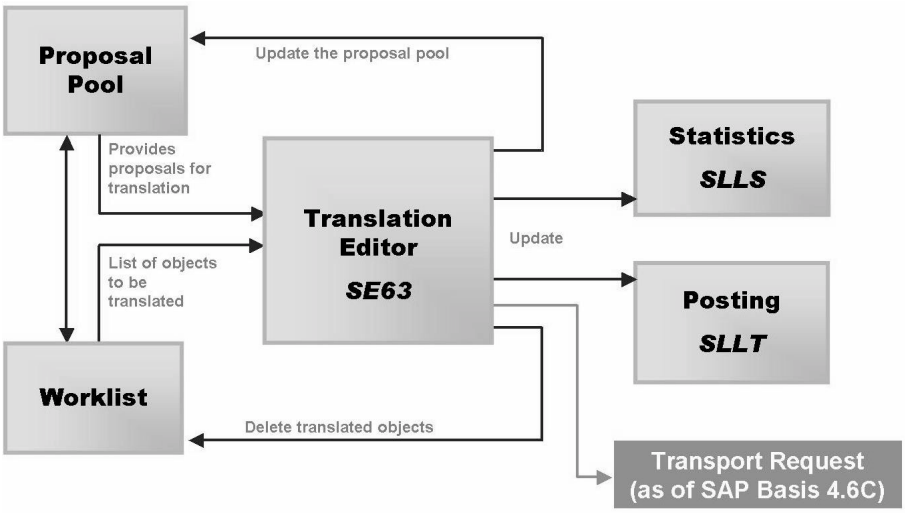


Figure 41: Components of the Translation Workbench

The following procedure for operating the translation workbench is intended as a supplement. You should not go through it with the participants, but interested participants can use it to learn how to use the tool. For additional explanations, see the online documentation.

You can translate texts as follows:

- Using a worklist
- By calling objects directly

Most objects to be translated are assigned to packages. You can call a worklist for one or more packages. All objects that are assigned to the packages that you specify are then displayed in your worklist. To fetch a worklist, follow the procedure below:

- Call transaction SE63 and choose *Worklist* → *Standard worklist*. The dialog box *Worklist: Query - User Parameters* appears. This dialog box contains settings for user name, source language, target language, and worklist number (this should be 1 for a standard worklist).
- Check the details and choose *Copy*. The *Worklist: Object list for User* <*your user*> appears.
- Place the cursor on your user name and choose *Get worklist*.
- In the dialog window, enter the package or packages that you want to process in your worklist. You can enter the desired package or a generic value.
- Choose *Copy*. The creates a worklist for your user name with the packages that you have specified.

 © 2003 SAP AG. All rights reserved.

The translation status of the texts is shown in the worklist with the colors blue, yellow, and red.

- Blue: Texts that have been correctly translated and match the translation in the proposal pool
- Yellow: Texts that have been modified
- Red: New texts

You can either process your worklist sequentially or select the individual objects by double-clicking them. The numbers next to the objects in the worklist stand for the number of translated, modified, and new texts in each case.

To translate a short text, follow the procedure below:

- When you call a short text object in your worklist, the short text editor appears. If you call a short text object by choosing *Translation* → *Short texts* → <*object type*>, a screen on which you must enter the name of the object first appears. Enter the name of the object. Enter the source language and the target language and choose *Edit*. The short text editor appears.
- The texts for the object that you have called are displayed as a list. The colors indicate the status of the translation of each text. The colors have the following meanings:
  - Blue: Texts that have been correctly translated and match the translation in the proposal pool
  - Yellow: Texts that have been changed or that have been rejected by the proposal pool for some other reason; these reasons are explained in more detail under *Translation Status*.
  - Red: New texts
- You can now enter or edit the target texts. You do not need to edit texts that are colored blue. These texts are correctly translated; that is, they match the translations in the proposal pool. Texts marked yellow are rejected by the proposal pool and need to be edited. This editing is explained in more detail under *Translation Status*. Texts marked red are new texts. To call existing proposals from the proposal pool, choose *F5*. If the proposal pool contains proposals for your text, the best proposal for this translation is displayed below the target text line; it is marked blue. If the translations are longer or shorter than the displayed proposals, you can create length variants. If your translation is different in meaning from the best proposal, you can create an application standard or an exception, depending on whether you have used the terminology that applies for your application. To make the relevant default settings for your short text editor, choose *Utilities* → *Settings* → *Short Text Editor*.
- Save.

 © 2003 SAP AG. All rights reserved.

To translate a long text, follow the procedure below:

- If you call a long text from the worklist, the long text editor appears immediately. If you call a long text object by choosing *Translation* → *Long texts* → <*object type*>, a screen on which you must enter the name of the object first appears.
- Enter the name of the object. If required, choose *F4* to select an object.
- Enter the source language and the target language and choose *Edit*. The long text editor appears. The screen is split into two sections. The source text is displayed in the upper section. This is the *Split-Screen Editor*. Enter your translation of the source text in the lower section of the screen.
- Save.

All translations are stored in the proposal pool. The texts are stored by application in accordance with the SAP component hierarchy. The proposal pool contains the following hierarchy levels:

- System standard
- Application standard
- Exception

The system standard and the application standard can be subdivided as follows:

- Node text
- Length variant

To display proposals, choose *Proposal Pool* in the short text editor. If necessary, you can edit the proposals directly.

Language Import

New developments for language transports:

The extended language supplementation program is new as of SAP Web Applications Server 6.20. As of SAP Web Application Server 6.20, client maintenance and the program *RSREFILL* are integrated into the language supplementation program. Despite the extended functions, the language supplementation program is now easier to operate.

As of SAP Web Application Server 6.10, the functions for importing language parts of Support Packages and client maintenance are integrated into SMLT. Previously, these functions were only provided by importing Support Packages. Export procedures are now also displayed in transaction SMLT. You can find all of the tools for language transports

© 2003 SAP AG. All rights reserved.

in transaction SMLT under *Goto* → *Other Tools*. You can start the tools from here. Previously, these tools had to be called separately. Additional improvements were made to make SMLT easier to use.

As of SAP Basis 4.6D, you can start transport processes in parallel. This applies both for language imports and for language exports. This means that the import or export can be performed more quickly. Changes were made to the underlying transport technology and to the user interface with the aim of improving the performance and the operability of the language transport.

Language transports were completely reorganized with SAP Basis 4.6C. Languages are now transported with the SAP transport program *R3trans*. Together with the transport control program *tp*, *R3trans* exports objects from a source system using a transport request that is imported into one or more target systems. Switching language transports to the tools of the Change and Transport System means both that translations for a language can be included in transport requests and that transports of a complete set of translations for a language are now performed using transport requests. For more information about the transport programs, see the SAP Library. Choose *Help* → *SAP Library* → *SAP Web Application Server* → *Change and Transport System* → *Transport Tools*. A significant improvement is that the translation environment, SE63, was connected to the transport system, meaning that as of SAP Basis 4.6C, translation activities can be stored in transport requests and distributed in the system landscape in the same way as developments. This change to the language transport can significantly reduce the downtime when importing languages during an upgrade. Transaction SMLT (Language Management) was completely revised. All functions are available on an object-oriented basis for the relevant languages. The transport technology was completely changed as of SAP Basis 4.6C. This means that *RSLTEXPO* exports can no longer be used.

You can use transaction SMLT to import additional languages into an SAP system.

© 2003 SAP AG. All rights reserved.

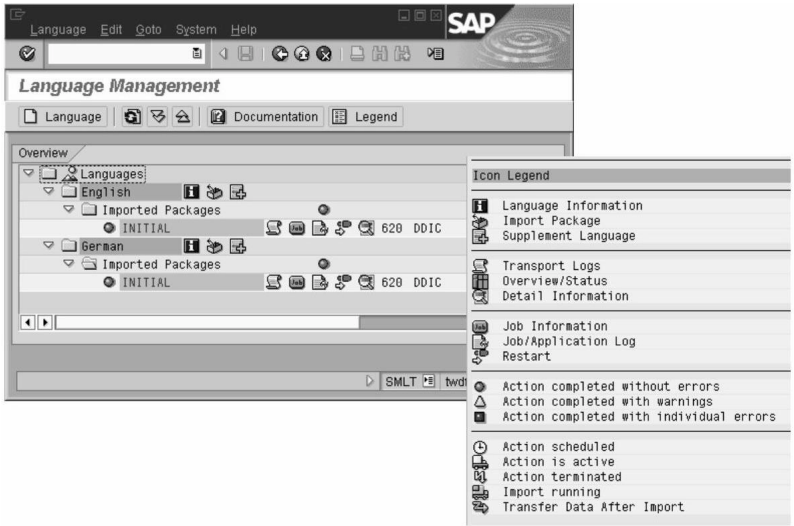


Figure 42: Transaction SMLT: Language Import

During the execution of an import or language supplementation, the SAP system locks language management activities for the relevant language. The system rejects all other activities that are to be performed with regard to a locked language. Other languages are not affected by this, meaning that parallel actions that operate on other languages can be performed.

SAP delivers Support Packages to correct errors that occur in important transactions. Support Packages can contain language-dependent data such as message texts, ABAP text pool entries, or screen texts. After importing a Support Package with transaction SPAM, the translated texts for these objects have the newest status for all languages that exist in the system. Problems can occur if an additional language is to be imported from the languages CD into a system into which Support Packages have already been installed. As the language CD is created before the first Support Package is issued, and the objects in the Support Packages are only provided with translations for the languages that have already been imported, importing a language subsequently places the objects contained in the Support Packages in an undefined status with regard to translation. Texts can be both obsolete and therefore incorrect, or generally missing (see also SAP Note 352941). Since SAP Basis 4.6C, you can subsequently import the language data contained in the Support Packages to achieve a consistent language status. To do this, call transaction SMLT and choose *Language → Special Actions → Import Support Packages.*

© 2003 SAP AG. All rights reserved.

- Data definitions are stored in Customizing tables, such as TVKO

TVKO	
Client	*Sales Org.*
100	DE01
100	SP01

- During Customizing, language-dependent text is maintained in the logon language
  - This data is visible during Customizing with the logon language EN
  - The user is not aware that he or she is maintaining language-dependent data

TVKOT			
Client	*Lang.*	*Sales Org.*	*Text*
100	DE	DE01	V.O. Deutschland
100	EN	DE01	S.O. Germany
100	DE	SP01	V.O. Spanien
100	EN	SP01	S.O. Spain
100	ES	SP01	S.O. Espana

Figure 43: Language-Dependent Texts in Customizing

Customizing data is data that is only created at the customer site in the context of Customizing. The Customizing data for a customer client is not overwritten during a language import or upgrade. The data is always imported only into client 000. The reason for this is that Customizing data becomes the responsibility of the customer through the customization, and must not subsequently be overwritten by data from SAP. This also applies, without restrictions, to translations. However, it is sometimes desirable for the customer to refresh the template data and default settings with current translations from SAP. The clients should only ever be copied from client 000 after this has been updated with all required languages. There are two ways to update a translated language for Customizing an existing customer client:

- Using tools
- Manual translation

Using Tools

SAP provides the following tools for updating the Customizing texts in higher clients:

- Report **RSREFILL** compares the Customizing texts in logon clients with the SAP defaults in client 000 in a reference language. If the texts are identical, the translations of these texts in the target language are copied from client 000 to the logon client. Data is only added, and no existing data is overwritten (see SAP Note 48047).

© 2003 SAP AG. All rights reserved.

- Client maintenance in transaction SMLT works in a similar way to report **RSREFILL**, but does not include the comparison with a reference language. In this way, you can copy more texts from client 000 to the desired clients. As with the **RSREFILL** program, Customizing data is added. No data is deleted or overwritten. You can start client maintenance in transaction SMLT by choosing *Language → Special Actions → Client Maintenance*.

We recommend, however, that you use the extended language supplementation program instead, into which the **RSREFILL** program and client maintenance are now integrated. Combining **RSREFILL** or client management with the language supplementation program means that for every client-dependent Customizing table:

- Any previously existing supplementation is reversed
- **RSREFILL** or client maintenance is performed
- The table is supplemented

This means that it is no longer possible for problems to occur due to the order in which **RSREFILL**, client maintenance, and language supplementation are performed.

Manual Translation

Due to the connection between the translation environment and the Change and Transport system, it is now possible to create translated Customizing texts in a perfect system, to include these in transport requests, and to distribute them in the system landscape in a targeted way. Use transaction SE63 to translate the remaining Customizing texts. Record the translation activities in transport requests and then distribute the requests in the system landscape using the Transport Management System (TMS) (transaction STMS).

Ensure that you take the following SAP Notes into account:

- SAP Note 482462: *Current note on language transport in Release 6.20*
- SAP Note 73606: *R/3 language combinations (non-Unicode)*
- SAP Note 15023: *Initializing table TCPDB (RSCP0004) (EBCDIC)*
- SAP Note 18601: *Frequently asked questions on language import*
- SAP Note 43853: *Consulting: Language-dependent and client-specific Customizing tables*
- SAP Note 352941: *Consultation: Languages and Support Packages*

For information about delivered languages, see the SAP Service Marketplace at http://service.sap.com/languages.

You must classify each language to make it known in the system. Only then is it possible to import a language package or supplement the language.

© 2003 SAP AG. All rights reserved.

1. In transaction SMLT, choose *Language* → *Classify*.
2. In the following dialog box, choose the language to be imported and the associated supplementation language.
3. Note that you must define the languages imported (using transaction SMLT or by upgrading) in the instance profiles to be able to log on in the relevant languages. The parameter is *zcsa/installed_languages*.

TVKOT			
Client	*Lang.*	*Sales Org.*	*Text*
100	EN	DE01	S.O. Germany
100	EN	FR01	S.O. France
100	EN	SP01	S.O. Spain
100	ES	DE01	S.O. Germany
100	ES	FR01	S.O. France
100	ES	SP01	S.O. Espana

Spanish is supplemented using the supplementation language English

Figure 44: Language Supplementation

We recommend that you supplement every language that does not have a degree of translation of one (1) after you have successfully completed the language import. A language with degree of translation of one (1) is a completely translated language. Currently, this is German and English. Supplementation guarantees the complete range of functions in a language. To display the degree of translation for a language in transaction SMLT, choose *Language* → *Information*.

The following rules apply for defining supplementation logic:

- Languages with a degree of translation of one cannot be supplemented.
- You should define a supplementation language for every language in the system (except languages with a degree of translation of one).
- You can only supplement texts from a language with a degree of translation of one or from a language that has already been supplemented with a language with a degree of translation of one.

 © 2003 SAP AG. All rights reserved.

You can display the current settings for a language at any time by choosing the *Information* button on the initial screen of transaction SMLT. If no supplementation language was specified when classifying a language, it is possible to do this at any time.

Users require the following authorizations in the *CTS_LANFKT* field of the *S_CTS_LANG* authorization object to be able to perform the relevant actions:

- *ADMI*: Administration in the language transport area, especially for classifying an additional language
- *INST*: Importing a language package
- *SUPL*: Supplementing a language
- *SHOW*: Display authorization; without this authorization, you cannot call transaction SMLT

All of the authorizations required for language transports are contained in the role *SAP_BC_TRANSPORT_ADMINISTRATOR*, which is delivered by SAP. Up to SAP Basis 4.6B (inclusive), the authorization checks for transaction *SMLT* were linked to the authorization object *S_LANG_ADM*. The authorizations are also evaluated in such a way that it is not necessary to adjust them in a language transport after an upgrade.

You can use transaction SMLT to import an additional language package. English and German are always completely available in newly installed systems. You therefore do not need to import these into installed systems.

1. For the language for which an additional package is to be installed, choose *Language* → *Import Package*.

2. In the following dialog box, specify the path to the language package. Choose *Find Package(s)*. The system searches for available packages in the specified path.

3. The system indicates whether a packages can be imported by displaying a green, yellow, or red traffic light next to the relevant packages. You can display additional information by selecting the traffic light. A green light means that all import conditions have been fulfilled, yellow means that warnings have occurred, and packages with red lights cannot be imported due to checks that produced errors.

4. Select one or more language packages with green or yellow lights for import by selecting the corresponding lines in the list.

5. After specifying the start time and, if desired, the target server, you can import one or more language packages by choosing *Execute*. This schedules a background job that performs the required steps.

© 2003 SAP AG. All rights reserved.

6. An import action is created for the relevant language on the overview screen of transaction SMLT, which you can administer and monitor by choosing the relevant icons.

You can restart terminated imports. To do this, choose the relevant function in transaction SMLT.

You can import multiple languages in parallel. Depending on the capacity of your SAP system, importing multiple languages simultaneously can cause greater workload than importing the languages sequentially.

After importing the languages, you must also import the language data in Support Packages. To do this, select the affected language in transaction SMLT and choose *Language* → *Special Actions* → *Import Support Packages*.

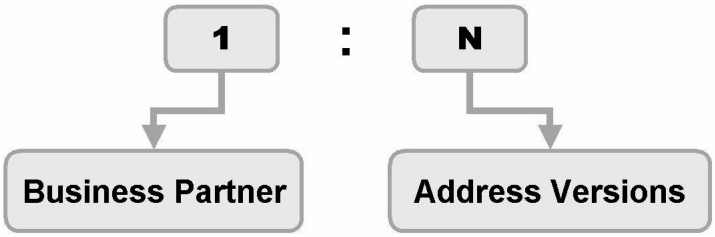
To fill translation gaps that exist in a language, you should always start language supplementation after the language import. The supplementation actions are client-dependent. The language supplementation is performed in the client in which you are logged on. If you use multiple clients, you must perform the language supplementation explicitly in each production client. You can access the texts stored in cross-client database tables simultaneously from all clients. The default setting is set in such a way that cross-client tables are supplemented if you are logged on in client 000.

1. For the language that is to be supplemented, choose *Language* → *Supplement Language*.
2. On the following dialog box, enter a description of the supplementation. The description is used only to easily identify the started supplementation on the overview screen of transaction SMLT.
3. In the middle part of the dialog box, specify conditions for selecting the quantity of tables to be supplemented. The fields have already been predefined in accordance with SAP recommendations and depending on the logon client; however, you can makes changes as appropriate.
4. After you have specified a start time and, if desired, a target server, you can supplement the language by choosing *Execute*. This schedules a background job that performs the required steps.
5. An entry is made for the supplementation on the overview screen of transaction SMLT, which you can administer and monitor by choosing the relevant icons.
6. You can identify the status of the action from the icon, which appears at the start of the relevant line.
7. For detailed information about the supplemented tables at any time, see the supplementation log. Each line of the log contains the table name, table class, table description, and additional information about

 © 2003 SAP AG. All rights reserved.

a supplemented table. The status, client, and supplementation mode are detailed first. Statistics about the number of processed, inserted, and modified lines are then included. A return code is also displayed. A green, yellow, or red traffic light is displayed for each table that has been completely processed, depending on the return code output. To display the meaning of the return codes, choose the *F1* help for the return code column of the table.

Address Versions



KNA1 (Customer data)		
CustNo	AddrNumber	Data Fields
1234	1000	xxxx

ADRC (Central Address Management)		
AddrNumber	Nation	Data Fields
1000		Standard Address
1000	R	Russian Address
1000	K	Kanji Address

Figure 45: Storing Address Versions

The worldwide use of SAP software means that different notations are used. The international address versions (abbreviated to versions) are a property of the Business Address Services to allow the printing of addresses in country-dependent scripts. In this context, “different scripts” does not mean country-specific differences within character sets, such as umlauts in German or accents in French, but rather scripts that consist of a completely different character set.

When printing addresses, you should note that the script type of the addresses to be printed is not determined by the current logon language, nor by the logon language at the time when the address is created. The international address versions allow you to print the same address in different script types (versions) depending on certain parameters. A

© 2003 SAP AG. All rights reserved.

Japanese address should be output in Kanji if the country of the sender is also Japan, and in "international script" if the country of the sender is not Japan (see SAP Note 316331).

The default version of an address can, in principle, be created in any of the permitted/available script types. However, it is useful to create all default versions of addresses within an installation in the same script to achieve uniform reporting.

Address versions are automatically available in all applications that use Business Address Services. As address versions must match the relevant default versions semantically and only represent one way of creating an address in multiple script types, you should ensure that the details of the address in the default version and the associated address version are consistent. For this reason, numerical fields from the address (such as building number, postal code, and telephone numbers) are offered from the default version as default values in the additional version. Changes to these fields are reflected in the default version of the address. Only text fields can be maintained differently in the different versions.

© 2003 SAP AG. All rights reserved.

111

Exercise 6: Importing an Additional Language into the SAP System

Exercise Duration: 150 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Import a language into the SAP system and perform the associated postprocessing steps

Business Example

Your company wants to allow employees to work in the SAP system in their local language. This language has not yet been imported into the SAP system. It is therefore necessary to import the desired language into the SAP system using a language import.

The duration of this exercise is relatively long, as (in the setup for the ADM102 course), both the DEV group and the QAS group should perform the language import. The duration of the exercises should be reduced by running the first step in the exercise either overnight (as intended in the ADM102 course), or at least over the lunch break. For this reason, it can be necessary to change the order of the individual lessons in this unit, or to perform the first exercise from this lesson before you have discussed the content of the lesson with the participants.

Task

Import an additional language into your SAP system. (The specified times will roughly double if the exercises are performed simultaneously by the DEV and QAS groups.)

1. Log on to client **000** with the user that your instructor gives you. Call transaction SMLT, classify a language, and import the language package. Observe the import process using transactions SMLT, STMS, and SM37 and using the *Task Manager* at operating system level.

Continued on next page

© 2003 SAP AG. All rights reserved.

In the setup of the ADM102 course, there is a user in client 000 with the user name **`language`** and the password **`language`** that has the authorizations required for this exercise. In this course, the language CD for the language to be imported is available at the path *G:\Additional_Files\ADM102\Language_CDs\CD1*.

Caution: Ensure that you select a Western European language (such as French with the supplementation language English), as an additional code page would be required in the SAP system otherwise.

2. Reimport the language-dependent texts from the Support Packages that have already been imported. Observe the progress.
3. Supplement the imported language. Observe the progress.
4. Attempt to log on to the SAP system using the newly imported language. Which step do you still need to perform? Check the success of the language import by testing randomly.

© 2003 SAP AG. All rights reserved.

Solution 6: Importing an Additional Language into the SAP System

Task

Import an additional language into your SAP system. (The specified times will roughly double if the exercises are performed simultaneously by the DEV and QAS groups.)

1. Log on to client **000** with the user that your instructor gives you. Call transaction SMLT, classify a language, and import the language package. Observe the import process using transactions SMLT, STMS, and SM37 and using the *Task Manager* at operating system level.

In the setup of the ADM102 course, there is a user in client 000 with the user name **`language`** and the password **`language`** that has the authorizations required for this exercise. In this course, the language CD for the language to be imported is available at the path *G:\Additional_Files\ADM102\Language_CDs\CD1*.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Caution: Ensure that you select a Western European language (such as French with the supplementation language English), as an additional code page would be required in the SAP system otherwise.

a) Log on to client *000* with the user that your instructor gives you.
b) Call transaction SMLT. Choose *Language* → *Classify*. Choose, for example Language *FR* with supplementation language *EN*. Save these entries.
c) In the menu of transaction SMLT, choose *Language* → *Import Package*. Specify the path to the language CD. It is available, for example, under \\<*host name*>*Additional_Files**ADM102**Language_CDs**CD1*. Ensure that it is possible to access the CD. If necessary, create a corresponding share name at file system level.
d) Choose *Find Package(s)*, and then choose the radio button *Import in All Clients* and set *Number of Parallel Import Processes* = *4* under *Settings*.
e) Select the line with the available package and then choose, for example, *Start Immediately* and *Execute*. Confirm the subsequent message. After the message *Import scheduled*, you can return to the initial screen of transaction SMLT by choosing *Back*.
f) Use transaction SM37 to observe a job *SAP-LANGUAGE-IMP**, which starts the import process. This job runs for around a minute. You can then see, for example, up to four parallel *R3trans* processes at operating system level.
g) You can follow the progress in the import queue of your system with transaction STMS. The import takes around 25 minutes.
h) Start the language import jobs (*SAP-LANGUAGE-IMP**).
i) You can use transaction SM37 to follow these. They run for around 30 minutes. You can use the icons and the action log in transaction SMLT to check the actions during this entire time.

2. Reimport the language-dependent texts from the Support Packages that have already been imported. Observe the progress.

   a) In transaction SMLT, select your imported language and choose *Language* → *Special Actions* → *Import Support Packages*. Choose *Go to next screen*, *Go to next screen*, and *Import Language Data*. On the *Handling Repaired Objects* screen, choose *Overwrite*. You can follow progress here. This process takes (depending on the number of Support Packages to be handled) around 20 minutes.

Continued on next page

 © 2003 SAP AG. All rights reserved.

3. Supplement the imported language. Observe the progress.

   a) In transaction SMLT, select your imported language and choose *Language → Supplement Language*. On the *Secondary language and supplementation language different* screen, enter a short text, keep the settings, and choose *Execute*. On the next screen, choose *Yes* and then *Back*.

      You can observe the supplementation jobs (*SAP-LANGUAGE**) in transaction SM37 or monitor the progress directly in transaction SMLT using *Overview/Status*. The supplementation takes around 30 minutes and should be successfully completed with the message *Action completed with warnings*.

4. Attempt to log on to the SAP system using the newly imported language. Which step do you still need to perform? Check the success of the language import by testing randomly.

   a) Before you can log on using the newly imported language, you must use transaction RZ10 to adjust the language parameter *zcsa/installed_languages*. Maintain this profile, for example, in the extended maintenance of the *DEFAULT* and add, for example, *F* for French (for example *Parameter name* = **`zcsa/installed_languages`**, *Parameter value* = **`DEF`**).
   b) Save and activate your changed profile. This action requires a restart of the SAP system.
   c) After restarting the system, log on to the SAP system with your user and the new language and call a few transactions to test whether the import has been performed successfully.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the process required to install an additional language in the SAP system
- Import an additional language into the SAP system

Related Information

For more information about languages in SAP systems, see the online documentation, the guide for language imports, and under the Quick Links */globalization* and */languages* in the SAP Service Marketplace.

© 2003 SAP AG. All rights reserved.

Lesson: Code Pages and Unicode: Fundamentals

116 Lesson Duration: 40 minutes

Lesson Overview

This lesson explains the terms **MDMP** and **Unicode** and the relationships and problems that arise in their context.

Lesson Objectives

After completing this lesson, you will be able to:

- Define terms such as code page, MDMP, and Unicode, and explain their relationships
- List advantages and disadvantages of using Unicode on the SAP Web Application Server

This lesson should explain the meaning of the terms code page, MDMP, Unicode, and so on to the participants. This lesson will not teach the participants how to set up an MDMP system, or how to perform a Unicode migration. The lesson is restricted to theoretical explanation of the material. Exercises and demonstrations are not really possible, as the training system is set up neither for MDMP, nor for Unicode. This would also be beyond the scope of the course.

Business Example

Multiple languages are installed in your SAP system, all of which are on the Latin-1 code page (Western European languages). You now want to provide support for employees from a country whose language is not on this code page.

Prerequisites and Terms

SAP supports more than 30 languages on non-Unicode systems. However, due to the technical limitations of code pages, only certain language combinations can be used without restrictions. The following options exist:

- Single standard code pages, which support certain numbers of languages: The number and combination of the supported languages cannot be changed, but there are not restrictions for users.
- “Blended” code pages, which are SAP’s own code pages and contain one or more standard code pages: This increases the number of available language combinations. From a functional point of view, a

 © 2003 SAP AG. All rights reserved.

blended code page system uses a single code page. New installations of blended code page systems are **not** supported by SAP. MDMP installations are now preferred.

- Multi-Display/Multi-Processing (MDMP), which allows dynamic switching of code pages on the application server: This means that any combination of standard code pages is possible in one system. The logon language determines the code page that is active for each user. There are a number of restrictions, for example, with regard to mixing data from different code pages.

The character set describes the storage of a character. The options are single byte character sets (one byte per character), double byte character sets (one or two bytes per character), and Unicode character sets (one or more bytes per character). The code page determines the assignment of a character to a hexadecimal value. There are various standards for this, such as ISO and Microsoft. SAP uses its own four digit identifiers, such as 1100 and 8500.

The “locale” is a set of rules for a national language. It describes the conversion of lowercase characters to uppercase characters, if they exist. It is also responsible for the national alphabetical sorting. The name of the locale is constructed from <language>_<country>.<code page>, such as ru_RU.ISO88595 for Russian, Russia, code page ISO8859-5.

The **common character set** is shared by all code pages. This contains the seven-bit US ASCII characters (US7ASCII), that is, lowercase and uppercase letters from A-Z, the digits 0-9, and the special characters ” . , + - ( ) < > : *.

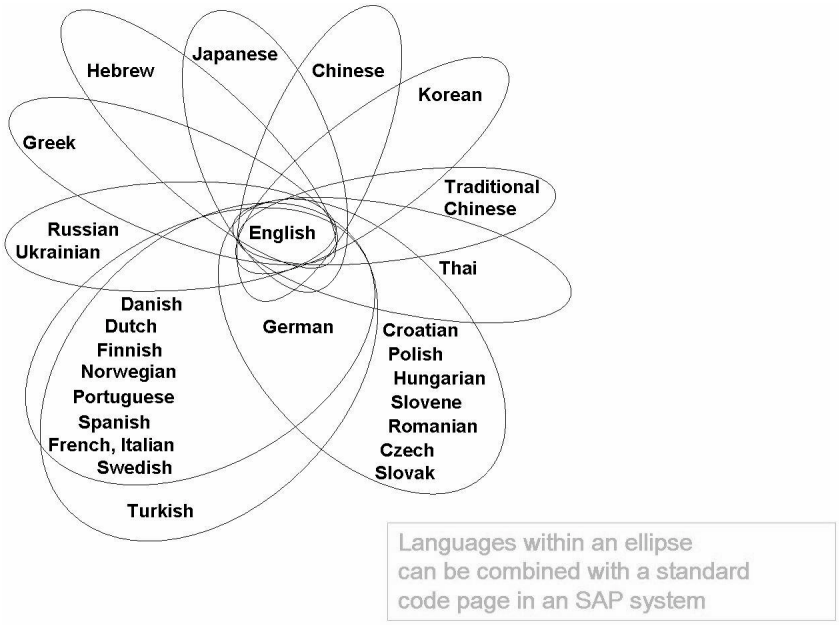


Figure 46: Standard Code Page in an SAP System

© 2003 SAP AG. All rights reserved.

Multiple Code Pages in an SAP System: MDMP

Different languages are required in an SAP system that is used internationally. This could require the use of different code pages in the same SAP system. Users want to be able to work in their national language and generate data in a language, or even multiple languages (by logging on with different language keys). They want to be able to view data that they or other users have created. This can be a problem if users want to view data that was created with a language that belongs to a different code page than the language currently in use. In this case, problems arise due to the different locales. That is, even viewing data of this type will generate "nonsense" search helps due to the different rules for lowercase to uppercase conversion, meaning that other users can no longer view some data. Changing data that was created under a different code page can lead to its destruction. It is therefore important to restrict globally used data to the characters shared by all code pages, the **common character set** (US7ASCII).

Does a single user require all languages?

- **A user requires only one language**
- **A translator requires two languages**
- **Global view?**

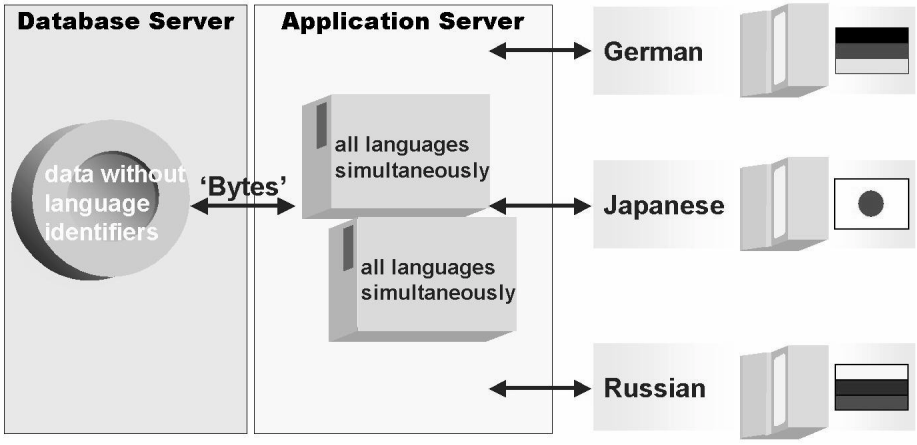


Figure 47: Need for Different Languages in an SAP System

This graphic emphasizes the fact that the data in the database of the SAP system is always stored as bytes (hexadecimal code), regardless of which language on which code page the user uses to log on and to view the data at his or her front end. These bytes are displayed differently at the front end and interpreted differently by the SAP system depending on the code page in use.

© 2003 SAP AG. All rights reserved.

Until now, SAP systems used different code for encoding the characters of different scripts, such as ASCII, EBCDI, or double-byte code pages.

- American Standard Code for Information Interchange (ASCII) encodes each character using 1 Byte = 8 Bit. This means that a maximum of 2^8 = 256 characters can be represented, and these characters are assigned the combinations [00000000, 11111111]. Code pages that can be used are, for example, ISO88591 for Western European scripts, or ISO88595 for Cyrillic scripts.
- Extended Binary Coded Decimal Interchange (EBCDI) also encodes each character using 1 byte, which in turn allows 256 characters to be mapped. EBCDIC 0697/0500 is an old IBM format, which is, for example, used on AS400 for Western European scripts.
- Double-byte code pages require one or two bytes for each character. This means that 65536 combinations can be created, of which only 10000 - 15000 are normally occupied with characters. Examples of double-byte code pages are SJIS for Japanese and BIG5 for traditional Chinese script.

The instances of the SAP system process the -- possibly different -- code pages of individual users simultaneously. They switch the code page depending on the user context. The logon language therefore determines the code page that is active for each user. This procedure is called Multi Display, Multi Processing (MDMP). In the database, however, the characters are code-page neutral; that is, they are stored without any indication of the code page under which they were created. It is therefore no longer possible, in general, to determine the code page when assessing the data. MDMP has emerged as the *best practice* solution. MDMP is used in production in many SAP installations for SAP R/3, SAP CRM, SAP APO, SAP BW, and so on.

When using individual code pages, it is also necessary that the operating system of the front end computer be working with a “matching”, that is, a compatible, code page. Changing the code page of the front end computer can require a reboot of the computer or, in the worst case, a reinstallation of the operating system. The SAP GUI itself also works with a code page

© 2003 SAP AG. All rights reserved.

setting. This must be set correctly for each logon. This setting can be made in SAP Logon. This can mean that multiple SAP Logon entries may be required for logging on to an SAP system with different languages.

Note: *Don't confuse the system.* Some language-dependent data is stored with a language key. It is important that this language key matches the code page for the logon language, as this data is interpreted in a number of places due to the assigned code page and its locale. You should ensure when logging on to the system that the code page of the operating system of the front end computer, the SAP GUI code page, and the logon language are compatible with each other. This is **not** checked by the SAP system's logon program. However, you can implement a check of this kind using a user exit.

© 2003 SAP AG. All rights reserved.

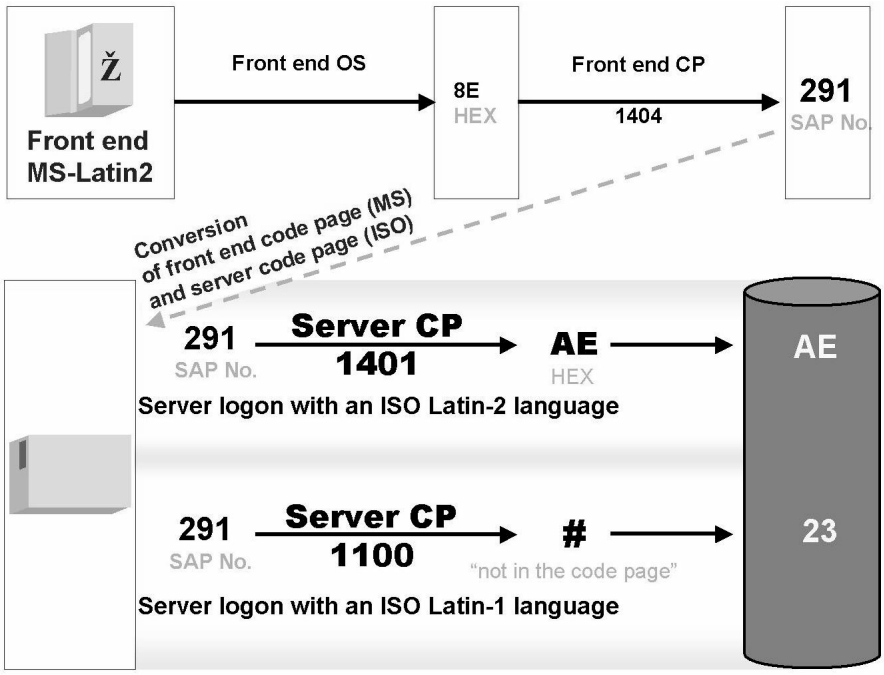


Figure 48: Logging On to the SAP System: "Don't Confuse the System"

This graphic shows the case of an employee that logs on to his or her (Microsoft Windows) front end computer in a Latin-2 language (such as Polish). The code page (Latin-2) associated with the language is set on his or her front end computer, and the mapped (Polish) character corresponds to the (Microsoft Windows) hexadecimal value 8E. On the front end, this character is assigned to the SAP character with the number 291 (every character has a unique SAP number). If the employee is logged on to the SAP system in a Latin-2 language (such as Polish) (the upper example in the graphic, code page 1401 for Latin-2), the SAP character 291 is correctly interpreted and is ultimately stored in the database under the hexadecimal value AE. However, if the employee is logged on to the SAP system using a language of the Latin-1 code page (such as French; lower example in the graphic), the SAP character with the number does not exist in the code page at all is stored as # (hexadecimal value 23) in the database.

The following table shows languages that are supported for non-Unicode systems. Languages that use the same code page can be used together with no restrictions. SAP also supports SAP blended code pages and Multi-Display/Multi-Processing (MDMP), which allows any combination of these languages to be used in a single installation. English only requires seven-bit US ASCII characters, and can therefore be combined with any other code page. However, English can only ever be used with one code page.

 © 2003 SAP AG. All rights reserved.

Supported Languages and Their SAP Code Page

Language	ISO 639-1 Language Code	SAP Language Code	SAP Code Page (ASCII)
Afrikaans	AF	a	1100, 1610
Arabic	AR	A	8700
Bulgarian	BG	W	1500
Catalan	CA	c	1100, 1610
Chinese	ZH	1	8400
Chinese (traditional)	ZF	M	8300
Danish	DA	K	1100, 1610
German	DE	D	1100, 1401, 1610
English	EN	E	1100
Estonian	ET	9	1900
Finnish	FI	U	1100, 1610
French	FR	F	1100, 1610
Greek	EL	G	1700
Hebrew	HE	B	1800
Indonesian	ID	i	1100, 1610
Icelandic	IS	b	1100
Italian	IT	l	1100, 1610
Japanese	JA	J	8000
Korean	KO	3	8500
Croatian	HR	6	1401
Latvian	LV	Y	1900
Lithuanian	LT	X	1900
Malaysian	MS	7	1100, 1610
Dutch	NL	N	1100, 1610
Norwegian	NO	O	1100, 1610
Polish	PL	L	1401
Portuguese	PT	P	1100, 1610

© 2003 SAP AG. All rights reserved.

Reserved - Customer	Z1	Z	
Romanian	RO	4	1401
Russian	RU	R	1500
Swedish	SV	V	1100, 1610
Serbian	SR	0	1500
Serbo-Croat	SH	d	1401
Slovakian	SK	Q	1401
Slovene	SL	5	1401
Spanish	ES	S	1100, 1610
Thai	TH	2	8600
Czech	CS	C	1401
Turkish	TR	T	1610
Ukrainian	UK	8	1500
Hungarian	HU	H	1401

A Code Page for All Characters: Unicode

Character sets can be used individually to cover every language that is relevant to SAP. However, difficulties arise if you want to mix texts from different incompatible character sets in one central system. Exchanging data between systems with incompatible character sets can also lead to completely undefined situations.

© 2003 SAP AG. All rights reserved.

Unicode means

- Character encoding for all languages relevant for business and their characters worldwide
- Every character has a unique representation ('Unicode code point')
- Notation: U+nnnn (nnnn are hexadecimal digits)

ما هي الشفرة الموحدة "يونيكود"؟ Arabic
什麼是Unicode(統一碼/標準萬國碼)? Chinese (traditional)
What is Unicode? English
რა არის უნიკოდი? Georgian
Τι είναι το Unicode? Greek
यूनिकोड क्या है? Hindu
Cos'è Unicode? Italian
ユニコードとは何か？ Japanese
유니코드에 대해? Korean
Что такое Unicode? Russian

Figure 49: What is Unicode?

The solution to the problem is to use a code page that contains all characters of all languages that are relevant for business. This code page is called Unicode (ISO/IEC 10646) and consists of at least 16 bits = 2 bytes or, alternatively, of 32 bits = 4 bytes for each character. Although the conversion effort for the SAP kernel and applications is enormous, the introduction of Unicode brings significant long-term advantages:

- The entire Internet is based on Unicode, which therefore becomes a basic prerequisite for internal competitiveness.
- With Unicode, all SAP users can install a central system that covers the business processes of the entire world.
- Even companies with different distributed systems usually want to prepare consolidated worldwide corporate information. Without Unicode, this is only possible to a limited extent.
- Under Unicode, multiple languages can be used simultaneously on one front end.
- Unicode is required for a cross-application data exchange without data loss due to incompatible character sets. XML is an example of a possible format for displaying documents in the Internet in this situation.

What does Unicode mean for an SAP system? ABAP programs must be changed everywhere an explicit or implicit assumption is made about the internal length of a character. This means that a new level of abstraction is reached that allows the same programs to run both in a traditional

 © 2003 SAP AG. All rights reserved.

system and in a Unicode system. It also means that if the character pool of the Unicode character set is increased further, it is possible to choose an internal representation of the Unicode characters with two or four bytes.

SAP's aim is complete internationalization with Unicode and better integration of SAP technology with other Internet technologies. As of SAP Web AS 6.20, all languages defined in ISO-639 can be used as language tags in a Unicode system. This means that characters from practically any language and any script can be stored in the database. Note, however, that this does not mean that there are localized versions or translations for all languages.

Unicode is an international standard that assigns ever character from every language and every script a unique number and therefore makes the presence of multiple code pages in one installation superfluous. With Unicode, all languages and scripts from the Unicode standard can be used without restrictions. As the standard encoding of all Internet communication (HTML, XML, Java, and so on), Unicode improves data communication across the boundaries of system landscapes and paves the way for closer integration of ABAP with these Internet technologies. At the same time, Unicode eliminates the danger of data loss due to code page incompatibilities.

Unicode is supported as of version 6.10 of the ABAP programming language.

- **UTF-16**
  Unicode Transformation Format, 16 bit Encoding
  - Fixed length, 1 character = 2 bytes, surrogates 2*2 bytes
  - Platform-independent byte sequence
  - 16 bit "Alignment Restriction" (requires even addresses)
- **UTF-8**
  Unicode Transformation Format, 8 bit Encoding
  - Variable length, 1 character = 1...4 bytes
  - Platform-independent
  - No "Alignment Restriction"

Character	Unicode Code Point	UTF-16 big endian	UTF-16 little endian	UTF-8
a	U+0061	00 61	61 00	61
ä	U+00E4	00 E4	E4 00	C3 A4
α	U+03B1	03 B1	B1 03	CE B1
會	U+3479	34 79	79 34	E3 91 B9

Figure 50: Unicode: UTF-16 and UTF-8

© 2003 SAP AG. All rights reserved.

Currently, more than 90, 000 characters are defined in the Unicode standard, and there is additional space for more than a million characters. The Unicode code page can be divided into blocks of characters, as the above figure shows. “U+” is a naming convention for Unicode scalar values (the number has hexadecimal notation). Endian describes the byte sequence for storage, with reducing place values from left to right (big endian) or from right to left (little endian).

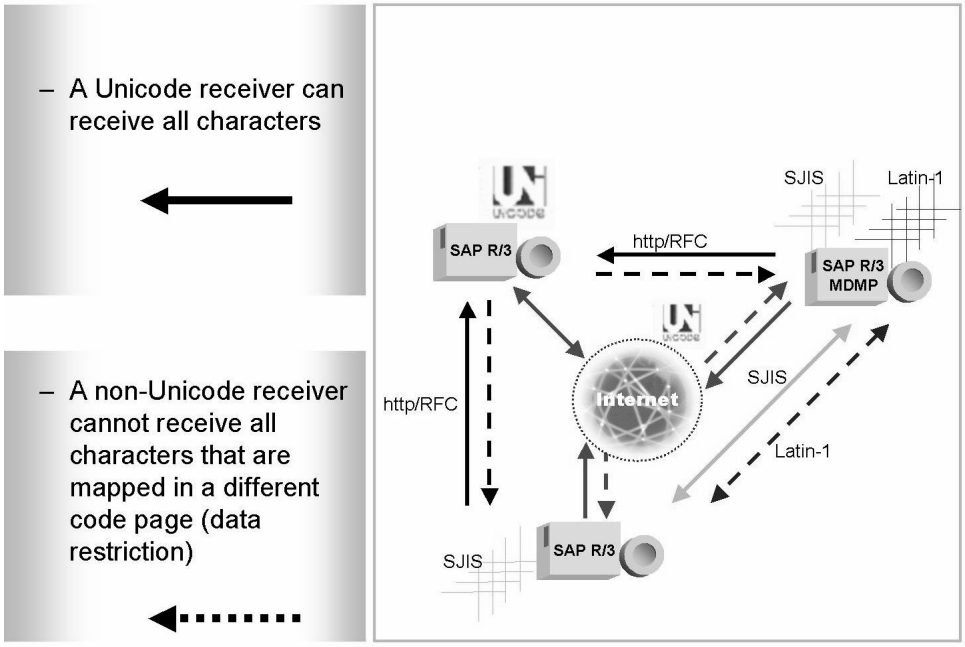


Figure 51: Communication With and Without Unicode

In this graphic, the solid lines do **not** mean indicate a Unicode receiver, but rather than both systems can receive all characters on the specified code page (SJIS refers to the Japanese code page here). As the lower SAP system is not an MDMP system and only has the SJIS code page, it can exchange all SJIS characters with the SAP R/3 System in the top right, but not all of the Latin-1 characters.

Using national standard code pages can cause restrictions, particularly when interfaces (such as ALE, EDI, BAPI, RFC, CPI-C, and XML) are used. While a Unicode system can receive all characters, a non-Unicode system may process characters from another code page incorrectly. The interface of a non-Unicode system should therefore be restricted to receiving data from its “own” code page.

© 2003 SAP AG. All rights reserved.

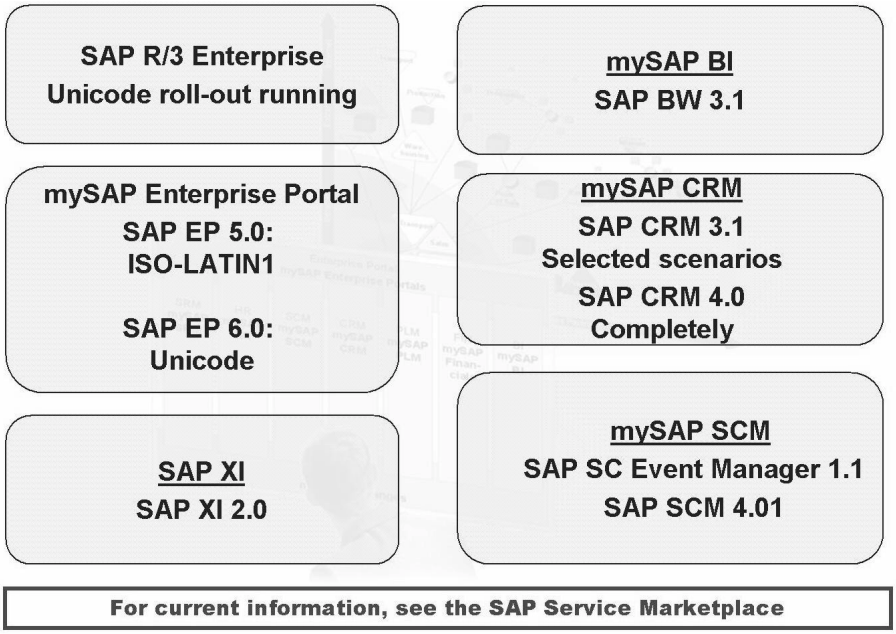


Figure 52: Availability of Unicode

How Can an Existing SAP System Be Migrated to Unicode?

IN an MDMP system, multiple code pages are used simultaneously on the application server. The logon language therefore determines the code page that is active for each user. As the active code page changes again and again, tables that contain non-ASCII characters must have a language key to ensure that the data is processed with the correct language. This is not, however, the case for all tables. The problem is, therefore, to determine the code page from which individual pieces of data originate.

© 2003 SAP AG. All rights reserved.

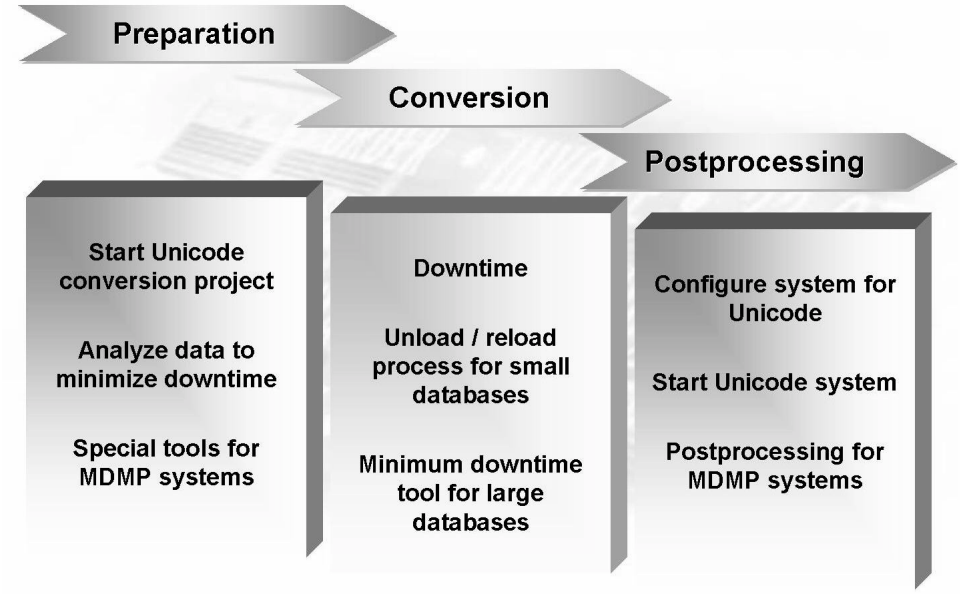


Figure 53: Steps of the Unicode Conversion

During the Unicode conversion of an SAP R/3 database, text data from the code page(s) of the original system is converted into Unicode. In an MDMP system, the language key information must be used during the conversion to avoid data errors.

It is possible that non-ASCII characters were entered in tables without language keys, or that the language keys are not available, for example in tables that were created by customers. The conversion of tables of this type represents a problem. The phases of preparation and post-conversion processing are used to avoid data loss in these problem cases.

Prerequisites:

- The SAP component that you want to install must be based on Unicode.
- Unicode is supported on a project basis for SAP Web AS 6.10 and above
- Older systems must first be upgraded to at least SAP Web AS 6.10; the preparation phase can be performed as of SAP Basis 4.6C.
- R3load version 6.20 should be used, as MDMP handling is supported by this program (R3Load is backward compatible and version 6.20 can also be used for SAP Web AS 6.10 installations).

Process:

1. Conversion preparation: Collect information about the database (language keys, tables, and so on). This information is stored in a control file for each table in the database, and is then stored in the

 © 2003 SAP AG. All rights reserved.

control table. The migration tools read the information from the control table during the import and export. The information is also used during post-conversion processing.

By performing this process, you prepare the system copy and create the control files for all tables in the database. The control files contain the information that is required to ensure a consistent conversion to Unicode during the system copy. The descriptions are stored in the control table. The prerequisite for this is SAP Basis 4.6C or above. Use transaction SPUMG to prepare for the conversion. To prepare for the conversion, you can divide the tables into a number of categories:

- Code page-dependent with language key: The table consists of text data that contains non-ASCII characters.

  Dependent on one code page: All rows of the table have language keys that belong to languages that use the same code page. The entire table can be converted with one code page.

  Dependent on multiple code pages: The rows of the table contain language keys that belong to languages that use different code pages. The table must be converted with the aid of multiple code pages.

- Code page-dependent with no language key: The table consists of text data that contains non-ASCII characters. The text data is therefore language-dependent. The language key must be specified.

  a) Code page guess function: Heuristic attempt to determine the correct code page.

  b) System vocabulary: Words from code-page-dependent tables without language keys are entered in the system vocabulary. The scanner then searches for these word in tables with language keys to determine the correct code page.

  c) Post-migration processing

- ASCII: The table consists only of text data that contains ASCII characters. The table is not language-dependent and it is therefore irrelevant which code page is used for the conversion.

2. System copy to Unicode: Export and import the database. The actual conversion takes place during the export. For more information, see the Unicode Installation Guide.

3. Post-conversion processing: Use the information collected during the conversion preparation to repair data that was not converted correctly. Use transaction SUMG to do this.

 © 2003 SAP AG. All rights reserved.

Unicode: Conclusion

SAP components are Unicode-compatible as of SAP Web AS 6.10. For more information, see SAP Note 79991. Unicode offers a range of advantages for systems that are operated internationally. However, when doing so, you should consider the fact that representing non/USA7ASCII characters with multiple bytes requires additional space in the database. Statistically, you should expect a Unicode system to have a database around 30% larger. This, of course, has direct consequences for the length of time required for a data backup or a data restore in the case of an error. Due to the larger quantity of data, to maintain the performance of the system, you should expect to provide around 30% more CPUs, main memory (RAM), network bandwidth, and so on. This is undoubtedly of secondary importance in comparison to the advantages of Unicode for a system that is used internationally. However, it also means that Unicode should also only be used if it is really necessary. It should not be implemented solely on the basis of a desire to be "up to date"; to do so would be to misunderstand the situation. SAP will continue to support Unicode and non-Unicode systems. A migration from a non-Unicode system to a Unicode system is possible at any time (as of SAP Web AS 6.10). This migration is a separate step; that is, it is not just done at the same time as an upgrade, for example.

Appendix: Unicode Glossary

byte-like

The ABAP data types X and X-STRING

code

Binary encoding of letters, digits, and special characters

code page

Selection of the encoded characters for a selected environment

data cluster

Combination of multiple data objects (fields, structures, tables)

DDIC

ABAP data directory

endian

The byte sequence of a number is stored in memory with reducing place values from left to right (big endian) or from right to left (little endian)

front end

Presentation server

© 2003 SAP AG. All rights reserved.

kernel

Basis functions coded in the programming languages C and C++

NUP (Non-Unicode Program)

Program for which the Unicode indicator is not set

NUS (Non-Unicode System)

System in which the binary encoding of characters uses one byte

remaining length

The remaining length of a field or of a character-type structure is calculated from the total length minus the offset

surrogate area

Character extension for all characters that are not contained in standard Unicode with a maximum of 64,000 characters; in addition to the bit pattern from the standard Unicode, two additional bytes are read from this area

Unicode fragment view

Splits structure into data areas of the same type

UP (Unicode Program)

ABAP program for which the Unicode indicator is set

US (Unicode System)

System in which the binary encoding of characters uses two or four bytes

UTF-8, UTF-16

Data format for communication and data exchange

XML

Extensible Markup Language: Markup language for displaying any documents in the Internet

character

Letter, digit, or special character

character-type

The ABAP data types C, N, D, T, and STRING

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

Short repetition of the terms code page, MDMP, and Unicode and their meanings and relationships

Discussion Questions

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

- What is a code page?
- When should I use MDMP?
- When can and when should I migrate to Unicode? How does that work in principle?

