

TADM10_1

SAP NetWeaver AS Implementation & Operation I

SAP NetWeaver

Date _____
Training Center _____
Instructors _____

Education Website _____

Participant Handbook

Course Version: 62
Course Duration: 10 Days
Material Number: 50089098



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

Copyright

Copyright © 2008 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 Server™ 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, SAPHIRE, 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 IMPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THESE MATERIALS AND THE SERVICE, INFORMATION, TEXT, GRAPHICS, LINKS, OR ANY OTHER MATERIALS AND PRODUCTS CONTAINED HEREIN. IN NO EVENT SHALL SAP BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND WHATSOEVER, INCLUDING WITHOUT LIMITATION LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS OR INCLUDED SOFTWARE COMPONENTS.

About This Handbook

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




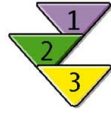

Typographic Conventions

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

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

Icons in Body Text

The following icons are used in this handbook.

Icon	Meaning
	For more information, tips, or background
	Note or further explanation of previous point
	Exception or caution
	Procedures
	Indicates that the item is displayed in the instructor's presentation.

Contents

Course Overview	viii
Course Goals	viii
Course Objectives	viii
Unit 1: The Big Picture	1
The SAP Software Portfolio	2
SAP Release Strategy	14
SAP NetWeaver	24
Unit 2: Navigation	35
Navigation in General	36
Advanced Navigation in the SAP GUI	52
Appendix - Personalizing the User Interface	67
Unit 3: The System Kernel.....	75
Principal Architecture of the SAP NetWeaver Application Server	77
AS ABAP Processes	97
Fundamental Concepts of Java.....	130
AS Java Processes.....	143
Java Cluster Architecture	151
The Internal Structure of SAP NetWeaver AS Java.....	158
Unit 4: Starting and Stopping SAP Systems	171
System Start: Process	173
System Start: Logs	183
System Shutdown: How and Why?	188
Appendix - Starting and Stopping: Operating System-Specific.....	195
Appendix - Database Logs	208
Overview of the Process for Starting and Stopping a SAP NetWeaver AS Java	215
Tools for Starting and Stopping	221
Java Startup and Control Framework.....	235
Logs of the Start and Stop Processes of SAP NetWeaver AS Java	244
Unit 5: Introduction to System Configuration	255
How the System Evaluates Its Parameters	257
How to Set System Parameters	266

Setting up Operation Modes	277
Administration and Configuration Tools of SAP NetWeaver AS Java	289
General Configuration of the SAP NetWeaver AS Java Cluster with the Config Tool	311
General Configuration of the SAP NetWeaver AS Java Cluster with the Visual Administrator	328
Other Administration Tools	347
Selected Configurations	362
Unit 6: Implementation of SSL for SAP NetWeaver AS Java	381
Setting up SSL	382
Unit 7: Software Development in SAP Systems	411
Data Structure of an SAP System and Transports between SAP Systems (ABAP Stack)	413
Accessing and Editing ABAP Repository Objects	423
Introduction to the SAP NetWeaver Development Infrastructure (NWDI)	446
Glossary	473
Index	483

Course Overview

Course TADM10 is the foundation for various, further training courses for consultants. After TADM10, you can continue your training to become a (Technical) XI or EP Consultant. Alternatively, you can proceed to course TADM12 where you will further expand your knowledge of SAP NetWeaver AS.

Courses TADM10 and TADM12 are **largely independent of the type of operating system and database technology**.

To pass the exam to become a certified **SAP Certified Technology Associate - System Administration - SAP NetWeaver 7.0 (<database>)**, you must have good knowledge of the content of courses TADM10 and TADM12, as well as one of the following database-specific courses: TADM51 (Oracle), TADM53 (MS SQL Server), TADM56 (DB2 on Win/UX) or ADM515 (MaxDB).

Like the other TADM courses, TADM10 comprises several individual courses (or parts thereof), which are arranged here in a way that will enable you to gain the knowledge you require as an SAP Technology Consultant as efficiently as possible.

Week 1 of course TADM10 is based on content taken from the following courses:

1. SAPTEC - Fundamentals of SAP NetWeaver Application Server
2. ADM100 - Administration AS ABAP I
3. ADM200 - Administration AS Java

Week 2 of course TADM10 is based on content taken from the following courses:

1. SAPTEC - Fundamentals of SAP NetWeaver Application Server
2. ADM100 - Administration AS ABAP I
3. ADM102 - Administration AS ABAP II
4. ADM200 - Administration AS Java
5. And some new course material, which is currently only covered in week 2 of TADM10.

At the end of the database-specific part of this training (TADM5#, or after ADM515, to be booked separately), there is a three-hour certification exam that covers topics from courses TADM10 and TADM12 and TADM5#/ADM515.



Caution: Note that the certification exam has been designed in such a way that the answers to all of the exam questions are contained in the folders provided for courses TADM10, TADM12, TADM5# (or ADM515).

Therefore, you do not require any additional course material even if the instructor hands out other books during the course or provides additional information not contained in the course folder.

Target Audience

This course is intended for the following audiences:

- SAP Technology Consultants (Associate Level)

Course Prerequisites

Required Knowledge

- Basic knowledge of IT
- Basic knowledge of operating systems and databases



Course Goals

This course will prepare you to:

- To work as an Associate Consultant (Junior Consultant) within the SAP technology environment
- To configure and manage SAP NetWeaver AS ABAP
- To configure and manage SAP NetWeaver AS Java



Course Objectives

After completing this course, you will be able to:

- To process basic tasks within the technology environment of SAP systems

Unit 1

The Big Picture

Unit Overview

This unit first provides an overview of the SAP solutions. You will then learn more about SAP's release strategy. Finally, we will talk about the application and integration platform SAP NetWeaver.



Unit Objectives

After completing this unit, you will be able to:

- List the key components of the SAP solutions
- Explain the differences between an application and a component
- Describe the SAP software delivery process
- Explain the SAP maintenance strategy
- Find out when SAP software releases will become available
- Explain the basic concept of SAP NetWeaver
- Describe the central role played by the SAP NetWeaver Application Server

Unit Contents

Lesson: The SAP Software Portfolio	2
Lesson: SAP Release Strategy	14
Exercise 1: SAP Release Strategy	21
Lesson: SAP NetWeaver	24

Lesson: The SAP Software Portfolio

Lesson Overview

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



Lesson Objectives

After completing this lesson, you will be able to:

- List the key components of the SAP solutions
- Explain the differences between an application and a component

Business Example

Your company plans to introduce SAP software. You, as a member of the project team, need an overview of the entire SAP portfolio.

Offers for Companies of All Sizes

SAP offers a vast range of products for companies of all sizes. SAP has scalable products, ensuring that they can be adjusted to any size and adaptable to the continuously changing processes of a company.

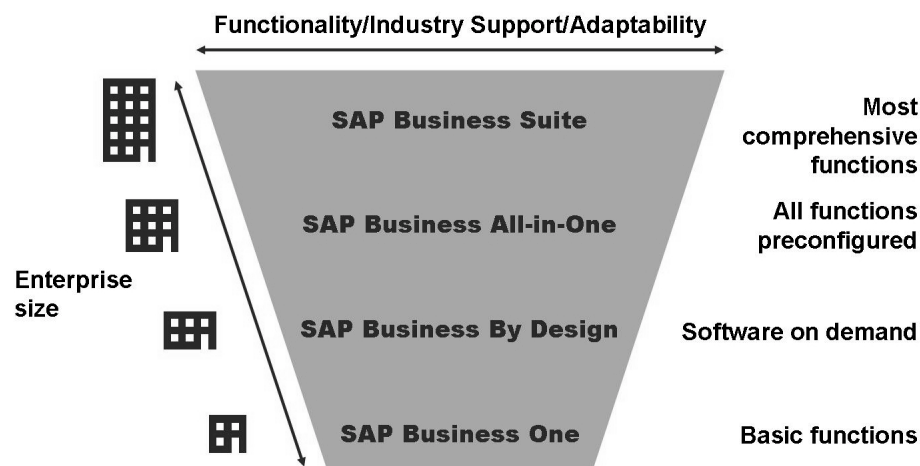


Figure 1: Different Sizes - Different Products

The following offers are available to small and mid size business customers:

SAP Business One

SAP Business One is a comprehensive, integrated ERP application with an interface similar to Windows, which has been developed specifically for small and midsize companies. It distinguishes itself by simple navigation and expansion options, innovative Drag&Relate functions as well as integration with Microsoft Word and Excel. The software enables you to uniformly manage the most important business areas such as sales, distribution and finances with one single, integrated application. This enables SAP Business One to provide an up-to-date overview of your company's most important data at any time. This information helps you meet customer requirements even quicker, an important prerequisite for profitable growth.

SAP Business One is the ideal solution for small companies with less than 100 employees and 30 users who are looking for an affordable system that covers their core processes (such as finance, sales, customer service and operations), thus ensuring successful and profitable operations. The solution can be used as a local software or hosted by a partner with an average implementation time of less than one month. SAP Business One is sold exclusively through a global partner network of currently 1,300 certified and experienced partners. They offer 24 hour support, which also covers the more than 350 enhancements from independent software providers. SAP Business One is currently being used by more than 14,500 customers in over 40 countries.



Note: For more information, go to <http://www.sap.com/solutions/sme/businessone>.

SAP Business ByDesign

SAP Business ByDesign is SAP's latest solution for small and midsize companies with between 100 and 500 employees, who want to use an on demand solution to improve their core processes and at the same time benefit from lower entry and operating costs. It is aimed, in particular, at midmarket companies who are not using integrated business software so far.

The solution combines the advantages of an integrated, end-to-end application with the low risk potential and the low total operating costs of an on demand solution. An on demand solution that distinguishes itself through integrated service and support, adaptability and more user friendliness and also guarantees transparency and the observation of legal requirements - all this on the basis of an Enterprise SOA by design. SAP Business ByDesign the most complete, fully integrated on demand solution. More than a pure niche solution, it supports the requirements of midsize companies in all areas such as effective management of finances, human resources, projects, production as well as supplier and customer relationship management as well as regional and local legal requirements and compliance management as core design principles. It was not developed with the traditional categories of business applications or customer relationship management and others. It was rather developed

for business processes in the entire company with user interfaces that are customized to the user roles. SAP Business ByDesign offers end-to-end processes for the work center of the future.



Note: For more information, go to <http://www.sap.com/solutions/sme/businessbydesign>.

SAP Business All-in-One

All software has to be adapted to the specific company. A midsize company with few employees and comparatively stable processes can use preconfigured SAP systems. SAP All-in-One is the brand name for vertical (industry-specific or country-specific) applications that are based on the technology of SAP NetWeaver and were developed in conjunction with SAP's partners.

SAP Business All-in-One is the ideal solution for small and midsize companies with very industry-specific requirements who have several divisions and a mature IT infrastructure. The solution can be used as a local software or hosted by a partner and is based on SAP ERP. It provides an industry-specific function portfolio based on SAP Best Practices. It is made available through 660 midmarket solutions by more than 1,000 partners. SAP Business All-in-One is affordable and can be planned and thanks to preconfigured business scenarios and implementation accelerators it can be implemented swiftly (4 to 16 weeks). SAP Business All-in-One is currently being used by more than 10,000 customers in over 50 countries.



Note: For more information, go to <http://www.sap.com/solutions/sme/businessallinone>.

SAP Business Suite in Detail

The SAP Business Suite is an extended family of business applications that enables companies to manage their entire value chains. The included business applications provide users with consistent results throughout the entire company network and give your company the flexibility it needs in today's dynamic market situations. The application consists of a number of different products that support cross-company processes.



Note: For more information, go to <http://www.sap.com/solutions/business-suite>.

SAP's software portfolio is constantly extended, optimized, and tailored to the needs of the market and of customers. SAP has therefore, for example, developed many industry-specific applications over the years. The topic of integrating different

business systems (even from different vendors and across company boundaries) has become increasingly important recently. Products for small to midsize businesses have also been added to SAP's family of software solutions.

The **SAP Business Suite**, a complete business software package that is unique worldwide, plays a central role. Many companies are already profiting from the comprehensive and flexible business applications with highly evolved functions: complete integration, industry-typical functions, unlimited scalability and smooth collaboration via the internet.

The SAP Business Suite provides:



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

The next graphic illustrates the key components of the SAP Business Suite:

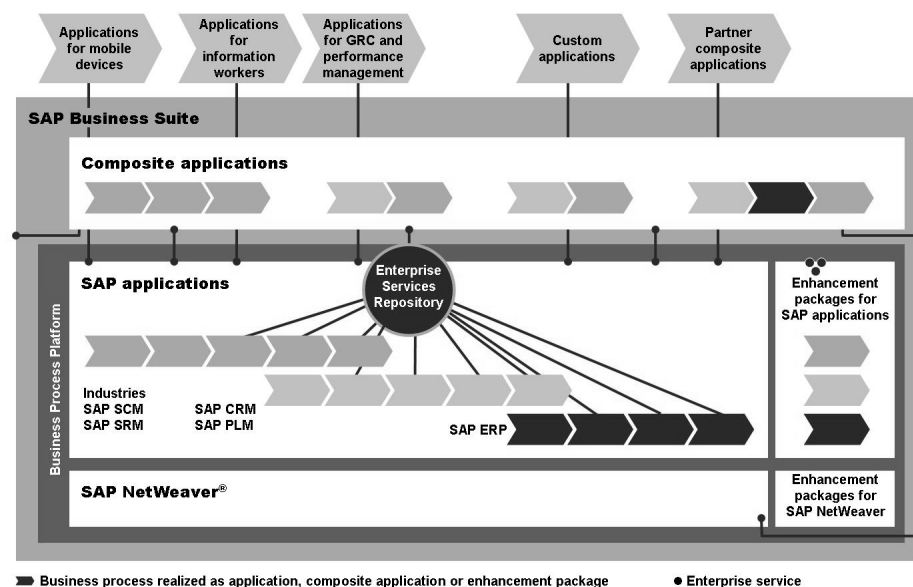


Figure 2: SAP Business Suite: Architecture

Key components of the SAP Business Suite are:

Business Process Platform (BPP)

A Business Process Platform is the prerequisite for the deployment of a service-oriented architecture for business applications (Enterprise Service-Oriented Architecture, Enterprise SOA). The SAP BPP is composed of the following modules:

SAP NetWeaver

SAP NetWeaver is an open integration and application platform for all SAP applications and certain SAP partner applications. SAP NetWeaver is a web-based platform, which serves as the basis for the Enterprise SOA and enables the cross-company and technology-independent integration and coordination of employees, information and business processes. Thanks to open standards, information and applications from any source and based on any technology can be integrated. SAP NetWeaver is interoperable with the most important technological standards such as the Java 2 Platform, Java Enterprise Edition (Java EE) and Microsoft .NET. The platform provides a maximum of reliability, security and scalability ensuring that business processes that are critical for success run without problems.

SAP applications

The SAP Business Suite includes extensive, integrated applications, which customers can license completely or individually. More recent SAP applications use the **Enterprise Service Repository**, repository for processes and Enterprise Services. Enterprise Services map business processes using open standards.

Both SAP NetWeaver and the SAP applications are continuously enhanced through **Enhancement Packages**. Enhancement packages are shipped at regular intervals in order to make it possible to respond to market requirements quicker and to provide additional benefits to customers and partners without the need to wait for the next major product release.

Composite Applications (xApps)

SAP xApps are a portfolio of “Packaged Composite Applications”, which you can use to promote improvements and innovation in your company more easily. With their ability to combine existing, heterogeneous systems to form cross-functional processes, SAP xApps Composite Applications bring people, information, and business processes together to make your company more dynamic and competitive. This flexibility enables you to implement business-wide strategies more easily and efficiently. SAP xApps increase the value of existing investments in the core business area and maximize the return on strategic investments: employees, knowledge, products, business relationships, and information technology.

You can use the SAP Composite Application Framework (CAF) to develop and run your own xApps.

The following graphic provides an overview of the applications included in the SAP Business Suite.

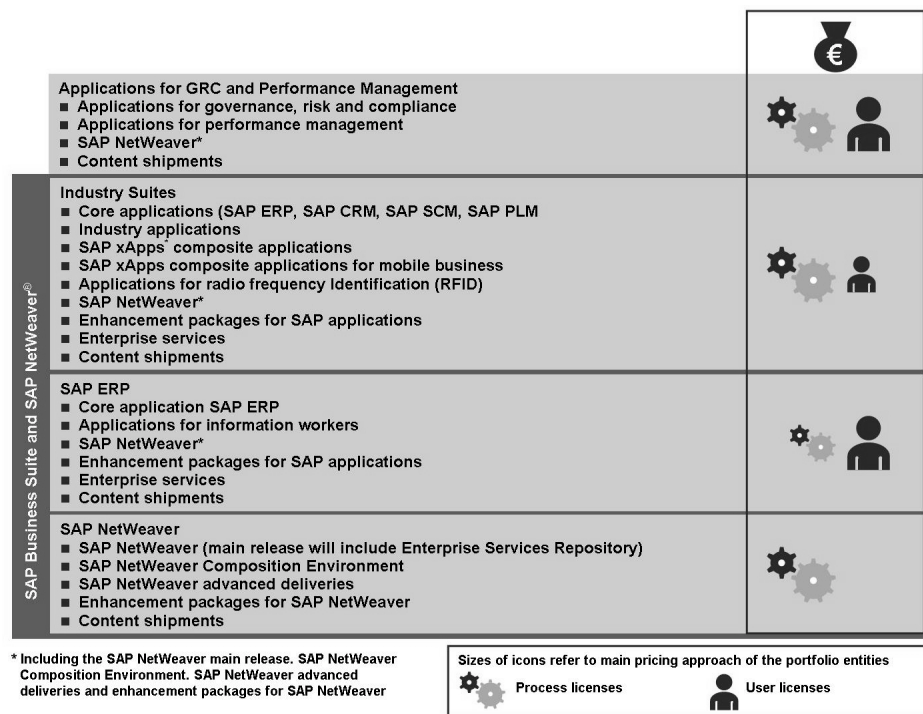


Figure 3: SAP Business Suite: Included Applications

Some of the applications introduced in the graphic are:

- **Applications for Governance, Risk, and Compliance (GRC):** Applications for governance (SAP GRC Repository), risk management (SAP GRC Risk Management) and compliance with legal requirements (SAP GRC Access Control, SAP GRC Process Control, SAP GRC Global Trade Services, SAP applications for environment, health, and safety compliance management, GRC composite applications by SAP and Cisco); for more information see <http://www.sap.com/solutions/grc>
- **Applications for Performance Management:** Applications for performance management (SAP Strategy Management, SAP Business Planning and Consolidation, SAP Business Profitability Management by Acorn, SAP xApp Analytics); for more information see <http://www.sap.com/solutions/performancemanagement>
- **SAP Enterprise Resource Planning (SAP ERP):** Applications for supporting resource planning across an entire enterprise
 - SAP ERP Human Capital Management
 - SAP ERP Financials
 - SAP ERP Operations
 - ERP Corporate Services

For more information see <http://www.sap.com/solutions/business-suite/erp>

- **SAP Customer Relationship Management (SAP CRM):** Applications for end-to-end, customer-oriented business processes; for more information see <http://www.sap.com/solutions/business-suite/crm>
- **SAP Supplier Relationship Management (SAP SRM):** SAP SRM integrates strategic purchasing processes such as vendor qualification and vendor management, electronic invitations for tender and auctions or contract management with the other core processes in your company and with the processes of your suppliers; for more information see <http://www.sap.com/solutions/business-suite/srm>
- **SAP Supply Chain Management (SAP SCM):** Applications for managing the entire logistical chain to synchronize supply and demand across company borders; for more information see <http://www.sap.com/solutions/business-suite/scm>
- **SAP Product Lifecycle Management (SAP PLM):** Applications for including internal and external partners directly in the operational product development and production processes; for more information see <http://www.sap.com/solutions/business-suite/plm>
- **Industry Applications:** Industry-specific applications
 - Banks

- Healthcare
- Universities and research institutes
- Internal & external security
- Public sector
- Insurance
- Service industry
- Wholesale
- Retail
- Logistics service providers
- Media industry
- Telecommunications
- Utilities industry
- Automobile industry
- Construction industry, engineering, construction and operations
- High tech and electronics industry
- Consumer products industry
- Aerospace and space industry
- Mechanical engineering
- Metal, wood and paper industry
- Mining
- Chemicals
- Life sciences
- Oil and gas industry

For more information see <http://www.sap.com/industries>

- **SAP xApps Composite Applications:** Composite applications that cover several company solutions across persons, information and processes
 - SAP xApp Cost and Quotation Management (SAP xCQM)
 - SAP xApp Integrated Exploration and Production (SAP xIEP)
 - SAP xApp Lean Planning and Operations (SAP xLPO)
 - SAP xApp Manufacturing Integration and Intelligence (SAP xMII)
 - SAP xApp Product Definition (SAP xPD)
 - SAP xApp Resource and Portfolio Management (SAP xRPM)
 - SAP xApp Sales and Operations Planning (SAP xSOP)
 - SAP xApp Spend Analytics (SAP xSA)

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

- **Applications for Information Workers:** Applications granting employees access to company data and business processes anytime and anywhere
 - Duet for Microsoft Office and SAP
 - SAP Interactive Forms by Adobe
 - SAP NetWeaver Enterprise Search

For more information see <http://www.sap.com/solutions/informationworkers>

- **SAP NetWeaver:** Open integration and application platform for all SAP applications
 - SAP NetWeaver Application Server
 - SAP NetWeaver Business Intelligence
 - SAP NetWeaver Exchange Infrastructure
 - SAP NetWeaver Master Data Management
 - SAP NetWeaver Mobile
 - SAP NetWeaver Portal
 - SAP Auto-ID Infrastructure
 - SAP NetWeaver Identity Management

For more information see <http://www.sap.com/platform/netweaver>

Context of Applications and Components

Numerous applications for business challenges are provided in the context of the SAP Business Suite. However, many applications have similar or identical requirements for business functions in subareas. Different applications therefore contain similar (software) components in parts. A component is the smallest, separately producible, deliverable, installable, and maintainable software unit.

Components refer to, for example, an SAP ECC system, an SAP SCM system or also an SAP NetWeaver portal system. The graphic provides an overview of this hierarchy (components as building blocks of solutions) using the SAP SCM application as the example.

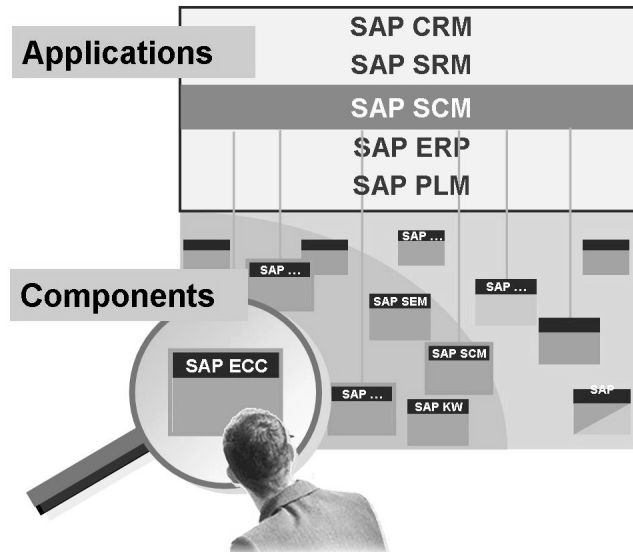


Figure 4: Applications and Components

Possibilities with SAP Business Suite

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

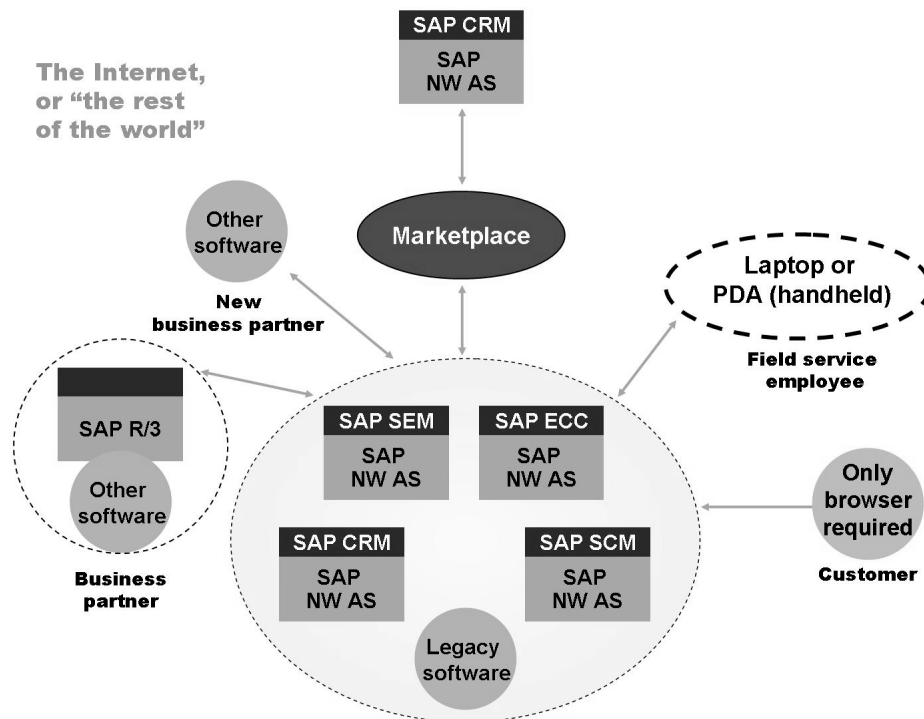


Figure 5: Possible Business Scenario with SAP Business Suite

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



Lesson Summary

You should now be able to:

- List the key components of the SAP solutions
- Explain the differences between an application and a component

Related Information

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

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

Lesson: SAP Release Strategy

Lesson Overview

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



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the SAP software delivery process
- Explain the SAP maintenance strategy
- Find out when SAP software releases will become available

Business Example

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

History of Selected Software Components

This section provides a bit of history of a few selected software components.

Technical Basis (Application Server)

Back in the days when SAP basically offered two products (SAP R/2 and SAP R/3), the development of the (technical) basis was closely linked to application development. The release names of the **SAP Basis** corresponded to the SAP R/3 version, for example, SAP Basis 4.0B was the technical basis for SAP R/3 4.0B

Around the turn of the millennium, the SAP portfolio grew significantly, new products were created that required more frequent changes and enhancements of the SAP Basis than SAP R/3. This marks the transition from the classical SAP Basis (last version: SAP Basis 4.6D) to **SAP Web Application Server** (SAP Web AS). New internet technologies (Internet Communication Manager from SAP Web AS 6.10 onwards) and the supplementing of the classical ABAP environment with Java/JEE (from SAP Web AS 6.20 onwards) were important milestones.

SAP Web AS 6.40 forms the technical basis (“application platform”) of SAP NetWeaver 04. SAP NetWeaver offers extensive capabilities (such as Business Intelligence), which are all based on the application platform. From SAP NetWeaver

7.0 (previously: SAP NetWeaver 2004s) the names and releases were adapted further, so now SAP NetWeaver 7.0 is based on **SAP NetWeaver Application Server 7.00** (SAP NetWeaver AS).

Central ERP Functions

The following graphic shows the historical development for the current SAP ERP Central Component (ECC 6.0):



Part of SAP NetWeaver	Basis Functionality (SAP Basis/SAP Web AS/ SAP NetWeaver AS)	Business Functionality (-/Core)	Business Extension Set	Part of SAP ERP
7.1	7.10	-	-	-
7.0 (2004s) 2004 ('04)	7.00 6.40	6.0 5.0	6.00 5.00	6.0 (2005) 2004
(2003)	6.30 6.20 6.10	4.7 4.7 -	2.00 1.10	(2003) (2003)
	4.6D 4.6C 4.6B 4.5B 4.0B 3.1I	- 4.6C 4.6B 4.5B 4.0B 3.1I		

Figure 6: Evolution from SAP R/3 via SAP R/3 Enterprise to SAP ECC

As already mentioned, in times of **SAP R/3**, the technical basis and application development were interlinked, up to and including SAP R/3 4.6C.

With **SAP R/3 Enterprise** (4.7), which is based on SAP Web AS 6.20, the concept of SAP R/3 Enterprise Extensions was introduced.

A central application (previously: solution) of the SAP Business Suite is SAP ERP for Enterprise Resource Planning. The central software component of SAP ERP is **SAP ERP Central Component** (SAP ECC). SAP ECC 5.0 can thus be considered the technical successor of SAP R/3 Enterprise and is based on an SAP Web AS 6.40. At the time of creating this documentation, the current version is SAP ERP 6.0 (previously: SAP ERP 2005), which also includes an SAP ECC 6.0 and other components, which operates on the basis of SAP NetWeaver AS 7.00. Functional enhancements for the different software components are made available through enhancement packages.

SAP NW AS 7.10 or SAP NetWeaver 7.1 are currently not used as the technical basis for an SAP ECC system. Other SAP NetWeaver components, such as SAP NW Process Integration (PI) and SAP NW Composition Environment (CE) already require this SAP NetWeaver release level.

Availability and Delivery

SAP applications, SAP NetWeaver, Enhancement Packages and xApps are offered to customers in two phases.

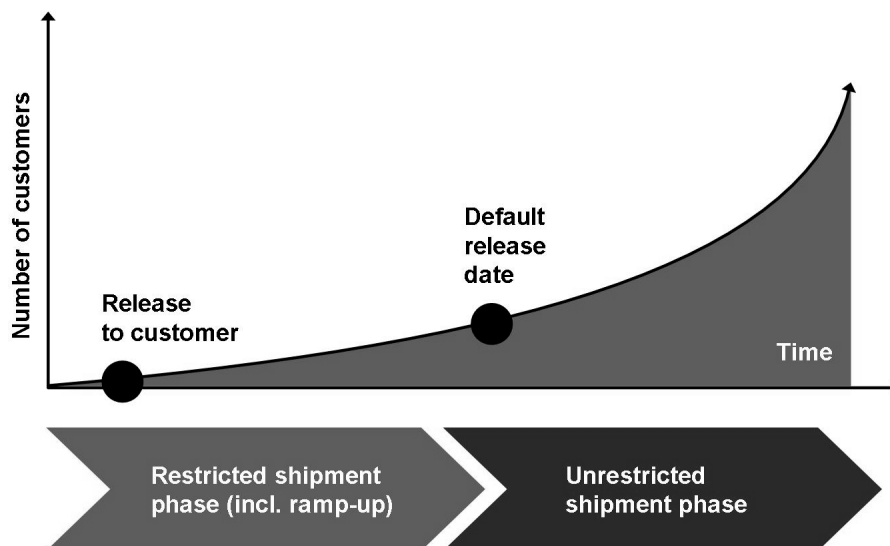


Figure 7: Phases of Software Delivery by SAP

Restricted Shipment Phase

New software releases are only available to a restricted number of customers whose increase is controlled. The *Release to Customer* date marks the start of the *Restricted Shipment Phase*.

During the restricted shipment phase, the new software release passes through a standardized process, the *Ramp-Up*. Characteristics of the ramp-up process are:

- Ramp-up is the name for the market introduction process of new products or new releases of SAP software.
- The software can be used for a productive implementation.
- During the ramp-up, the software is only available for a limited circle of customers, customers “apply” to SAP to participate in a ramp-up.
- The ramp-up serves to take customer wishes into consideration before the respective SAP software is generally available.
- Participants in the ramp-up are given direct contact to the SAP development and are provided with wide support in implementing the SAP software.
- A ramp-up usually takes about 6 months.



Note: For more information about the ramp-up, see <http://service.sap.com/rampup>.

Unrestricted Shipment Phase

The second phase of the delivery is called *unrestricted shipment phase*. After the *Default Release Date* all customers can request the new software without restrictions.

The respective latest, generally available version of SAP software is called *Default Release*. When SAP customers order software from SAP, they are usually given the current default release.

Maintenance

Your maintenance contract includes Support Packages, which are provided by SAP via the SAP Service Marketplace. Support Packages solve (component-dependent) problems that can result when using the SAP software. SAP provides these Support Packages at certain intervals.

5-1-2 Maintenance Strategy

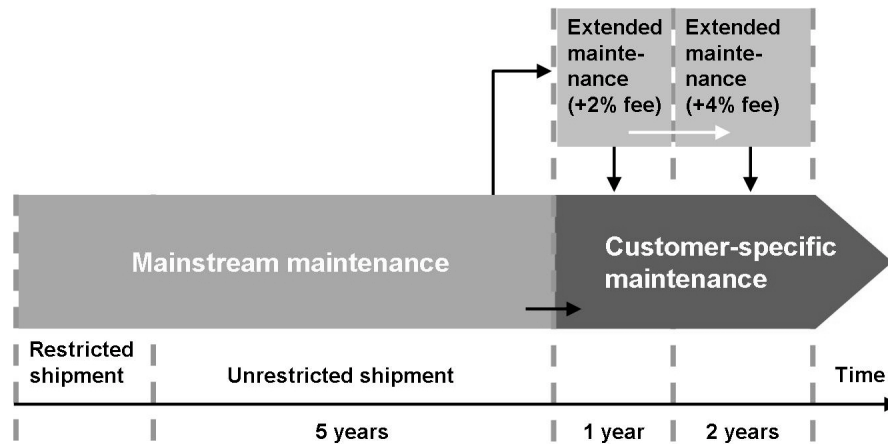


Figure 8: 5-1-2 Maintenance Strategy

Since 2004, SAP has been offering a new maintenance strategy for all products that are based on SAP NetWeaver. This 5-1-2 maintenance strategy offers maintenance for five years for the mainstream maintenance, followed by one year extended maintenance for a maintenance fee of two percent more and, optionally, another two years for a maintenance fee which is four percent more than the standard maintenance fee. Software installed at customer locations can transit into the phase of customer-specific maintenance.

We distinguish the following three phases of maintenance:

Mainstream maintenance

Mainstream maintenance starts from the release to customer point in time and usually ends 5 years after the default release date.

Extended maintenance

Customers can extend the support period (in which corrections or legal changes are taken into account) for an increased maintenance fee (+2% for 1 year, +4% for 2 years).

Customer-specific maintenance

From a certain point in time during the lifecycle of a release, this release is no longer adjusted to meet external requirements. Such requirements are, for example, the implementation of legal changes or the support of new technologies. The time in question is marked by the end of extended maintenance or the end of mainstream maintenance, if no extended maintenance is offered. Customer-specific maintenance starts at this point in time.



Note: See <http://service.sap.com/maintenance> for additional information on the maintenance strategy. SAP note 52505 deals specifically with the particularities of customer-specific maintenance.

Upgrade

You should think about upgrading to the latest release when your SAP software is approaching the end of its mainstream maintenance, at the latest. During an upgrade of your SAP software, you can generally skip one or more of the software releases (of your application) that have appeared in the meantime. For example, an upgrade from SAP R/3 4.6C to SAP ERP 6.0 is technically possible, even though you change product during this upgrade (from SAP R/3 to SAP ERP).

SAP offers upgrade paths for SAP applications, SAP NetWeaver, Enhancement Packages, xApps and Content Shipments. There might be cases in which an upgrade requires several steps (especially due to technical restrictions).



Note: For more information on upgrade, see <http://service.sap.com/upgrade>.

Exercise 1: SAP Release Strategy

Exercise Objectives

After completing this exercise, you will be able to:

- Name some information sources of the SAP Service Marketplace

Business Example

You want to determine the standard maintenance period for the SAP ERP 6.0 application and the point in time at which the next Support Package Stack will be available.

Task: Maintenance Information

Determine some maintenance information via the SAP Service Marketplace.

You use the SAP Service Marketplace to find out the end of the standard maintenance for the SAP ERP 6.0 application and to find out when the next Support Package Stack for the applications in your SAP ERP 6.0 system will be released.

1. When does the mainstream maintenance for the SAP ERP 6.0 application end?
2. When is the next Support Package Stack for SAP NetWeaver 7.0 scheduled?

Solution 1: SAP Release Strategy

Task: Maintenance Information

Determine some maintenance information via the SAP Service Marketplace.

You use the SAP Service Marketplace to find out the end of the standard maintenance for the SAP ERP 6.0 application and to find out when the next Support Package Stack for the applications in your SAP ERP 6.0 system will be released.

1. When does the mainstream maintenance for the SAP ERP 6.0 application end?

- a) Look at these addresses

- <http://service.sap.com/releasestrategy>
- <http://service.sap.com/pam>

to answer the question: March 2013.

2. When is the next Support Package Stack for SAP NetWeaver 7.0 scheduled?

- a) Look at this address

- <http://service.sap.com/sp-stacks>

to answer the question: Choose the *SP Stack Calendar* link at the bottom of the page and determine the planned calendar week.



Lesson Summary

You should now be able to:

- Describe the SAP software delivery process
- Explain the SAP maintenance strategy
- Find out when SAP software releases will become available

Related Information

- <http://www.sap.com/solutions>
- <http://service.sap.com/releasestrategy>
- <http://service.sap.com/rampup>
- <http://service.sap.com/maintenance>
- <http://service.sap.com/upgrade>

Lesson: SAP NetWeaver

Lesson Overview

This lesson explains the central role of the SAP NetWeaver Application Server in SAP NetWeaver.



Lesson Objectives

After completing this lesson, you will be able to:

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

Business Example

You want to find out more about SAP NetWeaver and the role that the SAP NetWeaver Application Server (SAP NetWeaver AS) plays in SAP NetWeaver.

SAP NetWeaver

The SAP NetWeaver technology platform is designed for the flexible, step-by-step and efficient introduction of important IT processes in companies. Thanks to the high level of integration of the individual platform functions, its openness and the use of industry standards, the integration and application platform contributes significantly towards reducing the total cost of ownership (TCO) across the entire system landscape. SAP NetWeaver represents the technical basis of the SAP Business Suite, SAP xApps Composite Applications and other, partly industry-specific SAP applications.

SAP NetWeaver is also the technical basis for the forward-looking, service-oriented architecture for business applications (enterprise SOA). In an enterprise SOA, the application functions are made available in the form of enterprise services. These are web services that can be combined flexibly and used for business purposes. They are provided by different providers through data networks and communicate with other applications using open, standardized interfaces.

SAP NetWeaver integrates and connects people, information and business processes across technologies and companies. It enables companies to adjust to changes quickly. SAP NetWeaver ensures that a company's crucial business processes are reliable, safe and scalable. Furthermore, SAP NetWeaver enables companies to maximize the benefits from the current software and systems. Non-uniform integration technologies are consolidated and predefined business content is provided thus reducing the

amount of manual work required. SAP NetWeaver is based on a technology using industry standards and can be enhanced with popular development tools such as Java 2 Platform, Enterprise Edition (J2EE), Microsoft .NET and IBM WebSphere.

IT Practices and IT Scenarios



IT Practices			IT Scenarios						
User Productivity Enablement	Running an Enterprise Portal	Enabling User Collaboration	Business Task Management	Mobilizing Business Processes	Enterprise Knowledge Management	Enterprise Search			
Data Unification	Master Data Management			Enterprise Data Warehousing					
Business Information Management	Enterprise Reporting, Query, and Analysis		Business Planning and Analytical Services		Enterprise Data Warehousing	Enterprise Knowledge Management	Enterprise Search		
Business Event Management	Business Activity Monitoring				Business Task Management				
End-to-End Process Integration	Enabling Application-to-Application Processes		Enabling Business-to-Business Processes		Business Process Management	Enabling Platform Interoperability		Business Task Management	
Custom Development	Developing, Configuring, and Adapting Applications				Enabling Platform Interoperability				
Unified Lifecycle Management	Software Life-Cycle Management				SAP NetWeaver Operations				
Application Governance and Security Management	Authentication and Single Sign-On				Integrated User and Access Management				
Consolidation	Enabling Platform Interoperability		SAP NetWeaver Operations		Master Data Management		Enterprise Knowledge Management		Enterprise Data Warehousing
Enterprise SOA Design and Deployment	Enabling Enterprise Services								

Figure 9: IT Practices and IT Scenarios

SAP NetWeaver enables you to implement IT processes in a range of solution methods, called **IT practices**. For each practice, SAP NetWeaver supports a range of key IT activities, which can be performed using the integrated components of the platform. The focus here is not on system and technological components but on the IT and business goals of the company. IT practices enable you to reach your company's goals in individual and manageable projects, that is, in sequential steps and according to their importance. IT practices refer to, for example, the increase of user productivity through improved, cross-company collaboration, personalized access to applications and data and optimized knowledge management. IT practices show how SAP NetWeaver can be used to solve certain IT problems by means of **IT scenarios**. For each IT practice, SAP NetWeaver supplies corresponding IT scenarios, which act as implementation guides.

The aim of the IT scenarios is to help you as a customer, partner or service provider with the installation, configuration and operation of SAP NetWeaver as well as the operation of SAP applications, customer-specific applications and the implementation of your defined IT scenarios.

SAP NetWeaver Layers



- **Integration of people, information and processes**
 - Across technological and organizational boundaries
 - .NET and WebSphere interoperability
- **The technological foundation for SAP customers and partners**
 - Solutions with lower total cost of ownership (TCO)
 - The Enterprise Services Architecture (ESA) increases the flexibility and extendibility of business processes

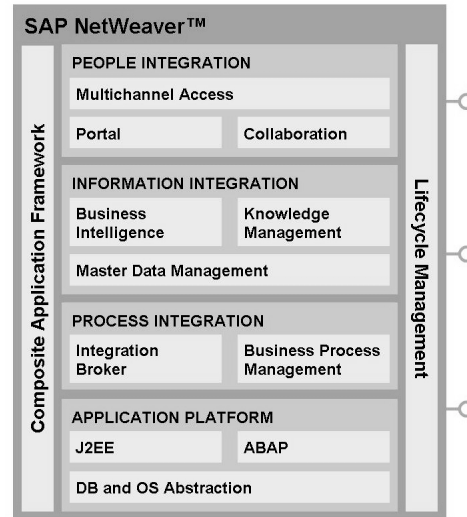


Figure 10: SAP NetWeaver: Capabilities

- **People Integration**

People integration ensures that your employees have access to the information and functions that they need in order to perform their activities quickly and efficiently. The functions of the **SAP NetWeaver Portal** are the most important in this context. **SAP NetWeaver Mobile** serves as the universal platform for company mobility and supports SAP solutions for Mobile Business.

- **Information Integration**

This integration layer provides access to all structured and non-structured information in your company. The key areas of information integration are mostly formed by **SAP NetWeaver Business Intelligence**, **Knowledge Management**, as a component of SAP NetWeaver Portal and **SAP NetWeaver Master Data Management**. All aforementioned key areas provide comprehensive options for uniform access to distributed information in a company.

- **Process Integration**

Process integration ensures that business processes work across system borders in a landscape of heterogeneous systems. The **SAP NetWeaver Exchange Infrastructure** (SAP XI) plays a crucial role here and connects SAP systems and non-SAP systems. It mainly uses open standards from the XML and Java world to do so.

- **Application Platform**

The **SAP NetWeaver Application Server** provides a proven infrastructure for operating business applications that are based on key technologies such as J2EE or ABAP. Besides the runtime environment for J2EE and ABAP applications, the corresponding development tools are also shipped and help you implement your applications. Open standards, browser-based access to applications and web services complete the application platform and serve as an important instrument for the enterprise service architecture.

Beyond these four steps and in parallel to them, SAP NetWeaver offers Lifecycle Management and the Composite Application Framework:

Lifecycle Management

Lifecycle Management supports design, development, deployment and change management of the entire solution and thus meets the requirements in the security, design and modeling, testing and systems environment management areas.

Composite Application Framework

The Composite Application Framework (CAF) is integrated into SAP NetWeaver and enables SAP and its partner to use tools, frameworks, rules and methods to create new applications that are based on business processes across function areas.

SAP NetWeaver Application Server (SAP NetWeaver AS)

Almost every SAP system is based on SAP NetWeaver AS and uses it as the runtime environment. Together with the database, SAP NetWeaver AS is the application platform of SAP NetWeaver.

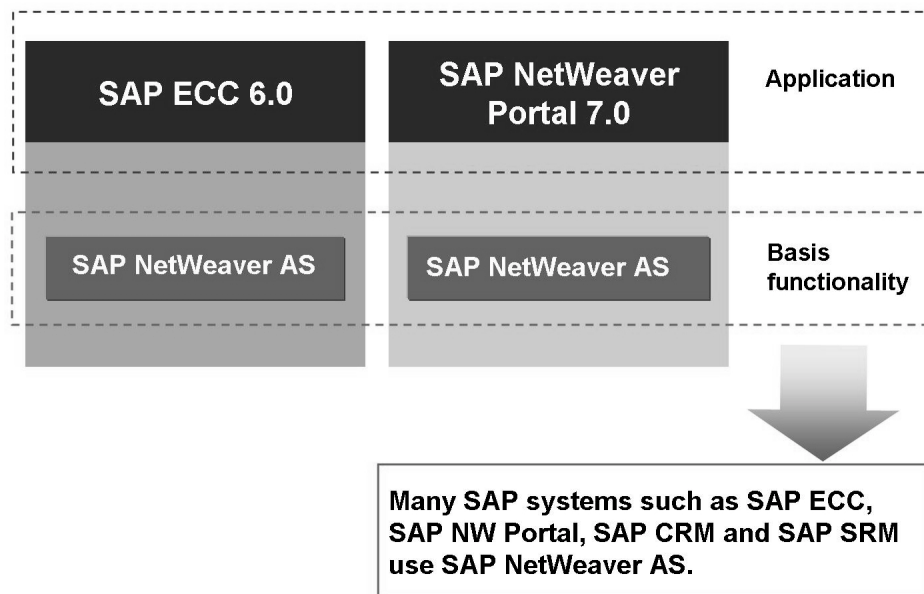
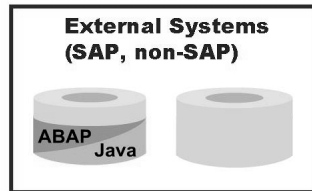


Figure 11: SAP NetWeaver AS as the Basis for SAP Systems



SAP NetWeaver Application Server offers

- Multi-level architecture
- Reliable, tested runtime environment for ABAP and Java programs
- High scalability



- Operating system and database independence
- Convenient development environment
- Support for various protocols (such as HTTP, RFC)
- Highest security standards for executing complex business processes

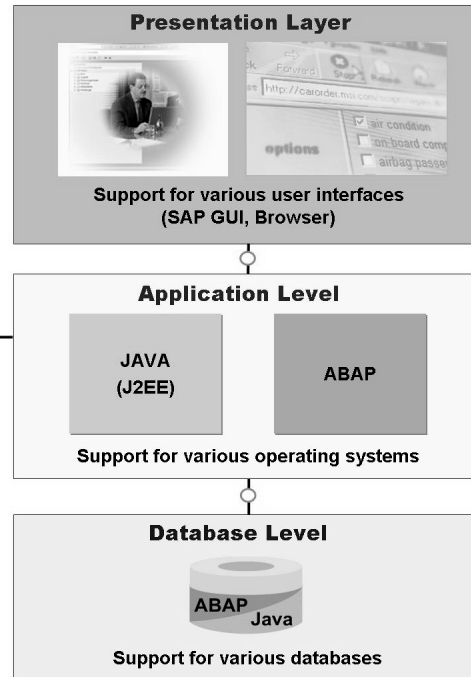


Figure 12: SAP NetWeaver Application Server

SAP NetWeaver AS is the logical result of the further development of the SAP Application Server Technology (previously: SAP Basis), whereby special attention is paid to web-based applications.