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Define terms such as code page, MDMP, and Unicode, and explain their relationships
- List advantages and disadvantages of using Unicode on the SAP Web Application Server

Related Information

For more information about the topics of globalization and Unicode, see the SAP Service Marketplace under the Quick Links */globalization* and */unicode*.

© 2003 SAP AG. All rights reserved.

Unit Summary

You should now be able to:

- Recognize some effects of a centralized or decentralized implementation, the possibility of using multiple time zones in one SAP system, and the need for country versions
- Outline the process required to install an additional language in the SAP system
- Import an additional language into the SAP system
- Define terms such as code page, MDMP, and Unicode, and explain their relationships
- List advantages and disadvantages of using Unicode on the SAP Web Application Server

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

133

Test Your Knowledge

1. Which statements are correct with regard to globalization?

   Choose the correct answer(s).

   - ☐ A The logon language for the SAP system is determined by the country version.
   - ☐ B A country version must always be imported as an add-on before it can be used.
   - ☐ C Time zones can be attached to a user master record or an object.
   - ☐ D The time zone from the user master record is taken into account in all SAP Basis/SAP Web AS transactions.
   - ☐ E Before time zones can be used, they must be defined in the system, together with any daylight saving time definition.
   - ☐ F A global company must essentially be mapped as a single client in a single SAP system, as otherwise no integrated business processes are possible.

2. What must be taken into account when importing an additional language into an SAP system?

   Choose the correct answer(s).

   - ☐ A After the actual import of the language package, language-dependent texts from Support Packages that have already been imported must also be imported using transaction SMLT.
   - ☐ B The Support Packages that had been imported into the SAP system at the time of the language import must be completely imported again using transaction SPAM.
   - ☐ C Languages that are not completely translated require a supplementation language to fill translation gaps.
   - ☐ D All available languages are already available in the database of the SAP system and you must now use transaction SMLT to make them known in the system.
   - ☐ E You can use transaction SMLT to perform the language import in parallel using multiple R3trans processes.

© 2003 SAP AG. All rights reserved.

3. Which statements are correct with regard to code pages?

Choose the correct answer(s).

- ☐ A Every language requires a separate, individual code page.
- ☐ B Multiple code pages can be operated simultaneously on an instance in one client in an SAP system.
- ☐ C With Unicode, all characters that are relevant for business can be mapped in one single code page. MDMP is then no longer required.
- ☐ D During an upgrade from SAP R/3 4.x to SAP Web AS 6.20/SAP R/3 Enterprise, an SAP R/3 system is automatically converted to Unicode. This is known as Unicode migration (Unicode Switch Upgrade).
- ☐ E With MDMP, one character is basically represented by one byte.
- ☐ F Unicode uses two bytes to represent every character.

© 2003 SAP AG. All rights reserved.

135

Answers

1. Which statements are correct with regard to globalization?

   Answer: C, E

   Centralized and decentralized system configurations with their hybrid forms, country versions, and time zones are some aspects that must be considered with a global system.

2. What must be taken into account when importing an additional language into an SAP system?

   Answer: A, C, E

   You can use transaction SMLT to import additional languages with language-dependent texts into an SAP system. You do not need to reimport Support Packages using transaction SPAM; the Support Packages are handled using transaction SMLT.

3. Which statements are correct with regard to code pages?

   Answer: B, C

   If you use MDMP, multiple code pages can be operated simultaneously in one instance of an SAP system. If you use Unicode, characters are stored with a length of up to four characters.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

Unit 3

137

Technical Aspects of External Communication

This unit consists of three lessons. However, the focus is clearly on the middle lesson. In the first lesson, you should explain the basic principles of SAPconnect, and you can then build on this in the second lesson. The second lesson on STMP connection is an application of SAPconnect. There have been a number of changes in this area with the introduction of SAP Web AS and its support for SMTP. The last lesson on telephone integration using SAPphone is independent of the other two lessons, as it is not based on SAPconnect and should only be regarded as an overview. This lesson could be of particular interest for participants from the CRM environment (Customer Interaction Center).

Unit Overview

The connection of external communication systems is important for many business processes.

This unit discusses the connection of different communication systems, such as fax or mail servers, to SAP systems using SAPconnect. You will also obtain an insight into integrating telephone equipment with SAP systems using SAPphone. This is used especially in the CRM environment.

Unit Objectives

After completing this unit, you will be able to:

- Outline the message flow in SAP systems
- Describe the function of SAPconnect
- State the steps required to connect the SAP system to a mail server
- Configure the SAP system for connection to a mail server
- Outline the architecture of telephone integration in SAP systems

© 2003 SAP AG. All rights reserved.

- Make settings on the SAP Web AS that are required to connect it with the telephone system

Unit Contents

Lesson: Communication with SAPconnect 165
Procedure: Creating a Node for Sending Remote Mail to a Partner System 171
Lesson: Communication Using SMTP 175
Exercise 7: Sending and Receiving E-Mails with SAP Web AS 189
Lesson: Telephone Integration with SAPphone 205

© 2003 SAP AG. All rights reserved.

Lesson: Communication with SAPconnect

138

Lesson Duration: 35 minutes

Lesson Overview

In this lesson, you will learn the principles of SAPconnect.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the message flow in SAP systems
- Describe the function of SAPconnect

This lesson should first explain the reasons for the way in which external communication (and therefore SAPconnect) fit into the overall picture: applications are to be offered the service of creating and processing documents in a generic way (internally and externally).

A short overview of the fundamentals of SAPconnect then follows. For this reason, the lesson contains no exercises. However, you can have participants perform the procedure at the end of this lesson as an exercise, if there is enough time available. You should ensure that you are familiar with transaction SCOT and have a good command of the functions mentioned. For a great deal of additional useful information, see the BC-SRV-COM area of the SAP Library.

Business Example

As a member of the system administration team, you want to enable the connection of SAP systems to external communication services.

Overview of the Message Flow in SAP Systems

The SAP system offers application developers many ways in which to create and process messages. For example, there could be a need to automatically send a confirmation message to the sold-to party after the creation of a sales order.

As the instructor for the ADM102 course, you do not need to know all of the routes exactly – ultimately this topic also belongs in part to the applications and to development. What you should ensure you communicate: Later in the course, messages are produced “manually” in

© 2003 SAP AG. All rights reserved.

the Business Workplace and then sent. In practice in customer systems, the applications that automatically generate large numbers of messages – which may be sent as external e-mails – are important.

The graphic shows the different communication streams that a developer can use. There are, of course, a large number of new developments (and not just new names) in this area. These can be seen, with their predecessors, in the graphic. With a delivered application, the developer at SAP decides the approach taken; for their own developments, customers have to decide this. For more information about the topic of "message control", see the *Message Control (CA-GTF-BS)* area of the online documentation for SAP R/3 Enterprise.

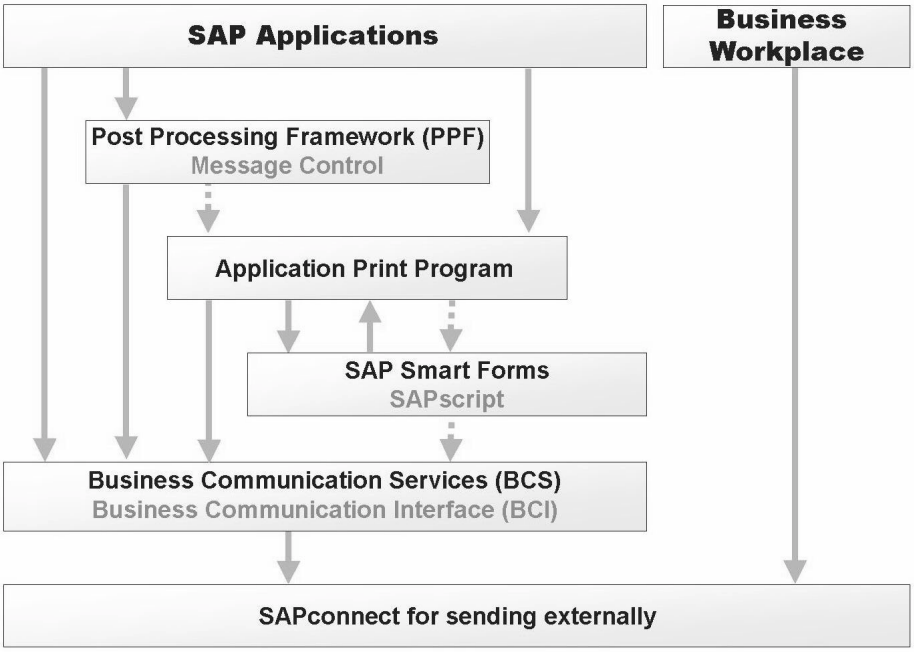


Figure 54: Message Flow in the SAP System

Business Communication Services (BCS) are of central importance in message processing. As of SAP Web AS 6.10, these services allow application developers to integrate sending messages internally and externally in their applications without any great effort. In addition to controlling sending and receiving, BCS takes on extensive status handling and makes available all information about the sending of an application object. Business Communication Services are the successor to the Business Communication Interface (BCI) and offer the advantage of an object-oriented interface.

© 2003 SAP AG. All rights reserved.

The sending of messages from SAP applications can be performed directly using BCS. Alternatively, the Post Processing Framework (PPF) can also be addressed first. The PPF provides a uniform interface for the conditional generation of actions (such as printing delivery notes, faxing order confirmations, or triggering approval processes). The PPF is the successor to message control and provides a greater range of functions, simpler connection to the applications, and greater flexibility than its predecessor. BCS (and its predecessor, BCI) can also be addressed by the tool for forms, SAP Smart Forms (and its predecessor, SAPscript).

In addition to automated message creation and processing, you can, of course, also create messages manually using the Business Workplace (previously known as SAP Office). The Business Workplace provides a uniform working environment in which every SAP user can perform his or her part of the business and communication processes in the company.

In the case of external messages, BCS or the Business Workplace forwards these to SAPconnect.

What Can SAPconnect Do?

SAPconnect is the central interface for external communication in SAP systems. SAPconnect supports the use of telecommunication services such as fax, text messages (pages/SMS), Internet e-mail, and X.400, and sending to printers and between different SAP systems.

SAPconnect allows you to connect external communication components to the SAP system.

SAPconnect provides a direct connection to the Internet using the SMTP plug-in of the SAP Web Application Server. The SMTP plug-in for Internet e-mail is available as of SAP Web AS 6.10.

Communication possibilities with SAPconnect:

© 2003 SAP AG. All rights reserved.

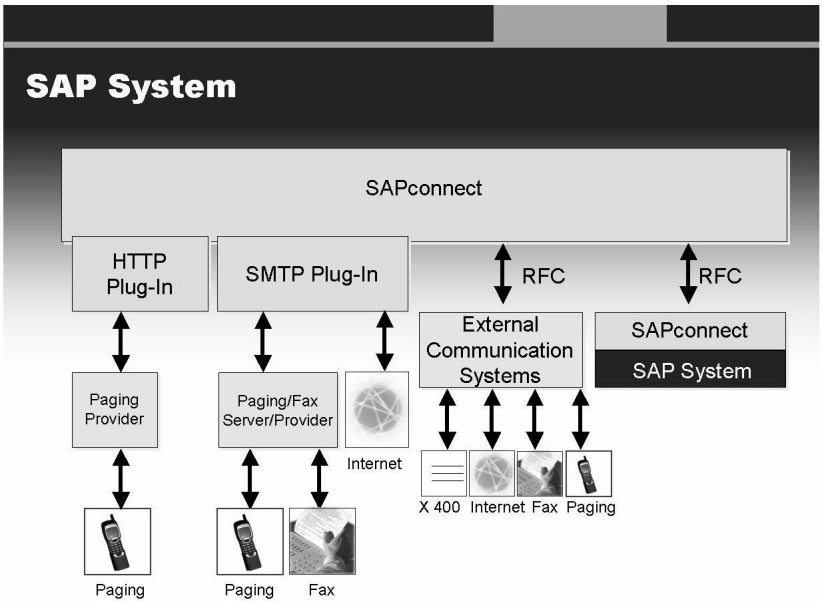


Figure 55: Communication Possibilities with SAPconnect

As you can see from the figure, you can reach the same target (such as sending mail in the Internet) in different ways. It has been possible to use external communication systems for many years; however, the use of plug-in technology requires at least SAP Web AS 6.10, and newer plug-ins may have even higher requirements (such as SAP Web AS 6.20).

SAPconnect: Transaction SCOT

Use transaction SCOT to configure SAPconnect. You can set up communication types here (such as fax, Internet e-mail, and so on) by creating and configuring communication nodes. You can use various views of the communication infrastructure in transaction SCOT. You usually start with the *System Status* view.

© 2003 SAP AG. All rights reserved.

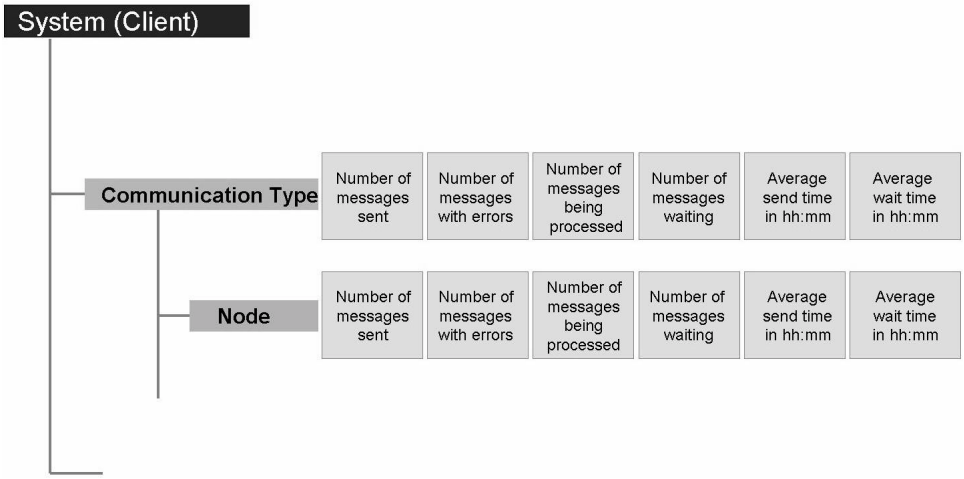


Figure 56: The *System Status* View

To be able to send messages using a communication type, you must perform the following steps:

1. You must activate the communication type (by choosing *Settings* → *Communication Methods*). SAPconnect is the communication method assigned by default for the communication type. Do not be confused: the only alternative to SAPconnect as the communication method here is *NONE*. This would explicitly deactivate a communication type.

In earlier releases, there was an alternative to SAPconnect, the predecessor technology SAPcomm. This obsolete technology is now no longer available.

---

2. You require at least one configured (communication) node for the communication type if you want use it to send documents.
3. All programs that may be required must be available and configured.

Basic Process of Communication Using SAPconnect

The following list clarifies the process of a generic communication step in SAPconnect.

1. A message is created, for example as a WebFlow item, as a message in the Business Workplace (transaction SBWP) or by an application.
2. The message is sent in accordance with the selected communication type and the selected address area assigned to a node.

© 2003 SAP AG. All rights reserved.

3. The send process (report **RSCONN01** - should run periodically in the background) starts and transfers the message to the SMTP plug-in or to an RFC destination.

Note: You can easily schedule the send process periodically in transaction SCOT, by choosing *Views* → *Jobs*.

Caution: SAPconnect is set up **cross-client**. Although the nodes that you use for communication use RFC connections that are valid system-wide, they are only known within one client.

Note: The following roles are required to administer SAPconnect: *SAP_BC_SRV_COM_ADMIN* and *SAP_BC_SRV_GBT_ADMI*. An RFC user that can receive messages requires a profile with authorization for *S_A.SCON*.

The figure below shows a completely configured node for sending remote mail to an SAP system.

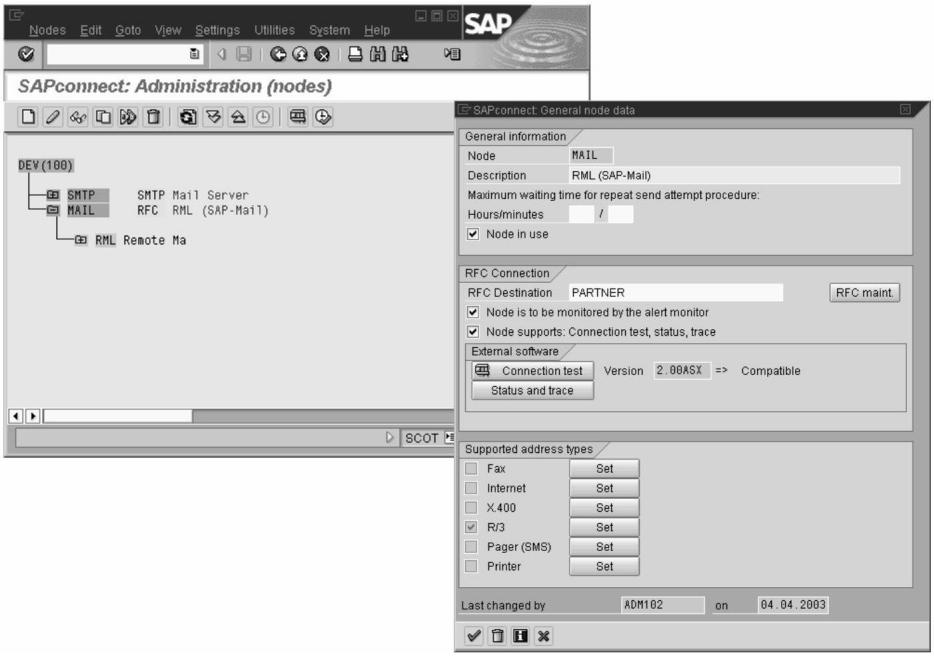


Figure 57: Setting up a Node for Remote Mail Sending

© 2003 SAP AG. All rights reserved.

143

Creating a Node for Sending Remote Mail to a Partner System

1. Start transaction SCOT.
2. Choose *View→ System Status*.
3. Choose *Create* (**Ctrl+F5**).
4. Enter a name for the node (such as **`Remote Mail`**) and a description.
5. Create an RFC node.
6. Specify the RFC connection **`PARTNER`** or another RFC connection that points to a client of another SAP system.
7. Choose the address type **`R/3`**.
8. Specify **`<SID>:100:*`** for the address type. **<SID>** = DEV, or QAS (ID of your partner system).
9. Leave the conversion rules and the screen for restricting the send time on the default settings.
10. Do not set any additional address types.
11. Keep the default settings for failed send attempts.
12. The node is to be monitored by the alert monitor (you may have to initially restart the data collection method in transaction SCOT by choosing *Utilities → Alert Monitor → Start data collection method* so that the node is displayed in the alert monitor).
13. Declare the node in use and check the *Node supports: connection test, status, trace* checkbox. Perform the test.

Result

The **`Remote Mail`** node is ready to use.

Test this node by following the procedure below:

1. Start transaction SBWP.
2. Choose *New Message*.
3. Enter a title and content for your mail and specify the receiver as follows: **system_name:client:user**. The system name corresponds to the SID of your partner system (in the setup for the ADM102 course; therefore QAS for the DEV groups and DEV for the QAS groups) and the client is the ID of the client in which the specified user exists, such as **`QAS:100:ADM102-30`**. Specify the recipient type as **`R`** (= remote mail address). Choose *Send*.

Continued on next page

 © 2003 SAP AG. All rights reserved.

4. Start the send process for the address type **RML** in transaction SCOT.

5. Check that the mail has been received in the partner system using transaction SBWP.

Frequently Asked Questions

How do I configure SAPconnect in the context of system monitoring so that an auto-reaction (sending mail) runs correctly?

If an alert occurs in a system according to transaction RZ20, and a correctly set up auto-reaction method is assigned, the following happens:

1. The alert is marked red in transaction RZ20 a short time after it occurs.

2. This red alert must be recognized by the auto ABAP *SAPMSSY8*, which runs every five minutes. The auto ABAP then starts the assigned auto-reaction method.

   Caution: As *SAPMSSY8* is executed in client 000, SAPconnect must also be correctly configured in client 000.

3. After the auto-reaction method is active, there is a message that is ready to send in client 000. This message is sent the next time that report **RSCONN01** runs. That is, the send process must be periodically scheduled.

   Caution: Documents are only sent correctly if the send user (which is declared in the auto-reaction method) is assigned a mail address in client 000.

Can I connect a *sendmail* to SAP systems?

Search for the term **sendmail** in the SAP Library. You will find detailed instructions for connection.

How can I connect MS Exchange Server to my SAP systems?

Search for the term **SXC** (SAP Exchange Connector) in the SAP Library. You will find detailed descriptions of the procedure.

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

Implementation areas of SAPconnect

Discussion Questions

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

Discuss the possible implementation areas of SAPconnect with the participants. For example, you could ask the participants about the areas in their company in which SAPconnect is used or could be used.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the message flow in SAP systems
- Describe the function of SAPconnect

Related Information

- Online documentation for SAP Web AS 6.20: *mySAP Technology Components → SAP Web Application Server → Basis Services / Communication Interfaces → Business Workplace and Services → Generic Business Tools for Application Developers (BC-SRV-GBT) → Business Communication Services - Sending from SAP Applications as of Release 6.10*
- Online documentation for the SAP Web AS 6.20: *mySAP Technology Components → SAP Web Application Server → Basis Services / Communication Interfaces → Communication Interfaces (BC-SRV-COM) → SAPconnect*
- SAP Note 455127: *E-Mail (SMTP) in different SAP releases*
- SAP Note 17194: *Telefax in various SAP releases*
- SAP Note 455129: *Paging/SMS in different SAP releases*
- SAP Note 312690: *SAPconnect: Collective Note*
- SAP Note 455140: *Configuration e-mail, fax, paging/SMS via SMTP*
- SAP Note 598718: *SAPconnect - Performance*

© 2003 SAP AG. All rights reserved.

Lesson: Communication Using SMTP

146 Lesson Duration: 60 minutes

Lesson Overview

Connecting an SAP system to a mail server has become significantly easier due to the SMTP plug-in available as of SAP Web AS 6.10. This lesson provides you with an overview of configuring the SAP system for SMTP. We will discuss both sending and receiving e-mails.

Lesson Objectives

After completing this lesson, you will be able to:

- State the steps required to connect the SAP system to a mail server
- Configure the SAP system for connection to a mail server

As of SAP Web AS, connecting an SAP system to a mail server has become very easy (compared to the situation previously), due to the SMTP plug-in. You should therefore certainly demonstrate sending – and if you have time and want to, you can certainly also demonstrate receiving.

Without the use of the SMTP plug-in, external mail systems were connected by inserting a mail adapter. The connection between the SAP system and the mail adapter was created using SAPconnect, and for communication between the mail adapter and the external mail system, both the external mail system and the mail adapter had to be configured. For more information about connecting external mail systems using SAP Basis 4.x, see the online documentation for the area BC-SRV-COM, or search for “SXC” (SAP Exchange Connector).

Business Example

Your company uses SAP systems based on SAP Web AS. As a member of the system administration team, it is your task to implement the exchange of e-mails between an SAP system and an SMTP-compatible mail server without external components.

Introduction to Communication Using SMTP

As of SAP Web Application Server 6.10, SMTP (Simple Mail Transfer Protocol) is supported directly by the SAP kernel. This means that it is possible to exchange e-mails between the SAP system and any

© 2003 SAP AG. All rights reserved.

SMTP-compatible mail server without using additional external components (such as connectors or gateways). You can use any product that meets the SMTP standard as a mail server.

The following figure shows a simplified e-mail system landscape for a company. If an e-mail is sent to the address info@sap.com, it first reaches a Message Transfer Agent (MTA). This is a program (such as *sendmail*) that is responsible for forwarding and delivering e-mails. After receiving a mail from a Mail User Agent (the actual e-mail program) or another MTA, the MTA analyzes the mail and either delivers it to the local user (or his or her Mail User Agent) or forwards it -- if appropriate, taking alias rules into account -- to another MTA. In the example shown, the incoming e-mail is forwarded to the SAP CRM system with the local domain crm.sap.com.

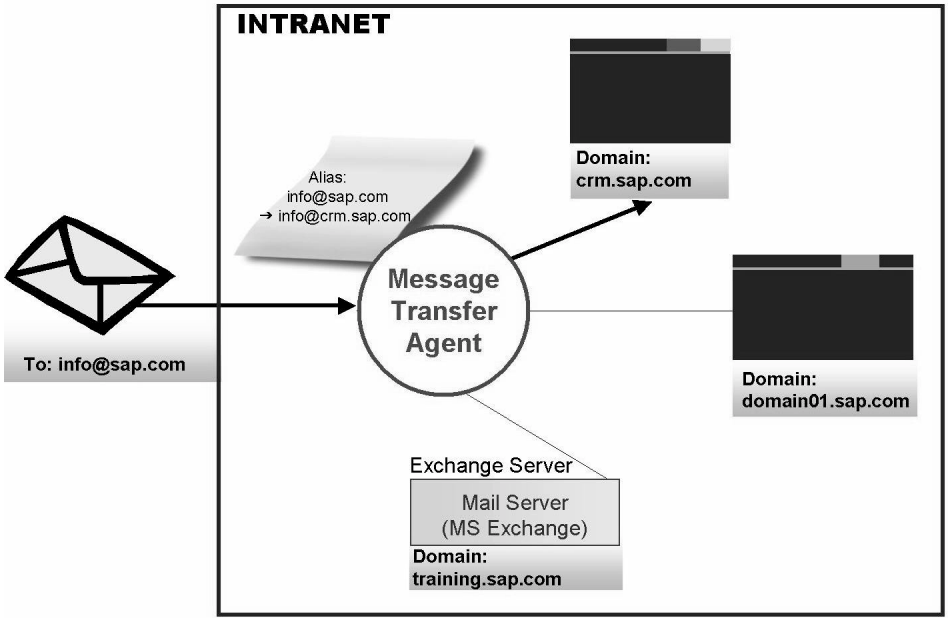


Figure 58: Task of a Message Transfer Agent (MTA)

As of SAP Web Application Server 6.20, it is also possible to exchange faxes and text messages (pager/SMS) using SMTP. Faxes and text messages are packed into e-mails for this purpose. For receipt, the fax/text message server or provider sends e-mails to addresses with the SAP system domain (such as to FAX=+1212541234@crm-prod.company.com).

Technical Details

In **the outbound direction**, the SAP system always transfers the mails to **one single** mail server. You can set its host address and port number. The technical background behind this is that an SMTP node can only have one

© 2003 SAP AG. All rights reserved.

target connection. The SAP Web Application Server, therefore, requires different Message Transfer Agents (MTA) to send e-mails to different domains, as shown in the following figure.

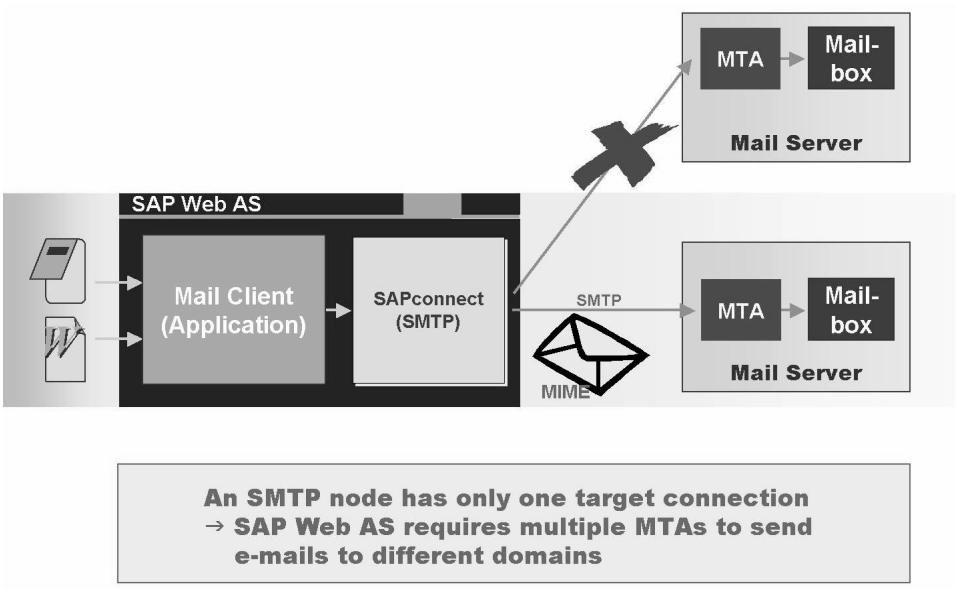


Figure 59: Restrictions for Outbound E-Mails

The SAP Web Application Server can receive e-mails through SMTP and redistribute these e-mails to Business Workplace (previously: SAP Office) users. In the **inbound direction**, the SAP system can receive mails from **any number** of mail servers. Each client can be reached using a separate virtual mail server (host name, port number) in transaction SICF . The best approach to addressing the SAP system and its clients is to use a separate subdomain for each client (such as training.sap.com). The assignment of these subdomains to host names and port numbers is defined using routing rules at the mail servers. There is a restriction: the SAP Web AS cannot be a Message Transfer Agent itself; that is, it cannot forward the mails to other mail servers. This is illustrated by the following figure.

© 2003 SAP AG. All rights reserved.

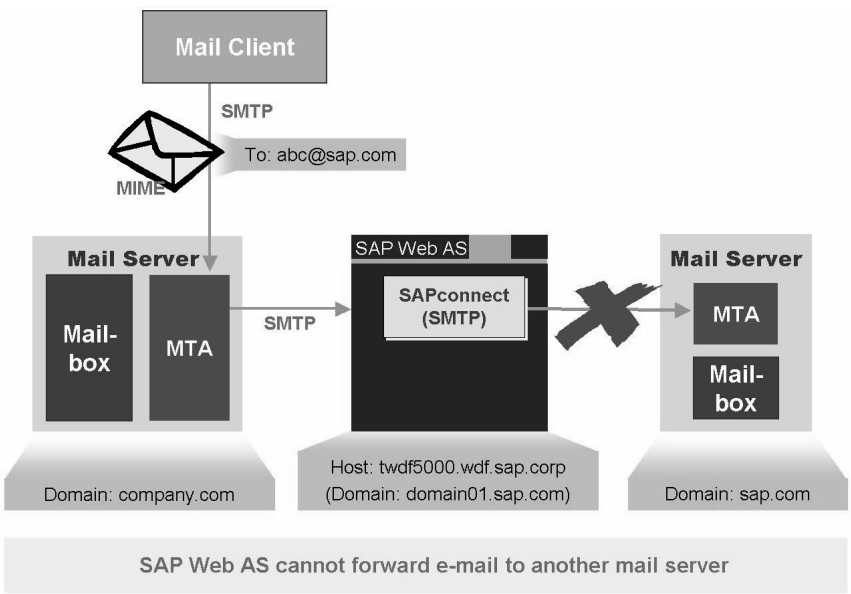


Figure 60: Restrictions for Inbound E-Mails

In this graphic, an e-mail arrives at the mail server for the domain company.com (such as the mail server of the sending company). This mail server forwards the mail (to the recipient abc@sap.com) to the SAP Web AS. The SAP Web AS cannot forward this mail to another mail server (with the matching domain).

Configuring the SAP System for SMTP

If you want to use the SMTP function, we recommend that you customize the following profile parameters of the SAP Web Application Server.

Note: The following restriction applies: The SAPconnect *send job* can only be scheduled for servers on which SMTP is activated. You should therefore activate SMTP on all application servers of the SAP system.

- *rdisp/start_icman = true*

  This parameter automatically starts the ICM process during the system startup.
- *exe/icman = <path for the executable of the ICM>*
- *icm/plugin_<*> = PROT=SMTP,PLG=<prog>*

© 2003 SAP AG. All rights reserved.

This parameter loads the SMTP plug-in. A dynamic library executes the SMTP protocol. <*prog*> describes the file name of the SMTP plug-in, which is in the instance directory of the SAP Web Application Server instance. It has the name **`smtpplugin`** and an appropriate file extension for your operating system (such as .so or .dll).

- *icm/server_port_<*> = PROT=SMTP,PORT=<port>*

  This parameter opens a TCP/IP port for receiving mails using the SMTP plug-in. <*port*> describes the port number. If no mails are to be received in this SAP system, set <*port*> to zero.

To avoid performance bottlenecks due to high numbers of SMTP requests (or, optionally, HTTP/S), you can use parameters to restrict the context usage in the backend SAP system for a protocol. You specify the percentage of all available contexts that can be used for the relevant protocol. In the SAP system, the maximum number of contexts in the system is limited by the profile parameter value of *rdisp/tm_max_no* (this parameter restrict the maximum number of users per instance). If the following quotas are exceeded, the requests are rejected in the ICM.

- *icm/HTTP(S)/context_quota*
- *icm/SMTP/context_quota*

For example, with the setting *icm/SMTP/context_quota = 20*, you specify that only 20 percent of the available contexts can be used for SMTP (mail functions). This means that you have more capacity for HTTP(S) requests if there is a high workload.

You can show the various parameters in the system and check the settings in the training system as a demonstration.

Sending E-Mails from an SAP System

Caution: In the ADM102 course, you (the trainer) install an external mail server for the entire course, to which you then connect “your” SAP system – and so do the participants with “their” SAP systems later, in the exercises.

We strongly recommend that you set up the mail server in the course as a demonstration for the participants. The reason for this is not to promote or to explain the product used in the course to the participants (the customers probably use mail servers from other vendors), but rather to demonstrate the significant points

© 2003 SAP AG. All rights reserved.

(such as protocol, port, and domain). These details are ultimately required in the SAP system and should therefore not "appear from nowhere".

Therefore do not become bogged down in the details of the mail server, but demonstrate the points mentioned.

Note: For this course, we use the freeware version of the **ArGoSoft mail server**. For more information about the product, other versions, and the download, see www.argosoft.com You can, of course, also give this address to the participants.

To set up the ArGoSoft mail server:

1. It is best if you are logged on at operating system level of "your" server with the Terminal Server Client (RDP Client).

2. Open Windows Explorer, navigate to **`G:\Additional_Files\ADM102\Mail Server`** and start the **`ArGoSoft setup.exe`** program.

3. The important clicks (see also the screenshots below) are *Setup*, leave everything set to Default, *Next*, and *Start Installation*.

   After you click *Finish*, the actual installation is complete – that was easy!

4. To configure the ArGoSoft mail server: Here, the participants should understand the meanings of the various entries.

   Start the mail server through the Microsoft Windows *Start* menu or the new shortcut on the desktop. You can ignore the message "Relay will be turned off".

   In the ArGoSoft Console (which you can also call by clicking with the right mouse button on the entry in the Microsoft Windows *System Tray* and choosing *Show*), choose *Options* from the *Tools* menu.

5. So that the mail server runs automatically after a restart of the entire server, under *Tools* → *Options* → *General* check the *Load Server at Windows Startup* field (leave the other fields unchanged).

6. Under *Tools* → *Options* → *Local Domains*, enter **`training.sap.com`** in the lower field (this is required later for the exercises). After choosing *Add*, this becomes **a domain to which all SAP systems can later send their e-mails**.

7. Show the participants that under *Tools* → *Options* → *Ports*, **port 25** is predefined for the SMTP protocol.

 © 2003 SAP AG. All rights reserved.

Leave all other settings set to their default values and chose *OK* to save the settings for the domain.

8. Each group receives its own e-mail recipient. We have created a Visual Basic script for this purpose. As preparation, call the path *Tools* → *ActiveX Library* → *Register* in the mail server.

9. Now start the script by double clicking the file **`G:\Additional_Files\ADM102\Mail Server\Create EMailUsers`**.

   This script creates the e-mail recipients **`ADM102-##`** (## from 00 to 30) with the password **`email##`**. The script **`Delete EMailUsers`** deletes all e-mail recipients...

10. After you have made all settings (just as after any other changes required), restart the mail server. You can do this either by choosing the *Start* pushbutton or by choosing *File* → *Start*.

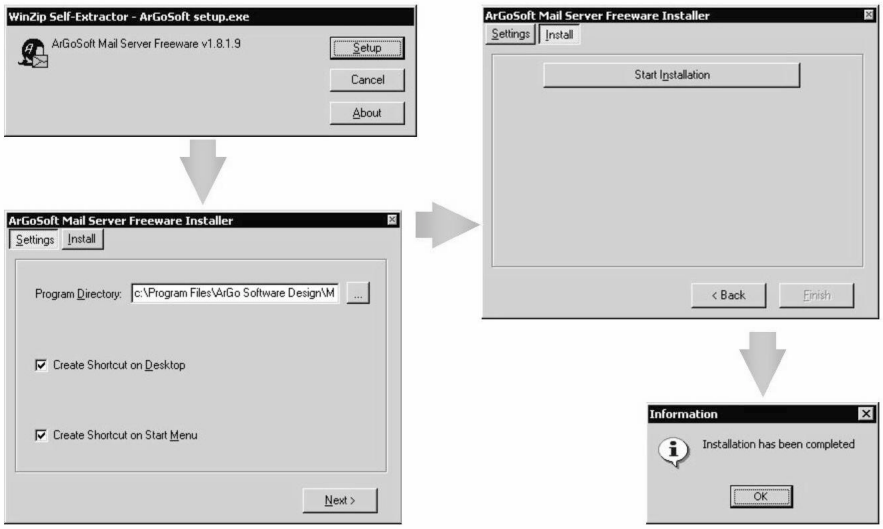


Figure 61: Installing and configuring the ArGoSoft mail server I

© 2003 SAP AG. All rights reserved.

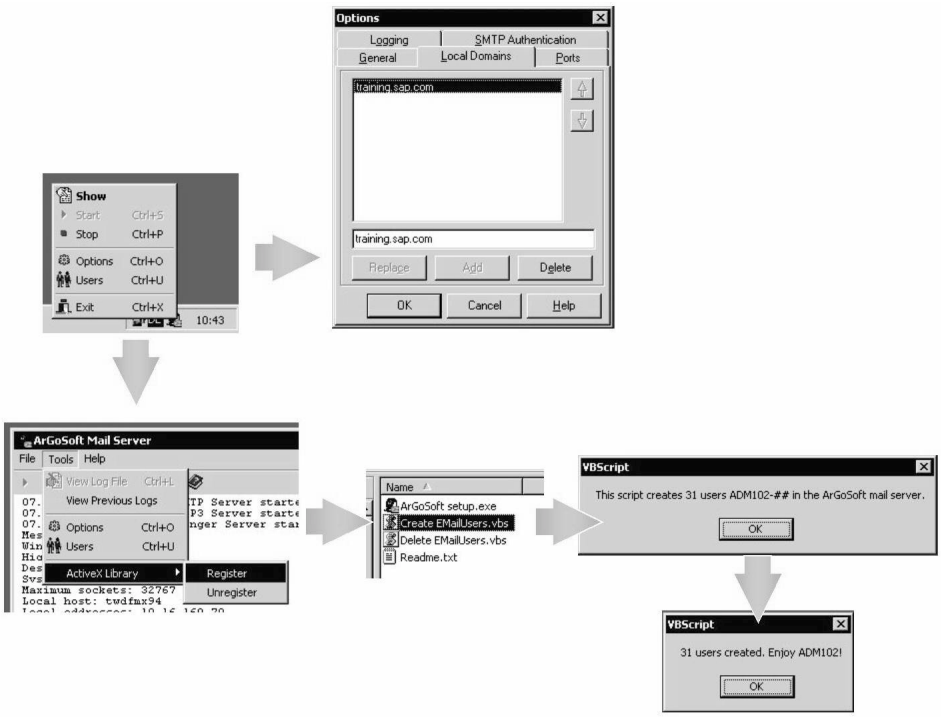


Figure 62: Installing and configuring the ArGoSoft mail server II

Note: The setup for the ArGoSoft mail server also exists on the participants' servers. If participants really want to set up the ArGoSoft mail server themselves, it is vital that they coordinate this with their neighboring group – otherwise, some services may not be available (due to double-assigned ports).

E-mails are created in the mail client of the SAP system (Business Workplace, transaction SBWP) and sent to a mail server using SMTP. These e-mails can now also be displayed with an appropriate mail client for this mail server, as shown in the following figure.

© 2003 SAP AG. All rights reserved.

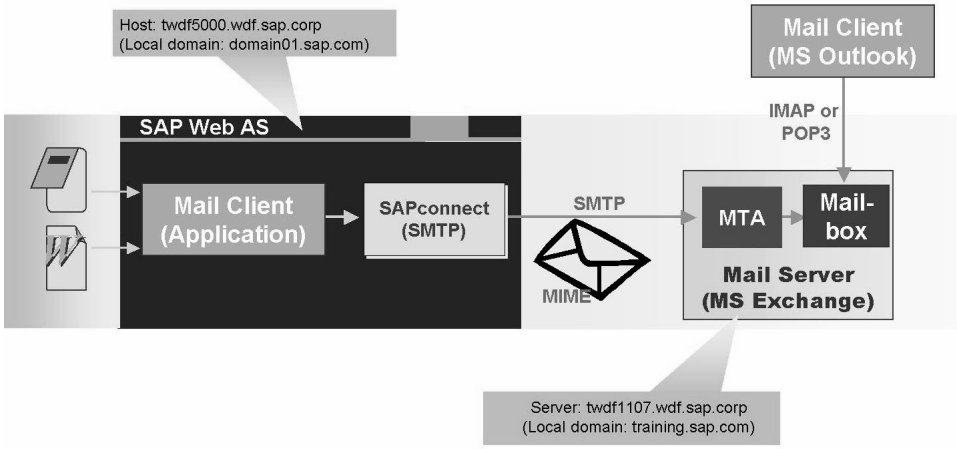


Figure 63: Outbound Mails: Process

This example in the graphic is based on the scenario that the SAP Web AS user abc (mail address abc@domain.sap.com) sends a mail to the Microsoft Exchange user xyz (mail address xyz@training.sap.com), who collects the message from his or her mailbox. In this graphic, the term "mail server" refers to a system that can communicate using SMTP, such as *Microsoft Exchange Server* or *BSD Sendmail*.

The names of the local domains are based on the names used in the exercises for this lesson.

Short explanations of the abbreviations used in the figure:

- **POP** is the abbreviation for Post Office Protocol. POP was defined in 1984 in connection with TCP/IP and allows the receipt of e-mails even on systems that cannot guarantee permanent connection to the mail server. POP3 is, therefore, together with SMTP (for sending messages), the standard protocol for most mail clients in the Internet. The POP3 protocol can delete messages on the server or leave them there, and can delete messages directly without first transferring them from the server. If more is required, such as hierarchical mailboxes, or filters, you should use the functions of the client; the protocol does not provide anything of this sort -- you may have to use IMAP.
- **IMAP** is the abbreviation for Internet Message Access Protocol. This is an e-mail protocol that allows the client to process mails on the server. You can also create folders on the mail server (remote mailboxes) to sort mails. IMAP was developed to transfer messages only when required: The user can choose (unlike with POP3) which data he or she actually wants to transfer to his or her own computer.

 © 2003 SAP AG. All rights reserved.

- **SMTP** is the abbreviation for Simple Mail Transfer Protocol. SMTP is the standard for exchanging e-mails between servers in the network. Mail clients use SMTP to send e-mails to a server, but not to receive e-mails.
- **MIME** is the abbreviation for Multipurpose Internet Mail Extensions. It consists of Internet extensions (coding procedures) for including binary data in Internet mails. In addition, MIME supports multipart mails to allow different data types in a mail or binary attachments, and mails in HTML format.

Follow the procedure below to configure a send process for SMTP:

- Maintain parameters for SMTP
- Maintain user addresses
- Activate and configure the SMTP node using transaction SCOT
- Schedule the send job

Profile parameters that are important for SMTP must be maintained.

Maintain the **user addresses**, as every user that wants to send e-mails requires an Internet e-mail address. This is used as the sender address. You can use transaction SU01 to enter the Internet address under *E-Mail* on the *Address* tab page. For security reasons, if a user is maintaining his or her own data (transaction SU3), he or she cannot maintain the entries for the communication types *E-mail* (INT or SMTP) and *Remote Mail* (RML).

In transaction SCOT, define the **domain** of the SAP system client under *Settings* → *Default Domain*. You need to do this for the following reasons:

- The SMTP plug-in logs on to the mail server with the domain as an ID.
- The message ID of the outbound e-mails is compiled using this domain.
- If an SAP user sends an e-mail without an Internet e-mail address, a sender address is generated that consists of the SAP user name and this domain.

Note: You must make SAPconnect settings using transaction SCOT for every client that is used for the send process.

Every client has an SMTP node. This node is created by the SAP system and cannot be deleted. The following steps are required for the **configuration of the SMTP node** to send an Internet mail:

- Set the node so that it is in use.
- In the *Mail Host* and *Mail Port* fields, specify the mail server to which the outbound mails are to be transferred.

© 2003 SAP AG. All rights reserved.

- When you set the node so that it can be used, enter the address area of recipient addresses that are to be reached using this node (such as *, if all e-mails are to be sent using SMTP).
- Enter the output format for SAP documents, such as PDF for SAPscript/Smart Forms, HTM for ABAP lists, and Business Object/links and TXT for RAW text.

You can also configure the node for sending faxes or text messages in the same way.

The figure below shows a configured SMTP node in transaction SCOT.

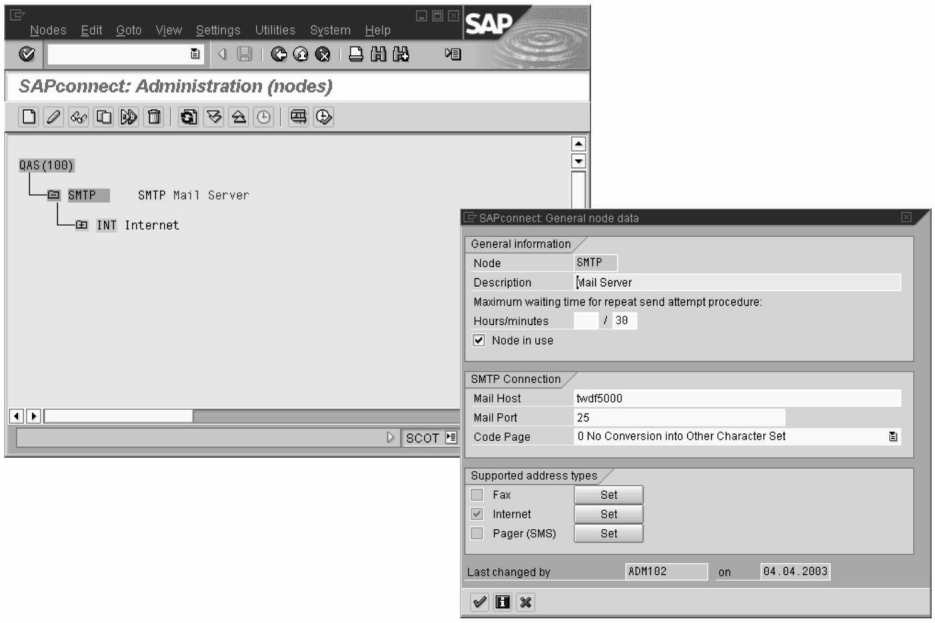


Figure 64: Configuring an SMTP Node

It is also useful to schedule a **send job**, as e-mails sent from an SAP application are initially placed in a wait queue. A background job that runs periodically, the SAPconnect **send job**, collects e-mails from the wait queue and sends them to the mail server. In the *Jobs* view in transaction SCOT, create a background job, for example with the variant *SAP&CONNECTALL*, and schedule it periodically (with a period value of, for example, 10 minutes).

Setting up transaction SCOT and the Outlook Express mail client is described in detail in the exercises. You can simply demonstrate it step by step in the system with the slides.

© 2003 SAP AG. All rights reserved.

Receiving E-Mails in an SAP System

When you send an e-mail from a mail client to a user in an SAP system, the e-mail is first forwarded to a mail server and then forwarded to the SAP system using the MTA. The e-mails are sent to the mail client in the SAP system (Business Workplace) on a client-dependent basis using SAPconnect. This is illustrated by the following figure.

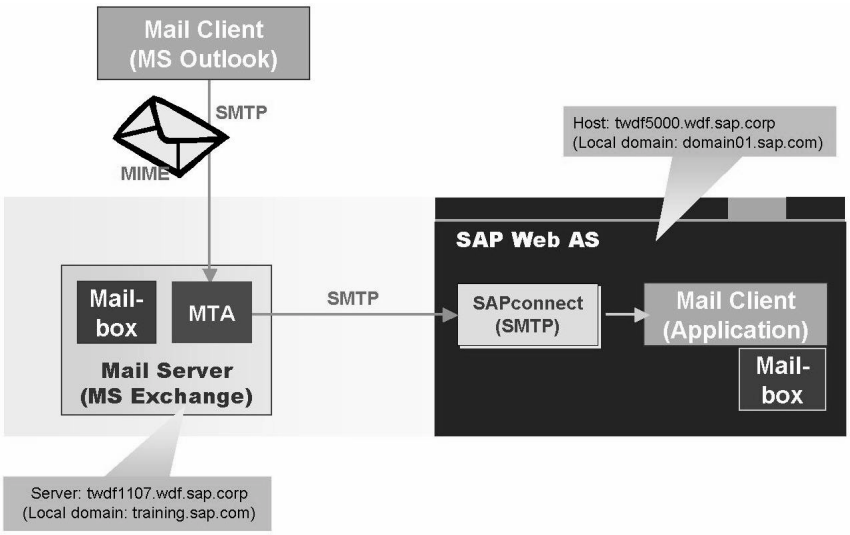


Figure 65: Inbound Mails: Process

This example in the graphic is based on the scenario that a Microsoft Outlook user sends a mail to the SAP Web AS user abc (mail address abc@domain01.sap.com). This mail is forwarded to the SAP Web AS by the mail server and arrives in the mailbox of the Business Workplace (transaction SBWP). In this graphic, the term "mail server" refers to a system that can communicate using SMTP, such as *Microsoft Exchange Server* or *BSD Sendmail*.

Follow the procedure below to configure a receive process for SMTP.

- Maintain parameters for SMTP
- Create the system user for processing mails
- Maintain user addresses
- Set the default domain in transaction SCOT
- Maintain and activate the SAPconnect service for each client using transaction SICF

© 2003 SAP AG. All rights reserved.

Setting the default domain in transaction SCOT is not necessary for the receive process itself, but is rather required, for example, for returning received receipts (that is, for a send process).

The profile parameters that are important for SMTP must be maintained.

A user is required to process inbound mail in each client. Use transaction SU01 to create a **system user** (or a **communication user**) with the profile S_A.SCON.

You must maintain an **Internet mail address** for every user that is to receive e-mails in an SAP system. This is also used as the sender address. Enter the Internet mail address using transaction SU01 on the *Address* tab page under *E-Mail*. For security reasons, if a user is maintaining his or her own data (transaction SU3), he or she cannot maintain the entries for the communication types *E-mail* (INT or SMTP) and *Remote Mail* (RML).

In transaction SCOT, define the **domain** of the SAP system client under *Settings* → *Default Domain*.

For every client of an SAP system in which inbound mails (or status notifications for sent mails) are to be received and possibly processed, you must **create an SMTP server** for which an assignment to a virtual mail host and logon data are stored. In transaction SICF, there is already an SMTP server for every SAP system, which is delivered by SAP. Use this for the first client that you want to make able to receive mails, and create a new SMTP server for each additional client. If you are working with multiple clients, you must create a virtual host for each client in which messages are received using the SMTP plug-in. Make the following settings in transaction SICF by double clicking the menu entry *SAPconnect*:

- On the *Host Data* tab page, maintain the parameter *is/SMTP/virt_host_<*>= <host>:<port>,<port>,...;*

  This parameter defines a virtual mail host for receiving mails. If all inbound mails (including status notifications) are to be received and processed by a single client of the SAP system, this parameter is not required. In this case, *is/SMTP/virt_host_0= <*>:<*>;* is used as the default. If multiple clients are intended as receivers, you must create a virtual host for each client. *<host>* and *<port>* describe the name of the host and the port to which inbound mails are addressed.

- On the *Service Data* tab page, enter the client to which the mails received by the virtual mail host are to be forwarded and maintain the logon data of the system user that was created for the inbound mails.

© 2003 SAP AG. All rights reserved.

- On the *Handler List* tab page, insert **`CL_SMTP_EXT_SAPCONNECT`** at position 1.

Note: As of SAP Web AS 6.20, every SMTP server must be activated (using transaction SICF under *Service/Virtual Host* → *Activate* or right-click and use the context menu), after it has been created or changed.

For more detailed information about the procedures for configuration an e-mail connection in the SAP system, see the following SAP Notes:

- SAP Note 455140: *Configuration E-mail, fax, paging/SMS via SMTP*
- SAP Note 455127: *E-Mail (SMTP) in different SAP releases*
- SAP Note 312690: *SAPconnect: Collective Note*

Setting up transaction SCOT and the Outlook Express mail client is described in detail in the exercises. You can simply demonstrate it step by step using the system and the graphics.

© 2003 SAP AG. All rights reserved.

157

Exercise 7: Sending and Receiving E-Mails with SAP Web AS

Exercise Duration: 30 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Configure the SMTP plug-in of the SAP Web AS
- Send e-mails from an SAP system using SMTP
- Receive e-mails in an SAP system using SMTP

Business Example

As a member of the system administration team, you want to implement the exchange of e-mails between an SAP system and an SMTP-compatible mail server without external components.

© 2003 SAP AG. All rights reserved.

Task 1 Setting Up the External Mail Client

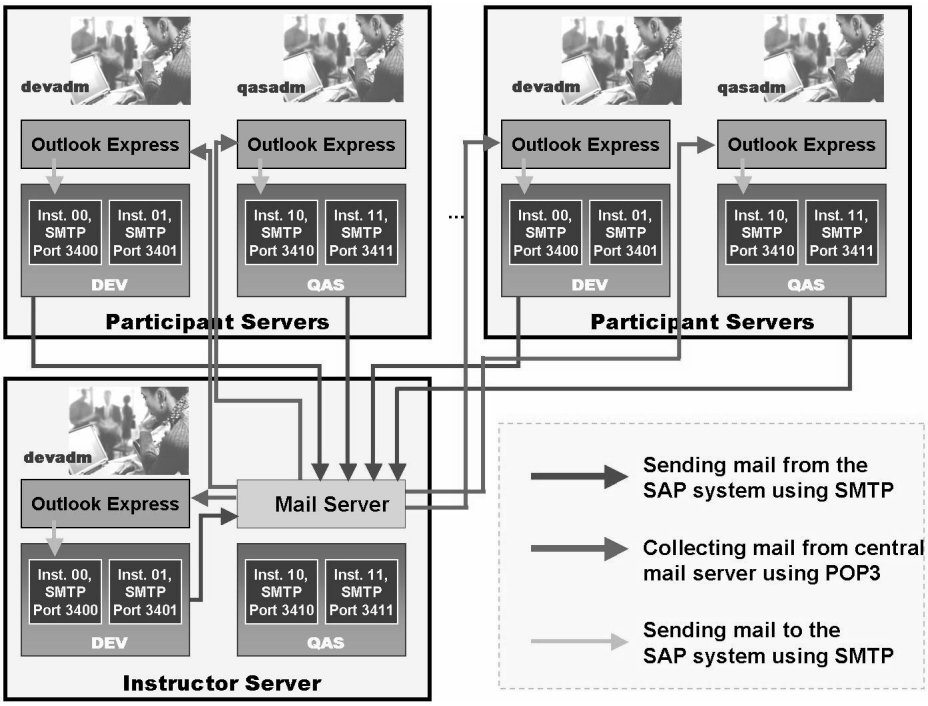


Figure 66: Complete Scenario of the Training Landscape

The graphic refers to the system setup for the ADM102 training course.

Set up Microsoft Outlook Express to receive e-mails with the e-mail recipient (prepared for each exercise group).

Note: Outlook Express is used here as an example. The user interfaces and functions of other mail clients are different; however, the basic concepts and required entries are identical.

1. Start Outlook Express at the operating system level of your server (not locally in the training room).
2. Set up a mail account with the following details:

Display name	any (such as your real name)
Existing *e-mail address*	`ADM102-##@training.sap.com`

Continued on next page

© 2003 SAP AG. All rights reserved.

Incoming mail (POP3) server	The **host name** on which your instructor set up the external mail server (such as **`twdf0042`**)
Outgoing mail (SMTP) server	The name of the host on which your SAP system is running (such as **`twdf0044`**)
Account name	**`ADM102-##`**
Password	**`email##`** (activate *Remember password*)

Note: The entries for *Display name* and *Outgoing mail server* are only of importance when sending e-mail from Outlook Express.

3. Test whether you can receive e-mails from the external mail server with your mail account.

Result

You can collect e-mails for your e-mail recipient, ADM102-##, from the shared mail server.

Task 2 Setting Up the Sending of E-Mails in the SAP System

Set domains and confirmation of receipt and set up the SMTP node.

1. In your SAP system, enter **`domain##.sap.com`** as the default domain, where **`##`** is your group number.
2. Configure SAPconnect so that no confirmation of receipt is expected for Internet mails.
3. Set up the SMTP node in your SAP system so that e-mails addressed to recipients in the domain **`training.sap.com`** can be sent to the external mail server (set up by your instructor).

Hint: If the external mail server also operates as a Message Transfer Agent, you could also use the setting **`*`** for the domain for SAPconnect. However, in the training landscape, the mail server does not forward any e-mails, but rather processes only the specified domain.

Continued on next page

© 2003 SAP AG. All rights reserved.

Result

Your SMTP node is now operational and responsible for all messages to Internet addresses. It is this easy to connect to an SMTP-compatible mail server.

Task 3 Sending an E-Mail from an SAP System

Create an e-mail and start the send process.

1. Enter any first name and last name for your **ADM102-##** user in the SAP system, and the e-mail address **ADM102-##@domain##.sap.com**, where ## is your group number.
2. Create an e-mail to the recipient **ADM102-##@training.sap.com** in the SAP system, for which no confirmation of receipt is to be requested.
3. Start the send process manually.
4. Display the e-mails sent to the user **ADM102-##@training.sap.com** in Outlook Express.
5. **Optional:** Schedule the send process for recipients of Internet e-mails periodically (every five minutes).

   Send a list output (such as report **SHOWCOLO**) to recipient **ADM102-##@training.sap.com** and observe the sending when the send process is next called.

Result

Congratulations on successfully connecting your SAP system to a mail server.

Task 4 Optional: Receiving E-Mails in the SAP System

Set up Outlook Express to send e-mails, and set up the SAP system to receive e-mails.

1. For each of your application servers, determine the port at which requests in the SMTP protocol are processed.
2. In your SAP system, create a user, **MAILDISP**, to distribute e-mails in the current client. This user should be of the type *Communication* and you should assign it the authorization profile **S_A.SCON**, delivered by SAP.
3. Enter the user that you have just created in the virtual host for *SAPconnect* (transaction SICF). Remember to activate the virtual host after changing it.

Continued on next page

 © 2003 SAP AG. All rights reserved.

4. Start Outlook Express on your server (as user `<sid>adm`). Check that the *Outgoing mail (SMTP) server* setting for your mail account points to the name of the host on which your SAP system is running (such as `twdf0044`).

   Important: Enter a valid port for your SMTP plug-in as the port for outbound e-mails (SMTP) (we suggest the port for the central instance). You determined the port information in Task 2.

5. Use Outlook Express to send an e-mail to **ADM102-##@domain##.sap.com** and check its arrival in the SAP system.

Result

You are now able to connect SMTP for an SAP system in both directions.

© 2003 SAP AG. All rights reserved.

Solution 7: Sending and Receiving E-Mails with SAP Web AS

Task 1 Setting Up the External Mail Client

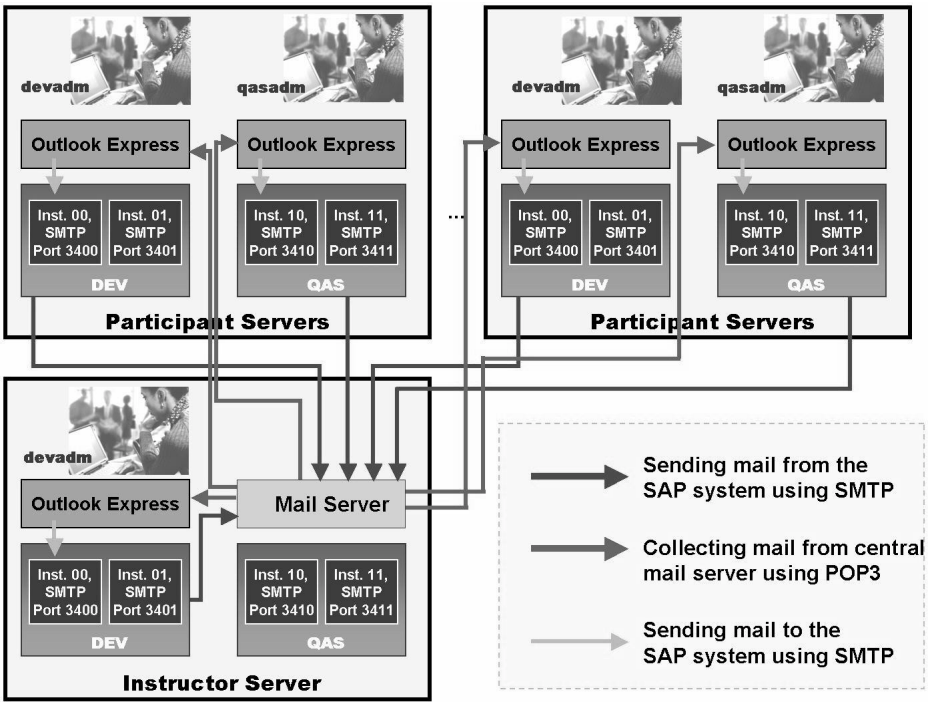


Figure 67: Complete Scenario of the Training Landscape

The graphic refers to the system setup for the ADM102 training course.

Set up Microsoft Outlook Express to receive e-mails with the e-mail recipient (prepared for each exercise group).

Note: Outlook Express is used here as an example. The user interfaces and functions of other mail clients are different; however, the basic concepts and required entries are identical.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. Start Outlook Express at the operating system level of your server (not locally in the training room).

   a) If you have not already done so, log on to the Terminal Server Client (also known as the RDP Client) at operating system level using the user **`<sid>adm`**.
   b) Choose *Start* → *Programs* → *Outlook Express*.

2. Set up a mail account with the following details:

Display name	any (such as your real name)
Existing *e-mail address*	**`ADM102-##@training.sap.com`**
Incoming mail (POP3) server	The **host name** on which your instructor set up the external mail server (such as **`twdf0042`**)
Outgoing mail (SMTP) server	The name of the host on which your SAP system is running (such as **`twdf0044`**)
Account name	**`ADM102-##`**
Password	**`email##`** (activate *Remember password*)

Note: The entries for *Display name* and *Outgoing mail server* are only of importance when sending e-mail from Outlook Express.

   a) When you first open it, Outlook Express immediately starts a Wizard to create a mail account. When opening it later, you can enter the required information by choosing *Tools* → *Accounts...* → *Add* → *Mail*.
   b) Fill out the fields as described above and leave all other fields unchanged. Choose *Next* to move to the next screen.

Continued on next page

© 2003 SAP AG. All rights reserved.

3. Test whether you can receive e-mails from the external mail server with your mail account.

   a) In Outlook Express, run the function *Send and Receive All* (by choosing the button or by choosing *Tools* → *Send and Receive* → *Send and Receive All*). Although there is not yet any e-mail for you on the external mail server, this should work without any error messages.

   Note: Outlook Express creates the welcome e-mail "Welcome to Outlook Express..." itself; it does not come from the mail server.

Result

You can collect e-mails for your e-mail recipient, ADM102-##, from the shared mail server.

Task 2 Setting Up the Sending of E-Mails in the SAP System

Set domains and confirmation of receipt and set up the SMTP node.

1. In your SAP system, enter **`domain##.sap.com`** as the default domain, where **`##`** is your group number.

   a) Start SAPconnect Administration (transaction SCOT) and under *Settings* → *Default Domain*, enter the value **`domain##.sap.com`**.

2. Configure SAPconnect so that no confirmation of receipt is expected for Internet mails.

   a) Choose *Settings* → *Confirmation of Receipt*, and define that SAPconnect does **not expect confirmation for Internet Mail**.

   Hint: This setting means that sent e-mails are immediately assigned the status "completed" (otherwise, they remain in the status "in transit" until a confirmation of receipt is received).

   In the SAP system, the sending application determines whether a confirmation of receipt is actually requested.

Continued on next page

 © 2003 SAP AG. All rights reserved.

3. Set up the SMTP node in your SAP system so that e-mails addressed to recipients in the domain **`training.sap.com`** can be sent to the external mail server (set up by your instructor).

Hint: If the external mail server also operates as a Message Transfer Agent, you could also use the setting **`*`** for the domain for SAPconnect. However, in the training landscape, the mail server does not forward any e-mails, but rather processes only the specified domain.

a) If you have not already done so, switch to *View* → *Node*.

Hint: You should see a "-" icon next to the SMTP node, as this node is not yet operational.

b) Double-click the SMTP node and enter the following information:

Description	any
Maximum waiting time...	**`30`** minutes
Mail Host	The **host name** on which your instructor set up the external mail server (such as **`twdf0042`**)
Mail Port	The port entered for the external mail server (typically **`25`**)
Code page	**`0`** (no conversion)

c) For *Supported Address Types*, choose *Internet* → *Set* → *Address Area*, enter **`*training.sap.com`**, and confirm.

d) Select *Node in use* and save your settings.

Result

Your SMTP node is now operational and responsible for all messages to Internet addresses. It is this easy to connect to an SMTP-compatible mail server.

Task 3 Sending an E-Mail from an SAP System

Create an e-mail and start the send process.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. Enter any first name and last name for your **ADM102-##** user in the SAP system, and the e-mail address **ADM102-##@domain##.sap.com**, where ## is your group number.

   a) Start User Maintenance (transaction SU01) and change your user, **`ADM102-##`**. On the *Address* tab page, enter any first name and last name and, under *E-Mail*, the address **`ADM102-##@domain##.sap.com`**.

   Hint: For security reasons, it is not possible to change the e-mail address by maintaining your own user data (by choosing *System* → *User Profile* → *Own Data*, transaction SU3).

   If the *E-Mail* field for a user is blank, the address is created using the structure **`<user ID>@<default domain>`**.

2. Create an e-mail to the recipient **ADM102-##@training.sap.com** in the SAP system, for which no confirmation of receipt is to be requested.

   a) Call the SAP Business Workplace (transaction SBWP) and choose *New Message*.

   Create a message (using any title and content). As soon as you have transferred the address **`ADM102-##@training.sap.com`** for the *Recipient* by choosing enter, the SAP system identifies from the structure of the address that it is an Internet address. Messages can be sent to different types of address (see the F4 help for the *Recip. Type* field).

   b) Switch to the *Trans. Options* tab page **after** you have specified the *Recipient*. Under *Confirm Send Status*, specify that this is *Never* to be done.

   The reason for this is that the "route back" to your SAP system has not yet been defined; it would therefore not be possible to deliver the confirmation e-mail.

   c) Finally, choose *Send*.

Continued on next page

 © 2003 SAP AG. All rights reserved.

3. Start the send process manually.
   a) View the information in SAPconnect Administration (transaction SCOT) under *View* → *System Status*. There should be a message with status "waiting."
   b) Now choose the *Start Send Process* function for the *Address Type* `INT` (for Internet mail).
   c) View the status once the send process is completed. If an error occurs, you can choose *Utilities* → *Overview of Send Orders* to analyze the cause of the error.
4. Display the e-mails sent to the user **ADM102-##@training.sap.com** in Outlook Express.
   a) After you have run the function *Send and Receive All* in Outlook Express (by choosing the button or by choosing *Tools* → *Send and Receive* → *Send and Receive All*), all e-mails that have been sent to the specified recipient are displayed.

      If you want, you can now send e-mails to neighboring groups (and use a distribution list to do this).
5. **Optional:** Schedule the send process for recipients of Internet e-mails periodically (every five minutes).

Continued on next page

 © 2003 SAP AG. All rights reserved.

Send a list output (such as report **SHOWCOLO**) to recipient **ADM102-##@training.sap.com** and observe the sending when the send process is next called.

a) In transaction SCOT, choose *View* → *Jobs*. The system displays how often the send process has already run in your client (you can change the evaluation period).

b) Choose the *Create* button and enter a job name (such as **`SEND INT EMAILS`**).

   SAP delivers a variant, *INT*, for this address type. Select the entry and choose *Schedule Job*.

   On the next screen, adjust the time if required (you can view the system time by choosing *System* → *Status*) and then choose *Schedule Periodically*. Choose the required period and confirm.

   The job that you have just created now appears in the Job Overview (transaction SM37).

c) Call any list display in the SAP system (suggestion: Use transaction SA38 to execute report **SHOWCOLO**).

   Choose the menu path *System* → *List* → *Send* and enter **`ADM102-##@training.sap.com`** as the recipient. The system proposes the report title as the default *Title* of the message. If desired, enter some text as the *Document contents*. Choose *Send* to transfer your message to SAPconnect.

   The message is processed by the send process during the following five minutes. When the e-mail appears in Outlook Express, the entry *Attach:* indicates that there is an attachment. Double-click the entry and display the attachment by choosing *Open*.

   Note: By default, ABAP lists are converted to HTML format. As you can see from the selection options for *Output Formats for SAP Documents* when defining a SAPconnect node, other formats are also supported.

Result

Congratulations on successfully connecting your SAP system to a mail server.

Task 4 Optional: Receiving E-Mails in the SAP System

Set up Outlook Express to send e-mails, and set up the SAP system to receive e-mails.

Continued on next page

 © 2003 SAP AG. All rights reserved.

1. For each of your application servers, determine the port at which requests in the SMTP protocol are processed.

   a) There are a number of ways to determine the port responsible for SMTP. The simplest method is to call transaction SMICM and choose the *Services* button.

      You can see the number under which the SMTP service is running in the system in the first column (it is represented here by *xx*). The associated profile parameter is called *icm/server_port_<xx>*. You can also find the value of this parameter, for example:

      - In transaction SMICM by choosing *Goto* → *Parameters* → *Display*
      - By calling report **RSPFPAR**
      - In Profile Maintenance (transaction RZ10)

   b) Note that the port determined in this way (such as 3411) applies only for the application server to which you are currently logged onto. Log on to the second instance (possibly using transaction SM51), to determine the port set there.

      Note: Verify that a different port is used for the SMTP protocol on each application server of your host by checking with your neighboring group.

      You can find out which plug-ins (and therefore which protocols) the ICM of your SAP system provides by choosing *Goto* → *Parameters* → *Display*. The information that you require appears at the top of the list (the *icm/plugin_<yy>*parameters).

2. In your SAP system, create a user, **MAILDISP**, to distribute e-mails in the current client. This user should be of the type *Communication* and you should assign it the authorization profile **S_A.SCON**, delivered by SAP.

   a) In User Maintenance (transaction SU01), create the user **`MAILDISP`** as described in the exercise text. Set the user type *Communication* on the *Logon Data* tab page.

      Hint: Users of the type *Communication* are intended for dialog-free communication between different systems and are excluded from the checks for expired passwords.

Continued on next page

© 2003 SAP AG. All rights reserved.

3. Enter the user that you have just created in the virtual host for *SAPconnect* (transaction SICF). Remember to activate the virtual host after changing it.

   a) In transaction SICF, double-click the *SAPconnect* entry. On the *Service Data* tab page, switch to change mode. Fill out the *Anonymous Logon Data* area with the data for **`MAILDISP`** (client, user, password, and language). Leave all other fields unchanged and save your changes. Create a local change request at this point, if appropriate.
   b) As of SAP Web AS 6.20, you must activate entries for SMTP servers after creating or changing them. To do this, select the *SAPconnect* entry in transaction SICF and choose *Service/Virtual Host* → *Activate*.

   Hint: An SMTP server is required for each client of an SAP system in which e-mails can be received. You have now created the connection to a virtual mail host. If e-mails were also to be delivered in other clients of your SAP system, you would have to create additional hosts for these clients in transaction SICF.

4. Start Outlook Express on your server (as user **`<sid>adm`**). Check that the *Outgoing mail (SMTP) server* setting for your mail account points to the name of the host on which your SAP system is running (such as **`twdf0044`**).

   Important: Enter a valid port for your SMTP plug-in as the port for outbound e-mails (SMTP) (we suggest the port for the central instance). You determined the port information in Task 2.

   a) As user **`<sid>adm`**, choose *Start* → *Programs* → *Outlook Express* at operating system level.
   b) Under *Tools* → *Accounts...* → *Mail*, select your mail account and choose *Properties*. Check on the *Servers* tab page that the host name of your SAP system is entered for *Outgoing mail (SMTP) server*.

      Switch to the *Advanced* tab page and enter the correct port for *Outgoing mail (SMTP)* (such as 3410). Save these values.

Continued on next page

 © 2003 SAP AG. All rights reserved.

5. Use Outlook Express to send an e-mail to **ADM102-##@domain##.sap.com** and check its arrival in the SAP system.

   a) In Outlook Express, choose *New Mail* and create an e-mail for the recipient **`ADM102-##@domain##.sap.com`**.

      Then select *Outbox* in Outlook Express (you will see the e-mail that you have just written there) and choose *Send and Receive All* (use the button or choose *Tools* → *Send and Receive* → *Send and Receive All*).

   b) In the SAP system, call the SAP Business Workplace (transaction SBWP) as user **`ADM102-##`**. You will find your e-mail in the *Inbox*.

Result

You are now able to connect SMTP for an SAP system in both directions.

Result

We wish you every success in implementing this in your company.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- State the steps required to connect the SAP system to a mail server
- Configure the SAP system for connection to a mail server

Related Information

- Online documentation for the SAP Web AS 6.20: *mySAP Technology Components → SAP Web Application Server → Basis Services / Communication Interfaces → Communication Interfaces (BC-SRV-COM) → SAPconnect (BC-SRV-COM)*

© 2003 SAP AG. All rights reserved.

Lesson: Telephone Integration with SAPphone

172 Lesson Duration: 30 minutes

Lesson Overview

SAPphone integrates telephone functions into SAP applications. This allows data to be exchanged between computer and telephone processes. The telephone functions can be used through the graphical user interface of the SAP applications and extended with data from the SAP system.

This lesson will provide you with an overview of the architecture of a telephone integration based on SAPphone.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the architecture of telephone integration in SAP systems
- Make settings on the SAP Web AS that are required to connect it with the telephone system

---

This lesson should have only an introductory character for the participants. For this reason, no exercises in the SAP system are provided for this lesson. You should explain central terms such as CTI server and SAPphone. Your demonstrations in the system should merely provide the participants with an initial impression of the settings that have to be made in the SAP system.

In the SAPphone environment, there are two different scenarios: using the SAPphone popup (that is, a popup appears if there is a phone call) and the Customer Interaction Center (CIC) in the CRM environment. Although the main application area of SAPphone is the CIC, the descriptions in this lesson relate mainly to the SAPphone popup (as this is available in every SAP Web AS). In CIC, the way that inbound calls are processed differs to some extent (keyword: “wait mode”; for more information, see the online documentation for the CIC) and additional settings are required for the agents in this case.

Business Example

As a system administrator, you want to learn about how you can integrate telephone functions into SAP applications to enable an exchange of data between telephone equipment and your SAP systems.

© 2003 SAP AG. All rights reserved.

Motivation

You should present the concept of a SAPphone work center to the participants. At this point, you should refer to the central problem to be solved: How can you connect the employee's telephone with the front end PC so that the PC can ultimately be used to control the telephone equipment (accept calls, start outbound calls, and so on)?

The aim of SAPphone is to remove the distinction between working with the telephone and working with the screen and to integrate the telephone more closely into the work process. The target group for SAPphone includes both users with daily office work and specialized telephone work centers in call centers.

The user can dial a person and create a telephone connection directly in an SAP application. In the case of an inbound call, the system displays data about the caller, or branches directly to a predefined SAP application. During a telephone conversation, the user can create notes or process tasks in an SAP application.

The primary advantages of telephone integration in the SAP system are related to connecting telephone activities with the business processes mapped in the SAP system. The information from telephone calls is directly available in the SAP system and can be additionally processed.

SAPphone functions are part of the generic object services, meaning that they can be called in all applications that support these services.

Computer Telephony Integration (CTI) between telephone equipment and the SAP system allows employees to control their telephones directly from the SAP application. This provides the following advantages:

- Simplification of operation: The employee no longer needs to switch between operating the telephone and SAP GUI, but can perform all processing from the SAP application.
- Automatic identification of the caller: The employee can quickly obtain an overview of the caller's customer data with automatic identification of a caller using his or her telephone number. This improves service quality and customer satisfaction.
- Outbound calls: Telephone numbers no longer have to be manually dialed if the business partner is called directly from the SAP application.

© 2003 SAP AG. All rights reserved.

A SAPphone work center represents a connection between an employee's extension and the computer (SAP GUI) at which he or she is working. This assignment is required so that an inbound call can be displayed on the correct PC (client) and outbound calls can be started from the correct telephone.

To perform telephony functions, it is necessary that the system knows the connection between the computer and the associated telephone; that is, it must be possible to identify the work center.

The type of work center identification used depends on the GUI in use and can be set in Customizing. Ideally, *Identification Using Registry Entries* or *Identification Using Host Names* is used. When creating a work center, you then either store a SAPphone workstation ID in the registry of the associated PC or the unique host name of the computer in the SAP system. However, neither of these types of identification can be used with all GUIs. The *Identification Using the Users* method should only be used if employees work at fixed work centers, due to the fixed employee-work center relationship.

The following discusses which components are required and which settings are necessary in the SAP system to be able to create a connection between a particular computer (at which the call center employee sits) and a specific telephone number of the telephone equipment.

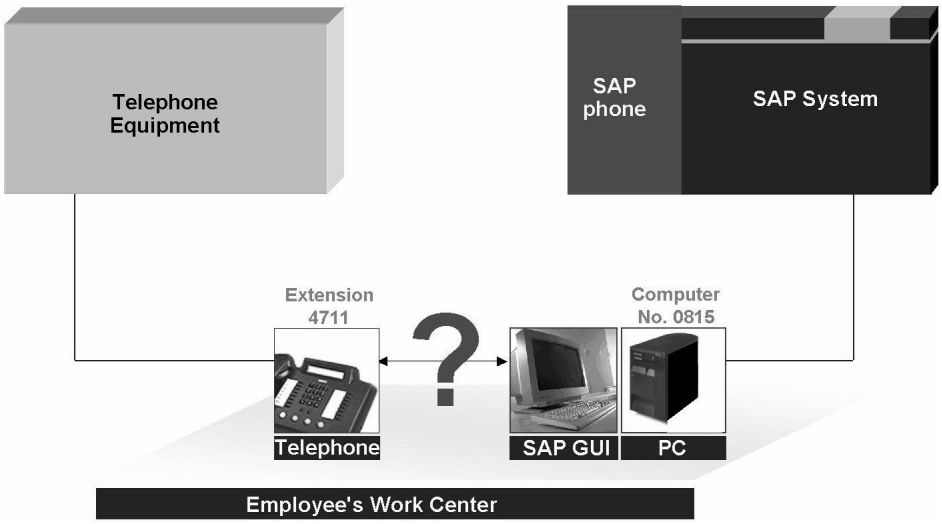


Figure 68: How Should you Connect the Telephone and User PC?

© 2003 SAP AG. All rights reserved.

Central Terms and Architecture of Telephone Integration

The following components are required for a scenario with Computer Telephony Integration (CTI) using SAPphone:

- Telephone equipment (Private Branch Exchange [PBX])
- CTI server
- SAP system (in the case of a Customer Interaction Center [CIC], a CRM system) with a configured SAPphone interface
- Employee work center, consisting of a telephone and a computer with SAP GUI.

The SAP applications transfer data to SAPphone, including, for example, the telephone number and country code. In return, SAP applications transfer data from SAPphone about an inbound call for processing. SAPphone provides a user interface for performing telephony functions and transactions with which you can make the required settings and perform Customizing for telephone integration in the SAP system.

Outside the SAP system, telephony hardware and software is required to control the telephones, as is the connection software between the SAP system and the telephone system (CTI server).

The connection between the telephone equipment and the SAP system is illustrated by the following figure.

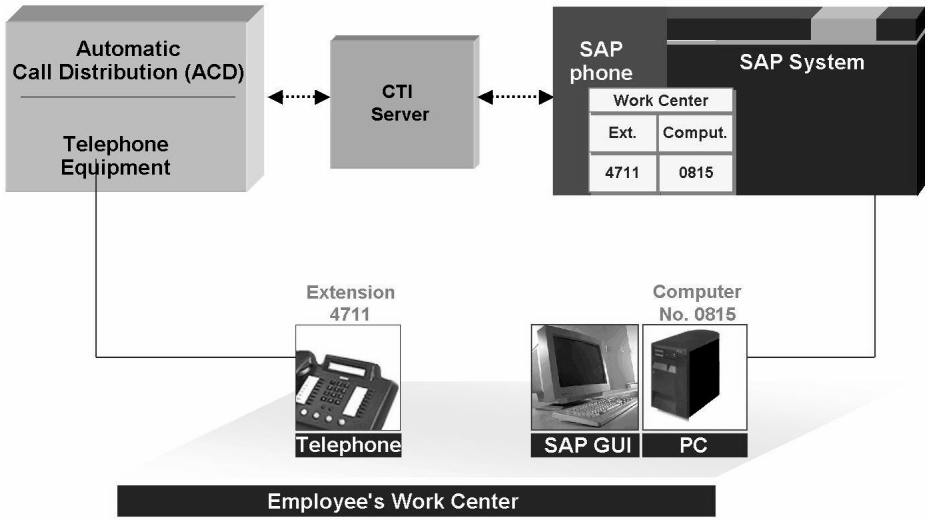


Figure 69: Telephone Integration Architecture: Overview and Components

There are essentially two methods of telephony integration with SAPphone:

© 2003 SAP AG. All rights reserved.

- Use a CTI solution that creates a gateway between the RFC interface of SAPphone and the interface of the telephone equipment.
- Use a TAPI-based solution. TAPI stands for Telephony Application Programming Interface and is an interface for telephony applications defined by Microsoft. If the vendor of the telephone equipment offers a TAPI Service Provider, you can consider this solution. The **SAPphone server** is a software component delivered by SAP free of charge that creates a connection between SAPphone and TAPI.

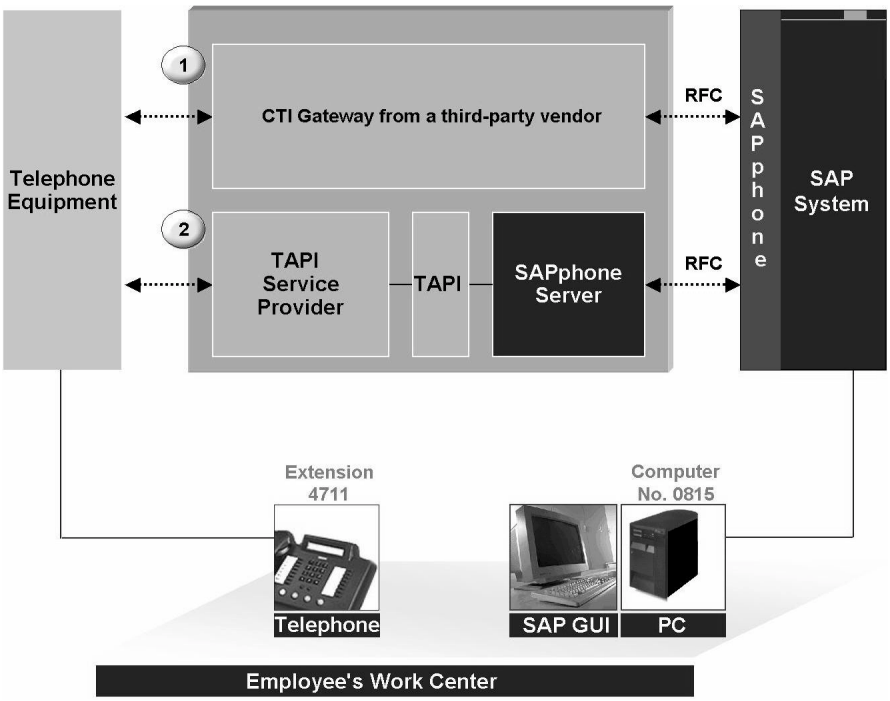


Figure 70: Options for Implementing a CTI Server

Third party vendors for a CTI gateway include Genesys and Telesnap. For a current list, see the *Software Partner Directory* under the "Computer Telephony" software category.

The process flow of an inbound call in a call center is explained below as an example. Customers usually call one or more central numbers to contact a call center. When a call arrives in a call center, a free employee that can accept the call must first be identified. This distribution of inbound calls to the employees is performed by the Automatic Call Distribution (ACD) module of the telephone equipment.

1. An external call is received by the telephone equipment. The caller dialed the number of the call center, 1-800-123-4567. The caller's telephone number is also transferred.

© 2003 SAP AG. All rights reserved.

2. The ACD module of the telephone equipment distributes the call to a free telephone extension. For example, the employee with the extension number 4711 is determined to be free.

3. The inbound call is directed by the telephone equipment to the employee's extension. The telephone of the employee with extension 4711 rings. At the same time, the call is reported to SAPphone through the CTI server. Some data is usually transferred at this point, such as the caller's number (if known) or the number called.

4. The computer (client) at which the employee with extension 4711 is working is determined. Technically, the settings of the SAPphone work center (consisting of the employee's extension and the computer at which he or she is working) are evaluated.

To be precise, this is only the case with the SAPphone popup. In CIC, the process is somewhat different: in this case, there is a "wait mode" or - in the case of connection using TAPI - the SAPphone Control to automatically activate the GUI when a call is received).

---

5. The system searches for a business partner in the SAP system using the caller's telephone number (automatic number identification). The business partners found are displayed directly in the application.

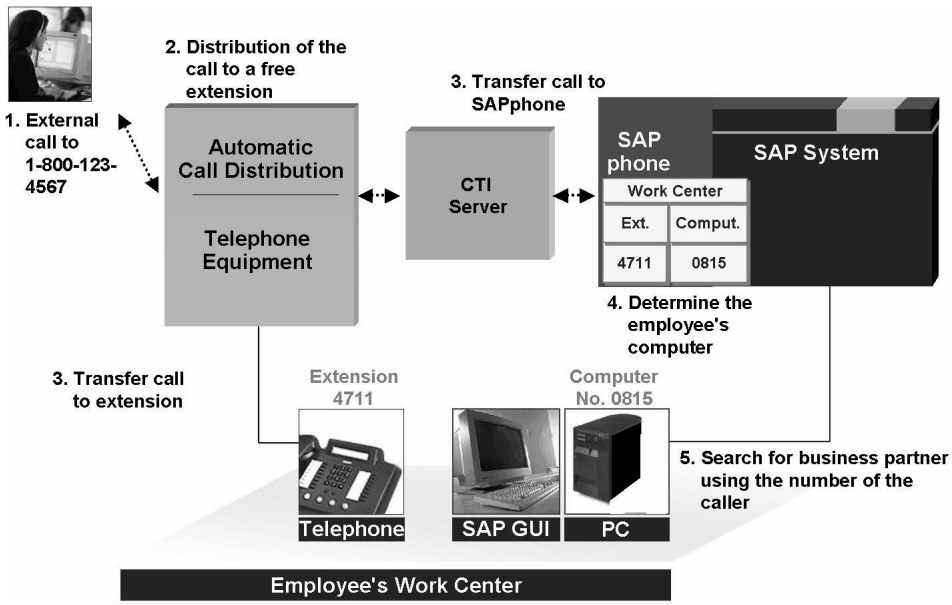


Figure 71: Process Flow of an Inbound Call

© 2003 SAP AG. All rights reserved.

Different architectures are used, depending on where the three layers of the telephony solution (SAP application, SAPphone, and telephone system) are installed:

- *Central architecture*: no telephone software or hardware at the work center and direct communication between the telephony server and the SAP system. This corresponds to the graphic "Process flow of an inbound call" in this lesson. The central architecture is the recommended configuration due to the small amount of configuration required.
- *Client-server architecture*: Connection between the private branch exchange (PBX) equipment and the work center using a telephony server and extension of the work center with parts of the telephony software (such as SAPphone Server), no hardware extension of the work center.
- *Local architecture*: Direct connection between the work center and the main connection or PBX (for example, using the telephone, or by modem, or telephone card) and extension of every work center with the software and hardware of the telephone system.

In each case, the external telephony software can be connected to the SAP system using the SAP RFC interface or using TAPI and SAPphone Server (see the above graphic about the alternatives for implementing the CTI server).

Configuration in SAP Web AS

In this section, the configuration steps required in the SAP system are only discussed. You should not discuss a CTI server and its configuration in this context.

To enable communication between the SAP system and the telephone system, you must make various settings in SAPphone.

On one hand, the telephony hardware and software must be available and installed, and (in the case of connection using TAPI) SAPphone must also be installed. On the other hand, the RFC connection between the SAP system and the CTI server (in the case of connection using TAPI, the SAPphone Server) must be configured. To do this, you must make the following settings:

- RFC destination for the connection from the SAP system to the CTI server (transaction SM59)

© 2003 SAP AG. All rights reserved.

- Connection data for the connection from the CTI server to the SAP system (in *Saprfc.ini* or the gateway software)

The data for the connection from the CTI server to the SAP system is required to display calls in the popup and for notification of missed calls. This data may not be required in Customer Interaction Center operation, depending on the CTI server.

The creation of SAPphone settings is controlled using authorizations. The authorizations are assigned to users through the assignment of roles. SAP delivers two roles that provide authorizations for SAPphone activities: *SAP_BC_SRV_GBT_USER* (this role contains the authorizations for the user of the telephony functions including maintaining the user's own settings) and *SAP_BC_SRV_GBT_ADMIN* or *SAP_BC_SRV_COM_ADMIN* (roles that contain authorizations for the administration of the telephony landscape). Application-specific authorizations that are not contained in the SAP Basis/SAP Web AS roles are required to perform the application tasks.

The structure of the telephone integration runs in various steps, the most important of which are presented here.

1. **Perform Customizing**. Customizing for SAPphone is performed in the SAP implementation guide (in transaction SPRO) under *SAP Web Application Server* → *Basis Services* → *Communication Interfaces* → *SAPphone: Cross-Client Settings* → *Activate Telephony/Choose Work Center Identification* (you can change the settings in the SAPphone system administration [transaction SPHB]). You activate the telephone integration for the current client here. Only if telephone integration is active are the dial functions shown in the applications. You also decide how the telephone work centers in your system are to be identified.

In the SAP Reference IMG (in transaction SPRO), open the specified path (this opens transaction SPH4) and activate telephone support. For background information about the types of identification, see the instructor note in the *Motivation* section.

2. Create a **Site** (transaction *SAPphone: System Administration*; *Tools* → *Business Communication* → *Communication* → *SAPphone*, transaction code SPHB). You must create a site for each company location at which telephone integration is to be used. Only then is it possible to create telephony servers. Sites create directories in the navigation tree of the SAPphone system administration, to which the telephony servers are assigned.

© 2003 SAP AG. All rights reserved.

Call SAPphone system administration (transaction SPHB). In the navigation structure, choose *Sites* by double clicking it. In the detail view, enter the name and generic telephone number of the site (such as **`Walldorf`** with the number prefix (that is, the area code and the common element of the numbers, but without an extension number)**`062277`**, country **`DE`**, length of extension **`5`** and time zone **`CET`**). Choose *Save*.

3. Create a **telephony server** (transaction SPHB). To allow communication between the SAP system and the telephone system, a telephony server must have been created and be operational. You must assign information to every server to allow it to access the external telephony software (RFC destination). You must also specify certain attributes for the server.

Double click the *Server Names* folder in the tree structure that has been created in transaction SPHB. A wizard leads you through the creation process. Use the following details, for example, when doing this (as there is no external telephony software connected to the training system, you can use the internal RFC destination **`NONE`** as a server simulation):

Server	**`SERV1`**
Name	**`Central server`**
RFC Destination	**`NONE`**
Server attributes	Do not check any
Functions	Check *Telephony*
Number conversion for outbound calls	Do not enter anything
Server is ready for use	**`X`**
Server is to be monitored by alert monitor	**`X`**

As a result, the system displays a corresponding entry in the tree structure of transaction SPHB.

4. **Create Work Centers** (transaction *SAPphone: Administration*; *Office* → *Telephone Integration* → *Extended Settings*; transaction code SPHA). Use this transaction to create a new SAPphone work center or to change

© 2003 SAP AG. All rights reserved.

your work center settings. The work center settings can only be created locally at the relevant work center. Telephony functions can only be executed once the work center has been created.

Choose *Work Center* on the initial screen of transaction SPHA. Enter or change: the number of the associated telephone (such as `4711`), the name of the telephony server to which the work center is assigned (use the server that you created previously), and the user that is responsible for these settings (such as your own user). Choose *Save* to complete the settings.

Note: If the work center has already been created, you can also call the settings using the SAPphone user interface. To do this, on the initial screen of transaction SPHA, choose *SAPphone* and then, on the *Services* tab page, in the *Work Center* group box, choose *Execute*.

You can optionally also choose *User Settings* on the initial screen of transaction SPHA. You can define here how the SAP system should behave with regard to inbound calls (for example, whether an inbound call should be displayed with a popup, or whether a notification should be made for calls received when the user is absent).

Transaction SPHD (menu path *Office* → *Telephone Integration* → *Own Telephone Number*) is also available for simplified maintenance of the work center.

---

5. **Test settings**. You have a various test options after the first configuration, after changes to the settings, or if errors occur. You can use transaction SPHA to simulate outbound and inbound calls.

You can optionally now perform a consistency check using transaction SPHB (by choosing *Utilities* → *Consistency Check*) and then simulate an outbound call using transaction SPHA. To do this, choose the *Initiate outbound call* pushbutton on the initial screen of transaction SPHA, enter a telephone number on the next screen, and choose *Initiate*. On the next screen, choose *Continue*. You can now, for example, accept, hold, retrieve, and hang up the call in turn using the corresponding pushbuttons.

The following figure clarifies the assignment of work centers to telephony servers and of telephony servers to sites (steps 2, 3, and 4 of the above list).

© 2003 SAP AG. All rights reserved.

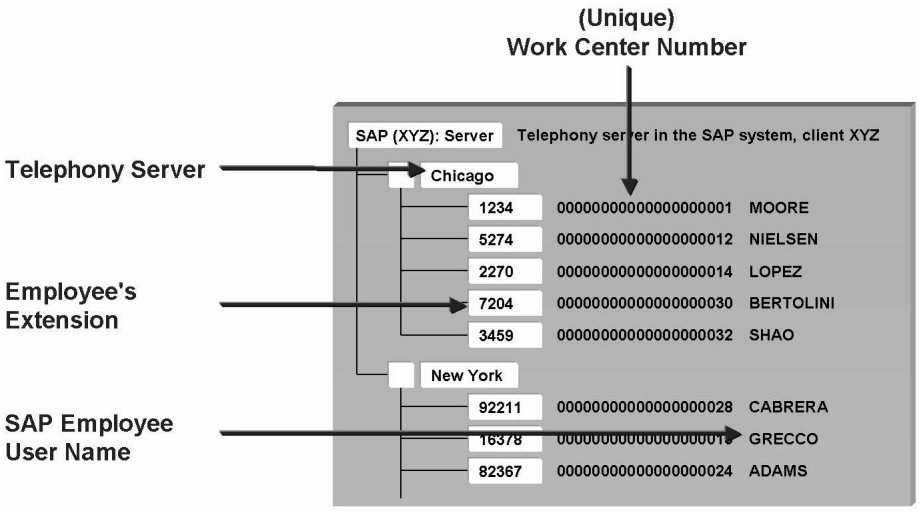


Figure 72: Assignment of Work Centers to Telephony Servers

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

You should clarify the difference between the terms **site**, **telephony server**, and **work center** for the participants.

Discussion Questions

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

What is the relationship between the terms site, telephony server, and work center?

Sites represent a physical company location in SAPphone, that is, an area with shared area code and main number. Each instance of the external telephony software (that is, the connection software between the SAP system and the telephone system) must be represented by a **telephony server** in the SAP system. You normally require at least one telephony server (an instance of the telephony software) for each site, SAP system, and client in which telephone integration is to be used. Each employee's **work center** is, in turn, assigned to exactly one telephony server.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the architecture of telephone integration in SAP systems
- Make settings on the SAP Web AS that are required to connect it with the telephone system

Related Information

For more detailed information about SAPphone, see the *SAPphone* area (BC-SRV-COM) of the SAP Library.

Loading and importing the newest version of the SAPphone server is described in SAP Note 119225.

© 2003 SAP AG. All rights reserved.

Unit Summary

You should now be able to:

- Outline the message flow in SAP systems
- Describe the function of SAPconnect
- State the steps required to connect the SAP system to a mail server
- Configure the SAP system for connection to a mail server
- Outline the architecture of telephone integration in SAP systems
- Make settings on the SAP Web AS that are required to connect it with the telephone system

© 2003 SAP AG. All rights reserved.

181

Test Your Knowledge

1. Which of the following statements regarding SAPconnect are correct?

   Choose the correct answer(s).

   - ☐ A As of SAP Web AS 6.10, a Microsoft Exchange Infrastructure must be set up outside the SAP system to be able to send mail from SAP systems.
   - ☐ B It is not possible to connect the UNIX-based *sendmail* to SAP systems.
   - ☐ C SAPconnect allows you to send mail both by RFC and the SMTP plug-in of the SAP Web AS.
   - ☐ D SAPconnect is configured cross-client.
   - ☐ E A send process automatically runs periodically in every SAP system after the system installation.

2. The connection of a mail server to an SAP system to send e-mails using SMTP from the SAP system runs through an SAPconnect node.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

3. How can the SAP system be connected with telephone equipment in the context of SAPphone?

   Choose the correct answer(s).

   - ☐ A Using a Computer Telephony Integration (CTI) gateway from a third-party vendor
   - ☐ B The connection is made automatically using Customizing settings in the SAP system
   - ☐ C Using the Telephony API (TAPI) interface with a TAPI Service Provider and a SAPphone Server
   - ☐ D Using none of the above

© 2003 SAP AG. All rights reserved.

182

Answers

1. Which of the following statements regarding SAPconnect are correct?

   Answer: C

   There are two ways to send mail in SAPconnect: by RFC connection or using the SMTP plug-in of the SAP Web AS. The other statements are incorrect.

2. The connection of a mail server to an SAP system to send e-mails using SMTP from the SAP system runs through an SAPconnect node.

   Answer: True

   To configure a send process for SMTP in the SAP system, you must, among other things, also have configured and activated an SMTP node using transaction SCOT (SAPconnect).

3. How can the SAP system be connected with telephone equipment in the context of SAPphone?

   Answer: A, C

   The connection between the SAP system and the telephone equipment is made using a CTI server. This can either be implemented using a CTI gateway from a third-party vendor or the TAPI interface in connection with a SAPphone server.

© 2003 SAP AG. All rights reserved.

Unit 4

183

extended Computer Aided Test Tool (eCATT)

In this unit, participants receive a first insight into the eCATT newly developed for SAP Web Application Server 6.20. This tool significantly extends the options of CATT and allows the testing of all functions that the SAP GUI for Windows provides for end users. This no longer existed with the CATT tool since the introduction of the control technologies as of SAP Basis 4.6, as CATT cannot correctly record transactions if these use the Control Framework.

If there are participants in your course that have experience of working with CATT, you should, in particular, explain to them the new possibilities that eCATT offers in comparison to CATT. You should also make the participants aware of the security issues in the eCATT environment, as discussed in the lesson "Configuring eCATT ". These participants may also be used to exchanging test data with CATT between the SAP system and a local file on the front end. This is not yet possible with the existing version of eCATT in the context of SAP Web AS 6.20. However, you can inform the participants at this point about the development for SAP Web AS 6.40, in which a number of extensions (such as the inclusion of local files of this type) are to be included.

Unit Overview

In this unit, you learn about the newly developed Extended Computer Aided Test Tool (eCATT), which has significantly extended functions in comparison with the CATT tool that has existed for some time. You will learn the uses of eCATT, obtain an overview of important system and client settings, and create a test script and monitor its execution.

Unit Objectives

After completing this unit, you will be able to:

© 2003 SAP AG. All rights reserved.

- Outline the uses of eCATT
- List the advantages of eCATT compared to CATT
- List security issues connected to using eCATT
- Make system and client settings related to eCATT
- Create a system data container
- Create a simple test script using eCATT and monitor its execution
- Define the implementation areas of various drivers for eCATT

Unit Contents

Lesson: Fundamentals of eCATT 223
Lesson: Configuring eCATT 233
Exercise 8: Configuring the System Landscape for Using eCATT ... 245
Lesson: Executing eCATT Test Scripts 251
Exercise 9: Executing an eCATT Test Script 259

 © 2003 SAP AG. All rights reserved.

Lesson: Fundamentals of eCATT

184 Lesson Duration: 20 minutes

Lesson Overview

This lesson introduces the test tool newly developed for SAP Web Application Server 6.20, the Extended Computer Aided Test Tool (eCATT). This tool provides a significantly extended range of functions in comparison to the CATT tool, which has been available for some time. In this lesson, we will discuss the uses of this tool, compare the functions of CATT and eCATT, and define the central terms in the eCATT environment.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the uses of eCATT
- List the advantages of eCATT compared to CATT

In this lesson, the participants should learn what automated tests can usefully be used for. You should also make clear the advantages that eCATT has in comparison to CATT (such as testing function modules, testing across system boundaries, the fact that external tools can be included for BSP tests and so on, and that transactions that use controls can also be tested, using the `SAPGUI`command). It is also important that central terms such as “test data container”, “test script”, “system data container”, and “test configuration” are well defined. You should make emphasize, particularly for customers that already use CATT, that with eCATT, there is no longer a fixed connection between the test data and the script, but that the test data is now defined in the test data container, meaning that test data is reusable.

Business Example

As a member of the system administration team, you want to obtain an overview of the possibilities that the new test tool eCATT offers for creating and performing function tests.

Introduction

In many cases, it is necessary to test individual processes within an SAP system landscape for operativeness. For example, it is necessary when integrating a new component into an existing system landscape to test

© 2003 SAP AG. All rights reserved.

both the new component for operativeness and its interaction with the other components in the landscape. Some scenarios that require tests are listed below:

- New SAP installations and upgrades
- Addition and integration of new components in an existing landscape
- Customer developments
- Core functions after importing Support Packages
- Function test in each case; possibly also acceptance tests

You can perform these tests manually using a written test plan. Examples for this are acceptance tests, that is, subjective tests of the usability of business processes. To do this, test descriptions are created, which must be performed by the tester during the test. However, manual tests usually require significant resources.

An alternative to the manual test scenario is to perform automated tests. In this case, the application processes (such as transactions) to be tested can be recorded and then automatically run. During the run, variables defined in the test script can be replaced with actual (application) data. Examples for this type of test are functional tests that check whether individual transactions are error-free. Automated tests have an advantage over manual tests in that it is easy to repeat tests (for example, if errors occur), and they produce more reliable data, as input fields can be excluded (if these are not actually to be tested). Also, a possible loss of motivation on the part of the testers due to the repetitive nature of the actions does not occur. Other advantages of automated tests are:

- Fewer test resources are required
- The test process is accelerated
- The test results are easier to reproduce, as the test can easily be repeated a number of times
- A log of the results of the tests is automatically generated

There is, however, a relatively high initial cost for performing automated tests. Very careful planning is required to write high-value, reusable test cases. The tests themselves can be performed relatively quickly.

If appropriate, you can briefly discuss the possibility of using eCATT for training users at this point. In this way, for example, you could define that transactions recorded with the `SAPGUI` driver are played back and that the playback is paused after each processed screen so that the users can understand which data was entered on which screen. However, you must

© 2003 SAP AG. All rights reserved.

point out here that there is no “simulation mode”; that is, the generated data may be written to the database. Therefore we also recommend that CATT and eCATT are not used in production clients.

Test Scenarios for SAP R/3 and for mySAP Business Suite Landscapes

With CATT (Computer Aided Test Tool) SAP has provided a test tool, delivered with the SAP system since SAP Basis 3.0. You can use this to test classic, screen-based transactions within an SAP system (such as SAP R/3).

SAP tools generally offer greater flexibility in comparison to external test tools. Actions on the GUI can be recorded in this context. It is also possible to access database tables directly (that is, without using function modules).

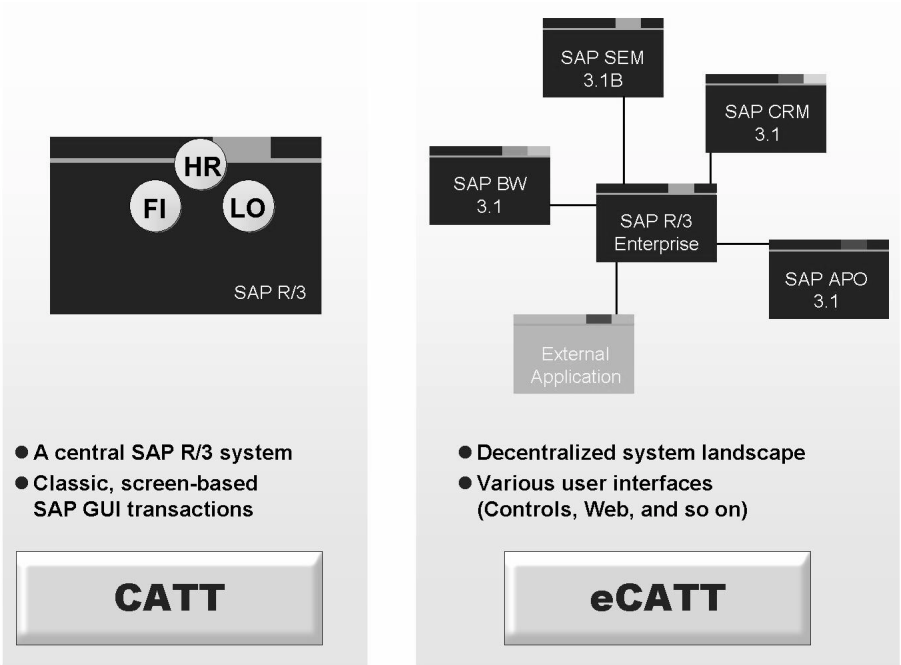


Figure 73: Uses of CATT and eCATT

The use of different components (such as SAP APO, SAP CRM, SAP R/3, and third-party tools) in the context of mySAP Business Suite means that it is necessary not only to test the individual components, but also their interaction in the context of cross-system business processes. CATT reaches its boundary at this point. It is also not possible use CATT to test transactions in SAP GUI for HTML or to test new functions such as Business Server Pages. To take these new developments into account, SAP developed the eCATT (extended CATT) test tool, which was first delivered with SAP Web AS 6.20. Properties of eCATT are:

© 2003 SAP AG. All rights reserved.

- eCATT enables you to test all SAP transactions, regardless of whether they use the Control Framework or not
- eCATT provides a simplified concept for performing tests in different systems, starting with a central test system
- eCATT extends the possibilities provided by CATT
- eCATT is fully integrated into the Test Workbench
- eCATT test objects can be generated from CATT test cases (migration)

The migration of CATT test cases to eCATT test cases is performed on the initial screen of transaction SECATT by choosing *eCATT Object* → *Migrate CATT*.

To use eCATT in the production environment: if you are not sure about exactly what a test case does, you should not use it in the production environment. It could change or create data, or even cause data inconsistency in the system. This is the reason why we recommend that eCATT is not used in the production environment. Customers can, of course, use eCATT nevertheless, although entirely at their own risk - customers should be made particularly aware of the risks.

You can use eCATT to test SAP transactions both on the SAP GUI for Windows and on the SAP GUI for Java. You can also test applications that run outside these two GUIs. You can access the contents of database tables, that is, both application and Customizing data, directly. This is illustrated by the following figure.

© 2003 SAP AG. All rights reserved.

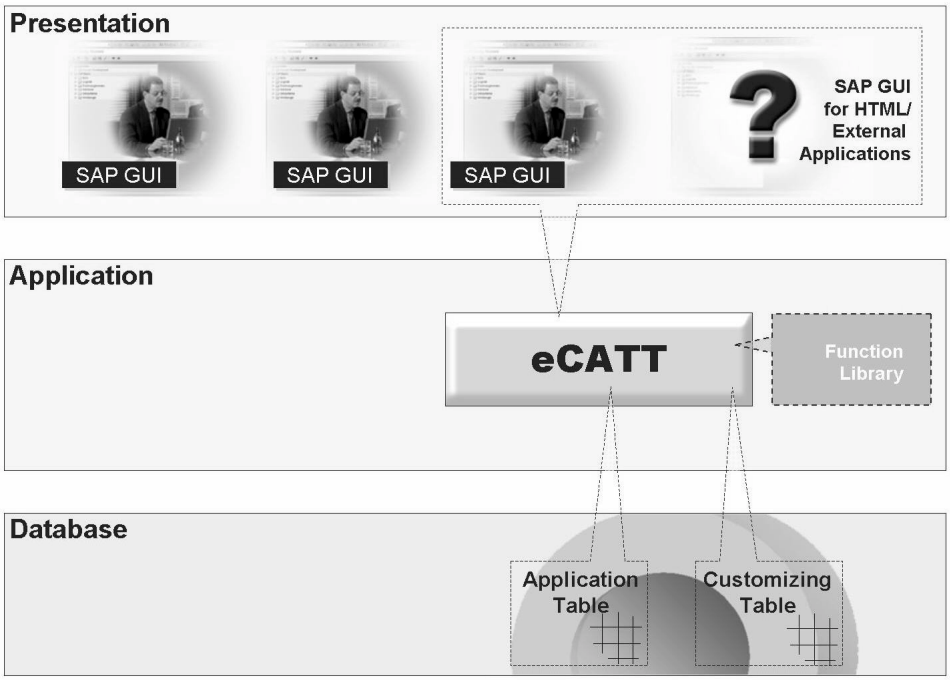


Figure 74: Test Possibilities with eCATT

Highlight at this point that eCATT can be used to test applications in SAP GUI for Windows and SAP GUI for Java, and also SAP GUI for HTML and external applications. It is also possible to use function modules and BAPIs and to access database tables directly.

Prerequisites for using eCATT are:

- The central test system must contain (at least) SAP Web AS 6.20. Therefore, an SAP R/3 Enterprise system, a standalone SAP Web AS, or a system with a correspondingly high Solution Manager release would be suitable
- The target systems must contain SAP Basis 4.6C or higher

It is still possible in principle to perform a test, even if the target systems have a lower (SAP Basis) release (such as SAP Basis 4.6B) by accessing the CATT functions. For more information, see SAP Note 496286. In this way, you can use, for example, the eCATT command `REMOTECATT` to call the CATT Interpreter in the target system.

© 2003 SAP AG. All rights reserved.

Central Terms in the eCATT Environment

When defining an automated test using eCATT, you must define test scripts, test data containers, system data containers, and test configurations.

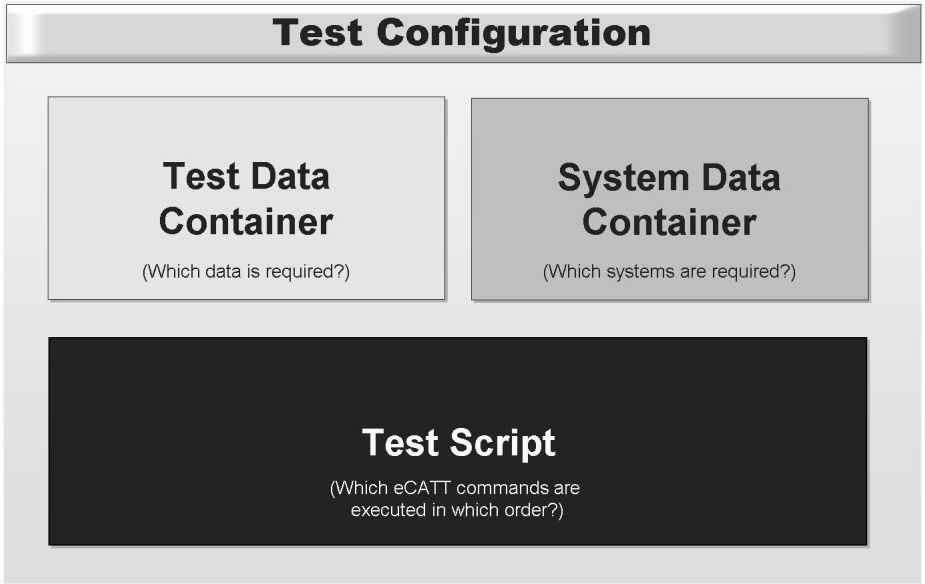


Figure 75: Central Terms in the eCATT Environment

A **test script** consists of a range of commands that control the process flow of the test (which transactions are executed in which sequence on which systems).