SAP NetWeaver AS offers:

- A reliable and extensively tested runtime environment, which has been developed further continuously over more than ten years
- A framework for executing complex business processes that meet the highest security standards
- A reliable and user-friendly development environment
- Support for open standards, including HTTP, HTTPS, SMTP, WebDAV, SOAP, SSL, SSO, X.509, Unicode, HTML, XML and WML
- High scalability
- Support for different operating system and database platforms

Since the applications delivered by SAP do not always require both runtime environments, that is, ABAP and Java, there are different installation options for SAP NetWeaver AS. These are:

- **SAP NetWeaver AS ABAP:** Complete infrastructure in which ABAP-based applications can be developed and used.
- **SAP NetWeaver AS Java:** Complete infrastructure in which J2EE-conform applications can be developed and used.
- **SAP NetWeaver AS ABAP+Java:** Complete infrastructure in which ABAP-based and J2EE-based applications can be developed and used. This installation emphasizes the seamless Java-ABAP integration.



Lesson Summary

You should now be able to:

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

Related Information

For more information, see the following sections:

- <http://www.sap.com/netweaver>.
- <http://service.sap.com/it-scenarios>



Unit Summary

You should now be able to:

- List the key components of the SAP solutions
- Explain the differences between an application and a component
- Describe the SAP software delivery process
- Explain the SAP maintenance strategy
- Find out when SAP software releases will become available
- Explain the basic concept of SAP NetWeaver
- Describe the central role played by the SAP NetWeaver Application Server



Test Your Knowledge

1. The following solutions are provided by SAP:

Choose the correct answer(s).

- ☐ A SAP Business Change Enabler (SAP BCE)
- ☐ B SAP Product Lifecycle Management (SAP PLM)
- ☐ C SAP Customer Relationship Management (SAP CRM)
- ☐ D SAP Supply Chain Management (SAP SCM)
- ☐ E SAP Strategic Organization Planning (SAP SOP)

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



Answers

1. The following solutions are provided by SAP:

Answer: B, C, D

SAP BCE and SAP SOP are imaginary products. The others are real SAP applications.

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

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

Unit 2

Navigation

Unit Overview

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



Unit Objectives

After completing this unit, you will be able to:

- Log on to the system successfully
- List some variants of the *SAP GUI*
- Log off from the system successfully
- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Describe the standard menus **System** and **Help**.
- Use different personalization options in the SAP system

Unit Contents

Lesson: Navigation in General	36
Procedure: Changing the Initial Password	43
Procedure: Changing the Changed Password Again.....	44
Procedure: Log off via the Menu Bar	47
Exercise 2: Logon and Screen Design	49
Lesson: Advanced Navigation in the SAP GUI	52
Exercise 3: Calling Functions.....	61
Lesson: Appendix - Personalizing the User Interface.....	67

Lesson: Navigation in General

Lesson Overview

This lesson provides an overview of the different navigation options in SAP systems and you will learn basic navigation commands using the SAP GUI for Windows as the example.



Lesson Objectives

After completing this lesson, you will be able to:

- Log on to the system successfully
- List some variants of the *SAP GUI*
- Log off from the system successfully

Business Example

You want to find out which options are generally available to your users to access SAP systems.

Different User Interfaces in the SAP Environment

There are numerous different types of SAP systems, such as “classical” ERP systems, but there are also other system types like Portal or Business Information Warehouse systems, for example. Depending on what they are used for, these different system types do offer rather different user interfaces.

The following list names some frequently used access technologies, which end users of these system (so not “technicians”) use frequently.



Examples of Different Access Technologies for a Small Selection of SAP System Types

SAP Product	Typical Access Technology
SAP ERP (with SAP ECC)	SAP GUI for Windows (GUI = Graphical User Interface)
SAP NetWeaver BI 7.0	Browser-based interface (different browsers possible) BEx Analyzer (additionally requires Excel on the front-end)
SAP Business ByDesign	Browser-based interface (different browsers possible)
SAP Business All-in-One	SAP NetWeaver Business Client

Please note that the above list does not cover all SAP products and that not all possible (!) access technologies are listed in all cases. For example, an SAP ECC system can also be accessed by browser or using the *SAP GUI* for Java; SAP NetWeaver BI is also accessible by *SAP GUI* for Windows and so on.

The following display will be limited to the most popular access technology, the *SAP GUI*.

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

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

The following types of *SAP GUI* are available:

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



Note: In the above list, the official terms of the *SAP GUI* are used; these are only available in English. Colloquially, the following terms are used synonymously (in the same order): *SAP GUI for Windows*, *Java GUI* and *Web GUI*. Please note that these are not the “official SAP terms” but are used frequently nonetheless.



Caution: The above variants of the SAP GUI are suitable only for working with ABAP-based SAP systems (including the SAP GUI for the Java Environment).

Java-based SAP applications are accessed using a browser.

SAP GUI for the Windows environment (abbreviated to **SAP GUI for Windows**) is the *SAP GUI* implementation in a Microsoft Windows environment. *SAP GUI* for Windows is written in C or C++, runs on Windows-based platforms, and supports all transactions in an SAP system.

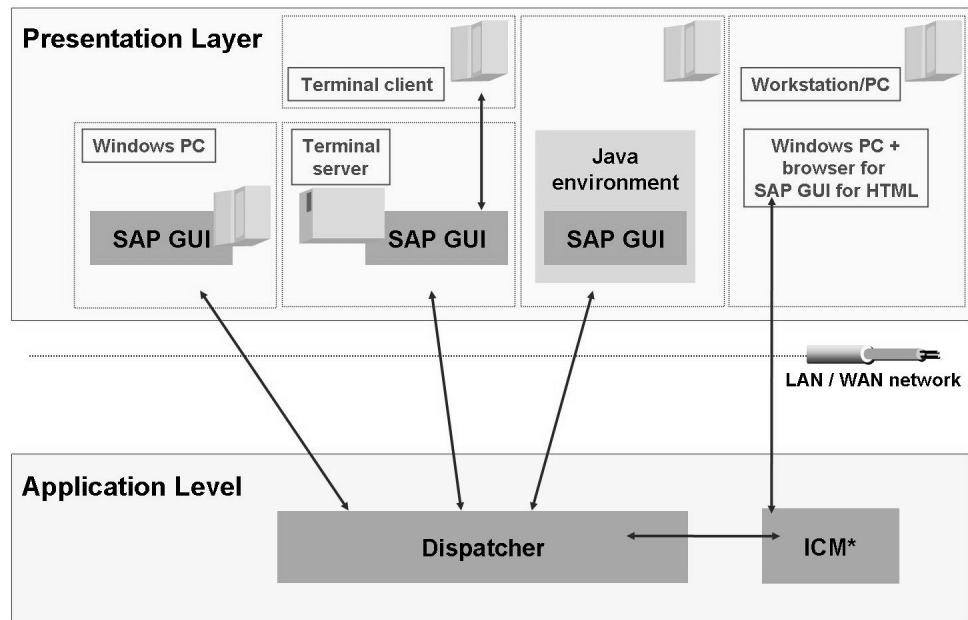
The data flow between the presentation layer and the application layer does not consist of prepared screens, but rather of logical, compact information about control elements and user input (DIAG protocol). The volume of data transferred for each screen change is generally a few KB. You can therefore also easily connect presentation servers over WANs.

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

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

SAP GUI for HTML consists of the Internet Transaction Server (ITS) in the Application Server ABAP (short: AS ABAP) on the server side and a Web browser on the client side. The ITS is required to convert the *SAP GUI* data flow already in the *AS ABAP* into HTML and back. In this case, an HTML data stream is exchanged between the Internet Communication Manager of the AS ABAP and the front-end. The primary advantage of the *SAP GUI for HTML* is the simpler installation on the front-end; a browser installation suffices.

The following graphic shows the various alternatives and the flow of communication with the *AS ABAP dispatcher*.



* ICM = Internet Communication Manager

Figure 13: Alternative types of SAP GUI



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

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

Logging on to the System

Most of the solutions shipped by SAP can be accessed using the *SAP GUI* (*SAP Graphical User Interface*). For the sake of simplicity, further descriptions refer to the *SAP GUI* for the Windows environment.

SAP provides another program for starting the front-end: *SAP Logon*. When you call up *SAP Logon*, it displays a list of SAP systems for which you can start the logon process. This list is taken from a file on the front end: **saplogon.ini**. This file is usually centrally preconfigured and provided for end users.

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

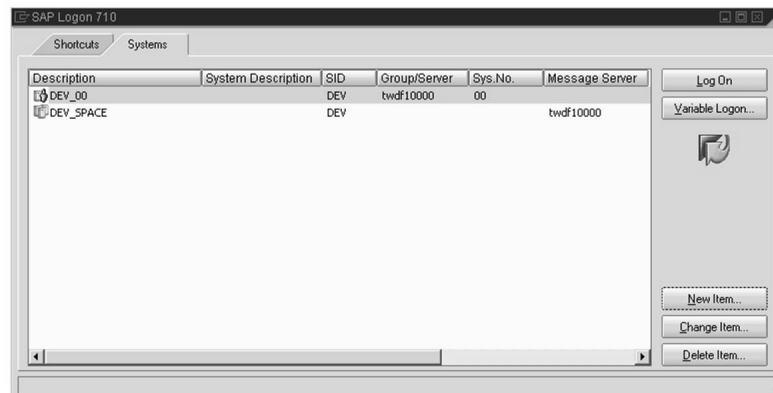


Figure 14: The SAP Logon program

When logging on to an SAP system, you will be prompted to enter the user and password, among other things. If you have implemented a Single Sign-On (SSO) solution, you may not need to enter this information. When you log on, you also select a certain client of the SAP system; this field is usually already filled appropriately.



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

When logging on, you can select a logon language supported by that system. SAP systems can support a large number of languages, the minimum being English and one other selected language. The number of installed languages determines which languages your system supports.

On the logon screen, you also have the option of using the *New password* pushbutton to change your password, at the most once per day.

Your system administrator can add additional text to your logon screen. To do so, follow the [SAP-Note 205487 - Own text on SAPGUI logon screen](#).



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

In the course of **one** logon to a system, you can work in several sessions (processing windows of an SAP system) simultaneously. Your system administrator can use a system parameter to define how many sessions are permissible for each logon to the SAP system. For more details please see the [SAP-Note 12466 - Logon restrictions in R/3](#).

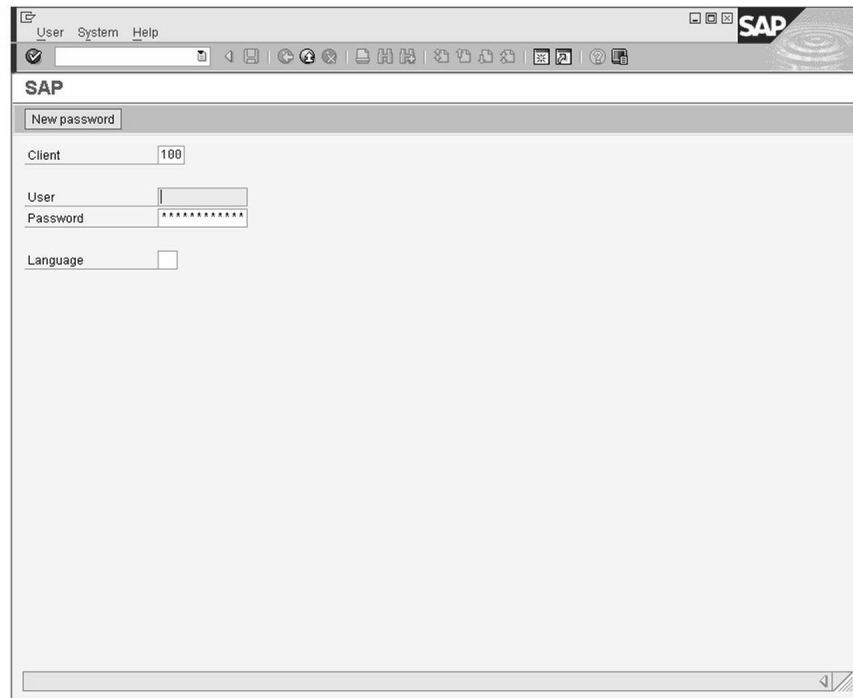


Figure 15: The logon screen for an SAP system

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

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

- Close all existing session(s) and log on again
- Keep existing session(s) open, and open an additional session (this is recorded in the log).
- Terminate this logon

Changing the Password

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




Changing the Initial Password

1. Enter your new password twice, whereby you activate both input fields in sequence by means of mouse-click or tab button. Please note that numerous rules regarding the structure of a password can be set up for your system.



Figure 16: Dialog Box for Changing the Password

2. Finally, choose  to save your new password.



Changing the Changed Password Again

Prerequisites



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

Procedure

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

Screen Structure

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

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

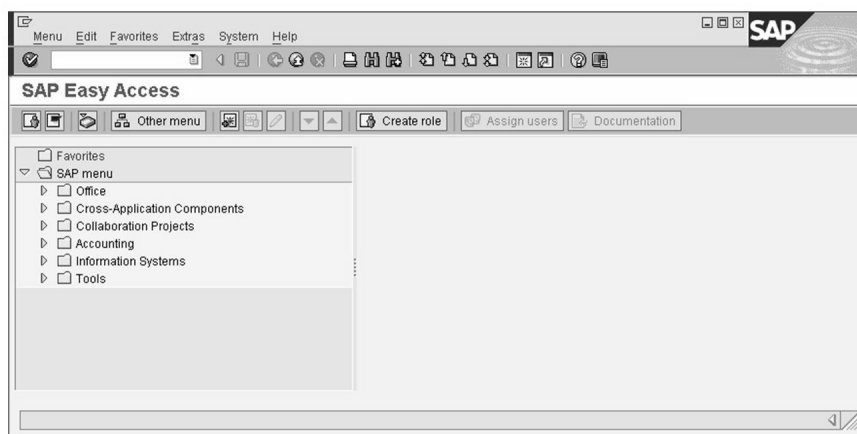


Figure 17: SAP Easy Access

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

A typical screen (with some common elements) appears if you choose *System* → *User Profile* → *Own Data* and then the *Defaults* tab page. You will then see this screen with the data for your own user:

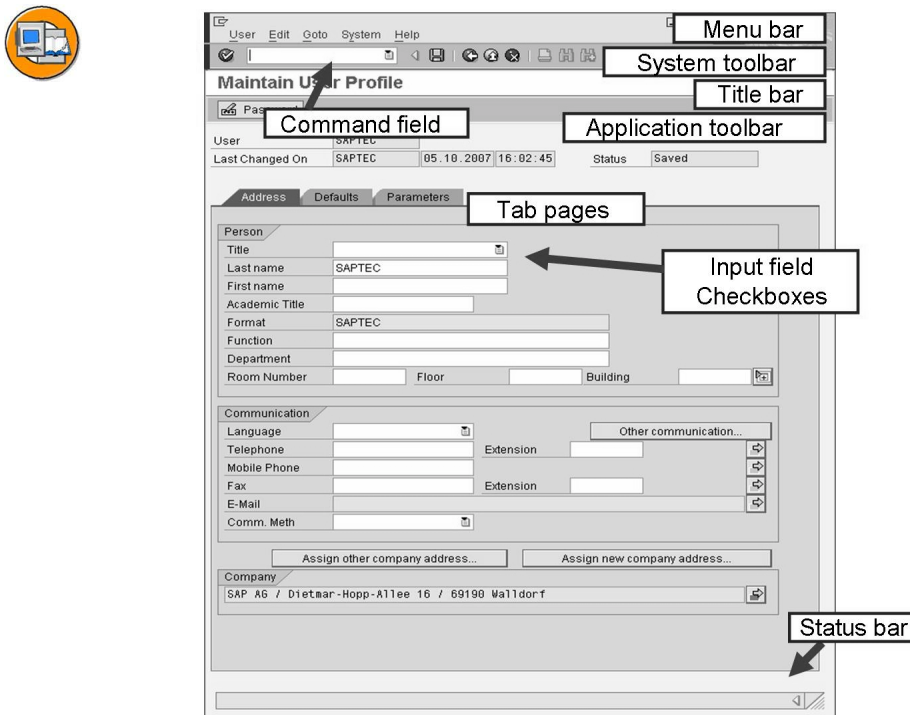


Figure 18: Screen Elements in SAP GUI

An SAP user interface can contain the following simple screen elements, for example:

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

You can start applications directly by entering their transaction code in the **command field**, which is usually hidden. You can find the transaction code for an application either in the overview menu on the *SAP Easy Access* screen, in the status bar (see below), or in the application itself under *System → Status*.

The **Menu bar** is the top line of any dialog window in the SAP system. The menus shown here depend on which application you are in. These menus may also contain cascading menus.

The pushbuttons in the **standard toolbar** are shown on every SAP screen. If they are not available in an application, pushbuttons that you cannot use on that screen are deactivated. If you place the cursor over a pushbutton for a short time, the system displays a flag with the name or function of the pushbutton. If appropriate, the corresponding function key setting is also displayed.

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

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

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

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

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

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

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

Logging off from the System


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



Log off via the Menu Bar

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



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



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

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

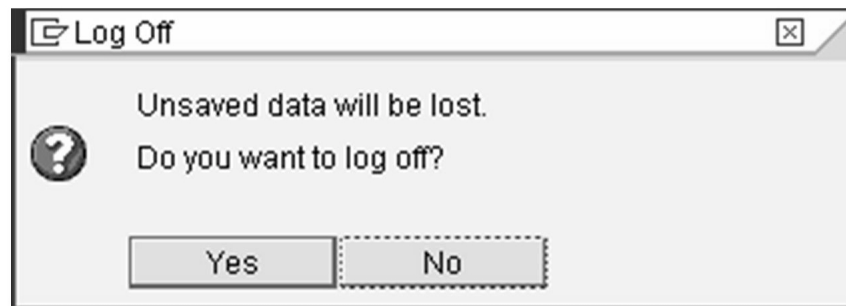


Figure 19: Log off Dialog Box

3. To log off, choose *Yes*.



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

Exercise 2: Logon and Screen Design

Exercise Objectives

After completing this exercise, you will be able to:

- Use the SAP GUI for Windows to logon to the SAP system and navigate

Business Example

You want to work in an SAP system.

Task: Logging on to the System and Initial Tasks

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



Hint: Use the logon data provided by your instructor.

1. Start the SAP Logon program and create a new entry according to the information given by your instructor. Choose the *Logon* pushbutton.

Enter the client, user name, initial password, and logon language supplied by the instructor. When you log on for the first time, enter a new password of your choice twice in the window that appears.
2. What is the maximum number of parallel sessions (windows of the SAP system) that you can open using *System* → *Create Session*?
3. What is the name of the function you reach if you choose *Tools* → *Administration* → *Monitor* → *System Monitoring* → *User Overview*? What transaction code could you use to call this transaction instead of the menu?

Solution 2: Logon and Screen Design

Task: Logging on to the System and Initial Tasks

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



Hint: Use the logon data provided by your instructor.

1. Start the SAP Logon program and create a new entry according to the information given by your instructor. Choose the *Logon* pushbutton.

Enter the client, user name, initial password, and logon language supplied by the instructor. When you log on for the first time, enter a new password of your choice twice in the window that appears.
 - a) Follow the instructions in the exercise.
2. What is the maximum number of parallel sessions (windows of the SAP system) that you can open using *System → Create Session*?
 - a) This training system should allow you to open up to 6 sessions.
3. What is the name of the function you reach if you choose *Tools → Administration → Monitor → System Monitoring → User Overview*? What transaction code could you use to call this transaction instead of the menu?
 - a) The function is called *User List* (see the entry in the title bar). You can find the relevant transaction code using *System → Status*; in this case, the code is SM04.

You can also choose to display the transaction code for an active transaction in the status bar.

Alternatively, you can permanently display the transaction codes in the SAP Easy Access menu. Choose *Menu bar → Extras → Settings* and activate the field *Display Technical Names*.

A fourth option would be to display the details by choosing *Menu Bar → Extras → Technical details*. For this, the corresponding menu entry would first have to be selected with the mouse.



Lesson Summary

You should now be able to:

- Log on to the system successfully
- List some variants of the *SAP GUI*
- Log off from the system successfully

Related Information

- For more information, see the following online documentation: *Help → SAP Library → Introduction to the SAP System.*

Lesson: Advanced Navigation in the SAP GUI

Lesson Overview

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



Lesson Objectives

After completing this lesson, you will be able to:

- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Describe the standard menus **System** and **Help**.

Business Example

You want to know how functions can be started in the SAP system and what help the system provides you with.

User Menu and SAP Menu

The SAP Access menu is automatically displayed after logging on; it represents the standard access point to an SAP system for the SAP GUI for Windows. The navigation bar in the left part of the screen contains a clear tree structure, consisting of menu entries and favorites' entries. You can adjust the favorites' area in particular to your needs. The entries of the SAP menus and the user menus can only be changed by system administrators.

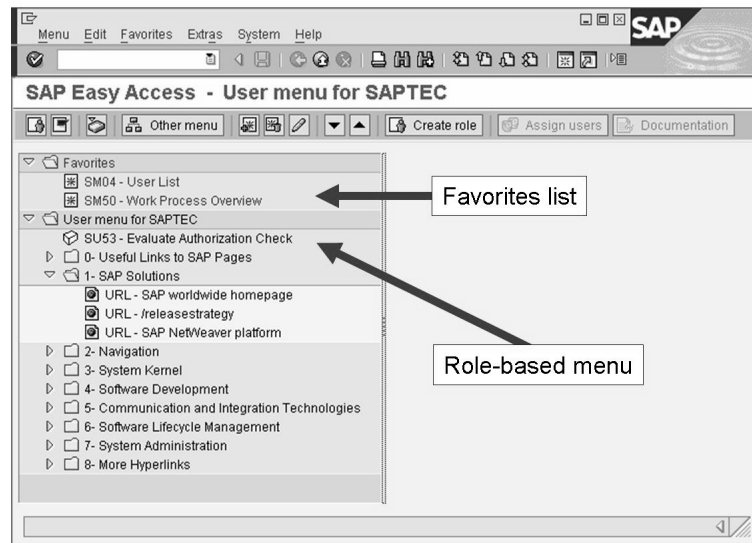


Figure 20: Favorites list and user menu

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



Hint: System administrators can use table *USERS_SSM* to determine whether or not users are allowed to switch between the SAP menu and their user menus. If both menus are disallowed, then the user menu is still displayed. The **availability** of the user or SAP menu does not affect a user's authorizations. This means that a user can always call authorized functions using a transaction code, independent of the menu structure.

The role-based user menu is created on the basis of the role(s) assigned to the user and transferred to the front end. This means that it makes sense to keep roles as small as possible because large roles could take a long time to transfer to the front end in certain circumstances. If a user has several roles assigned, then some functions can be repeated in different parts of the role-based user menu. If you do not want these to appear twice, see [SAP-Note 357693 - Redundancy avoidance in Easy Access](#) (and the other notes listed therein) for information on identifying and deleting duplicates, and related questions.



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

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

Favorites Management

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

The favorites list thus contains references to SAP system functions or links to internet content or to files on the end user's front-end computer. The favorites list, which is initially empty, can be edited by each end user in accordance with his or her preferences; you can only view your own favorites list. Since data on the favorites is stored within the SAP system, each user might have different collections of favorites in different systems.



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

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

The *Favorites* menu gives you the option of adding a function from the user (or SAP) menu to your personal favorites list by selecting the function and choosing *Favorites* → *Add*. Here, the menu path is also written with the text of the favorites entry to a limited extent. If you have highlighted a menu entry using the mouse, you can copy the selected node in the favorites' area by right-hand mouse click.

Alternatively, you can also drag the elements from the menu area using the mouse and store them in the favorites' area (Drag&Drop). You can add URLs or links to files to your list of favorites by choosing *Favorites* → *Add other objects*. Click on the triangle symbol to the left of the file symbol to expand or collapse the list of favorites.



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

You can delete favorites' entries that you no longer need. In doing so, the link to the corresponding function is deleted from your favorites' list. If you want to delete the entire Favorites menu, highlight the entry *Favorites* and right-click to call up the available context menu. Choose *Delete all favorites*.

Calling Up Functions

You have several options for navigating in an SAP system:

- Entering transaction codes in the OK Code field
- By choosing items from menus in the menu bar
- By choosing items from the favorites list or from the user or SAP menus

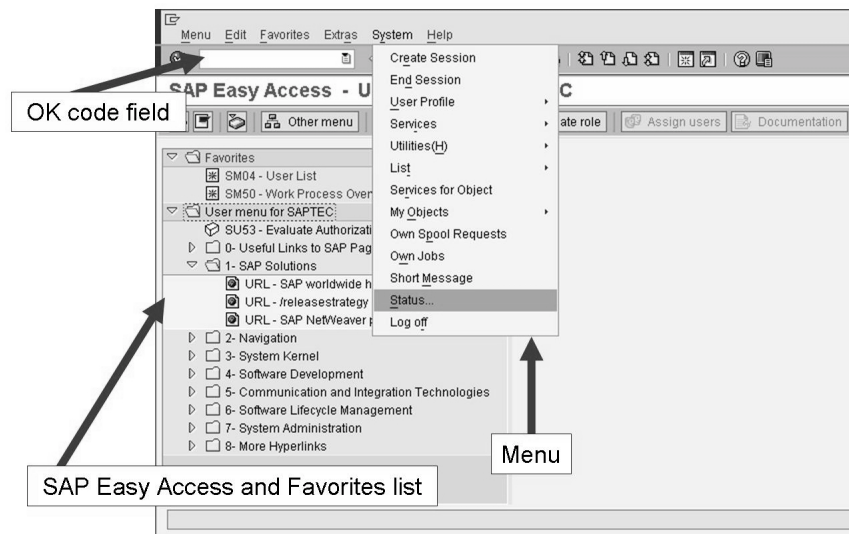


Figure 21: Various navigation options

You can reach the menus in the menu bar by simply using the key combination **Alt+<#>** (# corresponds to the underlined letter of the selected menu item) or **Alt** and cursor navigation using the arrow keys on the keyboard.

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



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

- **TAB** moves from one field element to the next within a field group.
- **Ctrl + TAB** goes from one field group to the first element of the next field group.
- **Ctrl + /** goes directly to the OK Code field.

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

When the OK Code field is active (also known as *command field*), you can use the F1 key to display possible entries for this important field. The following entries are possible:



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



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

Help Options

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

F1 Help

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

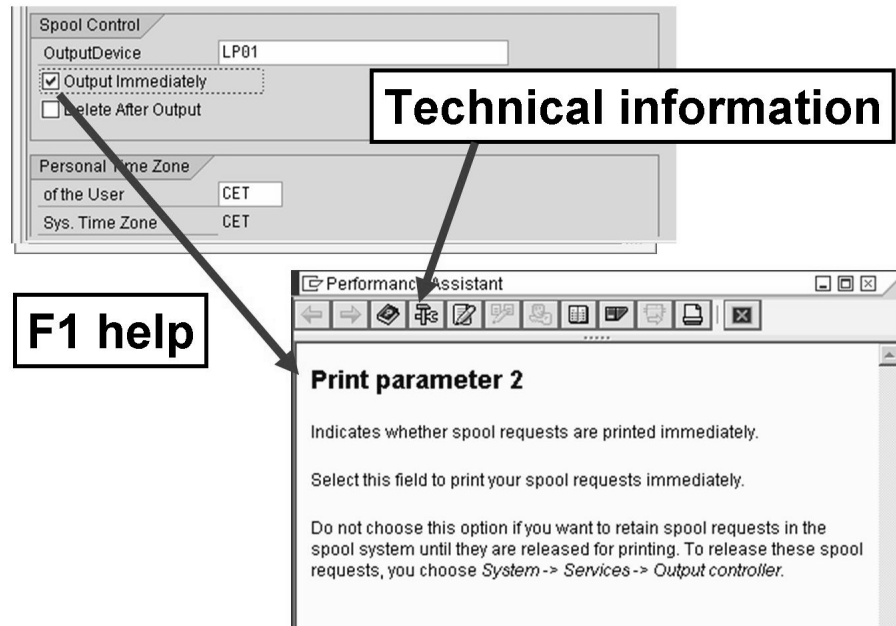


Figure 22: The F1 help

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



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

F4 Help

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

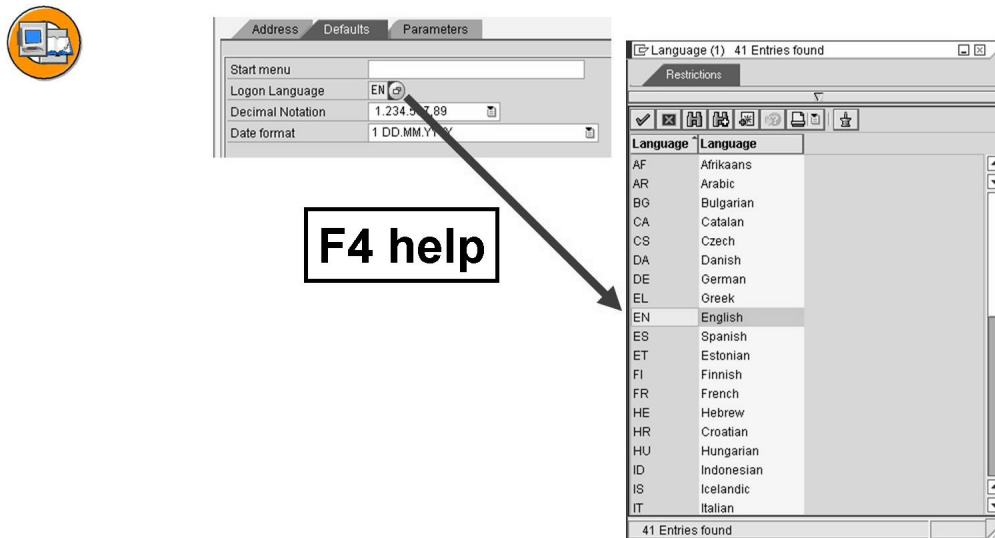


Figure 23: The F4 help

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



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

SAP Library (Online Documentation)

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

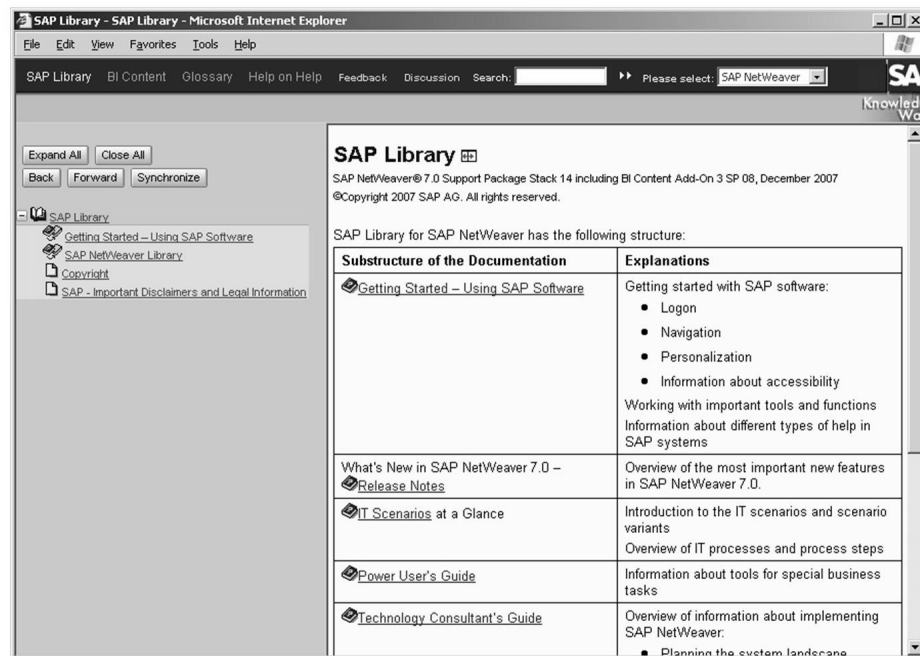


Figure 24: The SAP Library



Hint: You can also access the SAP Library on the Internet. Call the address <http://help.sap.com>. You can access the full product documentation for different releases here; a user-friendly full-text search by solution is also available.

The System and Help Menus

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

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

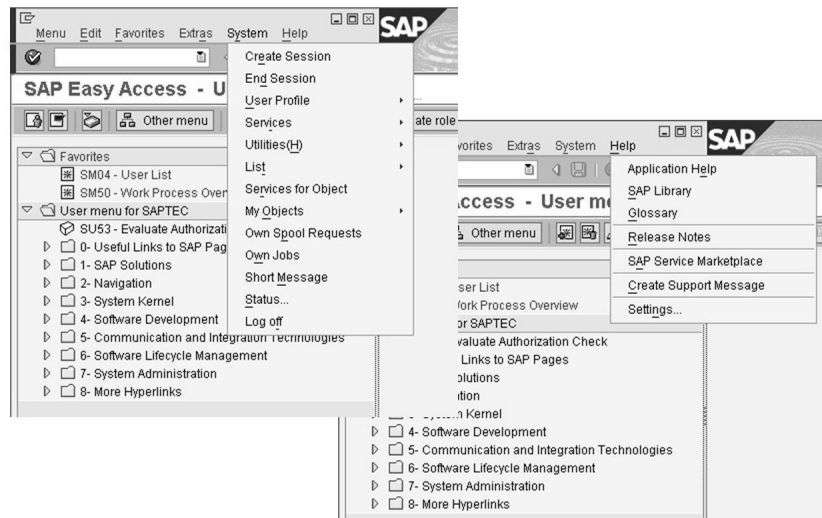


Figure 25: The System and Help menus

You can use the *Help* menu to access the system documentation. You can also display the **Release Notes** and configure the standard settings for your F4 Help. However, you will mostly use the Help menu to navigate to the specific section of the documentation relevant for your current work in the current context. The context-sensitive help is also called **Application Help**. The tool for accessing the documentation as a whole is the **SAP Library**.

Exercise 3: Calling Functions

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

You want to call functions.

Task 1: Various Ways of Calling Up the User Overview

Who is currently logged on to the training system?

1. Call the transaction for displaying a list of users who are logged on. Select from the SAP standard menu *Tools → Administration → Monitor → System Monitoring → User Overview*.
2. Quit the user overview by using either the F3 key or the *Back* button in the standard toolbar. Now enter **sm04** in the command field on the *SAP Easy Access* screen. This takes you directly to the user overview.

Task 2: Optional: Creating a Favorites Menu

Save often-used transactions in a favorites menu.

1. Expand the SAP menu to the user overview: *Tools → Administration → Monitor → System Monitoring → User Overview*. Highlight the transaction entry with the mouse.

Then select *Favorites → Add* in the menu bar. The transaction is saved in the favorites' area.
2. Add the following transactions to the Favorites menu without expanding the SAP menu first:

RSPFPAR, SEARCH_SAP_MENU, SEARCH_USER_MENU
3. Can you distinguish between favorites' entries by the method used to insert them?
4. Create a folder with the title *Search Options* in which you can store both search transactions.

Continued on next page

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

Transaction	Path in the SAP menu
SM04	
SM04	
SM04	

Task 3: Optional: Using F1 and F4 Help

Use the technical information of the F1 help.

1. Find out in which table of your SAP system the transaction codes are stored. To do so, use the dialog input field for entering a start transaction, here: *SAP Easy Access Extras* → *Set start transaction*.

Solution 3: Calling Functions

Task 1: Various Ways of Calling Up the User Overview

Who is currently logged on to the training system?

1. Call the transaction for displaying a list of users who are logged on. Select from the SAP standard menu *Tools → Administration → Monitor → System Monitoring → User Overview*.
 - a) See exercise for solution
2. Quit the user overview by using either the F3 key or the *Back* button in the standard toolbar. Now enter **sm04** in the command field on the *SAP Easy Access* screen. This takes you directly to the user overview.
 - a) For the solution, see the question.

Task 2: Optional: Creating a Favorites Menu

Save often-used transactions in a favorites menu.

1. Expand the SAP menu to the user overview: *Tools → Administration → Monitor → System Monitoring → User Overview*. Highlight the transaction entry with the mouse.

Then select *Favorites → Add* in the menu bar. The transaction is saved in the favorites' area.

 - a) See exercise text.



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

2. Add the following transactions to the Favorites menu without expanding the SAP menu first:
RSPFPAR, SEARCH_SAP_MENU, SEARCH_USER_MENU
 - a) Then select *Favorites → Insert transaction* in the menu bar. Enter the desired transaction in the dialog box and save the data. This process must be executed for every transaction individually.

Continued on next page

3. Can you distinguish between favorites' entries by the method used to insert them?
 - a) Yes, when inserting with previous selection in the menu, a part of the path information of the menu access is stored as a favorites text.
4. Create a folder with the title *Search Options* in which you can store both search transactions.
 - a) Choose *Favorites → Insert folder*. Name the file *Search Options*. Then, drag both search transactions (SEARCH_SAP_MENU, SEARCH_USER_MENU) into the file using the mouse.
5. Find out how often and where in the SAP menu the transaction for the user overview is stored.

Transaction	Path in the SAP menu
SM04	
SM04	
SM04	

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

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

Continued on next page

Task 3: Optional: Using F1 and F4 Help

Use the technical information of the F1 help.

1. Find out in which table of your SAP system the transaction codes are stored. To do so, use the dialog input field for entering a start transaction, here: *SAP Easy Access Extras* → *Set start transaction*.
 - a) You can get this information by calling up the F1 help in the dialog box to enter the start transaction. Then branch to the technical information. The table is called *TSTC*.



Lesson Summary

You should now be able to:

- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Describe the standard menus **System** and **Help**.

Related Information

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

Lesson: Appendix - Personalizing the User Interface

Lesson Overview

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



Lesson Objectives

After completing this lesson, you will be able to:

- Use different personalization options in the SAP system

Business Example

The end user wants to personalize the way in which they access SAP screens.

Ways of Personalizing the SAP GUI

There are many different options available for personalizing the screen display, some of which are described below. You can use *Extras* → *Settings* to change the appearance of the initial screen of the SAP system, for example, by turning off the graphics display on the right side of the screen, or by displaying technical names (transaction codes) on the *SAP Easy Access* screen.

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



Using the *Customizing of local layout* pushbutton, you can manage the input history by choosing *Options...* → *Local data*. If it is activated, the input history creates a small database on the front end containing the last x entries made in input fields in transactions. You can specify the value for “x”. These entries are offered as input help in appropriately declared fields. The input history is updated after a certain delay. To minimize this delay, choose *Options* → *Local Data* → *History* → *Immediate*.

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



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

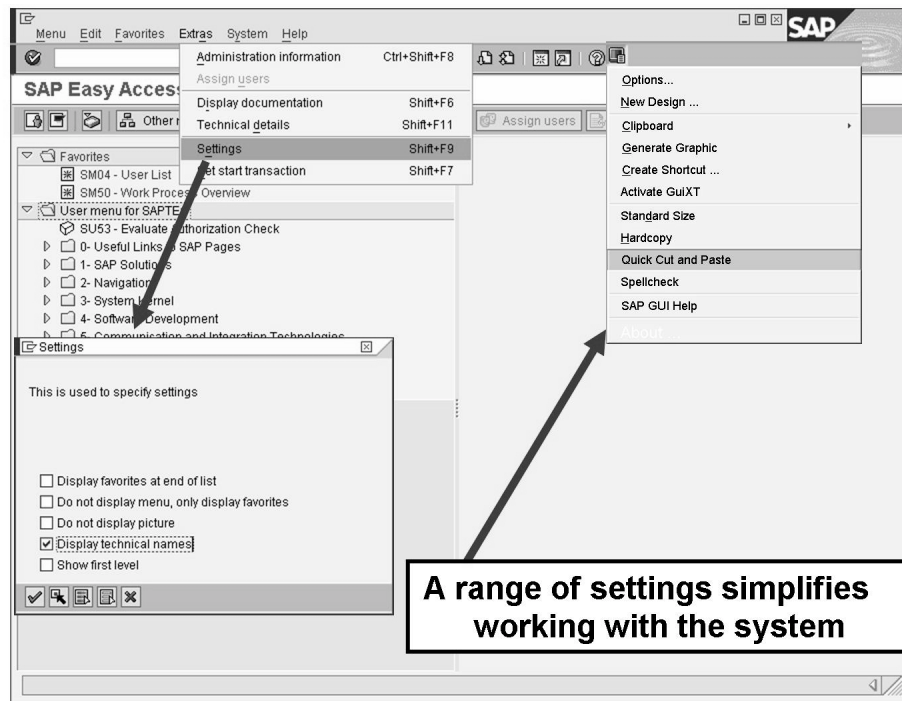


Figure 26: Two Possible Personalization Options

You can set personal standard values using *System* → *User Profile* → *Own Data*. You can choose between the tab pages Address, Fixed Values, and Parameters.



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

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



Hint: Use the information displayed in the status bar on the bottom right edge of the GUI window. There, you can, for example, always display the transaction code of the transaction that is currently being executed.



Lesson Summary

You should now be able to:

- Use different personalization options in the SAP system

Related Information

- For more information, see the documentation, in the section “Introduction to the SAP System”.



Unit Summary

You should now be able to:

- Log on to the system successfully
- List some variants of the *SAP GUI*
- Log off from the system successfully
- Use various methods to start SAP system functions
- Create a list of favorites
- Use the help functions
- Describe the standard menus **System** and **Help**.
- Use different personalization options in the SAP system



Test Your Knowledge

1. You can only work in one window (session) at a time in an SAP system.
Determine whether this statement is true or false.
 - ☐ True
 - ☐ False

2. Which of the following statements accurately describe a client in an SAP system?
Choose the correct answer(s).
 - ☐ A A client represents a completely independent business entity.
 - ☐ B A client has its own database.
 - ☐ C A client corresponds to a customer.
 - ☐ D A client may represent an entire company.

3. The following strings are valid entries in the command field:
Choose the correct answer(s).
 - ☐ A **/nend**
 - ☐ B **/nex**
 - ☐ C **?SM04**
 - ☐ D **/nsm04**
 - ☐ E From SAP Easy Access: **SM04**
 - ☐ F From SAP Easy Access: **search_sap_menu**

4. The following personalization options are available in the SAP GUI:
Choose the correct answer(s).
 - ☐ A You can change the font size in the SAP GUI within a certain range.
 - ☐ B You can display system messages in a dialog box.
 - ☐ C You can vary the size of input fields.
 - ☐ D You can deactivate the display of pictures in the SAP GUI
 - ☐ E Each user can integrate their own picture into the SAP Easy Access screen.
 - ☐ F You can use your own personal input history on your front end.



Answers

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

Answer: False

Within one logon, it is possible to work in several windows (sessions) in parallel.

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

Answer: A, D

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

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

Answer: A, B, D, E, F

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

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

Answer: A, B, D, F

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

Unit 3

The System Kernel

Unit Overview

This unit provides a detailed insight into how the SAP system processes user requests. You will learn about the different process types of the SAP NetWeaver Application Server and how they work.



Note: The following lessons have been formed from two different courses. The advantage to this is that all of the knowledge required for this range of topics is introduced, within the context of learning, in a training course for consultants. Consequently, you can use synergies to facilitate understanding among participants. This would not be possible if these topics were discussed at a later time.



Unit Objectives

After completing this unit, you will be able to:

- Outline simple client/server configurations
- Name the processes of the SAP NetWeaver Application Server
- Define the term instance and recognize the characteristics of a central instance
- Understand how AS ABAP works
- List the AS ABAP processes and describe their purpose
- Describe how requests to AS ABAP are processed
- Use basic Java terminology
- Understand how AS Java works
- Name the AS Java processes and describe their purpose
- Describe how requests to AS Java are processed
- Explain the term Central Services of SAP NetWeaver AS Java
- Understand and use concepts such as Java instance, Java dispatcher, and server
- Name the most important managers of the SAP NetWeaver AS
- Name the most important services of the SAP NetWeaver AS

Unit Contents

Lesson: Principal Architecture of the SAP NetWeaver Application Server...	77
Exercise 4: Structure of an Instance	93
Lesson: AS ABAP Processes	97
Exercise 5: Dialog Processing	119
Exercise 6: Optional: Lock Management	121
Exercise 7: Optional: Update Processing	123
Exercise 8: Printing	125
Exercise 9: Background Processing	127
Lesson: Fundamental Concepts of Java	130
Lesson: AS Java Processes	143
Lesson: Java Cluster Architecture	151
Lesson: The Internal Structure of SAP NetWeaver AS Java	158

Lesson: Principal Architecture of the SAP NetWeaver Application Server

Lesson Overview

This lesson introduces the principal architecture of SAP systems. Based on simple client/server configurations, the individual processes of the SAP NetWeaver Application Server are introduced and the term **instance** is defined. We will also explain the architecture of the different types of SAP NetWeaver Application Servers.



Lesson Objectives

After completing this lesson, you will be able to:

- Outline simple client/server configurations
- Name the processes of the SAP NetWeaver Application Server
- Define the term instance and recognize the characteristics of a central instance

Business Example

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

Introduction

SAP systems are used for mapping business processes or business applications. These applications should be implemented independent of the hardware environment used (operating system, database) to the greatest extent possible. To do so, *SAP NetWeaver* provides two different runtime environments: an ABAP runtime environment (usage type *AS ABAP*) and a Java runtime environment (usage type *AS Java*).

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

SAP not only provides a runtime environment for ABAP programs but also a runtime environment for **Java** programs. *AS Java* is an application server according to the Java 2 Enterprise Edition (J2EE) standard.

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

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

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

Client and Server: Terminology Definition

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

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

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

The following graphic clarifies the two approaches to the definitions.

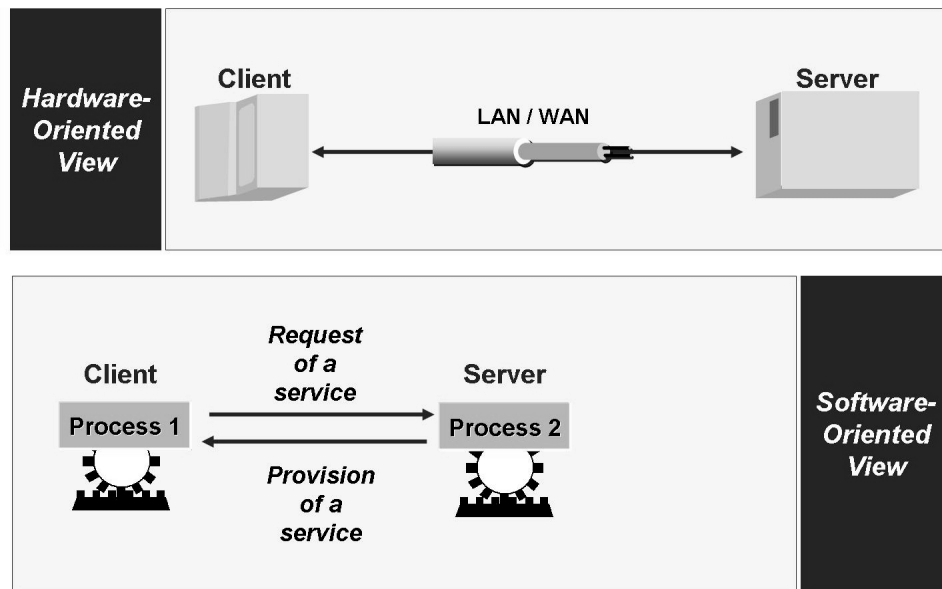


Figure 27: Hardware-Oriented View and Software-Oriented View

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

Client/Server Configuration for SAP Systems

The following processes are often used for operating business application software:

- Presentation processes (for example, for displaying screens)
- Application processes (for example, for executing application programs)
- Database processes (for example, for managing and organizing database data)

When you are installing and configuring an SAP system, you need to decide how you are going to distribute the required processes among the available hardware. There are various ways of doing this, some of which are described in more detail

below. Configurations are either single-tier or multi-tier, depending on the number of hardware layers used (see the following graphic). The *SAP ECC* system is an example of business application software.

- In single-tier configurations, all processing tasks (database, application and presentation processes) are performed by one computer. This is classic mainframe processing.
- Two-tier configurations are usually implemented using special presentation servers that are responsible solely for formatting the graphical interface.
- In a three-tier configuration, each layer runs on its own hosts. Many different application servers can simultaneously work with the data of a database server.