Test data containers consist of reusable data records that can be transferred to the test script. Unlike CATT, with which the test data belonged to a test script, with eCATT, the test data containers are stored separately from the script. This means that there is a greater degree of reusability with regard to the test data containers.

A **system data container** consists of a list of systems (with associated RFC destinations) that are required during the execution of a test script.

Finally, the **test configuration** links a test script ("what is to be done?") with the data from one or more test data containers ("which data is to be used?") and the system landscape defined in a system data container ("where are the tests to be performed?"). In this way, a test case (that is, a checkable entity of an SAP system that contains a procedure for testing an object) is uniquely defined. The test configurations are executable test objects that are included in test packages and that can be assigned to testers using the Test Workbench.

© 2003 SAP AG. All rights reserved.

Background information: A fully defined test configuration can be added to a test catalog using Test Catalog Management (transaction STWB_1, menu path *Tools* → *ABAP Workbench* →*Test* → *Test Workbench* →*Test Organizer* → *Test Catalog Management*). To do this, select an appropriate test catalog on the initial screen of transaction STWB_1 and then choose *Change* (or create a new test catalog, select this using the F4 help after saving it, and then choose *Change*). On the next screen, choose *Insert as Subnode*. Then choose the *Test Case* pushbutton. In the *Type* field, choose *eCATT Test Configuration*, in the *Name* field, enter the name of the test configuration, and choose *Continue*. You can save the test catalog and choose *Run Test Case* (*Test Case* pushbutton) to run the test case.

Note: The test configurations in eCATT replace the concept of variants in CATT.

Designing an Automated Test Series

Before a test series can be executed, you must answer a number of central questions:

1. Which applications are to be tested?
2. What is the system landscape like?
3. How can an automated test for this landscape best be created?
4. Which data is required to perform the test?
5. How can the test cases be organized?

The question about the applications to be tested includes clarifying the project aim and the project scope. At this stage, you have to decide how precisely you want to test (only superficially or very thoroughly?) and exactly what is the structure of the process flow of the business process to be tested - if appropriate, across system boundaries.

You then define which system is selected as the central test system and which systems have to be accessed during the test. You can then use this information to create the system data container.

In the next step, the test scripts can be created by test developers. Only systems that were defined in the system data container can be called from the test scripts. While in CATT, it was essentially only possible to record SAP transactions, there are different “drivers” in eCATT, depending on the application to be tested. In this case, you should consider which driver is to be used before a test.

© 2003 SAP AG. All rights reserved.

The situation with the different drivers is as follows: a driver to test function modules (`FUN`), a driver to test transactions that do not use the Control Framework (`TCD`), a driver to test transactions that use the Control Framework (`SAPGUI`), and a driver to test using external tools.

You can then compile the data required for the test in one or more test data containers.

Now that the system data container, test script, and test data containers exist, you can create the test configuration. This can then be executed by the individual testers, for example, using the Test Workbench (*Easy Access Menu → Tools → ABAP Workbench → Test → Test Workbench*).

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

The participants should discuss the advantages of automated tests.

Discussion Questions

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

In which situations do you use tests?

What aim do these tests have?

Which test tools do you use for these tests?

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the uses of eCATT
- List the advantages of eCATT compared to CATT

Related Information

For more information about extended CATT, see the online documentation in the area *eCATT: Extended Computer Aided Test Tool* (BC-TWB-TST-ECA). For more information about CATT, see the *Computer Aided Test Tool* area (BC-CAT).

© 2003 SAP AG. All rights reserved.

Lesson: Configuring eCATT

191 Lesson Duration: 40 minutes

Lesson Overview

In this lesson, you will obtain an overview of important system and client settings that are required to run eCATT scripts in a complex system landscape. In addition to setting up RFC connections between the central test system and the target systems to be monitored, a system data container will be created. Finally, the authorizations required to run eCATT scripts are discussed.

Lesson Objectives

After completing this lesson, you will be able to:

- List security issues connected to using eCATT
- Make system and client settings related to eCATT
- Create a system data container

In this lesson, which is important for system administrators, the participants should learn which system and client settings are required so that eCATT scripts can be run in a complex system landscape. You should emphasize to participants that enabling cross-system eCATT scripts has consequences for the security settings in the test environment (such as allowing activation of GUI scripting).

We recommend trusted RFC connections in the eCATT environment. Setting up a trusted RFC connection of this type in the course would, however, go well beyond the scope of this lesson, which is why a "normal" RFC connection is used in the exercises and in the instructor demonstration. You should, however, prepare for participant questions related to creating trusted RFC connections. For information about this, search for "trust relationships" or "Trusted System" in the "RFC Programming in ABAP" area of the online documentation, or the courses in the security field (such as the ADM960 training course).

Business Example

Your company wants to implement an SAP Web Application Server as the central test system for eCATT. As a member of the system administration team, it is your task to make the system settings required to do this in the SAP systems affected, while taking security issues into account.

© 2003 SAP AG. All rights reserved.

RFC Connections Between the Systems Involved

To be able to run eCATT scripts in a landscape consisting of multiple systems, you must first configure RFC connections from the central test system to the systems to be tested.

You can create RFC connections using the *RFC Destinations* transaction; transaction code SM59, , or choose *Tools* → *Administration* → *Administrations* → *Network* → *RFC Destinations.*

Note: The exact procedure for creating RFC connections between detailed SAP systems using transaction SM59 is not within the scope of this lesson.

You should be able to expect the participants to have knowledge about creating an RFC connection. You do not need to go into further detail about setting up RFC destinations using transaction SM59.

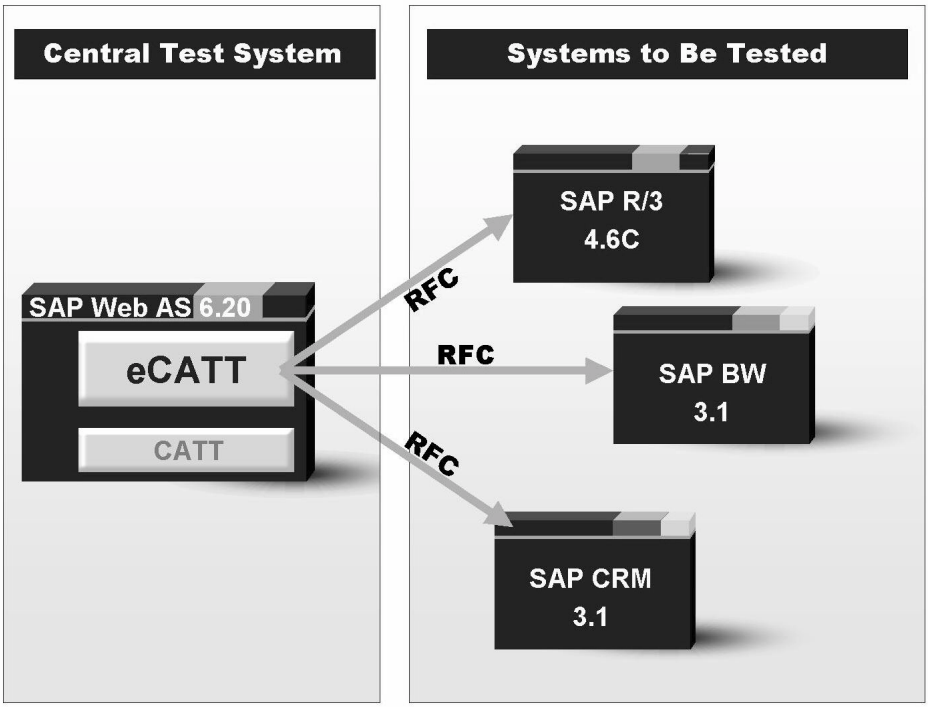


Figure 76: RFC Connections between the Central Test System and the Target Systems

© 2003 SAP AG. All rights reserved.

Digression: Trust Relationships Between SAP Systems

We recommend that you set up trust relationships (trusted RFC connections) in the context of running eCATT scripts across system boundaries.

If the current system has been registered as a trusted system in the target system, the Remote Function Call can be performed in the target system without the need to transfer a password. The trusted system relationship applies only for one direction; it is not bi-directional. If the relationship is to be bi-directional, you must register each partner system with the other as a trusted system.

Using trust relationships in the eCATT environment is useful for the following reasons:

- To run transactions using the `SAPGUI` command (for example, for transactions that use controls), you require a dialog user on the target system; it is not sufficient in many cases to specify a user of the type *System* (which is sufficient in CATT for the `TCD` command to run classic transactions using the batch input interface).
- If a user is entered (with a password) in the RFC destination of transaction SM59, any user of the calling system can (in principle) log on to the target system using this connection and run transactions under the user specified in the RFC destination. This is undesirable for security reasons.

If participants ask questions about setting up trust relationships using transaction SM59, refer them to courses in the security fields (such as the ADM960 course).

When using a trust relationship, it is necessary that the same user name exists in both systems. This user must have the authorizations for the authorization object “S_RFCACL” in the target system (these are **not** included in SAP_ALL).

Trust relationships between SAP systems have the following advantages:

- It is possible to use a single logon across system boundaries
- User-specific logon data is checked in the trusting system, that is, no passwords are transferred in the network

© 2003 SAP AG. All rights reserved.

At this point, you should create an RFC connection to an SAP partner system (connection type "3") (for example, with the ADM102 course setup, you could create a connection from the DEV system to the QAS system on the same host) and assign a descriptive name to it. This name is required again later when creating a system data container. On the *Logon/Security* tab page, you should specify a user that exists in the partner system. For the ADM102 course, for example, you can specify the user "ECATT" in client 100 with the password "remote" here. If you want to and have the required knowledge, you can also set up a trust relationship between the two SAP systems at this point. Otherwise, it is enough to highlight at this point that we recommend the use of trust relationships (trusted RFC connections) in the eCATT environment, but that this is not the main subject of this lesson.

Creating a System Data Container

A system data container stores the system landscape for a test scenario. It contains a list of all systems that a test script can access. Every entry in this list describes a system and consists of:

- A name for the target system (this should be chosen to be as descriptive as possible, to reflect the function of the target system, and to be independent of the name of the RFC destination)
- A description of the relevant software component (such as SAP R/3 Enterprise or SAP APO), purely for documentation purposes
- An RFC destination that points to the target system

The system data container should be maintained by a project manager for a test landscape, so that all members of the project team access the same system data container.

In a test script, a target system is identified using the name for the target system (and not using the name of the RFC destination). This has the advantage that if you change your system landscape, you can change the names of the target systems without having to adjust the test script. You only need to change the entries for the RFC destinations in the system data container. Optionally, you can create a new system data container and link it with the test script using a test data configuration.

© 2003 SAP AG. All rights reserved.

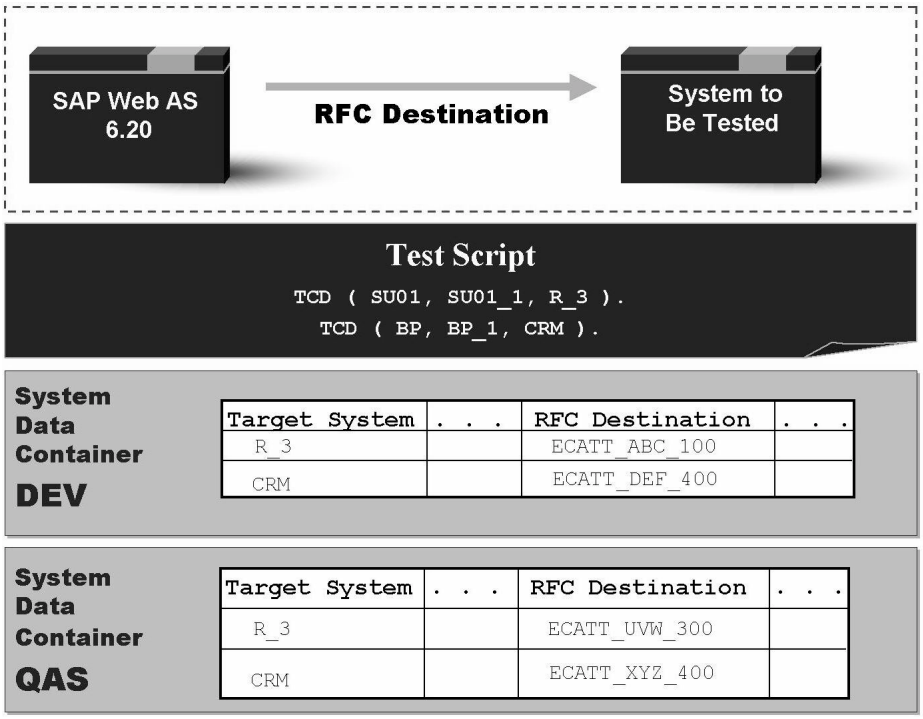


Figure 77: Meaning of System Data Containers

This graphic illustrates that, for example, a development landscape (DEV) and a test landscape (QAS) can exist for different system data containers. When executing the (identical) test script - on the one hand with the DEV system data container, and on the other with the QAS system data container - different RFC destinations are accessed, depending on the definition of the corresponding system data container.

Hint: System data containers can be transported using the SAP transport system (as can all other eCATT objects).

At this point, you should demonstrate creating a system data container to the participants. To do this, create system data on the initial screen of transaction SECATT (when doing this, take into account the naming convention; a possible name would, for example, be Z<course name>-00). On the *Attribs* tab page, enter a self-explanatory title. You can optionally also specify search terms, which users can use later to search for the system data container on the initial screen of transaction SECATT. On the *System Data* tab page, there is already an entry, “NONE”, that points to the “NONE” RFC destination and therefore allows local execution of individual eCATT commands at any time. You can create an additional

 © 2003 SAP AG. All rights reserved.

entry here that uses the RFC destination defined in the previous section of this lesson. In the course setup for ADM102, for example, the entry for the DEV system could be as follows:

Test System

Partner System

Product Version

SAP Web AS 6.20

RFC Destination

<the RFC destination defined in the previous section of this lesson>

A command in a test script that is to run transaction SU01 on the partner system could then be, for example, `TCD( SU01, SU01_1, partner system ).`

Other Settings

To enable eCATT scripts to be run successfully, the systems in the system landscape that are involved must be configured appropriately. This configuration includes:

- Maintaining client settings for eCATT in transaction SCC4 .
- Recording and running transactions that use controls (SAPGUI command in the eCATT script):
  - Using the SAP GUI for Windows 6.20 (with the SAP GUI scripting component)
  - System parameter *sapgui/user_scripting* = TRUE
  - SAP GUI settings on the front end: activate scripting
- Using external tools:
  - Maintaining the Customizing table
  - Creating a service user for communicating with the external tool

Client Settings

The maintenance of client settings using transaction SCC4 has been extended for eCATT. The extensions relate, in particular, to setting up remote connections. You can use trusted RFC connections to transfer security settings from the start system to the remotely started test system. With the extended client settings, you can not only allow or forbid the starting (as is the case with CATT), but can also make it wholly or partly dependent on the type of RFC connection. It is now possible to make the following settings:

© 2003 SAP AG. All rights reserved.

eCATT and CATT Not Allowed

eCATT and CATT Allowed

eCATT and CATT Allowed for Trusted RFC Only

eCATT Allowed; FUN/ABAP and CATT Not Allowed

eCATT Allowed; FUN/ABAP and CATT for Trusted RFC Only

Reason for this New, Additional Differentiation:

- eCATT authorizations are sufficient to add ABAP blocks to an eCATT. These blocks are always executed in the default target system of the script. This may be a system on which ABAP authorizations are **not** normally assigned.
- With the command `FUN`, function modules that are not designed for remote execution can be run **remotely**.

The eCATT functions `FUN` (calling function modules) and `ABAP` (blocks of ABAP commands within a test script) have particular importance for security reasons, meaning that they should be additionally secured using the extended client settings.

At this point, you can demonstrate the extended client setting for your client using transaction SCC4 (or the equivalently by using transaction SM31 and selecting table T000). Discuss the options under *Restrictions when Starting CATT and eCATT*. Ensure that executing CATT and eCATT is permitted for your client. You can optionally also check the corresponding settings in your partner system (that is, the system/the client to which the RFC connection that you have created points).

Hint: For more information about the extended security concept for CATT and eCATT, see SAP Note 496286.

Settings for Using GUI Scripting

With eCATT, it is possible (using the `SAPGUI` command) to use the scripting function of SAP GUI (as of SAP GUI for Windows 6.20). This means that you can use a new recorder in transaction SECATT to record every action on the SAP screen, regardless of whether the actions use controls or not.

© 2003 SAP AG. All rights reserved.

At this point, you can highlight the fact that GUI scripting can also be used on a "standalone" basis, that is, it may have to be deactivated so that other scripts (such as Visual Basic scripts) cannot address the GUI.

If this function is to be used, you must first activate scripting. You do this as follows:

- The system parameter *sapgui/user_scripting* must have the value TRUE (you can use transaction RZ11 to change this parameter dynamically). This parameter is set to FALSE by default.
- SAP GUI for Windows 6.20 (or above) must be installed on the front end of the user's computer (with the SAP GUI Scripting component)
- GUI scripting must be permitted on the user's front end computer. You can do this by choosing the *Customizing of Local Layout* button in the system function bar and then choosing *Options ...* → *Scripting* → *Enable Scripting*.

At this point, you can show the participants the current value of the system parameter *sapgui/user_scripting*using transaction RZ11, and - if you want to use the `SAPGUI` eCATT command in a test script later - change it to "TRUE", if required. You should also demonstrate for the participants how GUI scripting can be enabled on the front end (see the last point of the bulleted list above), and enable it if required.

Caution: If you change this parameter, you may have to log off and then log on again so that the change takes effect.

When recording and running transactions with the eCATT command `SAPGUI`, the settings are required in the central test system and in the target system (that is, scripting must be enabled in both systems).

Hint: If the system parameter *sapgui/user_scripting* is set to FALSE, the front end settings do not take effect.

Note: For more information about system parameter *sapgui/user_scripting*, see SAP Note 480149.

Settings for Using External Tools

eCATT supports the connection of certified external tools. You can also use these tools to test applications that do not run under the SAP GUI (such as Business Server Pages and the SAP GUI for HTML).

© 2003 SAP AG. All rights reserved.

To be able to use external tools, you may have to install the external tool on the user's local machine and:

- Use the function module *SET_EXTERNAL_TOOL* to make the required entries in the Customizing table **ECCUST_ET**
- Use the program *ECATT_GENERATE_ET_USER* in transaction SE38 to create the service user required for communication with the external tool

Note: For more information about setting up SAP systems for using eCATT, see SAP Note 519858.

If you have a SAP Service Marketplace user and know the relevant link, you can optionally branch to the SAP Service Marketplace at this point, and show the certified vendors (of external tools) for the Extended Computer Aided Test Tool there (using a search by software category).

Required Authorizations

The following authorization objects are required to use eCATT:

- *S_TCODE*: You can use this object to allow the user selective access only to certain transactions (for example, to the transaction SECATT).
- *S_DEVELOP*: *S_DEVELOP* is the authorization object that regulates access to all development objects in the SAP system. The granularity of this authorization object (with the fields *DEVCLASS* for packages, *OBJTYPE* and *OBJNAME* for object types and names of objects that the user can change, *P_GROUP* for specific program groups, and *ACTVT* for permitted activities such as change or create) allows you to assign authorizations specifically tailored to the user's requirements.
- *S_RFC*: Authorization check for RFC access: You can use this to restrict the execution of function modules that are in specific groups. To create trust relationships, *S_RFCACL* is also required.
- *S_ADMI_FCD*: *S_ADMI_FCD* is a system administration authorization object that is checked if a user wants to create an RFC connection.

The following table shows the authorizations required for test developers and testers. In addition, both the testers and the test developers require authorizations for the transactions to be recorded or tested.

© 2003 SAP AG. All rights reserved.

Required Authorizations in the eCATT Environment

Authorization Object	Field	Value for Test Developers	Value for Testers
S_TCODE	TCD	SECATT, SM59	STWB_WORK
S_DEVELOP	OBJTYPE	ECAT, ECSD, ECTC, ETCD	ECTC (test configuration)
	ACTVT	01 (Create), 02 (Change), 03, 06 (Delete), 16	03 (Display), 16 (Execute)
	DEVCLASS	Names of the required packages	Names of the required packages
	OBJNAME	Names of the objects to be tested	Names of the objects to be tested
S_RFC	RFC_TYPE	FUGR (function group)	FUGR (function group)
	ACTVT	16 (Execute)	16 (Execute)
	RFC_NAME	SCAT, STTM, STTF (for eCATT), SBDR (for the TCD command), SDYN (for the SAPGUI command), SYST, ECATT_EXTER-NAL_TOOL	STTF (eCATT help functions)
S_ADMI_FCD	S_ADMI_FCD	NADM (access to all RFC destinations)	

Hint: For more information about authorizations in the eCATT environment, see the eCATT Security Guide, which you can find under the *securityguide* Quick Link in the SAP Service Marketplace.

If there are separate systems developing tests and testing, the test developers also require authorization for the object *S_TRANSPORT*.

© 2003 SAP AG. All rights reserved.

If you are working with an external tool, this is started from eCATT. To do this, eCATT generates the user required for the RFC connection to exchange exchange the script data (the program *ECATT_GENERATE_ET_USER* described in the previous section is used for this purpose). If, however, scripts are to be uploaded from an external tool without starting the tool from eCATT, the user must log on to the SAP system himself or herself. To do this, the user requires the authorization object *S_RFC* with *RFC_TYPE* = “FUGR”, *ACTVT* = “16,” and *RFC_NAME* = “SYST”, “ECATT_EXTERNAL_TOOL”. These authorizations are contained in the role *SAP_ECET*.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

199

Exercise 8: Configuring the System Landscape for Using eCATT

Exercise Duration: 15 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Make system and client settings related to eCATT
- Create a system data container

Business Example

As a member of the system administration team, you are configuring your SAP system landscape for using eCATT.

System Data

System: This depends on the course setup in use. An SAP system with SAP Web AS 6.20 is required. For the ADM102 course, there are two SAP systems (DEV and QAS) of this type on each host, between which the participants should also set up RFC destinations.
Client: This depends on the course setup in use. For the ADM102 course, client 100 is used.
User ID: This depends on the course setup in use. A course user that has authorization to change entries in transaction SCC4, to create RFC destinations in transaction SM59, and to create a system data container in transaction SECATT is required. The user must also be able to run transaction RZ11 and the report RSPFPAR. For the ADM102 course, the ADM102-## users (where ## is the group number) have the relevant authorizations.
Password: This depends on the user used. The ADM102-## users have the initial password init.
Set up instructions:

1. The system settings in transaction SCC4 for the training client should be as follows for the “Restrictions when Starting CATT and eCATT” area: “eCATT and CATT Allowed”.

© 2003 SAP AG. All rights reserved.

Task 1

For the ADM102 course, the participants using the DEV system should create an RFC connection to the QAS system on the same host (and the participants using the QAS system should create an RFC connection to the DEV system on the same host), and should use the user "ECATT" with the password "REMOTE" to log on to the target system.

Create an RFC connection to your partner system.

1. Create an RFC connection with the name **ECATT_##** (where ## is your group number) in the SAP specified by your instructor and use the user assigned by your instructor to log on to this target system.

Task 2

Check whether the execution of eCATTs is allowed in your SAP system and whether the execution of scripting commands is allowed.

1. Use transaction SCC4 to check whether eCATTs can be executed in your SAP system.
2. Check whether the execution of GUI scripting command is allowed in your SAP system.
3. Determine whether your local GUI permits the execution of scripting commands.

Task 3

Create a system data container. An entry in the container should point to the RFC destination created above.

1. Use transaction SECATT to create a system data container with the name **ZECATT_SDC_##**, where ## is your group number, and give the system data container a self-explanatory title.
2. Add an additional target system to your system data container on the *System Data* tab page. Name this target system **Remote_R3**, maintain an appropriate product version, assign the name of the RFC destination that you created above to the target system, and save your settings.

© 2003 SAP AG. All rights reserved.

Solution 8: Configuring the System Landscape for Using eCATT

Task 1

For the ADM102 course, the participants using the DEV system should create an RFC connection to the QAS system on the same host (and the participants using the QAS system should create an RFC connection to the DEV system on the same host), and should use the user "ECATT" with the password "REMOTE" to log on to the target system.

Create an RFC connection to your partner system.

1. Create an RFC connection with the name **ECATT_##** (where ## is your group number) in the SAP specified by your instructor and use the user assigned by your instructor to log on to this target system.
   a) Call transaction SM59 (Display and maintain RFC destinations: *Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations*) and choose *Create*.
   b) In the *RFC Destination* field, enter the name of the RFC destination, **ECATT_##**. Choose *3* as the connection type. Enter a short, self-explanatory description for the RFC destination and choose *Save*.
   c) On the *Technical Settings* tab page, enter the target host and the system number of your partner system.
   d) On the *Logon/Security* tab page, enter your logon language and the data for the user that your instructor has given you. Save your settings.
   e) Check your settings by choosing *Test Connection*.

Task 2

Check whether the execution of eCATTs is allowed in your SAP system and whether the execution of scripting commands is allowed.

Continued on next page

 © 2003 SAP AG. All rights reserved.

1. Use transaction SCC4 to check whether eCATTs can be executed in your SAP system.
   a) Call transaction SCC4 (*Client maintenance*, menu path*Tools* → *Administration* → *Administration* → *Client Administration* → *Client Maintenance*).
   b) Select your logon client and choose *Detail*. Under *Restrictions when Starting CATT and eCATT*, you can see whether CATTs and eCATTs can be executed in your client. If necessary, change the setting to *eCATT and CATT Allowed* (by first choosing *Display* ↔ *Change*) and save your settings.
2. Check whether the execution of GUI scripting command is allowed in your SAP system.
   a) Run report **RSPFPAR**, either using transaction SA38 (*System* → *Services* → *Reporting*) or transaction RSPFPAR.
   b) On the selection screen, restrict the selection of the profile parameters to *sapgui**. Select the *with Short Description* checkbox and choose *Execute*.
   c) As a result, the system displays the value of the system profile parameter *sapgui/user_scripting*. If this parameter has the value *FALSE*, the execution of scripting commands is not permitted in your SAP system (regardless of the settings of your local GUI). You can, optionally, change the value of this parameter dynamically (that is, during running operation) using transaction RZ11. To do this, select the parameter in the output list of report **RSPFPAR** and then choose *Call Trans. RZ11*. On the initial screen of transaction RZ11 (Display Profile Parameter Attributes), choose *Change Value*. In this way, you can change the value to *TRUE* simultaneously for all instances (upper case letters are required). Then log off and on again so that the change to the profile parameter takes effect for your user.
3. Determine whether your local GUI permits the execution of scripting commands.
   a) In the system function bar, choose *Customizing of the Local Layout*. Then choose *Options ...* → *Scripting*. If the *Enable Scripting* checkbox is selected, the execution of GUI scripting is possible on your local GUI.

Task 3

Create a system data container. An entry in the container should point to the RFC destination created above.

Continued on next page

 © 2003 SAP AG. All rights reserved.

1. Use transaction SECATT to create a system data container with the name **ZECATT_SDC_##**, where ## is your group number, and give the system data container a self-explanatory title.

   a) Call transaction SECATT (Extended Computer Aided Test Tool: *Tools* → *ABAP Workbench* → *Test* → *Test Workbench* → *Test Tools* → *Extended CATT*). Select the *System Data* radio button, enter the name of the system data container, **`ZECATT_DC_##`**, in the associated entry field, and choose *Create Object*.
   b) On the *Attribs* tab page, enter a descriptive title in the *Title* field of the *Header Data* area.

2. Add an additional target system to your system data container on the *System Data* tab page. Name this target system **Remote_R3**, maintain an appropriate product version, assign the name of the RFC destination that you created above to the target system, and save your settings.

   a) Switch to the *System Data* tab page, and choose *Insert Row*. In the *Test System* field, enter the value **`Remote_R3`**. In the *Product Version* field, maintain the relevant version of your target system, such as *SAP Web AS 6.20*. In the *RFC Destination* field, enter the name of the RFC destination that you defined above (such as ECATT_##; you can display the available RFC destinations with the F4 help).
   b) Save your settings, and choose the package *$TMP* when doing so (or choose *Local Object*). Information about the selected target system should now be displayed in the *Instance Description* field.

Result

You have learned about a number of system settings for using eCATT and created a system data container.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- List security issues connected to using eCATT
- Make system and client settings related to eCATT
- Create a system data container

Related Information

For more information about configuring SAP systems for using eCATT, see:

- SAP Notes 519858 (Setting up R/3 systems to use eCATT), 496286 (Security concept extended for CATT and eCATT), and 480149 (New profile parameter for user scripting on the front end)
- The eCATT Security Guide on the SAP Service Marketplace (Quick Link */securityguide*)

© 2003 SAP AG. All rights reserved.

Lesson: Executing eCATT Test Scripts

205 Lesson Duration: 30 minutes

Lesson Overview

eCATT test scripts contain the commands to be processed to perform a test and the sequence in which they should be executed. This lesson provides you with an overview of the structure of test scripts. You will learn about the test script editor and various drivers for eCATT. You will also learn how you can monitor the execution of eCATT scripts.

Lesson Objectives

After completing this lesson, you will be able to:

- Create a simple test script using eCATT and monitor its execution
- Define the implementation areas of various drivers for eCATT

The participants should become familiar with the test script editor so that they can create their own test script at the end of this lesson and monitor its execution. As the instructor, you should present the various drivers and their uses. It could be useful for the participants if you highlight here that there was not an equivalent to the SAP GUI driver or to connecting external tools in CATT.

Business Example

You have configured SAP systems for performing eCATT in your company, and now want to create and execute test scripts.

Structure of a Test Script

The sequence of commands that are to be executed during a test is defined in a test script.

© 2003 SAP AG. All rights reserved.

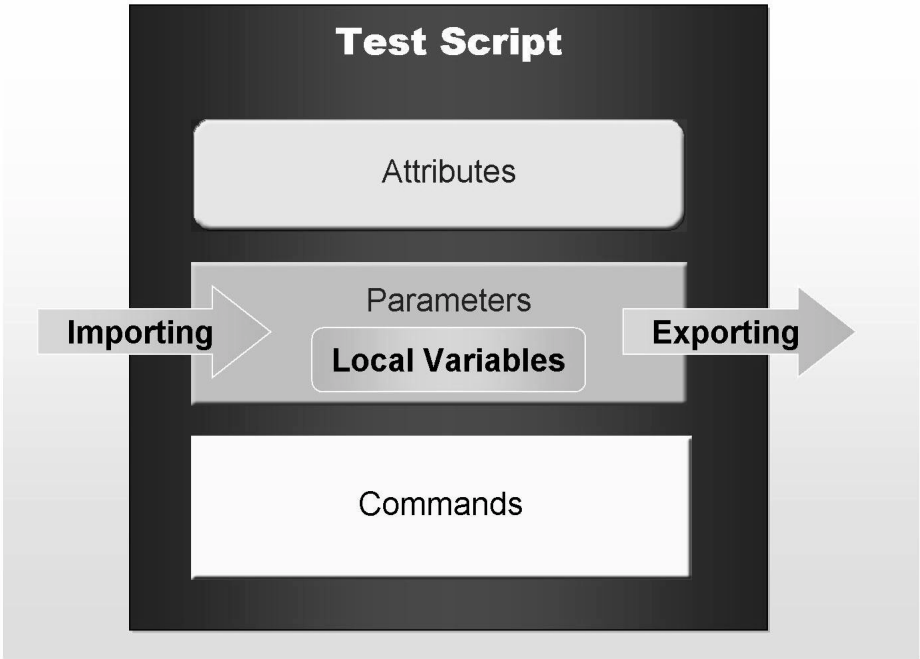


Figure 78: Structure of a Test Script

A test script consists of three main parts:

- Attributes
- Parameters
- Commands

Attributes define the properties of a test script. As test scripts are elements of the repository, they require a title and an assignment to a package. You must also specify an application component and a person responsible for them. If necessary, you can also specify restrictions about the validity of a test script (such as a restriction to a specific database release or a specific application server platform).

Parameters are variables. They are used to transfer data from an external source (such as from a test data container) to the script (importing parameters); to transfer data from the script to external sources (exporting parameters); or to define variables locally in the script (local variables).

Commands are the commands that define the process flow of the script. Commands are used to control which transactions on which target system are processed in which order, and branches and loops can also be evaluated, if required.

© 2003 SAP AG. All rights reserved.

The Test Script Editor

You can maintain a test script using the test script editor. You start the test script editor from the initial screen of transaction SECATT (*Tools → ABAP Workbench → Test → Test Workbench → Test Tools → Extended CATT*) by selecting the *Test Script* radio button, entering a name for the test script (this is an element of the repository, so you should observe the naming convention here) and choosing *Create* or *Change*. The following figure shows the structure of the test script editor.

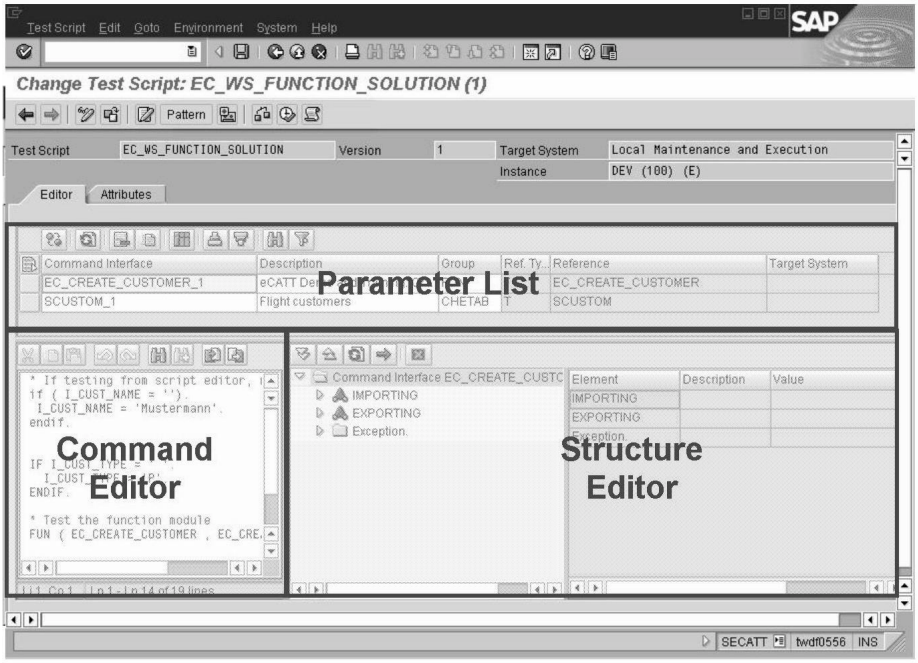


Figure 79: Structure of the Test Script Editor

Open the test script editor, for example, from the initial screen of transaction SECATT by creating a test script with the name Z<*course number*>_ECATTDEMO. On the *Attributes → General Data* tab page, enter a descriptive title and specify a component (such as *BC*). If you have already created a system data container, you can also specify this by choosing *Attributes → General Data → Maintenance System → System Data Container* (at the lower end of the *General Data* tab page). Then switch to the *Editor* tab page and display the structure of the test script editor. If the structure editor is not displayed, this is because no command interface has been defined yet. Now create a command that runs transaction SU01. To do this, choose *Pattern*. On the next screen, enter the following values: *Command* = *TCD (Record)* , *Transaction* = *SU01*, confirm the generated interface (such as *SU01_1*) and choose *Continue*. You then run through transaction SU01,

© 2003 SAP AG. All rights reserved.

and all of your entries are recorded. You could, for example, display the data for your user, choose *Back* and transfer the recorded data. As a result, the corresponding eCATT command is displayed in the command editor. You can optionally select another target system for this command from your system data container by changing the eCATT command to `TCD ( SU01 , SU01_1 , DEST ).` (where `DEST` is a target system from your system data container). You can use the *Parameter* ↔ *Command Interface* pushbutton to switch to the definition of the command interface, if necessary. If you double click the name of the generated command interface, the structure editor is opened. This provides you with an overview of the structure of the automatically generated interface (in this case, specifically for transaction SU01). Finally, save your test script, for example, as a *Local Object*.

Note: If you select the folder for a screen in the structure editor and choose *Simulate Screen*, the system displays the input screen of the transaction you selected. Although you can define input fields as variables here (for example, by entering **`&<variable name>`**), we do not recommend this procedure, as it can lead to problems with nested subscreens. Variables should therefore always be changed directly in the area on the right of the structure editor (the entry **`'T000'`** reads the **value** T000, while the entry **`T000`** refers to the **parameter** T000). With SAP Web AS 6.40, an appropriate parameter with the correct ABAP Dictionary reference is created directly in the test script at this point.

In the parameter list area, you can define import and export parameters and any local variables that you require for executing your test script. You can use the *Parameter* ↔ *Command Interface* button to switch to the definition of the command interface, if necessary. These command interfaces can be automatically generated when you generate certain eCATT commands (such as recording or running SAP transactions) by choosing the *Pattern* button.

You can branch to the structure editor by double-clicking the name of a command interface. In the structure editor, you can view the definition of the interface structure in more detail or define parameters for the interface.

You can enter eCATT commands in the command editor.

Selecting the Correct Driver

There are various drivers available in the eCATT environment. The figure below provides an overview of which drivers should be used for specific applications.

© 2003 SAP AG. All rights reserved.

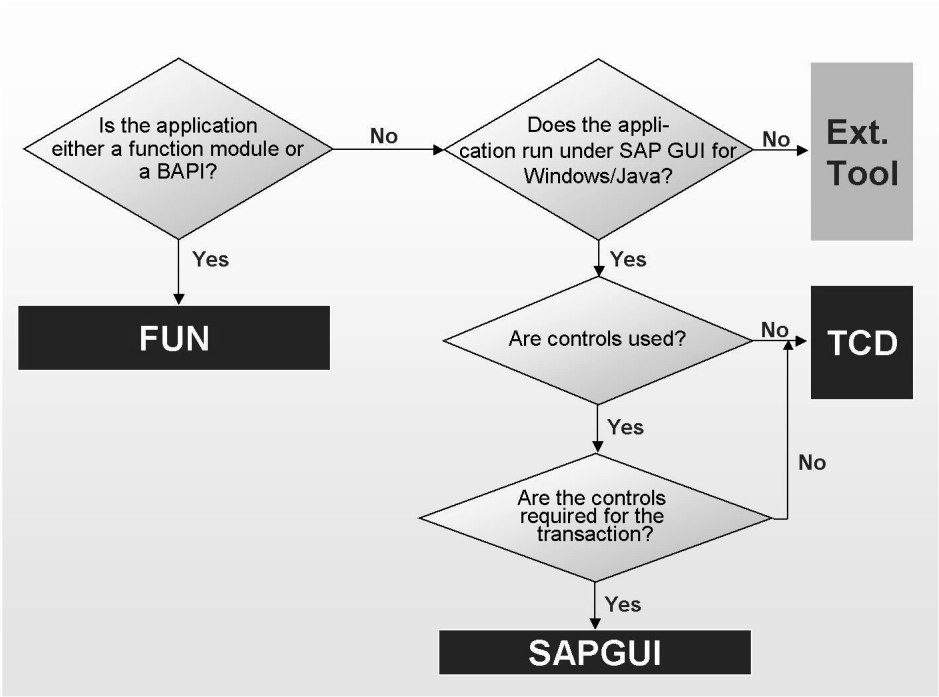


Figure 80: Selecting the Correct Driver

The `FUN` command is used to test function modules. The command interface of this command corresponds to the interface of the called function module, including the transferred parameters and exceptions.

The commands `SAPGUI` and `TCD` are used to test SAP transactions. Which of the two is used depends on whether the transaction to be tested uses controls or not. If controls are used, you must use the `SAPGUI` command, as `TCD` commands are run at application level, without an assigned SAP GUI. If controls are not used, we recommend the use of the simpler and more performant `TCD` command. The `SAPGUI` command uses a scripting engine of the SAP GUI to record status changes on the user interface. A SAP GUI session is also created when the command is run.

In addition to testing SAP transactions, eCATT provides an interface for external test tools that allows the testing of external applications. The scripts created with external tools are also stored centrally in the database of the SAP system.

© 2003 SAP AG. All rights reserved.

The connection of external tools cannot be demonstrated here, as there is usually no external tool installed on the local machines. If the SAP system was configured for the use of external tools, an additional selection field would appear on the initial screen of transaction SECATT: *External Tool* .

Note: In addition to the SAPGUI, TCD, and FUN commands, it is also possible to check special conditions (command`IF ... ENDIF.`), to define loops (command `DO ... ENDDO.`), to check values (command `CHEVAR` or `CHETAB`), to set temporary Customizing settings (commands `SETTAB` and `RESTTAB`), and to use blocks of ABAP coding (command `ABAP ... ENDABAP.`).

There are already a number of eCATT test scripts in the SAP system that describe the use of more advanced eCATT commands not discussed here. These begin with "EC_*" and you could look through them as part of your preparation for this lesson.

Test Data Containers

Test data containers consist of parameters that describe the interface of the container and of variants in which the actual test data is stored. Every variant contains a value for every parameter in the test data container. The names of the source parameters in the test data container must match the target parameters in the test script.

Note: In CATT, the variants consisted of tab separated entries in text files. This suggested the use of spreadsheet programs for this purpose (although the use of programs of this type was not required). In eCATT (in SAP Web AS 6.20), there is still no inbuilt function to import data into test data containers. However, a function of this type is planned for SAP Web AS 6.40. However, users can already "cut&paste" data from the spreadsheet program to the corresponding test data containers. With SAP Web AS 6.40, there will also be a simple way to transfer the parameters defined in a test script to a test data container (*Parameters* tab page), and to transfer parameters from a test data container to a test script.

© 2003 SAP AG. All rights reserved.

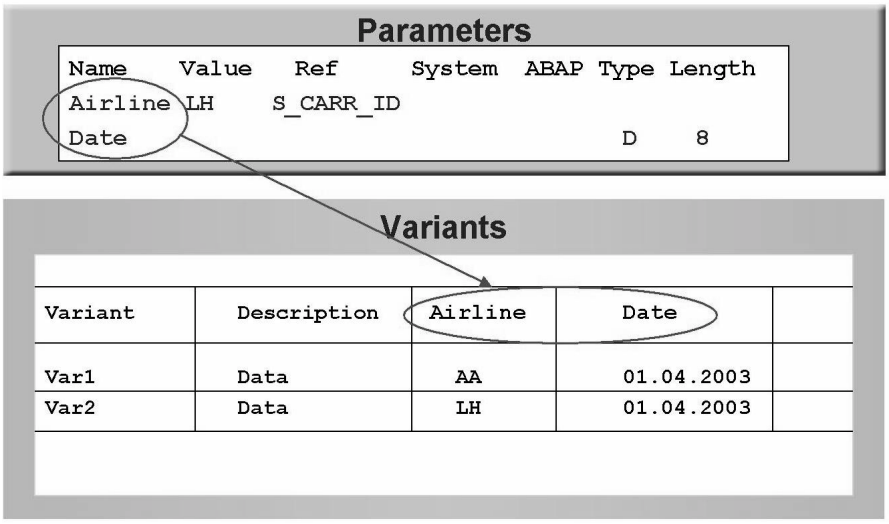


Figure 81: Structure of a Test Data Container

The simplest method is to define a separate test data container for each test script. However, this significantly reduces the options for reusing test data containers.

However, it is also possible to use a single test data container for multiple test scripts for an application. This simplifies consistent data storage.

In the case of larger quantities of parameters, using multiple test data containers can be useful. These provide the data required for a test project. Each script takes its data from one or more test data containers.

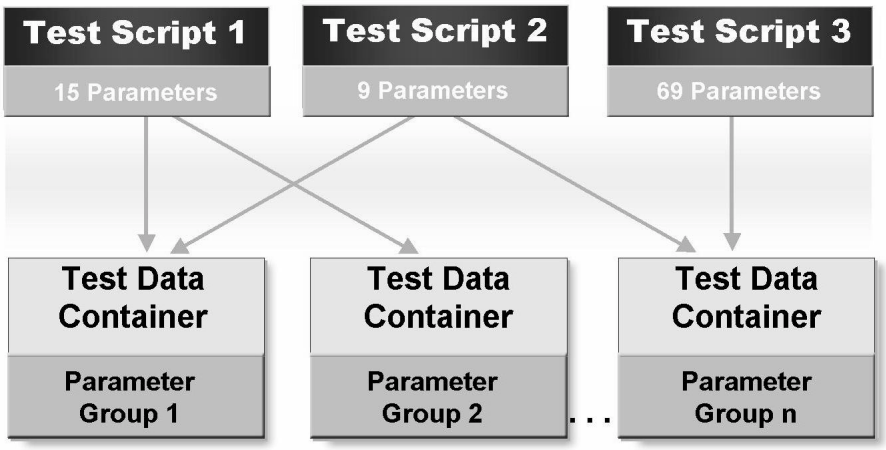


Figure 82: Reusability of Test Data Containers

© 2003 SAP AG. All rights reserved.

Monitoring the Execution of eCATTS Using Logs

Every eCATT run generates a log containing detailed test results. This log contains the name of the test script (with the version) and every command executed in the script, together with the corresponding parameters.

Every error that occurs when the script is being run is colored red and is passed up to the highest level of the hierarchical structure of the log, so that each error that occurs flags the run as containing errors. Specific entries can also be written to the log within a script using the command `log ( <var> ).`

Now run the test script defined at the beginning of this lesson by choosing *Execute Test Script*. If you want, you can also explain the different start options here. Open the tree structure of the log created after the transaction and display selected entries in the log.

Hint: While the system data container specified on the Attribs, General Data tab page during the definition of the test script determines the selected SAP system landscape at “design time”, this specification can be overridden when the test script is run (at “runtime”). Entering a test system specifies which system is used if no target system is explicitly specified in script commands.

© 2003 SAP AG. All rights reserved.

211

Exercise 9: Executing an eCATT Test Script

Exercise Duration: 15 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Create a simple test script for eCATT and monitor its execution

Business Example

You want to create and execute eCATT test scripts to test business processes in your SAP system landscape.

System Data

System: This depends on the course setup in use. An SAP system with SAP Web AS 6.20 is required.
Client: This depends on the course setup in use. For course ADM102: client 100
User ID: This depends on the course setup in use. A course user that has authorization to execute transactions SECATT and SU01, and the function module RFC_READ_TABLE is required. For the ADM102 course, the ADM102-## users (where ## is the group number) have the required authorizations.
Password: This depends on the user used. The ADM102-## users have the initial password init.
Set up instructions:

1. The system settings in transaction SCC4 for the "Restrictions when Starting CATT and eCATT" area for all clients used should be: *eCATT and CATT Allowed*.

2. A system data container and the associated RFC connection(s) to a remote system should exist for exercise step 4 of task 1 (the optional exercise). In the setup for the ADM102 course, there is a system data container, *ZADM102_SDC_DEMO*, that, among other things, references an RFC connection between the DEV system and the QAS system (client 100 in each case).

3. For exercise step 5 of task 1, the user must also be able to execute transaction RZ11 and change parameters there.

© 2003 SAP AG. All rights reserved.

Task

Create an eCATT test script that reads the contents of a table on your local system. Then run the test script and check the log created. You can then optionally extend your test script by recording and running an SAP transaction on a remote target system, and by recording and running a transaction that uses controls on your own system.

1. Create an eCATT test script called **ZECATT_##**, where ## is your group number. Give the test script a self-explanatory title and assign the test script to the component *BC*.
2. Switch to the test script editor and write an eCATT script that records the function module *RFC_READ_TABLE* on your system, where table T000 is to be read. Save your test script as a *Local Object*.
3. Run your test script and check the results of the test script using the log created.
4. Optional: Add an additional eCATT command to your test script so that a user master record is changed on a remote system. Use a system data container and a user on the remote system to do this. Your instructor will give you the details. Save your test script, execute it, and check the result of the execution.

A system data container, in which a target system other than the central system is specified, is required for this optional exercise. The RFC destination specified in the system data container must be maintained appropriately, and the execution of eCATTs must be allowed in the target system (transaction SCC4). There must also be an appropriate model user in the target client that can be changed as required. In the setup for the ADM102 course, there is a system data container, *ZADM102_SDC_DEMO*, that, among other things, references an RFC connection between the DEV system and the QAS system (client 100 in each case) and in the target clients, there is a user, *ECATTCHANGE*, for which the address, for example, can be changed. If you have performed the exercise for the Configuring eCATT lesson in your course, you can also link to the system data container created there.

---

5. Optional: Extend your eCATT test script so that, in your local system, starting from the *SAP Easy Access* menu, the menu path *Tools* → *ABAP Workbench* → *Overview* → *Object Navigator (SE80)* is called, and the program *RSPFPAR* is displayed in this transaction. To do this, you may have to use the SAP GUI installed on your SAP system (instead of the SAP GUI in your training room). In the

Continued on next page

 © 2003 SAP AG. All rights reserved.

hierarchy tree of transaction SE80, branch to *RSPFPAR* → *Subroutines* → *READ_ACTUAL_PAR_VALUES* after selecting the program, and display the relevant report by double-clicking the program name. Then complete the recording of the transaction and save the eCATT script. Execute the test script and display the log.

Caution: This optional exercise can only be executed if both the SAP system (parameter *sapgui/user_scripting*) and the local GUI (*Customization of the Local Layout* → *Options...* → *Scripting* → *Enable Scripting*) allow the execution of GUI scripting. If you want to let this exercise be run, ensure beforehand that all prerequisites are fulfilled. If this is not the case, it could be helpful if participants log on at the operating system level of their SAP systems (for example, using the RDP client), and use the SAP GUI installed there. Note, in particular, that - if you have changed the parameter sapgui/user_scripting in transaction RZ11 - you must log off and log on again so that the change takes effect for your user.

© 2003 SAP AG. All rights reserved.

Solution 9: Executing an eCATT Test Script

Task

Create an eCATT test script that reads the contents of a table on your local system. Then run the test script and check the log created. You can then optionally extend your test script by recording and running an SAP transaction on a remote target system, and by recording and running a transaction that uses controls on your own system.

1. Create an eCATT test script called **ZECATT_##**, where ## is your group number. Give the test script a self-explanatory title and assign the test script to the component *BC*.

   a) Start transaction SECATT (*Tools* → *ABAP Workbench* → *Test* → *Test Workbench* → *Test Tools* → *Extended CATT*).
   b) On the initial screen of this transaction, select the *Test Script* radio button, and enter the name of your test script in the associated input field.
   c) Choose *Create Object (F5)*.
   d) On the next screen, under *Attributes* → *General Data*, enter a self-explanatory title for the test script in the *Title* field, and in the *Component* field, select the component *BC*.

Continued on next page

 © 2003 SAP AG. All rights reserved.

2. Switch to the test script editor and write an eCATT script that records the function module *RFC_READ_TABLE* on your system, where table T000 is to be read. Save your test script as a *Local Object*.

   a) Switch to the *Editor* tab page. You are now in the test script editor. The simplest way to record a function module is to choose the *Pattern* button in the application toolbar. On the following screen, in the *Command* field, choose the entry *FUN*. In the *Function Module* field, enter the name *RFC_READ_TABLE* for the function module to be executed and choose *Continue*. The system now proposes an interface. Confirm the name of this interface by choosing *Continue* again.
   b) In the parameter list, define the parameter *TABLE*. To do this, choose the *Parameter* ↔ *Command Interface* button, if necessary, to switch to the parameter view, and create a new parameter (*Append Parameter* button).
   c) In the *Parameter* field, enter the name **`TABLE`**; in the *Value* field, enter **`T000`**; and in the *I/E/V* field, enter **`E`**, because the name of the table is an export parameter for the test script that is transferred to the function module. In the *ABAP Type* field, enter **`C`**, and in the *Length* field, enter **`16`**.
   d) Alternatively, you can also enter the value **`TABNAME16`** in the *Parameter Reference* field as a description of the associated data element. In this case, the information about the ABAP type and length is automatically determined by the SAP system using the ABAP Dictionary.
   e) Choose the *Parameter* ↔ *Command Interface* button to switch to the command interface view, and open the structure editor by double-clicking the name of the automatically generated command interface (such as *RFC_READ_TABLE_1*). In the structure editor, open the path *Importing* → *Query_Table* and double-click *Query_Table*. Then, in the area on the right of the structure editor, in the *Value* field, enter the name of the parameter you have just defined (*TABLE*). Finally, save the script and, on the following screen, *Create Object Directory Entry*, choose the *Local Object* button.
   f) As an alternative to steps b) to e), you can also branch directly to the structure editor and, in the *Value* field (*Importing* → *Query_Table*), enter the value **`'T000'`** to enter the name of the table directly for Query_Table (single quotation marks are required). In this case, you do not need to define any parameters in the parameter list.

Continued on next page

 © 2003 SAP AG. All rights reserved.

3. Run your test script and check the results of the test script using the log created.

   a) Choose *Execute Test Script*. On the following screen, accept the predefined values and choose *Execute*. The system then displays the result log of the test. You can open the tree structure here to display detailed information about the individual steps.

4. Optional: Add an additional eCATT command to your test script so that a user master record is changed on a remote system. Use a system data container and a user on the remote system to do this. Your instructor will give you the details. Save your test script, execute it, and check the result of the execution.

A system data container, in which a target system other than the central system is specified, is required for this optional exercise. The RFC destination specified in the system data container must be maintained appropriately, and the execution of eCATTs must be allowed in the target system (transaction SCC4). There must also be an appropriate model user in the target client that can be changed as required. In the setup for the ADM102 course, there is a system data container, *ZADM102_SDC_DEMO*, that, among other things, references an RFC connection between the DEV system and the QAS system (client 100 in each case) and in the target clients, there is a user, *ECATTCHANGE*, for which the address, for example, can be

Continued on next page

 © 2003 SAP AG. All rights reserved.

changed. If you have performed the exercise for the Configuring eCATT lesson in your course, you can also link to the system data container created there.

---

a) Switch back to the *Attributes* tab page. Enter the name of the system data container that your instructor gives you under *General Data* → *Maintenance System* and save your entry. By double-clicking the name of the system data container, you can check which target system relates to a remote SAP system. Then switch back again and choose the *Editor* tab page.

b) The simplest way to record an SAP transaction in a remote SAP system is to choose the *Pattern* button in the application toolbar. On the following screen, in the *Command* field, choose the entry *TCD (Record)*. In the *Transaction* field, enter the name **`SU01`** for the transaction to be executed and choose *Continue*. The system now proposes an interface. Do not change the default value. In the target system field, use the F4 help to select the remote target system from the system data container. Then confirm your selection by choosing *Continue*. Transaction SU01 then starts. On the initial screen of transaction SU01, enter the user that your instructor specifies and choose *Change*. On the *Address* tab page, you can now make any suitable entry in the *Function* field (such as **`Course participant`**). Then save your data for transaction SU01, choose *Back*, and confirm on the next screen that you want to transfer the data. Save your test script again, and then choose *Execute Test Script*. On the following screen, accept the predefined values and choose *Execute*. The system then displays the result log of the test. You can open the tree structure here to display detailed information about the individual steps. You can, for example, branch to the data entered on the individual screens.

5. Optional: Extend your eCATT test script so that, in your local system, starting from the *SAP Easy Access* menu, the menu path *Tools* → *ABAP Workbench* → *Overview* → *Object Navigator (SE80)* is called, and the program *RSPFPAR* is displayed in this transaction. To do this, you may have to use the SAP GUI installed on your SAP system (instead of the SAP GUI in your training room). In the hierarchy tree of transaction SE80, branch to *RSPFPAR* → *Subroutines* → *READ_ACTUAL_PAR_VALUES* after selecting the program, and display the relevant report by double-clicking the program name. Then complete the recording of the transaction and save the eCATT script. Execute the test script and display the log.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Caution: This optional exercise can only be executed if both the SAP system (parameter *sapgui/user_scripting*) and the local GUI (*Customization of the Local Layout* → *Options...* → *Scripting* → *Enable Scripting*) allow the execution of GUI scripting. If you want to let this exercise be run, ensure beforehand that all prerequisites are fulfilled. If this is not the case, it could be helpful if participants log on at the operating system level of their SAP systems (for example, using the RDP client), and use the SAP GUI installed there. Note, in particular, that - if you have changed the parameter sapgui/user_scripting in transaction RZ11 - you must log off and log on again so that the change takes effect for your user.

a) First use transaction RZ11 to check in another session whether the SAP system allows the execution of scripting commands. To do this, on the initial screen of transaction RZ11, enter the parameter name **`sapgui/user_scripting`**, choose *Display*, and, if necessary, change (using the *Change Value* button) the value of this parameter to *TRUE* (upper case letters are required). Select *Switch on all servers* and choose *Save*. Then log off with your user and log back on again so that the changes take effect.

b) Check whether your SAP GUI supports the execution of GUI scripting. To do this, *Enable Scripting* must be selected under *Customization of the Local Layout*→ *Options...* → *Scripting*.

c) Switch back to the test script editor for your test script. The simplest way to record an SAP transaction that uses controls is to choose the *Record Test Script* button in the application toolbar. On the next screen, ensure that the *Manual Generation* checkbox is not set and that all four checkboxes are selected in the *Automatic Generation* area. Choose *Start Recording*. If necessary, confirm the saplogon message by choosing *OK*. A message now appears in the status bar that you must either open a new session or log on to a remote system using SAP Logon to start the recording. Open a new session from the session toolbar by choosing *Create a new session*. In the *Record SAP GUI Command* window, check the entry in the *Message Server* field (and, if necessary, change it on your own host, twdfxxxx), choose *Yes*, and then switch to your newly created session.

d) In your newly created session, choose *Tools* → *ABAP Workbench* → *Overview* → *Object Navigator* in the *SAP Easy Access* menu and start the Object Navigator by double-clicking. In the Object Navigator, select the *Program* with the name *RSPFPAR* in the

Continued on next page

 © 2003 SAP AG. All rights reserved.

Repository Browser (even if Program or RSPFPAR are already predefined, choose these options again with the *Enter* key). In the folder structure that appears, choose *Subroutines* → *READ_ACTUAL_PAR_VALUES* and display the coding of the subroutine by double-clicking it. Select any text within the coding using the mouse.

e) Now switch to the *Recording running...* screen, select the *Close Recorded GUIs* field, and choose *Stop Recording* there. Save your test script again, and then choose *Execute Test Script*. On the following screen, accept the predefined values and choose *Execute*. If necessary, confirm the message that a script is attempting to connect to your SAP GUI. The test script is then run and the system displays the result log for the test. You can open the tree structure here to display detailed information about the individual steps. You can, for example, branch to the data entered on the individual screens.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Create a simple test script using eCATT and monitor its execution
- Define the implementation areas of various drivers for eCATT

Related Information

For more information about working with eCATT, see the online documentation in the area *eCATT: Extended Computer Aided Test Tool* (BC-TWB-TST-ECA).

© 2003 SAP AG. All rights reserved.

Unit Summary

You should now be able to:

- Outline the uses of eCATT
- List the advantages of eCATT compared to CATT
- List security issues connected to using eCATT
- Make system and client settings related to eCATT
- Create a system data container
- Create a simple test script using eCATT and monitor its execution
- Define the implementation areas of various drivers for eCATT

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

221

Test Your Knowledge

1. Which possibilities does eCATT provide?

   Choose the correct answer(s).

   - ☐ A Performing centrally stored test cases on different target systems
   - ☐ B Testing SAP transactions regardless of whether they use the Control Framework or not
   - ☐ C Testing transactions in the SAP GUI for HTML and testing Business Server Pages
   - ☐ D Testing function modules

2. There is a fixed connection between the test data required for an eCATT test case and the test script, and the data cannot be used without this fixed test script.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

3. Which statements about the relationship between system data containers and test scripts are true?

   Choose the correct answer(s).

   - ☐ A A system data container is always associated with only one test script.
   - ☐ B Multiple test scripts can access the same system data container.
   - ☐ C Each entry in the system data container must also appear in the test script at least once.
   - ☐ D Only systems that are entered in an associated system data container can be accessed from a test script.

4. Which eCATT driver is suitable for recording and running SAP transactions that run in SAP GUI for Windows?

   Choose the correct answer(s).

   - ☐ A SAPGUI
   - ☐ B FUN
   - ☐ C TCD
   - ☐ D WINGUI

© 2003 SAP AG. All rights reserved.

222

Answers

1. Which possibilities does eCATT provide?

   Answer: A, B, C, D

   You can use eCATT to perform cross-system tests. Unlike CATT, you can also test function modules and SAP transactions that use the Control Framework. It is also possible to test applications in the SAP GUI for HTML or BSPs and JSPs using an interface to external test tools.

2. There is a fixed connection between the test data required for an eCATT test case and the test script, and the data cannot be used without this fixed test script.

   Answer: False

   In eCATT, the test data used for the test is stored independently of a fixed test script in test data containers. This allows the same test data to be reused in different test cases with different test scripts.

3. Which statements about the relationship between system data containers and test scripts are true?

   Answer: B, D

   Both test scripts and system data containers are defined in the context of a test configuration. The test configuration links the initially independent objects "system data container" and "test script." This has the advantage that system data containers can be reused; that is, multiple test scripts can access the same system data container. System data containers can contain more system entries than are required in the test script; however, only systems that are defined in the system data containers can be accessed from test scripts.

4. Which eCATT driver is suitable for recording and running SAP transactions that run in SAP GUI for Windows?

   Answer: A, C

   You can use the SAPGUI and TCD drivers for recording and running SAP transactions that run in the SAP GUI for Windows (or in the SAP GUI for Java). If the transaction does not use the Control Framework, we recommend that you use TCD; otherwise, the SAPGUI driver is required. The FUN driver is used for testing function modules. WINGUI is an imaginary name in this context.

© 2003 SAP AG. All rights reserved.

Unit 5

223

Advanced User Administration Topics

A prerequisite for ADM102 is that participants have a sound knowledge of "normal" user administration (transaction SU01) and that basics of the authorization concept (keywords: authorization object, authorization, authorization profile, role). In preceding courses such as SAPTEC or ADM100, the participants learn that there is a central user administration (CUA) and connection to directory services (using the LDAP protocol). In this course, both of these concepts are explained and implemented in the system.

The CUA is already almost a classic feature, but has been significantly corrected and improved in the most recent releases and Support Packages. Assure customers, despite possible skepticism, that the CUA is now very stable. SAP has been involved with directory services for a long time, and user-friendly functions were added with SAP Web AS. However, as this topic is new to many customers, and probably also to instructors, you should prepare especially thoroughly for this topic and ensure that you provide a clear presentation.

Unit Overview

This unit deals with two advanced user administration topics.

In a landscape with multiple SAP systems, Central User Administration (CUA) allows you to maintain user master records centrally in one location and to distribute them to any number of child systems. After an explanation of the concept, you will set up a CUA yourself and get to know various settings.

The CUA is restricted to SAP systems. If user master records are to be synchronously maintained in a heterogeneous landscape, directory services can be useful. They act as a central repository for information (such as user data) that can be accessed from different applications using a standardized protocol. In this unit, you will connect an SAP Web AS to a directory service.

© 2003 SAP AG. All rights reserved.

You can implement each of these concept independently of the other, however they can also be used together.

Unit Objectives

After completing this unit, you will be able to:

- Outline the concept of Central User Administration
- Evaluate the usefulness of Central User Administration for your company
- Define logical systems
- Correctly set up the required users for RFC connections
- Create the RFC connections required for CUA
- Configured the ALE communication required for CUA
- Set up Central User Administration
- Maintain and distribute user data using the CUA
- Perform simple troubleshooting
- Remove a CUA
- Explain the concept of user administration with application to the integration of directory services.
- List the prerequisites for connecting an SAP Web Application Server (SAP Web AS) to a directory server
- Configure the SAP Web AS for connection to a directory server
- Exchange data between the SAP Web AS and a directory server using the LDAP protocol

Unit Contents

Lesson: Introduction to CUA 275
Lesson: Setting Up a CUA.....282
Exercise 10: Setting Up Central User Administration.....295
Lesson: User Administration with the CUA.....303
Exercise 11: Distributing User Data with the CUA 309
Lesson: Introduction to Directory Services.....314
Exercise 12: Technical Prerequisites for LDAP Clients.....323
Lesson: Technical Connection of Directory Services 326
Exercise 13: Connecting an SAP Web AS to a Directory Service ...337
Lesson: Data Exchange with Directory Services.....351
Exercise 14: Data Exchange Between the SAP System and the Directory Service 367

© 2003 SAP AG. All rights reserved.

Lesson: Introduction to CUA

225 Lesson Duration: 25 minutes

Lesson Overview

This lesson introduces the principles of Central User Administration, and provides decision-making aids.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the concept of Central User Administration
- Evaluate the usefulness of Central User Administration for your company

This introductory lesson presents the concept of Central User Administration and the circumstances in which the use of Central User Administration achieves a simplification of administrative activities.

To promote the use of CUA, you must have a clear understanding of the usefulness of CUA yourself. As part of your preparation, you should perform the exercises for CUA in the ADM102 course a number of times, know the documentation, and have read the SAP Notes about CUA listed in the ADM102 course. Basic knowledge about ALE and connecting a directory service with LDAP are also useful. You should ensure that you are familiar with the operation of transaction BD64, at least to the level of knowledge described in the system documentation (http://help.sap.com).

Business Example

In your system group of SAP systems, the same users exist in a number of clients. You want to learn how you can reduce the costs for maintaining users in the different SAP systems. You can reduce the costs for user administration by centrally administering the users in distributed systems.

© 2003 SAP AG. All rights reserved.

Fundamentals

User information is stored in a client-specific manner in SAP systems. That is, every client in a system has its own, independent group of permitted users. If you have a complex system landscape in your company, it is entirely possible that you have more than one of each of three three standard system types (DEV, QAS, and PRD).

Hint: For example, you have three production SAP systems: an SAP R/3 4.6C system for Human Resources, an SAP R/3 Enterprise system for a production ERP system, and a CRM system. These systems would all be regarded as "PRD" systems.

Classically, users are maintained in the respective client. If, for example, a developer retires from your company, you must ensure that the relevant system user is removed from all affected clients. If you do not know exactly which areas your colleague worked in as a developer, you need to log on to every single client that he may have used, and manually search for and remove the user. In complex landscapes, this task can easily become confusing. There may, therefore, be potential security gaps in your user administration. This can result in large costs for consistent user maintenance.

The following figure should clarify the structure of user administration in a "classic" scenario with multiple SAP systems.

Figure 83: Classic User Administration

© 2003 SAP AG. All rights reserved.

As you can see from the figure, the number of clients to be administered is 27 even in this simple example. Note that no clients other than those that are absolutely necessary have been created -- this is not typically the case for real system landscapes.

Concept of Central User Administration

The aim of Central User Administration is to reduce the cost for user administration and to make user administration more secure by centralizing the work. The concept of Central User Administration (CUA) envisages only a single client for maintaining user data; that is, you determine the following list of points at a central location in your system landscape:

- Which user master records exist in which clients?
- Which roles and profiles are assigned to these master records?
- What are the initial passwords?
- Is the user locked or not?

The above user data is managed in only one place. This means that you can have an overview of the situation for managing your user data. Before you decide to implement CUA, the following prerequisites should exist in your system landscape; otherwise the work reduction due to the implementation of CUA will not be optimal.

- There are a large number of identical users in multiple clients
- There is a complex system landscape with large number of clients to be managed
- You want to reduce the costs of complex, distributed user administration by implementing Central User Administration

The prerequisites above are not entirely self-explanatory. If, for example, you have a standard three-system landscape in which the users each primarily work in one client of the system, and the user groups in the different clients are almost disjunctive, the cost saving due to the implementation of CUA is not especially large. The situation is similar if you use identical users in multiple clients, but your system landscape only contains a few clients, or the number of administered users is not especially high in total.

Hint: CUA can also simplify user administration in small system landscapes, as the user master records are then only administered in one client of your system landscape, which is **very clear**. This increases the security of the systems that you administer.

© 2003 SAP AG. All rights reserved.

As the following figure illustrates, you can perform all work relating to user master records from one central client when you implement CUA.

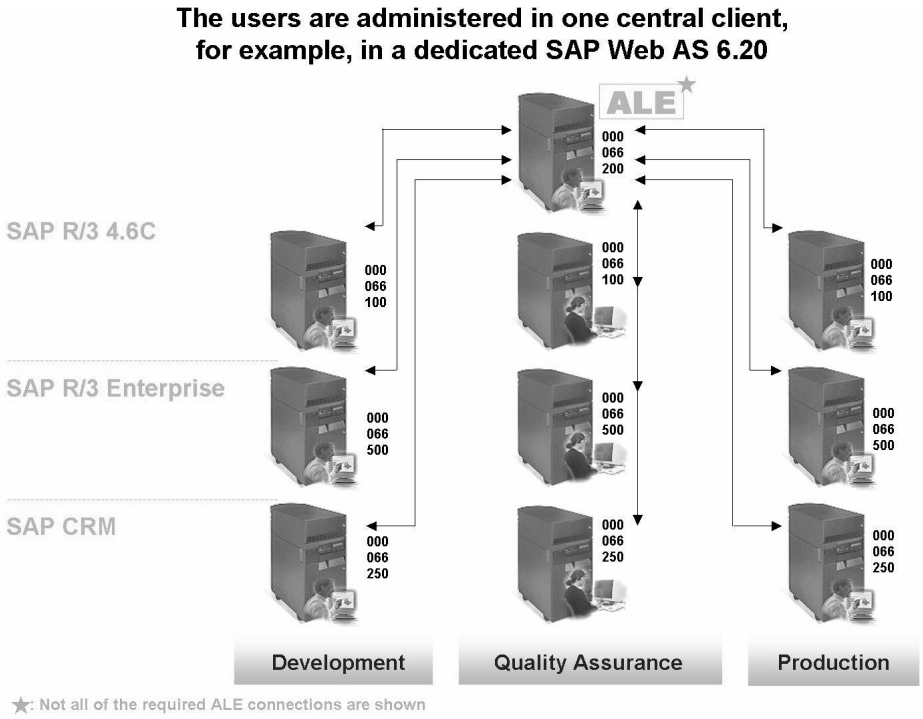


Figure 84: User Administration Using CUA

Using CUA, which is placed in a dedicated system, you can now distribute the following information by Application Link Enabling (ALE) from the central system to the child systems.

Which users exist in which system, in which client, and with which roles and profile assignments. You can also distribute the initial password.

Hint: ALE is a powerful tool, and only a small portion its the many possibilities can be explained in the context of CUA.

The following list contains important notes for implementing CUA:

1. Use a system with a release status of at least SAP Basis 4.6C. Your system should be as current as possible, with current Support Packages and kernel patches.
2. CUA can be used cross-release, although a linked system must have at least SAP Basis 4.5B (and should also have as current a Support Package status as possible).

© 2003 SAP AG. All rights reserved.

3. CUA can, for example, also be used for user administration in BW, CRM, and SRM systems; however, you should search for SAP Notes about this topic in the SAP Notes database before implementing CUA, and take these into account.

In an ALE-integrated system, SAP and non-SAP systems may be in contact. The systems linked to each other are called **logical systems**. The logical systems in the context of CUA are, in this case, the clients involved. That is, a single SAP system can house multiple logical systems in terms of ALE (and CUA). Data exchange in an ALE-integrated system is performed by exchanging **intermediate documents** (IDocs) by **Remote Function Call** (RFC).

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

Clarify the advantages of the CUA

Discussion Questions

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

You can repeat the fundamentals of the CUA at this point, using simple questions. For example, you can explain in a discussion why implementing CUA provides greater benefits (in terms of efficiency) if a large number of identical users are active in multiple clients.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the concept of Central User Administration
- Evaluate the usefulness of Central User Administration for your company

Related Information

- SAP Note 159885: *CUA: Composite Note for Central User Administration*
- Quick Link */security* in the SAP Service Marketplace

© 2003 SAP AG. All rights reserved.

Lesson: Setting Up a CUA

231

Lesson Duration: 85 minutes

Lesson Overview

In this lesson, you will learn about setting up Central User Administration (CUA). All of the required steps are presented in chronological order.

Lesson Objectives

After completing this lesson, you will be able to:

- Define logical systems
- Correctly set up the required users for RFC connections
- Create the RFC connections required for CUA
- Configured the ALE communication required for CUA
- Set up Central User Administration

Setting up CUA has become significantly easier in comparison to previous releases. The old procedure can still be used, but is significantly more complicated than the new method. You should, however, know the old method; the configuration of transactions BD64 and WE20 is described in detail in the SAP Library.

Business Example

The user administration of the SAP systems in your company is to be organized more efficiently and more clearly. CUA is to be implemented to do this.

Central User Administration: Step by Step

The steps listed in the following "roadmap" are required to set up Central User Administration.

Roadmap for Setting Up CUA:

- Specify the logical systems
- Assign logical systems to a client
- Create the required communication users
- Create RFC connections
- Configure and activate the CUA
- Set the parameters for field distribution

© 2003 SAP AG. All rights reserved.

- Synchronize the company addresses
- Synchronize the user groups manually
- Transfer users to central administration

Hint: The following **training setup** is assumed in the rest of this lesson.

Setup for DEV

DEV: Your central system is client 100 in the DEV system, identified as DEVCLNT100.

QAS: The linked child system is client 200 in the QAS system, identified as QASCLNT200.

Setup for QAS

QAS: Your central system is client 100 in the QAS system, identified as QASCLNT100.

DEV: The linked child system is client 200 in the DEV system, identified as DEVCLNT200.

The following figure should also clarify this setup for this course.

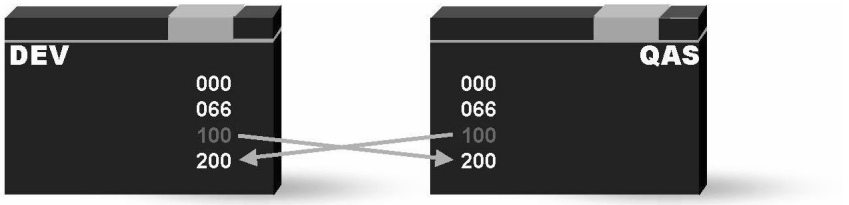


Figure 85: CUA Infrastructure in ADM102

Caution: In a real system landscape, you will only set up one central system.

Note: In the following, only the terms **central system** and **child system** are used. Use the figure above to help you if you are no are longer sure which steps you should perform in which client of which system.

Specifying and Assigning the Logical Systems

You can specify logical systems as follows:

1. Log on to your central system with an administration user.

© 2003 SAP AG. All rights reserved.

2. In the Implementation Guide (transaction SALE) , choose *Sending and Receiving Systems → Logical Systems → Define Logical System* or start transactionBD54. Alternatively, you can maintain the table view *V_TBDLS* using transaction SM30.

3. Choose *Edit → New Entries*.

4. In the *Log.System* column, create a new logical name in **uppercase letters**. So that you can later identify systems by name more easily, use the following naming convention for the logical system names: **<SID>CLNT###**; replace **<SID>** with the system ID in each case, and **###** with the appropriate client number, such as **`DEVCLNT100`**. Enter a name of the logical system, such as **`Central System (CUA)`** or **`Child System (CUA)`**.

5. Save your entries, which will then be included in a transport request.

6. Create the logical system name of the central system and of the relevant child system in all child systems.

You have the following options for step 6:

- Send the transport request generated at step 5, which contains the names of all logical systems, to all other systems in the CUA.
- Perform steps 1 to 4 in all child systems of the CUA. Create the logical system name of the central system and of the relevant child system in each case.

Hint: The data for the logical systems are created cross-client. If there are multiple logical systems of the CUA in one SAP system, you only need to perform steps 1 to 5 (or to import the transport request) once.

You also use transaction SALE to assign the logical systems. Choose *Sending and Receiving Systems → Logical Systems → Assign Client to Logical System* or start transaction SCC4 directly. Alternatively, edit table *T000* with transaction SM30. After you have switched to change mode, you can assign an appropriate entry to the *Logical System* field in the detail view of the client attributes. You can use the F4 help to do this. Select the appropriate entry, such as **`DEVCLNT100`**, if you want to assign this logical system to client 100 in the DEV system. Make this assignment of logical systems for all clients involved in the CUA (in all of the involved systems).

Creating the Communication Users and the RFC Connections

RFC connections are used to exchange data by ALE in the context of CUA. These RFC connections are used to distribute user data from the central system to the child systems, to send changes to this data in the

 © 2003 SAP AG. All rights reserved.

child systems, and to send status reports back to the central system. In the context of CUA, the central system is treated like another child system in some respects. That is, in the simplest case of a CUA (two logical systems linked to each other -- one central system, one child system -- within the same SAP system), two RFC connections are required. If the central system and the child system are in different SAP systems, three RFC connections are required: a separate, additional RFC connection is required to connect the central system to itself. Special users are used for these RFC connections. These RFC users are of the type *Communication* and their authorizations are clearly delimited using predefined roles. For more information, see also SAP Note 492589: *Minimum authorizations for communication users*.

You should understand the need for the RFC connection from the central system to itself. In the scenario for this course: two logical systems in different SAP systems, otherwise unexpected side effects would occur. It is, however, certainly useful if you try out this situation yourself as the trainer, as preparation for this topic.

RFC Connections from the Central System to the Child Systems

RFC connections are required from the central system to all child systems and to the central system itself (loopback). These RFC connections use users in the child systems for which this naming convention should be sufficient: **`CUA_<SID>_###`**, where <SID> is replaced with the system ID of the child system and ### by the client number. This user also requires authorizations that are contained in the following roles: SAP_BC_USR_CUA_SETUP_CLIENT and SAP_BC_USR_CUA_CLIENT. However, you should not assign the delivered SAP roles to the communication users, but rather copies of these roles for which you have created authorization profiles with role maintenance (transaction PFCG).

© 2003 SAP AG. All rights reserved.

For simplicity, simply name the copies of the SAP roles by adding "Z_" to the start of the name, that is, Z_SAP_BC_USR_CUA_SETUP_CLIENT and Z_SAP_BC_USR_CUA_CLIENT.

Hint: Note that the communication user in the child system only requires the role Z_SAP_BC_USR_CUA_SETUP_CLIENT while you are setting up the CUA. You can then remove this role from the communication user again.

Hint: To increase the security and performance in your system, you can use the role Z_SAP_BC_USR_CUA_CLIENT in a "shared" way. This role contains the authorizations required for receiving the IDocs of the CUA and to update them. However, the subrole Z_SAP_BC_USR_CUA_CLIENT_RFC (create this in the same way as the roles described above) contains only authorization to receive the IDocs, and the role Z_SAP_BC_USR_CUA_CLIENT_BATCH has the update authorization for the inbound IDocs. Assign the former role to the communication user, as described above. The role Z_SAP_BC_USR_CUA_CLIENT_BATCH, however, which only allows the changes to the user master records, you should assign to a background user. This user is used to run a periodic background job that implements the CUA changes in the child systems.

The security is increased by sharing the roles in the child system, in that the communication user can now only receive IDocs, and has no additional authorizations. If the communication user also has authorization to update the IDocs, other actions, for which the same authorizations are checked as for updating the IDocs, can also be performed using this user. This problem can be neutralized by splitting the authorizations in the child system as described.

RFC Connections from the Child Systems to the Central System

RFC connections are also required from the child systems to the central system. Every child system must be able to create an RFC connection to the central system.

Hint: As RFC connections can be used cross-client (all clients of a system can use a shared RFC connection), you only need to create one RFC connection to the central system in the case of multiple child systems in one SAP system.

© 2003 SAP AG. All rights reserved.

At this point, the trainer can ask a seemingly easy question: In a CUA scenario, in which three clients of a development system, two clients of a test system, and two clients of a production system are to be administered from another client of the production system, how many RFC connections are required?

Ten RFC connections are required in this situation. Three from the central system to DEV, two to QAS, two to PRD, one from DEV to the central system, one from QAS to the central system, and one from PRD to the central system (which also functions as the "loopback connection").

The connection from the child system to the central system uses a communication user that must be known in the central system. This user should have the following ID: **CUA_<SID>**. In this case, <SID> stands for the system ID of the child system that communicates using this user.

This user requires the authorizations of the following roles: Z_SAP_BC_USR_CUA_SETUP_CENTRAL, Z_SAP_BC_USR_CUA_CENTRAL, and Z_SAP_BC_USR_CUA_CENTRAL_BDIST. Create the roles by copying the template roles (the template names have the same names as the specified roles, but without **Z_** at the start of the name) and generating the profiles for them using transaction PFCG. The first role listed, Z_SAP_BC_USR_CUA_SETUP_CENTRAL, is only required while you are setting up CUA.

Caution: The role Z_SAP_BC_USR_CUA_CENTRAL_BDIST is only required by the communication user if the attributes in transaction SCUL are set to *Redistribution*. See the section "Setting the Parameters for Field Distribution".

Caution: In addition to the RFC connections from the child systems to the central system, you also require an RFC connection from the central system to the central system. This connection also uses the user with the ID **CUA_<SID>**, in this instance, <SID> is the ID of the central system. The required authorizations are contained in the roles Z_SAP_BC_USR_CUA_SETUP_CENTRAL, Z_SAP_BC_USR_CUA_CENTRAL, and Z_SAP_BC_USR_CUA_CENTRAL_BDIST, mentioned above .

The following figure illustrates the required RFC connections and communication users and the roles that they require in a very simple CUA scenario.

© 2003 SAP AG. All rights reserved.

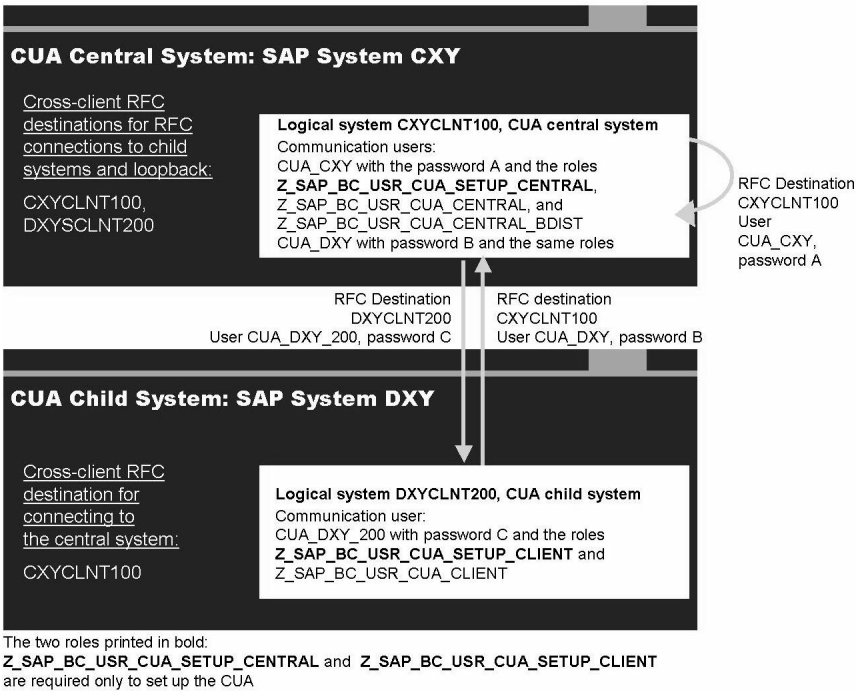


Figure 86: Required RFC Connections and Communication Users for CUA

After you have created the users listed above with the appropriate role assignments in all of the clients involved , you only need to define the RFC connections in transaction SM59. Follow the instructions below to do this:

1. In transaction SALE in the central system, choose *Sending and Receiving Systems* → *Systems in Network* → *Define Target Systems for RFC Calls* (transaction SM59).

2. Choose *Create*.

3. Enter the following data:

   - In the *RFC Destination* field, enter the name of the connection, such as **`QASCLNT200`** or **`DEVCLNT200`**. This entry must be in uppercase letters.
   - In the *Connection Type* field, enter **`3`**
   - In the *Description* group box, enter a short text, such as **`Connection to child system.`**
   - In the *Logon* group box, specify the communication user for the relevant connection, such as **`CUA_QAS_200`** or **`CUA_DEV_200`**.

   Caution: Specify the RFC connection in the same way that you specified the logical system that you want to address with the RFC connection. The RFC connection must be specified in uppercase letters.

© 2003 SAP AG. All rights reserved.

4. After choosing *Return*, make the following settings on the *Technical Settings* tab page:
   - If you want to restrict the function of the CUA to a particular server group, you can specify the logon group after selecting the *Load Distribution: Yes* radio button and choosing *Return*. Only the delivery logon group SPACE exists in the training systems.
   - If you do **not** want to use load distribution, specify the name of the host on which the central instance of your target system runs for *Target Host*, and specify the instance number of the central instance.
5. On the *Logon/Security* tab page, specify the client of the desired logical system and the communication user **`CUA_<SID>_###`**, that is, for example, **`CUA_QAS_200`** or **`CUA_DEV_200`** and its password (in the ADM102 course, this password is **`CUA`**).
6. Save this connection.
7. Test the connection by choosing *Test* → *Connection* and *Test* → *Authorization*.

To setup the required RFC connection from the child system to the central system, log on to the child system and perform steps 1 to 7 again. Use the user **`CUA_<SID>`** here, where <SID> is the SID of your child system. You should also set up the RFC connection from the central system to itself, described above, in the same way.

Hint: In the training setup, RFC connections for use in the exercises have already been prepared. You only need to copy the relevant templates and adjust it a little. That is, you do not need to perform all of the steps listed above.

Note: You can also use trusted RFC connections for the CUA communication. This increases the security of your CUA communication further. For information about configuring the CUA using trusted RFC connections, see the system documentation.

Configuring and Activating the CUA

Once you have successfully activated the CUA, you will no longer be able to create users in the linked child systems using transaction SU01. Before you activate CUA, call transaction SU01 in a child system: It is still possible to create new users.

© 2003 SAP AG. All rights reserved.

The activation of CUA has been significantly simplified, and a number of necessary configuration steps are performed automatically by the system. Follow the procedure below.

1. Log on to the central system.

2. In the Implementation Guide (transaction SALE), choose *Modelling and Implementing Business Processes* → *Predefined ALE Business Processes* → *Cross-Application Business Processes* → *Central User Administration* → *Select Model View for Central Administration* (transaction SCUA).

3. Enter the name of your distribution model, such as **`CUA`**.

4. Choose *Create*.

5. Enter the names of all child systems to be connected (such as **`QASCLNT200`**).

6. Save your entries.

7. The result screen *Display Logs* appears. If you expand the nodes for the individual systems, you normally see the following messages for each system: "ALE distribution model was saved," "Central User Administration activated," and "Text comparison was started." If problem messages are displayed here, follow the procedure in SAP Note 333441: *CUA: Tips for problem analysis*.

Check whether it is still possible to create new users with transaction SU01 in the child system.

Caution: The CUA is only fully operational after you have performed the following additional steps.

Setting the Parameters for Field Distribution

For each field of transaction SU01, you can use transaction SCUM to determine the system in which the administration of the field content can be performed. There are three options:

- The field content is maintained only in the central system ("global")
- The field content can be maintained locally ("local")
- The field content can be maintained locally, and the changed values can be transferred back tot he central system, to then be distributed to all child systems ("Proposal" without redistribution, "Redistribution" with redistribution).

© 2003 SAP AG. All rights reserved.

Caution: If you subsequently change the distribution from "Local" or "Proposal" to "Global" or "Redistribution", inconsistent data can be created. The only exception is that you can reset the indicators on the *Lock* tab page at any time without danger.

Ensure that you take into account SAP Note 611972: *SCUM: Change to field distribution parameters*.

You can make the following settings for each field:

Global

You can only maintain data in the central system. The data is then automatically distributed to the child systems. The corresponding fields cannot be changed there; they can only be displayed.

Local

You can only maintain data in the child system. Changes are not distributed to other systems.

Proposal

You have maintained a default value in the central system that is automatically distributed to the child systems when you create a user. After distribution, the data is only maintained locally, and is no longer distributed, if you change it in the central or child system.

Redistribution

You can maintain the data both centrally and locally. Every time that the data is changed, the change is distributed back to the central system, and is then forwarded from there to the other child systems.

© 2003 SAP AG. All rights reserved.

Everywhere

This option is only available on the *Lock* tab page. You can maintain the lock data both centrally and locally. However, only the changes made to the data in the central system are distributed to the other systems. Local changes in child systems are not distributed.

Hint: The settings of the distribution parameters are automatically forwarded to the child systems.

Hint: Recommendations for how you should configure the parameter distribution for a number of fields are presented in the following table. The recommendation for a "Global" setting applies to all fields that are not listed (with the exception of locks).

Recommendations for Transaction SCUM

Field	Setting
OutputDevice (with German-language logon, as *DEFAULTS_OUTPUT_CONT* under *Defaults*)	Proposal
Parameters	Proposal
Group (under *Logon Data*)	Proposal
Fields for data that the users maintain themselves	Redistribution or local

Caution: For information about the lock management of your users (such as due to too many failed logon attempts), see the documentation.

Synchronizing the Company Addresses

At this point, you should demonstrate synchronizing the company addresses. Explicit differences between the logical systems have been preconfigured, and you can use these in your presentation. This demonstration by the instructor is also included as an optional exercise for this lesson. This means that you should ensure that you are familiar with performing this exercise as part of your preparation. The operation of transactions SCUG and SUCOMP is almost self-explanatory in this context.

© 2003 SAP AG. All rights reserved.

As company address data has already been maintained in all systems of the future CUA, you must first ensure that at least the central system contains all valid company addresses. Then distribute this complete company address set to all child systems, so that a consistent status exists for the company addresses in the entire CUA.

You can use transaction SUCOMP to administer company address data. You can use transaction SCUG in the central system to perform the synchronization activities between the central system and the child systems by selecting your child system on the initial screen of transaction SCUG and then choosing *Synchronize Company Addresses in the Central System*. For more information about this, see the documentation.

Synchronizing User Groups

To be able to transfer users from a child system to the central system, or to distribute them from the central system to a child system, the user group to which the user is assigned must exist in all systems in which the user exists.

Caution: For information about structuring this work as simply as possible, see SAP Note 395841: *CUA: Assign target system-specific parameters and user groups*.

After importing this SAP Note, the missing user groups from the CUA are created where required.

Hint: As of SAP Web AS 6.20, the SAP Note is included in the system; that is, if you use exclusively SAP Web AS 6.20 (or above) based systems, you do not need to take any action.

Transferring Users to Central Administration

After you have synchronized the company addresses, you can transfer the users from the newly connected child systems to central administration. This is done, as with the synchronization of the company addresses, using transaction SCUG in the central system. To do this, on the initial screen of transaction SCUG, select your child system and choose the *Copy Users to the Central System* button.

Different cases are handled differently for the user transfer.

New Users

These users are not yet contained in the Central User Administration. By choosing *Transfer users*, you can transfer the selected users to the central system. All user parameters (address, logon data, and so on) and profiles and roles are transferred. The user is maintained centrally in the future.

© 2003 SAP AG. All rights reserved.

Identical Users

In this case, there are users with identical user IDs (the user ID or ID is the string that you enter in the *User* field on the SAP Logon screen) and matching first and last names. You can transfer the roles and profile data of this user to the central system. The user is then distributed as it exists in the central system. Local data is overwritten. In this case, it is assumed that, for example, the ID "MooreJ" that exists in multiple logical systems, with the last name "Moore" and the first name "Jane," always belongs to one and the same person. In large companies, this assumption is not always correct. For this reason, we recommend that you use unique character strings, such as the personnel number as the user ID.

Different Users

These user IDs exist both in the central and in the child system, but the user has a different first and/or last name. If, in an individual case, these IDs actually refer to the same user, you can transfer the roles and profile data for the user to the central system. The user is then distributed as it exists in the central system. If the IDs do not refer to the same user, use the CUA to create a new user ID in the child system for one of the users and delete the old ID in the child system.

Already Central User

These users already exist in the Central User Administration and are only administered centrally.

Central User Administration is now completely set up for use.

 © 2003 SAP AG. All rights reserved.

243

Exercise 10: Setting Up Central User Administration

Exercise Duration: 30 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Set up a simple Central User Administration

Business Example

You want to familiarize yourself with the functions for setting up CUA before planning a productive implementation.

System Data

System: Two different systems with SAP Web AS 6.20, analogous to the setup for the ADM102 course, in which the systems DEV and QAS are used.
Client: 100 central, 200 child, cross-wise; that is, the groups on the DEV system use client 100 in the DEV system as the central client and client 200 in the QAS system as the child client, and the QAS group use client 100 in the QAS system and client 200 in the DEV system in the same way.
User ID: Users with the required authorizations must exist in every client; in the setup for the ADM102 course, there are users with the name ADM102-### in both clients 100 and 200
Password: Appropriate for the user used. In the ADM102 course, the initial password for user ADM102-## is init.
Set up instructions:

1. All required settings for a smooth processing of the exercises already exist in the system setup for the ADM102 course.

Task

Implement Central User Administration between your central system and the child system that you are using.

If you are working in the DEV system, your central system is client 100 in the DEV system, and the child system is client 200 in the QAS system.

If you are working in the QAS system, your central system is client 100 in the QAS system, and the child system is client 200 in the DEV system.

Continued on next page

 © 2003 SAP AG. All rights reserved.

The user that you are to use is called **ADM102-##** (## corresponds to your group number) with the initial password **init**.

Hint: Note that in the training system, some of the work that you would usually have to do has already been prepared for you.

1. Specify the required logical systems. Use the naming convention presented in the lesson when doing so. Then assign the IDs of the logical systems to the corresponding clients in the two training systems DEV and QAS.
2. Check the role assignments for the required communication users in the central system and in the child system.

   The user in the child system (CUA_QAS_200 or CUA_DEV_200) should have the roles **Z_SAP_BC_USR_CUA_SETUP_CLIENT** and **Z_SAP_BC_USR_CUA_CLIENT**, and the user CUA_QAS or CUA_DEV (in the central system) should have **Z_SAP_BC_USR_CUA_SETUP_CENTRAL**, **Z_SAP_BC_USR_CUA_CENTRAL**, and **Z_SAP_BC_USR_CUA_CENTRAL_BDIST**.

   Caution: The QAS (child) system uses the user CUA_QAS in the DEV system to communicate with the DEV (central) system. In this way, it is easy to assign (using the last changed by information) the changes that the child system confirms to the central system to the child system (QAS in this case).

   Caution: The DEV (child) system uses the user CUA_DEV in the QAS system to communicate with the QAS (central) system. In this way, it is easy to assign (using the last changed by information) the changes that the child system confirms to the central system to the child system (DEV in this case).
3. Create the required RFC connections from the templates using the copy function. To do this, choose an RFC connection by double-clicking it and, on the next screen, choose *Destination* → *Copy*. Specify the correct target host and enter `CUA` for the password of the communication user. The passwords for ALL communication users is `CUA`. Test these RFC connections.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Caution: By assigning a client to a logical system incorrectly, a group (for example, the DEV group) can affect the exercise for both groups (that is, also for the QAS group).

4. Activate the CUA.
5. Specify the parameters for field distribution. Use the recommendations in the lesson as a template. In addition, assign the *Redistribution* setting to the *Telephone* field.
6. Optional: Use transaction SUCOMP to view the company addresses in the central system and in the child system. Now perform a comparison of the company addresses between the central system and the child system. Use transaction SCUG to do this. Proceed step by step here, and always check the success of the last step before starting the next step.
7. Transfer the users from the child system to the central administration. If necessary, decide how individual user master records are to be handled. You should delete a *Different User* from the child system in this example. Then restart transaction SCUG in the central system. Can you still find different users?

Result

Congratulations. You have now set up a simple CUA.

© 2003 SAP AG. All rights reserved.

Solution 10: Setting Up Central User Administration

Task

Implement Central User Administration between your central system and the child system that you are using.

If you are working in the DEV system, your central system is client 100 in the DEV system, and the child system is client 200 in the QAS system.

If you are working in the QAS system, your central system is client 100 in the QAS system, and the child system is client 200 in the DEV system.

The user that you are to use is called **ADM102-##** (## corresponds to your group number) with the initial password **init**.

Hint: Note that in the training system, some of the work that you would usually have to do has already been prepared for you.

1. Specify the required logical systems. Use the naming convention presented in the lesson when doing so. Then assign the IDs of the logical systems to the corresponding clients in the two training systems DEV and QAS.

   a) For step-by-step instructions for the solution to this task, see the subsection "Specifying and Assigning the Logical Systems". Note that you must log on to both systems for this task.

2. Check the role assignments for the required communication users in the central system and in the child system.

   The user in the child system (CUA_QAS_200 or CUA_DEV_200) should have the roles **Z_SAP_BC_USR_CUA_SETUP_CLIENT** and **Z_SAP_BC_USR_CUA_CLIENT**, and the user CUA_QAS or CUA_DEV (in the central system) should have **Z_SAP_BC_USR_CUA_SETUP_CENTRAL**, **Z_SAP_BC_USR_CUA_CENTRAL**, and **Z_SAP_BC_USR_CUA_CENTRAL_BDIST**.

   Caution: The QAS (child) system uses the user CUA_QAS in the DEV system to communicate with the DEV (central) system. In this way, it is easy to assign (using the last changed by information) the changes that the child system confirms to the central system to the child system (QAS in this case).

Continued on next page

 © 2003 SAP AG. All rights reserved.

Caution: The DEV (child) system uses the user CUA_DEV in the QAS system to communicate with the QAS (central) system. In this way, it is easy to assign (using the last changed by information) the changes that the child system confirms to the central system to the child system (DEV in this case).

a) For information about the procedure for creating the required roles and users, see the "Creating the Communication Users and RFC Connections" section.

3. Create the required RFC connections from the templates using the copy function. To do this, choose an RFC connection by double-clicking it and, on the next screen, choose *Destination* → *Copy*. Specify the correct target host and enter **CUA** for the password of the communication user. The passwords for ALL communication users is **CUA**. Test these RFC connections.

Caution: By assigning a client to a logical system incorrectly, a group (for example, the DEV group) can affect the exercise for both groups (that is, also for the QAS group).

---

a) Create an RFC connection from your central system to the child system (QASCLNT200 or DEVCLNT200). Perform a connection test and an authorization check. Ensure that a remote logon is not possible using the newly created RFC connection.
b) Create an RFC connection from the child system to the central system in the same way: DEVCLNT100 or QASCLNT100.
c) In the central system, an RFC connection is also required to the central system itself. Name this connection in accordance with the naming convention (DEVCLNT100 or QASCLNT100).

4. Activate the CUA.

a) Use transaction SCUA to activate the CUA. For information about the procedure, see the subsection "Configuring and Activating the CUA".

5. Specify the parameters for field distribution. Use the recommendations in the lesson as a template. In addition, assign the *Redistribution* setting to the *Telephone* field.

a) Follow the instructions in the subsection "Setting the Field Distribution Parameters".

Continued on next page

 © 2003 SAP AG. All rights reserved.

6. Optional: Use transaction SUCOMP to view the company addresses in the central system and in the child system. Now perform a comparison of the company addresses between the central system and the child system. Use transaction SCUG to do this. Proceed step by step here, and always check the success of the last step before starting the next step.

   a) Call the comparison by selecting your child system on the initial screen of transaction SCUG and choosing the *Synchronize Company Addresses in the Central System* button. You determine that there are different company addresses in the logical systems involved.
   b) Now replicate the company addresses that exist in the central system in the child system.
   c) Replicate the company addresses that only exist in the child system in the central system.
   d) Overwrite the old address in the child system with the current address in the central system.
   e) Use transaction SUCOMP in the child system to check that your actions have been successful.

7. Transfer the users from the child system to the central administration. If necessary, decide how individual user master records are to be handled. You should delete a *Different User* from the child system in this example. Then restart transaction SCUG in the central system. Can you still find different users?

   a) You can transfer the *Identical Users* immediately.
   b) You can also transfer the *New Users* immediately.
   c) In the case of *Different Users*, follow this procedure: Use transaction SU01 to delete the user master record in the child system. Restart transaction SCUG in the central system to check whether this user still exists. For additional information about the procedure, see the lesson "User Administration with the CUA".

Result

Congratulations. You have now set up a simple CUA.

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

If you have enough time, you can explain the security issues for the CUA and the use of trusted RFC and communication users.

Discussion Questions

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

Which advantages do the use of role templates (and their copies) provide in the context of the CUA?

The main advantage here is that in the past, users with *SAP_ALL* authorization were often used for CUA. This is no longer necessary.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Define logical systems
- Correctly set up the required users for RFC connections
- Create the RFC connections required for CUA
- Configured the ALE communication required for CUA
- Set up Central User Administration

Related Information

- SAP Library, area BC-SEC-USR, section *Central User Administration*
- SAP Note 159885: *CUA: Composite Note for Central User Administration*
- SAP Tutorial under http://service.sap.com/security: *Technical Track → Users and Authorizations*

© 2003 SAP AG. All rights reserved.

Lesson: User Administration with the CUA

249 Lesson Duration: 75 minutes

Lesson Overview

In this lesson, you learn about using the CUA using simple examples. You will also learn tips and tricks for secure operation and the steps required to remove Central User Administration.

Lesson Objectives

After completing this lesson, you will be able to:

- Maintain and distribute user data using the CUA
- Perform simple troubleshooting
- Remove a CUA

There are a large number of potential tips and tricks that could be demonstrated for this lesson. However, you should ensure that you do not spend too much time here. The actions described in this lesson can generally also be found in the online documentation easily.

Business Example

You have set up Central User Administration (CUA) in your company and now want to administer the user data for the connected SAP systems. You also want to know about the options for troubleshooting, and to learn the steps required to remove a CUA.

Distributing User Data Using the CUA

With an activated CUA, you continue to use transaction SU01 for user administration. However, there are a number of differences in its operation. The following list should provide you with an overview of the differences in user maintenance when CUA is implemented.

New *Systems* tab page:

This tab page specifies the logical systems in which it is possible for a particular user to log on. However, you only rarely enter values directly on this tab page , because if a system assignment is removed from a user on the *Systems* tab page, the user is deleted from the relevant child system.

© 2003 SAP AG. All rights reserved.

***Roles* tab page:**

On this tab page, you can assign roles to a user in the systems of the CUA. The central system should know all role names of all connected systems. It is always possible to manually assign a role that exists only in a child system, but not in the target system. However, you can only use the F4 help with current data if you have first performed the *Text Comparison from Child Systems* . When setting up the CUA, an initial text comparison is automatically performed. Therefore, F4 help is available immediately. If you want to use F4 help with current data, ensure that you perform a new text comparison between the central system and the child system after creating new roles in the child system.

During the text comparison, specify as the *Target System* the system from which the central information is to query the new information.

Note: If roles were created or changed in a child system, you can choose, in transaction PFCG in the child system, *Environment* → *Text Comparison for CUA Central System*.

***Profiles* tab page:**

The same statements apply to this tab page as to the *Roles* tab page. The *Role Assignment* dialog box also appears from this tab page. However, this dialog box is not displayed when assigning profiles.

The following changes occur for transaction SU01 in the child system.

All tab pages:

On all tab pages of transaction SU01 in the child system, you are now able to administer only the fields that you released for local administration during the configuration of the parameters. Local administration is allowed if you chose an administration setting other than *Global*.

© 2003 SAP AG. All rights reserved.

***Last Changed* field:**

The name of the user that was stored with the RFC connection from the central system to the child system, such as CUA_QAS_200, is specified in this field in the child system for all users that have already been centrally changed.

Hint: There are alternative options for the text comparison described above. You can also start the comparison from the child system by choosing, in transaction PFCG, *Environment → Text Comparison for CUA Central System*, or by calling report **SUSR_ZBV_GET_RECEIVER_PROFILES** in transaction SA38. You do not need to make any entries for this report; the data from the child system is sent to the central system anyway.

To force the text comparison, you should also run report **SUSR_ZBV_GET_RECEIVER_PROFILES** in transaction SA38. In this case, however, you should explicitly specify the child systems with which you want to perform a text comparison.

You can automate the comparison by scheduling the report to be regularly executed in the background on the central system. In this case, select a variant that contains all of your child systems.

Fields whose field distribution parameters are set to *Redistribution* send their content back to the central system, from which this field value is then forwarded to all child systems. Existing entries are updated in this way.

For more information about optimizing the performance of the data distribution of your CUA landscape, see the documentation in the area *BC-SEC-USR → Central User Administration → Activated Background Processing*. To do this, you should have at least basic experience using ALE. Note that the settings required for this are not in any way directly related to the CUA, but are stored deep in ALE technology.

Hint: You can use transaction SCUL to distribute individual user master records to the child systems again.

If you want to redistribute a large number of users simultaneously, transaction SCUL is not a good option. Use report **RSCCUSND** to do this, as described in SAP Note 503247.

To activate the background processing, and to use it optimally, it is necessary to make settings in transaction WE20, and to be familiar with and/or able to operate reports RSEOUT00, RBDMOIND, RSARFC01, and RBDAPP01, the SAP Notes 371068 and 399271, and transaction SM58 (for

© 2003 SAP AG. All rights reserved.

completely optimized operation - a simplified scenario is described in the optional exercise). A more detailed description of these options would be beyond the scope of this lesson. A short summary of the options is given in the documentation area referred to above.

Troubleshooting for the CUA

If the distribution of user data is not functioning, follow the checklist below:

1. View the distribution log in transaction SU01: *Environment* → *Distribution log*. Alternatively, call transaction SCUL . If *Errors* are displayed here, you can choose the *Resend User* button as a first step. Trace whether the problem persists or whether the data is sent successfully this time.
2. If the display of the distribution log informs you of *tRFC with Errors*, you should check the RFC connections used in transaction SM59.
3. In transaction SCUL, choose *Environment* → *ALE Distribution Model* (transaction BD64) to check the consistency of the ALE model that you are using. To do this, select the model for your CUA and check whether the required BAPIs *USER.Clone* and *UserCompany.Clone* are assigned to your model.
4. Check whether your CUA distribution model is displayed in transaction SCUA.
5. For more advanced analysis, use the tips in SAP Note 333441: *CUA: Tips for problem analysis*.

Caution: Ensure that you read the important SAP Notes, which are listed in the composite SAP Note 159885: *CUA: Composite note for Central User Administration*.

Removing the CUA

It is important to differentiate between two possible scenarios for removing the CUA.

1. A child system is to be removed from the CUA.
2. The CUA is to be completely removed.

These two scenarios are presented in the following. **Steps to remove a child system from the CUA:**

1. Use the *Delete* button in transaction SCUA or call report **RSDELCUA** directly in transaction SA38.

© 2003 SAP AG. All rights reserved.

2. In the *Delete* area, choose the option *Child System*, and then select the child system to be removed using the F4 help.
3. Choose *Test* and *Execute*.
4. On the following screen, the system lists the data to be deleted and the tables affected. By double-clicking a table name, you jump to the display of the table content (as it is displayed using transaction code SE16).
5. If you are sure that you want to delete the displayed data, choose *Back* (F3) and deselect the *Test* option. Choose *Execute*.
6. In transaction WE20 , under *Partner Profiles* for *Partner Type LS*, delete the message types CCLONE and USERCLONE for the child system.
7. Start transaction BD64, and switch to change mode.
8. Delete the methods of the child system and save your entries.
9. To distribute the distribution model of your CUA, choose *Edit* → *Model View* → *Distribute* and select the child system that you have just deleted in the dialog window.
10. Use transaction SCUA to check whether the child system was deleted from the CUA.
11. If there are no additional child systems of the CUA in the SAP system that contains the child system to be deleted from the CUA, change the communication user (such as CUA_QAS) in transaction SU01 by removing the role **Z_SAP_BC_USR_CUA_CLIENT** on the *Roles* tab page. If there are no longer any roles or profiles assigned to the user, you can delete it.

Hint: As the RFC destinations that you created may also be used for ALE connections other than the CUA, you should not delete the RFC destinations, but rather remove the roles with CUA authorizations from the communication user used.

In the child system, you must perform the following step:

1. In transaction WE20, delete the data about the central system.

You have now completed the steps required to remove a child system from the CUA.

If you want to delete the CUA completely, you must, in principle, perform the steps to set up the CUA again, but in reverse order. The steps for transactions SCUA, SCUM, and SCUG can be reversed using report **RSDELCUA**.

© 2003 SAP AG. All rights reserved.

Steps to delete the entire CUA:

1. Start report **RSDELCUA** on the central system with the options *Complete CUA* and *Test*.
2. If you are satisfied with the list of data to be deleted, run the report again, but this time without the *Test* option.
3. In the central system, delete the data about the central system and the child system(s) in transaction WE20; that is, delete the message types CCLONE and USERCLONE under *Partner Profile for the Partner Type LS*. If only the standard message type SYNCH then remains, you can delete the partner profiles completely.
4. In transaction BD64, delete the distribution model of the CUA.
5. Repeat steps 3 and 4 in the child systems.
6. Remove the CUA roles (such as Z_SAP_BC_USR_CUA_CENTRAL and Z_SAP_BC_USR_CUA_CLIENT from all communication users (such as CUA_QAS_200 or CUA_DEV).

Caution: Delete the RFC connections of the CUA only if you are certain that these connections are not used in other ALE scenarios.

Hint: The users that you have copied from the child systems to central administration are *locked* in the former central system after you have removed the CUA. These users have no authorizations or roles (unless they were assigned roles and profiles in the central system) and can be deleted manually.

© 2003 SAP AG. All rights reserved.

255

Exercise 11: Distributing User Data with the CUA

Exercise Duration: 15 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Outline the use of the CUA with simple examples

Business Example

You want to use a CUA.

Task 1

Create a new user using the CUA.

1. Create the user ADM102-25 for the child system. Assign the profile SAP_ALL to this new user. Assign the parameter default *XUS* with the value **ADM102-25** to the user. Save your entries.
2. In the child system, display the user ADM102-25. Which fields are ready for input? Change the parameter *XUS* for this user to the value **Analyze**. Assign a telephone number to this user.
3. In the central system, use transaction SU01 to check the fields *Last Change* and *Parameters* for the user ADM102-25, and check the telephone number of this user.

Task 2

Optional (for experts): Set up the processing of inbound IDocs in the background in the child system, as described in SAP Note 399271 and the SAP Library.

1. Use the CUA to create a new user in the child system: **CUA_BACK**. Assign the required role, Z_SAP_BC_USR_CUA_CLIENT_BATCH, to this user (the role has been generated for you), and assign the prepared role Z_SAP_BC_USR_CUA_CLIENT_RFC to the communication user **CUA_<SID>_###** in the child system.

   Then remove the roles that the communication user **CUA_<SID>_###** no longer requires.
2. In transaction WE20, schedule the execution of the IDocs CCLONE and USERCLONE from the central system in the background. Use the prepared variant **CUA** for report **RBDAPP01** (created in accordance

Continued on next page

 © 2003 SAP AG. All rights reserved.

with SAP Note 399271). Use transaction SM36 to create a job for the user **CUA_BACK** (as the step executing user) that executes this report with the variant **CUA** every five minutes. Use a test user to check that the background processing is operative.

Task 3

Optional (for experts): Remove the CUA.

1. Follow the instructions in the training material to remove the CUA.
2. Check that transaction SU01 is fully functional in the child system. You should also check the transactions SCUA, WE20, and BD64.

© 2003 SAP AG. All rights reserved.

Solution 11: Distributing User Data with the CUA

Task 1

Create a new user using the CUA.

1. Create the user ADM102-25 for the child system. Assign the profile SAP_ALL to this new user. Assign the parameter default *XUS* with the value **ADM102-25** to the user. Save your entries.

   a) Use transaction SU01 in the central system. You determine that user ADM102-25 already exists, but not in the child system. Switch to the *Profiles* tab page and start the text comparison (as a precaution). Then assign the *SAP_ALL* profile to the user in the child system. Before you save, assign the parameter *XUS* with the value `ADM102-25`.

2. In the child system, display the user ADM102-25. Which fields are ready for input? Change the parameter *XUS* for this user to the value **Analyze**. Assign a telephone number to this user.

   a) Use transaction SU01 in the child system. View all tab pages and note the fields that are ready for input. Assign a telephone number to the user and reassign the parameter *XUS*.