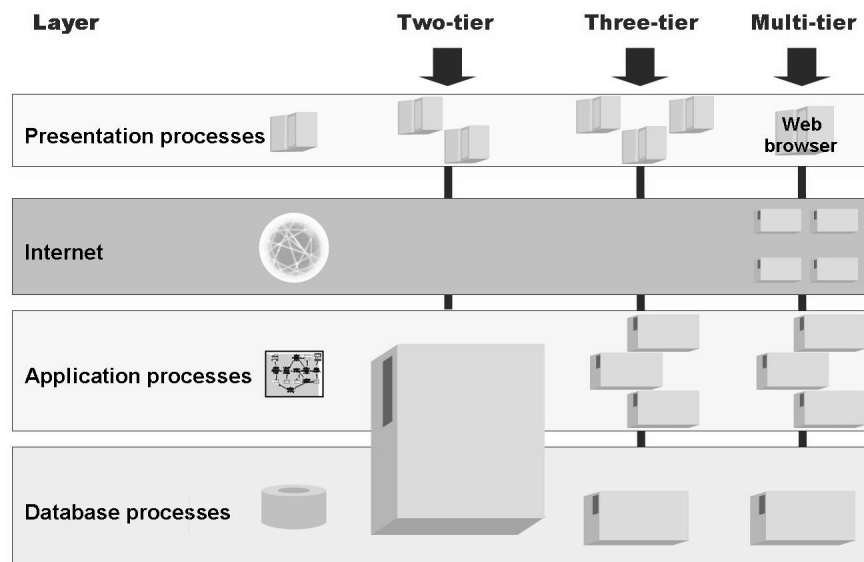


Figure 28: Simple Client/Server Configurations

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

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

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

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

In the SAP Business Suite environment, more complex client/server configurations consisting of more than three tiers are both theoretically possible and used in practice. An additional level could be a web server, for example.

The Instance

An **instance** is an administrative unit that combines SAP system components providing one or more services. The services provided by an instance are started or stopped together. You use a common **instance profile** to set parameters of all the components of an instance. Each instance has its own buffer areas. An instance runs on one physical computer, but there can be multiple instances on one computer. An instance is identified by the system ID (SID) and the **instance number**.



Hint: The terms (SAP) instance and Application Server are often used synonymously.

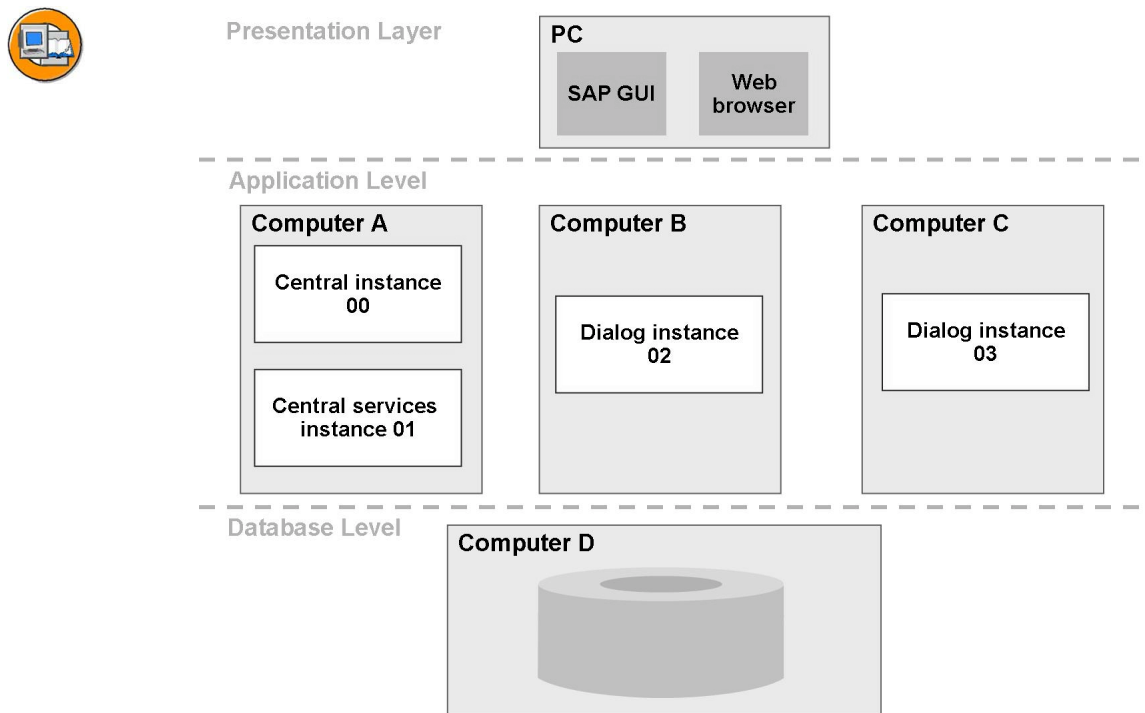


Figure 29: Instances of an SAP System (Example)

When you install an SAP system, you already have the option of separating the processes at application level from those at database level. This means that the database for an SAP system can be installed and operated on a separate physical computer, separated from the instances of the SAP system. There is exactly one database for each SAP system. The database usually has the same system ID (DB ID) as the SAP system.

The **central instance** of the SAP system is distinguished by the fact that it offers services that no other instance of the system offers. For the AS ABAP, these are the Message Server and the Enqueue work process (see below). For the AS Java you can recognize the central instance by the Software Deployment Manager (SDM).

The **central services instance** provides central services of the AS Java, the Message Service and the Enqueue Service (see below). For the AS ABAP, these services can also be moved to the **ABAP central services instance** for high availability reasons. These AS ABAP systems therefore no longer have a central instance.

All other instances of the system are typically called **dialog instances**.

If the central instance and the database (and for the AS Java also the central services instance) are installed on the same computer, this is referred to as a **central system**.

Processes of the SAP NetWeaver Application Server

The SAP runtime system consists of a large number of parallel processes that work together. Here, you can distinguish between the runtime environment for ABAP (AS ABAP) and the runtime environment for Java (AS Java) .

AS ABAP Processes

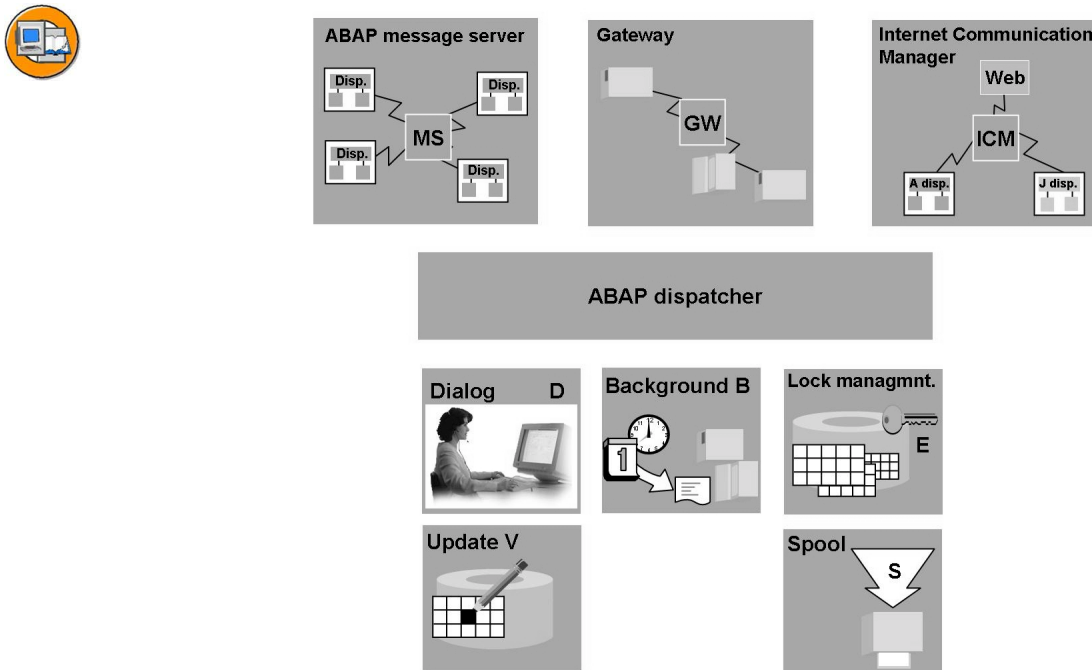


Figure 30: AS ABAP Processes

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

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

To summarize, the dispatcher of an ABAP instance manages different types of work processes: dialog, update, background, enqueue and spool work processes. These work processes work on different tasks when executing the business procedures in the SAP system.

The administrator of an SAP system can use profile parameters to configure the number of different work processes.

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

- The **Message Server** (MS) handles communication between the distributed dispatchers within the AS ABAP, thereby enabling scalability of several parallel application servers. The message server is configured only once per SAP system.
- The **gateway** reader (GW) enables communication between SAP systems, or between SAP systems and external application systems. There is one per dispatcher.
- The **Internet Communication Manager** (ICM) enables the communication with the SAP system using web protocols such as HTTP. The ICM receives requests from the client and forwards them to the SAP system for processing. In an ABAP+Java system (see below), it recognizes whether the request is a call for the AS ABAP or the AS Java and forwards the request accordingly. It can also direct HTTP requests from an SAP system to a Web server and send the response back to the SAP system. You can configure a maximum of one ICM process per application server (software-based view).

AS Java Processes

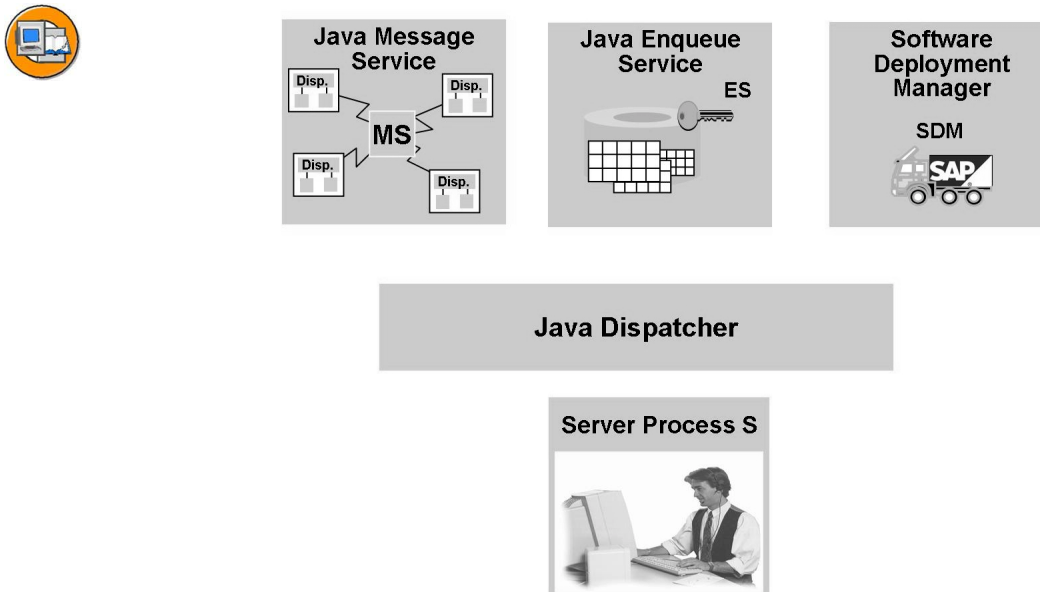


Figure 31: AS Java Processes

The following processes exist in **AS Java**:

- The **dispatcher** distributes incoming requests to the server processes.
- The **server process** executes the Java applications. Every server process is multi-threaded and can thus process a large number of requests in parallel (in contrast to the ABAP work processes). For each dispatcher there is at least one server processes and there can be up to 16 server processes.
- The Java **message service** manages a list of Java dispatchers and server processes. It is responsible for the communication within the Java runtime environment.
- The Java **enqueue service** manages logical locks that are set by the executed Java application program in a server process.
- The **Software Deployment Manager** (SDM) is the standard tool used to install Java software components on the SAP Web AS Java.

Types of SAP NetWeaver AS

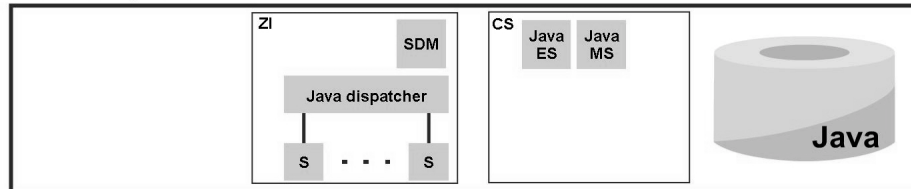
Depending on the application or product used, different types of application server are installed.



AS ABAP System



AS Java System



AS ABAP + Java System

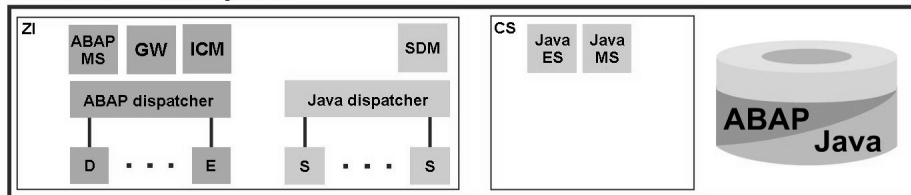


Figure 32: Possible Types of SAP NetWeaver AS

- **AS ABAP system:** Complete infrastructure in which ABAP-based applications can be developed and used.
- **AS Java system:** Complete infrastructure for developing and using J2EE-based applications.
- **AS ABAP+Java system:** Complete infrastructure in which ABAP-based and J2EE-based applications can be developed and used. Such a system should only be installed if explicitly required by the application. For example, SAP NetWeaver PI 7.0 or SAP Solution Manager 4.0

One of the main characteristics of the SAP NetWeaver AS is that ABAP tables, programs and application data is stored in the ABAP schema of the database while Java data is stored in the Java schema. Here, the ABAP runtime environment can access the ABAP schema of the database, and the Java runtime environment can access the Java schema. In the ABAP+Java system, the different runtime environments communicate directly via the SAP Java Connector (JCo).

AS ABAP Architecture

In AS ABAP, the central instance is distinguished by the fact that the message server and the enqueue work process run there. All other instances of the system are usually called dialog instances. Alternatively, the instances are also named after the services provided.

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

➔ **Note:** An overview of the AS ABAP instances is available in SM51 (in SAP Easy Access under *Tools → Administration → Monitor → System Monitoring → Servers*).

You can use the transaction SM50 to display an overview of the work processes on the instance that you are logged on to; you can also display this overview by choosing *Tools → Administration → Monitor → System Monitoring → Process Overview* on the SAP Easy Access screen.

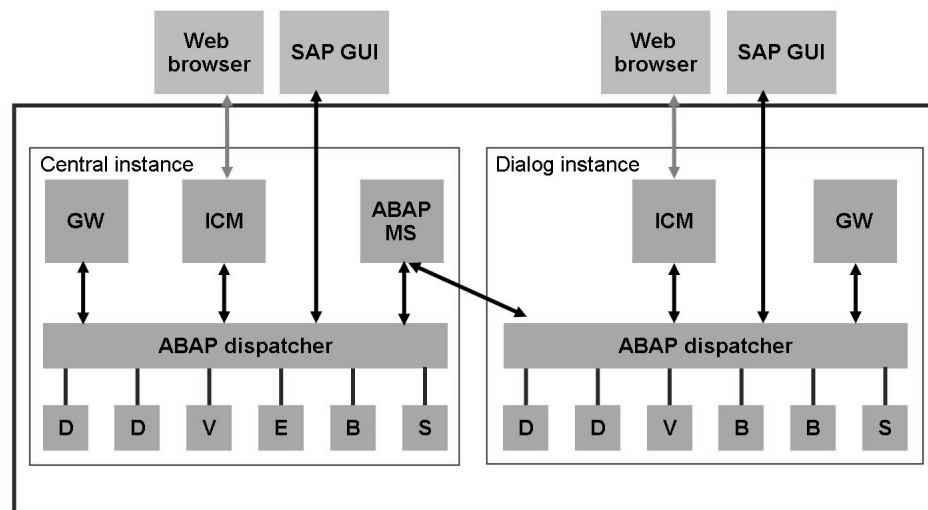


Figure 33: AS ABAP Architecture

The ABAP message server provides the AS ABAP with a central message service for internal communication (for example, for starting updates, requesting and removing locks, triggering background requests). The message server also provides information on which instances of the system are currently available.

The ABAP dispatchers of the individual application servers communicate via the ABAP message server, which is installed exactly once per SAP system.

When you log on to the AS ABAP using the SAP GUI for Windows or the SAP GUI for Java using logon groups, the message server performs a load distribution of users to the available instances. This load distribution, which takes place during the logon procedure, is also known as logon load balancing. After the load distribution by the message server, the SAP GUI communicates directly with the dispatcher. The user remains logged on to this instance until he logs off again.



Note: An overview of users who are logged on the the instance to which you are also logged on, is available using transaction SM04 (*Tools* → *Administration* → *Monitor* → *System Monitoring* → *User Overview*).

You can see to which instance you are logged on under *System* → *Status*.

If you are accessing the AB ABAP via web protocols such as HTTP using the browser, the Internet Communication Manager (ICM) receives the request. This forwards the request to the dispatcher of its instance.

Communication from other SAP systems via Remote Function Call (RFC) is accepted by the Gateway Reader (GW).

AS Java Architecture

In AS Java, the central instance is distinguished by the fact that the Software Deployment Manager (SDM) runs there. The central services Message Service (MS) and Enqueue Service (ES) run in the central services instance (CS instance). All other instances of the system are usually called dialog instances.



Note: The entirety of the Java environment (all processes and the database scheme) is also referred to a **Java cluster**, and the individual processes (dispatcher and server) as **nodes** of the Java cluster.

You can get an overview of started Java processes (Java dispatcher and Java server processes as well as SDM) via the *system information* of the Java runtime environment ((<http://<Hostname>:<Java-Dispatcher-HTTP-Port of the Instance>/sap/monitoring/SystemInfo>, for example <http://twdf1234.wdf.sap.corp:50000> → *System Information*)

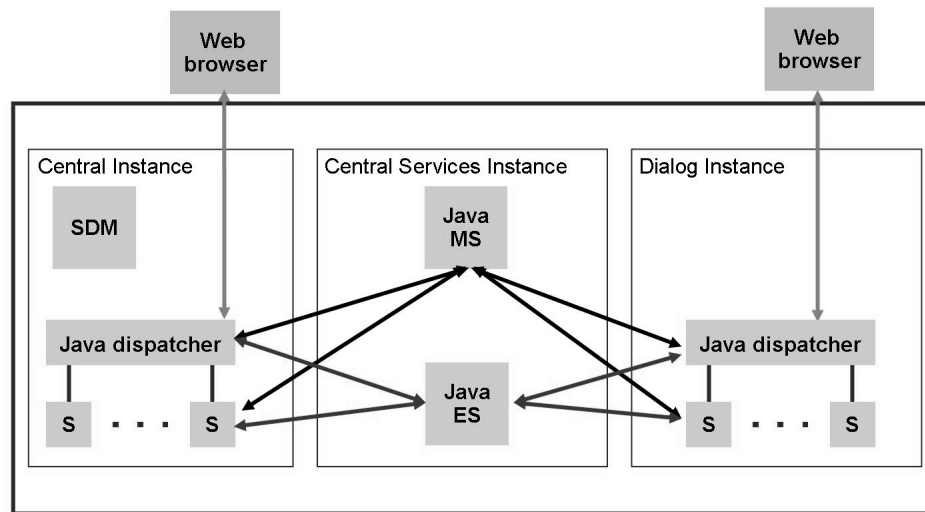


Figure 34: AS Java Architecture

Analogous to the AS ABAP, the message service of the AS Java provides a central message service for internal communication. The Java message service also provides the information which instances and nodes of the AS Java are available. Each node of the Java cluster can communicate directly with the message service.

In the AS Java, the enqueue service holds logical locks. Each node of the Java cluster can communicate directly with the enqueue service.

When the AS Java is accessed using a browser, the Java dispatcher receives requests, which are then processed by the server processes.

AS ABAP+Java Architecture

For the AS ABAP+Java (meaning ABAP and Java processes in the same SAP system, under the same system ID), the same architectural principles apply as for separate AS ABAP and AS Java systems. However, there are some particularities because both runtime environments are integrated with each other in this case.



Note: The AS ABAP+Java is often called “add-in installation” because it is possible to install an AS ABAP first and then supplement it with the AS Java at a later point in time.

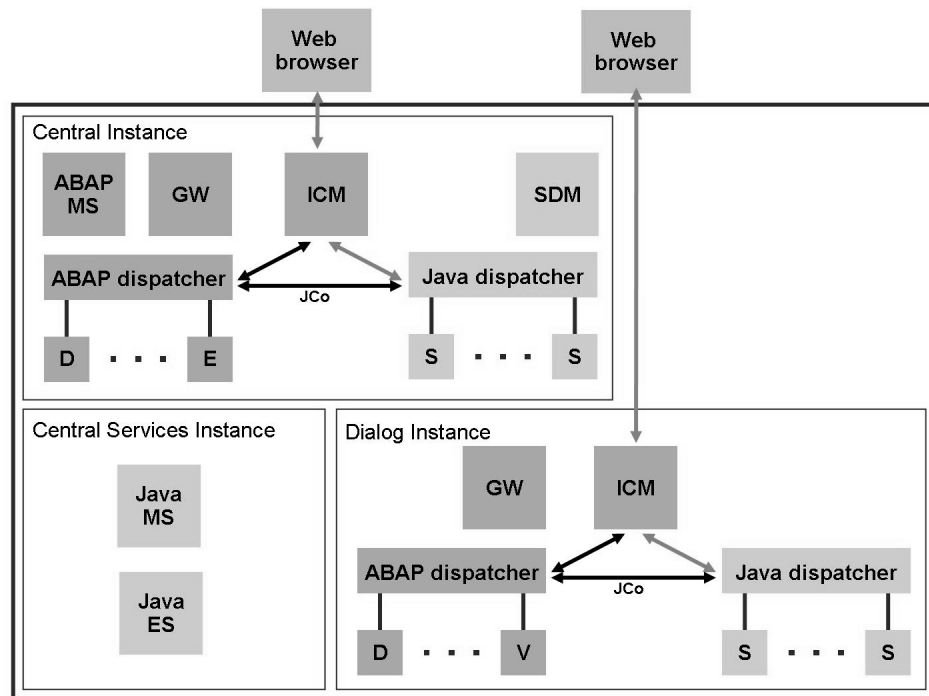


Figure 35: AS ABAP+Java Architecture

The central instance of an AS ABAP+Java system can be recognized by the following processes: ABAP-MS, enqueue work process and SDM. The central services of the Java runtime environment (Java-MS, Java-ES) are also provided in the Java central services instance here. All other instances are usually called dialog instances.

Since both runtime environments are capable of answering requests via web protocols, the Internet Communication Manager must now decide whether the request is addressed to the ABAP or the Java runtime environment. It decides this by means of the URL of the request. In case of a request to the ABAP runtime environment, for example, the call of an ABAP web dynpro, the ICM forwards the request to the ABAP dispatcher and the work processes respond to the request. If the request is a request for the Java runtime environment, for example, the call of a Java Server Page (JSP), the ICM forwards the request to the Java dispatcher and one of the server processes responds to the request.

In an AS ABAP+Java system, data is also kept in separate database schemas (but in the same database installation). That is, work processes can only access ABAP data and server processes can only access Java data. In the data exchange, both runtime environments then communicate using the **SAP Java Connector (JCo)**. This communication is necessary, for example, if billing data that is stored in the ABAP data schema is supposed to be displayed in a Java user interface.

The SAP JCo is integrated into the AS Java and is also used when an AS Java system has to communicate with a remote AS ABAP system.

Exercise 4: Structure of an Instance

Exercise Objectives

After completing this exercise, you will be able to:

- Determine the number of instances of which the SAP system consists
- Find out how many work processes are running and which processes these are
- Find out which AS Java processes are running

Business Example

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

Task: Overview of the Instances Configured on the Training System

Determine the number of instances of which your SAP system consists. Start in AS ABAP.

1. Which instance are you currently working with?
2. Which (AS ABAP) instances does your system have? Which services are offered there? What is the central instance of the system?
3. Which AS Java instances does your system have? Which work processes are running there?
4. About which instance and which processes have you not seen any information so far?

Solution 4: Structure of an Instance

Task: Overview of the Instances Configured on the Training System

Determine the number of instances of which your SAP system consists. Start in AS ABAP.

1. Which instance are you currently working with?
 - a) Log on to the SAP system using the SAP GUI for Windows. Your instructor will supply you with user name and password.
 - b) Choose *System* → *Status* from the menu bar.
 - c) The name of the instance to which you are logged on is available in the *Server Name* field and has the following structure: <computer name>_<SID>_<instance number>.
2. Which (AS ABAP) instances does your system have? Which services are offered there? What is the central instance of the system?
 - a) Call up transaction SM51 (*Tools* → *Administration* → *Monitor* → *System Monitoring* → *Servers*).
 - b) You will see a list of all started AS ABAP instances with the services offered there.
 - c) The instance in which the enqueue work process is configured, is the central instance. In transaction SM51, you can recognize this instance by the *type enqueue*.
3. Which AS Java instances does your system have? Which work processes are running there?
 - a) Start a web browser.
 - b) Call the home page of the AS Java of your SAP system. You reach the homepage at <http://<host name>:<port-number>>, for example <http://twdf1234.wdf.sap.corp:50000>. Your instructor will provide you with the precise information.
 - c) On the AS Java homepage, call up the System Information link.
 - d) Log on with the user and password provided by the instructor.
 - e) You see information about the AS Java instances and processes.

Continued on next page

4. About which instance and which processes have you not seen any information so far?
 - a) Information about the Java central services instance, including Java-MS and Java-ES, is not available using the above-mentioned tools.
 - b) The ABAP-MS and the gateway are not listed in transaction SM51.



Lesson Summary

You should now be able to:

- Outline simple client/server configurations
- Name the processes of the SAP NetWeaver Application Server
- Define the term instance and recognize the characteristics of a central instance

Lesson: AS ABAP Processes

Lesson Overview

This lesson discusses how the SAP NetWeaver Application Server ABAP (AS ABAP) works. The processes are introduced and it is explained how they work.



Lesson Objectives

After completing this lesson, you will be able to:

- Understand how AS ABAP works
- List the AS ABAP processes and describe their purpose
- Describe how requests to AS ABAP are processed

Business Example

Your company has decided to implement SAP ERP. The business core functions such as accounting in SAP ERP are processed by the ABAP runtime environment. For the employees in the different, rather technical departments (development, systems administration, work preparation, security and so on) it is therefore important that they understand how AS ABAP works in more detail.

Processing of Requests in AS ABAP

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

Processing Requests by the SAP GUI

Users logon via the (ABAP) message server (load balancing) or they logon directly on the ABAP dispatcher, the work processes execute the user requests.

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

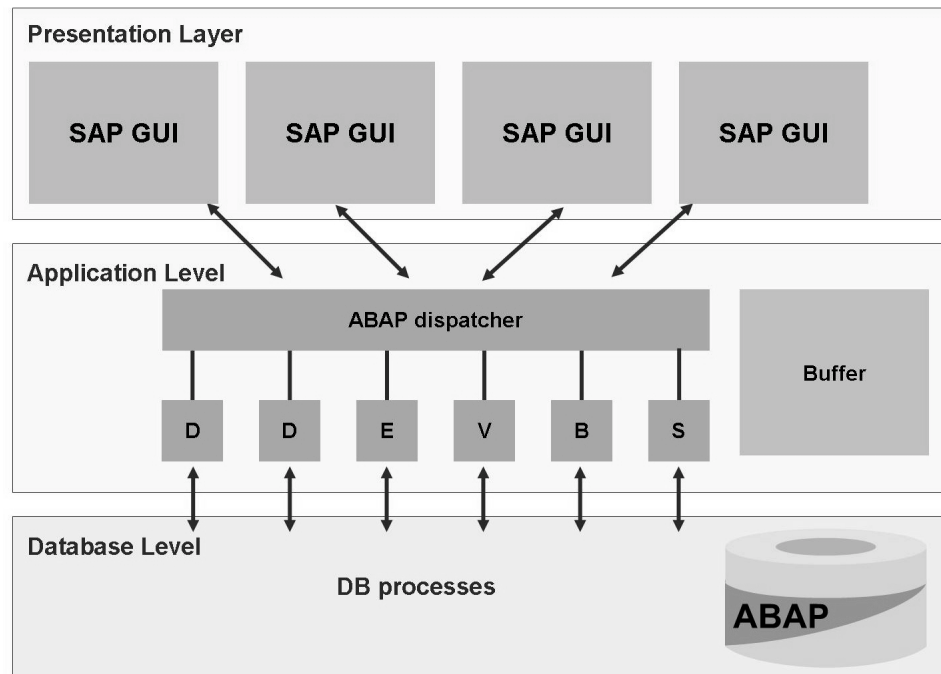


Figure 36: Processing a User Request

The screen entries of a user are accepted by the SAP presentation program **SAP GUI** (SAP Graphical User Interface), converted to an internal format and forwarded to AS ABAP.

The (ABAP) **dispatcher** is the central process of AS ABAP. It manages the resources for the applications written in ABAP in coordination with the respective operating system. The main tasks of the ABAP dispatcher include the **distribution of the requests to “its” work processes**, the integration of the presentation layer and the organization of communication transactions.

The dispatcher first saves the processing requests in request queues and then processes them according to the “first in, first out” principle.

The ABAP dispatcher distributes the requests one after the other to the available work processes. Data is actually processed in the **work process**, whereby the user who created the request using the SAP GUI is not always assigned the same work process. There is no **fixed assignment** of work processes to users. To process user requests it is often necessary to read data from the ABAP schema of the **database** or to write to it. For this, every work process is connected **directly to the ABAP schema of the database**.

Once the process is complete, the result from the work process is sent via dispatcher back to the SAP GUI. The SAP GUI interprets this data and generates the output screen for the user.

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

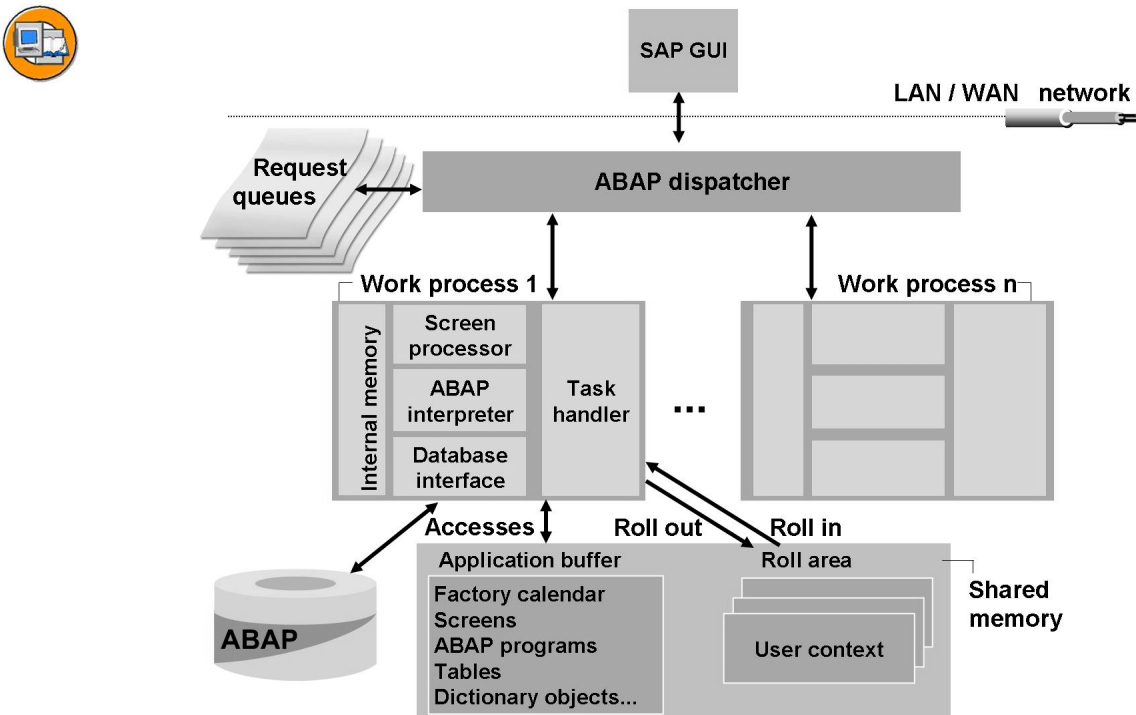


Figure 37: Process Flow for Requests

Work processes execute the process logic of application programs. In addition to internal memory, a work process has a **task handler** that coordinates the actions within a work process, software processors and a **database interface**. The **dynpro processor** executes the screen flow logic of the application program, calls processing logic modules, and transfers field content to the processing logic. The actual processing logic of ABAP application programs is executed by the **ABAP interpreter**. The screen processor tells the ABAP processor which subprogram needs to be executed, depending on the processing status of the screen flow logic.

The dialog work process selected by the dispatcher, performs a roll-in of the user context first. That is, the data that contains the current processing status of a running program as well as data that characterizes the user is made known to the work process. The work process then processes the user request, which may involve, for example, requesting data from the database or from the buffers in the shared memory. Once the dialog work process has processed the dialog step, the work process returns the result, rolls the user context back out to the shared memory, and is now available again for a new user request from the request queue. The result is transferred to the SAP GUI and the user sees the new screen.

Database Interface of AS ABAP

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

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



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

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

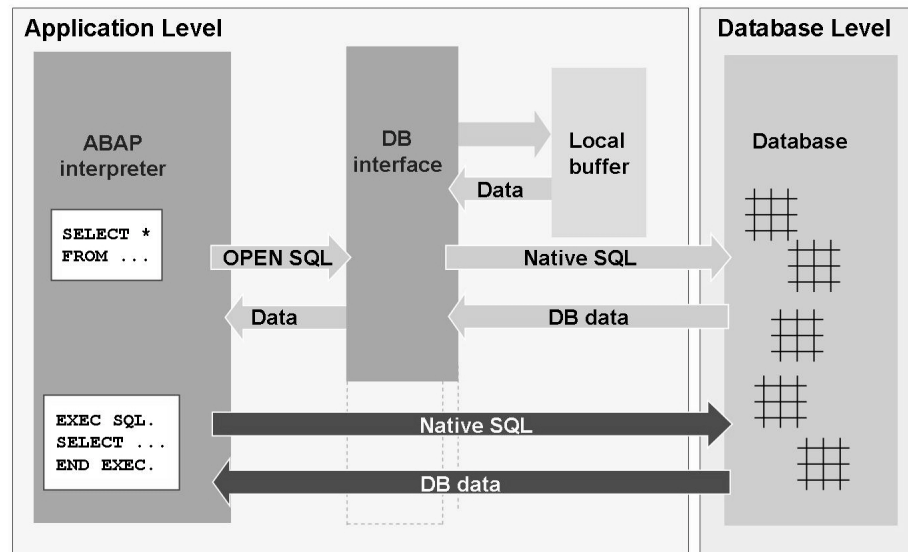


Figure 38: Database Query Flow

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

Processing Dialog Requests

The execution of dialog requests is characterized by the following :



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

This is illustrated by the following figure.

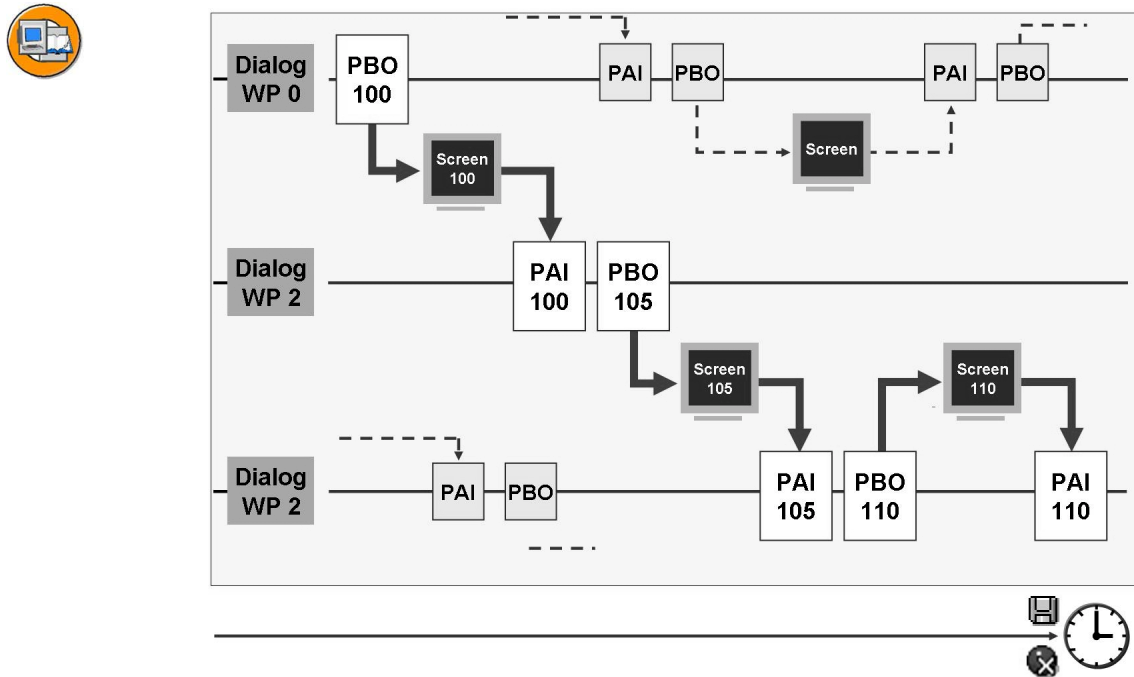


Figure 39: Dialog Work Process Multiplexing

SAP application programs differentiate between user interaction and processing logic. The user actions are technically realized using screens, also called **dynpros** (from dynamic programs), which consist of a screen image and the underlying flow logic. The dynpro processor of the work process executes the screen flow logic of the application program, calls processing logic modules, and transfers field content to the processing logic. The screen flow logic itself is further divided into **PBO** (Process Before Output), which is processed before the screen image is sent, and **PAI** (Process After Input), which is processed after a user interaction on the screen. The PAI part of a dialog step logically belongs to the preceding screen image, while the PBO part logically belongs to the subsequent screen image. The actual processing logic of ABAP programs is executed by the ABAP interpreter. The screen processor tells the ABAP processor which subprogram needs to be executed, depending on the processing status of the screen flow logic.

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

Transactional Processing in AS ABAP

The Term Transaction

Transactions are processing units, that functionally belong together. They have four principal characteristics. The initial letters of these characteristics together form the acronym **ACID**.



- Atomic
- Consistent
- Isolated
- Durable

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

Consistent means that the system status changes from one that is accurate and consistent in business terms to another that is also accurate and consistent in business terms.

Isolated means that the changes made within a transaction can only be seen by other transactions, even those that run simultaneously, after the final confirmation (“Commit”).

The results of a transaction are **durable** because after the final confirmation they are stored permanently in the database.

Database Transactions and ABAP Transactions

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

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

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

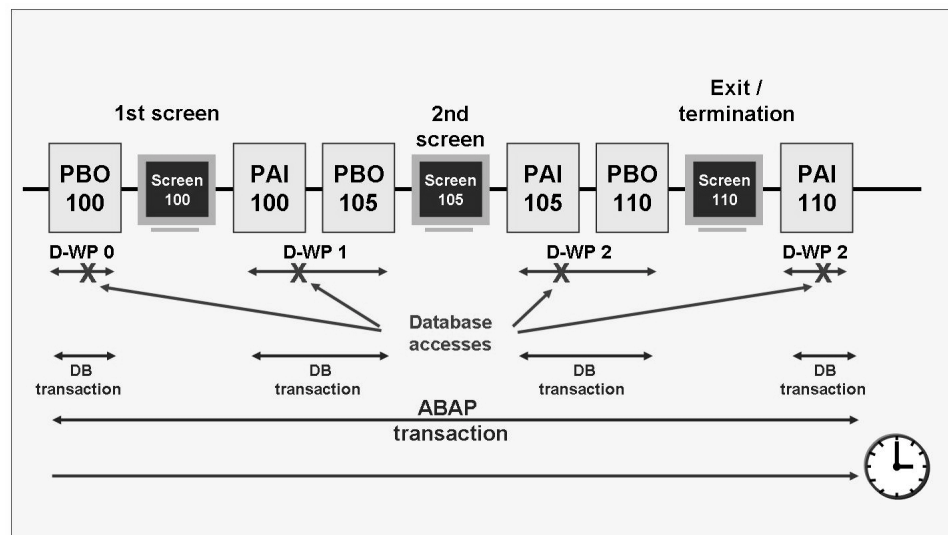


Figure 40: Relationship between database transactions and SAP transactions

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

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

Lock Management

To ensure data consistency within an SAP system, you must ensure that data records cannot be accessed and changed by more than one user at any one time. To do this, the SAP system has its own lock management concept.

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

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

The **enqueue work process** manages the logical lock of the SAP transactions in the **lock table**. The lock table is located **in the main memory** of the instance with the enqueue work process.



Note: The instance whose main memory contains the lock table is also known as the enqueue server.

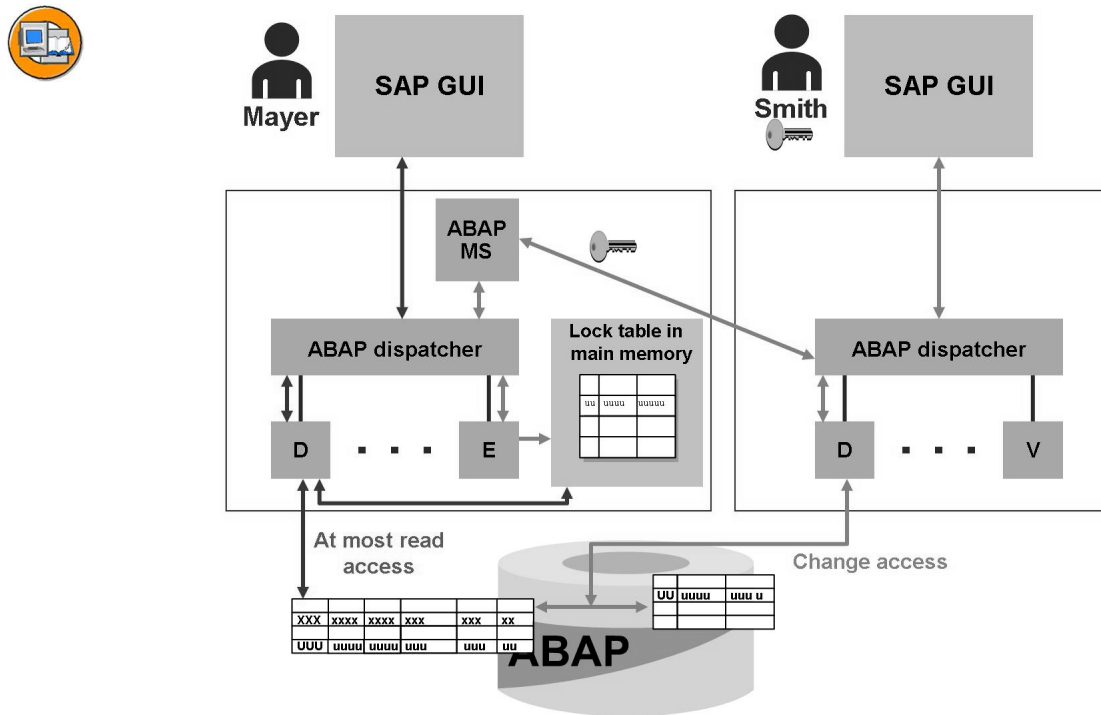


Figure 41: Lock Management in AS ABAP

If a user wants change access to data, the executing dialog work process requests a lock (to do so, the application developer must program this request explicitly).

If a dialog request is processed on the enqueue server, the dialog work process can access the lock table directly. It now checks whether a new lock can be generated; that is, whether there is a collision with locks that have already been set. If a lock can be set, the dialog work process creates it and the user (lock owner) is given the lock key. The lock key is kept in the user context in the shared memory.

If the dialog work process that processes the user request and the enqueue work process are not running on the same instance, these two work processes communicate through the message server. In this case, the lock request is forwarded from the dialog work process to the enqueue work process via the dispatchers and the message server. The enqueue work process now checks whether a lock can be set. If this is possible, the lock is set by the enqueue work process and the lock key transferred to the requesting dialog work process via dispatcher and message server.

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

The application developer can choose between different lock modes:

- Write locks (lock mode **Exclusive**); the lock data can be edited only by one user. The requests for another write lock and another read lock are rejected. A write lock protects the locked objects against all types of other transactions. Only the same lock owner can set the lock again (cumulate).
- Read locks (lock mode **Shared**); several users can have read access to the locked data at the same time. The requests for additional read locks are accepted, even if they are from other users. A write lock is rejected.
- Enhanced write locks (lock mode **eXclusive noncumulative**); while write locks can be successively requested and released by the same transaction, an enhanced write lock can only be requested once, even by the same transaction. All other requests for locks are rejected.
- Optimistic locks (lock mode **Optimistic**); optimistic locks respond like read lock at first and can be changed to write locks. An optimistic lock is set if the user displays the data in change mode. Optimistic locks on the same object do not collide. If the user wants to save the (changed) data, the optimistic lock must be changed to a write lock (mode E). (This fails if someone set a non-optimistic lock on the object before.) Other optimistic locks on the object are deleted in the process.

Locks set by an application program are either released by the application program itself or by the update program once the database has been changed. Locks that have been passed on to an update work process in this way are also written to a file at operating system level and can therefore be restored if the enqueue server goes down.

Transaction SM12 (*Tools → Administration → Monitor → Lock Entries*) displays the locks that are currently set. If a lock has already been inherited to the update process, the *backup flag* has also been set. Such a lock will also be included in the lock table again after restarting the enqueue server.

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

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

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



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

Standalone Enqueue Server / ABAP Central Services

For reasons of high availability, the enqueue work process together with the ABAP message server can also be detached from the central instance and installed as an **ABAP central services** instance (ASCS). For more details, please refer to the SAP NetWeaver Library: *SAP NetWeaver by Key Capability* → *Application Platform by Key Capability* → *ABAP Technology* → *Client/Server Technology* → *The SAP Lock Concept (BC-CST-EQ)* → *Standalone Enqueue Server*

Update

In the SAP system, a business process is mapped using an SAP transaction that can contain several screen changes (for example, the creation of an order). Data changes effected by this process are supposed to be executed completely or not at all in the database. If the operation is terminated during the transaction or an error occurs, the transaction is not supposed to make any database changes at all. The SAP update system, which is described below, takes care of this.

The update system also offers increased security, performance and restorability in the execution of database changes.

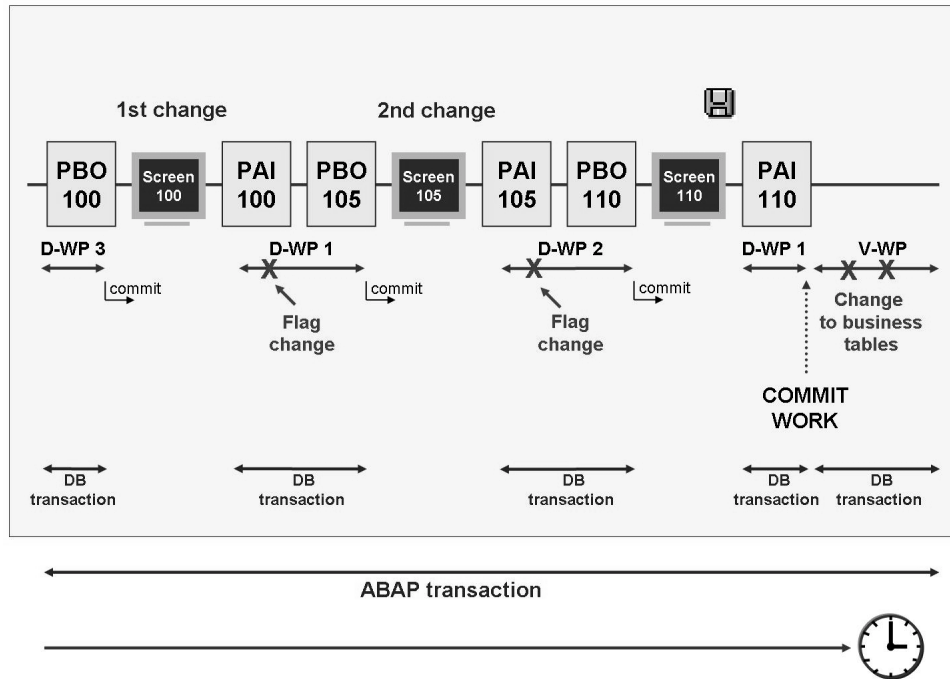


Figure 42: The Principle of Asynchronous Updates

The updating system is a technology which allows SAP transactions to off-load time-intensive database changes. These are then carried out **asynchronously** in special **update work processes**. It also circumvents the roll-back problems caused by the difference in the conception of the logical unit of work (LUW) in an SAP transaction and in the database.

If, during a dialog work process, data temporarily stored for processing is passed to an update work process for further processing, the dialog work process does not wait for the update request to be completed: the update is **asynchronous** (not simultaneous). The asynchronous update process is illustrated in the graphic “The Principle of Asynchronous Updates”.

The dialog part is completed with the ABAP command *COMMIT WORK*; the update part of the transaction starts: the update server transfers the update request to an update work process. Here, each dialog step corresponds to a database transaction (which is executed either completely or not at all in the database and there completed with a *COMMIT* command). The update part of the SAP transaction is executed in a database transaction. It is only then that the data is copied to the application tables.

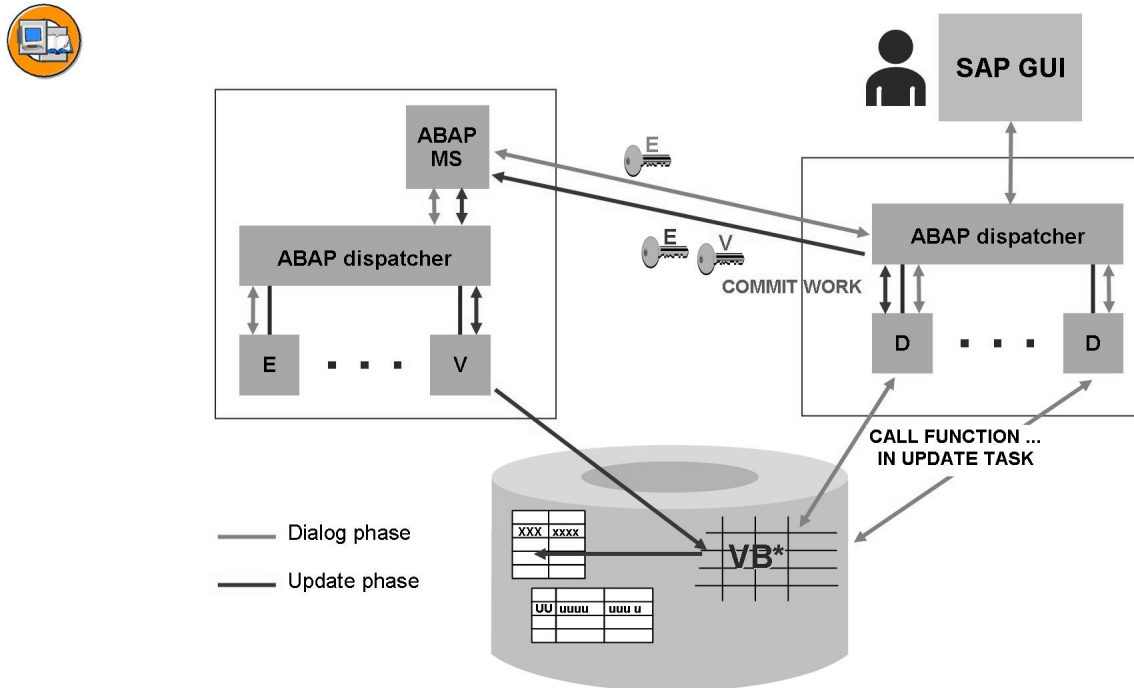


Figure 43: The Asynchronous Update Process

If users want to change a data record in an SAP transaction, they call the corresponding transaction in the dialog, make the appropriate entries on the screens and then initiate the update process by saving the data. This process triggers the following steps:

1. The program locks the data record for other users. The program does this by addressing the enqueue work process (using the message server if appropriate). The enqueue work process makes the relevant entry in the lock table or (if another user has already locked the data) informs the user that the data record cannot currently be changed.
2. If the enqueue work process succeeded in writing the lock entry to the lock table, it passes the lock key it created to the user, the program reads the record to be changed from the database and the user can change the record on the screen image of the SAP transaction.
3. In the active dialog work process, the program calls a function module using *CALL FUNCTION ... IN UPDATE TASK* and writes the change request to database update tables. These are also called VB* tables, because their names begin with "VB". They act as temporary memory and store the data to be changed until it can be collected and written to the application tables in the database (in a single database transaction).

4. At the end of the dialog part of the transaction (for example, when the user saves the data – possibly after completing other dialog steps), the program initiates the close of the transaction with the *COMMIT WORK* statement. The work process that is handling the active dialog step completes the update header and triggers an update work process.
5. Based on the information (key of the update order, lock key) transferred from the dialog work process, the update work process reads the log records that belong to this SAP transaction from the VB* tables.
6. The update work process transfers the changes marked and collected in the VB* tables to the database as a change request and evaluates the database response. If the changes were successfully written to the target tables, the update work process triggers a database commit after the last change to the database and deletes the entries from the VB* tables. If an error occurs, the update work process triggers a database rollback, leaves the log records in the VB* tables and marks them as defective.
7. The lock entries in the lock table are reset.



Note: The application developer decides whether and how to use asynchronous updates while programming the transaction. Besides the asynchronous update, there are some other update techniques (for example, synchronous or local).

To increase performance further, application developers can configure different types of updates:

- time-critical, primary **V1 updates**. They are relevant to objects that have a controlling function in the SAP system, such as a change to the material stock or an order creation.
- non-time-critical, secondary **V2 updates** that depend on the V1 updates. These are, for example, purely statistical updates such as the calculation of results.
- non-time-critical updates that are collected and processed at a later point in time (**collective run**).

The V1 modules for an SAP transaction are processed sequentially in a single update work process. If your SAP system has a work process for V2 updates (type UP2), then V2 modules will only be updated there. Once it has successfully completed processing, the V1 update work process releases the relevant locks again. This means that the “normal” update work processes are available again more quickly for time-critical V1 updates, and that the relevant lock entries are deleted sooner. If you have not configured any V2 update work processes, then the V1 work process handles all updates.

In the collective run, the modules are not updated automatically but only when a special report (report *RSM13005*) triggers the update. All calls of the function modules are then collected, aggregated and updated at once. In doing so they are handled like V2 update modules.

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

If a dialog work process terminates whilst writing data to the VB* tables, the tables will contain data that will not be updated. These entries can be automatically deleted the next time you start the system or they can be deleted manually. The application tables remain unchanged.

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

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

Printing

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

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

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

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



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

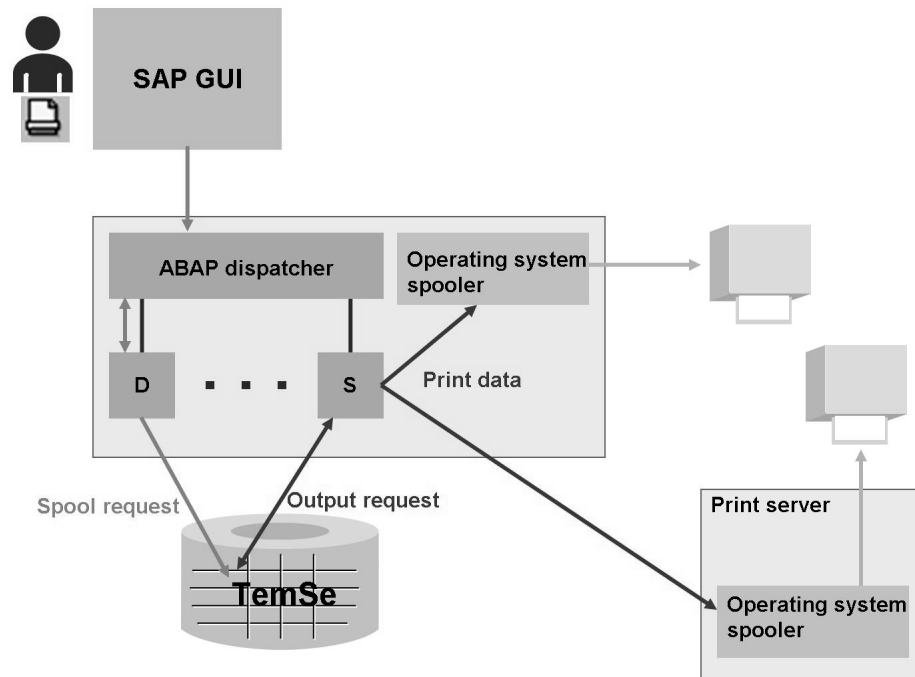


Figure 44: Printing in AS ABAP

As you can see from the graphic **Spool processing in an SAP system**, a spool work process formats the data specified in the spool request and creates an **output request**. The output request contains all data in an appropriate format for the printer. This data can either be passed on to an appropriate operating system spool process **locally** (on the same computer) or **remotely** (over a network connection).



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

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



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

You can display your own spool and output requests via *System → Own Spool Requests* (transaction SP02).

Via *System → User Profile → Own Data* (transaction code SU3) you can specify personal settings for printing on the *Defaults* tab page in the *Spool Control* section.

Background Processing

SAP background processing is a method for automating routine tasks and for optimizing the use of your organization’s SAP computing resources. In background processing, you instruct the R/3 System to run programs for you. You can use background processing to execute long-running or resource intensive programs at off peak times. You can use it to assign the system the task to execute reports and programs. There is no strain on your dialog resources and reports running in the background are not subject to the runtime restrictions of dialog processing (termination of the program after a runtime of ten minutes).

The segregation of background processing to special work processes gives you an additional dimension for separating background processing and interactive work. Normally, background processing and interactive work in the system take place at

different times. For example, the system is used interactively during the day and background processing takes place at night. You can use **background work processes** to separate background processing and interactive work also by servers because background jobs are only executed on servers that offer background processing.

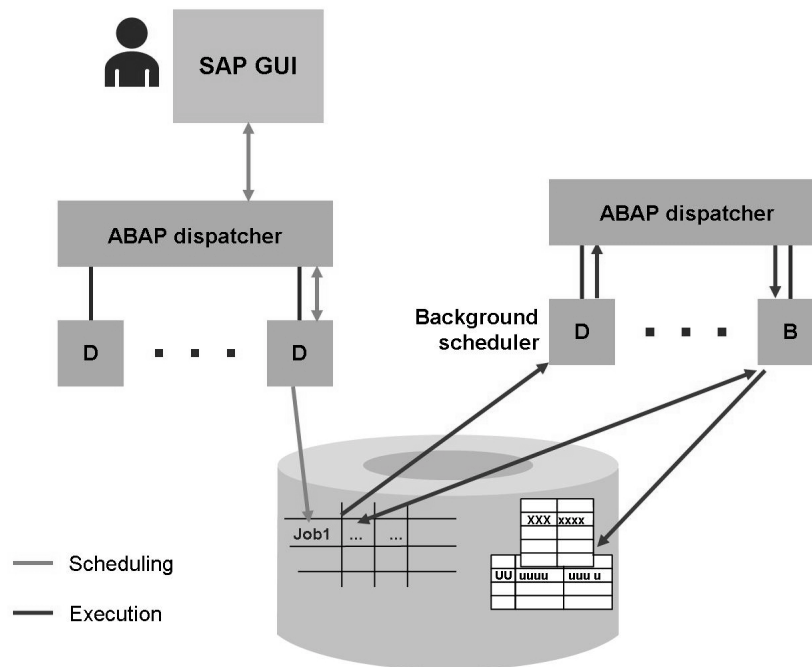


Figure 45: Scheduling Background Tasks (Jobs)

The end user can usually schedule the program to be started in the background as a **job** from the application transaction. The job then “waits” for the planned execution time in the job scheduling table. If the time has come and free background work processes are available, the job is distributed to a background work process by the background scheduler and then executed. Users can display the result in the application transaction or, in case of list generating programs, look at the spool request belonging to the job (see *Printing* section). To display your own jobs, choose *System → Own Jobs* (transaction code SMX).

The system administration and the work preparation have access to a tool for scheduling different types of background takes with transaction SM36 (*Tools → CCMS → Background Processing → Define Jobs*). The system-wide monitoring of jobs takes place with transaction SM37 (*Tools → CCMS → Background Processing → Jobs - Overview and Administration*).

For cross-system scheduling and monitoring of background tasks, you can use SAP Central Process Scheduling (SAP CPS) and other licensed partner products.

Communication via the Gateway

Each instance of an AB ABAP system contains a gateway. This is used for the communication between work processes of different instances or SAP systems as well as work processes and external programs. The **gateway reader** (usually just called gateway) is the main process of the gateway system. The dispatcher starts it and checks it periodically.

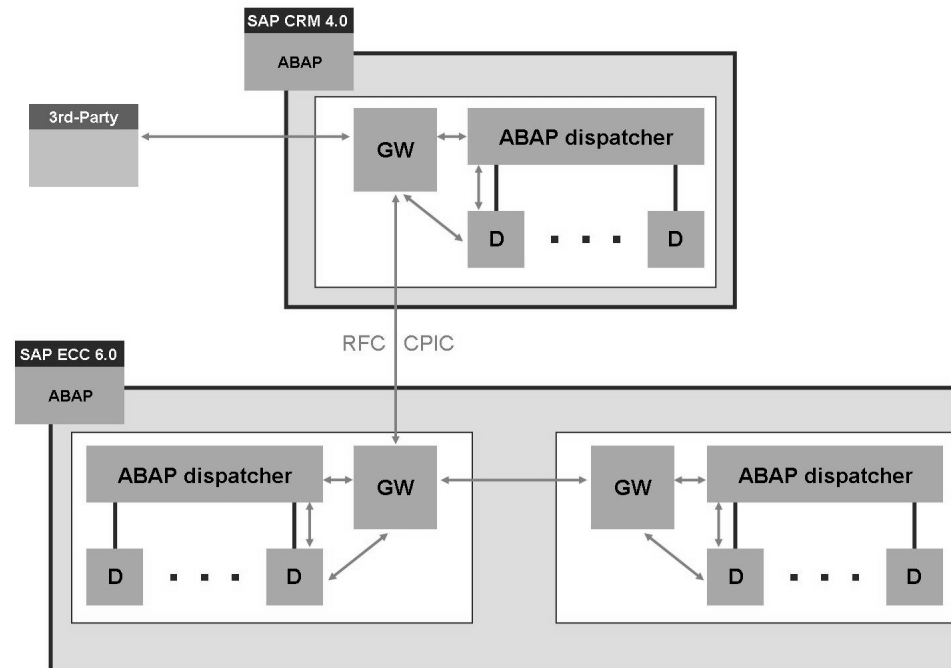


Figure 46: Gateway Communication

In the communication between instances of a system or systems using remote function call (RFC) or CPIC, the gateway is always involved. If a dialog work process has to establish an RFC connection to a remote system in the context of a request, for example, to fetch customer data, it uses the gateway, which then takes care of the communication with the remote system. The gateway forwards the request to the gateway of the remote system. The remote gateway transfers the request to the dispatcher. This in turn forwards the request to one of its work processes, which then communicates directly with “its” gateway.

Inbound RFC connections are therefore always received by the gateway. Outbound connections are initiated by the work process.

Processing Web Requests

Web requests are received by the **Internet Communication Manager (ICM)**. These HTTP(S) requests can be processed either in the ABAP work process (for example, ABAP web dynpro applications) or forwarded to AS Java in an AS ABAP+Java system (for example, Java web dynpro applications). The ICM can use the URL to decide to where it forwards the request (if it cannot reply to the request from its cache).

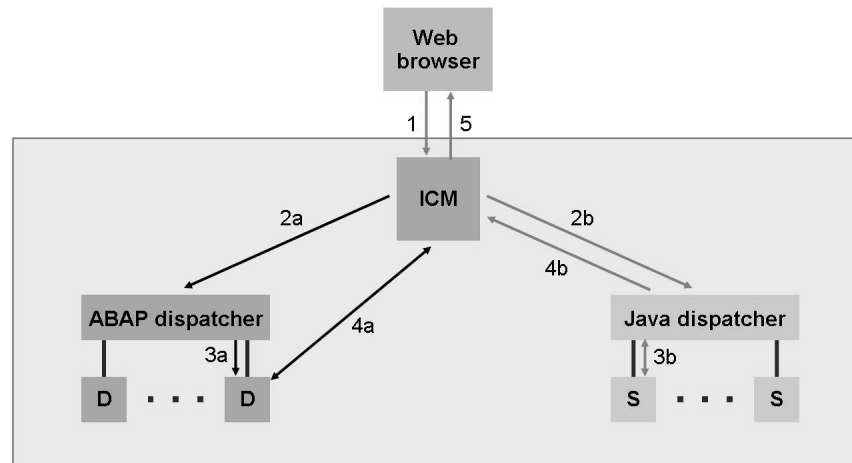


Figure 47: Processing a Web Request

If the request is to the Java runtime environment, it is sent to the Java dispatcher (2b), which then forwards it to a Java server process (3b). The Java server process then refers to the database's Java schema, if applicable, and uses the Java dispatcher process to send the response back to the ICM. Finally, the ICM sends the response from the SAP system to the user who sent the request (5).

If the request is to the ABAP runtime environment, the ICM forwards it to the ABAP dispatcher (2a), which then handles it like a typical SAP GUI request (see previous section). The work process that processes the enquiry now communicates directly with the ICM (4a). The ICM returns the response to the user who sent the request (5).

Exercise 5: Dialog Processing

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

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

Task: Overview of the configured work processes

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

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

Solution 5: Dialog Processing

Task: Overview of the configured work processes

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

1. How many dialog work processes are there on the instance that you are logged on to?
 - a) You can find the number of dialog work processes on your instance by using transaction SM50, then counting the work processes of type *DIA*. Transaction SM50 displays the work processes on an instance. If you want to display the work processes for the entire SAP system, choose transaction SM66 (in this case you need to make the appropriate settings using the *Select process* and *Settings* buttons).
2. Choose Refresh several times in the process overview. Are your queries always processed by the same work process?
 - a) In transaction SM50, choose the *Refresh (F8)* pushbutton. Find “your” work process by finding your user name in the *User* column. You may find – although you might not – that your queries are processed by different work processes.

Exercise 6: Optional: Lock Management

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

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

Task: Setting and Monitoring Locks

Learn about how lock administration works in SAP systems.

1. Change your own user data.
2. Open a new session and check the list of lock entries to determine on which tables a lock has been set.
3. In another session, try changing your own user data again.
4. End the session that changes your user data. Are the lock entries still there?

Solution 6: Optional: Lock Management

Task: Setting and Monitoring Locks

Learn about how lock administration works in SAP systems.

1. Change your own user data.
 - a) Call transaction SU01 for user maintenance (menu path *Tools* → *Administration* → *User Maintenance* → *Users*), enter your own user name and choose *Change*. Your user data is now exclusively locked for you at SAP system level.
2. Open a new session and check the list of lock entries to determine on which tables a lock has been set.
 - a) To open a new session choose *System* → *Create Session*, for example.
 - b) Call up transaction SM12 (menu path *Tools* → *Administration* → *Monitor* → *Lock Entries*). Choose *Enter*. You see a lock entry for your user for a data record of table *USR04*.
3. In another session, try changing your own user data again.
 - a) Open a second session (*System* → *Create Session*) and start transaction SU01 there as well. Enter your user name again and choose *Change*. The system displays the message that no other simultaneous session in the system can access your user data to change it (not even if the other session also belongs to your user). However, if you choose *Display*, then your user data is displayed.
4. End the session that changes your user data. Are the lock entries still there?
 - a) To end your user change session, start transaction SM04 (menu path *Tools* → *Administration* → *System Administration* → *Monitor* → *System Monitoring* → *User Overview*). Select your user with the mouse and choose the *Sessions* pushbutton.
 - b) Position the cursor on the *User List* line and choose *End session*.
 - c) If you now select your current lock entries using transaction SM12, the lock on table *USR04* has disappeared. If an action terminates in an SAP system (whether deliberately or not), the locks held by it are deleted.

Exercise 7: Optional: Update Processing

Exercise Objectives

After completing this exercise, you will be able to:

- Call important functions of transaction SM13 (Update Requests)

Business Example

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

Task: Update records to Be Processed

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

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

Solution 7: Optional: Update Processing

Task: Update records to Be Processed

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

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

Exercise 8: Printing

Exercise Objectives

After completing this exercise, you will be able to:

- Print a screenshot in AS ABAP
- Explain the difference between spool requests and output requests

Business Example

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

Task: Printing a Simple List

Print out the list of work processes.

1. Call transaction SM50 and print the list.
2. Display your own spool requests and generate an output request from the spool request.

Solution 8: Printing

Task: Printing a Simple List

Print out the list of work processes.

1. Call transaction SM50 and print the list.
 - a) Call transaction SM50 (*Tools* → *Administration* → *Monitor* → *System Monitoring* → *Process Overview* and choose *Print (Ctrl+P)*
 - b) Use the *F4* help to select an *Output Device* such as **LP01** and choose *Continue*.
 - c) Choose *Continue* to confirm the dialog box.
2. Display your own spool requests and generate an output request from the spool request.
 - a) To do this, choose *System* → *Own Spool Requests*. Check the status and the number of pages of your spool request.
 - b) Select your output request and choose *Print directly* to create an output request for the spool request you selected.
 - c) Select your spool request and choose *Output Requests* to display the output requests for your spool request. The status text is “Waiting for output formatter”. Since there is no printer connected to the training system, you will not be able to print.

Exercise 9: Background Processing

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

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

Task: Scheduling and Monitoring Jobs

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

1. In transaction SA38 (*System* → *Services* → *Reporting*) select report *RSUSR000* in dialog mode.
2. Use transaction SA38 to execute report *RSUSR000* in the background. Display the result of the job.

Solution 9: Background Processing

Task: Scheduling and Monitoring Jobs

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

1. In transaction SA38 (*System* → *Services* → *Reporting*) select report *RSUSR000* in dialog mode.
 - a) To do this, enter the report name **RSUSR000** on the initial screen of the transaction, then choose *Execute (F8)*.
 - b) The users who are logged on are listed.
2. Use transaction SA38 to execute report *RSUSR000* in the background. Display the result of the job.
 - a) To do this, enter the report name **RSUSR000** on the initial screen of transaction SA38, then choose *Execute (Shift+F8)*.
 - b) Choose *Execute Immediately*.

You can also choose *Schedule*. In this case, make sure that the *Job Name* and *Execution Time* are entered. The execution time should not be too far in the future (about 2 minutes, for example) so that you can see the result of the job as soon as possible. Then choose *Schedule Once*.

- c) To monitor your job, you can select *System* → *Own Jobs* in the menu bar. You should see the job you just scheduled in the displayed list.

If the job is in *status released*, wait until the execution time you selected has passed.

If the job is in *status complete*, it was completed successfully and you can display the result in the form of a spool list. To do so, select the job and then choose *Display Spool List*. In the next screen, select the spool request and choose *Display Content*.

You will see a list of users logged on at the time the job was executed.



Lesson Summary

You should now be able to:

- Understand how AS ABAP works
- List the AS ABAP processes and describe their purpose
- Describe how requests to AS ABAP are processed

Lesson: Fundamental Concepts of Java

Lesson Overview

This lesson provides a short overview of the fundamental Java concepts for all participants that have not yet been confronted with Java. The architectural concept and the properties of Java are briefly outlined.



Lesson Objectives

After completing this lesson, you will be able to:

- Use basic Java terminology

Business Example

As of SAP Web Application Server 6.20, you have the option of using Java as well as the ABAP programming language. You should therefore familiarize yourself with the fundamental concepts of the Java environment.

Introduction

If you search for the meaning of the word Java in an encyclopedia, you find the following definitions, among other things:



- One of the four main islands of the Indonesian republic in the Indian Ocean.
- A programming language developed by the company Sun Microsystems.
- An aromatic type of coffee, particularly used to make espresso.
- A popular dance of the 1920s.
- A coarse, loose mesh canvas for embroidery, made of linen or cotton.

It would undoubtedly be worthwhile to investigate each of these definitions individually in more detail. In this course, we want to concentrate exclusively on the second definition in this list.

The Java programming language was first introduced by the company Sun Microsystems Inc.™ in 1995. Java is an object-oriented and platform-independent programming language that is widely used in many areas. The powerful concept of Java allows the development of a large number of different types of application - from the classic application using Applets implemented in Web pages to client/server applications.

Java was designed to be platform-independent. This means that the same Java program can be executed on any system for which a **Java Runtime Environment** (JRE) is available. In particular, Java applications can run on different systems that are based on different hardware. Java can be used to create programs for UNIX, Microsoft Windows, Linux, MacOS, or OS/2, for workstations or servers; for x86, MIPS, Alpha, or Sparc; for computers, organizers, or cell phones, and for the micro computers built into household and industry devices, such as washing machines, video recorders, cars, and traffic lights.

Properties of Java

You need to differentiate between two basic types of Java programs: applications and applets.

- **Applications**

Java applications are computer programs with the full functional range also provided by other programming languages. Applications can run as local programs on the user's computer or as client/server systems across the Internet or using an Intranet, or as server programs (servlets, CGI programs) on a Web server.

- **Applets**

The term applet essentially means “little application”. It usually means a Java applet, a small computer program that runs in a Web browser and is written in the Java programming language. Java applets are intended for execution within the Java Runtime Environment of a browser. They are transferred (as is also the case, for example, with embedded images) with the HTML page using the HTTP protocol, and then executed on the client computer. Special security guidelines apply to Java applets; that is, they are not permitted to perform everything on the client that locally-installed programs are permitted to. In this way, for example, they can only create network connections to the host from which they were downloaded.



Hint: You need to be careful when using the name “Java”. Not everything that has Java as its linguistic root is actually connected to Java; JavaScript is completely unrelated to Java. JavaScript is a script language that can be embedded in HTML and, with some Web browsers (Netscape, Microsoft Internet Explorer), produces the execution of certain functions and actions within the Web browser. JavaScript was developed with the intention of dynamically organizing static HTML pages. Unlike server-side script languages such as Perl or PHP, JavaScript is executed on the client. In contrast to Java, JavaScript is not an independent programming language, is not independent of the browser version, and does not have the required security mechanisms.

Initially, Java is a programming language like any other. Unlike traditional compilers for a programming language, which generate machine code for a specific platform (see the Compiled Programming Languages figure), the Java compiler produces program code for a virtual machine (the **Java Virtual Machine**). This is known as bytecode. A virtual machine is a model of a processor (which often does not exist as real hardware) and the associated system architecture. It is a computer program

that emulates a processor. The effort required to transfer this software to other real CPUs is small in comparison. The virtual machine is therefore viewed as the interface between Java and the actual hardware. It must therefore be developed for each processor architecture and, as such, is the only platform-dependent component of a Java development system (see the figure Concept of the Java Virtual Machine).

SAP also uses the concept of a cross-platform virtual machine in the ABAP world. The ABAP programs of the SAP system are also converted into a bytecode, which is interpreted by the ABAP Virtual Machine, the ABAP work process.

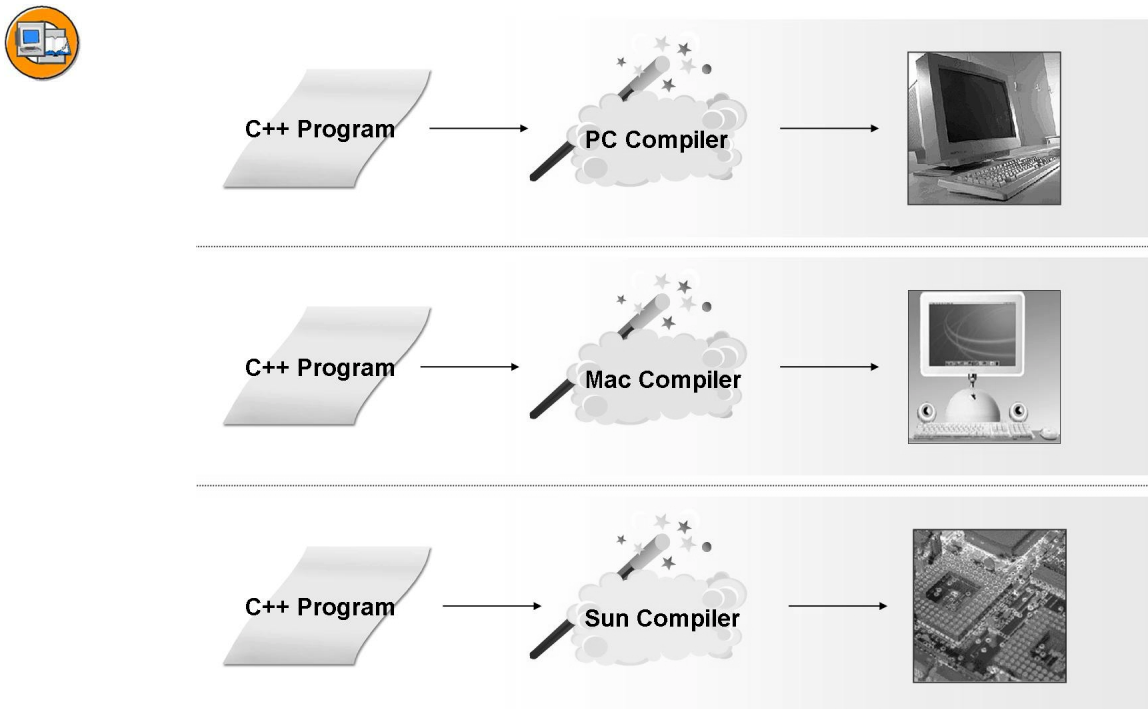


Figure 48: Compiled Programming Languages



Hint: Virtual machines play an important role nowadays, since Microsoft has followed Sun's example of the Java Virtual Machine (Java VM) with its .NET architecture. The concept of the virtual machine was already in use in the late 1960s; Martin Richards' O code for the BCPL programming language is an early example. Better-known examples are the Pascal P code system (UCSD Pascal) from the 1970s, and the virtual machine on which programs in the Smalltalk programming language run.

The bytecode generated by the Java compiler is comparable to microprocessor code for a conceived processor, which understands instructions such as arithmetic operations, gotos, and so on. A Java compiler, such as the one from Sun, which is itself implemented in Java, generates this bytecode. However, so that the program code of the virtual processor can be executed, after the conversion phase, the runtime environment (also known as the runtime interpreter), the Java Virtual Machine, runs the bytecode. In this way, Java is a compiled, but also an interpreted programming language.

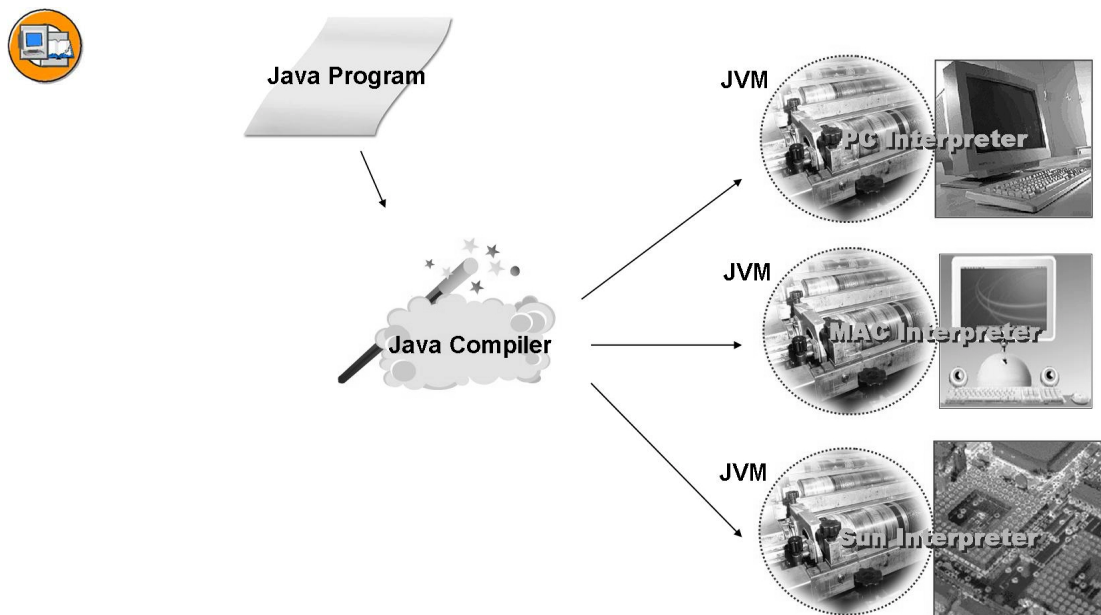


Figure 49: Concept of the Java Virtual Machine

The interpretation causes speed problems, since the identification, decoding, and execution of the commands takes time. Java programs are generally slower than programs that were converted specifically for the target environment (such as C(++) programs). The technology of the Just In Time (JIT) compiler alleviates the problem. A JIT compiler accelerates the execution of the programs by converting the program statements of the virtual machine for the physical machine. There is then a program adjusted for the architecture available in memory, which is executed quickly without interpretation. Even with this technology, the speed is slower than that of C in many cases, but the difference is smaller.

To create cross-platform Java programs, a **Java Development Kit (JDK)** is required. The Java Development Kit includes the software required to create and test Java applications and applets, the packages with the Java classes that are part of the basic configuration, and the online documentation. The software includes the Java

compiler, the Java Runtime Environment (the Java Virtual Machine) for running applications, the applet viewer for running applets, a Java debugger, and various utilities. The online documentation describes all language elements and all classes of the Application Program Interface API.

Java is a relatively new programming language and is therefore still in development, that is, new versions with extensions and improvements are constantly being released: the original version is JDK 1.0 (1995). Since Version 1.2, the JDK is also known as “Java Platform 2”, and the “**Java 2 Software Development Kit (SDK)**” then means the current JDK 1.x in each case, divided into three areas:

- **Java 2 Standard Edition (J2SE).** The Standard Edition is the normal software environment and defines the Java 2 Software Development Kit, abbreviated to SDK. The term JDK (Java Development Kit) is still commonly used. Although this was once outdated it is now being used again in successor versions.
- **Java 2 Enterprise Edition (J2EE).** The Enterprise Edition is an add-on to J2SE and integrates additional packages such as Enterprise Java Beans, Servlets, JSP, Java-Mail-API, JTS.
- **Java 2 Micro Edition (J2ME).** The Micro Edition is a small runtime environment for small PDAs or telephones. A reference implementation is available for PalmPilots. J2ME replaces Personal Java and Embedded Java.

The following figure “SDK Structure” describes the interaction between the terms “J2EE SDK”, “J2SDK” and “J2RE”.

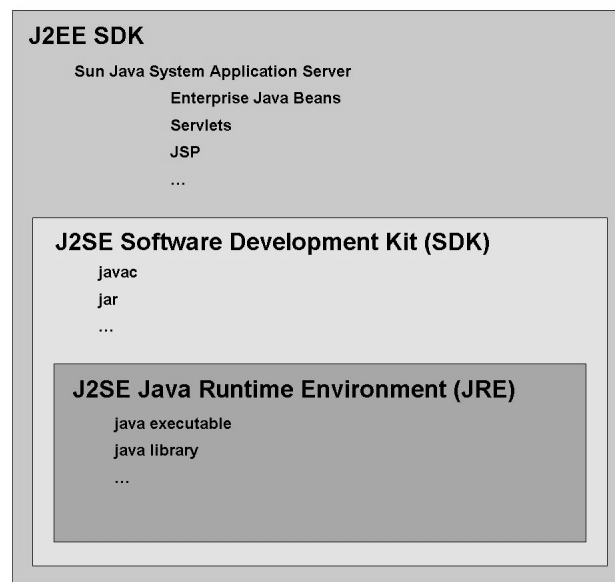


Figure 50: SDK Structure

The J2SE JRE is also referred to as J2RE and contains the runtime environment to which the JVM and other libraries with which Java applications can be executed belong. J2RE cannot be used to develop any Java programs. To do this you need J2SE SDK which is also referred to as J2SDK. It contains J2RE, the Java compiler “javac” and other tools which are of use for programming. When performing client-server programming, you need a server. The J2EE SDK contains not only the J2SDK but also the reference implementation of an application server, i.e. the “Sun Java System Application Server”. Using J2EE SDK, it is possible to program, among other things, “Enterprise Java Beans”, “Servlet”s and “JSP”

SAP NetWeaver AS Java is the SAP implementation for a Java-based application server for which J2SE SDK is a prerequisite. J2EE SDK is not required for SAP NetWeaver AS Java.

Programming in Java

When creating Java programs, the developer first creates a **source file** in the format “MyProgram.java” (see the figure Tools of the Java Development Kit).

The Java compiler generates a separate file for each **class** from the source file that contains the bytecode for this class. The file name is then the name of the class with the extension “.class”, and therefore has the format “MyProgram.class”. In accordance with the object-oriented approach, Java “functions” are grouped in classes. Each Java source file may only contain one public class. The file name of the source file must then be the name of this class (which is case-sensitive) with the extension “.java”.

This bytecode is platform-independent: irrespective of the type of system on which the Java compiler was called, the bytecode can also be executed on any other computer system, at least in the case of “100% pure Java”, which unfortunately cannot be guaranteed for all products.



The Compiler – javac

```
javac [options] <file list>
```

The Interpreter – java

```
java [options] <class> [program arguments]
```



Figure 51: Tools of the Software Development Kit

You can also combine multiple class files into a compressed **archive file**, which then requires less transfer time over the Internet. Java archive files of this type then have, for example, names in the format “MyArchive.jar”. A Java archive contains files and possibly also entire directory structures in the same compressed format used by PKZIP and WinZip. They are administered with the program jar (Java Archiver); the call is similar to that of the UNIX program tar (tape archiver). It is not usually necessary to extract or unpack archive files: The Java compiler and the Java Virtual Machine can read and load the class files directly from the archive files. For this purpose, the file name of the archive file must be specified in a defined format.

The bytecode of a Java application is executed by calling the Java Virtual Machine JVM (in the Java Runtime Environment JRE) in the format `java MyProgram`.

In a similar way to sub program libraries in other programming languages, in Java there are also small Java programs (classes) with precisely defined conventions for the interfaces that allow reuse in multiple applications (applications and applets). These programs are known as **Java Beans**.

J2EE Specification

Java became popular with the Internet as a language of the World Wide Web. Life can be added to static HTML pages using Applets and Java Script. By extending Web servers with the Common Gateway Interface (CGI), these can perform simple database accesses and display the results formatted in the browser. However, these technologies are limited in the case of simultaneous accesses from very large numbers of users who expect reliable information and quick response times.

Sun provides a solution to these problems in terms of server-side architecture for Internet applications with the **Java 2 Platform Enterprise Edition (J2EE)**. J2EE is a vendor standard for a complete palette of software components, which are primarily created in the Java programming language. Sun wishes to ensure that developments comply with the specification of the Java 2 Enterprise Edition using compatibility tests for J2EE.

The specification provides a generally-accepted framework for developing distributed, multi-level applications with modular components. Clearly-defined interfaces between the components and layers are intended to ensure that software components from different vendors are interoperable if they comply with the specification. With the J2EE specification, Sun wishes to ensure that Web applications that conform to the specification run on all J2EE-compatible servers. In accordance with the J2EE specification, the application logic is packed into components, the **Enterprise Java Beans (EJB)**. They represent modular java program components. A container implicitly provides the components with the services of the runtime environment.

Three-level client/server architectures for enterprise applications can be realized in accordance with the J2EE specification. The J2EE application server forms the core of these architectures (see the figure Three-Level J2EE Architecture).

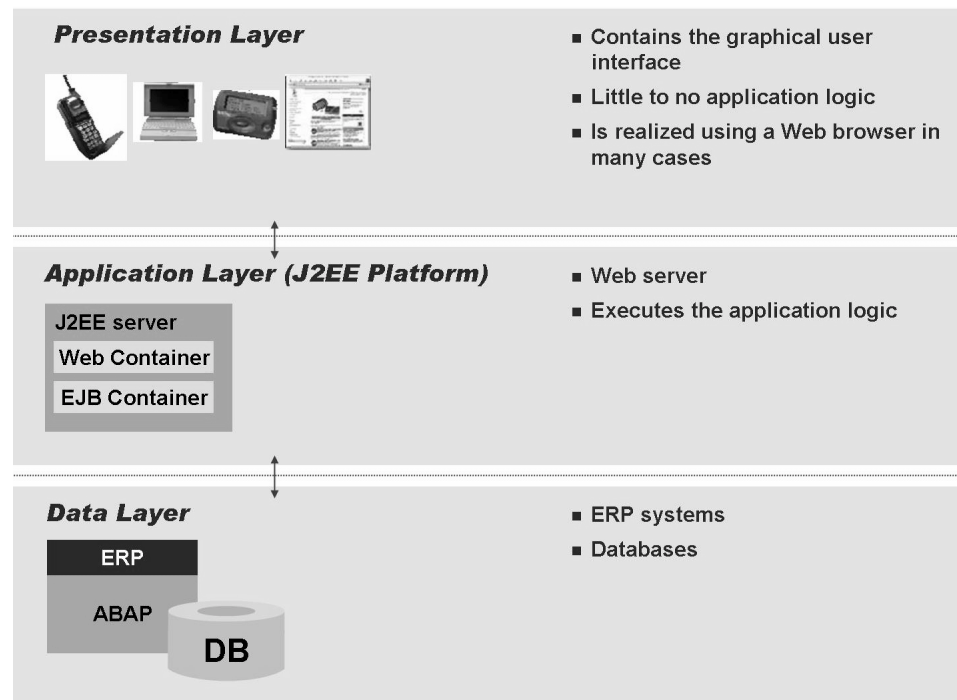


Figure 52: Three-Level J2EE Architecture

The J2EE application server (see the figure Architecture of the J2EE Application Server) processes clients' requests, executes the application logic, and returns the results to the clients. The application data is accessed using the database level. The application logic is implemented in the form of Enterprise Java Beans. These are executed in an **EJB Container**, which is part of the J2EE Application Server. The communication between the client and the J2EE is based on Web standards such as HTTP, HTML, and XML. The J2EE server uses Java Server Pages (JSP) and Java Servlets to generate HTML pages or XML data. Java Server Pages and Java servlets are executed in a **Web Container**, which is, in turn, part of the J2EE Application Server. Java applets can be included in the HTML pages that are sent to the Web browser. These applets are executed in the browser.

Java Server Pages (JSP) are a technology developed by Sun, which essentially allow the simple dynamic generation of HTML and XML output for a Web server. This technology allows the embedding of Java code and special JSP actions in static content. This has the advantage that the logic can be implemented independently of

the design with this technology. Java Server Pages are converted to Java source code using a special JSP compiler. This source code, which corresponds to a Java servlet, is then converted to bytecode by the Java compiler.

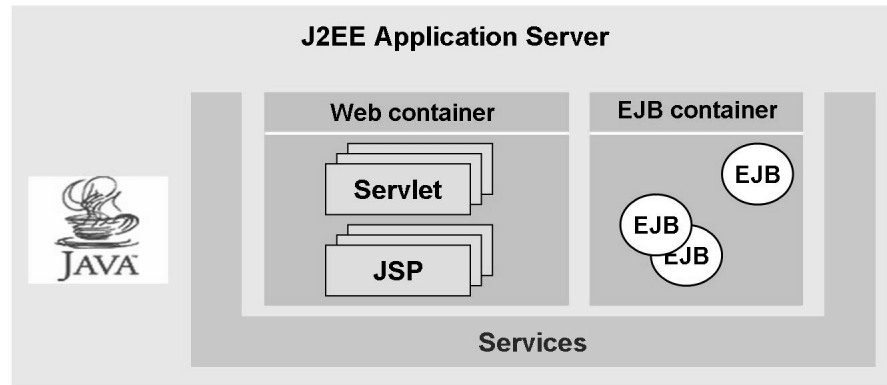


Figure 53: Architecture of the J2EE Application Server

Each application component is executed in a **container**. These containers provide the application components with services in accordance with the J2EE specification. Examples of these services are transaction management, availability, security, scalability, client connection, and database access. This means that the developer can concentrate purely on the application logic and does not need to deal with these system functions.

The following figure shows an example of the processing of a client request by a J2EE Server.

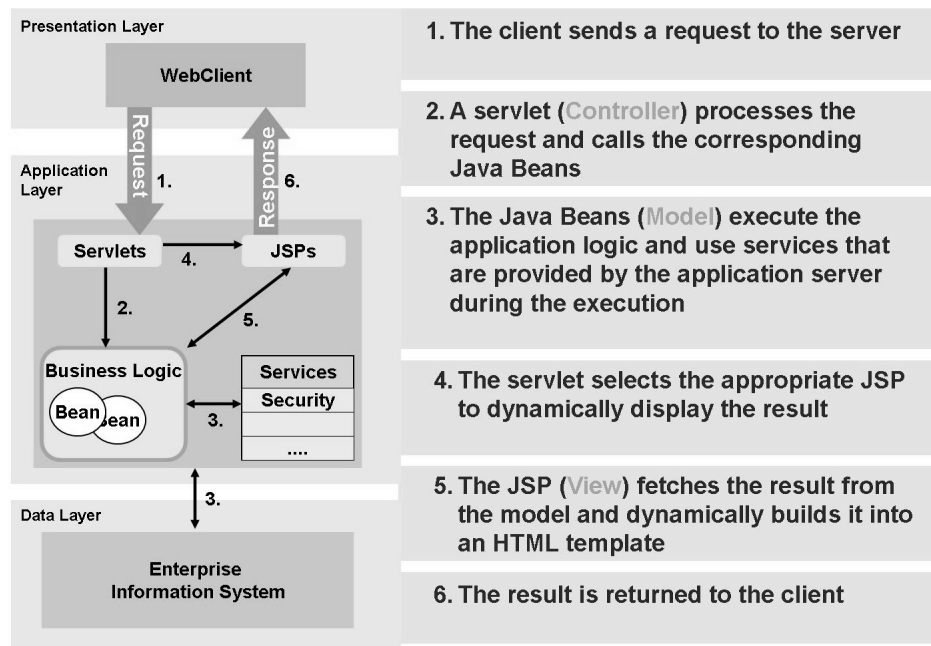


Figure 54: Processing of a Client Request by a J2EE Server

To achieve the aims of quick response times and reliable information, the J2EE application server must also provide scalability and reliability in addition to the functional side. The J2EE Server must handle clustering and load balancing for this.

J2EE Servers in the SAP Environment

SAP takes into account the increasing importance of Java for creating business applications since the J2EE specification was introduced, together with the larger market of Java developers. SAP began a fundamental evaluation of Java as early as 1998 and, with the SAP Web Application Server, now provides a complete J2EE-compatible runtime infrastructure. The architecture of an application server in accordance with the J2EE specification is similar to the ABAP runtime environment used for SAP solutions, since both are platform-independent, executed in a multi-level architecture, and separate presentation logic and business logic.