3. In the central system, use transaction SU01 to check the fields *Last Change* and *Parameters* for the user ADM102-25, and check the telephone number of this user.

   a) Follow the exercise text.

Task 2

Optional (for experts): Set up the processing of inbound IDocs in the background in the child system, as described in SAP Note 399271 and the SAP Library.

1. Use the CUA to create a new user in the child system: **CUA_BACK**. Assign the required role, Z_SAP_BC_USR_CUA_CLIENT_BATCH, to this user (the role has been generated for you), and assign the prepared role Z_SAP_BC_USR_CUA_CLIENT_RFC to the communication user **CUA_<SID>_###** in the child system.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Then remove the roles that the communication user **CUA_<SID>_###** no longer requires.

a) Use the roles Z_SAP_BC_USR_CUA_CLIENT_RFC and Z_SAP_BC_USR_CUA_CLIENT_BATCH (the template roles have identical names, but without the **Z_** at the start of the name). Now assign the first of the listed roles to the **CUA_<SID>_###** user. Remove the two roles previously assigned to the user. Assign the second of the roles listed above to the newly created user **CUA_BACK**.

2. In transaction WE20, schedule the execution of the IDocs CCLONE and USERCLONE from the central system in the background. Use the prepared variant **CUA** for report **RBDAPP01** (created in accordance with SAP Note 399271). Use transaction SM36 to create a job for the user **CUA_BACK** (as the step executing user) that executes this report with the variant **CUA** every five minutes. Use a test user to check that the background processing is operative.

   a) For the solution, see SAP Note 399271, the SAP Library, and the exercise text.

Task 3

Optional (for experts): Remove the CUA.

1. Follow the instructions in the training material to remove the CUA.

   a) For the solution, see the exercise text, and the content of the third section of this lesson (*Removing the CUA*).

2. Check that transaction SU01 is fully functional in the child system. You should also check the transactions SCUA, WE20, and BD64.

   a) Follow the descriptions in the training material in the *Removing the CUA* section of the lesson.

 © 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Maintain and distribute user data using the CUA
- Perform simple troubleshooting
- Remove a CUA

Related Information

- SAP course BIT300: *Application Link Enabling (ALE) Technology*
- SAP Library, area BC-SEC-USR, section *Central User Administration*
- SAP Note 333441: *CUA: Tips for problem analysis*

© 2003 SAP AG. All rights reserved.

Lesson: Introduction to Directory Services

260 Lesson Duration: 20 minutes

Lesson Overview

This lesson first defines what is meant by a directory service. A scenario is discussed as an example.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain the concept of user administration with application to the integration of directory services.

From experience, many participants in the ADM102 course have little knowledge about directory services. You should therefore ensure that you explain the reasons for this topic clearly. Directory services are playing an ever greater role; not only in the SAP environment, but in general. SAP is well-positioned with SAP Web AS – and we should not hide this fact!

A directory service is already running at many customer sites, without them realizing this (for example, Active Directory is a fixed component of every Microsoft Windows 2000 Primary Domain Controller).

Business Example

Your company uses a directory service to store personnel data centrally, among other things. As a member of the user administration team for SAP systems, you want to obtain an overview of directory services, and find out about how you can connect your SAP system to these directory services.

What Are Directory Services?

Not only in the Internet, but also in company networks, there is a danger that information will be lost because it is made available in an unstructured way, becomes obsolete, or no-one knows that it exists. A modern information system is needed here.

 © 2003 SAP AG. All rights reserved.

Directory services act as a central repository for data that is used by various applications.

Typically, information about users, documents, or hardware resources is stored – however, any other objects could also be stored.

The directory service is accessed using a standardized protocol, such as LDAP (Lightweight Directory Access Protocol) or X.500 DAP (Directory Access Protocol) or DSML (Directory Services Markup Language)

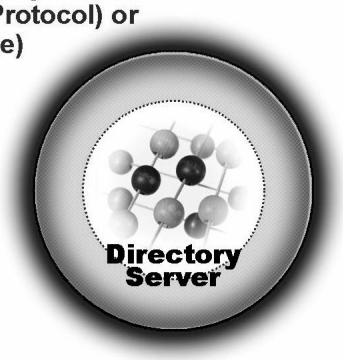


Figure 87: Properties of directory services

A directory server is an information system for particular information. Directory services can help us to answer questions such as "What is the project leader's e-mail address?" or "Where is the nearest color printer?" in the same way as yellow pages help us to find a lawyer or the nearest florist.

Unlike a relational database management system (RDBMS), where information is stored in two-dimensional tables, directory services use a hierarchical (tree) structure to organize the data. Another difference is the relationship of write to read accesses. Directory services are designed for a high read load, but unlike relational databases, they do not support transaction or lock concepts, or costly queries such as joins.

Directory services are typically used to store information about users, documents, or hardware resources – however, any other objects could also be stored.

Example Scenario

This example is, of course, an ideal scenario. It is, however, also realistic (companies implement at least part of it) and it makes the reasons why directory services are becoming so important clear to customers.

One comment on the SAP Enterprise Portal: SAP Enterprise Portal 5.does not work without a central Directory Server. Some customers are pleased that their corporate directory server can be connected – while for others, the introduction of the portal causes a "technology push" in their

© 2003 SAP AG. All rights reserved.

company. SAP delivers SAP Enterprise Portal 5.0 with a license for the Novell eDirectory product (free of charge); however, other products can, of course, be used. With SAP Enterprise Portal 6.0, the customer can choose an SAP system, the portal database, or a directory server as the storage location for the user data ("User Persistence Store").

Directory services can fully demonstrate their strengths in heterogeneous system landscapes (where products from different vendors are used). This example makes this clear.

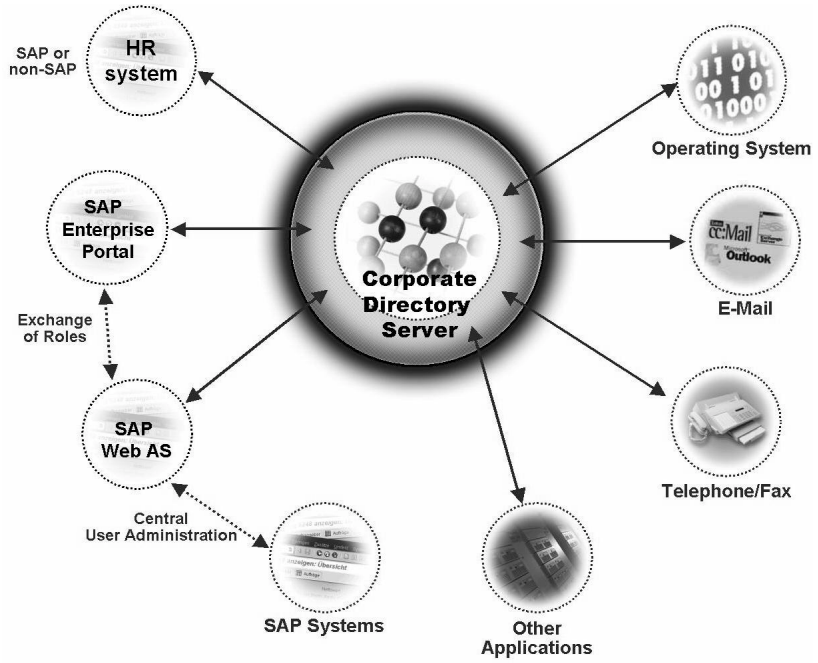


Figure 88: Example scenario with a directory service

A company employs a new employee. This employee is first created in the HR system (which can be an SAP system or come from a different vendor). Some properties (such as name, personnel number, telephone number, e-mail address, department, and group assignment) are transferred to the corporate directory server, while others (such as date of employment, manager, and salary) are deliberately held only in the HR system. By connecting various applications to the directory server:

- The new employee can immediately log on to the corporate domain at operating system level
- The employee has a functioning e-mail account immediately
- Telephone and fax numbers are assigned to the user
- The employee can log on to the enterprise portal

© 2003 SAP AG. All rights reserved.

- The user has access to certain SAP systems (such as an SAP EBP, to place orders)
- Other external systems can be easily integrated

If an employee leaves the company, he or she is deleted in the HR system and therefore also from the directory. Consequently, all of his or her accounts are deleted or locked – no longer an easy task, if it must be performed manually. This closes gaps in security due to accounts that have not been deleted.

What Is LDAP?

The abbreviation LDAP stands for **Lightweight Directory Access Protocol** and describes how operations are to be formulated for a directory service. The most common operation is querying entries, however it is also possible to create, change, or delete entries.

From a technical point of view, LDAP follows the client/server model: one or more servers hold the information, which the client(s) access. Originally, LDAP implemented only communication with an X.500 server, as described by the International Organization for Standardization (ISO).

LDAPv2 was implemented in 1995 by Yeong, Howes, and Kille at the University of Michigan, to "minimize the complexity of clients so as to facilitate widespread deployment of applications capable of utilizing the Directory". LDAP uses a TCP/IP stack and simplifies access to X.500 servers. The current version of LDAP is LDAPv3. For more information about the technical details, see RFC 2251 (RFC means "Request for Comments" here, for more information about this, see www.ietf.org).

An X.500 server is no longer required; various vendors offer "standalone" servers that can be addressed using LDAP. In addition to commercial products (such as eDirectory from Novell, SunOne Directory Server from Sun/Netscape, and Active Directory von Microsoft), there is also an Open Source development: for more information, see www.openldap.org.

The Data Model

There are a number of models for LDAP. In addition to the access protocol, these define how information is stored and identified, and how access is protected.

A directory contains **entries** that consist of one or more **attributes**. An attribute, in turn, consists of a **type** (frequently abbreviated strings such as "cn" for "common name, ", "c" for "country", or "mail" for e-mail address) and one or more **values**.

 © 2003 SAP AG. All rights reserved.

Certain attributes are required, while others are optional – this is controlled with LDAP using a special attribute, **object class**. Object classes have an inheritance hierarchy. The information about object classes and attribute types is stored in **schemas**, which are provided by Internet organizations and vendors of directory servers (and which can be customized by the user).

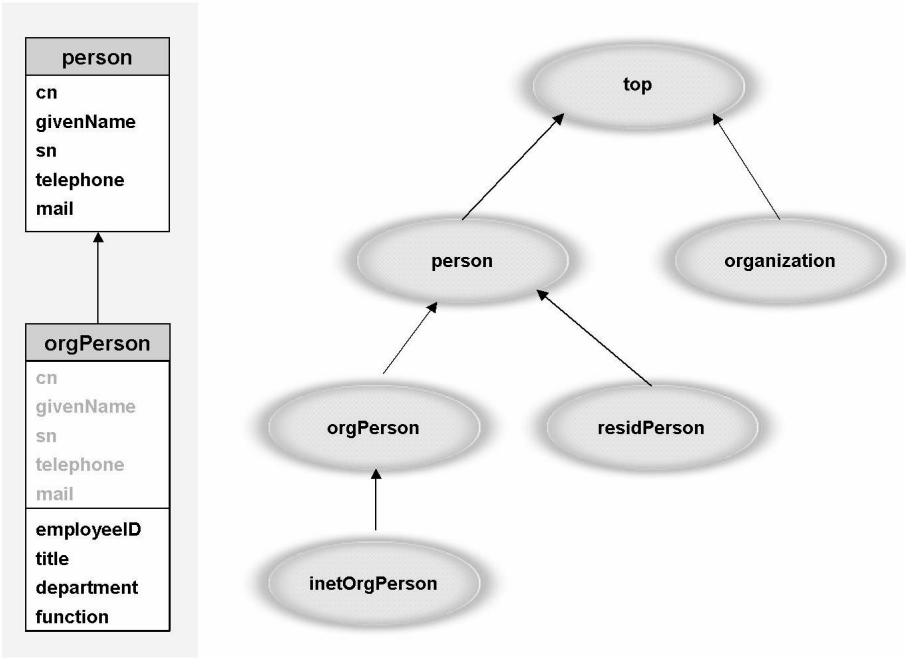


Figure 89: Hierarchy of object classes

cn	LHeepmann	naming attribute (RDN)
objectclass	inetOrgPerson	special attribute
givenName	Linus	attribute with only one value
sn	Heepmann	required attribute
telephoneNumber	+49-6227 7-47474	optional attribute
mail	linus.heepmann@sap.com	

Figure 90: Entry in a directory

© 2003 SAP AG. All rights reserved.

The entries are also organized in a hierarchical tree structure. Each entry contains a name that is unique directory-wide, the **Distinguished Name (DN)**. The DN is created by describing the path from the entry to the root of the tree (in the form of a comma separated list) (For example: cn=LHeepmann, ou=instructors, o=sap, c=de). An individual component of the DN is called the **Relative Distinguished Name (RDN)**, as this always identifies the node uniquely relative to the superordinate node in the tree. Two nodes under the same superordinate node could not, therefore, have the same RDN. The entire tree of entries (which can be physically stored across multiple hosts) is called the **Directory Information Tree (DIT)**.

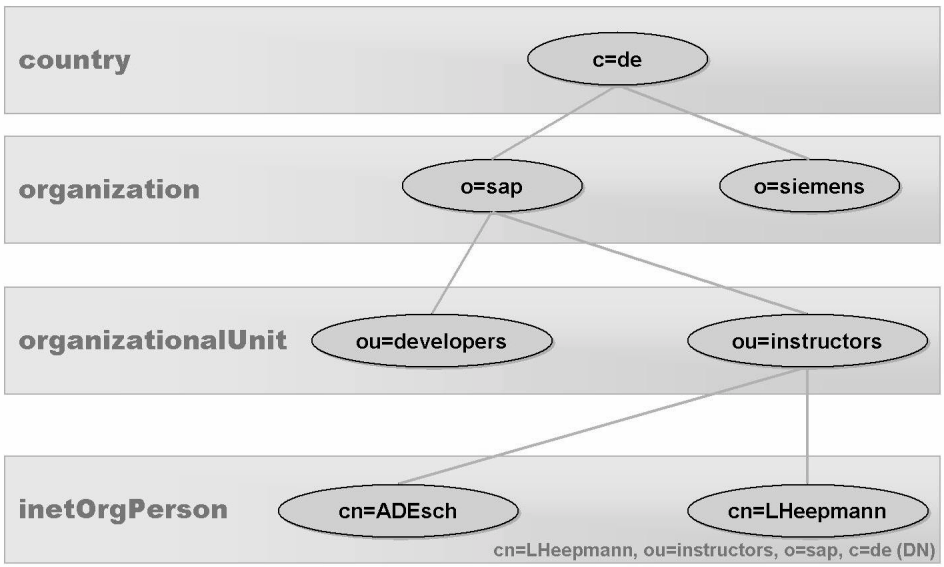


Figure 91: Directory Information Tree (DIT)

© 2003 SAP AG. All rights reserved.

SAP and Directory Services

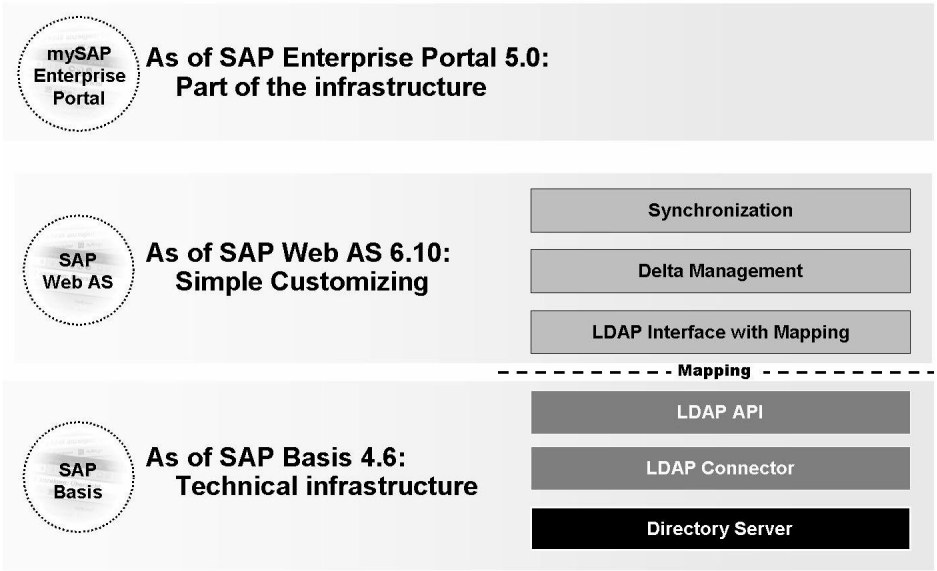


Figure 92: SAP and directory services

SAP has already provided ways to connect SAP systems to directory services for a number of years. In 1997, the LDAP Gateway was delivered in Java with SAP Basis 4.0A, a program that runs independently of the application server (supported up to SAP Basis 4.6B). Integration into the application server came with the LDAP Connector, which has been available since SAP Basis 4.6A. With SAP Web AS 6.10, user-friendly functions for user master synchronization (mapping, delta management, synchronization) were added.

On August 2, 2001, SAP Web AS 6.10 was awarded the "Works with LDAP" certificate by the Open Group (see http://www.opengroup.org/dif/wwl-dap2000/prod_disp.tpl?CALLER=rcp.tpl&id=38). SAP has also certified partners for the directory interface (technical name: BC-LDAP-USR) itself (see Quick Link */partnerdirectory* on the SAP Service Marketplace).

Show the list of certified vendors (*Search by ... Software Category → Directory Services*). Here is an alternative URL (without guarantee that it will always work): http://www.sap.com/global/scripts/softwarepartnerdir/products/certify/direct_intfce.asp?DescID=224&ProdDesc=Directory%20Services.

SAP's enterprise portals also require a store for user data such as master data and group assignment ("User Persistence Store"). SAP Enterprise Portal 5.0 (first delivered at the end of 2001) uses one ore more directory

© 2003 SAP AG. All rights reserved.

servers for this purpose, which users are checked against when they log on to the portal. With SAP Enterprise Portal 6.0 (first delivered at the end of 2002), additional storage options were added to the directory server (SAP system or a portal database).

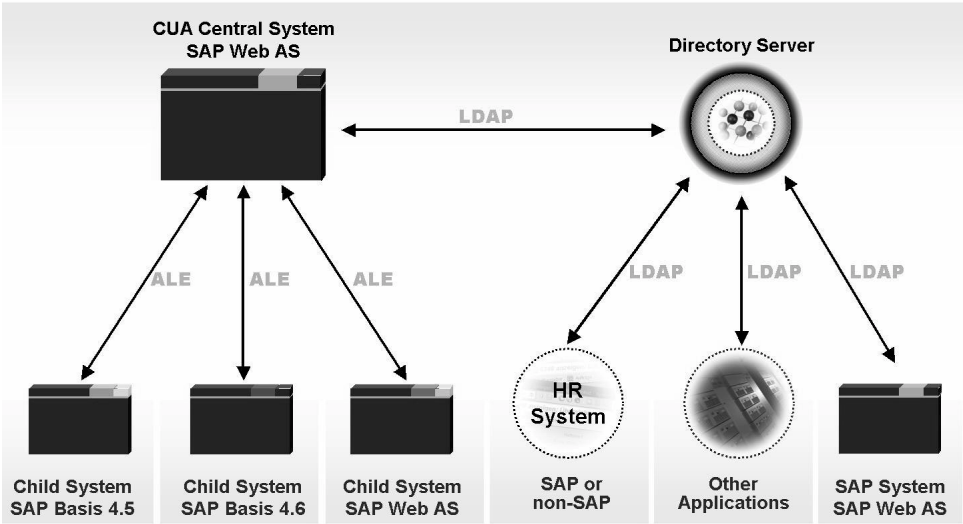


Figure 93: CUA and directory services

Both Central User Administration (CUA) and directory services allow you to maintain user data at one location, and to synchronize it in multiple systems. Both concepts can be implemented independently. However, you can also connected them with each other.

In this way, you could use CUA to connect SAP systems with SAP Basis 4.5 or 4.6, while other SAP systems with SAP Web AS could also be connected directly to the directory service.

In a scenario of this type, LDAP synchronization functions as a “remote control” for the CUA, meaning that all prerequisites and restrictions of the CUA continue to apply.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

269

Exercise 12: Technical Prerequisites for LDAP Clients

Exercise Duration: 5 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- State which program libraries are required at operating system level to be able to use LDAP clients in general

Business Example

As an administrator, it is your task to connect an SAP system to a directory server. You require information about whether there are particular prerequisites for the operating system of the SAP system.

Task 1 Program Libraries for LDAP Clients

Depending on your operating system, program libraries for LDAP clients may already be part of the delivered system, or you may have to purchase them separately. A library of this type is also required if you are using an SAP system as an LDAP client.

1. Use the Quick Link */security* in the SAP Service Marketplace to determine which program libraries are required to use the SAP Web AS as an LDAP client.

Task 2 (Optional): LDAP Connector Version

The backward compatible LDAP Connector is made available through the SAP Service Marketplace.

1. What is the current version of the LDAP Connector for SAP Web AS 6.20?

© 2003 SAP AG. All rights reserved.

Solution 12: Technical Prerequisites for LDAP Clients

Task 1 Program Libraries for LDAP Clients

Depending on your operating system, program libraries for LDAP clients may already be part of the delivered system, or you may have to purchase them separately. A library of this type is also required if you are using an SAP system as an LDAP client.

1. Use the Quick Link */security* in the SAP Service Marketplace to determine which program libraries are required to use the SAP Web AS as an LDAP client.

   a) Log on to the SAP Service Marketplace, http://service.sap.com/security, with your user.

   You can find the document *LDAP Client Libraries* under the path *Security in Detail → Identity Management → Directory Service Integration*. This document specifies which program libraries are required for which operating systems.

Task 2 (Optional): LDAP Connector Version

The backward compatible LDAP Connector is made available through the SAP Service Marketplace.

1. What is the current version of the LDAP Connector for SAP Web AS 6.20?

   a) Use the SAP Notes Search (Quick Link */notes* in the SAP Service Marketplace) with the search term **`LDAP Connector Version`** and find the SAP Notes with the highest version specified in the SAP Notes title.

   At the time of creating this training material, version 2.6 was the current version (see SAP Note 539198).

   For corrections (patches) for the version delivered on CD, see the Quick Link */patches* under the path *SAP Web AS → SAP Web AS 6.20 → Binary Patches → SAP KERNEL <Version> → <operating system> → Database independent*.

 © 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Explain the concept of user administration with application to the integration of directory services.

Related Information

- Quick Link */security* → *Security in Detail* → *Identity Management* → *Directory Service Integration* on the SAP Service Marketplace
- Quick Link */partnerdirectory* → *Search by...* → *Software Category* → *Directory Services* on the SAP Service Marketplace
- Request for Comments (RFCs), see also www.ietf.org and www.rfc-editor.org
  - RFC 1274: COSINE and Internet X.500 Schema (www.ietf.org/rfc/rfc1274.txt)
  - RFC 2251: Lightweight Directory Access Protocol (v3) (www.ietf.org/rfc/rfc2251.txt)
  - RFC 2256: A Summary of the X.500 (96) User Schema for Use with LDAPv3 (www.ietf.org/rfc/rfc2256.txt)
  - RFC 2798: Definition of the inetOrgPerson LDAP Object Class (www.ietf.org/rfc/rfc2798.txt)
- www.openldap.org (Open Source LDAP implementation)

© 2003 SAP AG. All rights reserved.

Lesson: Technical Connection of Directory Services

272 Lesson Duration: 75 minutes

Lesson Overview

SAP Web AS allows you to easily connect an SAP system to a directory service using the LDAP Connector. This lesson explains how this connection is technically implemented.

Lesson Objectives

After completing this lesson, you will be able to:

- List the prerequisites for connecting an SAP Web Application Server (SAP Web AS) to a directory server
- Configure the SAP Web AS for connection to a directory server

---

Caution: In the ADM102 course, you (the trainer) install an external directory service for the entire course, in which you then create a Directory Server, to which you connect "your" SAP system. The participants also connect "their" SAP systems later, in the exercises.

We recommend that you install the **Directory Service** before the course. The process takes some time, requires a restart of the entire server at the end of the process, and has no useful content for the participants. And if a server hangs during a restart...

On the other hand, you should demonstrate the work with the iPlanet Console to the participants. The reason for this is not to promote or to explain the product used in the course to the participants (the customers may use directory services from other vendors), but rather to demonstrate the significant points (such as protocol, port, and structure). These details are ultimately required in the SAP system and should therefore not "appear from nowhere".

The following description sets up the environment for an exercise group 99 (so that you can demonstrate this again later in the course in the same way as the exercises with group number 00). When you demonstrate the installation steps to the participants, you can also use 00 immediately instead of 99 (note, however, that certain points in the exercises will then run differently from your demonstration). Background: During the installation of the iPlanet Server, the iPlanet Console and a Directory Server are installed - steps that the participants are to perform as extra points.

 © 2003 SAP AG. All rights reserved.

You should therefore avoid becoming bogged down in the details of the directory service, but demonstrate the points mentioned.

Note: For this course, we use the **iPlanet Server** product (now known as the Sun ONE Directory Server) from the Sun/Netscape Alliance. For more information about the product, see http://docs.sun.com/db/prod/sunone. You can, of course, also give this address to the participants.

To set up the iPlanet Directory Server:

1. Use **pcAnywhere** to log on at operating system of "your" server.

   Caution: The installation fails with the Terminal Server Client (RDP Client) (the system displays a message saying that there are "unknown errors" at the end).

2. Stop the SNMP Service (you can do this by choosing *Start* → *Programs* → *Administrative Tools* → *Services* → *SNMP Service*). Various Compaq agents are stopped at the same time.

   Note: After the restart of the server later, all of these services run again.

3. Open Windows Explorer, navigate to **`G:\Additional_Files\ADM102\Directory Server\iPlanet 5.0`** and start the **`setup.exe`** program.

4. Move through the installation by choosing *Next* repeatedly. Important points (see also the screenshots) in the installation are:

a) Installation type *iPlanet Servers* and *Typical*
b) Target directory **`G:\iPlanet\Servers`**
c) All suggested components
d) *This instance will be the configuration directory server*
e) *Store data in this directory server*
f) *Server identifier* **`Directory99`**, *Server port* **`38999`**, *Suffix* **`o=company99, c=de`**.
g) User **`admin`**, enter the password **`SAPNetWeaver`** twice
h) *Administration Domain* **`wdf.sap.corp`**
i) *Directory Manager DN* **`cn=Directory Manager`**, enter the (suggested) password **`Dir99Manager`** twice
j) *Administration Port* **`8888`**
k) *Install*

5. After around two minutes, the installation has been performed. You must then reboot the entire server. It will be around three minutes before you can log on to the server again.

© 2003 SAP AG. All rights reserved.

Caution: As of this point, you must use the Terminal Server Client (RDP Client), as there are problems with the screen structure when the iPlanet Console and pcAnywhere are connected.

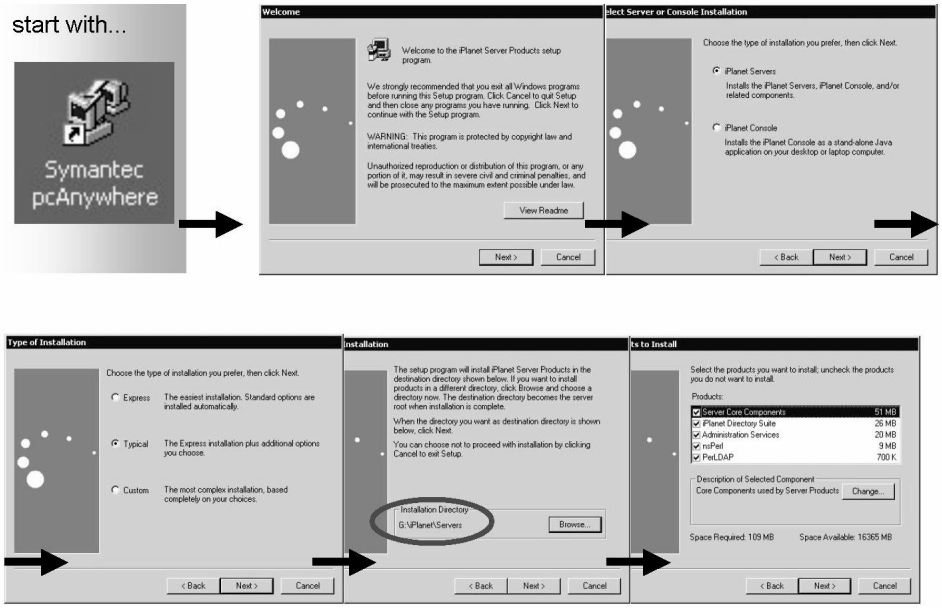


Figure 94: Installing the iPlanet Server 1/3

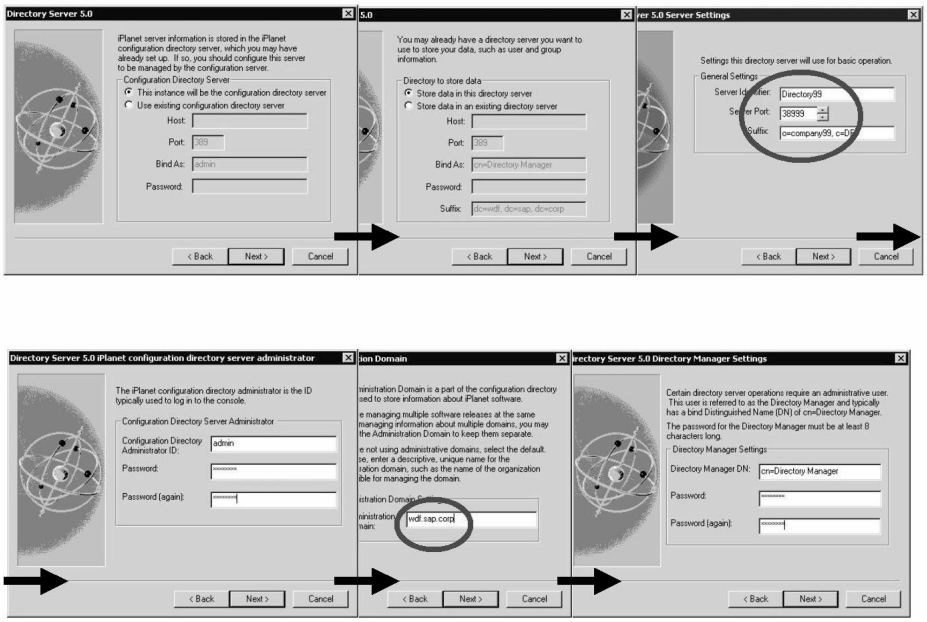


Figure 95: Installing the iPlanet Server 2/3

© 2003 SAP AG. All rights reserved.

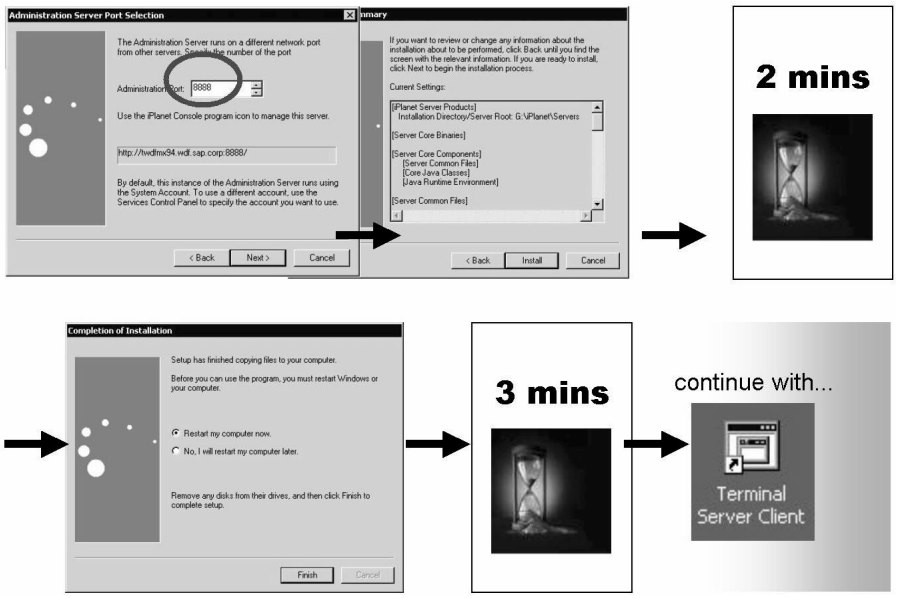


Figure 96: Installing the iPlanet Server 3/3

Business Example

Your company has decided to connect SAP systems to a directory service. As a member of the user administration team for SAP systems, it is your task to configure the SAP Web AS so that this connection can be implemented.

Roadmap

Once you have installed the iPlanet Server as described on “your” server, the iPlanet Console is available there. To help the participants with the exercise “Installing the iPlanet Console” (which has to be performed once for each host, not for each SAP system), you can either demonstrate the exercise on a participant server, or begin the task on “your” server and then, at the end, instead of choosing *Install*, choose *Cancel*.

Demonstrate the task “Setting Up a Directory Server” on your SAP system with the group number 00 at this point. For you, the Directory Server and the SAP system are running on the same host. This will be different hosts later, for the participants.

This demonstration should clarify the start point: There is a Directory Server and a “blank” SAP system that do not (yet) know anything about each other. Emphasize that nothing was prepared in the systems, and that the participants are therefore seeing all of the steps required for connection.

© 2003 SAP AG. All rights reserved.

Essentially: You do not need to (and should not) present the entire lesson first and then demonstrate all of the exercises. The topic is certainly complex and the Customizing transactions are anything but trivial ⇒ it is therefore useful to switch between small units of presentation, demonstration, and exercise tasks. If a group "gets stuck", they cannot successfully complete the rest of the exercises. These instructor notes suggest when you should insert which exercise.

You could therefore perform the exercises "Installing the iPlanet Console" and "Setting Up a Directory Server" immediately now, before discussing the SAP-specific information.

The starting point for the following steps is that an SAP system based on an SAP Web AS and a directory server are to be connected to each other. The focus here is on the LDAP Connector, an operating system program delivered by SAP, and the associated Customizing.

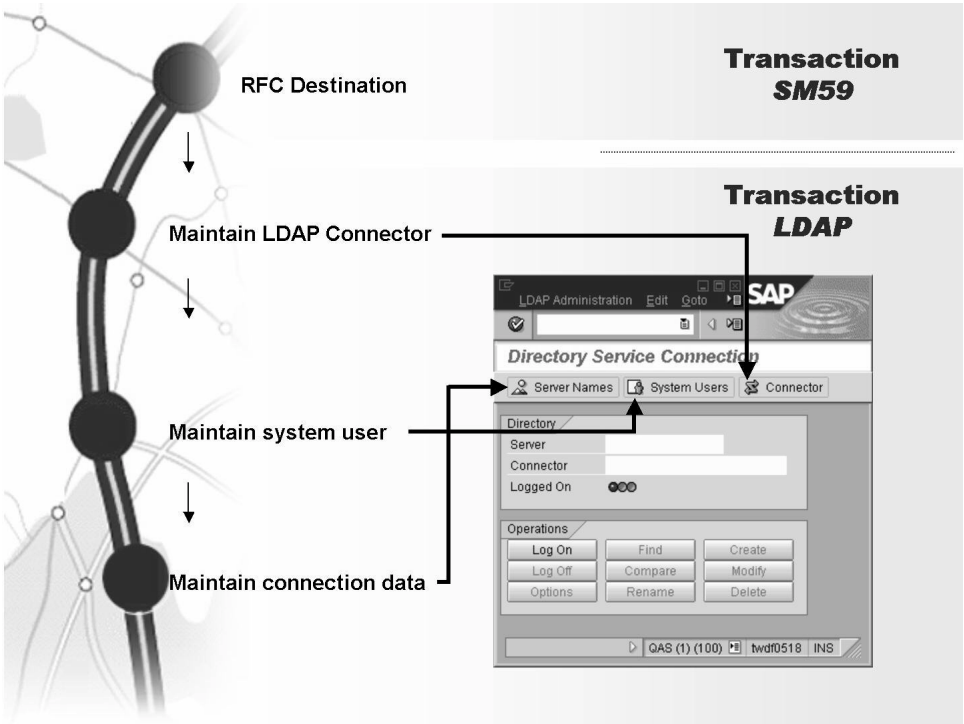


Figure 97: Roadmap for the Technical Connection of Directory Services

All of the steps for connecting the SAP system and the directory server are explained in more detail in the following:

© 2003 SAP AG. All rights reserved.

RFC Destination and LDAP Connector

Since SAP Basis 4.6, the LDAP Connector has been delivered with every SAP system. If you start the associated program at operating-system level (the program *ldap_rfc* in the kernel directory), it outputs a version number and additional technical details. If the LDAP program library is missing for your platform, the system displays an error message.

At operating system level of your host, open the Command Prompt (Choose *Start* → *Run* and then enter **`cmd`**), and enter **`ldap_rfc`**. The specified data is displayed as a result.

If you want to, you can navigate to the directory **`G:\usr\sap\<SID>\sys\exe\run`** with Windows Explorer – the program is stored there (together with other *ldap*.exe* programs).

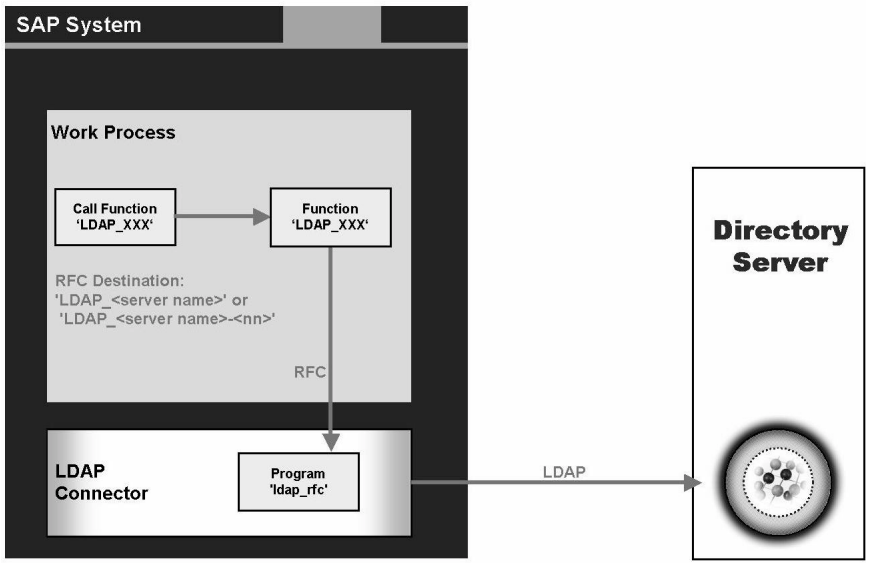


Figure 98: Connecting to Directory Services Using the LDAP Connector

As the program name suggests, the LDAP Connector is addressed using RFC communication. To do this, an RFC destination of the type **T** (TCP/IP) that activates the LDAP Connector as a registered server program is required.

Caution: Note that the name of the RFC destination must be written **in uppercase letters and without spaces**.

If only one LDAP Connector is running in an SAP system for each physical host, call the RFC destination *LDAP_<ApplServername>* (for example: *LDAP_TWDF0013*).

© 2003 SAP AG. All rights reserved.

If more than one is running, add a sequence number with a hyphen, that is, *LDAP_<ApplServername>-<Number>* (for example: *LDAP_TWDF0013-01*).

This proposed naming convention is the current SAP recommendation, which has changed from older versions of documentation. For more information, see SAP Note 587051.

The LDAP Connector should be registered at the local gateway, that is, you must maintain the gateway options so that the *Gateway Host* points to the local server and the *Gateway Service* points to **sapgw<instance number>**.

For the *Program ID* field, enter the name of the RFC destination.

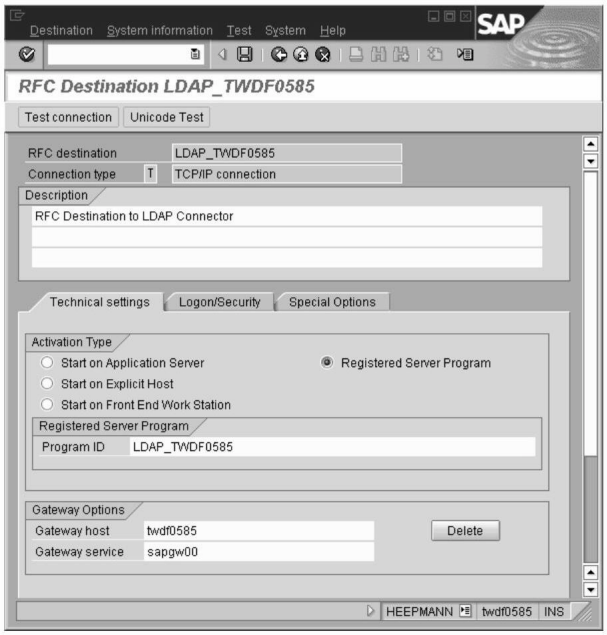


Figure 99: Creating the RFC Destination

You can perform all of the Customizing for connecting directory services in the SAP Reference IMG (Implementation Guide, transaction SPRO) under the path *SAP Web Application Server → System Administration → Directory Integration*. Alternatively, you can also use the central transaction for connecting to directory services, transaction LDAP. There, you can perform Customizing for the LDAP Connector in the *Connector* area. Use the field help (F1 and F4) and the documentation when creating the connection. The ***Connector Name*** **must be identical to the RFC destination**.

For the (logical) definition of the LDAP Connector in the SAP system, you must specify the application on which the (physical) LDAP Connector is running. You can also create connections to multiple directory services using one LDAP Connector. You usually set the *Status* to *Connector Is Active*

© 2003 SAP AG. All rights reserved.

– the Computing Center Management System (CCMS) then monitors the status of the LDAP Connector periodically and attempts (for example, after a restart of the SAP system) to make or keep it available using an auto-reaction method. You can set a trace level for the error log for error analysis (file *dev_<Name of the RFC destination>.trc*. You can evaluate this, for example, with transaction ST11).

After storing the Connector data, you can start the LDAP Connector manually from transaction LDAP.

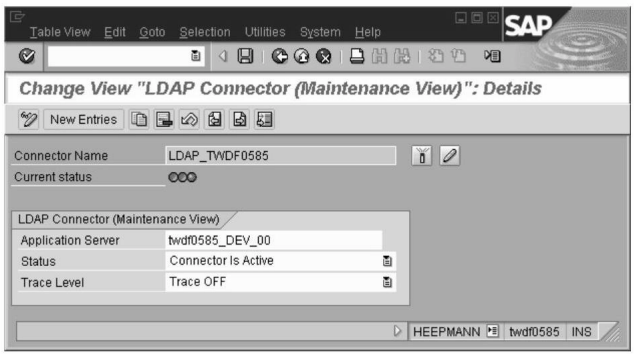


Figure 100: Defining the LDAP Connector

Demonstrate the exercise "RFC Destination and LDAP Connector" at this point. You should also show the participants the specified path in the IMG. In this and the following **exercises**, all important fields are explained. Direct customers that want information about the fields that are not discussed to the (very thorough) documentation.

Hint: Some screens in transaction LDAP may not seem very intuitive to you (or the customers). Before you complain about the developers: The entire LDAP configuration in the SAP system is, from a technical point of view, Customizing, and only a small number of particular screen elements are allowed for Customizing.

You can optionally show the participants the following in connection with the exercise:

- A connection test in transaction SM59 in the SAP system is now successful
- In transaction SMGW in the SAP system, choose the path *Goto* → *Logged on clients* (the connection to the LDAP connector now appears there, you can display details by double clicking it)

© 2003 SAP AG. All rights reserved.

- At operating system level, show the process *ldap_rfc.exe* in the Task Manager (on the *Processes* tab page, check the field *Show processes from all users*)
- In transaction RZ20, choose, for example *SAP CCMS Monitors for Optional Components* → *LDAP Connectors* and then show the method for the status attribute *Status* (you can suggest the ADM106 course to confused or interested customers)

System User

The SAP system must be able to log on to the directory service for the various LDAP operations. The directory services themselves use various authorization concepts (from anyone being able to retrieve information, to extensive identity checks using a directory server).

To do this, enter the logon data for an LDAP system user in the *System Users* area of transaction LDAP using a specific user ID. In addition to the name and credentials (password entry), this also includes information about whether a user may access the directory service to read only, and where the credentials are stored.

Note: The name of the created user ID corresponds neither to a user in the SAP system, nor to a user in the directory server. Rather, it is a key that is required for the connection data.

This means that you can store the same *User ID* with multiple server definitions in the SAP system. This simplifies administration if the same physical directory service is behind the various logical servers in the SAP system (perhaps to implement a different mapping for employees and vendors).

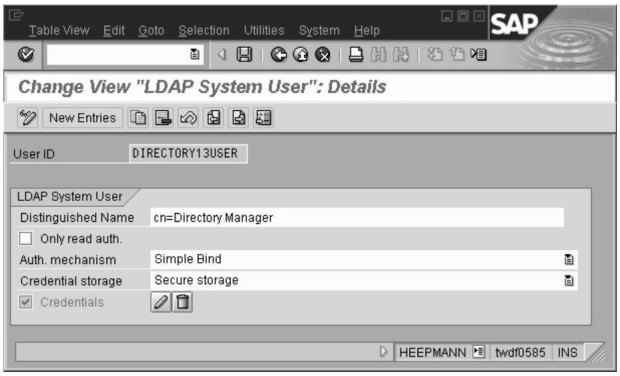


Figure 101: Defining the LDAP System User

© 2003 SAP AG. All rights reserved.

Now demonstrate the “System User” exercise. The *Secure Storage* value under *Credential Storage* is only possible if an SAP license exists. If the storage location changes (for example, after a system copy), the system user must be maintained again.

Connection Data

After these preparations, define the actual connection data for the directory server to be connected in the *Server* area of transaction LDAP.

This includes host name, port, and product name of the directory server in use. You also categorize the LDAP application (for example, as a directory for *Users*). The previously defined system user is also included in the connection data.

Note: You can enter a name of your choice for the *Server Name* field in the SAP system. However, you should avoid names that begin with **SAP_**, as these are reserved for SAP applications.

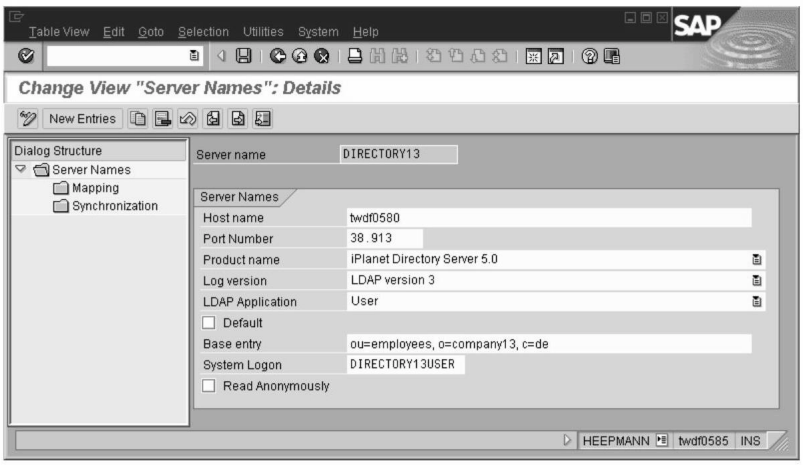


Figure 102: Defining the Connection Data

Now demonstrate the “Connection Data and Test” exercise. The *User* entry in the *LDAP Application* field is not only for documentation purposes – only by making this setting do the transfers of mapping proposals, for example, work later.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

279

Exercise 13: Connecting an SAP Web AS to a Directory Service

Exercise Duration: 35 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Create a directory server (with the iPlanet product)
- Make the technical connection of an SAP Web AS to a directory server

Business Example

A central directory service is already operating in your company for employee master data. It is your task as an administrator to connect a newly installed SAP system to this directory service.

Task 1 Installing the iPlanet Console

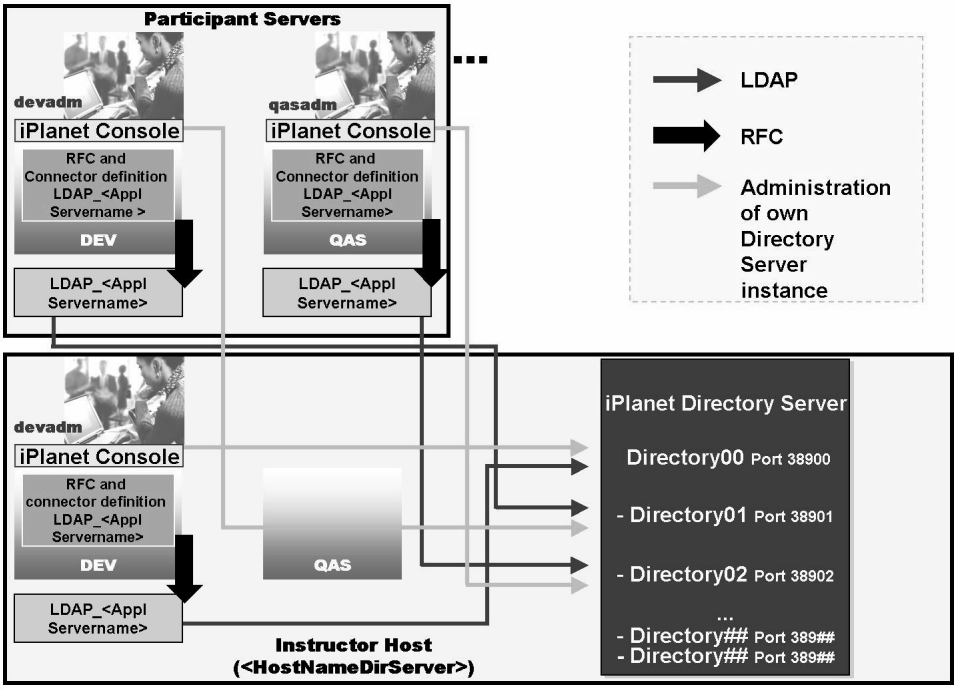


Figure 103: Complete Scenario of the Training Landscape

Install the iPlanet Console to connect to the directory server (shared by all exercise groups).

Continued on next page

© 2003 SAP AG. All rights reserved.

Caution: This exercise can be performed only **once per server**. Decide with your neighboring group which group is going to perform this part.

Note: The iPlanet product is used as an example here. The user interfaces and functions of other directory services are different; however, the basic concepts and required entries are identical.

1. Install the iPlanet Console at operating system level of your server in the target directory *G:\iPlanet\Servers*.

Result

The iPlanet Console is now available on your server (for both groups).

Task 2 Setting Up a Directory Server

Log on to the central directory server and create a new directory server.

Note: From this point, both groups can perform the exercises.

1. On your server, create a connection to the shared directory server with iPlanet Console. Your instructor will give you the logon data.

If you have followed the instructor notes and suggestions, the following information is correct:

User ID	`admin`
Password	`SAPNetWeaver`
Administration URL	`http://<HostDirServer>:8888`

For all groups, HostDirServer stands for the host on which you installed the central directory server.

---

2. Create a new instance of the iPlanet Directory Server with the following data:

Server identifier	`Directory##`, where `##` is your (two digit) group number
Network port	`389##`
Base suffix	`o=company##, c=de`

Continued on next page

 © 2003 SAP AG. All rights reserved.

Directory Manager DN	`cn=Directory Manager`
Directory Manager Password	`Dir##Manager` (enter this twice)

3. Create the two organizational units (attribute *organizationalUnit*), **employees** and **partners**, in your directory under the *company##* entry.

Hint: The reason for creating these two organizational units is that the organizational unit **employees** is to be synchronized with the SAP system later, while **partners** are administered in a different way (for example, with a third-party product).

Result

There is now a directory server (with the logical name *Directory##*) on the central directory server. All entries in this server refer to your company (due to the base entry *o=company##, c=de*). There are two sections in your company (the organizational units **employees** and **partners**). You know the logon data for a user (the *cn=Directory Manager*) with authorization to maintain entries in your directory server.

Task 3 RFC Destination and LDAP Connector

Technical configuration of the LDAP Connector

1. Create an RFC Destination, **LDAP_<ApplServername>** (ensure that you use the correct notation) of type **T** for your LDAP Connector. **<ApplServername>** stands for the host name on which your SAP system is running.

Caution: Ensure that you enter the name of the RFC destination in uppercase letters and without spaces.

Use the following settings:

Activation Type	*Registered server program*
Program ID	Identical to the name of the RFC destination, that is `LDAP_<ApplServername>`

Continued on next page

© 2003 SAP AG. All rights reserved.

Gateway Host	Host on which **your** SAP system is running, such as **`twdf0052`** (lowercase)
Gateway Service	**`sapgw<instance number>`**, such as **`sapgw00`** for the DEV group and **`sapgw10`** for the QAS group

2. Define and start your LDAP Connector using transaction LDAP.

Hint: Use the value help for the *Connector Name* and *Application Server* fields.

Result

The LDAP Connector that implements the connection of your SAP system to your directory server is now running at operating system level.

Task 4 System User

Maintain the logon data.

1. Store the logon data for the system user (including the password) on your directory server. To do this, use the following settings:

User ID	**`DIRECTORY##USER`**
Distinguished Name	**`cn=Directory Manager`**
Auth. Mechanism	*Simple Bind*
Credentials Storage	*Secure Storage*

Result

The system user (the user cn=Directory Manager) is known in the SAP system.

Task 5 Connection Data and Test

Maintain the connection data and perform a first test logon to the directory server.

1. Maintain the connection data for your directory server:

Continued on next page

 © 2003 SAP AG. All rights reserved.

Server Name	**DIRECTORY##**, the logical name of the directory service server
Host Name	Host name of the central directory service server (such as **twdf0042**)
Port Number	**389##**
Product Name	**iPlanet Directory Server 5.0**
Log Version	**LDAP v3**
LDAP Application	s**User**
Base entry	**ou=employees, o=company##, c=de** (in this order)
System Logon	**DIRECTORY##USER** (using F4 help, corresponds to the previously created user ID)

2. Test the configuration by logging on to, searching in, and logging off from the directory server.

This test is important to verify the entire configuration. If there are errors here, the data exchange will probably also fail. A few common error causes are listed below:

- Incorrect port on the directory server
- Organizational unit *employees* is missing or written incorrectly
- Incorrect notation for RFC destination or LDAP Container
- Gateway host written in upper case letters
- LDAP Connector not started

Result

The connection data (such as the product used, host name, port, and base entry) are known in the SAP system.

© 2003 SAP AG. All rights reserved.

Solution 13: Connecting an SAP Web AS to a Directory Service

Task 1 Installing the iPlanet Console

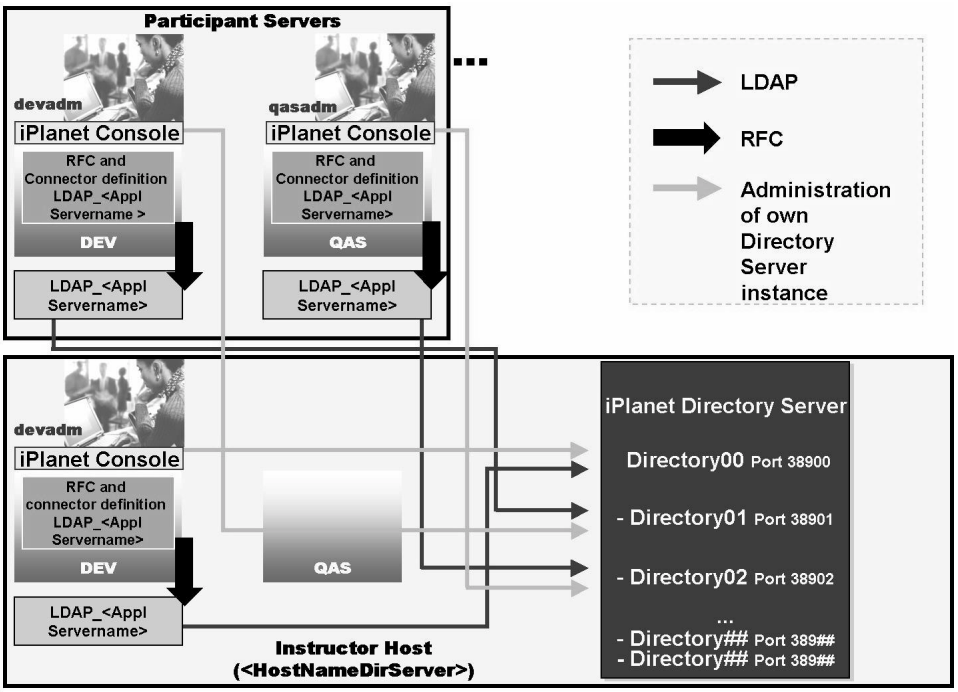


Figure 104: Complete Scenario of the Training Landscape

Install the iPlanet Console to connect to the directory server (shared by all exercise groups).