In summer 2000, SAP began to combine the characteristics of its infrastructure - reliability, scalability, software logistics, and change management - with J2EE in SAP Web Application Server. With SAP Web Application Server 6.20, the non-proprietary object-oriented language Java is available for cross-platform Web applications in addition to ABAP.

The use of J2EE in SAP technology has the following advantages for SAP Web Application Server:



- The open integration architecture SAP NetWeaver integrates perfectly into the openness of J2EE.
- J2EE is additional evidence of the strategy of platform-independence pursued by SAP.
- The J2EE Connector architecture allows standardized integration of external applications.
- Web Services technologies are supported directly by Java.
- The quickly-growing Java community provides simple access to experienced developers.



Lesson Summary

You should now be able to:

- Use basic Java terminology

Lesson: AS Java Processes

Lesson Overview

This lesson discusses the way the SAP NetWeaver Application Server Java (AS Java) works. The processes are introduced and it is explained how they work.



Lesson Objectives

After completing this lesson, you will be able to:

- Understand how AS Java works
- Name the AS Java processes and describe their purpose
- Describe how requests to AS Java are processed

Business Example

Your company has decided to migrate a J2EE application that is currently running on a third party provider J2EE application server to SAP NetWeaver 7.0 to consolidate the system landscape, which has been heterogeneous so far. SAP NetWeaver 7.0 is a J2EE 1.3 certified application server.

Processing of Requests in AS Java

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

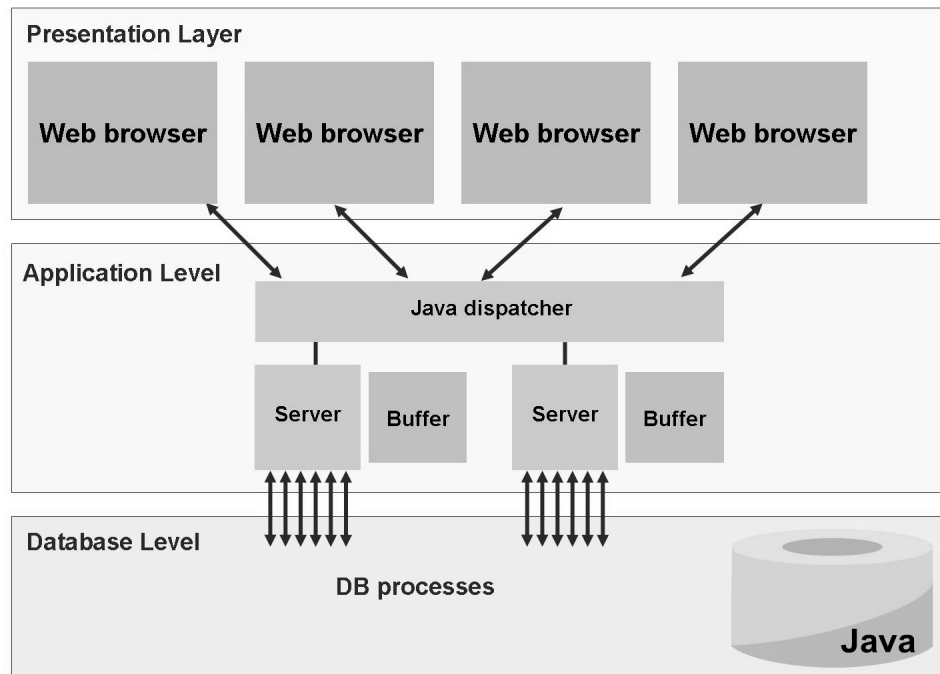


Figure 55: Processing a User Request

A web browser is the standard user interface for AS Java. A user request for AS Java is usually an HTTP request that is received by the **Java dispatcher**. The dispatcher forwards the processing requests to one of the server processes of "its" instance.

The actual processing takes place in the **server** process, whereby the user who sent the request is usually assigned the same server process again for the next request.

The dispatcher and server processes of AS Java are also called **nodes**. All processes of AS Java together with the database schema form the **Java cluster**. In contrast to the processes of AS ABAP (excluding the ICM), the cluster nodes of AS Java are **multithreaded**. This means that an AS Java process consists of many threads and one request can be processed in each thread. Hence, **one server process** always processes **many user requests in parallel**.

To process user requests it is often necessary to read data from the Java schema of the database or to write to it. To do so, each server process is connected **multiple times** to the **Java schema of the database** via a connection pool (DB pool).

Once processing is complete, the processing result from the server processes is returned to the web browser via the dispatcher.

The **buffers** help to speed up processing of user requests. This means that the data does not have to be read from the database every time it is needed, but can be called very quickly from the buffer. Each server process has its own buffer.

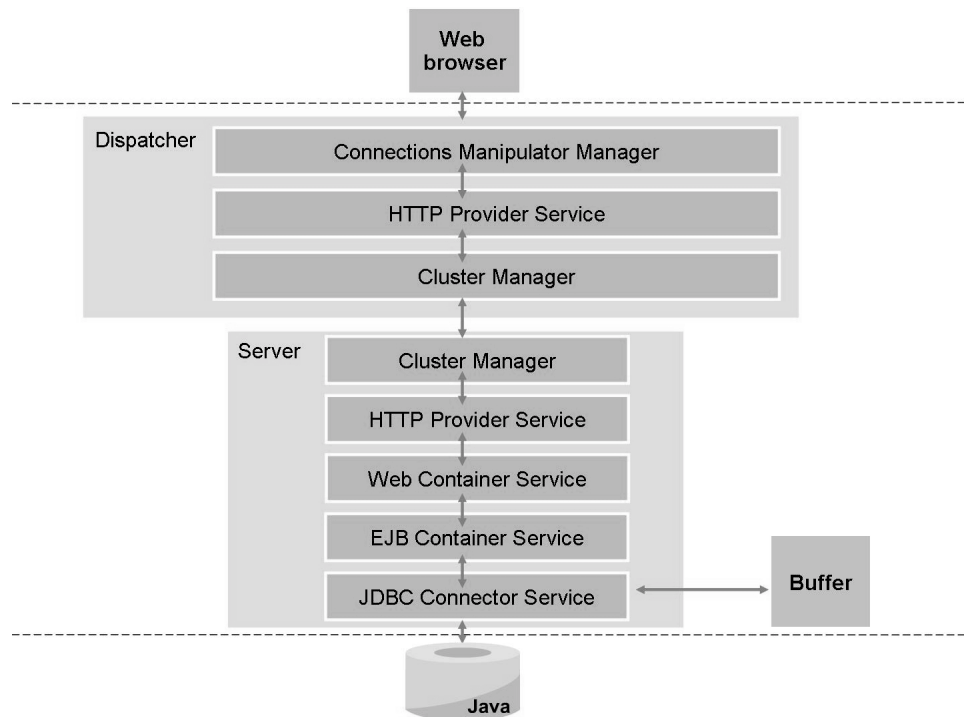


Figure 56: Processing a Request

The nodes of AS Java are split into different functional modules called **managers** and **services**. The managers form the **Java Enterprise Runtime**. The Java Enterprise Runtime provides basic core functions of AS Java. It is also referred to as kernel.

Together with the interfaces and libraries, the services are called **J2EE Engine Components**. The J2EE Engine Components provide programming interfaces (APIs) to the applications; the applications can then use these APIs to access the AS Java functions.

In case of an HTTP requests to the Java dispatcher the **Connections Manipulator Manager** holds a connection object with information about the client sending the request. The request is then forwarded to one of the processes of this instance by the **HTTP provider service** using a **cluster manager**.

The cluster manager of the server process receives the request and forwards it to the HTTP provider service. In **Web Container Service**, the presentation logic of the application is then processed. The web container service provides the processing of

servlets and JavaServer Pages (JSP). The business structure logic of the application is processed in the form of EJB beans in the **EJB Container Service**. If, in the processing of the request, data from the database is required, the **JDBC Connector Service** is used to establish a connection to the database and the data is requested there. If the same tables contents have already been queried by this server process, the content can be retrieved in the table buffer at application level (if buffering is allowed for the table).

The response to the web browser using HTTP is then returned in the same way.

Transactional Processing in AS Java

The **ACID** concept has now been implemented in AS Java in the same way as in AS ABAP. In AS Java, the **Transaction Service** is responsible for managing transactions. The two J2EE standards **Java Transaction API (JTA)** and **Java Transaction Service (JTS)** are implemented using the Transaction Service. Applications (application developers) can use the Java Transaction Service by means of the JTA interface.

In the J2EE standard, much of the implementation of the transaction logic is left to the respective database used. Hence, a transaction at application level often corresponds exactly to one database transaction in the J2EE standard. The following figure illustrates the correlation using a JSP application as the example.

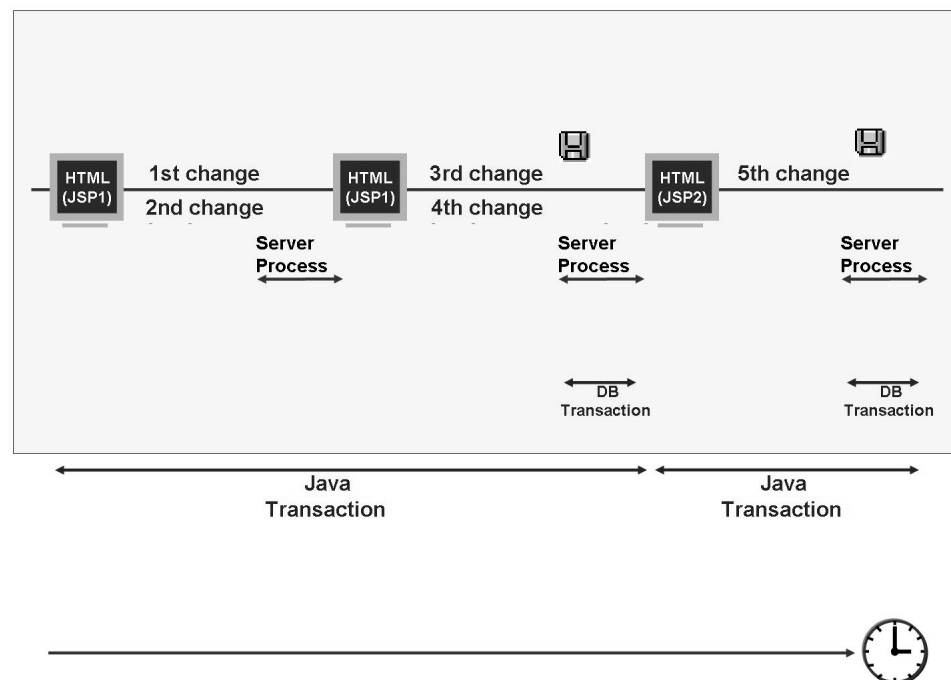


Figure 57: Transactions in AS Java

In AS Java, changes and entries in the user interface (in the web browser) are not made persistent immediately in the database. When the user saves his entries, the Java transaction is completed immediately and the data is made persistent in a database transaction. A Java transaction thus consists of a database transaction.

Persistence

SAP has created **Open SQL for Java** framework for AS Java. Hence, Java application developers have access to various database-independent programming techniques as well as important functions for improved performance and trouble shooting.

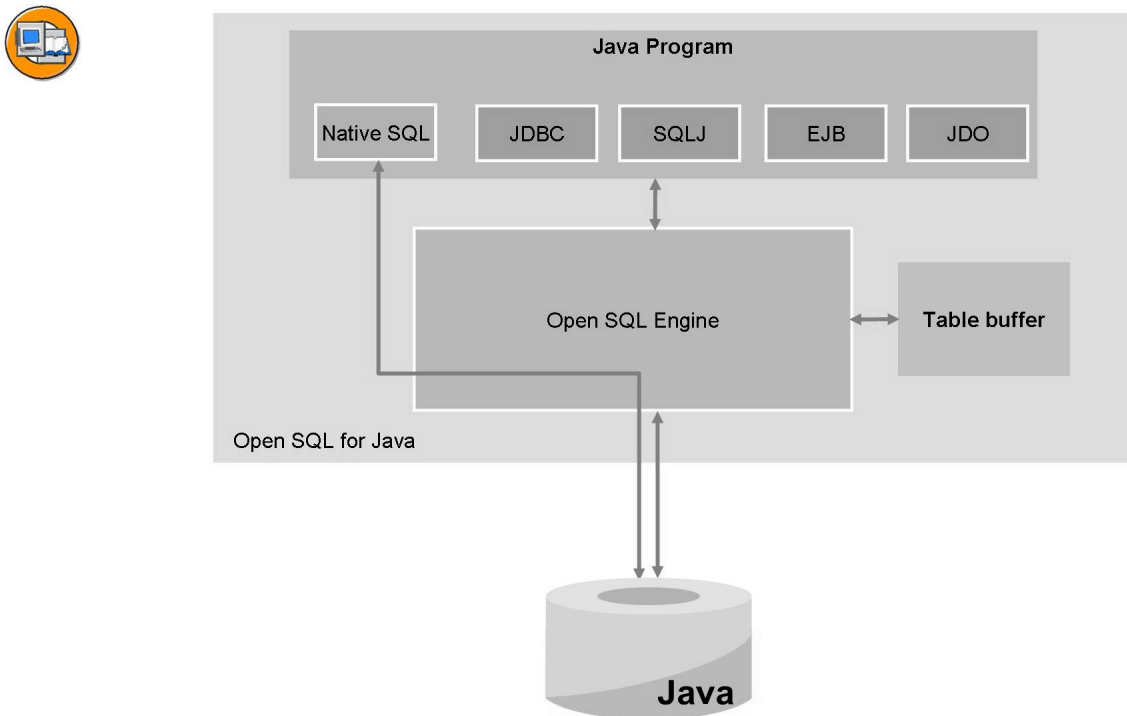


Figure 58: Open SQL for Java

If the Java program is supposed to be **portable**, that is, run with a database other than the one used originally, developers can choose between Open SQL/JDBC, Open SQL/SQLJ, EJB (Enterprise JavaBeans) and JDO (Java Data Objects). If developers use Native SQL in the program, they lose the portability and cannot use the table buffer of Open SQL for Java Frameworks.

Lock Management

The lock concept of the database is used in the J2EE standard. So if application developers make sure that they implement database-independent database accesses, the application will be portable but will respond semantically different on different database platforms. For this reason and to improve response times, SAP introduced the concept of the enqueue service analogous to AS ABAP.

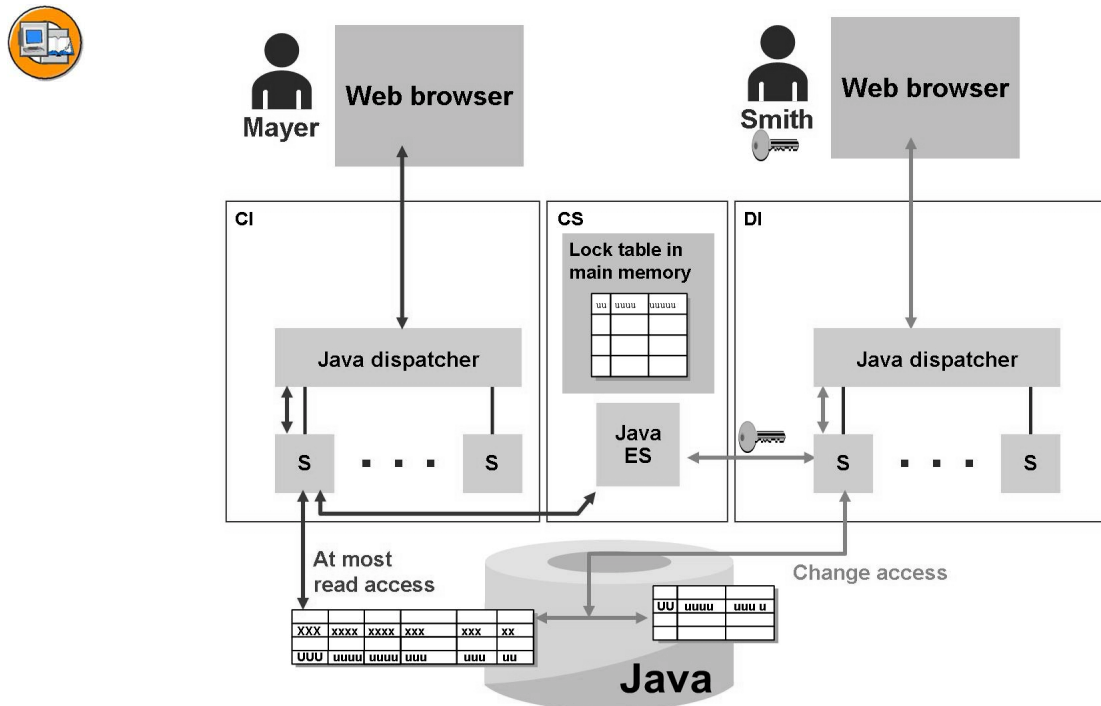


Figure 59: Lock Management in AS Java

If a user wants change access to data, the executing server process requests a lock (to do so, the application developer must program this request explicitly).

The application developer uses the interface of the **Application Locking Service** to request a **logical lock**.

The request is forwarded to the **Enqueue Server** via the **Locking Adapter Service** and the **Locking Manager**. The enqueue server now checks whether a new lock can be generated; that is, whether there is a collision with locks that have already been set. If a lock can be set, the dialog work process creates it and the user (lock owner) is given the lock key.

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



Lesson Summary

You should now be able to:

- Understand how AS Java works
- Name the AS Java processes and describe their purpose
- Describe how requests to AS Java are processed

Related Information

- SAP Education course: ADM200 - *AS Java Administration*
- SAP Education course: JA320 - *SAP Java Persistence Framework*

Lesson: Java Cluster Architecture

Lesson Overview

This lesson describes the architecture of SAP NetWeaver AS Java. The individual components of SAP NetWeaver AS Java and their functions are introduced. A Java cluster encompasses all Java components of an SAP system.



Lesson Objectives

After completing this lesson, you will be able to:

- Explain the term Central Services of SAP NetWeaver AS Java
- Understand and use concepts such as Java instance, Java dispatcher, and server

Business Example

After the installation of a SAP NetWeaver Application Server Java, configuration is still required. You should therefore be familiar with the basic architecture of the cluster of SAP NetWeaver Application Server Java.

The Java Instance Concept

SAP NetWeaver Application Server Java takes the instance concept of SAP Basis, which combines multiple components that are monitored, started, and stopped together into an administrative unit.

The following components are part of a Java instance:



- The **server processes** provide the infrastructure in which the J2EE applications run.
- The **Java dispatcher** distributes the client requests to the free server processes of the instance.

An instance always runs on one physical server, but there can be multiple instances on one server. Within an SAP system, an instance is defined using the system ID (SID) of the SAP system and the instance number. An SAP system consists of a database and one or more instances. These instances can either be purely ABAP or Java instances, or instances with ABAP and Java infrastructure.

The Central Services form a special Java instance. They provide the basis of communication and synchronization within a Java cluster. The central instance is another special instance. This runs on a physical server with the Central Services. During its installation, the Software Deployment Manager is also installed.

To ensure high-performance when processing Java requests, the SAP system can be scaled using the number of server processes for each instances or using the number of instances.

Java Cluster Architecture

All Java components of an SAP system are known as a Java cluster. A Java cluster consists of:



- A (central) **Java instance** with a dispatcher and at least one server process.
- The **Central Services**, which contain a message server and an enqueue server.
- A **database** for the central storage of data.
- Optionally, additional Java instances

A minimal cluster installation and an installation with multiple SAP NetWeaver AS Java instances are shown as examples in the following figures.

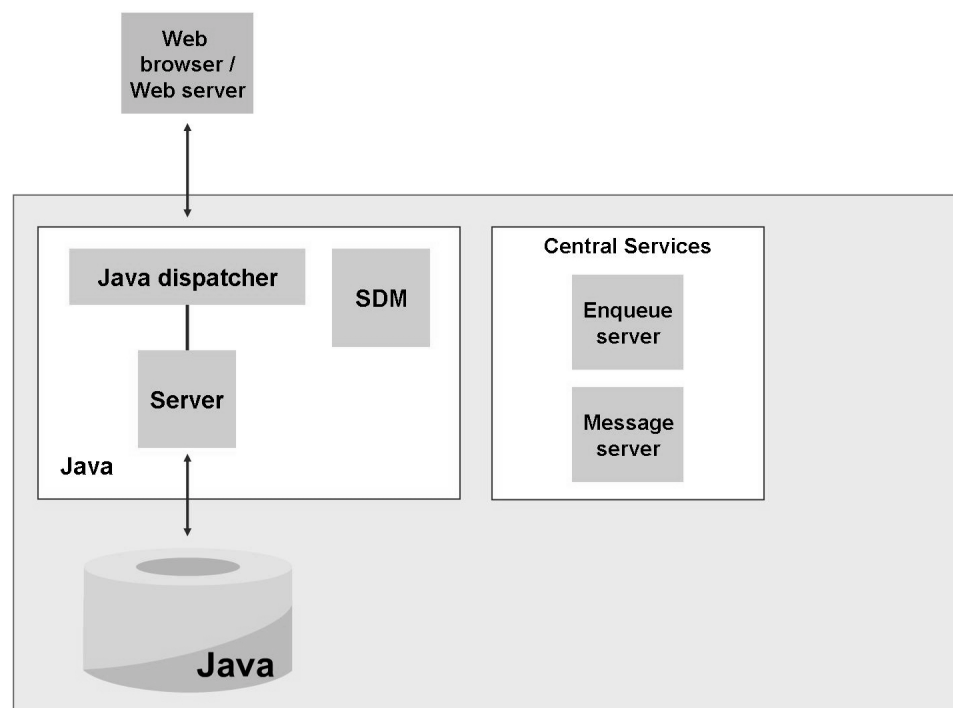


Figure 60: Minimal Cluster Installation

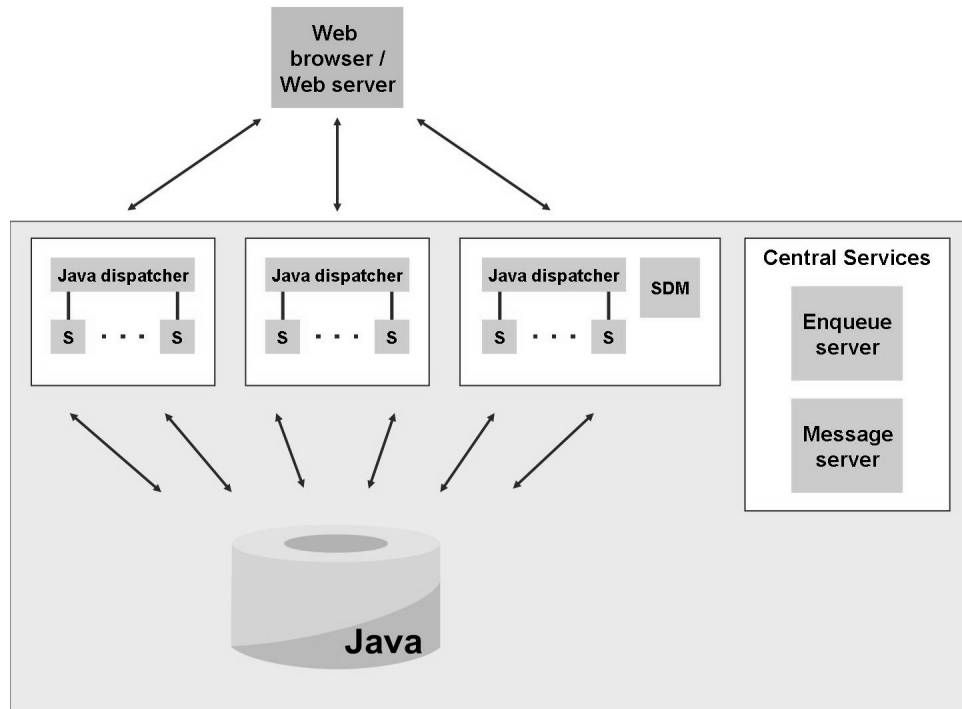


Figure 61: Cluster with Multiple SAP NetWeaver AS Java Instances

The Java Dispatcher and Server Processes

Client requests to SAP NetWeaver AS Java are received by the Java dispatcher. It selects a free server process to process the request and creates the connection between the client and the server process. The dispatcher distributes new incoming requests to the available server processes using a “round robin” algorithm. If there is already a connection to the client, the request is forwarded to the server process that is already processing requests for this client. The individual steps in the processing of a request are outlined in the following figure.

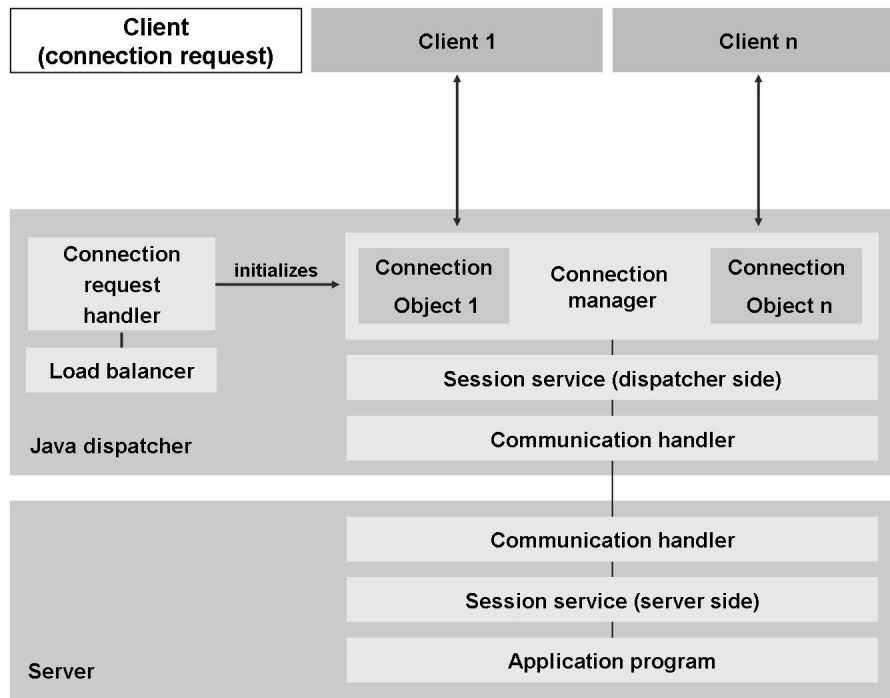


Figure 62: Processing a Request

Within the Java dispatcher, the connection request handler receives the very first request from a client. It initializes a connection object, which is then assigned to this connection. The connection request handler then uses load balancing to determine which server process processes the request.

After the initialization of a connection object, the client is then connected to this connection manager for all following requests. The connection manager selects the required session service (such as HTTP) on the basis of the request type. The request is then forwarded to the server process by the communication handler.

The server process of the SAP NetWeaver Application Server Java runs the Java applications. The structure of server processes essentially corresponds to the structure of the Java dispatcher. The server processes are implemented as multi-threaded servers and can therefore process multiple requests in parallel. The system or application threads take over the processing of the requests.

Central Services

The Central Services run on one host and form a separate Java instance. They consist of the message service and the enqueue service.

The Central Services provide the basis for communication and synchronization for the Java cluster:



- The **message service** administers a list of the dispatchers and the server processes of the Java cluster. It represents the infrastructure of data exchange (for small quantities of data) between the nodes involved. In the case of load balancing between a large number of Java instances, it also provides the load balancing information for the SAP Web Dispatcher.
- The **enqueue service** administers logical locks that are set in a server process by the executed application program. It is also used for cluster-wide synchronization.

The Central Services are essentially required when a Java cluster is installed. They are started on a host with a separate system number and the system ID (SID) of the entire system. If the Central Services are running, other Java instances are started with the program JControl (see the unit Starting and Stopping a SAP NetWeaver AS Java).

Message Service

The message service is a separate program that allows communication between the elements of a Java cluster. The message service knows all active Java instances.

The terms message server and message service are used with the same meaning in the training material. To be precise, the message server is a program/process that provides the message service.

The message service performs the following tasks in the Java cluster:

- Notification of events that arise in the cluster, for example, if a node of the cluster disappears (due to failure or the instance being shut down), or when a service is started or stopped.
- Communication between different services
- Forwarding of messages and requests to all participants (broadcast)
- Prepare logon information for the SAP Web Dispatcher
- Support for message server failover
- Guaranteed message transmission
- Exchange of cache information in the cluster

By using the message service, you can avoid performance problems that occur if all the cluster elements are connected with each other and exchange data with each other. This previously lead to significant performance problems, particularly with large clusters. The same technology is used as with the SAP Message Server for the earlier SAP NetWeaver AS versions without Java.

Enqueue Service

The enqueue service runs on the Central Services instance of the Java cluster. It manages the lock table in the main memory and receives requests for setting or releasing locks. It uses the tried and tested SAP lock concept.

The terms enqueue server and enqueue service are used with the same meaning in the training material. To be precise, the enqueue server is the program or process that provides the enqueue service.

The enqueue service has the following tasks:

- Internally, it is used for synchronization within the Java cluster
- The applications can lock objects and release locks again. The enqueue service processes these requests and manages the lock table with the existing locks.



Lesson Summary

You should now be able to:

- Explain the term Central Services of SAP NetWeaver AS Java
- Understand and use concepts such as Java instance, Java dispatcher, and server

Lesson: The Internal Structure of SAP NetWeaver AS Java

Lesson Overview

This lesson introduces the internal architecture of SAP NetWeaver AS. This architecture is the foundation for realizing a J2EE application server in accordance with the J2EE 1.3 specification.



Lesson Objectives

After completing this lesson, you will be able to:

- Name the most important managers of the SAP NetWeaver AS
- Name the most important services of the SAP NetWeaver AS

Business Example

SAP NetWeaver AS Java consists internally of several managers and services. To be able to configure these managers and services, you should first understand their significance and functions.

Introduction

SAP NetWeaver Application Server Java was previously known as SAP J2EE Engine. However, this term should no longer be used, since it does not uniquely identify an SAP product. If you are referring purely to the internal technical architecture of SAP NetWeaver AS Java for realizing a J2EE server in accordance with the J2EE 1.3 specification, then it is appropriate to use the term J2EE Engine in this case.

The internal structure of SAP NetWeaver AS Java is divided into three logical levels (see the figure Internal Structure of SAP NetWeaver AS Java):

- **SAP Java Enterprise Runtime** - provides fundamental functions of the runtime environment, such as class loading, cluster communication, management of configuration data, and so on .
- **J2EE components** - contain interfaces, libraries, and services
- **Applications Layer** - relates to the applications that are deployed and run in SAP NetWeaver Application Server Java.

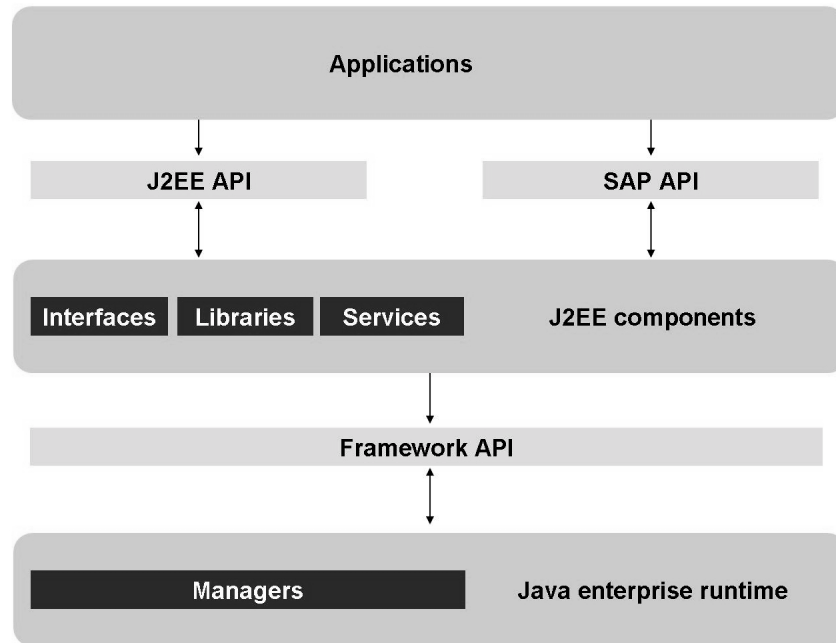


Figure 63: Internal Structure of SAP NetWeaver AS Java

The following general rule applies to the interaction between these three logical entities in SAP NetWeaver AS Java: higher-level components can use the functions of the lower-level layers. On the other hand, the lower levels are not aware of the higher levels and cannot therefore use their functions.

This rule is a consequence of the start sequence of the individual modules of the system. First, the runtime environment is started, then the services are started, and then the applications are started.

Communication between the individual components takes place using defined Application Programming Interfaces (APIs). The components of the higher levels use these APIs to use functions of the lower levels. The J2EE components use the Framework APIs to talk to the SAP Java Enterprise Runtime. The applications can talk with the J2EE components either using APIs defined by the J2EE 1.3 specifications, or using proprietary SAP APIs.

The functions of these logical levels and their interaction are described in the following.

SAP Java Enterprise Runtime

The SAP Java Enterprise Runtime provides the core functions of the system. These core functions are realized by multiple subsystems of the lowest level. These subsystems are called managers. A selection of these managers with a short description is listed below:

- **Log Manager:**
manages the process of logging system events. It configures the logging routines of the kernel. The Log Manager is the first manager that is started during the startup of SAP NetWeaver Application Server Java.
- **Ports Manager:**
manages the process of opening various communication ports on the system.
- **Application Thread Manager:**
manages inbound client requests. Searches for free threads to process inbound requests. If no threads are free, the requests are buffered in a request queue.
- **Thread Manager:**
manages threads in which internal operations of SAP NetWeaver AS Java are executed.
- **IP Verification Manager:**
manages a list of host names (using their IP addresses) and uses this to control access to the elements of the cluster. It stands between an inbound request and the requested resource.
- **Connections Manipulator:**
manages client connections to the cluster. Since the client accesses the cluster through the dispatcher, this manager runs only on the dispatcher.
- **Locking Manager:**
Represents the interface between the server process and the enqueue service.
- **Configuration Manager:**
Allows SAP NetWeaver AS Java to store and access data in a relational database. Uses the Open SQL for Java Framework to do this.
- **Classloading Manager:**
Central location for the registration and removal of loaders and references between them.
- **Cluster Manager:**

Manages the elements of a cluster (dispatcher and server processes). It updates information about the status of the individual elements of a cluster and the services that are provided by it.

Important settings:

- Join Port: Port with which dispatcher/servers listen for inbound connections.
- Open Port: Connections between cluster elements are created using this port.
- Cluster Hosts: Hosts to which the dispatcher attempts to create connections.

- Service Manager

Represents a container in which all services in the cluster are executed.

J2EE Components

The J2EE components form the second level within the three-level structure of SAP NetWeaver AS Java. They provide the complete infrastructure for executing J2EE applications and proprietary SAP applications.

Three types of J2EE components are defined:

- Interfaces:

Agreements that define how different components of SAP NetWeaver AS Java work together. They do not provide any runtime functions themselves, but rather are used by services that provide their implementation.

- Libraries:

They provide names, classes, and objects within SAP NetWeaver AS Java. These objects are created by the system when it loads the library, or when an object is first requested.

- Services:

The services that SAP NetWeaver AS Java provides for processing requests are defined and configured using the Services. Service components can access and utilize functions of the runtime environment through the Framework API. They are the most important of these three types of J2EE Components.

A selection of the most important services with a short description is listed below:

- **Security Provider:**
Administration of users and groups and authorization administration. Controls access to resources or applications deployed in SAP NetWeaver AS Java.
- **Monitoring Service:**
Allows access to information about the current system status. Provides general and statistical information, among other things, about the nodes in the cluster, memory utilization, performance, applications, and user connections.
- **Log Configurator Service**
Manages the configuration of the logging and tracing mechanism of SAP NetWeaver AS Java.
- **Log Viewer Service:**
Provides an integrated log viewer for displaying log and trace information.
- **Deploy Service:**
Manages the deployment of Java applications.
- **EJB Container Service:**
Manages all Enterprise Java Beans (session beans, entity beans, and message-driven beans), which are executed in the EJB Container of SAP NetWeaver AS Java.
- **HTTP Provider:**
Analyzes the URL of inbound HTTP requests, forwards the requests to the correct module of SAP NetWeaver AS Java for processing, and returns the responses to the client.

Applications Layer

The applications form the third level within the architecture of SAP NetWeaver AS Java. The boundary between the applications and the J2EE components is defined by the J2EE 1.3 APIs and a few proprietary SAP APIs. Applications use these APIs to utilize the functions of the J2EE components.

An enterprise application can contain different types of application components:

- **Servlet:** A program module written in the Java programming language, which is used to respond to requests to a Web server in such a way that the content of the returned HTML Web page is dynamically generated. Servlets are standardized in the context of the J2EE specification.
- **Java Server Pages (JSP):** Technology for dynamically generating HTML and XML output of a Web server. This technology allows the embedding of Java code and special JSP actions in static content. This has the advantage that the logic can be implemented independently of the design with this technology. Java Server Pages are converted to Java source code, which corresponds to a Java servlet, using a special JSP compiler.
- **Enterprise Java Beans (EJB):** Used for the standardized, simplified development of business applications using Java. They are part of the central elements of the J2EE (Java 2 Enterprise Edition) specification.
- **Java Database Connectivity (JDBC):** Uniform interface to databases from different vendors, defined specifically for relational databases. The tasks of JDBC include creating and managing database connections, forwarding SQL queries to the database, and converting the results into a format that is usable for Java and making them available to the program.

These components reside in different containers of SAP NetWeaver AS Java (see the figure Containers of SAP NetWeaver AS Java). The containers provide runtime services for the application components.

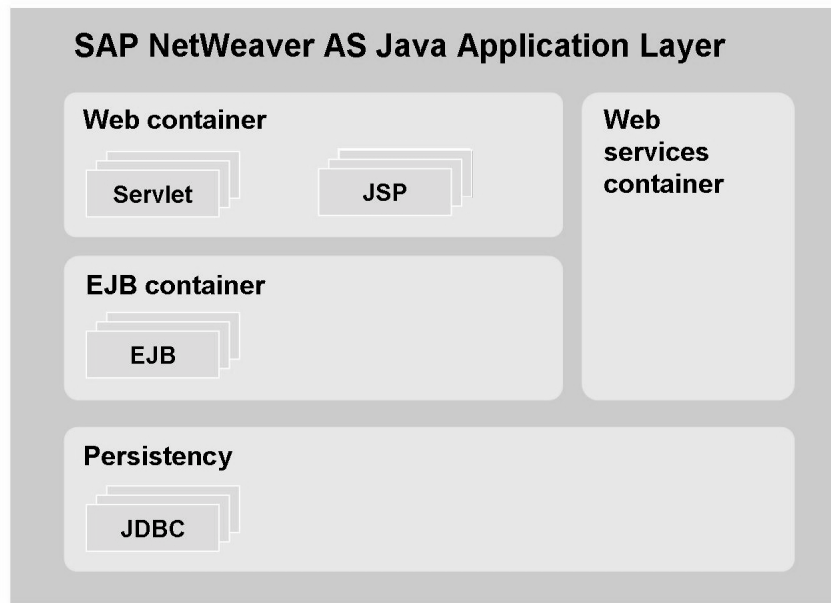


Figure 64: Containers of SAP NetWeaver AS Java



Lesson Summary

You should now be able to:

- Name the most important managers of the SAP NetWeaver AS
- Name the most important services of the SAP NetWeaver AS



Unit Summary

You should now be able to:

- Outline simple client/server configurations
- Name the processes of the SAP NetWeaver Application Server
- Define the term instance and recognize the characteristics of a central instance
- Understand how AS ABAP works
- List the AS ABAP processes and describe their purpose
- Describe how requests to AS ABAP are processed
- Use basic Java terminology
- Understand how AS Java works
- Name the AS Java processes and describe their purpose
- Describe how requests to AS Java are processed
- Explain the term Central Services of SAP NetWeaver AS Java
- Understand and use concepts such as Java instance, Java dispatcher, and server
- Name the most important managers of the SAP NetWeaver AS
- Name the most important services of the SAP NetWeaver AS



Test Your Knowledge

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

Choose the correct answer(s).

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

2. Which AS ABAP process receives the user request?

Choose the correct answer(s).

- ☐ A Work process
- ☐ B Dispatcher
- ☐ C Server process
- ☐ D SAP GUI

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

Choose the correct answer(s).

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

4. Which of the following processes belong to AS Java?

Choose the correct answer(s).

- ☐ A Server (process)
- ☐ B Internet Communication Manager (ICM)
- ☐ C Software Deployment Manager (SDM)
- ☐ D Dispatcher
- ☐ E Java Connector (JCo)

5. Which of the programming languages listed below are implemented platform-independently?

Choose the correct answer(s).

- ☐ A Fortran
- ☐ B ABAP
- ☐ C Java
- ☐ D C / C++

6. Briefly describe the function of the Java Virtual Machine (JVM).



Answers

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

Answer: A, C

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

2. Which AS ABAP process receives the user request?

Answer: B

The dispatcher receives the user request on the AS ABAP and passes it on to an available work process. The SAP presentation program, *SAP GUI*, is not part of the application server (software-oriented view), and the server process is an AS Java process.

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

Answer: A, C, E, F, G

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

4. Which of the following processes belong to AS Java?

Answer: A, C, D

The ICM is not an AS Java process but an AS ABAP process. The Java Connector is part of AS Java but is not an independent process.

5. Which of the programming languages listed below are implemented platform-independently?

Answer: B, C

ABAP and Java are programming languages that are implemented platform-independently.

6. Briefly describe the function of the Java Virtual Machine (JVM).

Answer: The JVM executes the platform-independent bytecode generated by the Java compiler on the relevant hardware.

Unit 4

Starting and Stopping SAP Systems

Unit Overview

The topic of this lesson is the starting and stopping of an SAP system. These are two of the basic tasks of system administration. You will also learn about the available log and trace options, to be able to react correctly if an error occurs.

On all operating systems, it is possible to use the SAP Management Console for starting and stopping. In the Windows operating system, it is also possible to use the Microsoft Management Console (SAP MMC). In the UNIX operating system, it is also possible to use the *startsap* and *stopsap* scripts.

This unit first describes in detail the process of starting an SAP NetWeaver AS ABAP. Then, the process of starting and stopping an SAP NetWeaver AS Java is introduced in the next (4) lessons. Technically, SAP NetWeaver AS Java uses the *Startup and Control Framework* to perform the start process of the Java instances.



Note: The following lessons have been formed from two different courses. The advantage to this is that all of the knowledge required for this range of topics is introduced, within the context of learning, in a training course for consultants. Consequently, you can use synergies to facilitate understanding among participants, and for actual use. This would not be possible if these topics were discussed at a later time.



Unit Objectives

After completing this unit, you will be able to:

- Describe the process of the start procedure of an SAP system
- Start the entire SAP system or individual instances
- Evaluate start logs to analyze problems
- Stop the entire SAP System or individual instances
- Describe the starting and stopping of an SAP system under different operating systems
- Describe the locations of the log files for various databases

- Describe the sequence in which the components of an SAP system are started and stopped
- Describe the general start process for an SAP NetWeaver AS Java
- Describe the general start process for an SAP NetWeaver AS ABAP + Java
- Operate the tools to start and stop SAP NetWeaver AS ABAP + Java
- Operate the tools to start and stop SAP NetWeaver AS Java
- Use the term Startup and Control Framework
- Describe the individual steps during the start and stop processes of a Java instance
- Find the storage locations of trace and log files of the Startup and Control Framework
- Name the most important trace and log files of the Startup and Control Framework and outline their content

Unit Contents

Lesson: System Start: Process	173
Exercise 10: Starting the SAP System	179
Lesson: System Start: Logs	183
Lesson: System Shutdown: How and Why?	188
Exercise 11: Stopping an SAP System	191
Lesson: Appendix - Starting and Stopping: Operating System-Specific	195
Lesson: Appendix - Database Logs	208
Lesson: Overview of the Process for Starting and Stopping a SAP NetWeaver AS Java	215
Lesson: Tools for Starting and Stopping	221
Exercise 12: Tools for Starting and Stopping	231
Lesson: Java Startup and Control Framework	235
Exercise 13: Java Startup and Control Framework	241
Lesson: Logs of the Start and Stop Processes of SAP NetWeaver AS Java	244
Exercise 14: Logs of the Start and Stop Processes of SAP NetWeaver AS Java	247

Lesson: System Start: Process

Lesson Overview

The topic of this lesson is the starting of an SAP system. As well as the actual start sequence, the SAP Management Console (SAP MC) will also be introduced as a central control element.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the process of the start procedure of an SAP system
- Start the entire SAP system or individual instances

Business Example

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

Start Process of the SAP System

The starting of an SAP system is the basic prerequisite for being able to work with the system. Restarts of SAP systems are also required after maintenance work on the hardware in use and changes to the SAP system's profile. Starting an SAP system is an initial process with which every administrator should be familiar.

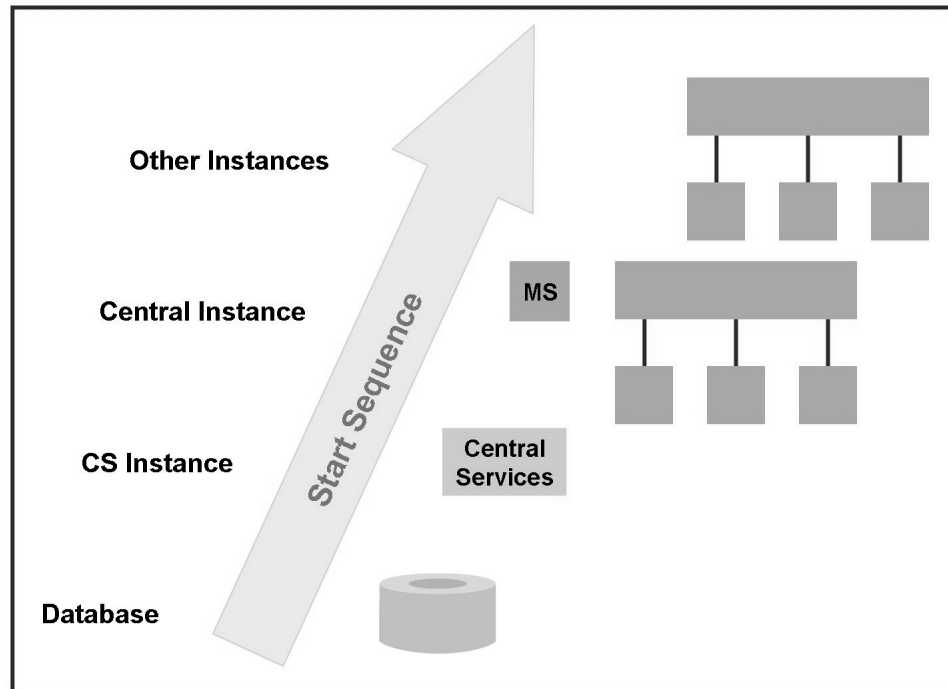


Figure 65: Start Process of the SAP System

Starting an SAP System is performed in a number of steps and is the task of the operating system user <sid>adm.

Start the database:

- The underlying element of the entire SAP system is the database. Before the SAP instances are started, this must have operational status. The database is therefore always started as the first step.

Start the Central Services instance (AS Java and AS ABAP+Java)

- Start the Central Services instance that contains both the enqueue and the message service for the Java part of the SAP system. The Central Services are installed as an independent instance.

Starting the Central Instance:

- Next, the operating system collector **SAPOSCOL** is started, if it is not already active. This standalone program runs in the operating system background, independently of SAP instances. This program collects data about operating system resources and makes this data available through the shared memory of all SAP instances.



Note: If you later discover that SAPOSCOL has not been started, you can start it at any time by calling transaction ST06 and choosing the path *Detail Analysis Menu → OS Collector → Start*.

- The central instance with the message server and the dispatcher and its work processes is then started. If the start up parameters are set accordingly, the dispatcher also starts the Internet Communication Manager (ICM) and the processes of AS Java (if it is installed). You should only start other (optional) instances once the message and enqueue servers are active.

Starting the dialog instances

- If the dialog instance is not running on the same host as the central instance, the SAPOSCOL operating system collector is first started on this host (if it is not yet running).
- The dispatcher is then started with its work processes.

The sapstartsrv Process

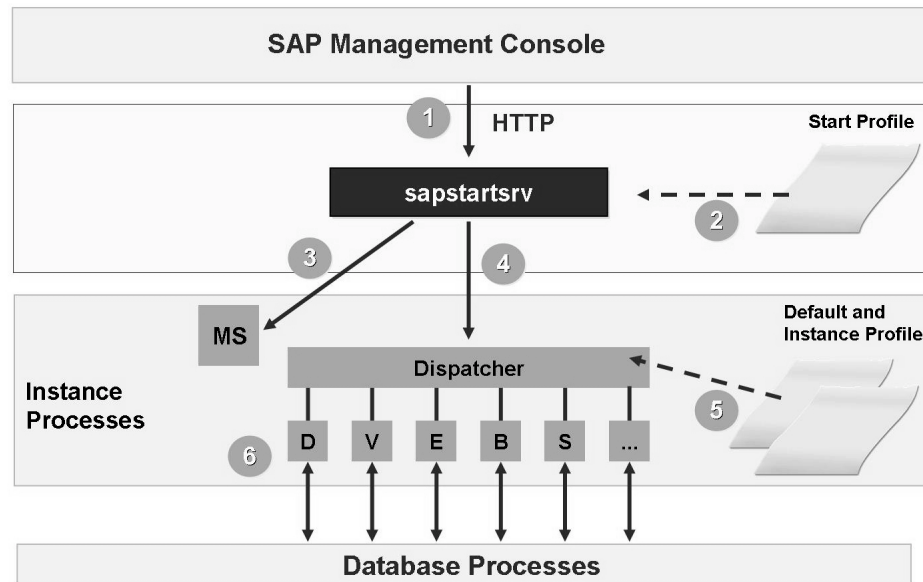


Figure 66: Process of Starting an SAP Instance with the SAP Management Console (SAP MC)

As of release SAP NetWeaver 7.0, the start service *sapstartsrv* that has been used for many years in the Windows installation is installed on all platforms. The service is a Windows service or a Unix daemon for the administration of an SAP instance.

A *sapstartsrv* process is started for each SAP instance and does not terminate even if the instance is stopped. When it starts, *sapstartsrv* reads the start profile of its instance and opens its Web service interface by default on the ports 55513 (HTTP, \$\$ represents the instance number) and, providing SSL is configured, on 55514 (HTTPS). The start service then waits for incoming requests from Web service clients and processes these. The Web service interface contains a number of functions for administrating and monitoring an SAP instance, in particular the SAP Management Console (SAP MC). *sapstartsrv* also has a limited Web server function that allows you to download all files under *DIR_EXECUTBALE/servicehttp* by way of HTTP(S). This can be used to start the SAP Management Console from a Web browser on any host, for example. If no other URL is specified (for example, <http://<hostname>:55513>), the system automatically redirects you to <http://<hostname>:55513/servicehttp/sapmc/sapmc.html>, for example, to start the SAPMC.

sapstartsrv manages an internal list of protected operations. These can be changed, if necessary, with the start profile parameter *service/protectedwebmethods*. With the start profile parameter *service/hostname* you can also determine the IP address / host name to which the Web service port should be connected (default: all / 0.0.0.0), to limit accessibility in the network. You then have to restart *sapstartsrv*. To do this, refer to SAP Note 927637 - *Web service authentication in sapstartsrv as of release 7.00*.

Exercise 10: Starting the SAP System

Exercise Objectives

After completing this exercise, you will be able to:

- Start an SAP system

Business Example

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

Task 1: Starting with the SAP Management Console

Start the SAP system using the SAP Management Console.

1. Start the SAP system using the SAP Management Console. Monitor the start of the processes in your system.

Your instructor will provide you with the name of the server on which you can start the SAP MC.
2. Which process types are started at operating system level after your system is started up?
3. Check whether your system started correctly. To do this, log on to your SAP system and call the process overview. Compare the list of processes at operating system level with the process overview in the SAP system.

Task 2: Starting from the Command Line (Optional)

Start the SAP system using a command call.

1. Start the SAP instances by entering the command *startsap*. Start the central instance first and then the dialog instances.

Solution 10: Starting the SAP System

Task 1: Starting with the SAP Management Console

Start the SAP system using the SAP Management Console.

1. Start the SAP system using the SAP Management Console. Monitor the start of the processes in your system.

Your instructor will provide you with the name of the server on which you can start the SAP MC.

- a) Start a browser on the host specified by your instructor and enter **http://<hostname>:55\$13** in the address bar. Then choose *Start*.

Choose the node for the central instance in the SAP Management Console tree. Select the desired instance and choose the *Start* function from the context menu (right mouse button). Start the dialog instance in the same way.



Hint: You should always start the central instance first.

2. Which process types are started at operating system level after your system is started up?
 - a) You can monitor the processes at operating system level with the *Task Manager* or the *Process Explorer* tool (Desktop shortcut).

The following processes are started at operating system level after your system is started up: *saposcol.exe*, *msg_server.exe*, *gword.exe*, *icman.exe* and several *disp+work.exe*. Other Java processes can belong to your instance (these have the process names: *jlaunch* or *jcontrol*).
3. Check whether your system started correctly. To do this, log on to your SAP system and call the process overview. Compare the list of processes at operating system level with the process overview in the SAP system.
 - a) Log on to your SAP system. The process overview *Tools* → *Administration* → *Monitor* → *System Monitoring* → *Process Overview*, transaction SM50) displays a list of work processes for the instance to which you are logged on. The dispatcher and all work processes are visible at operating system level as *disp+work.exe*. An assignment can be made using the process ID.

Continued on next page

Task 2: Starting from the Command Line (Optional)

Start the SAP system using a command call.

1. Start the SAP instances by entering the command *startsap*. Start the central instance first and then the dialog instances.
 - a) Start the SAP instances by entering the command *startsap* in a command prompt (*cmd*).

startsap syntax:

startsap name=<sid> nr=<instance no.> SAPDIAHOST=<server name>

Start the central instance first and then the dialog instances.



Lesson Summary

You should now be able to:

- Describe the process of the start procedure of an SAP system
- Start the entire SAP system or individual instances

Related Information

For further information about the SAP Management Console, refer to the following SAP Notes:

- 936273 - *sapstartsrv for all platforms*
- 927637 - *Web service authentication in sapstartsrv as of release 7.00*
- 823941 - *SAP Start Service on Unix*
- 995116 - *Backward porting of sapstartsrv for earlier releases*

Lesson: System Start: Logs

Lesson Overview

In this lesson, you will become familiar with the most important log and trace files, in which the start of an SAP system is logged.



Lesson Objectives

After completing this lesson, you will be able to:

- Evaluate start logs to analyze problems

Business Example

Problems occurred when starting an SAP system. To correct these problems, the administrator analyzes logs and trace files that were generated during the system start.

Logging the Start Process

The start process is an especially important phase, which is logged by the operating system, the SAP system, and the database. If the SAP system does not start up, you can find the relevant error message in the log files. It may be that there were problems starting the database, meaning that the SAP system could not subsequently be started.

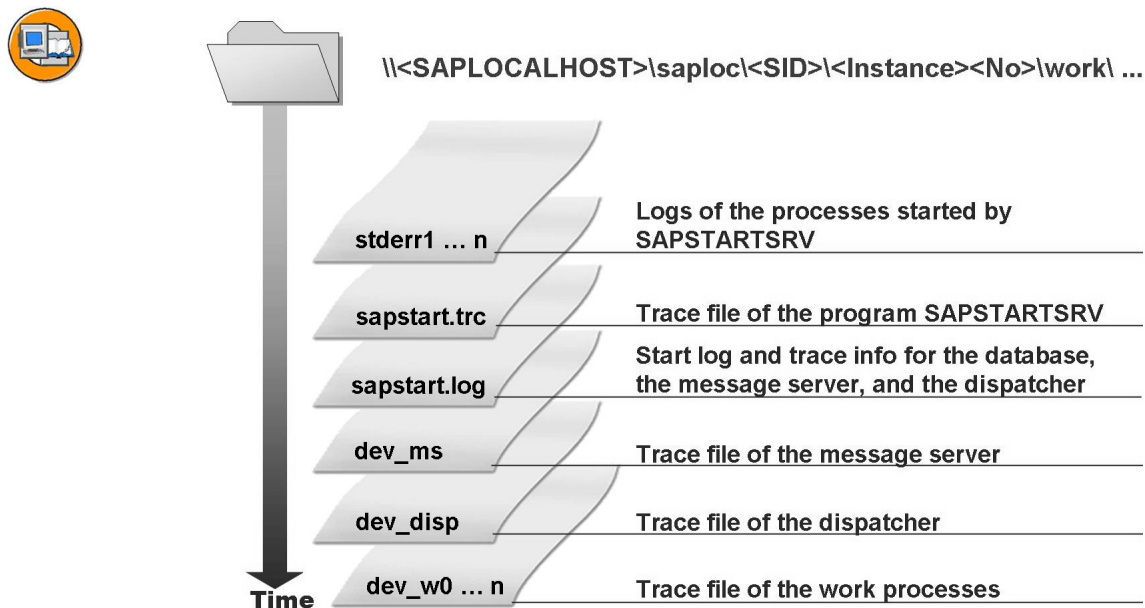


Figure 67: Logging the Start Process in the SAP System

Logs about the start process of the SAP system are stored in the file system. If there are problems during the start, these logs can provide useful information such as error messages or problem descriptions. These files are in the home directory (*DIR_HOME*) of the respective instance.

During the start process, the *STDERR<n>* log files are created by the SAP service. The starting processes write to the individual files, depending on the sequence in which they are listed in the start profile. The contents of these log files therefore depends on the individual system setup, and could, for example, be as follows:

- *STDERR1*: Information about the start process of the database system.
- *STDERR2*: Information about the start process of the message server.
- *STDERR3*: Information about the start process of the dispatcher.

You can set the level of detail of the logged information to four levels using the *rdisp/TRACE* profile parameter. The possible values for this parameter are:

- 0: Errors only
- 1: Error messages and warnings (default)
- 2: Error messages and a short trace
- 3: Error messages and a complete trace

The higher the trace level, the larger the amount of logged information, and therefore the larger the size of the files. You should therefore only increase the default value for short periods for problem analysis.

The trace level can be set separately for individual work processes in the process overview (transaction SM50).

Problem Analysis

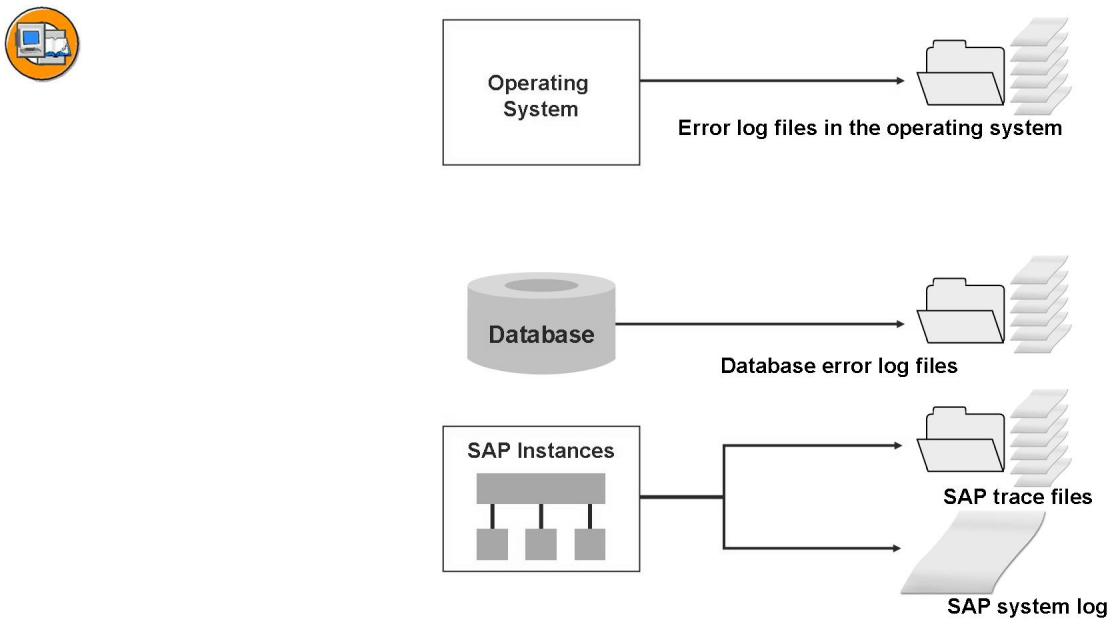


Figure 68: Problem Analysis

If the SAP system does not start correctly, this can be due to a variety of reasons. To analyze the problem, proceed as follows:

- Check the error messages and warnings of the respective operating system with the corresponding operating system tools.
- Check the status of the respective database system using the error log files. You will find information about this in the lesson: “Appendix - Database Logs”.
- Check the start log in the SAP MC. Select the instance that is affected, and from the context menu, choose *List Developer Traces*.
- Check the error files *stderr<n>* that were created by the SAP Service.
- Check the trace files of the individual SAP work processes:
 - *dev_ms*: Developer trace for the message server
 - *dev_rd*: Developer trace for the gateway
 - *dev_disp*: Developer trace for the dispatcher
 - *dev_w<m>* (m is the work process number): Developer trace for the work processes

If you can still log on to the SAP system, check the system log of the SAP system using transaction SM21.



Lesson Summary

You should now be able to:

- Evaluate start logs to analyze problems

Lesson: System Shutdown: How and Why?

Lesson Overview

This lesson demonstrates the stopping of an SAP system, including the preparatory activities in the system. Stopping the database is a separate process. The sequence in which the SAP instances are shut down is important: first the dialog instances, and finally the central instance.



Lesson Objectives

After completing this lesson, you will be able to:

- Stop the entire SAP System or individual instances

Business Example

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

Stopping the SAP System

It may be necessary to stop an SAP system for a variety of reasons, for example, restarting after permanently changing profile parameters, before installing kernel patches, or allowing hardware upgrades to be carried out.

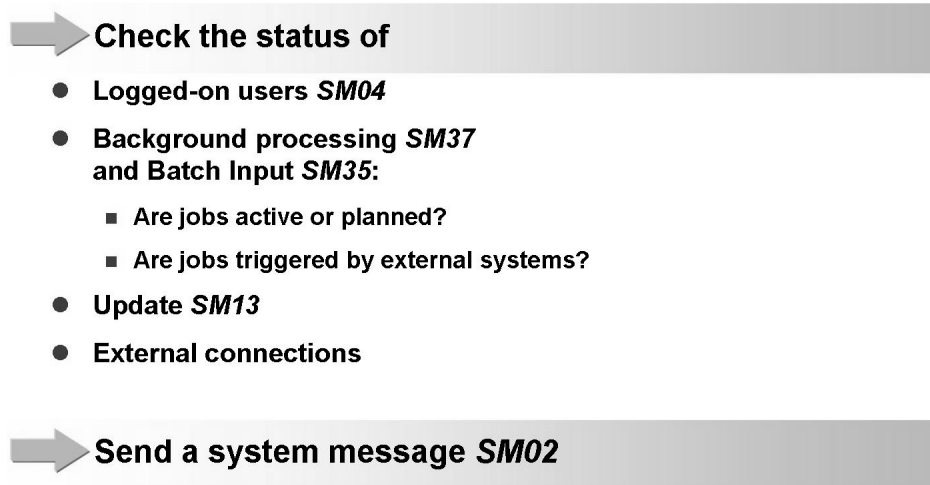


Figure 69: Before Stopping the SAP System

Before you stop the system, you should check the status of the system. This involves, among other things:

- Active Users:
Check which users are logged on using the User List (SM04).
- Background Processing :
Check which jobs are active using the Job Overview (SM37). If jobs are terminated by the system stop, these jobs must be rescheduled. Jobs that are scheduled for the time when the system is stopped run automatically once the system restarts.
- Batch Input:
The transaction Batch Input: Session Overview (SM35) displays running batch input jobs.
- Update:
Use the Update Overview (SM13) to check whether update processes are terminated by the system stop. These update records are rolled back during the stop, and are set to the status “init”. These records are then updated again during the restart.

Before you stop your system, you should inform users using a system message (SM02).

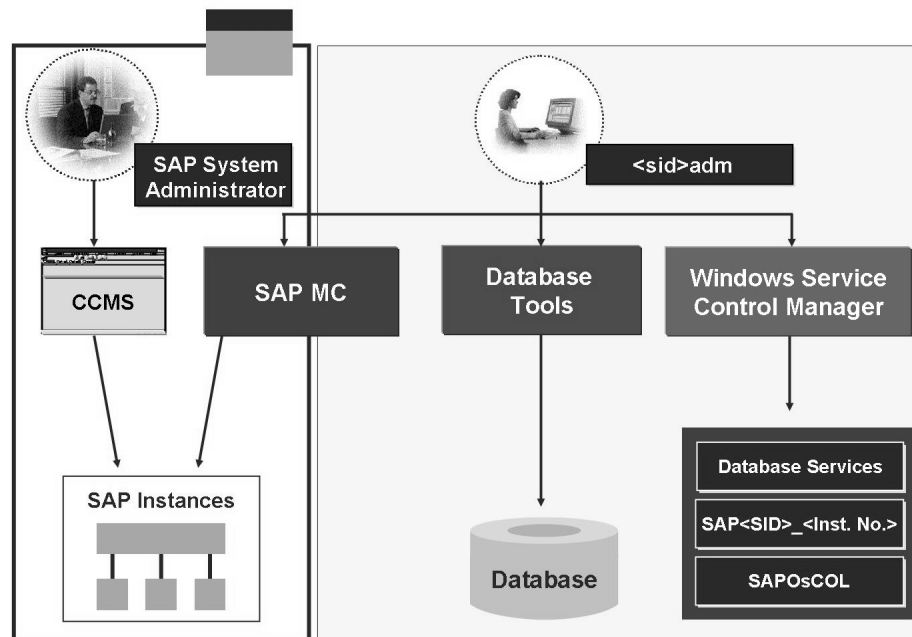


Figure 70: Stopping the SAP System

The stopping of the SAP system is performed in the opposite order of starting.

Stop the instances:

- In the SAP system itself within the CCMS (transaction RZ03) by choosing *Control* → *Stop SAP Instance*.
- In the Microsoft Management Console, right-click to show the context menu and choose the *Stop* function. Depending on whether you have selected an individual instance or the SAP system, the following are stopped:
 - A single instance
 - Central instance and all dialog instances

The SAP service waits for a stop message from the SAP MC or from the CCMS and then stops the SAP system. The service itself does not stop.

The services themselves can be stopped and restarted with the corresponding operating system tools, such as the Windows Service Control Manager.

The database is stopped using the relevant database system tools.

Exercise 11: Stopping an SAP System

Exercise Objectives

After completing this exercise, you will be able to:

- Stop an SAP system

Business Example

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

Task 1: Stopping with the SAP Management Console

Stop the SAP system using the SAP Management Console.

1. Before you stop the SAP system, check which users are logged on to your system. Send a system message.
2. Stop the SAP system using the SAP Management Console.

To do this, log on to the server for your training system with the Terminal Server Client.

Task 2: Stopping from the Command Line (Optional)

Stop the SAP system using a command call.

1. Log on to the server with the user <sid>adm over a Telnet connection.
2. Check whether your SAP system is running at operating system level.
3. Enter the command *stopsap* to stop an instance. To stop the entire SAP system, first stop the dialog instance and then the central instance.

The command *stopsap* only stops the selected instance. If the database is to be stopped, this must be done using database tools.

Solution 11: Stopping an SAP System

Task 1: Stopping with the SAP Management Console

Stop the SAP system using the SAP Management Console.

1. Before you stop the SAP system, check which users are logged on to your system. Send a system message.
 - a) You can use the global user overview to display the users logged on to your SAP system: *Tools → Administration → Monitor → Performance → Exceptions/Users → Active Users → Global Users* (transaction AL08). In this transaction, you will find your own user more than once. Transaction AL08 uses RFC connections to collect the user information for all instances. This means that your own user, which initiates the RFC connections, is displayed additionally once per instance.

To send a system message to the users, choose *Tools → Administration → Administration → System Messages* (transaction SM02). Choose the *Create* pushbutton, enter the system message text, and choose *Save*.

2. Stop the SAP system using the SAP Management Console.

To do this, log on to the server for your training system with the Terminal Server Client.

- a) Start a Web browser and enter **http://twd-fxxxx.wdf.sap.corp:5\$\$13** in the address bar (where **xxxx** stands for the number of the server assigned to your group and **\$\$** represents the instance number) and choose *Start*.

In the SAP MC, select the desired instance(s) and choose the *Stop* function from the context menu (right mouse button). The system now requires you to authenticate your ID. To do this, enter the user **<sid>adm** and the password **<sid>adm**. Depending on whether you selected an individual instance or the SAP system, either an individual instance or the entire SAP system is stopped.

Task 2: Stopping from the Command Line (Optional)

Stop the SAP system using a command call.

1. Log on to the server with the user **<sid>adm** over a Telnet connection.
 - a) Use the tool specified by the instructor to create a Telnet connection to your server. Log on with the user **<sid>adm** and password **<sid>adm**.

Continued on next page

2. Check whether your SAP system is running at operating system level.
 - a) To do this, execute the **tasklist** command to display an overview of operating system processes.
3. Enter the command *stopsap* to stop an instance. To stop the entire SAP system, first stop the dialog instance and then the central instance.

The command *stopsap* only stops the selected instance. If the database is to be stopped, this must be done using database tools.

- a) Enter the command **stopsap** to stop an instance.

stopsap syntax:

```
stopsap name=<sid> nr=<instance no.>  
SAPDIAHOST=<server name>
```

To stop the entire SAP system, first stop the dialog instance and then the central instance.



Lesson Summary

You should now be able to:

- Stop the entire SAP System or individual instances

Lesson: Appendix - Starting and Stopping: Operating System-Specific

Lesson Overview

In this lesson, you receive information about starting and stopping an SAP system under different operating systems.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the starting and stopping of an SAP system under different operating systems

Business Example

As a system administrator, you need to be able to start and stop the SAP system.

The Services Concept with Windows

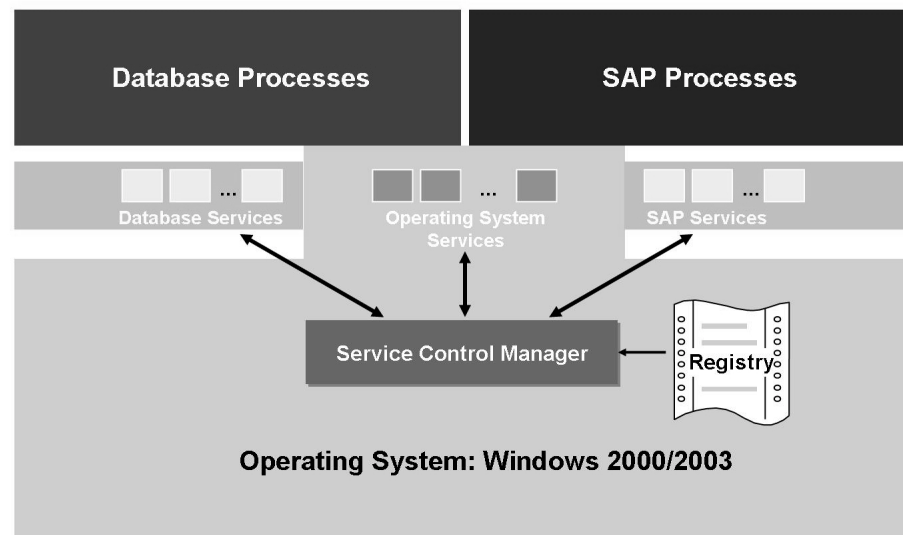


Figure 71: Services Concept with Windows

When starting programs in the Microsoft Windows environment, you should note that these programs are only active as long as the user is logged on to the system. When users log off, all their programs are terminated. This is why the SAP system uses the

services concept for starting. These are programs that are automatically started and managed by the operating system. Services provide support to other programs and run even if there are no users logged on to the host.

The Service Control Manager starts the services installed in the registry when Microsoft Windows 2000 is launched. All services can be configured for automatic startup.

During the installation of the SAP system, SAP and database services are installed in addition to the operating system services.

SAP Services:

- SAPOSCOL: Collects performance data for one or more SAP instances and runs once for each host.
- SAP<sid>_<instance no.>: Controls the SAP instances and runs once for each instance.

Database Services:

- Create the connection to the database.
- Control DB actions

Starting the SAP System

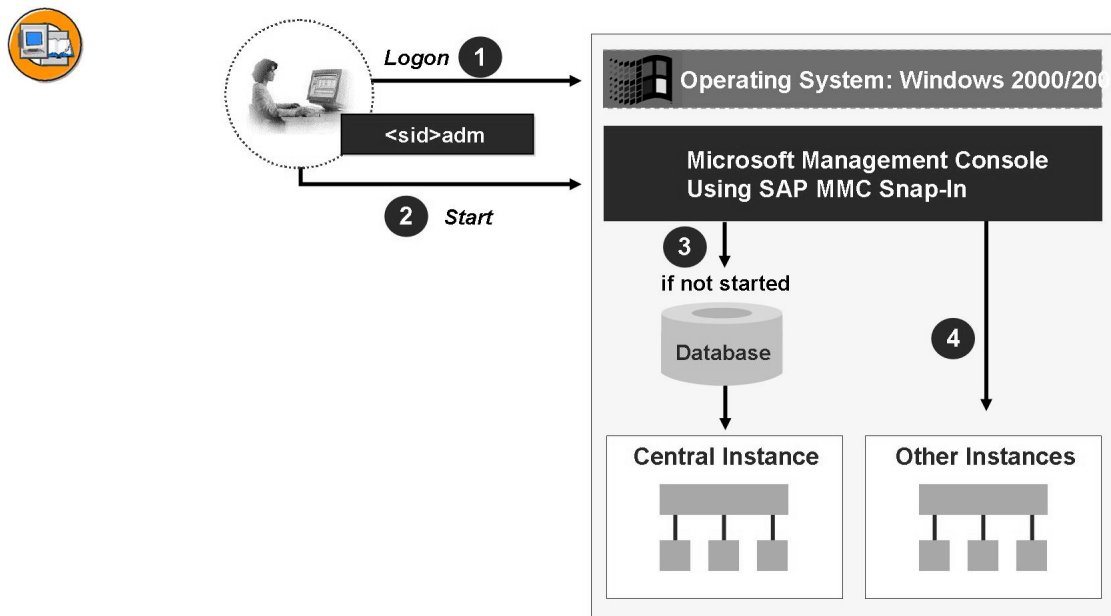


Figure 72: Starting the SAP System

With Microsoft Windows 2000/2003, you can start and stop the SAP system with the Microsoft Management Console (MMC).

To do this, the administrator logs on to the operating system with the user **<sid>adm** and opens the Microsoft Management Console.

Choose the node for the central instance in the tree. Call the context menu with the right mouse button and choose *Start*. The system first checks whether the database is active. If not, it is started automatically. If the database is active, the central instance (message server and dispatcher) is started by SAP Service SAP<SID>_<Instance no.>. The communication between the Microsoft Management Console and the SAP Service takes place through a “named pipe”.

Other instances can then be started.

The status of the SAP system, individual instances, and the message server and dispatcher are displayed in the Microsoft Management Console in accordance with the following color legend:

- **gray** not running
- **yellow** is starting
- **green** active
- **red** terminated after errors

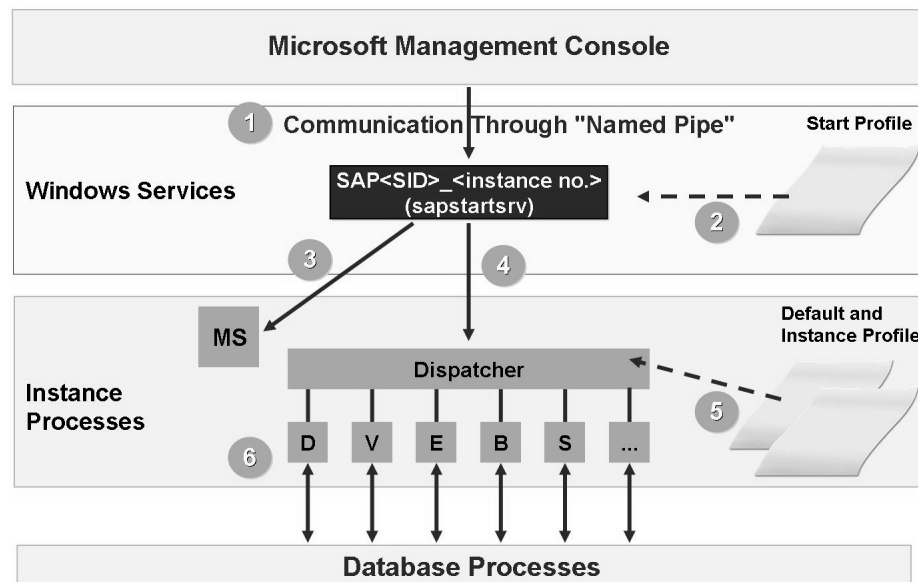


Figure 73: Start Sequence of SAP System Processes

The specifications of the instances; that is, the type and number of processes, main memory sizes, and other options are controlled using profiles when the instances are started. These are files at operating system level that are stored in the directory `\\<SAPGLOBALHOST>\sapmnt\<SID>\sys\profile`.

When the instances are started, SAP Service reads which processes (message server, dispatcher) are to be started from the instance-specific start profile. The start profile can be displayed in the Microsoft Management Console by right-clicking the entry for the instance and selecting the function *All Tasks* → *View Start Profile*.

The particular configuration of the instances is stored in the default profile and in the instance profile. These profiles are read by the dispatcher, which starts the work processes and creates the instance-specific configuration.

After the instance has been started successfully, all work processes connect to the database.

Logging the Start Process

The start process is a particularly important phase that is logged by the operating system, the SAP system, and the database. If the SAP system does not start up, you can find the relevant error message in the log files. It could have been that there were problems starting the database, meaning that the SAP system could not subsequently be started.

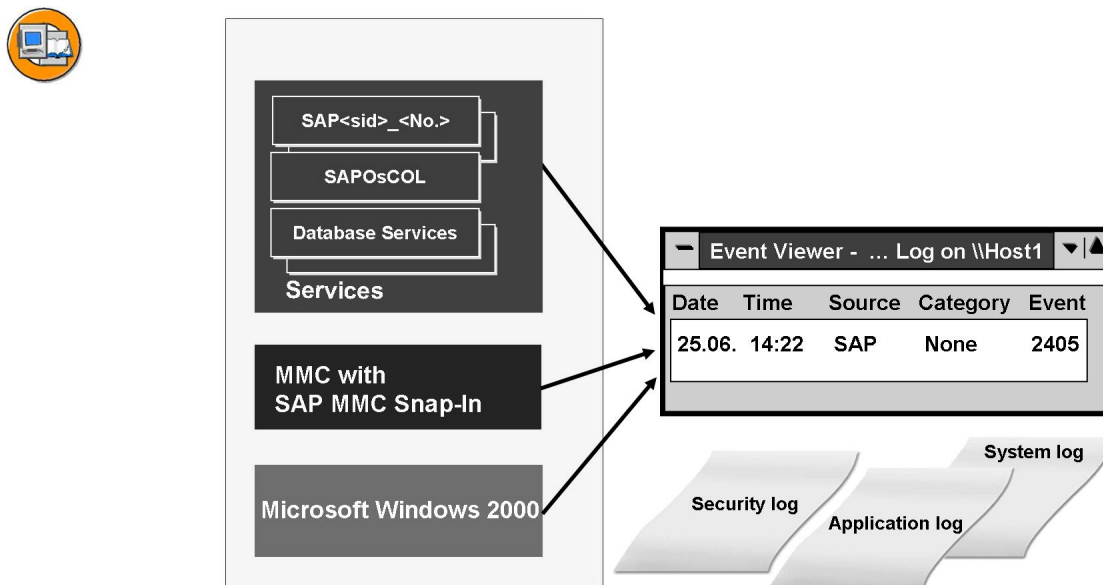


Figure 74: Logging of the Start Process with Windows

All messages that are created by services or the Microsoft Management Console are recorded at operating system level by an event logging service, the Event Manager. This Event Viewer writes an event log which contains the following three components:

Components of the Event Log

- **System Log:**
Operating system and application messages
- **Application Log:**
List of errors, warnings, and information that is generated by application software.
- **Security Log:**
Events such as log ons and log offs and user access to files.

You can call the event log by choosing *Start*→ *Programs*→ *Administrative Tools*→ *Event Viewer*. Choose the relevant component from the menu bar. You will see a list of the errors, warnings, and information that are generated in each case. For detailed information, double-click a particular log.

Logs about the start process of the SAP system are stored in the file system. If there are problems during the start, these logs can provide useful information such as error messages or problem descriptions. These files are stored in the home directory (*DIR_HOME*) of the relevant instance.

The log files *STDERR<n>* are created by the SAP service during the start process. These are independent of the operating system. For further details, see the lesson **System Start: Logs**.

UNIX

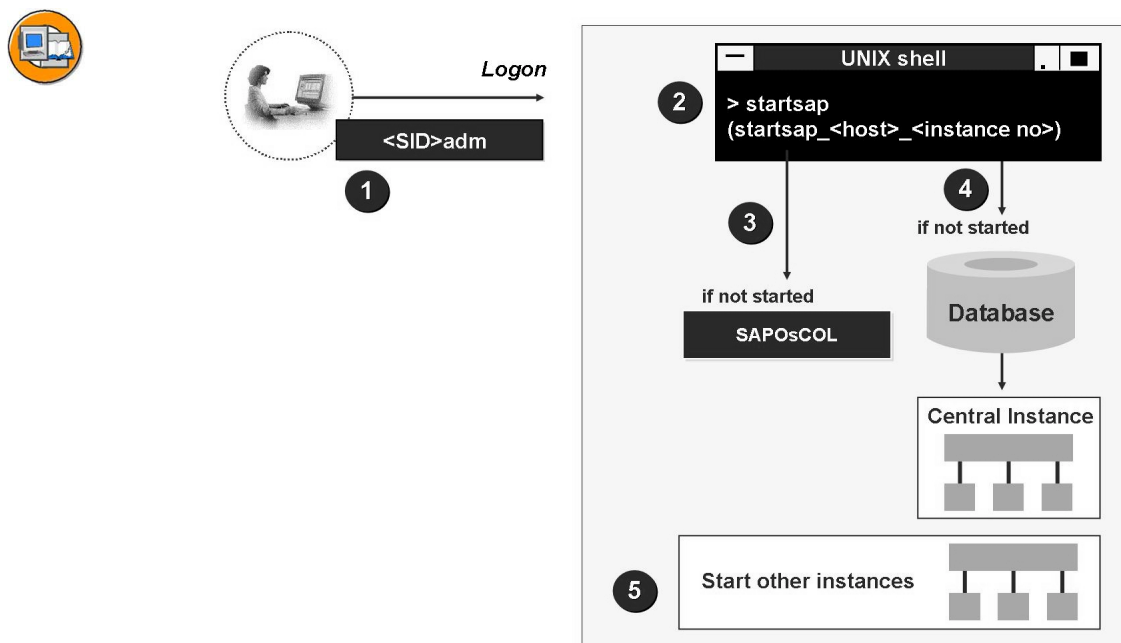


Figure 75: UNIX: Starting

Log on in UNIX as a user with SAP administrator authorization (`<sid>adm`).

To start the central instance, call the shell script `startsap_<host>_<instance no>` in the home directory of the `<sid>adm` user. If there is only one instance for each application server, the alias `startsap` references the script `startsap_<host>_<instance no>`.

The script `startsap` first starts the `saposc` process which collects statistical data for the operating system that it provides to the SAP system.

The next step is a check to see if the database is running. If not, the script `startdb` is called to start the database.

The final step starts all of the processes for the instance (message server, dispatcher, work processes, and so on).

After the central instance is started, you can optionally start other instances.

The `startsap` script can be called with the following options:

DB: starts the database system

R3: starts the instances and associated processes for the instance

ALL: starts the database system and the instance (default setting, can be omitted)

To start the SAP system, the `startsap` script calls the `sapstart` process with the start profile specified in the script in the variable `START_FILES`.

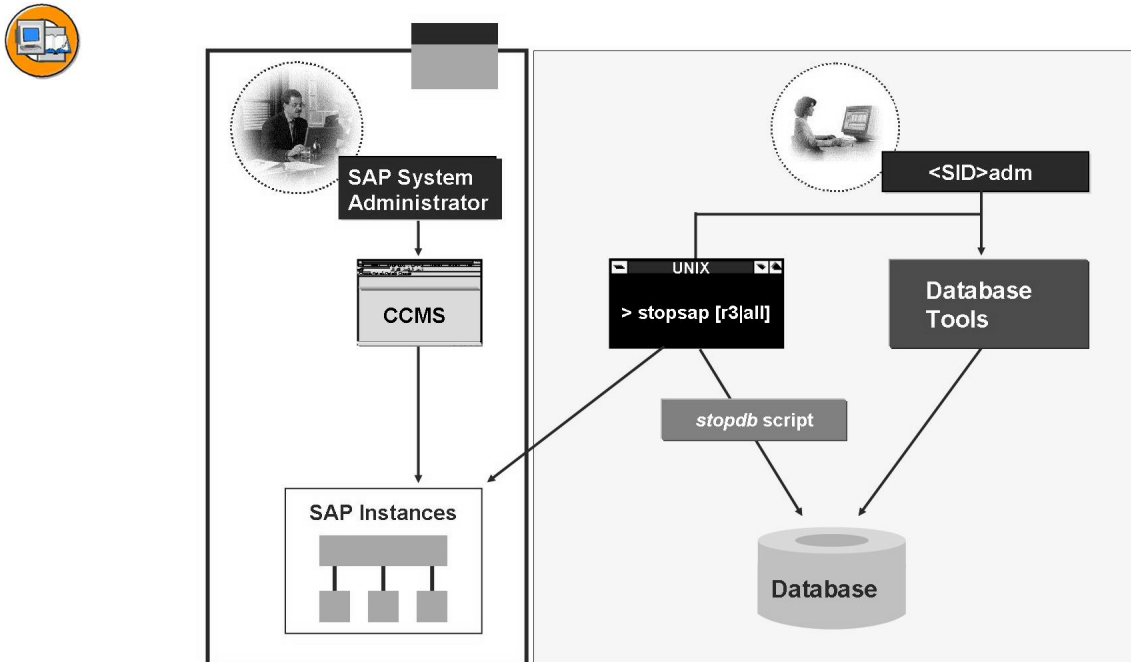


Figure 76: UNIX: Stopping

When you stop the SAP system, you should first stop all dialog instances and then stop the central instance. You have two options for doing this:

From the SAP system using the CCMS Control Panel.

Log on under UNIX as the SAP administrator (<sid>adm) at operating system level and enter the command `stopsap` from your home directory.

The `stopsap` script can be called with the following options:

- DB: stops the database system with the help of the `stopdb` script
- R3: stops the instances of the SAP system
- ALL: stops the database system and the instance (default setting, can be omitted)

The database can be stopped separately with database tools.

OS/400

➔ **Note:** The following section refers to AS/400. However, the statements apply equally to iSeries hosts.

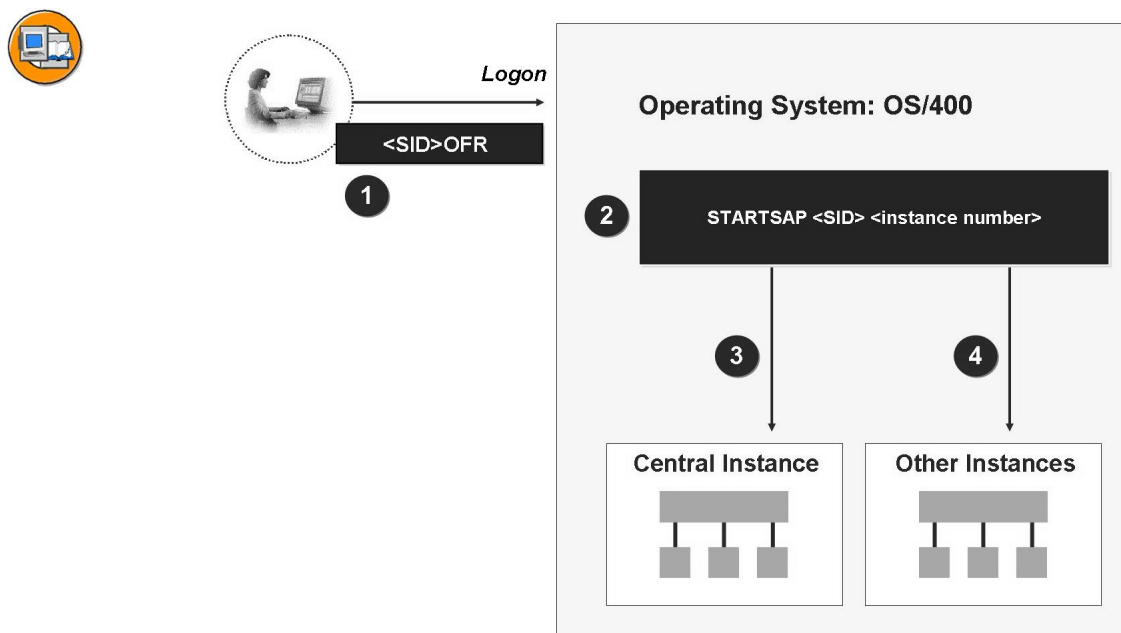


Figure 77: OS/400: Starting

Logon on to the AS/400 system with the SAP user profile for administrators. The authorizations of the group profile <SID>OPRGRP are required for this user (such as user profile <SID>OPR or <SID>OPR).

Enter the AS/400 command STARTSAP and request parameters with *F4*.

Under *SAP System ID*, enter the name of your system (such as DEV).

Under *R/3 Instance*, enter the instance number (such as 00). To start all instances on one or more hosts, choose **ALL*.

Under *R/3 Instance Host Name*, enter the name of the host on which the instance is to be started. To start all instances on all hosts, choose **ALL*. (You must also have selected **ALL* under *R/3 Instance*.)

Confirm your entries with *ENTER*. The subsystem R3_nn is then started for each started instance (<nn> is the instance number). All associated SAP services are started together with the subsystem (such as dispatcher, work processes, spool processes).

Enter the AS/400 command `WRKACTJOB SBS (R3_<nn>)` to check whether the SAP system(s) were started correctly.

The system displays a screen with the column *Status* at the right-hand side. If an SAP system was started correctly, the first line of this column should contain the value *RUN* (running), *SELW* (waiting), or *DEQW* (waiting for completion of unlocking). It can take a few minutes for the file system to reach this status.

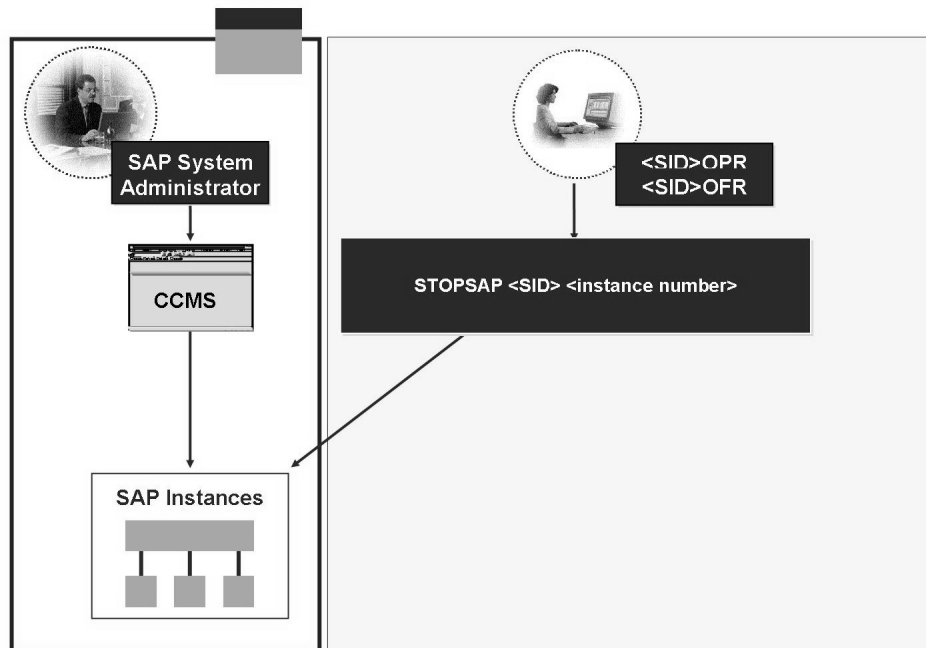


Figure 78: OS/400: Stopping

Logon on to the AS/400 system with the SAP user profile for administrators (<SID>OFR or <SID>OPR).

Enter the AS/400 command `STOPSAP` and request parameters with *F4*.

Under *SAP System ID*, enter the name of the SAP system that you want to stop.

Under *R/3 Instance*, enter the number of the instance that you want to stop, such as 90. To stop all instances on one or more hosts, choose **ALL*.

Under *R/3 Instance Host Name* enter **LOCAL* to stop one or more instances on the local host. To stop all instances on all hosts, choose **ALL*. (You must also have chosen **ALL* under *R/3 Instance*.)

If you enter **YES* under *Wait for instance to end*, the command `STOPSAP` waits until the SAP instance is shut down before stopping the SAP system. (The instance is regarded as shut down if the number of active instance user jobs in the instance subsystem, other than the `SAPOSCOL` job, is zero.)

Under *Maximum wait time (seconds)*, you can enter the maximum time that the command should wait for the instance to be shut down. The default value is 120 (two minutes). If it takes longer than two minutes for the instance to be shut down, an exception message is sent.

Confirm your entries with *ENTER*.

OS/390

➔ **Note:** The following section refers to S/390. However, the statements apply equally to zSeries hosts.

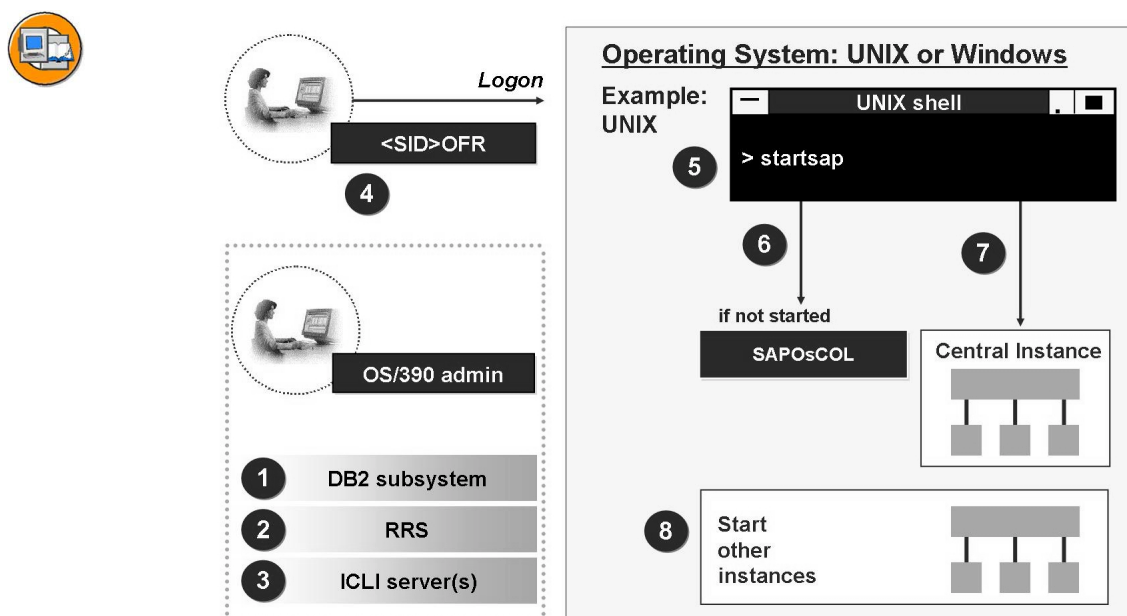


Figure 79: OS/390: Starting

In an OS/390 environment, an S/390 system serves as a database server. You can use Microsoft Windows or UNIX systems as application servers.