Caution: This exercise can be performed only **once per server**. Decide with your neighboring group which group is going to perform this part.

Note: The iPlanet product is used as an example here. The user interfaces and functions of other directory services are different; however, the basic concepts and required entries are identical.

Continued on next page

© 2003 SAP AG. All rights reserved.

1. Install the iPlanet Console at operating system level of your server in the target directory *G:\iPlanet\Servers*.

   a) If you have not already done so, log on to the Terminal Server Client (also known as the RDP Client) at operating system level using the user **`<sid>adm`**.
   b) Open Windows Explorer and navigate to the directory *G:\Additional_Files\ADM102\Directory Server\iPlanet 5.0*. Start the **setup.exe** program by double-clicking it.
   c) Perform the installation (by choosing the *Next* or *Yes* buttons). Note the following information:

      1. Confirm any warning that appears about DHCP configuration not existing by choosing *Yes*.

      2. Install **only** the *iPlanet Console*.

      Caution: It is not possible to install *iPlanet Server* with the Terminal Server Client.

      3. Change the suggested directory, *C:\iPlanet\Servers*, to *G:\iPlanet\Servers* (confirm the creation of this directory).

      At the end of the installation, you can choose whether you want to start the iPlanet Console immediately.

Result

The iPlanet Console is now available on your server (for both groups).

Task 2 Setting Up a Directory Server

Log on to the central directory server and create a new directory server.

Note: From this point, both groups can perform the exercises.

1. On your server, create a connection to the shared directory server with iPlanet Console. Your instructor will give you the logon data.

If you have followed the instructor notes and suggestions, the following information is correct:

User ID	**`admin`**
Password	**`SAPNetWeaver`**
Administration URL	**`http://<HostDirServer>:8888`**

Continued on next page

© 2003 SAP AG. All rights reserved.

For all groups, HostDirServer stands for the host on which you installed the central directory server.

---

a) If you have not already done so, start iPlanet Console at operating system level of your server. You can log on by choosing *Start → Programs → iPlanet Server Products → iPlanet Console 5.0*.

2. Create a new instance of the iPlanet Directory Server with the following data:

Server identifier	**`Directory##`**, where **`##`** is your (two digit) group number
Network port	**`389##`**
Base suffix	**`o=company##, c=de`**
Directory Manager DN	**`cn=Directory Manager`**
Directory Manager Password	**`Dir##Manager`** (enter this twice)

a) In the iPlanet Console, open the displayed server tree on the *Servers and Applications* tab page. First select the *Server Group* entry with the **left mouse button**. Then **right-click** and select *Create Instance Of → iPlanet Directory Server*.
b) Fill out the *Create New Instance* window as specified in the table above and confirm your entries with *OK*.
c) After a few seconds, the system should display a status message informing you that your directory server instance has been successfully created. Confirm the success message by choosing *OK*.

Continued on next page

 © 2003 SAP AG. All rights reserved.

3. Create the two organizational units (attribute *organizationalUnit*), **employees** and **partners**, in your directory under the *company##* entry.

Hint: The reason for creating these two organizational units is that the organizational unit **employees** is to be synchronized with the SAP system later, while **partners** are administered in a different way (for example, with a third-party product).

a) In your iPlanet Console, select **your** directory server, *Directory##*, and choose *Open* (by double-clicking, with the right mouse button, or using the button in the top right of the screen). A new window is opened.

b) On the *Directory* tab page, select the *company##* node with the left mouse button. Then right-click and select *New* → *Organizational Unit...*. Under *Name*, enter the value **`employees`**. The other fields are optional. Save your entries by choosing *OK*.

Repeat the last step for the **partners** organizational unit.

Result

There is now a directory server (with the logical name *Directory##*) on the central directory server. All entries in this server refer to your company (due to the base entry *o=company##, c=de*). There are two sections in your company (the organizational units **employees** and **partners**). You know the logon data for a user (the *cn=Directory Manager*) with authorization to maintain entries in your directory server.

Task 3 RFC Destination and LDAP Connector

Technical configuration of the LDAP Connector

1. Create an RFC Destination, **LDAP_<ApplServername>** (ensure that you use the correct notation) of type **T** for your LDAP Connector. **<ApplServername>** stands for the host name on which your SAP system is running.

Caution: Ensure that you enter the name of the RFC destination in uppercase letters and without spaces.

Use the following settings:

Continued on next page

 © 2003 SAP AG. All rights reserved.

Activation Type	*Registered server program*
Program ID	Identical to the name of the RFC destination, that is **LDAP_<ApplServername>**
Gateway Host	Host on which **your** SAP system is running, such as **twdf0052** (lowercase)
Gateway Service	**sapgw<instance number>**, such as **sapgw00** for the DEV group and **sapgw10** for the QAS group

a) In transaction SM59, choose the *Create* button. Enter the RFC destination **LDAP_<ApplServername>** in uppercase letters (for example, a group with the SAP system on the host twdf0052 would enter: **LDAP_TWDF0052**). Enter **T** as the connection type (start an external program using TCP/IP) and any description text.

b) Maintain the fields on the *Technical Settings* tab page, as specified in the table and save your connection.

Note: Do not be surprised if the connection test fails at this point. It will only succeed after the LDAP Connector is started.

Continued on next page

 © 2003 SAP AG. All rights reserved.

2. Define and start your LDAP Connector using transaction LDAP.

Hint: Use the value help for the *Connector Name* and *Application Server* fields.

a) In the LDAP transaction, navigate to the *Connector* area. Switch to change mode and choose *New Entries*.

b) The *Connector Name* must be identical to the RFC destination, that is, **LDAP_<ApplServername>**. Use the F4 help to make your selection.

The *Application Server* field (use the F4 help) points to the host on which the LDAP Connector is running. You can use the F1 help to obtain information about the syntax (for example, for a QAS group with the SAP system on the host twdf0052: **twdf0052_QAS**).

Set the *Status* to *Connector Is Active* and the *Trace Level* to *Trace OFF*.

Save your definition.

c) Leave change mode (for example, by calling transaction LDAP again). Navigate to the *LDAP Connector* view again and start your LDAP Connector using the *Start Connector* button.

A green traffic light should now indicate that the Connector was successfully started. If a green light does not appear after you choose *Enter*, use transaction ST11 to analyze the trace file **dev_<name of the RFC destination>.trc**.

Result

The LDAP Connector that implements the connection of your SAP system to your directory server is now running at operating system level.

Task 4 System User

Maintain the logon data.

1. Store the logon data for the system user (including the password) on your directory server. To do this, use the following settings:

User ID	**DIRECTORY##USER**
Distinguished Name	**cn=Directory Manager**

Continued on next page

 © 2003 SAP AG. All rights reserved.

Auth. Mechanism	*Simple Bind*
Credentials Storage	*Secure Storage*

a) On the initial screen of transaction LDAP, choose the *System Users* button. Switch to change mode and choose *New Entries*.
b) Fill out the fields as specified in the exercise text. Under *Credentials*, enter the password (Suggestion: **`Dir##Manager`**) that you have assigned to your Directory Manager (user cn=Directory Manager). When saving, the system prompts you for a Customizing request. If there is not yet a Customizing request, create one.

Result

The system user (the user cn=Directory Manager) is known in the SAP system.

Task 5 Connection Data and Test

Maintain the connection data and perform a first test logon to the directory server.

1. Maintain the connection data for your directory server:

Server Name	**`DIRECTORY##`**, the logical name of the directory service server
Host Name	Host name of the central directory service server (such as **`twdf0042`**)
Port Number	**`389##`**
Product Name	**`iPlanet Directory Server 5.0`**
Log Version	**`LDAP v3`**
LDAP Application	s**`User`**

Continued on next page

 © 2003 SAP AG. All rights reserved.

Base entry	`ou=employees, o=company##, c=de` (in this order)
System Logon	`DIRECTORY##USER` (using F4 help, corresponds to the previously created user ID)

a) On the initial screen of transaction LDAP, choose the *Server Names* button. Switch to change mode and choose *New Entries*.
b) Fill out the fields as specified in the table above and save your entries. Use the Customizing request from the previous section of the exercise.

2. Test the configuration by logging on to, searching in, and logging off from the directory server.

This test is important to verify the entire configuration. If there are errors here, the data exchange will probably also fail. A few common error causes are listed below:

- Incorrect port on the directory server
- Organizational unit *employees* is missing or written incorrectly
- Incorrect notation for RFC destination or LDAP Container
- Gateway host written in upper case letters
- LDAP Connector not started

a) On the initial screen of transaction LDAP, select your server and your Connector (using F4 help).
b) Perform the *Logon* operation. Check *Use System User* and choose *Execute*. On the initial screen of transaction LDAP, you should now see the status *Logged On*.
c) After successful logon, you can run a search. To do this, choose *Find* and perform the query with the default values. As a result, the system displays your tree (and in the tree, among other things, the organizational units **employees** and **partners**).
d) Log off from the directory server again (*Log Off* button).

Result

The connection data (such as the product used, host name, port, and base entry) are known in the SAP system.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- List the prerequisites for connecting an SAP Web Application Server (SAP Web AS) to a directory server
- Configure the SAP Web AS for connection to a directory server

Related Information

- Online documentation for the SAP Web AS 6.20: *mySAP Technology Components* → *SAP Web Application Server* → *Security* → *Directory Services (BC-SEC-DIR)*
- Quick Link */security* → *Security in Detail* → *Identity Management* → *Directory Service Integration* in the SAP Service Marketplace

© 2003 SAP AG. All rights reserved.

Lesson: Data Exchange with Directory Services

293 Lesson Duration: 60 minutes

Lesson Overview

The actual synchronization process is presented in this lesson. To do this, you must define which fields in the SAP system are associated with which attributes of the directory service (possibly to be newly created). This assignment is called **mapping**.

Lesson Objectives

After completing this lesson, you will be able to:

- Exchange data between the SAP Web AS and a directory server using the LDAP protocol

---

You could begin this lesson by asking the participants "After the technical preparations, could we now start the synchronization immediately?" Clever participants could point out that:

- There are elements such as role assignment and authorization profiles in the user master in the SAP system which the directory server does not know at all
- Some data (good example: telephone numbers) is stored differently in the SAP system and in the directory server
- The direction of the synchronization has not yet been defined anywhere

This discussion shows that it is first necessary to consider the mapping – performing the synchronization can then be performed simply by pressing a button.

Business Example

As a member of the user administration team for SAP systems, you have configured a connection in an SAP Web AS to a directory service. You now want to synchronize the data between the SAP system and the directory service using the LDAP protocol.

© 2003 SAP AG. All rights reserved.

Roadmap

The same applies for this lesson: You do not need to (and should not) present the entire lesson first and then demonstrate all of the exercises. The topic is certainly complex and the Customizing transactions are anything but trivial ⇒ it is therefore useful to switch between small units of presentation, demonstration, and exercise tasks.

You can navigate to the Customizing presented in this lesson:

- Through the SAP Reference IMG in transaction SPRO) under the path *SAP Web Application Server* → *System Administration* → *Directory Integration*
- Using transaction LDAP and the *Server Names* tab page
- Using transaction LDAPMAP

From a functional point of view, all three methods are identical. To avoid confusing the participants, we recommend that you demonstrate the route using transaction SPRO once and to use transaction LDAP otherwise. The existence of transaction LDAPMAP is deliberately not mentioned in the following.

After the technical connection of a directory server to an SAP system based on SAP Web AS, the actual data exchange has to be prepared and performed. To do this, information is required about how SAP data fields are mapped to directory attributes, which fields are used where (mapping indicators), and the direction in which data is synchronized (synchronization indicators).

© 2003 SAP AG. All rights reserved.

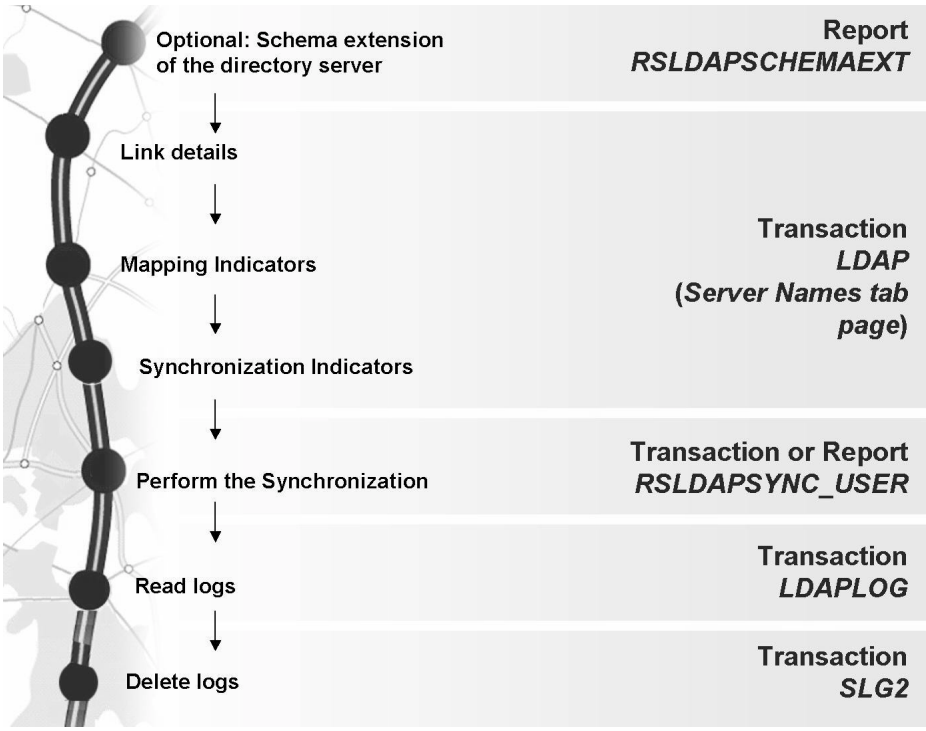


Figure 105: Roadmap for Data Exchange with Directory Services

All of the steps are described in more detail in the following:

Mapping

Mapping describes the assignment of SAP fields to directory attributes. In the simplest case, this is a 1:1 assignment. However, more complex link rules may be required.

Schema Extension

The directory servers from various vendors have much in common in their data models (schemata). On the other hand, there are differences in the details. Understandably, there are no attributes for SAP-specific information in the vendor standards.

- SAP user name
- Alias
- Validity period
- Settings for logon language, printer, number formats, time zone
- Assigned roles
- Assigned authorization profiles

© 2003 SAP AG. All rights reserved.

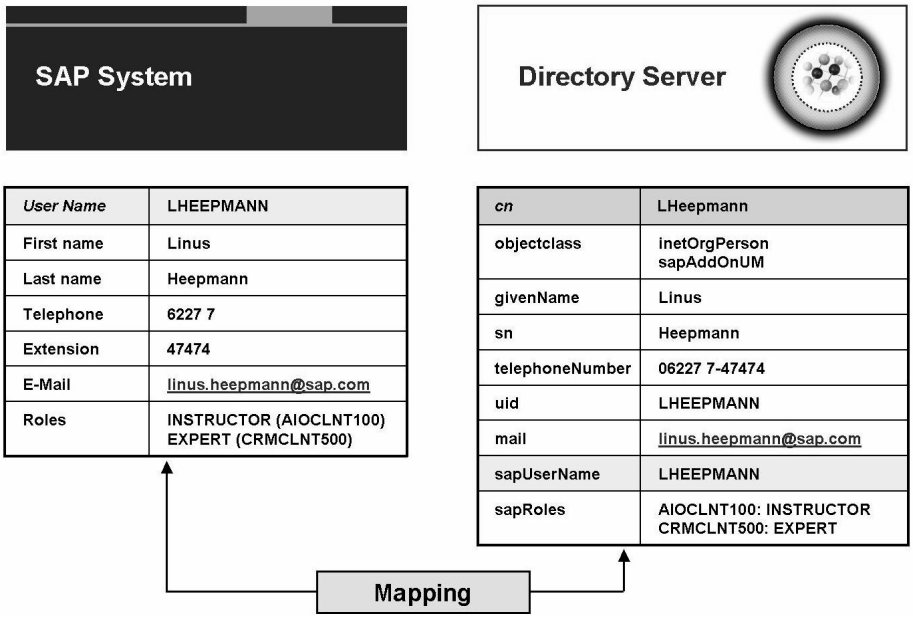


Figure 106: Mapping SAP Data Fields to Directory Attributes

SAP has decided not to develop its own schema, but rather to support the schemata available on the market. This is important for heterogeneous system landscapes, if you want to connect non-SAP applications to your directory service in addition to SAP systems.

As the delivered schemata for the various products do not know many SAP data fields, you must use schema extensions to make these fields known. An extension of this type is provided and imported into the directory server in a special format (LDIF for “LDAP Data Interchange Format”).

You can easily create files of this type in the SAP system for all current products using the report **RSLDAPSCHEMAEXT**. After you have chosen a product, this report outputs a list that you can save as a file on your front end computer and then import into your directory server. For more information, see the comment lines in this file.

The delivered schema extensions are intended as support from SAP for its customers. They work with the product versions (of directory service and SAP Web AS) for which they were developed – however, the customer does not have an entitlement to support from SAP, with the exception of certified vendors.

Discussions are currently taking place with a number of directory vendors that want to deliver SAP-specific schema extensions with their products; in this case, generating and importing a schema extension would no longer be necessary.

© 2003 SAP AG. All rights reserved.

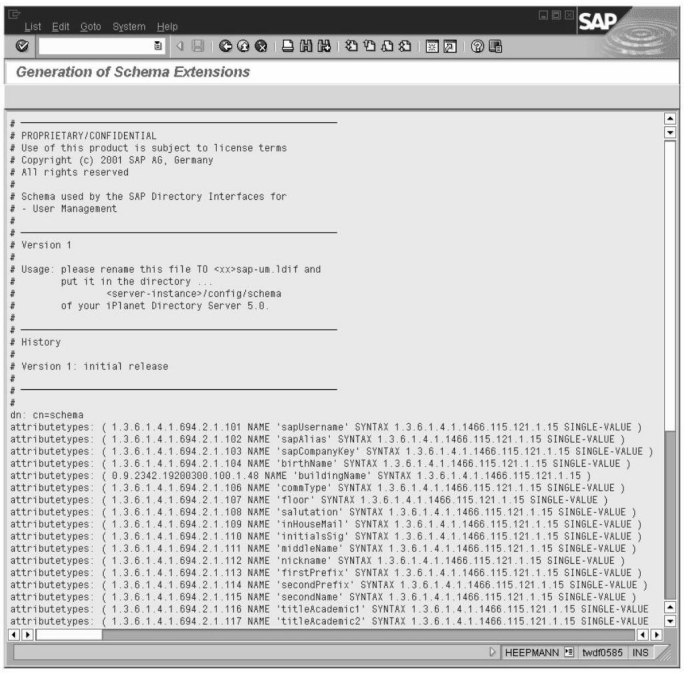


Figure 107: Generating a Schema Extension

Demonstrate the “Schema Extension” exercise now. You can demonstrate the optional part in your **SAP GUI in the training room** – although it only stores the file locally, the participants will have seen the procedure (and will then no doubt realize that it is not difficult to generate the appropriate LDIF file for their product).

Linking Details

Essentially, mapping is about linking an SAP data field with a directory attribute. As stated previously, this can be a simple 1:1 assignment or a more complex rule, if, for example, multiple SAP fields are to be mapped to one attribute (or multiple attributes to one SAP field).

© 2003 SAP AG. All rights reserved.

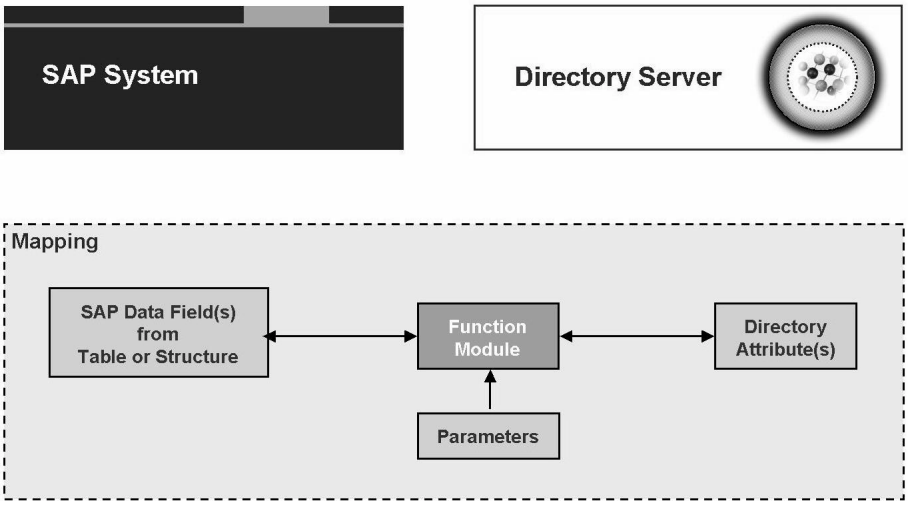


Figure 108: Structure of a Mapping

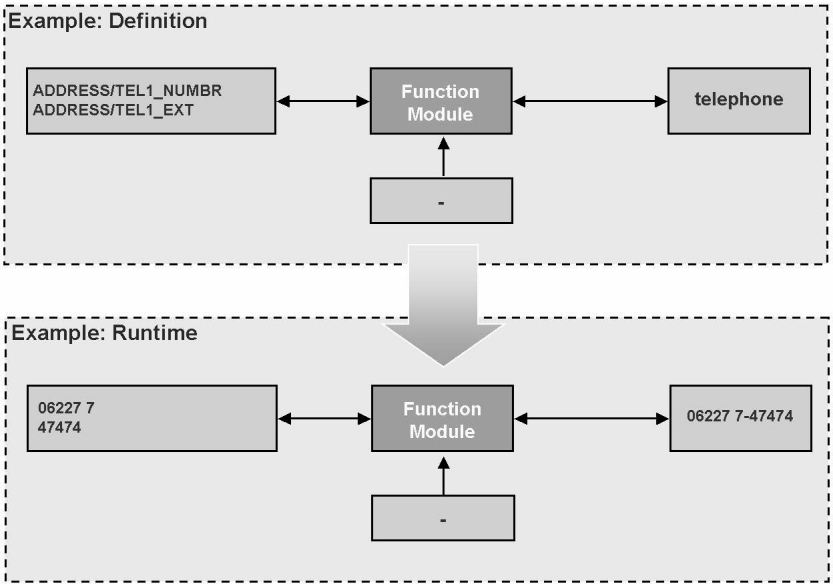


Figure 109: Example of a Mapping

SAP delivers a number of function modules that implement certain types of link. Additional parameters are required to some extent, for example, to specify a particular character at which a string is to be divided. Customers can, of course, also develop their own modules for particular purposes.

Note: The names of the delivered functions modules all start with *MAP_** and are in the function group *FSLDAP_MAP*. For information about their functions, see the BC-SEC-DIR area of the SAP Library.

© 2003 SAP AG. All rights reserved.

Tip: Demonstrate the *Server Names* tab page in transaction LDAP at this point, and, in change mode, double click on *Mapping*. You can go to the Customizing for **one** entry by clicking *Add Entry*. You could, for example, suggest how the definition for mapping the telephone number would look...

The customers have misgivings, and are imagining themselves maintaining connections for days? Isn't it good that SAP delivers mapping proposals for all important products (this is shown in the next demonstration/exercise).

Mapping Indicators

In addition to the link (one or more SAP data fields with one or more directory attributes), a mapping also includes its purpose. Is the mapping to be used only for export, only for import, or for both synchronization directions? Are there mandatory fields, without which it is not possible to create an entry in the directory server?

These questions and others are answered by the indicators to be maintained for each mapping. You can perform the associated Customizing in transaction LDAP on the *Server Names* tab page, in the *Mapping* area.

Meaning of the Mapping Indicators

Mapping Indicator	Meaning
Filter	Mapping used for the search filter of the directory server; corresponds to the key in the SAP system (exists only once; must be appropriate for the LDAP application; usually for the same mapping as the RDN flag)
Import	Mapping that can be used for reading directory entries
Export	Mapping that can be used for writing directory entries
Required	Mapping without which it is not possible to write an entry in the directory (can only be chosen for mappings with the *Export* indicator)
RDN	Mapping used to create the RDN (must exist for exactly one mapping)

© 2003 SAP AG. All rights reserved.

SAP provides a mapping proposal for all important products, which you can transfer in the mapping maintenance at the push of a button.

Hint: This function is only possible if you chose the LDAP application *User* when maintaining the connection data and have selected a particular product.

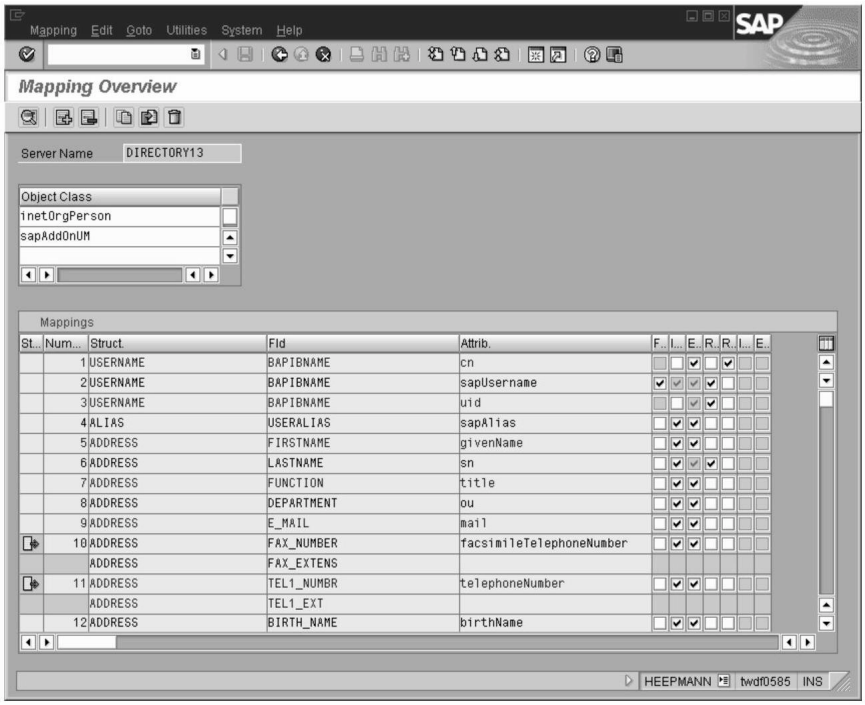


Figure 110: Mapping (in Accordance with an SAP Proposal, Extended with an Additional Link)

Demonstrate the first two parts of the "Configuring the Mapping and Synchronization" exercise at this point.

You can show a good example (that is discussed on the slides) by double clicking the link for the *telephoneNumber* directory attribute: You can see here very clearly that the characters "-" is used to help the *MAX_SPLIT_CHAR* function module link the two SAP fields with one attribute.

Synchronization Indicators

As the customer, you must ultimately decide the direction in which a synchronization works. There is no SAP proposal for this, as which data is maintained where depends on the scenario. It is certainly conceivable

© 2003 SAP AG. All rights reserved.

that, during a synchronization for a user, some information is transferred from the directory server to the SAP system (import) and other data is transferred in the opposite direction, that is, is written to the directory (export).

You can set the synchronization indicators for each individual mapping to *Import*, *Export*, or not at all in transaction LDAP on the *Server Names* tab page, in the *Synchronization* area.

Meaning of the Synchronization Indicators

Synchronization Indicators	Meaning
Import	Object is imported during the synchronization (this can only be selected for mappings with the *Import* mapping indicator)
Export	Object is exported during the synchronization (this can only be selected for mappings with the *Export* mapping indicator)

Regardless of the indicators that you have set, the following rules apply:

- The fields that are mandatory in the directory (*Required* mapping indicator) and the key fields in the SAP system (*Filter* mapping indicator) are automatically exported when creating (but not when updating) an entry in the directory.
- The fields that are mandatory in the SAP system (this cannot be identified in the mapping maintenance) and the key fields in the SAP system (*Filter* mapping indicator) are automatically imported when creating (but not when updating) an entry in the SAP system.

For existing objects (that is, when updating objects), only the synchronization indicator settings apply.

Caution: In addition to field mapping, the synchronization indicators are vital for a correct data exchange. Incorrect Customizing can lead to data loss or inconsistencies.

© 2003 SAP AG. All rights reserved.

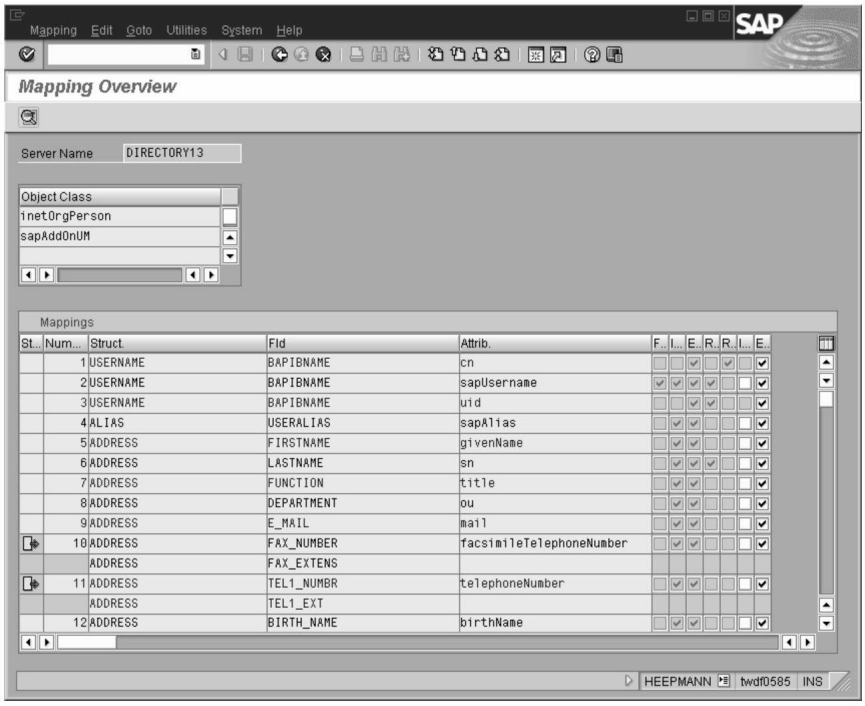


Figure 111: Maintaining the Synchronization Indicators

Demonstrate the third part of the "Configuring the Mapping and Synchronization" exercise now.

Executing and Monitoring the Synchronization

After you have made all settings, you can perform the synchronization – perhaps manually initially as a test, and then later, in production, as a periodically background job.

Preparing and Performing the Synchronization

It is important that you are clear about what data is maintained where. Not all user attributes must necessarily be administered in one place – in this way, users could be created, assigned a telephone number, and assigned a SAP role in different places. User should, of course, be created at exactly one place.

Usually, either the directory or the SAP system defines the existence of users. Users newly created there can then be automatically generated in the system and users deleted there can be automatically deleted or locked. However, this "one way street" is not mandatory. A synchronization

© 2003 SAP AG. All rights reserved.

indicator must be maintained for each mapping where attributes can be synchronized in different directions (exactly one direction for each attribute; either import or export).

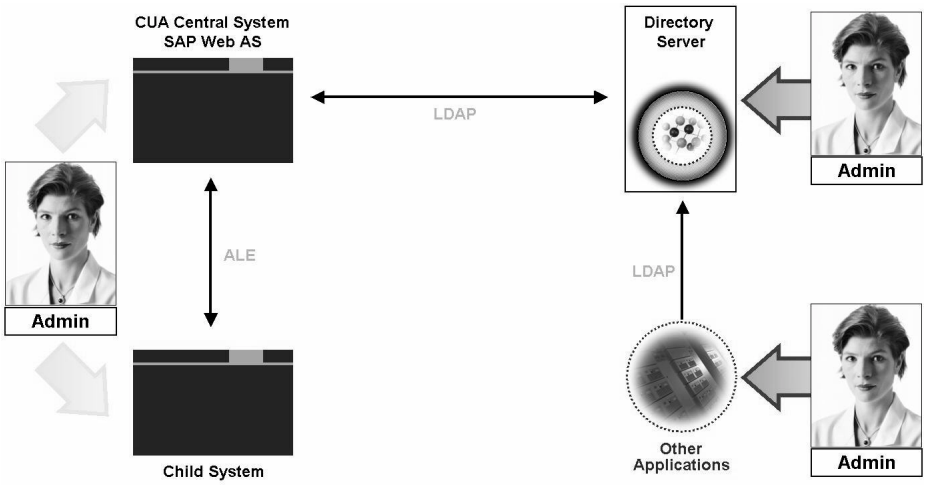


Figure 112: User Administration in the Directory Server (or in an External Application)

In the illustrated case, users are either created manually in the directory server, or are created by an external application, such as in an HR system. Newly created users would then be created in the SAP system during the next synchronization. If this is the central system of a CUA, the user would also be created in the specified child systems.

In this scenario, it is still possible for an SAP administrator to administer the user master in the central system and (depending on the configuration) also in the child systems.

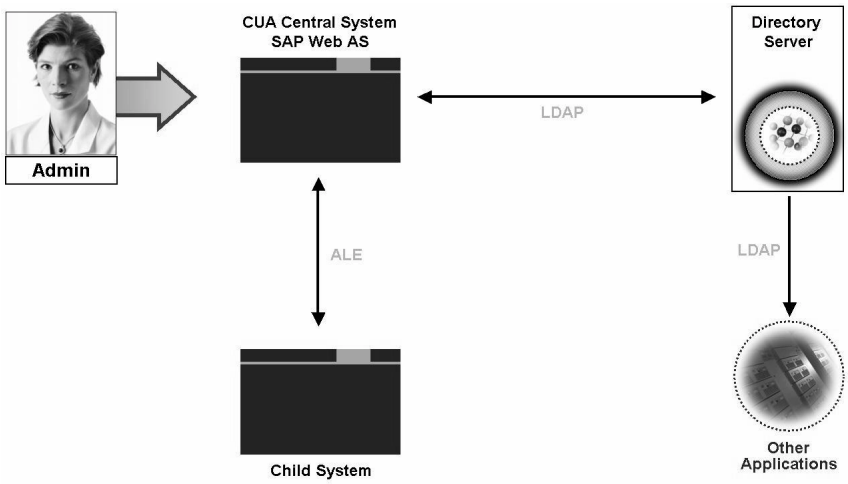


Figure 113: User Administration in the SAP System

© 2003 SAP AG. All rights reserved.

Another scenario could have the connected SAP system as the leading system. Users created there are immediately distributed to the affected child systems. After the next synchronization, these users also exist in the directory server, and can be accessed by other applications.

You always control the actual synchronization with the directory server from the SAP system, using transaction RSLDAPSYNC_USER or the report with the same name: **RSLDAPSYNC_USER**. You can start the report in dialog (for example, with transaction SA38), or schedule it as a background job with a suitable variant. You specify the desired LDAP server when you call the report (the system can determine the LDAP Connector automatically). You can also restrict the synchronization to certain user names.

During the process of the synchronization, three groups of users are identified, which are processed sequentially in accordance with the settings of report **RSLDAPSYNC_USER** and the synchronization indicators.

1. Handling of the users that exist only in the SAP system
2. Handling of the users that exist only in the directory server
3. Handling of the users that exist in both repositories

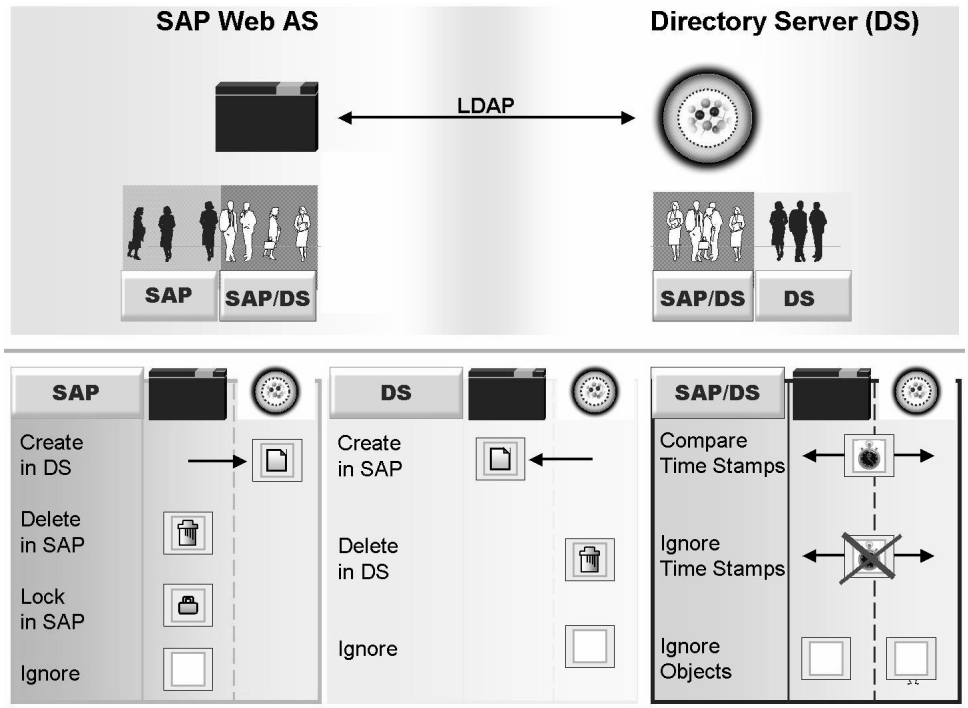


Figure 114: Sequence and Options for Synchronization

© 2003 SAP AG. All rights reserved.

RSLDAPSYNC_USER offers different options for each of the identified user groups. The *Ignore Objects* default settings for each group are non-critical. With a run of this type, no objects are changed; however, you can find out from the log how many objects would be synchronized.

- The users SAP*, DDIC, and EARLYWATCH are basically excluded from every synchronization.
- Instead of deleting users (that do not exist in the directory), we recommend that you simply lock them (for audit purposes).
- Before you delete users (that do not exist in the SAP system) from the directory, you must be sure that the entries are not required for other applications.
- *Compare Time Stamp* for the users that exist in both repositories performs a delta synchronization (that is, only users changed since the last run are synchronized).
- *Ignore Time Stamp* for the users in both repositories performs a complete synchronization (that is, all users are synchronized in accordance with the synchronization indicators).

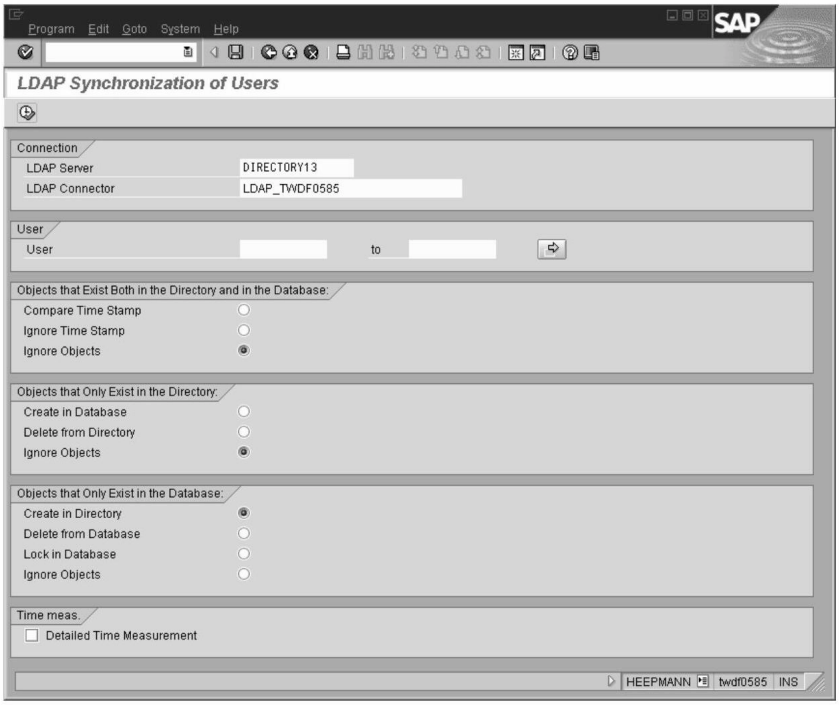


Figure 115: Performing the Synchronization with Report RSLDAP-SYNC_USER

© 2003 SAP AG. All rights reserved.

Demonstrate the first part of the exercise “Performing the Synchronization” now.

Logs

If you run **RSLDAPSYNC_USER** in dialog, the system displays the associated log immediately after the run. Small traffic light icons show you whether there were problems. The runtime required is also displayed.

If you ran the synchronization in the background (or want to access older logs), transaction LDAPLOG allows you to evaluate historical logs. The selection screen allows you to use convenient search options.

Note: As an alternative to transaction LDAPLOG, you can also use the (general) transaction SLG1 (use it with object *LDAPSYNC* and subobject *USER*).

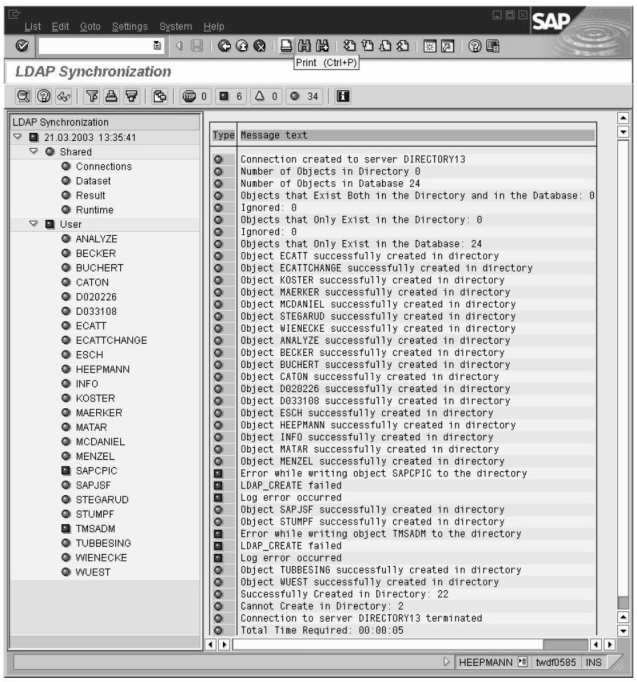


Figure 116: Result Log in the SAP System

The logs use the Application Log, a tool of the SAP Web AS, to collect messages, exceptions, and errors (component BC-SRV-BAL). This tool provides a range of user-friendly functions for logs. You can archive old

© 2003 SAP AG. All rights reserved.

logs (with transaction SARA, archiving object BC_SBAL), or, if they are no longer required, delete them manually or automatically (with transaction SLG2).

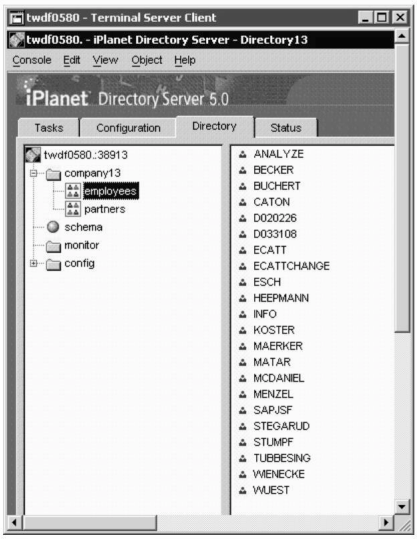


Figure 117: Users Created in the Directory Server

Which options exist for directory accessing entries in the server? One approach is the *Search* operation in transaction LDAP. In addition to the directory services vendors, there are also other LDAP browsers available from third-party vendors.

Demonstrate the last part of the exercise “Performing the Synchronization” now. If the participants are interested (and there is enough time), you can now demonstrate the import, that is, the middle (optional) part of the exercise “Performing the Synchronization”. Alternatively, you can also demonstrate the import after the participants have done the exercises.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

307

Exercise 14: Data Exchange Between the SAP System and the Directory Service

Exercise Duration: 30 minutes

Exercise Objectives

After completing this exercise, you will be able to:

- Generate and import the schema extensions proposed by SAP
- Configure the mapping and synchronization of SAP data fields to directory attributes
- Perform a synchronization between the SAP system and the directory service
- Monitor the synchronization using logs

Business Example

You have made a connection between your SAP system and a directory service. Employee master data is maintained in the SAP system and is to be transferred to the directory service.

Your company is also intending to implement external software for fax connections. Every employee is to receive his or her own fax number using the external software. These numbers are to be transferred to the SAP system using the directory service.

Task 1 Schema Extension

Extension of the existing directory server schema with SAP specific fields using an LDIF file.

Caution: The first part is optional; the required LDIF file is already on your server. The second part, on the other hand, is mandatory.

If you have “weak” participants who are overwhelmed by this task, you can also simply copy the schema extension *60sap-um.ldif* on the central directory server to the appropriate subdirectories

- After the participants have created their directory server and
- Before the participants restart their directory server (the participants should do at least this themselves)

Continued on next page

© 2003 SAP AG. All rights reserved.

1. **Optional:** Use report **RSLDAPSCHEMAEXT** to generate a schema extension for the iPlanet Directory Server 5.0. Store the extension in a file called *60sap-um.ldif* on your server.

   Hint: To store the file on your server (and not in the training room), use the SAP GUI installed on your server (in the Terminal Server Client window).

2. **Mandatory:** Copy the file *60sap-um.ldif* (either from the first part of the exercise or from *G:\Additional_Files\ADM102\Directory Server\60sap-um.ldif*) from your server to the *\config\schema* subdirectory of **your** directory server on the central directory server.

   Restart your directory server.

Result

Your directory server now knows SAP-specific data fields such as SAP user name, role, and authorization profile.

Task 2 Configuring the Mapping and Synchronization

Mapping (of SAP data fields to directory attributes), purpose (what the mapping is used for), and synchronization direction (import or export)

1. Import the SAP proposal for mapping.
2. For the iPlanet Directory Server, the *User ID* (abbreviation uid) is a required field. Map the SAP field *USERNAME/BAPINAME* to the uid attribute of the directory server. This field is to be be used for export, and is required for the directory server.
3. Define that all SAP data fields are to be exported.

Result

The assignment of SAP data fields to directory attributes is performed using default values. During a synchronization, all available user data is transferred from the SAP system to the directory server.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Task 3 Performing the Synchronization

Export data from the SAP system to the directory server; optionally also import data from the directory server to the SAP system. Evaluate all logs that have been created for LDAP operations.

Hint: A prerequisite for this exercise is that you are working in a client in the SAP system that is assigned to a logical system. This is relevant because when the assigned SAP roles are transferred, the logical system is stored together with this information in the directory server. If you have set up a CUA, this prerequisite is fulfilled.

If this prerequisite is not fulfilled, use transaction SCC4 to assign the client in which you are working the logical system **`<SID>CLNT<client>`** (such as **`DEVCLNT100`**).

1. **Export.** First determine how many objects (in this case, users) would be affected by a synchronization.

   Create these users in the directory server. Check the entries with the iPlanet Console.

2. **Optional: Import.** Maintain the fax numbers using an external tool.

   Change the synchronization rules so that in the future, the fax number is imported from the directory server to the SAP system. Then enter a new fax number in your directory server for user ADM102-## and synchronize it with the SAP system.

   Hint: Enter the fax number in exactly the format that is expected from the mapping (in this case, exactly one hyphen separating the main number from the extension number).

   Note whether the user master record has changed in the SAP system. If you have Central User Administration (CUA) in operation, check the user master record in the child system.

3. Display the logs for all LDAP operations for today in the SAP system.

© 2003 SAP AG. All rights reserved.

Solution 14: Data Exchange Between the SAP System and the Directory Service

Task 1 Schema Extension

Extension of the existing directory server schema with SAP specific fields using an LDIF file.

Caution: The first part is optional; the required LDIF file is already on your server. The second part, on the other hand, is mandatory.

If you have "weak" participants who are overwhelmed by this task, you can also simply copy the schema extension *60sap-um.ldif* on the central directory server to the appropriate subdirectories

- After the participants have created their directory server and
- Before the participants restart their directory server (the participants should do at least this themselves)

1. **Optional:** Use report **RSLDAPSCHEMAEXT** to generate a schema extension for the iPlanet Directory Server 5.0. Store the extension in a file called *60sap-um.ldif* on your server.

   Hint: To store the file on your server (and not in the training room), use the SAP GUI installed on your server (in the Terminal Server Client window).

   a) Start SAP Logon **on your server** (in the Terminal Server Client, also known as the RDP Client) and log on to your SAP system.
   b) Use transaction SA38 to execute report **RSLDAPSCHEMAEXT**. Enter `iPlanet Directory Server 5.0` as the product.
   c) Save the unconverted output as `60sap-um.ldif` by choosing *System* → *List* → *Save* → *Local File*.
   d) Log off the SAP GUI. Use the local SAP GUI in the training room for the other exercises.

2. **Mandatory:** Copy the file *60sap-um.ldif* (either from the first part of the exercise or from *G:\Additional_Files\ADM102\Directory Server\60sap-um.ldif*) from your server to the *\config\schema* subdirectory of **your** directory server on the central directory server.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Restart your directory server.

a) If you have not already done so, log on to the Terminal Server Client (also known as the RDP Client) at operating-system level using the user **`<sid>adm`**.
b) Open Microsoft Windows Explorer (for example, by choosing *Start* → *Programs* → *Accessories* → *Windows Explorer*) and navigate to the file *60sap-um.ldif* (for the path, see the exercise text).
c) Display the *\config\schema* subdirectory of **your** directory server on the central directory server. To do this, choose *Start* → *Run* and enter the following: **`\\<DirServerHostName>\G$`** (For example, for a central directory server on host twdf0042: **`\\twdf0042\G$`**). Navigate to the subdirectory *\iPlanet\Servers\slapd-Directory##\config\schema* (For example, for group 13: *\iPlanet\Servers\slapd-Directory13\config\schema*).
d) Add the file *60sap-um.ldif* to the existing schemata.
e) In the iPlanet Console, choose *Restart Directory Server* for **your** directory server on the *Tasks* tab page. Confirm the security query and wait for the success message (this will take about 10 seconds).

Result

Your directory server now knows SAP-specific data fields such as SAP user name, role, and authorization profile.

Task 2 Configuring the Mapping and Synchronization

Mapping (of SAP data fields to directory attributes), purpose (what the mapping is used for), and synchronization direction (import or export)

1. Import the SAP proposal for mapping.

   a) In transaction LDAP, choose *Server Names* (or use the SAP Reference IMG in transaction SPRO and choose *SAP Web Application Server* → *System Administration* → *Directory Integration* → *Configure LDAP Server*)
   b) Switch to change mode and then double-click *Mapping*.
   c) Choose *Import Proposal* (or choose *Utilities* → *Import Proposal*). Leave all fields unchanged and return to the previous screen, where you can save the data (Customizing request).

Hint: A proposal of this type is possible because you are using the generated schema extension and have declared the product in use in the SAP system.

Continued on next page

 © 2003 SAP AG. All rights reserved.

2. For the iPlanet Directory Server, the *User ID* (abbreviation uid) is a required field. Map the SAP field *USERNAME/BAPINAME* to the uid attribute of the directory server. This field is to be be used for export, and is required for the directory server.

   a) In the server maintenance of transaction LDAP, double-click *Mapping* (in change mode).
   b) Place the cursor in the third row (the new mapping procedure should be inserted here for the sake of clarity) and choose *Add Entry*. On the mapping details, select the SAP field *USERNAME/BAPINAME* using the F4 help and enter the directory attribute **`uid`**. Leave all other fields unchanged and return to the previous screen.
   c) For this new mapping procedure, check *Export* and then *Required*. Return to the previous screen and save your work.

3. Define that all SAP data fields are to be exported.

   a) In the server maintenance of transaction LDAP, double-click *Synchronization* (in change mode). You can define for each field whether it is to be imported or exported.
   b) Select the *Export* field in all rows.

      Note: Within the table control, you can change the checkboxes with the space bar and move the cursor with the arrow keys.

   c) Return to the previous screen and save your work.

Result

The assignment of SAP data fields to directory attributes is performed using default values. During a synchronization, all available user data is transferred from the SAP system to the directory server.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Task 3 Performing the Synchronization

Export data from the SAP system to the directory server; optionally also import data from the directory server to the SAP system. Evaluate all logs that have been created for LDAP operations.

Hint: A prerequisite for this exercise is that you are working in a client in the SAP system that is assigned to a logical system. This is relevant because when the assigned SAP roles are transferred, the logical system is stored together with this information in the directory server. If you have set up a CUA, this prerequisite is fulfilled.

If this prerequisite is not fulfilled, use transaction SCC4 to assign the client in which you are working the logical system **`<SID>CLNT<client>`** (such as **`DEVCLNT100`**).

1. **Export.** First determine how many objects (in this case, users) would be affected by a synchronization.

   Create these users in the directory server. Check the entries with the iPlanet Console.

   a) In transaction SA38, select report **RSLDAPSYNC_USER**. Under *LDAP Server*, enter your server using the F4 help and choose *Execute*.

      As your directory server does not yet contain any entries, only users that exist only in the database (in the SAP system) are displayed.

   b) Now start report **RSLDAPSYNC_USER**, but have *Objects that only exist in the database* created in the directory. You can see in the log which users were created in the directory server.

      Hint: Special users without address data (such as SAPCPIC) are not created during the synchronization (with the default values, the SAP field *Last name* is required).

   c) Navigate in the iPlanet Console for your directory server to the *Directory* tab page in the tree *company##* → *employees*. The users that have just been created are displayed there.

2. **Optional: Import.** Maintain the fax numbers using an external tool.

Continued on next page

 © 2003 SAP AG. All rights reserved.

Change the synchronization rules so that in the future, the fax number is imported from the directory server to the SAP system. Then enter a new fax number in your directory server for user ADM102-## and synchronize it with the SAP system.

Hint: Enter the fax number in exactly the format that is expected from the mapping (in this case, exactly one hyphen separating the main number from the extension number).

Note whether the user master record has changed in the SAP system. If you have Central User Administration (CUA) in operation, check the user master record in the child system.

a) In change mode on the *Server Names* tab page in transaction LDAP, choose *Synchronization* and set the synchronization indicator for the attribute *facsimileTelephoneNumber* to *Import*.

b) Navigate in the iPlanet Console for your directory server to the *Directory* tab page in the tree *company##* → *employees*. Double click the entry ADM102-## and change the fax number. Confirm this with *OK*.

c) Start report **RSLDAPSYNC_USER** with the setting that the time stamp is to be compared for *Objects that exist in the directory and in the database*.

   You should be able to see in the log that a synchronization was required for your ADM102-## user.

d) In transaction SU01, the new fax number for the ADM102-## user is displayed on the *Address* tab page.

   If you have CUA set up, you can also see the changed fax number in the child system (as long as the distribution parameter for the fax number is not set to *Local*).

3. Display the logs for all LDAP operations for today in the SAP system.

   a) Run transaction LDAPLOG with the default values.

Result

Congratulations on successfully connecting an SAP Web AS to a directory server using LDAP.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Exchange data between the SAP Web AS and a directory server using the LDAP protocol

Related Information

- Quick Link */security* → *Security in Detail* → *Identity Management* → *Directory Service Integration* on the SAP Service Marketplace
- Online documentation for the SAP Web AS 6.20: *mySAP Technology Components* → *SAP Web Application Server* → *Security* → *Directory Services (BC-SEC-DIR).*

© 2003 SAP AG. All rights reserved.