You must first start processes that allow the SAP work processes to connect to the database.

Start the DB2 database on the OS/390 system with the appropriate operating system commands.

Start the Recoverable Resource Manager (RRS) to synchronize all resources supported by OS/390.

Instances on non-OS/390 systems require the ICLI servers to allow communication with the database. To allow this, start the ICLI server(s) on the OS/390 system. Use either the OS/390 JCL operating language or a script which uses the OS/390 UNIX system services.

To start the SAP system, log on as operating system user <sid>adm on the UNIX or Microsoft Windows application server.

First, start the central instance. To do this, call the `sartsap` script under UNIX or use the Microsoft Management Console under Microsoft Windows. If the `saposc` process has not yet been started, this is started first. The processes of the central instances are then started.

The work processes then connect to the database.

Other instances can then optionally be started

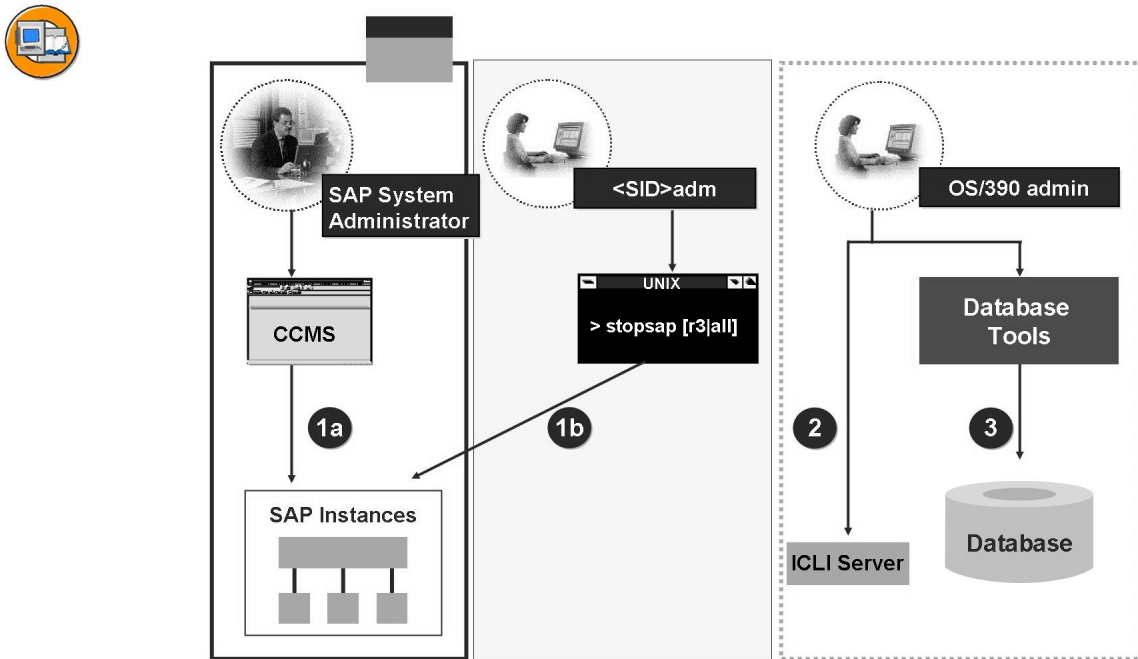


Figure 80: OS/390: Stopping

You can stop the SAP system using either the CCMS Control Panel or operating system resources. At operating system level, under UNIX, run the command `stopsap r3`, or use the Microsoft Management Console under Microsoft Windows.

The ICLI servers and the DB2 database are stopped on the S/390 system.

Stop the ICLI servers either on the OS/390 system console with the command `MODIFY BPXINIT, TERM=<PID>` or by using the OS/390 UNIX system services with the command `kill <PID>`.

Check that the ICLI servers have been stopped.

Stop the database.



Lesson Summary

You should now be able to:

- Describe the starting and stopping of an SAP system under different operating systems

Lesson: Appendix - Database Logs

Lesson Overview

This lesson provides information about the log files of various databases.



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the locations of the log files for various databases

Business Example

If an error occurs, the system administrator needs to access the database log files to find the cause of the error.

MaxDB



Note: The following statements apply in the same way for MaxDB-based SAP systems.

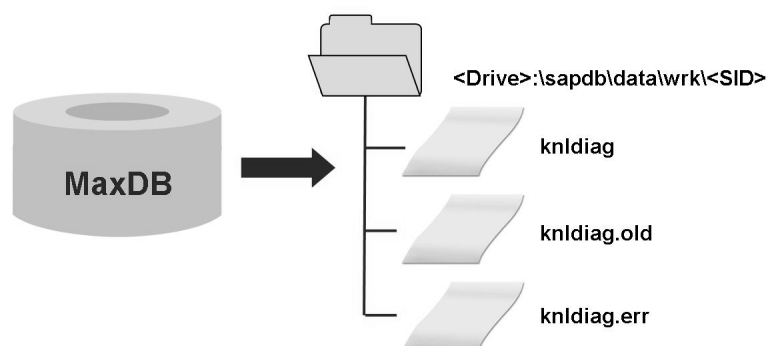


Figure 81: MaxDB

Max DB logs system messages and error messages in the directory:
<Laufwerk>:\sapdb\data\wrk\<SID>.

The system messages are recorded in the kernel log `knldiag`. This contains the following messages in chronological order:

- Database start and stop
- Information about the physical storage areas
- User processes
- System error messages

The log is run as a circular memory file that is overwritten as soon as it reaches a certain size. A new log file is created after every start of the database system. A backup copy of the old log (`knldiag.old`) is created before a restart of the database system.

All error and warning messages concerning the database system are recorded in the error log (`knldiag.err`), including the messages for the system start and stop.

MS SQL Server

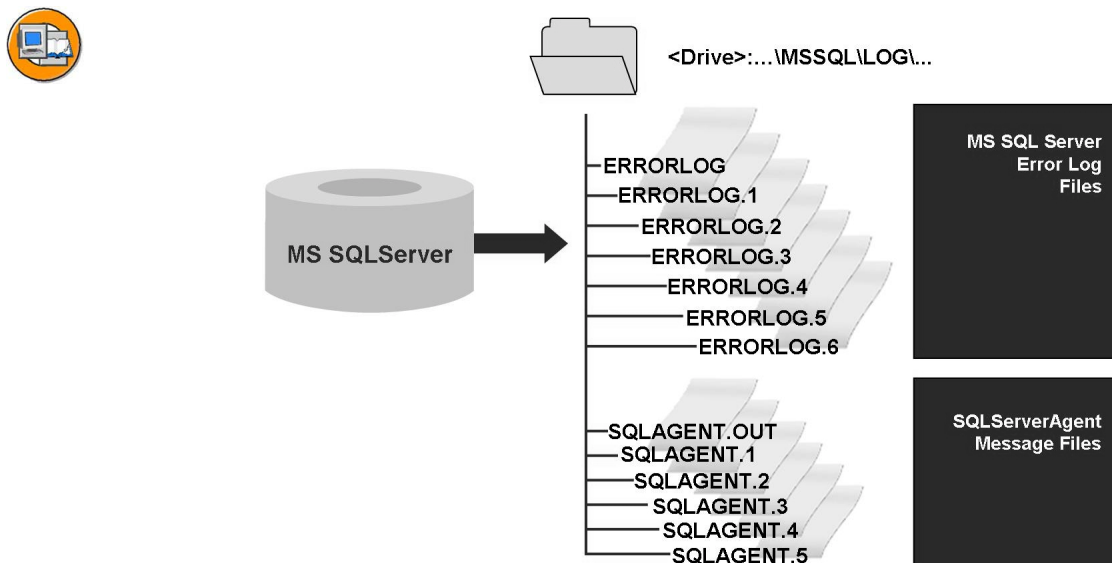


Figure 82: MS SQL Server

The MS SQL Server logs all significant events such as starting and stopping the database and error messages in the file `<Laufwerk>:...\MSSQL\LOG\ERRORLOG`. A new error log file is written at every start of the MS SQL server. Multiple versions of these error log files are stored as `ERRORLOG.1`, `ERRORLOG.2`, and so on. The

oldest available version is stored as `ERRORLOG . 6`. At every restart of the SQL server, the oldest file is overwritten. If a serious database problem occurs, you should backup the oldest file before the restart, so that this information is not lost.

These error log files can be displayed in the Enterprise Manager.

Messages of the “SQLServerAgent” service that is required for backups are also logged in the above directory in the file `SQLAGENT . OUT`. The last six versions of this log are also retained.

Oracle

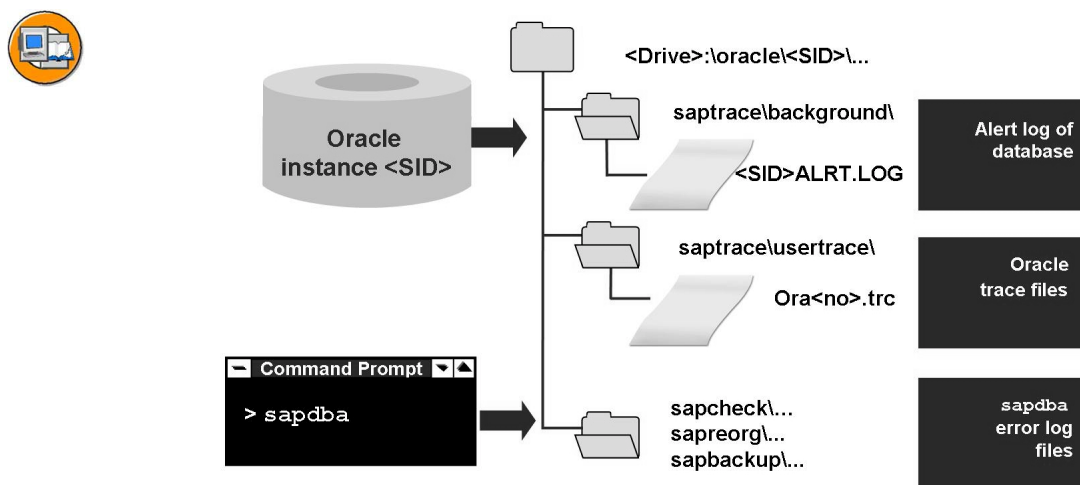


Figure 83: Oracle

The Oracle database logs all significant events such as starting and stopping the database and error messages in the file
 <Laufwerk>:\oracle\<SID>\saptrace\background\ALRT.LOG.

Detailed information about errors is logged in the Oracle Trace File:
 <Laufwerk>:\oracle\<SID>\saptrace\usertrace\Ora<no>.trc.

If the system administrator administers the database with sapdba, this writes its own log files in the following directories:

<Laufwerk>:\oracle\<SID>\sapreorg

<Laufwerk>:\oracle\<SID>\sapcheck

<Laufwerk>:\oracle\<SID>\sapbackup

DB2 (UDB)

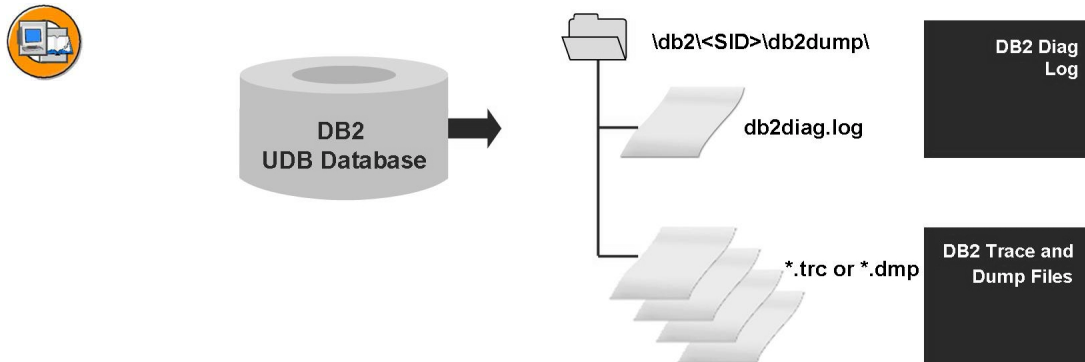


Figure 84: DB2 (UDB)

The DB2 database logs all significant events in the file `db2diag.log`. The path under which this file is stored is defined with the parameter Diagnostic Directory Data Path (*DIAGPATH*). This path is configured in the database manager configuration. The default value is `$DB2INSPROF/DB2INSTANCE`.

The `db2diag.log` file contains the following information:

- The place at which the reported error occurred. Application IDs allow the comparison of entries that belong to one application in the file `db2diag.log`.
- A diagnostic message with the reason for the error. The message usually begins with “DIA”.
- All other available data such as SQLCA data structures and pointers to other dump or trap files.

Detailed information about errors is logged in the DB2 trace or dump files, which are also stored in the path defined using parameter *DIAGPATH*. These files are only created if a serious internal DB2 error occurs.

You can access the dump directory by calling transaction DB6COCKPIT and choosing *Diagnostics* → *Dump Directory* in the navigation area.

If you want to display the contents of an error log or a trace file, double click the file.

Informix

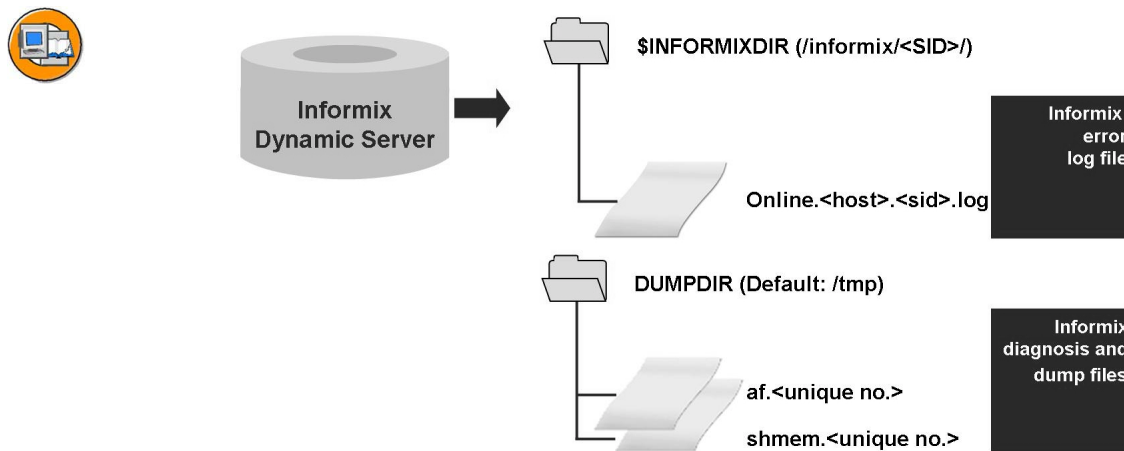


Figure 85: Informix

All significant events, such as starting and stopping the database and error messages are logged by the INFORMIX database in the file `$INFORMIXDIR/online.<hostname>.<SID>.log`.

Detailed information for individual errors is logged in the trace file `af.<unique no>`. In certain cases, the content of the shared memory is copied to the files `shmem.<unique no>`. The directory in which these files are stored is defined using the parameter *DUMPDIR*. The default value of this parameter is */tmp*.

DB2 (OS/390)

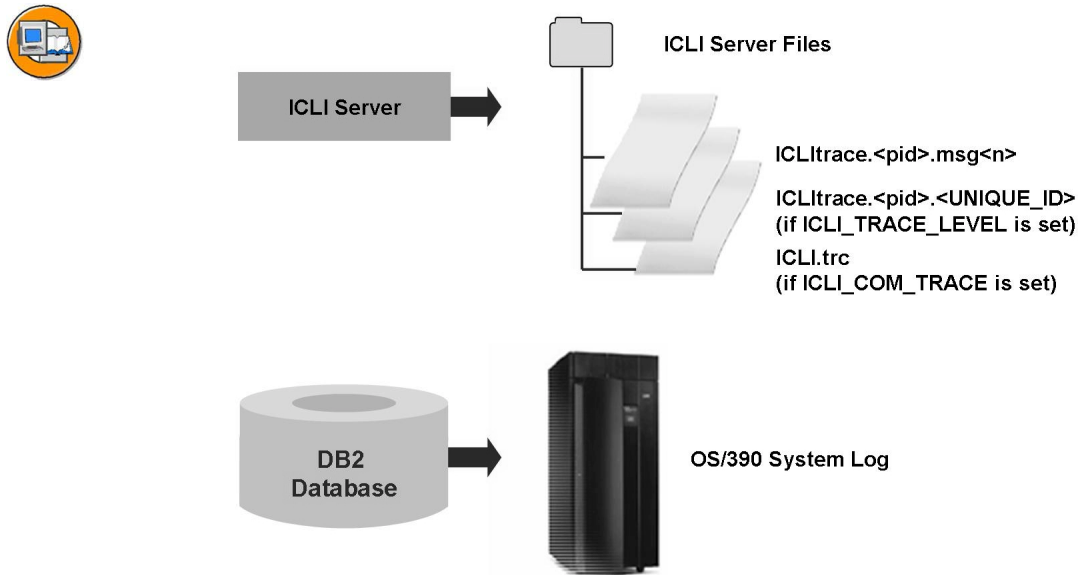


Figure 86: DB2 (OS/390)

ICLI Server Message Reporting:

ICLI Server Messages for one or more ICLI servers are written to one or more ICLI error log files. The file names follow the naming convention `ICLI.<pid>.msg<n>`, where `<pid>` is the process ID of the ICLI server and `<n>` identifies multiple error log files of an ICLI server.

The error log files of the ICLI are either stored in a directory that is specified during the start of the ICLI servers using the option `-LOGDIR`, or in the home directory of the user under which they were started.

ICLI Client Message Reporting:

The ICLI servers also send messages to the communication partners of the client. These are logged in the developer traces and in the system log of the SAP system.

The DB2 database logs all significant events and errors in the OS/390 system log.



Lesson Summary

You should now be able to:

- Describe the locations of the log files for various databases

Lesson: Overview of the Process for Starting and Stopping a SAP NetWeaver AS Java

Lesson Overview

There are different techniques for initiating the start and stop processes for the SAP NetWeaver AS, depending on the installation (with or without an ABAP stack). The Java-Stack of an SAP NetWeaver AS ABAP+Java is automatically started and stopped by the ABAP dispatcher. The start and stop process for an SAP NetWeaver AS Java (without ABAP stack) can be performed using the SAP Management Console (SAP MC).



Lesson Objectives

After completing this lesson, you will be able to:

- Describe the sequence in which the components of an SAP system are started and stopped
- Describe the general start process for an SAP NetWeaver AS Java
- Describe the general start process for an SAP NetWeaver AS ABAP + Java

Business Example

An SAP system should be stopped before maintenance work to the hardware and started again later. To be able to do this, it is necessary to become familiar with the tools for starting and stopping the system, and the process flow.

System Start: Process

The starting of an SAP system is the basic prerequisite for being able to work with the system. The start process presented in this lesson describes the process flow when starting an SAP system with ABAP and Java stack as well as when starting an SAP system with a Java stack but without an ABAP stack. Every SAP system has a database and at least one instance. One of these instances provides central functions and is therefore known as the central instance. This central instance and all other so-called dialog instances can consist either only of a Java stack (SAP NetWeaver AS Java) or of ABAP and Java stacks (SAP NetWeaver AS ABAP + Java). In the case of an SAP system that has at least one instance with a Java stack, there are also Central Services which must also be taken into account during the start process.

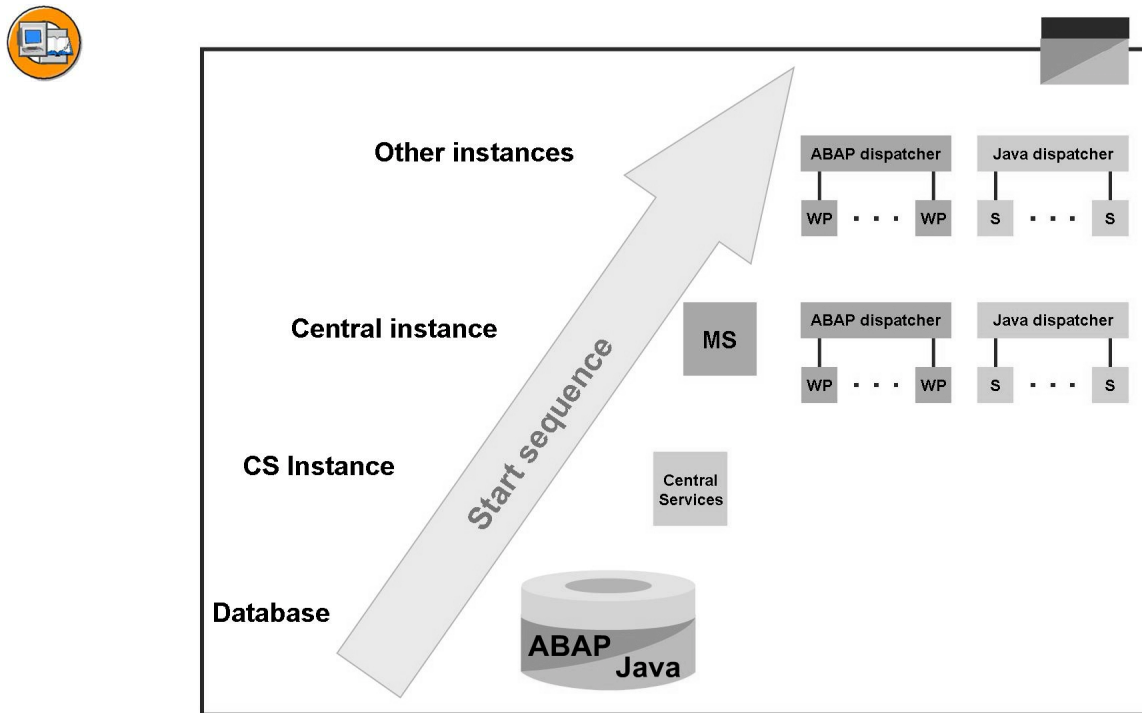


Figure 87: Start Process of an SAP System

In principle, the starting of the system is performed in multiple steps, and is the task of the operating system user <sid>**adm**:

1. Starting the database

The database is the fundamental element of the entire SAP system. This must be in an operational state before SAP instances are started.

2. Starting the Central Services Instance

The Central Services consist of the Java message server and the Java enqueue server. The cluster elements e.g. Java-dispatcher and Java-server connect to the Java message server during their own start process.

3. Starting the Central Instance

After the Central Services have been started, the central instance is started with the Java dispatcher and servers. The Java stack is started and stopped by the *Java Startup and Control Framework*.

The Software Deployment Manager (SDM) is also started with the central instance.

4. Start additional dialog instances

Other so-called dialog instances are started. When an instance is started, its Java dispatcher and servers are started.

For the start process, you differentiate between the starting of SAP systems with purely Java instances (without ABAP) and mixed instances (Java and ABAP stack). Additional details are provided in the following sections.

Stop Process: Process

The stopping of an SAP system is performed in reverse sequence. The dialog instances are always stopped first, and then the central instance.

For SAP system installations on the Microsoft Windows operating system, the database is not stopped at the same time. This can be done with the tools of the relevant database. In the case of SAP system that are installed on UNIX operating systems, the database is generally stopped at the same time.

Starting and Stopping an SAP NetWeaver AS ABAP + Java

During the installation of an SAP system, you can add a Java stack to each ABAP instance. The instances are started and stopped using appropriate tools at operating system level - such as the SAP Management Console (SAP MC). Within an instance,

the Java stack is controlled by the ABAP dispatcher. In concrete terms, this means that the start and stop processes are triggered by the ABAP dispatcher. To do this, the ABAP dispatcher sends a start command to the so-called *Startup and Control Framework*. The corresponding Java cluster elements are started using the *Startup and Control Framework*.

➔ **Note:** The *Startup and Control Framework* is the infrastructure that SAP provides for starting and stopping of Java cluster instances.

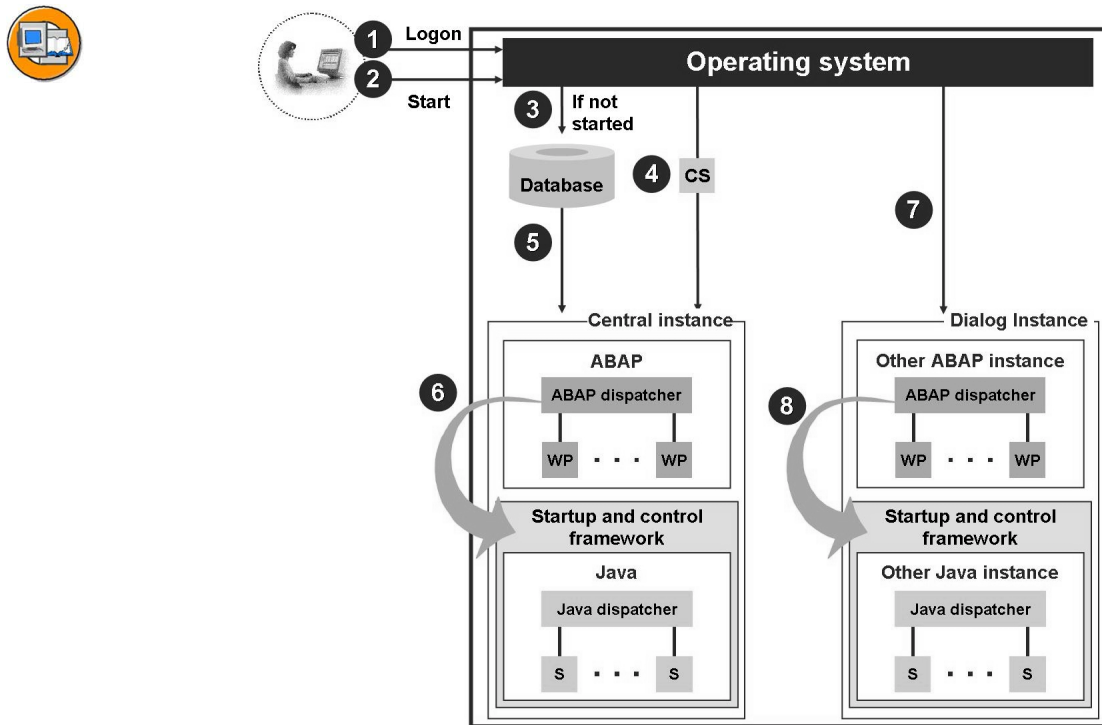


Figure 88: Starting an SAP NetWeaver AS ABAP+Java

The stop process is performed by the ABAP dispatcher in the same way as the start process. The ABAP dispatcher also informs the *Startup and Control Framework* and transfers the stop command in this case.

Starting an SAP NetWeaver AS Java

For some components, such as SAP Enterprise Portal or SAP NetWeaver Developer Studio, you only require a SAP NetWeaver Application Server as a technical foundation. This means that only pure Java instances without an associated ABAP stack are installed.

Pure Java instances can therefore not be managed by the ABAP dispatcher. The start and stop processes can be initiated using appropriate operating system commands - such as the SAP Management Console (SAP MC) or the Microsoft Management Console (SAP MMC) under Windows as well as the scripts `startsap` and `stopsap` under UNIX.

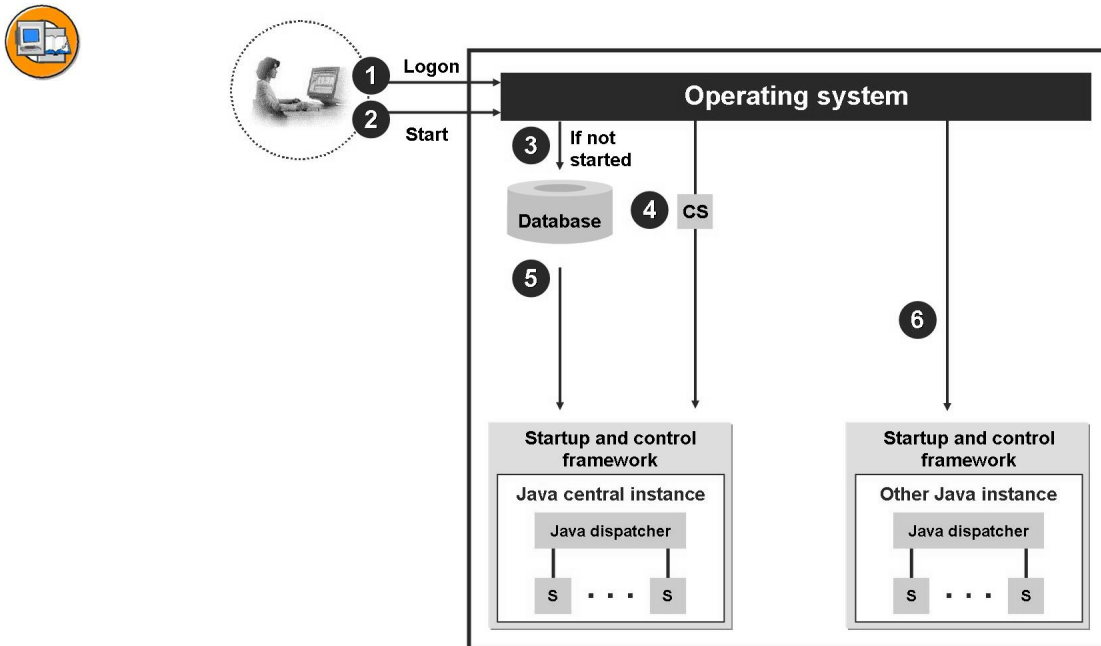


Figure 89: Starting a SAP NetWeaver AS Java (Without ABAP)



Lesson Summary

You should now be able to:

- Describe the sequence in which the components of an SAP system are started and stopped
- Describe the general start process for an SAP NetWeaver AS Java
- Describe the general start process for an SAP NetWeaver AS ABAP + Java

Related Information

- SAP Help Portal: . *help.sap.com* -> *SAP NetWeaver*

Lesson: Tools for Starting and Stopping

Lesson Overview

This lesson presents the tools for the technical implementation of a start and stop process for SAP systems. Independently of the operating system, the SAP Management Console (SAP MC) can be used for the start and stop process. There are also special tools for Windows and UNIX operating systems.



Lesson Objectives

After completing this lesson, you will be able to:

- Operate the tools to start and stop SAP NetWeaver AS ABAP + Java
- Operate the tools to start and stop SAP NetWeaver AS Java

Business Example

You are using an SAP NetWeaver Application Server with Java and different operating system platforms such as Microsoft Windows and UNIX. To start and stop the SAP systems you require information about the use of the available tools.

Starting and Stopping an SAP NetWeaver AS ABAP + Java

When SAP NetWeaver AS ABAP + Java is started, the entire instance is always started. If the instance is started then the Java stack can be stopped and started again individually by the ABAP dispatcher.

Starting and Stopping instances of SAP NetWeaver AS ABAP + Java

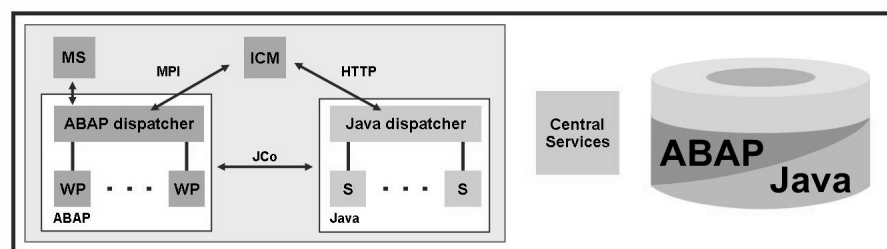


Figure 90: Starting and Stopping instances of SAP NetWeaver AS ABAP + Java

Instances of SAP NetWeaver AS ABAP + Java can be started and stopped independently of the operating system using the SAP Management Console (SAP MC).

As of SAP NetWeaver release 2004, a new operating system-independent application is supplied with the SAP Management Console (SAP MC) (see SAP Note 1014480). This allows you to display monitoring information and perform administration tasks such as start and stop operations. The tool is a standalone Java application which can be started either as an applet or locally. The SAP MC is supplied as standard with the start program *sapstartsrv* and is ready for use without any additional installation.

SAP MC is a Java application which is supplied as a Java applet as standard. You simply use a Web browser to do this.. To use the applet correctly, you require a JRE (Java Runtime Environment) of version 1.4 or higher. In addition, the corresponding Java plug-in for the browser must be activated correctly. To start SAP MC after installing the corresponding SAP NetWeaver AS, enter the following URL in your browser:

<http://<hostname>:5<instancenumber>13>

Or, if you have configured https in sapstartsrv (see SAP Note 1036107):

<https://<hostname>:5<instancenumber>14>

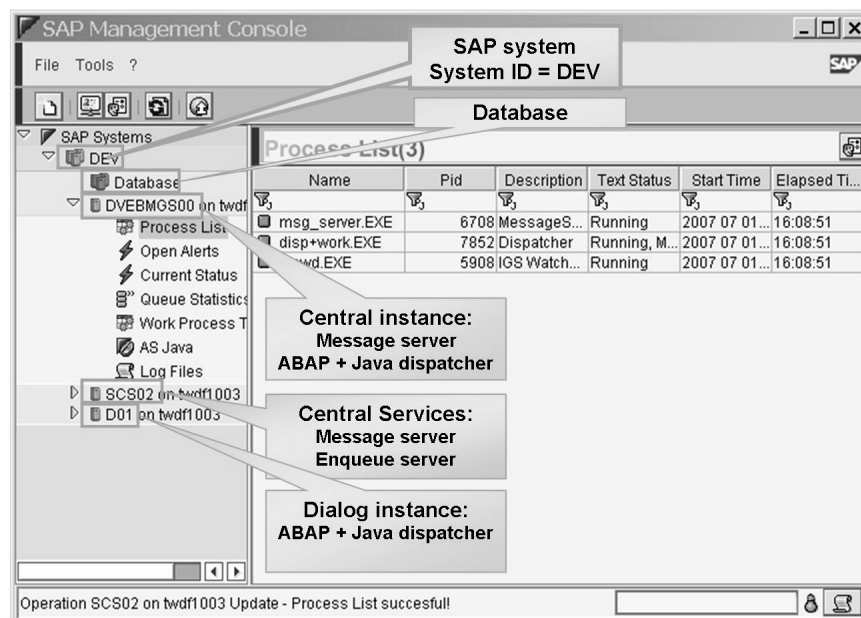


Figure 91: Starting and Stopping with the SAP Management Console

The SAP MC allows you to start and stop all the SAP NetWeaver AS ABAP + Java instances as well as the Central Services. You can also display information about the instances of the SAP system and the corresponding database (name, manufacturer and name of the host on which the database is located). Starting and Stopping with the SAP Management Console).

For each instance, SAP MC displays information about the ABAP and Java stack processes (see figure: SAP Management Console: Process Information).



Name	Pid	Description	Text Status	Start Time	Elapsed Ti...
msg_server.EXE	6708	MessageServer	Running	2007 07 01...	36:42:03
disp+work.EXE	7852	Dispatcher	Running, M...	2007 07 01...	36:49:44
igswd.EXE	5908	IGS Watchdog	Running	2007 07 01...	36:49:44

Nu...	Type	Pid	Status	Rea...	Start
0	DIA	1336	Wait		yes
1	DIA	5668	Wait		yes
2	DIA	7496	Wait		yes
3	DIA	8244	Wait		yes

Name	Type	Pid	State Text
SDM	SDM Server	5080	Running
dispatcher	J2EE Dispatcher	5460	Running
server0	J2EE Server	5492	Running

Figure 92: SAP Management Console: Process Information

The SAP Management Console also allows you to display the trace files for the individual processes.. You can use these trace files to analyze problems. However, these files also enable you to identify the employed ports (such as, for example, the message server port, http port, P4 port and telnet port) (see figure: SAP Management Console: Trace Files).

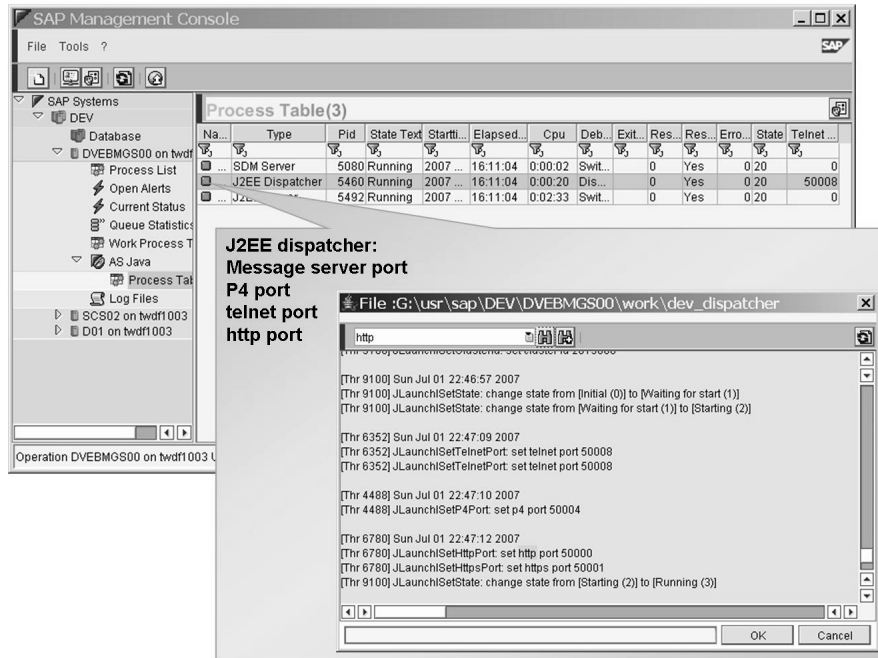


Figure 93: SAP Management Console: Trace Files

Stopping and Starting AS Java in an SAP NetWeaver AS ABAP + Java System

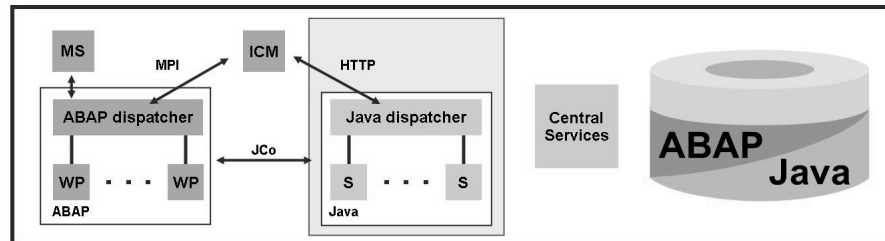


Figure 94: Stopping and Starting AS Java in an SAP NetWeaver AS ABAP + Java System

In the case of SAP NetWeaver AS ABAP + Java, it is possible to allow the ABAP stack to continue running, and only stop and then restart the Java stack. You do this using transaction SMICM. You can either start/stop the (local) instance onto which you are logged in the transaction SMICM or start/stop all the instances in the (global) Java cluster (see figure: Starting and Stopping the Java Stack of an SAP NetWeaver AS ABAP + Java from the Transaction SMICM).

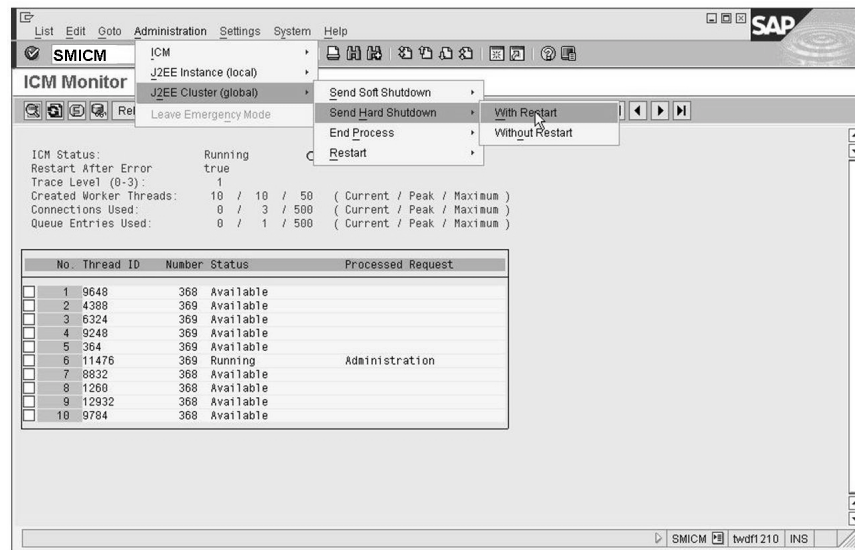


Figure 95: Starting and Stopping the Java Stack of an SAP NetWeaver AS ABAP + Java from the Transaction SMICM

It is not possible, and also not useful, to stop only the ABAP stack and leave the Java stack started in the case of AS ABAP + Java.

Starting and Stopping a SAP NetWeaver AS Java

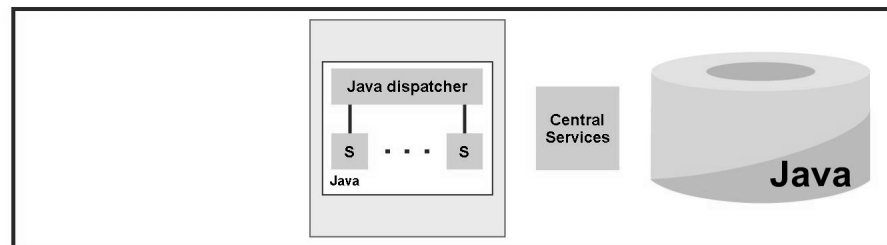


Figure 96: Starting and Stopping a SAP NetWeaver AS Java

SAP NetWeaver AS Java is started and stopped in the same way as SAP NetWeaver AS ABAP + Java by means of the SAP Management Console (see figure: Starting and Stopping SAP NetWeaver AS Java with the SAP Management Console).

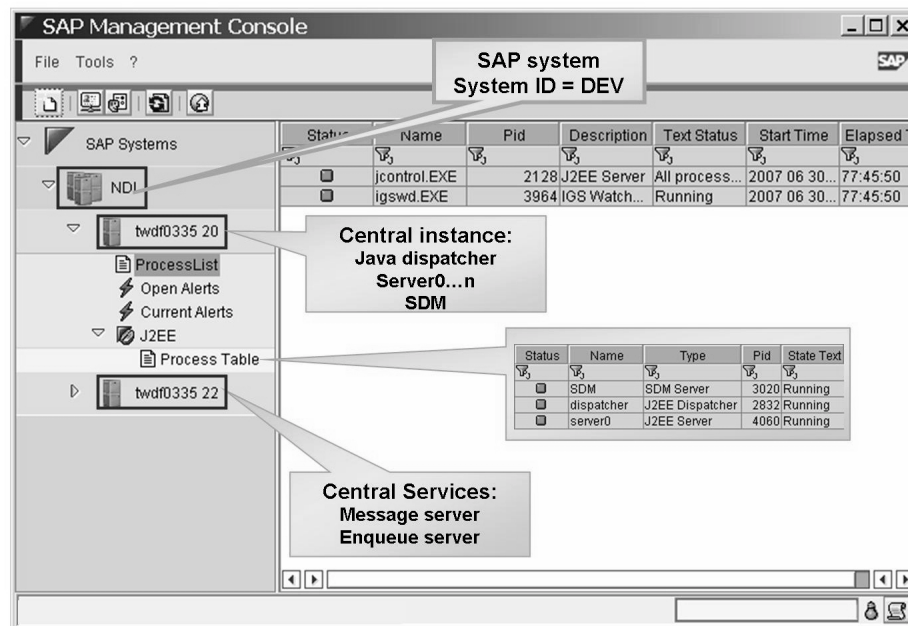


Figure 97: Starting and Stopping SAP NetWeaver AS Java with the SAP Management Console

In the case of SAP NetWeaver AS Java, the instance names are JC<instance-number> for the central instance and J<instance-number> for the so-called dialog instances.

Special Characteristics of Starting and Stopping Under Microsoft Windows

With the Microsoft Windows operating system, you can use both the SAP Management Console and the Microsoft Management Console with an SAP plug-in (SAP MMC) to start and stop an SAP system.

The SAP MMC allows you to start and stop all the instances of the SAP system as well as the Central Services. For some databases, you can also display administration information and the status of the database. Some database types can also be administered using the SAP MMC.

The information displayed in the SAP MMC and the start and stop procedures are similar to the browser-based SAP Management Console (SAP MC). This is illustrated by the next two figures.

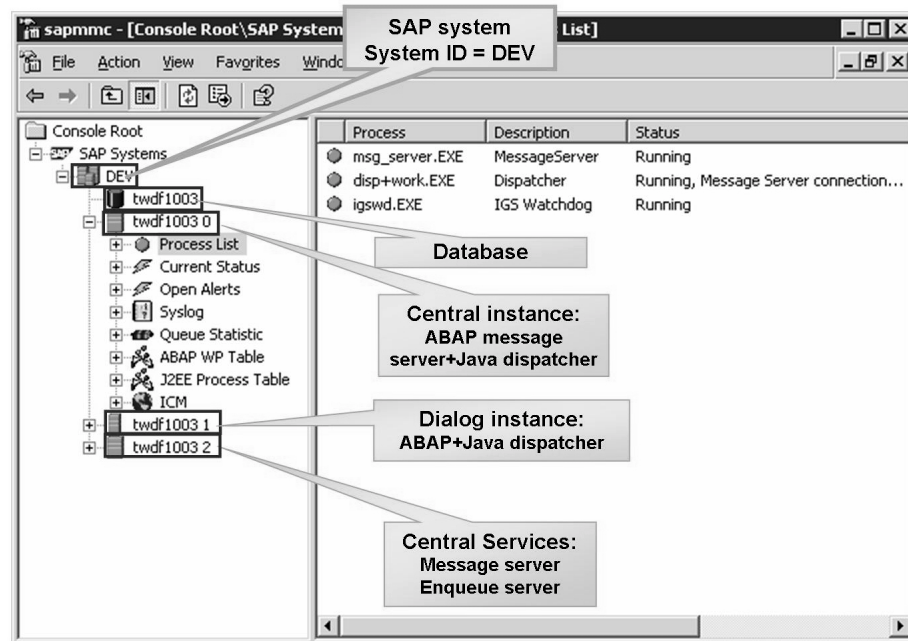


Figure 98: Starting/Stopping an SAP Net Weaver AS ABAP + Java under Microsoft Windows

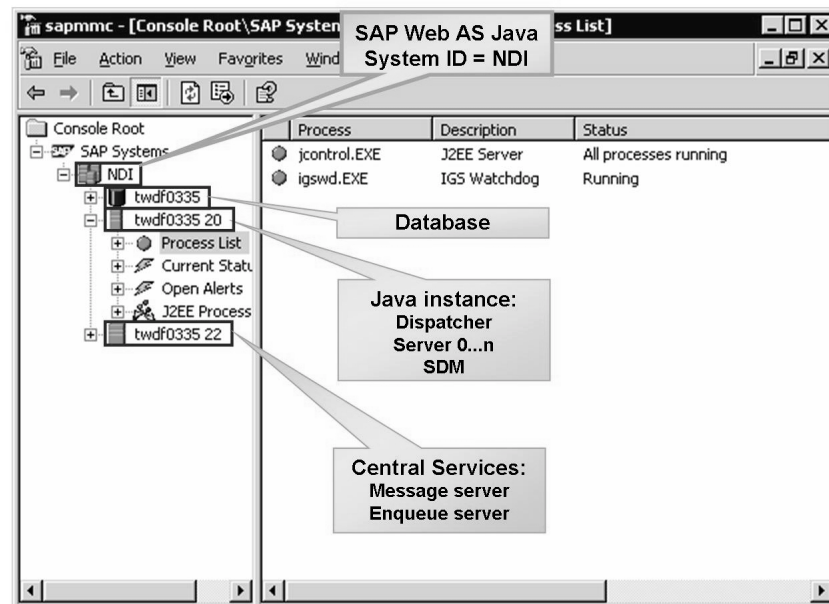


Figure 99: Starting and Stopping a SAP NetWeaver AS Java under Microsoft Windows

In the same way as the SAP MC, SAP MMC also makes it possible to display the employed ports via the trace files for the individual processes..