Unit Summary

You should now be able to:

- Outline the concept of Central User Administration
- Evaluate the usefulness of Central User Administration for your company
- Define logical systems
- Correctly set up the required users for RFC connections
- Create the RFC connections required for CUA
- Configured the ALE communication required for CUA
- Set up Central User Administration
- Maintain and distribute user data using the CUA
- Perform simple troubleshooting
- Remove a CUA
- Explain the concept of user administration with application to the integration of directory services.
- List the prerequisites for connecting an SAP Web Application Server (SAP Web AS) to a directory server
- Configure the SAP Web AS for connection to a directory server
- Exchange data between the SAP Web AS and a directory server using the LDAP protocol

© 2003 SAP AG. All rights reserved.

Test Your Knowledge

1. Which of the following statements with regard to Central User Administration are correct?
   Choose the correct answer(s).
   - ☐ A CUA can be implemented as of SAP R/3 4.0B.
   - ☐ B Only SAP R/3 systems can be linked to a CUA.
   - ☐ C CUA uses an ALE setup.
   - ☐ D The clients linked by CUA are called logical systems.
   - ☐ E The clients linked with the CUA are known as the "central system" and "child systems."

2. Which of the following statements about logical systems are correct?
   Choose the correct answer(s).
   - ☐ A Logical systems are defined cross-client.
   - ☐ B The IDs of logical systems can be assigned to clients.
   - ☐ C The naming convention for logical systems is LOGSYS###, where ### corresponds to a client ID.
   - ☐ D Multiple logical system IDs can be assigned to a client.
   - ☐ E In the context of a CUA, the clients involved are addressed as logical systems.

3. Which of the following statements about the required RFC connections in a CUA are correct?
   Choose the correct answer(s).
   - ☐ A The RFC connections must be specified entirely in uppercase letters and must be identical to the name of the addressed logical system.
   - ☐ B The user in the RFC connection must be a dialog user.
   - ☐ C An RFC connection from the child system to itself is required.
   - ☐ D An RFC connection from the central system "to itself" is required.
   - ☐ E Certain roles for the communication users are required only during CUA setup, and can then be removed from the relevant user master records.

© 2003 SAP AG. All rights reserved.

4. With which methods is it possible to create users in child systems in the context of an active CUA?

   Choose the correct answer(s).

   - ☐ A In the child system with transaction SU01, *Systems* tab page
   - ☐ B In the central system with transaction SU01, *Systems* tab page.
   - ☐ C In the central system with transaction PFCG
   - ☐ D In the child system with transaction PFCG
   - ☐ E In the central system with transaction SU01, *Roles* tab page

5. Directory services can only be used as a repository for information about users.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

6. Which prerequisites must be fulfilled on the SAP side to use an SAP Web AS with a directory server?

   Choose the correct answer(s).

   - ☐ A Host Name and Port Number of the directory server are maintained under *Connection Data*.
   - ☐ B In user maintenance (transaction SU01), there is a user of type *System*, that has read and write authorization for the directory server.
   - ☐ C The operating system program *ldap_rfc* is running on the host, on which the Directory Server is running.
   - ☐ D The names of the RFC destination and the associated Connector definition are identical.

7. During a synchronization process, data can be simultaneously imported and exported for a user.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

© 2003 SAP AG. All rights reserved.

319

Answers

1. Which of the following statements with regard to Central User Administration are correct?

   Answer: C, D, E

   The CUA clients linked by ALE are known as the "central system" and "child systems," and as "logical systems." You can connect SAP systems to a CUA if they have at least SAP Basis 4.5B or an SAP Web AS.

2. Which of the following statements about logical systems are correct?

   Answer: A, B, E

   The table that contains the IDs of the logical systems is a cross-client table. Each ID is assigned to exactly one client, and the recommendation for the naming convention is <SID>CLNT###. The clients involved in a CUA require a unique assignment to a logical system.

3. Which of the following statements about the required RFC connections in a CUA are correct?

   Answer: A, D, E

   The names assigned to the RFC connections used must be identical to the names of the logical systems, that is, also in uppercase letters. The user behind the RFC connection should be of the type *Communication*. The central system requires an RFC connection to itself. The communication users require a number of authorizations only to create the CUA; the corresponding roles can be removed from the user master records after the setting up of the CUA is completed.

4. With which methods is it possible to create users in child systems in the context of an active CUA?

   Answer: B, E

   If you are using a CUA, you can only create users in child systems using transaction SU01 in the central system. The creation of users is possible using the *Systems*, *Roles*, and *Profiles* tab pages.

© 2003 SAP AG. All rights reserved.

5. Directory services can only be used as a repository for information about users.

   Answer: False

   Thanks to flexible data model, data about any objects (such as hardware resources or documents) can be stored in directory services – although their use for user information is particularly widespread.

6. Which prerequisites must be fulfilled on the SAP side to use an SAP Web AS with a directory server?

   Answer: A, D

   In the SAP system, the RFC destination used and the LDAP Connector must be maintained with the same name (in uppercase letters). The operating system program *ldap_rfc* can then be started on the selected application server of the SAP system – while the directory server(s) addressed can run on remote servers (without *ldap_rfc*). Additional Customizing includes the definition of a system user and the connection data, including the host name and port number of the directory server.

   The LDAP system user is stored in Customizing for the directory service connection and has nothing to do with the SAP users maintained in transaction SU01.

7. During a synchronization process, data can be simultaneously imported and exported for a user.

   Answer: True

   The synchronization indicators do not need to be identical for all mappings. It is therefore entirely possible that, for example, SAP-specific data (such as role assignments) is exported during a synchronization process, while other data (such as a fax number) is imported.

 © 2003 SAP AG. All rights reserved.

Unit 6

321

Outlook: SAP Exchange Infrastructure (SAP XI)

The Exchange Infrastructure is another component of SAP NetWeaver that is based on the technology of the SAP Web Application Server. The Exchange Infrastructure should therefore be briefly introduced in this course. You should describe both possible uses and the architecture as well as highlighting administrative activities for running operation. The Exchange Infrastructure in not installed and set up on the available training systems. Therefore, no demonstrations are given in this unit.

Unit Overview

The SAP Exchange Infrastructure (XI) is the SAP platform for process integration; it is based on the exchange of XML messages. From a technical point of view, the Exchange Infrastructure is based on a Unicode SAP Web Application Server with SAP J2EE. However, it must be additionally installed. This unit provides an introduction to the architecture and possible uses of the Exchange Infrastructure and describes administrative activities.

Unit Objectives

After completing this unit, you will be able to:

- Outline the uses of the SAP Exchange Infrastructure
- Outline the structure of the SAP Exchange Infrastructure
- List simple administrative activities in the context of the SAP Exchange Infrastructure

Unit Contents

Lesson: SAP XI: Introduction.....382
Lesson: SAP XI: Structure and Administration.....389

© 2003 SAP AG. All rights reserved.

Lesson: SAP XI: Introduction

322 Lesson Duration: 15 minutes

Lesson Overview

This lesson provides an overview of the uses of the SAP Exchange Infrastructure.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the uses of the SAP Exchange Infrastructure

This lesson should provide the participants with an overview of the uses of the SAP Exchange Infrastructure. Describe the problems with system landscapes that have grown over the years. Explain the reasons for the development of the Exchange Infrastructure. Demonstrate the functions that the Exchange Infrastructure provides and discuss the use of open standards in its implementation.

Interested participants may ask about the future of the SAP Business Connector at this point. The SAP Business Connector is no longer developed (SAP Note 571530). For information about its positioning with regard to SAP XI, see SAP Note 507813.

Business Example

Your company has a system landscape that has grown over the years, with a large number of direct connection between the individual systems. As a member of the system administration team, you want to obtain an overview of the technology and uses of the SAP Exchange Infrastructure.

Introduction

Many customers have system landscapes that have grown over the years, as illustrated in the following graphic. This could be, for example, ERP systems such as SAP R/3 or technical systems for a specific purpose. Many components in these system landscapes are directly connected to each other, one to one. The integration information hardwired into the application components or in special middleware mapping programs. The high level of complexity of the integration knowledge and the individual connections require a great deal of administration and are associated with enormous costs. An upgrade, a change, or an extension of a system

© 2003 SAP AG. All rights reserved.

landscape that has grown in this way represents a significant challenge. The concept behind the SAP Exchange Infrastructure is to organize this integration knowledge.

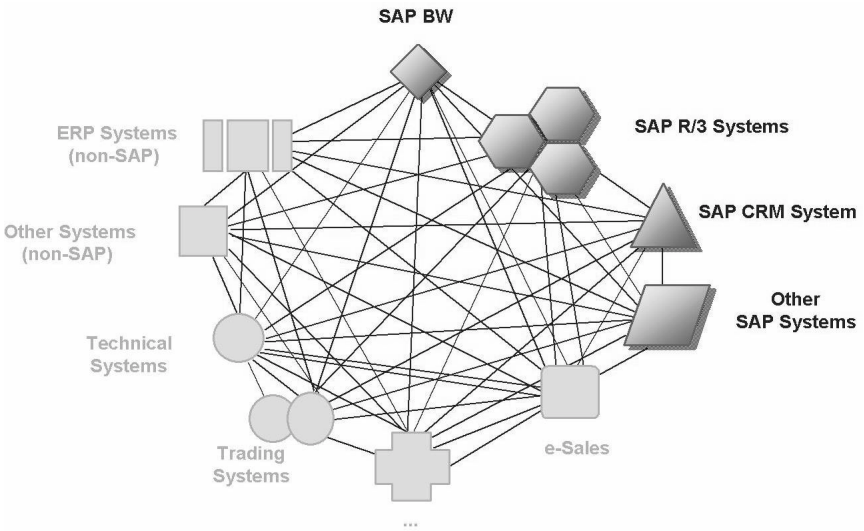


Figure 118: Example of a system landscape that has grown over the years

The development of the SAP Exchange Infrastructure is illustrated by the following graphic. With the SAP R/3 System, multiple application components access a shared database. Shared data structures are accessed across components in the database. The next step was the development of additional application components, such as SAP CRM. These systems had to be integrated into the environment of the existing system landscape. This was done using individual connections between the individual systems.

© 2003 SAP AG. All rights reserved.

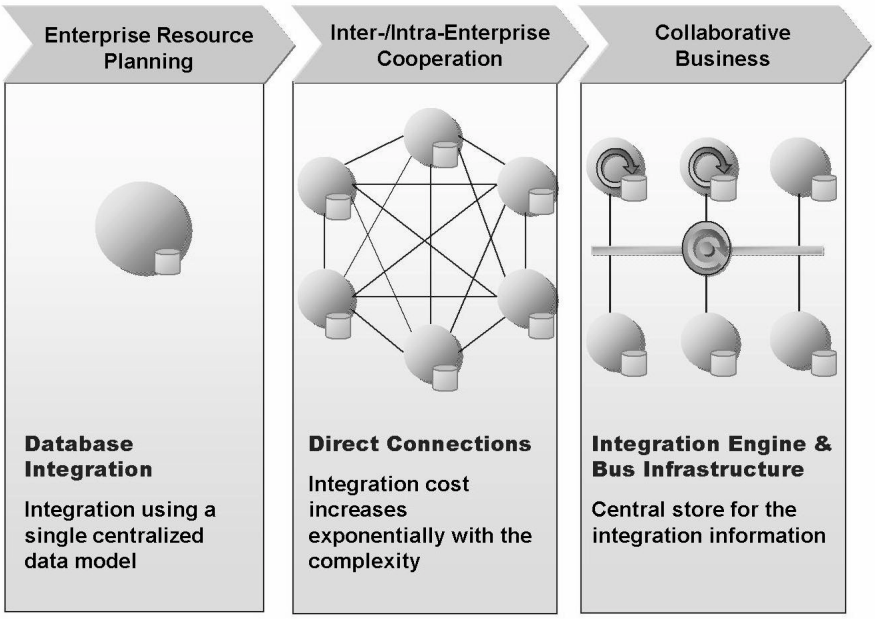


Figure 119: Development of the SAP Exchange Infrastructure

With the SAP Exchange Infrastructure, we consider the problem of the integration of systems from another point of view. The central idea behind it is to reduce the number of connections that must be maintained and to replace individual interface programs and the centralized storage and management of this integration knowledge. The knowledge about this integration; that is, the information about the connection of the individual applications and the message flow between the applications is stored in a central location.

In individual cases, it can still be necessary to create a direct connection between two SAP systems, even with SAP XI. This connection is required if the communication between these two systems generates an extremely high workload. This applies only to individual cases, however, and should not be emphasized in the course.

Uses

With the SAP Exchange Infrastructure, abbreviated to SAP XI, you can implement cross-system business processes. The aim of the SAP Exchange Infrastructure is to be able to connect different versions of systems from different vendors (SAP and non-SAP), implemented on different platforms (Java, ABAP, and so on). The Exchange Infrastructure is based on an open architecture, uses mainly open standards (especially from the XML and Java sphere) and provides services that are essential in a heterogeneous and complex system landscape: Runtime infrastructure for exchanging

© 2003 SAP AG. All rights reserved.

messages, configuration options for controlling the business process and the message flow, and options for transforming the message contents between sender and receiver.

Integration

The infrastructure is based, in general, on recognized standards to allow the integration of external systems. This means that the focus is on message-oriented communication using HTTP (HyperText Transfer Protocol): The application-specific content is transferred from the sender to the receiver through the Integration Engine using messages in a freely definable XML schema (XML: eXtensible Markup Language). The structure of a message is determined by the data structures used by an interface. A central concept here is that all required interfaces are initially developed in a platform-independent form in the design phase, and are made available as a Web Service Description Language (WSDL) description. Starting from this description, you can, for example, define mapping between interfaces, without which there would already be consequences for an existing system landscape. All data from the design phase is stored in the Integration Repository to be implemented in a concrete system landscape later.

In this second phase, the configuration period, you can select the components, interfaces, and mapping information stored there in accordance with your system landscape and your business process, and assign them to each other with logical routing. The result of this configuration process is stored in the Integration Directory and can be called and evaluated during the runtime of the SAP Exchange Infrastructure. The integration knowledge for a distributed process is also stored centrally: in the Integration Repository during the design period and in the Integration Directory during the configuration period. The SAP Exchange Infrastructure therefore follows the principle of Shared Collaboration Knowledge: You no longer need to search for information about a distributed process in the different systems involved, but can be called centrally. This procedure reduces the costs for development and maintenance of distributed applications.

Features

With the SAP Exchange Infrastructure, you can

- Use the System Landscape Directories to define your system landscape as the basis for the description of your cross-system business process
- Use Business Scenarios to describe the overall business process from a bird's eye perspective, and therefore derive the required interfaces and mapping.
- Use Routing Relations to maintain the message flow between the logical systems of your system landscape centrally (logical routing).

© 2003 SAP AG. All rights reserved.

- Use end points to link the logical recipient to a technical system, which can be more simply replaced due to this level of abstraction (technical routing)
- Control system access using logon data
- Use mapping values and structures to adjust your message for the recipient
- Develop cross-system applications that exchange messages with a large number of systems using synchronous or asynchronous communication through the runtime infrastructure. To do this, you can both develop platform independent interfaces and link existing interfaces at runtime using adapters.

Advantages of the XML Technology

- XML is the standardized exchange format in the Internet. Before this standard was designed, only more or less proprietary exchange formats existed that made communication in heterogeneous system landscapes significantly more complicated.
- Based on this, there are additional standards and tools that simplify working with XML. For example, you can use these tools to define mapping information that is required due to different structures in messages, and they allow the evaluation of conditions depending on values in the XML message. Evaluations of this type are required to determine the recipient with logical routing.

Examples of these tools are XML Schema, XSLT, and Xpath. You can use XSLT (eXtensible Stylesheet Language for Transformations) to define mapping information. Conditions are evaluated using XPath expressions.

- As a standardized format, XML also simplifies the connection of external systems. If you can convert data from the external system to XML using an adapter, the conversion to other XML formats and therefore for other recipients is only a small step.

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

Discuss the implementation of process integration based on the Exchange Infrastructure with the participants.

Discussion Questions

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

What advantages does the SAP Exchange Infrastructure provide for customers with regard to process integration?

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the uses of the SAP Exchange Infrastructure

Related Information

- On the SAP Service Marketplace, Quick Link */xi*: http://service.sap.com/xi

© 2003 SAP AG. All rights reserved.

Lesson: SAP XI: Structure and Administration

328 Lesson Duration: 20 minutes

Lesson Overview

This lesson provides an overview of the structure of the SAP Exchange Infrastructure and of administrative activities during its running operation.

Lesson Objectives

After completing this lesson, you will be able to:

- Outline the structure of the SAP Exchange Infrastructure
- List simple administrative activities in the context of the SAP Exchange Infrastructure

---

This lesson provides participants with an overview of the structure of the SAP Exchange Infrastructure and of administrative activities during its running operation.

Introduce the components of the SAP Exchange Infrastructure and describe them using the example of transporting a message from an SAP system to a third-party system using the Integration Server.

The part of this lesson about administrative activities is restricted to activities during the running operation. You should not discuss activities during the setup and configuration of the SAP Exchange Infrastructure. You are also unable to show any demonstrations on the training systems, as SAP XI is not set up and configured. Instead, use the slides to show which administrative activities are to be performed in the context of monitoring and archiving.

However, the Integration Engine is part of SAP Web AS 6.20. To demonstrate this, you can optionally give short demonstrations, such as the *Troubleshooting Methods* (or, similarly the roadmap of the *System Administration Assistant*, transaction SSAA) in transaction SXMB_ADMIN.

Business Example

Your company uses SAP XI to connect your SAP systems with each other and with third-party vendor systems. It is your task as a member of the system administration team to ensure the correct operation of the SAP Exchange Infrastructure.

© 2003 SAP AG. All rights reserved.

Architecture

We will first obtain an overview of the possibilities of the SAP Exchange Infrastructure (SAP XI), starting from the runtime components:

- The Integration Server is a central communication and "distribution machine" that processes XML messages. This includes services for determining recipients (logical and technical routing) and for transforming message content between sender and receiver (mapping). The Integration Engine can also receive messages from other runtime components of the SAP Exchange Infrastructure using communication services, and send messages to these using HTTP. You would therefore be unable to exchange messages using the SAP Exchange Infrastructure without this component.
- Adapters allow the connection of interfaces from external systems or from SAP systems with SAP Basis 4.6 (or earlier) to the Integration Engine. The interfaces involved in the integration scenario must be imported into a central Integration Repository.
- SAP has introduced a new programming model for new developments with SAP XI. This model allows you to define platform-independent interface descriptions in the central Integration Repository. Interface representatives (proxies) can then be generated for Java applications and for SAP systems from the descriptions. In this case, the XML messages are exchanged through the Integration Engine using the proxy framework.

The following figure shows again how systems in a heterogeneous system landscape can communicate with each other using the different runtime components (Integration Engine, adapters, and the proxy framework).

 © 2003 SAP AG. All rights reserved.

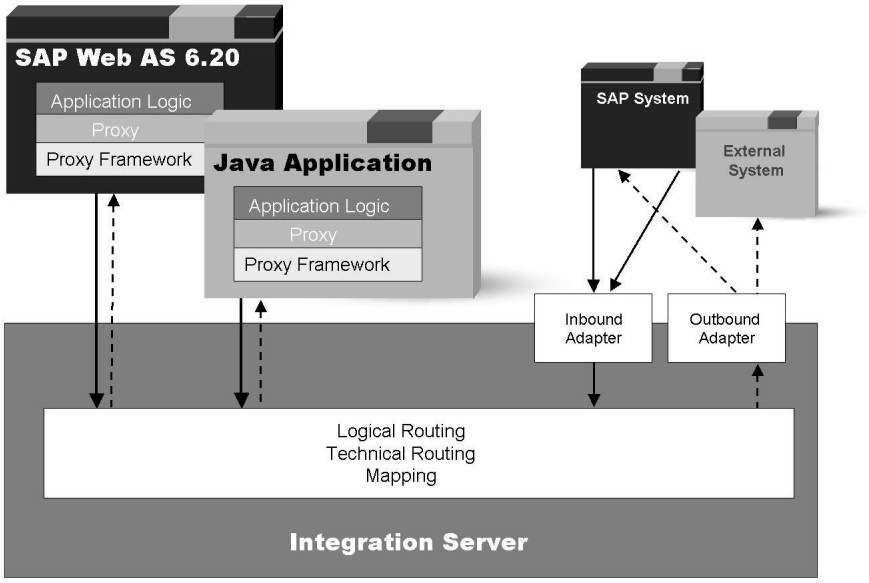


Figure 120: Systems Communicating

All systems send messages to and/or receive messages from the Integration Engine. In this way, messages can be exchanged between all systems in a heterogeneous system landscape, regardless of whether a message was sent to the Integration Engine using an adapter or the proxy framework.

Components of SAP XI

The main components of SAP XI, the Integration Repository, the Integration Directory, and the Integration Server, are illustrated in the following figure.

 © 2003 SAP AG. All rights reserved.

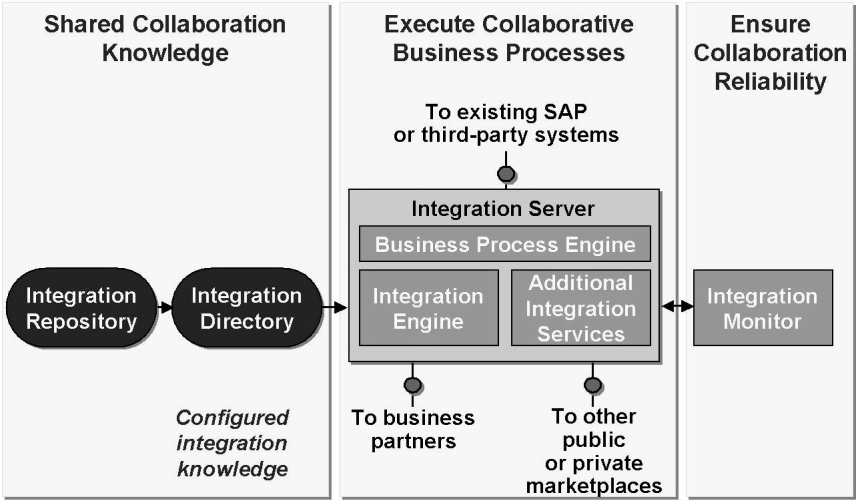


Figure 121: SAP Exchange Infrastructure

All objects that are relevant for integration are first stored platform-independently in the Integration Repository. It contains potentially usable objects that are not dependent on a specific system landscape. This integration information is stored by SAP during the design period, that is, during the development of new application functions. This can be descriptions of interfaces, mapping information, or a description of business processes.

The configuration of the current system landscape is described in the Integration Directory. The Integration Directory describes the actual selection of the objects available in the repository and their configuration for the specific landscape. It contains the description of the current interfaces, the current mappings, and the business processes actually used in the system landscape.

The processing of messages in the Integration Server is controlled using the information stored in the Integration Directory. The Integration Server represents the connection to SAP systems, systems from third-party vendors, business partner systems, or public or private marketplaces.

The integration processes and the distribution of messages between the systems are monitored using the Integration Monitor.

The following figure illustrates the flow of messages in the Integration Server.

 © 2003 SAP AG. All rights reserved.

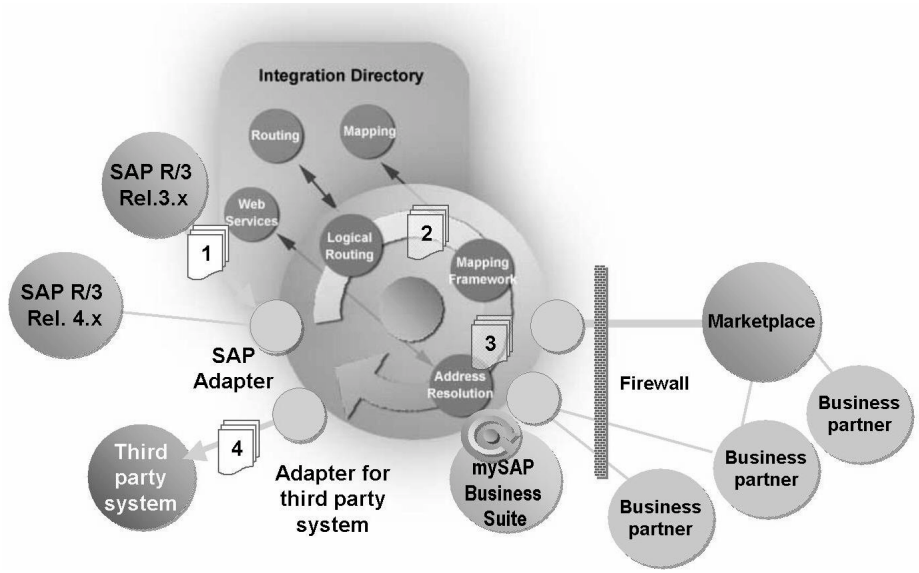


Figure 122: Flow of Messages in the Integration Server

This example shows the transport of a message between an SAP R/3 3.x system and a third-party vendor system. Messages are to be sent within this system landscape, starting from this SAP R/3 system. As a consequence of the decoupling of the individual systems, the sender system does not know where the information is actually required. As SAP R/3 3.x cannot send XML messages, but rather IDocs, this IDoc,must first be converted to an XML message using an IDoc adapter. The XML message is then transferred to the Integration Engine.

The additional processing of this message is performed in three steps. First, there is routing at a logical level; that is, this message is assigned a receiver system. The physical routing, during which the address of the receiver system is resolved and the protocol is determined is performed in the last step. The content of the message is mapped in a second step so that the receiver system can understand the message. The information required to perform this step is stored in the Integration Directory. After processing, the message can be sent to the receiver system directly or using an adapter.

Message

The message format of the SAP Exchange Infrastructure is based on XML. We will therefore often refer to XML messages, or simply to messages.

Message Structure

The following figure illustrates the structure of messages in the SAP Exchange Infrastructure.

 © 2003 SAP AG. All rights reserved.

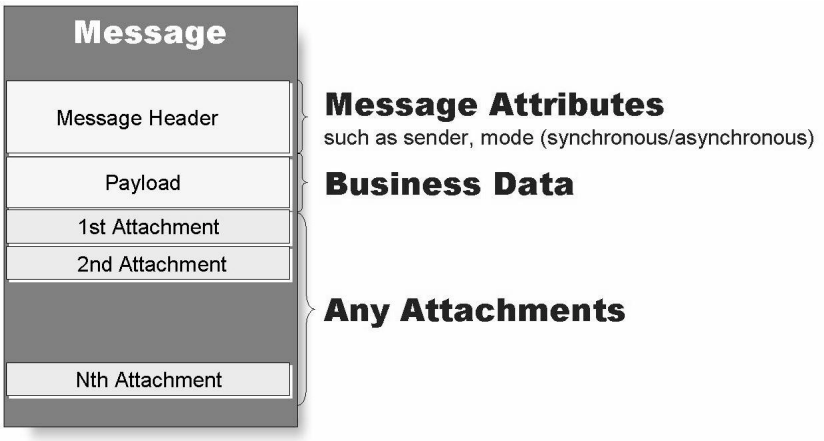


Figure 123: Message Structure

Here, the message header contains the properties of the current message (such as the sender, which SAP XI uses later to determine the recipient), while the actual business data is transferred with the payload.

```
<PurchaseOrder no="1811">
<ShipToParty>
<Name>Brian Adams</Name>
<Address>200 S Wacker Drive Chicago IL 60606</Address>
</ShipToParty>
<Item>Bass Guitar No.14 </Item>
<Status>Confirmed</Status>
</PurchaseOrder>
```

In addition, any number of attachments can be attached to the message before it is sent. Attachments are usually non-XML data, such as images, text documents, and binary files.

Interface-Based Processing of Messages

SAP XI messages are based on XML. How can an application send a message of this type? The concept is similar to that of Remote Function Calls: The communication with another system is encapsulated using an interface, the parameters of which are converted into a message. The following figure is a schematic illustration of how a message is sent to a recipient using a sender interface.

 © 2003 SAP AG. All rights reserved.

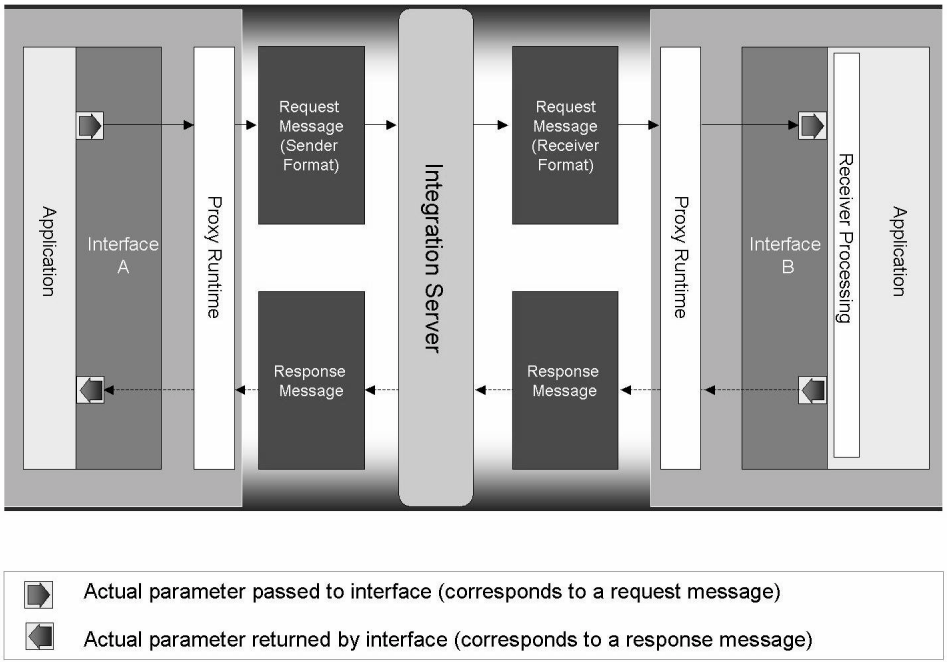


Figure 124: Sending Messages

Hint: There can also be more than one recipient; however, the principle remains the same.

The figure shows that for an interface A in a sender system, there is an assigned interface B in the receiver system.

- Interface B makes a service available that, for example, determines and returns customer data for an order.
- Interface A has the sole task of generating a message using the values transferred to the interface, which are to be sent to an assigned interface for a recipient using the Integration Server.

The application does not need to implement anything to convert the parameters to a message or the message to parameters. The proxy framework, which forwards and receives the message, performs this task. To send a message, simply call Interface A, to which you transfer parameters for the request message. You must also implement the recipient processing of Interface B on the receiver side.

Interface Types

The question is, of course, where the two interfaces come from. There are basically two types of interface:

 © 2003 SAP AG. All rights reserved.

- Message interfaces, which you initially create as a platform-independent XML -based description in an Integration Repository. You can generate interface representatives in one or more systems (on different platforms) from this description. The interfaces in the figure above were generated starting from message interfaces.
- Imported interfaces, which you import to the Integration Repository (IR), starting from interfaces in SAP systems. You can include imported interfaces in the SAP Exchange Infrastructure by using adapters. In this case, the adapter performs the role of the proxy framework, to generate the message and forward it to the Integration Engine. The figure "Sending Messages" shows communication between two IR-defined interfaces.

The differences between message interfaces and imported interfaces also results in a different programming model: For message interfaces, you use a platform-independent description to provide interfaces in a system; imported interfaces, however, access the system externally. Therefore, the first approach is called **outside-in** and the second approach is called **inside-out**. In both cases, the description of an interface in the Integration Repository is separated from the implementation, which has following advantages:

- Interface information is centrally accessible.
- You can use the interface description for application development in the systems and simultaneously for the design of business scenarios and mappings.

Outbound and Inbound

Interface A and interface B complement each other in that interface B receives the message that was sent by interface A. To emphasize this difference, interfaces that send messages are called outbound interfaces, and interfaces that receive messages and implement a service are called inbound interfaces. As indicated by the broken arrows in the figure "Sending Messages", communication can also be regarded as simply transferring messages, depending on whether communication is synchronous or asynchronous.

Outbound and inbound interfaces are also used to separate (potential) senders and receivers. The interface parameters do not need to match for a message exchange (a mapping is required at runtime). This enables various senders and receivers to communicate with each other. In particular, this loose link also enables you to assign interfaces of this type to each other, for which one side of the communication must not, or cannot be changed.

 © 2003 SAP AG. All rights reserved.

Administrative Activities

This section introduces administrative activities during the running operation of the SAP Exchange Infrastructure. These include monitoring the processing of XML messages and archiving XML messages.

You should not discuss administrative activities during the setup and configuration of the SAP Exchange Infrastructure in this lesson. These are:

- Installation
- Maintaining the system landscape in the System Landscape Directory
- Setting up RFC destinations

Optional: at this point, you can briefly demonstrate transaction SXMB_ADMIN and show , for example, the *Integration Engine* → *Monitoring* → *Troubleshooting Methods* area there.

Monitoring

The system administrator uses Monitoring to monitor his or her system. In addition to classic system monitoring in the Computing Center Management System (CCMS) (transaction RZ20), with the Exchange Infrastructure, the system administrator can monitor the processing of XML messages, tracking errors, and statuses. It is also possible to monitor message throughput.

Monitoring XML Messages

Monitoring comprises a monitor for processed XML messages. It identifies and evaluates all XML messages processed using the Integration Engine and produces processing statistics for them. The processed XML messages can be selected by certain criteria and displayed for evaluation purposes.

Monitoring also provides access to archived XML messages. These can also be collected using certain criteria and displayed for evaluation.

The status of an XML message is shown schematically in the following figure. When it is sent, an XML message initially has the status “being processed.” This is the current workload. After processing, the message is either “correctly processed”, “processed with errors”, or a “message branch” may have occurred. A message branch occurs if multiple recipients are determined for one message. If an error occurs, message processing is terminated, and manual processing is required.

© 2003 SAP AG. All rights reserved.

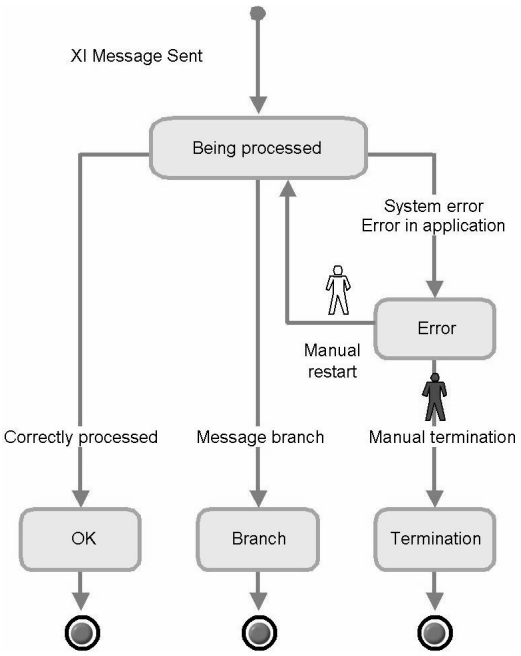


Figure 125: Life Cycle of a Message

The processing statistics for XML messages provide:

- Current information about the number and status of XML messages processed by the Integration Engine

  The status display for messages shows the following statuses for messages:

  - XML messages with errors
  - Not yet processed XML messages
  - XML messages being processed
  - Correctly processed XML messages
  - XML messages flagged for archiving
  - Archived XML messages
  - XML messages flagged for deletion
  - Deleted XML messages
- A performance analysis for message processing for each processing step of a pipeline

  In addition to the message status, number, and processing time, the processing statistics provide a range of current performance data for individual processing steps in a pipeline. These are, for example:

  - Name of the processing step concerned in the pipeline displayed
  - Shortest, average, and longest processing time for a message

© 2003 SAP AG. All rights reserved.

- Number of messages processed per second
- Throughput in kilobytes per second

Monitoring also provides:

- An overview about archiving and delete jobs used in the Integration Engine with corresponding information
- Analysis of the persistence layer
- Troubleshooting methods

Archiving

By default, correctly processed XML messages are deleted.

For clarity, we first need to differentiate between two types of delivering when sending XML messages:

- Synchronous sending of the message: The sender waits for a response before continuing processing.
- Asynchronous sending of the message: The sender does not wait for a response. The Integration Engine guarantees that the message is delivered and processed exactly once.

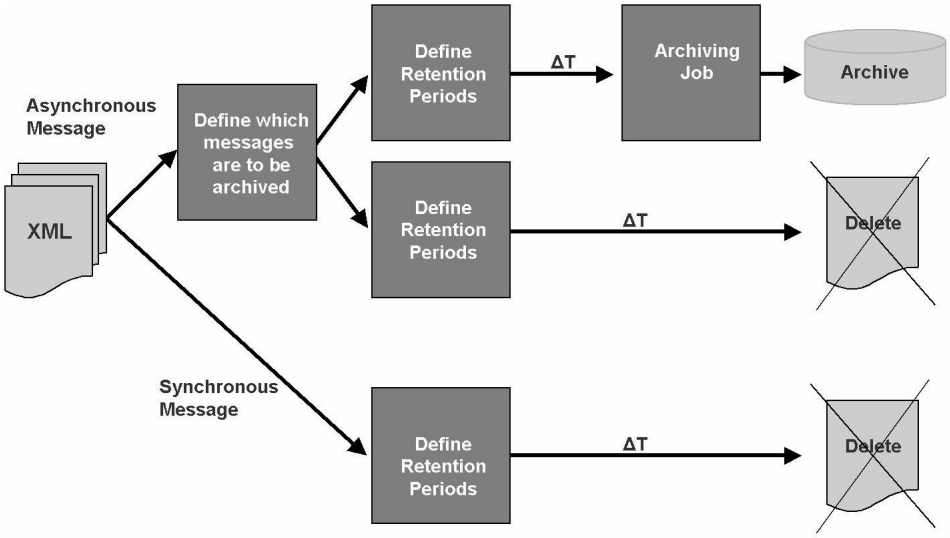


Figure 126: Archiving and Deleting XML Messages

Synchronous Messages

When sending synchronous messages, the sender system receives confirmation of receipt from the receiver system. After a defined retention period, the messages are then deleted from the database of the SAP Exchange Infrastructure. It is therefore not necessary to archive these synchronous messages.

© 2003 SAP AG. All rights reserved.

Asynchronous Messages

Asynchronous XML messages that have been correctly processed can either be archived or deleted. All XML messages that have not been deleted once they have been processed correctly must therefore be archived.

To do this, first define the interfaces for which XML messages are to be archived. You can also specify the retention period in the database for XML messages to be deleted or to be archived. The retention period defines a number of days for which the XML messages to be deleted or to be archived are to be retained in the database.

You must carry out multiple archiving jobs to archive XML messages:

- Determine messages to be archived
- Write messages to archive
- Delete archived messages from the database

To delete XML messages, perform the following two jobs:

- Determine messages to be deleted
- Delete archived messages from the database

You can schedule all jobs periodically, but you must maintain the job sequence. You can select and display archived XML messages.

To archive XML messages, use the archiving object BC_XMB, and use SAP_BC_XMB as the archiving infostructure.

© 2003 SAP AG. All rights reserved.

Facilitated Discussion

Discuss the structure of the SAP Exchange Infrastructure and which administrative activities are to be performed during operation.

Discussion Questions

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

List the central components of the SAP Exchange Infrastructure.

Outline the individual steps when transporting a message between two systems.

List and describe administrative activities that are to be performed when operating the SAP Exchange Infrastructure.

© 2003 SAP AG. All rights reserved.

Lesson Summary

You should now be able to:

- Outline the structure of the SAP Exchange Infrastructure
- List simple administrative activities in the context of the SAP Exchange Infrastructure

Related Information

- On the SAP Service Marketplace, Quick Link */xi*: http://service.sap.com/xi

© 2003 SAP AG. All rights reserved.

Unit Summary

You should now be able to:

- Outline the uses of the SAP Exchange Infrastructure
- Outline the structure of the SAP Exchange Infrastructure
- List simple administrative activities in the context of the SAP Exchange Infrastructure

Related Information

- SAP course BIT400: *SAP Exchange Infrastructure*
- SAP course BIT450: *SAP Exchange Infrastructure Development*

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

Test Your Knowledge

1. Which of the following statements about the SAP Exchange Infrastructure are correct?
   Choose the correct answer(s).
   - ☐ A The SAP Exchange Infrastructure maps the integration information about direct one to one connections between the individual systems.
   - ☐ B The SAP Exchange Infrastructure is the SAP platform for process integration, and is based on the exchange of XML messages.
   - ☐ C The SAP Exchange Infrastructure is based on an open architecture and primarily uses open standards.
   - ☐ D The connection of the individual applications and the message flow between the applications is stored on all systems involved.

2. Which components are part of the SAP Exchange Infrastructure?
   Choose the correct answer(s).
   - ☐ A Integration Repository
   - ☐ B Data Dictionary
   - ☐ C Integration Server
   - ☐ D Integration Manager

3. Which statuses does the status display for XML messages show?
   Choose the correct answer(s).
   - ☐ A Lost XML messages
   - ☐ B XML messages being processed
   - ☐ C XML messages with errors
   - ☐ D Deleted XML messages

© 2003 SAP AG. All rights reserved.

342

Answers

1. Which of the following statements about the SAP Exchange Infrastructure are correct?

   Answer: B, C

   The SAP Exchange Infrastructure, as the SAP platform for process integration, is based on an open architecture and the use of open standards. The integration knowledge is stored at a central location.

2. Which components are part of the SAP Exchange Infrastructure?

   Answer: A, C

   The main components of the SAP Exchange Infrastructure are the Integration Repository, the Integration Directory, and the Integration Server.

3. Which statuses does the status display for XML messages show?

   Answer: B, C, D

   You can track the status of XML messages in the context of Monitoring.

© 2003 SAP AG. All rights reserved.

Course Summary

You should now be able to:

- List and evaluate technology components of SAP systems in the Internet environment
- List the requirements for an SAP system used worldwide
- Explain the procedures for telephone integration and for sending mail from SAP systems
- Describe the eCATT procedure for automated tests
- Set up Central User Administration
- Connect an SAP Web Application Server to a directory service
- Explain the purpose of the SAP Exchange Infrastructure

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

Glossary

address version
Address dependent on a script (such as Russian or kanji)

AGate
Application Gateway, main processing component of the SAP ITS: The AGate connects the SAP ITS to the SAP system by receiving Web browser requests from the WGate (Web Gateway) through the Web server, and communicates with the application server using the DIAG or RFC protocol.

ALE
ALE stands for Application Link Enabling and describes a link between different software systems for exchanging data.

ALE integrated system
A group of systems linked using ALE is described as an ALE integrated system.

blended code page
Unambiguously mixed individual code page

CATT
SAP test tool. You can use the Computer Aided Test Tool (CATT) to combine business processes into repeatable test processes and automate these. CATT processes are used for the following tasks, among other things: Running transactions and transaction chains automatically, checking transaction results, checking system messages, and generating data.

central system
In the case of systems linked with ALE, you differentiate between the system in which the distribution model of the data is managed, and the linked systems. The managing system is called the **central system**, and the linked systems are called **child systems**.

code page
Character set with rules, such as converting from lower-case to upper-case letters.

communication type
A communication type is a type of communication supported by SAPconnect, such as sending by e-mail, sending as a fax, transfer to archiving systems, transfer to communication services that allow sending to pagers, or as SMS, and so on. Communication types are configured in SAPconnect.

© 2003 SAP AG. All rights reserved.

Computer Telephony Integration (CTI)
Software that functions as an interface between the telephone equipment and the computer. The interface is mapped by the SAP Remote Function Call (RFC) interface. SAPphone currently provides an extensive number of APIs that represent the central gateway between SAP applications and the CTI middleware products and telephony components.

control
Controls are independent binary software components that are reusable. Developers can integrate one or more controls into their applications and use the functions provided. A typical example is their use in structuring the user interface. However, the use of controls is not limited to visible components.

Control Framework
Basis technology based on ABAP Objects. The Control Framework is a prerequisite for displaying ActiveX controls and JavaBeans in the SAP system.

country version
Business function subject to legal requirements of an individual country; can be part of the standard SAP system or available as an add-on.

deployment
Installation process for libraries, service, interfaces, applications, and application components on the SAP J2EE Engine.

Directory server
A directory server is an information system for particular information that is stored in a hierarchical structure and that can be accessed using a standardized protocol (such as LDAP).

DIT
Directory Information Tree: Global tree of entries in a directory that can be physically distributed across multiple hosts.

DN
Distinguished Name: Name of an entry in a directory server that is unique directory-wide (For example: cn=LHeepmann, ou=instructors, o=sap, c=de).

eCATT
Extended Computer Aided Test Tool (eCATT) is used to create and perform function tests of software that is based on SAP technology. eCATT allows automated testing in SAP GUI for Windows and SAP GUI for Java. To provide a comprehensive test solution with other user interfaces, eCATT has interfaces for tools from third party vendors. Its primary purpose is to allow the automated testing of SAP business processes.

 © 2003 SAP AG. All rights reserved.

garbage collectors
Garbage collectors are Java's memory management. Unlike C and C++, there is no explicit instruction in Java for releasing memory that is no longer required. Memory management in Java is dynamic and automatic; that is, when a new object is created, the correct amount of memory is automatically assigned, and the memory is automatically released again when the object is no longer required. A concurrent thread has the task of finding objects that are no longer referenced and releasing their memory.

globalization
Collective term for various topics that are relevant for systems used internationally, such as country versions, time zones, languages, and code pages.

Greenwich Mean Time (GMT)
Greenwich Mean Time, uniform worldwide reference time zone, corresponds to Universal Time Coordinated (UTC).

ICF service
Links a certain URL (requested service of an SAP system with SAP Web AS) to an HTTP request handler of the ICF (development objects).

IDoc
An IDoc (intermediate document) is a data container with a particular format used for exchanging data between SAP systems, and also for exchanging data container data with non-SAP systems. These data packages are sent by RFC. IDocs are primarily used for exchanging data in an ALE integrated system.

Internet Communication Framework (ICF)
Environment for handling HTTP requests in work processes of an SAP system (in its role as a Web server and a Web client). The ICF is the bridge between the kernel of the SAP system and the application program written in ABAP. The ICF consists of ABAP classes and interfaces, the objects and methods of which can be accessed in a BSP application.

Internet Communication Manager (ICM)
Component of the SAP architecture as of SAP Web AS 6.10 that allows the SAP system to communicate directly with the Internet. Technically, the ICM is a standalone multi-threaded process that is started and monitored by the dispatcher.

Internet Server Cache (ISC)
Cache for response pages of the SAP Web AS. This stores pages before they are sent to the client. The next time that the relevant URL is called - as long as the expiry time has not elapsed - the page is sent back to the client directly from the ICM; in this case, it does not need to be branched to the task handler and the ICF.

© 2003 SAP AG. All rights reserved.

ITS Administration tool
Tool to administer, configure, and monitor the SAP ITS. The ITS Administration Tool provides an HTML-based interface and can therefore be called in a Web browser.

J2EE
J2EE is a standard defined by Sun Microsystems (SUN) for developing distributed applications based on Java.

JCo
Middleware component that allows the development of SAP-compatible components and applications in Java. The SAP Java Connector (JCo) supports communication with the SAP server in both directions: inbound (Java calls ABAP) and outbound (ABAP calls Java). The SAP JCo can be used both with desktop and with (Web) server applications.

language import
Importing language-dependent texts of an additional language into an SAP system.

LDAP
Lightweight Directory Access Protocol: Protocol for accessing directory servers (directory services).

LDAP Connector
Operating system program delivered by SAP, which implements the technical connection of an SAP system to a directory server.

locale
Set of rules for a national language.

mapping
Mapping describes how SAP fields are assigned to the attributes of a directory server.

Message Transfer Agent (MTA)
Program that is responsible for forwarding and delivering e-mails: After receiving a mail from a Mail User Agent (the actual e-mail program) or another MTA, the MTA analyzes the mail and either delivers it to the local user (or his or her Mail User Agent) or forwards it to another MTA. The MTA that is still used most frequently in the Internet is sendmail.

Multi Display Multi Processing (MDMP)
Parallel use of multiple code pages in an SAP system

mySAP Business Suite
mySAP Business Suite is a complete package of software and services that enables customers, partners, and employees to work together profitably cross-company - any time, anywhere.

 © 2003 SAP AG. All rights reserved.

RDN
Relative Distinguished Name: Components of the Distinguished Name in a directory server that are unique compared to the superordinate node (For example: cn=LHeepmann under ou=instructors, o=sap, c=de).

SAP Exchange Infrastructure
The SAP Exchange Infrastructure (XI) is the SAP platform for process integration, and is based on the exchange of XML messages.

SAP Internet Transaction Server (SAP ITS)
The SAP Internet Transaction Server (SAP ITS) forms a possible interface between the SAP system and the Internet. It allows users to communicate directly with the SAP system by starting business transactions, function modules, and reports from a Web browser. The SAP ITS consists of two main components – the WGate (Web Gateway to the Web server) and the AGate (Application Gateway to the application server).

SAP J2EE Engine
SAP delivers a J2EE application server, the SAP J2EE Engine, as standard with SAP Web AS 6.20.

SAP Web dispatcher
SAP solution for load distribution for HTTP(S) requests. If an SAP system consists of multiple instances, the SAP Web Dispatcher receives the requests from the browser and forwards them tot he application server that currently has most capacity. This simplifies administration, as there is only one entry point (IP address, HTTP port, and so on) to the SAP system.

SAPphone
Interface for telephone integration in the SAP system. SAPphone allows external vendors to connect their telephony software to the SAP system. It provides services with which telephone functions can be integrated into SAP applications.

Schema
Set of rules for a directory server, which defines the way in which entries are organized and which information is required or optional.

site
Sites represent a physical company location in SAPphone; that is, the area with shared area code and main number.

SMTP
Simple Mail Transfer Protocol (SMTP) is an Internet standard for a *host-to-host* mail transport protocol. SMTP uses TCP and port 25.

© 2003 SAP AG. All rights reserved.

standard system types
The three standard system types are: systems in which development and Customizing activities are performed, systems that are used for quality assurance, and systems used in production operation. These system types are often known as “DEV”, “QAS”, or “PRD” systems in the SAP environment

synchronization
During a synchronization (using the LDAP protocol), entries in a directory are compared with those in the connected system and synchronized in accordance with previously defined rules.

system data container
A system data container identifies instances of SAP systems. It can be maintained independently of the test script. As system data is an independent object, test cases can be changed efficiently if the system environment changes.

TAPI
Telephony Application Programming Interface: an interface for telephony applications defined by Microsoft.

telephony server
A telephony server includes all extensions that are connected to the SAP system using the same gateway (such as an instance of SAPphone Server). It is a logical entity that contains the settings required for a connection between the SAP system and the external telephony software.

test configuration
A test configuration contains references to a test script and a system data container. It can also contain references to multiple test data containers.

test data containers
A test data container for eCATT contains a parameter record that can be maintained independently of a test script. A parameter can be a simple ABAP type, a structure, or a table.

test script
An eCATT test script consists of an executable script and an interface for data transfer.

test script editor
The test script editor is used to create and maintain test scripts. You can either use special recording functions to do this, or enter eCATT commands directly in the command editor.

time zone
Time in relation to Greenwich Mean Time (GMT), with regard to an SAP system, a user, or an object.

© 2003 SAP AG. All rights reserved.

trust relationships
SAP systems can build trust relationships to each other. To do this, the calling SAP system must be registered in the called system as a trusted system. The called system is known as the trusting system. If a calling SAP system is known to the called system as a trusted system, no password is required to log on.

Unicode
Unique character set that is consistent worldwide, for displaying the characters of all languages that are relevant for business.

Universal Time Coordinated (UTC)
Universal Time Coordinated, uniform worldwide reference time zone, corresponds to Greenwich Mean Time (GMT).

UTF-16
Character representation for Unicode (compare with UTF-8)

UTF-8
Character representation for Unicode (compare with UTF-16)

WGate
Web Gateway, component of the SAP ITS: The WGate (Web Gateway) connects the AGate of the SAP ITS with the Web server. It receives requests from the Web browser through the Web server and forwards them to the AGate using TCP/IP. The WGate always runs on the same host as the Web server.

work center
A work center consists of a SAP front end (SAP GUI) and a telephone. There is a fixed or virtual connection between the computer and the telephone. In this case, virtual means that the connection is implemented indirectly using a telephony server and a telephone exchange. The connection between the computer and the telephone means that the screen and telephone work are integrated.

© 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

Index

A

address version, 131
AGate, 12
ASCII, 142

B

big endian, 149
blended code page, 139
Business Server Page, 6

C

CATT, 225
central system, 278
Central User Administration (CUA), 277
centralized implementation, 102
CGI, 59
character set, 140
child system, 278
code page, 139
communication method, 169
communication types, 168
Computer Telephony Integration (CTI), 206
connection data (for a directory server), 335
country version, 106

D

decentralized implementation, 102
deployment, 65
directory server, 315
DIT (Directory Information Tree), 319
DN (Distinguished Name), 319
double byte code pages, 142

E

EBCDI, 142
eCATT, 225

G

globalization, 101
Greenwich Mean Time (GMT), 108

I

ICF service, 47
IDoc (Intermediate Document), 279
IMAP, 183
inbound interface, 396
Integration Directory, 392
Integration Repository, 392
Internet Communication Framework (ICF), 46
Internet Communication Manager (ICM), 31
Internet Communication Server, 6
Internet Server Cache (ISC), 32
ITS Administration tool, 16

J

J2EE, 57
  Enterprise Java Beans (EJB), 59
  JCo, 63
  JSP, 59
  Servlet, 59
J2ME, 58
J2SE, 58
JVM, 58

© 2003 SAP AG. All rights reserved.

L

language import, 124
language supplementation, 128
LDAP Connector, 331
LDAP protocol, 317
little endian, 149
locale, 140
logical system, 279

M

mapping, 353
mapping indicators, 357
message
- asynchronous message, 400
- synchronous message, 399
- XML message, 393

Message Transfer Agents (MTA), 177
MIME, 184
Multi Display Multi Processing (MDMP), 139
multilingual capability, 118

O

outbound interface, 396

P

POP, 183
profile parameter
- exe/j2ee, 64
- icm/HTTP(S)/context_quota, 179
- icm/server_port_<xx>, 33
- icm/SMTP/context_quota, 179
- is/HTTP/default_root_hdl, 80
- ms/http_port, 82
- rdisp/j2ee_start, 64
- rdisp/mshost, 82
- rdisp/plugin_auto_logout, 49
- rdisp/start_icman, 33
- rdisp/tm_max_no, 179
- sapgui/user_scripting, 240
- SAPSYSTEM, 89
- zcsa/installed_languages, 128

R

RDN (Relative Distinguished Name), 319
Remote Function Call (RFC), 279
RMI, 58

S

SAP Exchange Infrastructure, 384
SAP Internet Transaction Server (SAP ITS), 12
SAP ITS, 5
SAP J2EE Engine, 7, 57
SAP Web dispatcher, 78
SAPconnect, 167
SAPphone, 206
schema extension, 353
SMTP, 175
SMTP plug-in, 167
synchronization, 362
synchronization indicators, 358
system data container, 228–229, 236
system user (of a directory server), 334

T

TAPI, 209
telephony server, 213
test configuration, 228, 230
test data container, 230
test data containers, 228, 256
test script, 228, 251
test script editor, 253
text comparison, 304
time zone, 108

© 2003 SAP AG. All rights reserved.

transaction code
BD54, 284
BD64, 306–307
LDAP, 332
LDAPLOG, 364, 374
PFCG, 285
RSLDAPSYNC_USER, 362
RZ11, 64, 240
SALE, 284
SBWP, 169
SCC4, 238, 284
SCOT, 168
SCUA, 290
SCUG, 293
SCUL, 306
SCUM, 290
SE63, 120
SECATT, 253
SICF, 47, 177, 188
SLG1, 364
SLG2, 365
SM59, 234, 288
SMICM, 34, 66
SMLT, 118
SPHA, 213
SPHB, 212
SPRO, 111
SPUMG, 152
ST11, 333
SU01, 303
SUCOMP, 293
SUMG, 152
WE20, 307
translation workbench, 120
trust relationships, 235

U

Unicode, 146
Unicode conversion, 150
Unicode migration, 150
Universal Time Coordinated (UTC), 108
UTF-16, 149
UTF-8, 149

W

WGate, 12
work center, 214
WSDL, 385

 © 2003 SAP AG. All rights reserved.

© 2003 SAP AG. All rights reserved.

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.

© 2003 SAP AG. All rights reserved.