Hint: In the environment of SAP NetWeaver AS Java, there is a special Java development environment, which is installed locally on every PC. It usually consists of an SAP NetWeaver Developer Studio and an SAP Web AS Java as a test environment for Java developments. The SAP NetWeaver Application Server Java can be “restarted” from the SAP NetWeaver Developer Studio.

Under Windows, the SAP system can also be started and stopped without a GUI by calling a command by means of the executable files `startsap.exe` and `stopsap.exe`. This can be done using a simple telnet access.

To start an instance of the SAP system, open a telnet connection and enter the following command: `startsap name=<SID> nr=<instance nr.>`
`SAPDIAHOST=<hostname>`

To stop an instance of the SAP system, open a telnet connection and enter the following command: `stopsap name=<SID> nr=<instance nr.>`
`SAPDIAHOST=<hostname>`

For the `SAPDIAHOST` parameter, enter the name of the host on which the instance is to be started.

Special Characteristics of Starting and Stopping Under UNIX

For SAP systems that are installed under UNIX, you run the scripts *startsap* and *stopsap* to initiate the start and stop processes. If multiple SAP instances are installed on one physical server, you add the instance name to the names of the scripts *startsap* and *stopsap*.

The *startsap* script can be called with the following options:

- DB: starts the database system
- R3: starts the instances and associated processes for the instance
- J2EE: starts the instances and associated processes for the instance
- ALL: starts the database system and the instance (default setting, can be omitted)

The `stopsap` script can be called with the following options:

- `DB`: stops the database system with the help of the `stopdb` script
-
- `J2EE`: stops the instances of the SAP system
- `ALL`: stops the database system and the instance (default setting, can be omitted)

To start the SAP system, the `startsap` script calls the `sapstart` process with the start profile specified in the script in the variable `START_FILES`.

Database and SAP instances can each be individually started and stopped with the options `DB` or `R3`, e.g. `startsap R3 DVEBMGS00`.



Caution: The option `J2EE` can be used in the same way as the option `R3`. In the case of SAP NetWeaver AS ABAP + Java, both the ABAP stack and the Java stack are started and stopped.

Exercise 12: Tools for Starting and Stopping

Exercise Objectives

After completing this exercise, you will be able to:

- Start and stop an SAP NetWeaver AS Java under Microsoft Windows

Business Example

You are using a SAP NetWeaver Application Server with Java and the Microsoft Windows operating system platform. There is a special tool for starting and stopping an SAP system under the Microsoft Windows operating systems. This is the Microsoft Management Console. It is also possible to use the operating system-independent SAP Management Console.

Task 1: Stopping an SAP NetWeaver AS Java

You stop the SAP system using the Microsoft Management Console or the SAP Management Console.

1. Use the Terminal Server Client to log on to the operating system of your training system.
2. Start the Microsoft Management Console and stop your system.

Task 2: Starting an SAP NetWeaver AS Java

You start the SAP system using the Microsoft Management Console or the SAP Management Console.

1. Start the SAP system using the Microsoft Management Console. Monitor the starting of the corresponding Java processes of your SAP system.
2. Check whether your SAP NetWeaver AS Java has been correctly started. To do this, call the appropriate URL (*http://<host>:<port>*, such as *http://twdf9999:50000* for your system).

Task 3: Starting and Stopping with SMICM

OPTIONAL: Starting and stopping the Java stack of the dialog instances with transaction *SMICM*.

1. Stop the dialog instance's Java stack using transaction *SMICM*.
2. Start the Java instances using transaction *SMICM* and check whether it was possible to perform this successfully.

Solution 12: Tools for Starting and Stopping

Task 1: Stopping an SAP NetWeaver AS Java

You stop the SAP system using the Microsoft Management Console or the SAP Management Console.

1. Use the Terminal Server Client to log on to the operating system of your training system.
 - a) Start the Terminal Server Client, enter the physical host name under *Server* and then choose *connect*. Log on to the operating system as the *<sid>adm* user.
2. Start the Microsoft Management Console and stop your system.
 - a) Start the Microsoft Management Console by double-clicking the corresponding icon.
 - b) Select the desired instance(s) and choose the *Stop* function in the context menu (right-click). Depending on whether you selected an individual instance or the SAP system, either an individual instance or the entire SAP system is stopped.

Task 2: Starting an SAP NetWeaver AS Java

You start the SAP system using the Microsoft Management Console or the SAP Management Console.

1. Start the SAP system using the Microsoft Management Console. Monitor the starting of the corresponding Java processes of your SAP system.
 - a) Start the Microsoft Management Console by double-clicking the corresponding icon.
 - b) In the Microsoft Management Console tree, select the node for the central instance. In the context menu (right-click), choose the *Start* function. Start the dialog instances in the same way.



Hint: You should always start the central instance first.

- c) At operating system level, you can monitor the processes using the *Task Manager* or the *Quick Slice* tool (*Start -> Run: qslice*).

The following Java process types exist after your SAP system has been started: JControl, multiple JLaunch

Continued on next page

2. Check whether your SAP NetWeaver AS Java has been correctly started. To do this, call the appropriate URL (*http://<host>:<port>*, such as *http://twdf9999:50000* for your system.
 - a) Start Microsoft Internet Explorer on your desktop, and enter the following URL: *http://<hostname>:<port>*, e.g. *http://twdf12345:50000*. The start page of your SAP NetWeaver AS Java should now appear.

Task 3: Starting and Stopping with SMICM

OPTIONAL: Starting and stopping the Java stack of the dialog instances with transaction *SMICM*.

1. Stop the dialog instance's Java stack using transaction *SMICM*.
 - a) Log on to the dialog instance's ABAP stack in your SAP system. Start transaction *SMICM*. Navigate through the menu *Administration → J2EE Instance (local) → Send Hard Shutdown → Without Restart*.
 - b) At operating system level, go to the directory *G:\usr\sap\<SID>\<dialog instance>\work\dev_disp* and check whether the command was successfully executed. The following message should be displayed:
DpIJ2eeShutdown: j2ee state = SHUTDOWN
2. Start the Java instances using transaction *SMICM* and check whether it was possible to perform this successfully.
 - a) Log on to the central instance of your SAP system. Start transaction *SMICM*. Navigate through the menu *Administration → J2EE Instance (local) → Restart → Yes*.
 - b) At operating system level, go to the directory *G:\usr\sap\<SID>\<dialog instance>\work\dev_disp* and check whether the command was successfully executed. The following message should be displayed:
start = TRUE
state = ACTIVE
...
start_control = SAP J2EE startup framework

You should also call the following URL: *http://<hostname>:<port>*, e.g. *http://P12345:50100*. As soon as the SAP NetWeaver AS Java start page appears, the start process has been successfully completed.



Lesson Summary

You should now be able to:

- Operate the tools to start and stop SAP NetWeaver AS ABAP + Java
- Operate the tools to start and stop SAP NetWeaver AS Java

Related Information

SAP Help Portal: . help.sap.com → *SAP NetWeaver* → *SAP NetWeaver 7.0 (2004s)*

Lesson: Java Startup and Control Framework

Lesson Overview

The Java Startup and Control Framework coordinates the proper starting and stopping of the Java stack. It consists of the processes JControl and JLaunch. The functions of the processes are described in this lesson.



Lesson Objectives

After completing this lesson, you will be able to:

- Use the term Startup and Control Framework
- Describe the individual steps during the start and stop processes of a Java instance

Business Example

Starting and stopping an SAP system is a basic task for administrators of SAP systems. To understand parameter maintenance, it is important to understand how parameters are transferred to Java instances.

Startup and Control Framework

SAP provides a separate *Startup and Control Framework* for SAP Web Application Server Java. This Framework is used to start, stop, and monitor the Java stack processes within the SAP system instances - but not, however, the Central Services. It consists of the following processes:

JControl:

- JControl starts, stops, and monitors the processes of a Java instance – primarily the server and dispatcher processes. SAP Signal Handling is implemented with JControl, to forward the start and stop commands to the processes of the Java stack.
- JControl restarts terminated processes, ends hanging processes, and sends a shutdown signal to the processes of the Java stack.
- JControl reads the description of the instance from profile files.
- JControl starts the JLaunch processes, which in turn start Java nodes such as dispatchers or servers.
- JControl creates a shared memory segment for the internal administration data of all JLaunch processes.

JLaunch:

- JLaunch starts a Java program, loads a Java VM (JVM) in its own address space and assumes the function of the corresponding cluster element. The parameterizing of the JVM is read before the loading.
- JLaunch receives commands from the JControl process (through named pipes) to stop cluster elements such as dispatchers or servers.
- The JLaunch process ends itself if its parent process JControl is no longer running.

Start Process of a Java Instance

The start procedure in the *Startup and Control Framework*:

(See also figure: Starting with the Startup and Control Framework)

1. JControl is started.
2. JControl starts a Java instance bootstrap process with parameters that are defined in the file *bootstrap.properties* (*/usr/sap/<SID>/<instance>/j2ee/cluster/bootstrap*). The bootstrap process synchronizes all required binaries for the Java nodes from the database with the file system. This is necessary, since the *Class Loading* is performed using the file system in the Java environment. The database always contains the current (deployed) binaries and properties and distributes these to every Java node when starting.
3. The Java instance bootstrap process reads the instance description from the Java schema of the database.
4. The Java instance bootstrap process stores a file *instance.properties*, containing the instance description, at operating system level.
5. JControl reads the instance description from the file system.
6. JControl starts the bootstrap processes which are specific to Java nodes and which read the descriptions for the nodes from the database and store them in the file system.
7. JControl creates a connection to the Central Services message server. If this connection is established successfully, the operation continues with the next step. If no connection is established to the message server, JControl initiates an Emergency Shutdown.
8. In the final step, JControl starts all the Java nodes such as dispatchers and servers as JLaunch processes and, when doing so, loads a JVM in the JLaunch process.

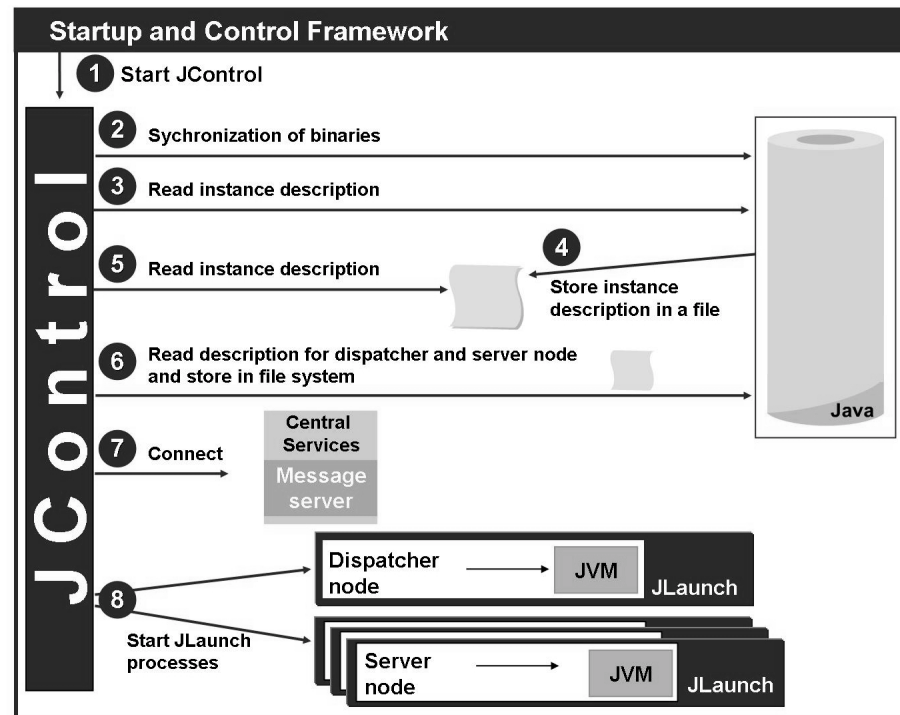


Figure 100: Starting with the Startup and Control Framework

JControl monitors the Java instance processes during their runtime, restarts terminated processes, ends hanging processes, and sends the shutdown signal to the JLaunch processes.

So-called profile files are first read when a Java instance is started. The JControl process then reads the files *instance.properties* and *bootstrap.properties*.

The profile files are located on the operating system in the directory DIR_PROFILE (Microsoft Windows: <drive>:\usr\sap\<SID>\SYS\profile or UNIX: /usr/sap/<SID>/SYS/profile) and are generated at installation time. There are three different profile files: the default profile (*Default.pfl*), the start profile (*START_<instance>_<host>*), and the instance profile (<SID>_<instance>_<host>).



Note: The Central Services profiles are read when the Central Services are started.

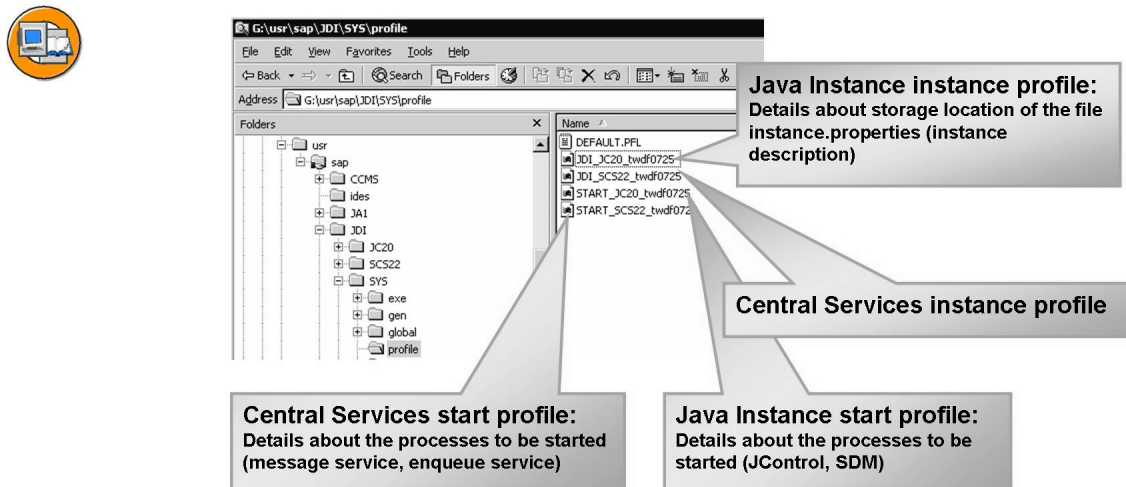


Figure 101: Profiles

Stop Process of a Java Instance

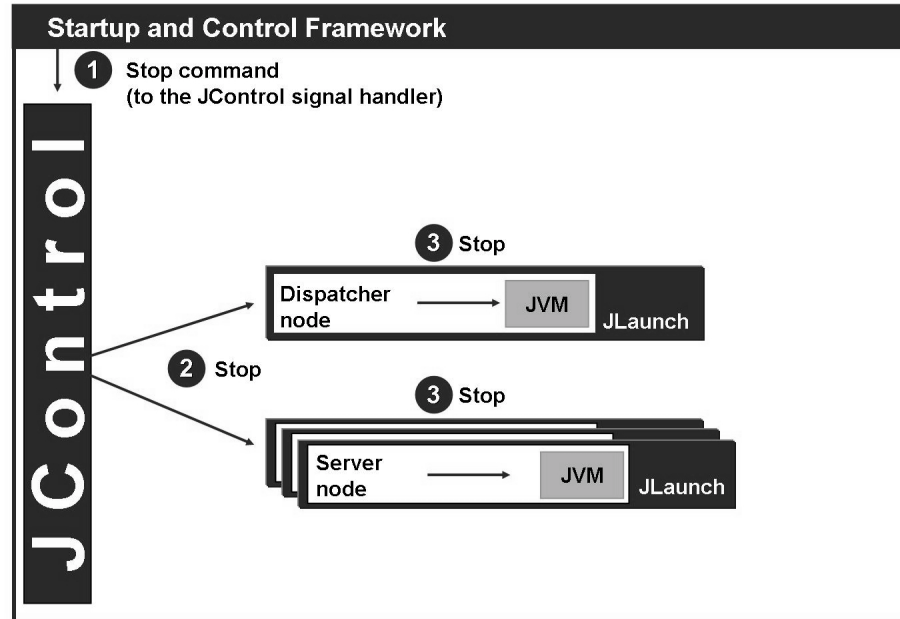


Figure 102: Stopping with the Startup and Control Framework

1. The Signal Handler of JControl receives a stop signal from the *stopsap* script (UNIX) or from the Windows Service.
2. JControl forwards the signal through named pipes to all running JLaunch processes (Java servers and dispatchers).
3. The JLaunch process of a Java cluster element should react to this notification (soft shutdown) within a certain period of time. After this time has expired, the JLaunch process is terminated by JControl. When this is done, the JVM within the JLaunch process is ended, as is the JVM of the JLaunch process.

JCmon

The JCmon tool can be used to monitor the JControl process. JCmon is part of the *Startup and Control Framework*, and is located in the JControl/JLaunch home directory, that is, in the executable directory `/usr/sap/<SID>/<instance>/SYS/exe/...`

JCmon can be started with the command `JCmon pf=<SAP instance profile>`. JCmon provides an administration interface for elements in the Java cluster that can be called from the operating system.



```

C:\Windows\System32\cmd.exe - jcmmon pf=G:\usr\sap\DEV\SYS\profile\DEV_DVEBMGS00_twdf1003
G:\usr\sap\DEV\SYS\exe\bin\NTAMD64\jcmmon pf=G:\usr\sap\DEV\SYS\profile\DEV_DVEBMGS00_twdf1003
[Ctrl+F224] [Ctrl+F2] [Ctrl+F2]

JControl Monitor Program - Main Menu
0 : exit
10 : Cluster Administration Menu
20 : Local Administration Menu
30 : Shared Memory Menu (Solid lock, experimental)
command => 10

JControl Monitor Program - Cluster Admin Menu
0 : exit
1 : Display Instance List
2 : Shutdown Cluster
3 : Shutdown Cluster (Async)
4 : Shutdown Cluster and wait for Restart
5 : Shutdown Cluster and wait for Restart (Async)
6 : Start waiting Cluster
7 : Free Cluster wait lock without Restart
8 : Shutdown Cluster and wait for Restart without SDM
9 : Shutdown Cluster and wait for Restart without SDM (Async)
10 : Shutdown Instance
11 : Shutdown Instance (Async)
12 : Shutdown Instance with Restart
13 : Shutdown Instance with Restart (Async)
14 : Restart Instance
15 : Enable processes with specified process type
16 : Disable processes with specified process type
17 : Enable processes excluding specified process type
18 : Disable processes excluding specified process type
19 : Shutdown and Restart Instance (Async)
21 : Process Administration Menu (remote)
30 : Enable all processes with specified process type
31 : Disable all processes with specified process type
32 : Enable all processes excluding specified process type
33 : Disable all processes excluding specified process type
40 : Enable bootstrapping for all processes with specified process type
41 : Enable bootstrapping for all processes excluding specified process type
50 : Shutdown and Restart Cluster (Async)
98 : Display monitor clients
99 : Display process types
command =>

C:\Windows\System32\cmd.exe - jcmmon pf=G:\usr\sap\DEV\SYS\profile\DEV_DVEBMGS00_twdf1003
command => 20

SAP System Name : DEV
SAP System : 00
MS Host : twdf1003
MS Port : 3602
Process Count : 3
PID of JControl : 7988
State of JControl : All processes running
State inside MS : All processes running
Admin URL :

!Idx!Name!PID!State!Error!Restart!
-----!-----!-----!-----!-----!-----!
0!dispatcher!5460!Running!0!yes!
1!isapver0!5492!Running!0!yes!
2!SDM!5080!Running!0!yes!

JControl Monitor Program - Administration Menu (Local)
Instance : JC_twdf1003_DEV_00
0 : exit
1 : Refresh list
2 : Shutdown instance
3 : Enable process
4 : Disable process
5 : Restart process
6 : Enable bootstrapping on restart
7 : Disable bootstrapping on restart
8 : Enable debugging
9 : Disable debugging
10 : Dump stacktrace
11 : Process list
12 : Port list
13 : Activate debug session
14 : Deactivate debug session
15 : Increment trace level
16 : Decrement trace level
17 : Enable process restart
18 : Disable process restart
19 : Restart instance
40 : Enable bootstrapping for all processes with specified process type
41 : Enable bootstrapping for all processes excluding specified process type
99 : Extended process list on/off
command =>

```

Figure 103: JComon

Exercise 13: Java Startup and Control Framework

Exercise Objectives

After completing this exercise, you will be able to:

- Identify the processes of the Startup and Control Framework at operating system level

Business Example

Starting and stopping an SAP system is a basic task for administrators of SAP systems. To understand parameter maintenance, it is important to understand how parameters are transferred to Java instances.

Task 1: Startup and Control Framework

The Startup and Control Framework consists of the JControl and JLaunch processes. How many JControl and JLaunch processes exist on your host?

1. Log on to the operating system of your host and display the process overview.

Task 2: JCmon

Start the JCmon tool and display your central instance's started Java processes.

1. Log on to the operating system as for task 1. Navigate to the profile directory and open a command prompt there (*cmd*). Enter the command *jcmon pf=<instance profile of the Java central instance>*.
2. Go to the *Local Administration Menu*. All processes of this Java instance and the status of these processes are displayed there.

Solution 13: Java Startup and Control Framework

Task 1: Startup and Control Framework

The Startup and Control Framework consists of the JControl and JLaunch processes. How many JControl and JLaunch processes exist on your host?

1. Log on to the operating system of your host and display the process overview.
 - a) Follow the instructions of your instructor to log on to the operating system. You can obtain an overview of the started processes in the Task Manager on your host. Open the Task Manager by right-clicking on the taskbar at the bottom of the screen and choosing Task Manager. Open the *Processes* tab page, and ensure that you are viewing all processes of all users. Search for the processes JControl and JLaunch. You will find a JControl process for each Java instance and a JLaunch process for each started Java node (dispatcher, server, SDM).

Task 2: JCmon

Start the JCmon tool and display your central instance's started Java processes.

1. Log on to the operating system as for task 1. Navigate to the profile directory and open a command prompt there (*cmd*). Enter the command *jcmon pf=<instance profile of the Java central instance>*.
 - a) Navigate to the directory *G:\usr\sap\<SID>\SYS\profile* and open a command prompt there using the context menu available by right-clicking.
 - b) Enter the command *jcmon pf=<instance profile of the Java central instance>*. You can find the instance profile under: *G:\usr\sap\<SID>\SYS\profile\<SID>_<instance>_<host>*.
2. Go to the *Local Administration Menu*. All processes of this Java instance and the status of these processes are displayed there.
 - a) Choose number 20 to go to the *Local Administration Menu*.



Lesson Summary

You should now be able to:

- Use the term Startup and Control Framework
- Describe the individual steps during the start and stop processes of a Java instance

Related Information

SAP Help Portal: . help.sap.com *SAP NetWeaver* → *Release '04*

Lesson: Logs of the Start and Stop Processes of SAP NetWeaver AS Java

Lesson Overview

The start process of an SAP system is a critical process. If problems occur during this phase, you should be familiar with the relevant log and trace files. This lesson focuses on the most important logs of an SAP NetWeaver AS Java.



Lesson Objectives

After completing this lesson, you will be able to:

- Find the storage locations of trace and log files of the Startup and Control Framework
- Name the most important trace and log files of the Startup and Control Framework and outline their content

Business Example

The start process of an SAP system is a critical action. If problems occur during it, the administrator must be familiar with the most important logs that are written during the start process. The administrator uses these to perform an error analysis, identify the cause, and solve the problem as quickly as possible. These files are also used for error logging during operation.

Log and Trace Files

In the case of an error or unexpected behavior of the *Startup and Control Framework*, it is important to check the following trace and log files:



- dev_jcontrol
- dev_<node name>, such as dev_dispatcher
- jvm_<node name>.out, such as jvm_dispatcher.out
- std_server<X>.out, e.g. std_server0.out
- std_dispatcher.out

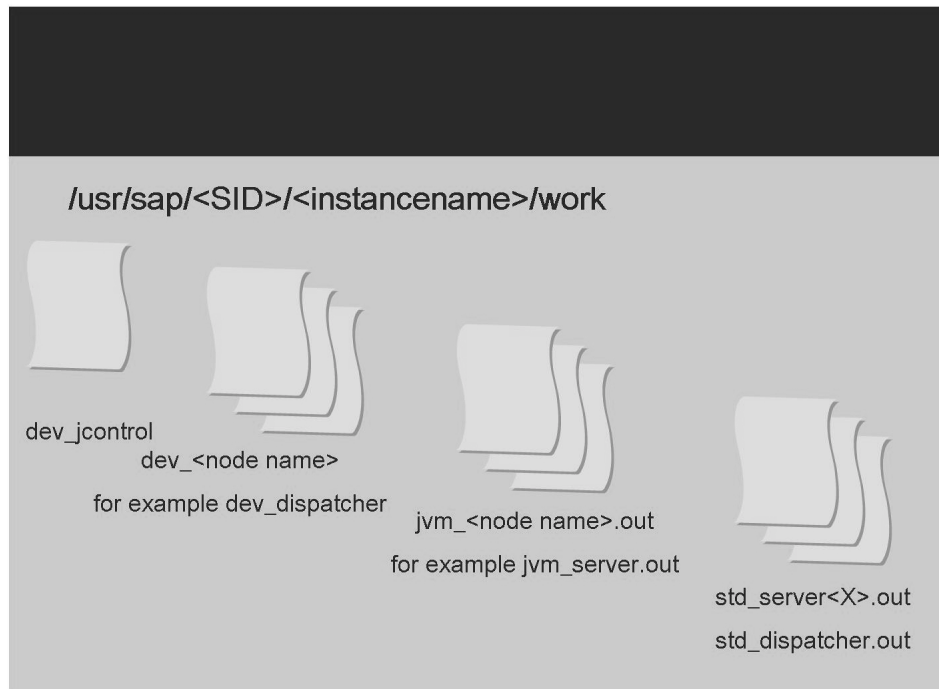


Figure 104: Log Files for Starting and Stopping

The trace and log files are stored in the *work directory* of an instance. This directory is called `/usr/sap/<SID>/<instance name>/work` (UNIX) and analogously in the Microsoft Windows environment.

dev_jcontrol is the trace file for the JControl process. `dev_jcontrol` is the most important trace file for problem messages when starting NetWeaver AS Java. Current messages are written at the end of the file.

dev_<node name> is the trace file for JLaunch processes. The trace file `dev_<node name>` is written for each started JLaunch process, and therefore for every dispatcher and server process.

jvm_<node name>.out is the output file for the Java Virtual Machine (JVM). This JLaunch process represents a Java node such as a dispatcher or a server and therefore a JVM. The output of a JVM is forwarded to the file `jvm_<node name>.out` in the work directory of a Java instance.

std_server<X>.out and **std_dispatcher.out** are the default output files for the started managers and services of the the corresponding nodes.

For all of the log files listed above, you will also find log files in the *work directory* with the ending *.old*, which can also often be used for troubleshooting.

Exercise 14: Logs of the Start and Stop Processes of SAP NetWeaver AS Java

Exercise Objectives

After completing this exercise, you will be able to:

- Name the most important logs for the start/stop processes of an SAP NetWeaver AS Java

Business Example

The start process of an SAP system is a critical action. If problems occur during it, the administrator must be familiar with the most important logs that are written during the start process. The administrator uses these to perform an error analysis, identify the cause, and solve the problem as quickly as possible.

Task: Start Logs

Check the most important logs that are written when starting an SAP NetWeaver AS Java.

1. List the most important log files that are written when starting and stopping.
2. Open the most important log files (see task 1) and perform a time-based search for errors.

Solution 14: Logs of the Start and Stop Processes of SAP NetWeaver AS Java

Task: Start Logs

Check the most important logs that are written when starting an SAP NetWeaver AS Java.

1. List the most important log files that are written when starting and stopping.
 - a) The most important logs are:
 - dev_jcontrol
 - dev_<node name>, such as dev_dispatcher
 - jvm_<node name>.out, such as jvm_server.out
2. Open the most important log files (see task 1) and perform a time-based search for errors.
 - a) Start the Terminal Server Client, enter the physical host name under *Server* and then choose *connect*. Log on to the operating system as the <sid>adm user. Open an Explorer window, and navigate to the following directory:
G:\usr\sap\<SID>\<instance>\work.
 - b) Open the relevant files and scroll to the end. Check the entries since the last start of the system.



Lesson Summary

You should now be able to:

- Find the storage locations of trace and log files of the Startup and Control Framework
- Name the most important trace and log files of the Startup and Control Framework and outline their content



Unit Summary

You should now be able to:

- Describe the process of the start procedure of an SAP system
- Start the entire SAP system or individual instances
- Evaluate start logs to analyze problems
- Stop the entire SAP System or individual instances
- Describe the starting and stopping of an SAP system under different operating systems
- Describe the locations of the log files for various databases
- Describe the sequence in which the components of an SAP system are started and stopped
- Describe the general start process for an SAP NetWeaver AS Java
- Describe the general start process for an SAP NetWeaver AS ABAP + Java
- Operate the tools to start and stop SAP NetWeaver AS ABAP + Java
- Operate the tools to start and stop SAP NetWeaver AS Java
- Use the term Startup and Control Framework
- Describe the individual steps during the start and stop processes of a Java instance
- Find the storage locations of trace and log files of the Startup and Control Framework
- Name the most important trace and log files of the Startup and Control Framework and outline their content



Test Your Knowledge

1. Which SAP processes are started when the SAP system or an instance is started?

Choose the correct answer(s).

- ☐ A SAPOSCOL
- ☐ B START_SAP_NOW
- ☐ C Message Server
- ☐ D Gateway Server
- ☐ E LAUNCH_DB

2. Log information for the dispatcher is stored in the _____ file. You can control the level of detail of the log using the _____ profile parameter. There are _____ levels for the trace. Error messages and warnings are displayed at level ____ by default.

Fill in the blanks to complete the sentence.

3. You can display the users logged on to the SAP system with transactions _____ (for each instance) and _____ (system-wide). You can obtain an overview of the scheduled background jobs with transaction _____. You can use transaction _____ to send a system message to the users that are logged on.

Fill in the blanks to complete the sentence.

4. When starting an SAP system in which only Java instances are installed, the database is started only after the Java instances have been started.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

5. In the case of SAP NetWeaver AS Java, it is not possible to stop individual instances under a UNIX operating system.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

6. The *Startup and Control Framework* consists of the processes _____ and _____.

Fill in the blanks to complete the sentence.

7. The most important trace and log files are stored in the work directory of each instance, that is, for example, under /usr/sap/<SID>/DVEBMGS00/work.

Determine whether this statement is true or false.

- ☐ True
- ☐ False



Answers

1. Which SAP processes are started when the SAP system or an instance is started?

Answer: A, C, D

SAPOSCOL is an underlying process that should always be running, even if the SAP system is shut down. START_SAP_NOW and LAUNCH_DB are made up. The message server is started once for each SAP system, and the gateway server is started once for each instance.

2. Log information for the dispatcher is stored in the dev_disp file. You can control the level of detail of the log using the rdisp/TRACE profile parameter. There are four levels for the trace. Error messages and warnings are displayed at level 1 by default.

Answer: dev_disp, rdisp/TRACE, four, 1

3. You can display the users logged on to the SAP system with transactions SM04 (for each instance) and AL08 (system-wide). You can obtain an overview of the scheduled background jobs with transaction SM37. You can use transaction SM02 to send a system message to the users that are logged on.

Answer: SM04, AL08, SM37, SM02

Transaction SM04 is instance-specific, transaction AL08, on the other hand, is cross-instance. In transaction SM37, you can select by client and user, among other things. In transaction SM02, you can restrict the sending of the message to instances or clients.

4. When starting an SAP system in which only Java instances are installed, the database is started only after the Java instances have been started.

Answer: False

The database is always started first, or must be available before instances are started.

5. In the case of SAP NetWeaver AS Java, it is not possible to stop individual instances under a UNIX operating system.

Answer: False

You can stop individual instances using the command stopsap R3 <instance name> or stopsap J2EE <instance name>.

6. The *Startup and Control Framework* consists of the processes JControl and JLaunch.

Answer: JControl, JLaunch

JControl and JLaunch are the processes that are visible when starting an instance.

7. The most important trace and log files are stored in the work directory of each instance, that is, for example, under /usr/sap/<SID>/DVEBMGS00/work.

Answer: True

All developer traces and all important start files are stored in the work directory of each instance.

Unit 5

Introduction to System Configuration

Unit Overview

In this unit, you will learn, among other things, how you can use profile parameters to configure an SAP system.

Furthermore, the configuration and administration tools of SAP NetWeaver AS Java are introduced and their application areas are discussed. Selected settings for Java instances and the Central Services are presented.



Note: The following lessons have been formed from two different courses. The advantage to this is that all of the knowledge required for this range of topics is introduced, within the context of learning, in a training course for consultants. Consequently, you can use synergies to facilitate understanding among participants, and for actual use. This would not be possible if these topics were discussed at a later time.



Unit Objectives

After completing this unit, you will be able to:

- Determine the configuration of system parameters
- Adjust the system parameters using profiles
- Set up a dynamic switch of work process types using operation modes
- Name various configuration and administration tools for the SAP NetWeaver AS Java
- Describe the primary usage areas of configuration and administration tools
- Maintain settings for an SAP NetWeaver AS Java with the Config Tool
- Maintain settings for an SAP NetWeaver AS Java with the Visual Administrator
- Obtain an overview of the most important information for an SAP NetWeaver AS Java with the System Information and SAP NetWeaver Administrator or Telnet Console
- Make selected configurations

Unit Contents

Lesson: How the System Evaluates Its Parameters	257
Exercise 15: Configuration of System Parameters	263
Lesson: How to Set System Parameters	266
Exercise 16: Maintaining the System Parameters	271
Lesson: Setting up Operation Modes	277
Exercise 17: Setting up Operation Modes	285
Lesson: Administration and Configuration Tools of SAP NetWeaver AS Java	289
Exercise 18: Administration and Configuration Tools of SAP NetWeaver AS Java	305
Lesson: General Configuration of the SAP NetWeaver AS Java Cluster with the Config Tool	311
Exercise 19: General configuration of SAP NetWeaver AS Java with the Config Tool	321
Lesson: General Configuration of the SAP NetWeaver AS Java Cluster with the Visual Administrator	328
Exercise 20: General Configuration of the SAP NetWeaver AS Java Cluster with the Visual Administrator	341
Lesson: Other Administration Tools	347
Exercise 21: Other Administration Tools	359
Lesson: Selected Configurations	362
Exercise 22: Selected Configuration	371

Lesson: How the System Evaluates Its Parameters

Lesson Overview

This lesson explains the order in which the system evaluates profile parameters and where these parameters are stored.



Lesson Objectives

After completing this lesson, you will be able to:

- Determine the configuration of system parameters

Business Example

You want to determine the system parameters for your SAP system.

Configuring the System Parameters

The configuration of the individual instances and therefore of the SAP system is performed using system parameters. The default values for these parameters are defined in the program code of the kernel.



Instance profile

<SID>_<Instance>_<Host name>

Default profile

DEFAULT.PFL

- SAP kernel
- Environment variable

Sequence
in which the values for
the profile parameters
are read

Figure 105: Assigning System Parameters

You can change these default values using the profile files, which are read when an instance is started. These profile files are created during the installation of the system and can also be edited later.

As the profile files are only read when the system is started, you must restart the instance or the entire system after changing individual parameters.

Dynamic switching, that is, while the system is running, is only possible for a small number of system parameters.

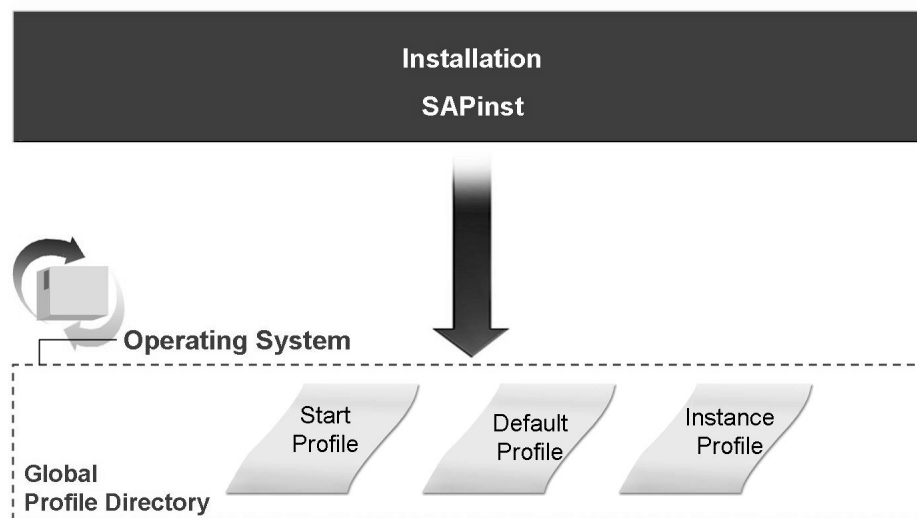


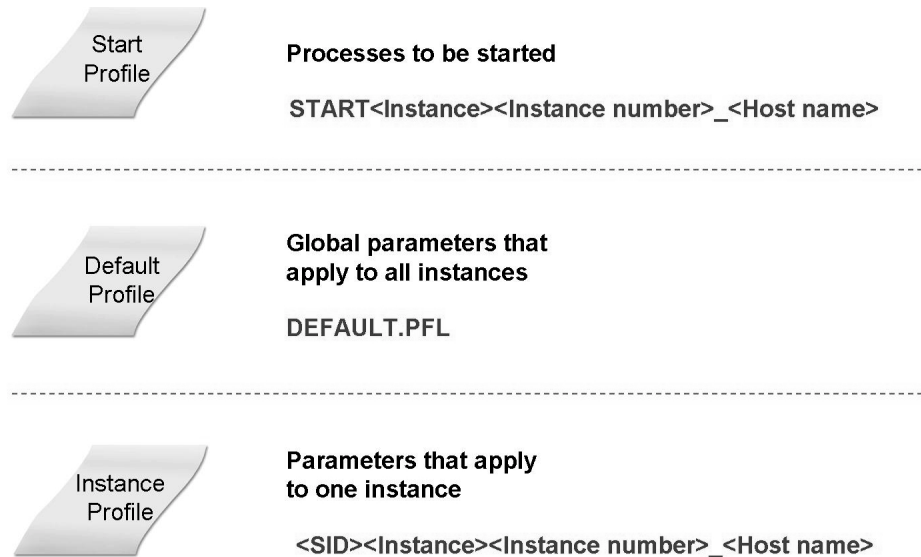
Figure 106: Profile Files at Operating System Level

The profile files are automatically created during installation. After installation is complete, the profile files are stored at operating system level in the directory: `\usr\sap\<SID>\SYS\profile`. This directory can be read by all instances of an SAP system using the share or mount technique.

The SAP system has three system profiles. These are:

- Start profile
- Default profile
- Instance profile

In principle, you can change these files with operating system tools (editors). In the process, users must themselves ensure that the changes are performed correctly. Parameters that are set incorrectly can lead to the system not starting. It is far more convenient and safer to change the profile parameters using the tools in the SAP system.



For example: <SID> = <DEV>, <Instance> = <DVEBMGS00>, <Instance number> = <00>, <Host name> = <twdf1000>

Figure 107: Profile Files: Overview

The instance-specific start profile (*START<instance><instance number>_<host name>*), specifies which processes are to be started for each instance. These are, for example, the message server and the dispatcher.

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

The instance profile (*<SID>_<instance><instance number>_<host name>*) defines parameters that apply for one instance, such as the number and type of work processes, or the definition of the size and allocation of the main memory area used by the SAP system. The instance profile is therefore instance-specific.

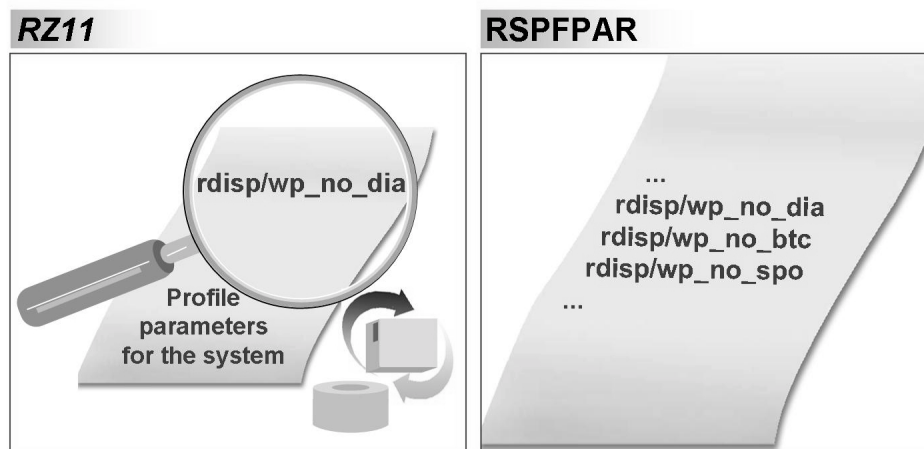


Figure 108: View of System Parameters

The current values of the system parameters can be displayed in the system. There are two ways to do this: Report RSPFPAR and transaction RZ11. Both functions display the system parameters for the instance to which the user is currently logged on. Report RSPFPAR displays a list of all instance-specific system parameters, along with the parameters that apply system-wide. You can restrict this list to specific parameters. A table displays the system default value for the individual parameters as it is defined in the program code of the kernel and, if the default value has been overridden by a profile parameter, the user-defined value. A short description and, if required, documentation for the parameters can also be displayed. Transaction RZ11 displays information and documentation for individual profile parameters. It also shows whether the parameter can be changed while the system is running with the *DynamicallySwitchable* indicator.

➔ **Note:** In the *TPFYPROPTY* table, all dynamically switchable profile parameters are identified with the *Dynamic* indicator. You can use transaction SE16, for example, to display this table.

Outside the SAP system, you can display the values of the profile parameters at operating system level using the user <sid>adm with the program **sappfpar**. You can display the current value of a parameter with **sappfpar <parameter name>**. The command **sappfpar all** returns a list of all parameters. You can check the parameters that are set using **sappfpar check**. The command **sappfpar help** returns a short overview of possible options.

You can also specify an instance profile, an instance number, or the SAP system name with this command using the options **pf=<instance profile>**, **nr=<instance number>**, or **name=<SID>**.



Hint: Note that, for the evaluation of profile parameters using the tools described above, some profile parameters are the same system-wide, while others can be set differently for each instance. The report RSPFPAR displays the configuration of the instance on which you run the report.

Exercise 15: Configuration of System Parameters

Exercise Objectives

After completing this exercise, you will be able to:

- Determine the configuration of system parameters

Business Example

You want to determine the system parameters for your SAP system.

Task: Determine Values of System Parameters

Determine the values of a number of system parameters.

1. Determine the values of the following system parameters:

Name of the application server, number of work processes on the central instance (dialog, batch, enqueue, update, spool), maximum runtime of a program in the dialog.

Solution 15: Configuration of System Parameters

Task: Determine Values of System Parameters

Determine the values of a number of system parameters.

1. Determine the values of the following system parameters:

Name of the application server, number of work processes on the central instance (dialog, batch, enqueue, update, spool), maximum runtime of a program in the dialog.

- a) You can determine the values of system parameters by running the report RSPFPAR. The parameter names are as follows:

Name of the local application server: *rdisp/myname*.

Number of work processes on the central instance:

Dialog: *rdisp/wp_no_dia*

Batch: *rdisp/wp_no_btc*

Enqueue: *rdisp/wp_no_enq*

Update: *rdisp/wp_no_vb*

Spool: *rdisp/wp_no_spo*

This parameter limits the maximum runtime of a process step within a dialog process: *rdisp/max_wprun_time*.



Lesson Summary

You should now be able to:

- Determine the configuration of system parameters

Lesson: How to Set System Parameters

Lesson Overview

In this lesson, you will learn how you can change the values of system parameters.



Lesson Objectives

After completing this lesson, you will be able to:

- Adjust the system parameters using profiles

Business Example

As an administrator, you are to adjust the configuration of the system parameters.

Administering and Maintaining Profiles

If you want to change profile parameters, you can make these changes using operating system-specific editors. However, this procedure has certain dangers, as the user must ensure that the changes are performed, and also documented, correctly. Incorrectly set parameters can prevent an instance starting. The SAP system therefore provides integrated profile administration and parameter maintenance.

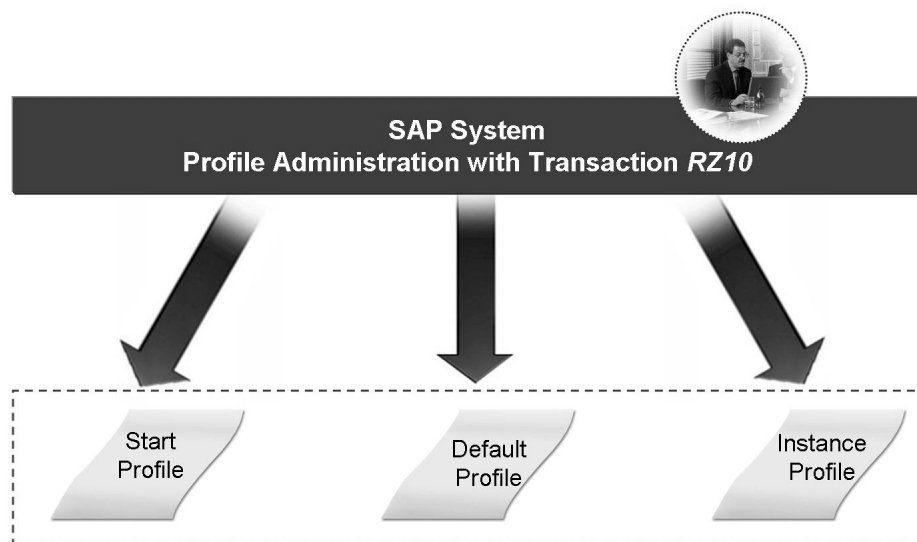


Figure 109: Profile Administration

Profile administration in the SAP system offers a number of advantages to the user.

Advantages of Profile Administration in the SAP System

- Central administration and maintenance of the instances
- Changes in profiles checked for consistency
- Administration of multiple versions of a profile
- Comparison of the actively used profile and the profile stored in the database
- Immediate activation of selected parameters



Caution: Before you make changes to the profile files, create backups. If an error should occur, you can then start the SAP system with the restored files and perform troubleshooting.

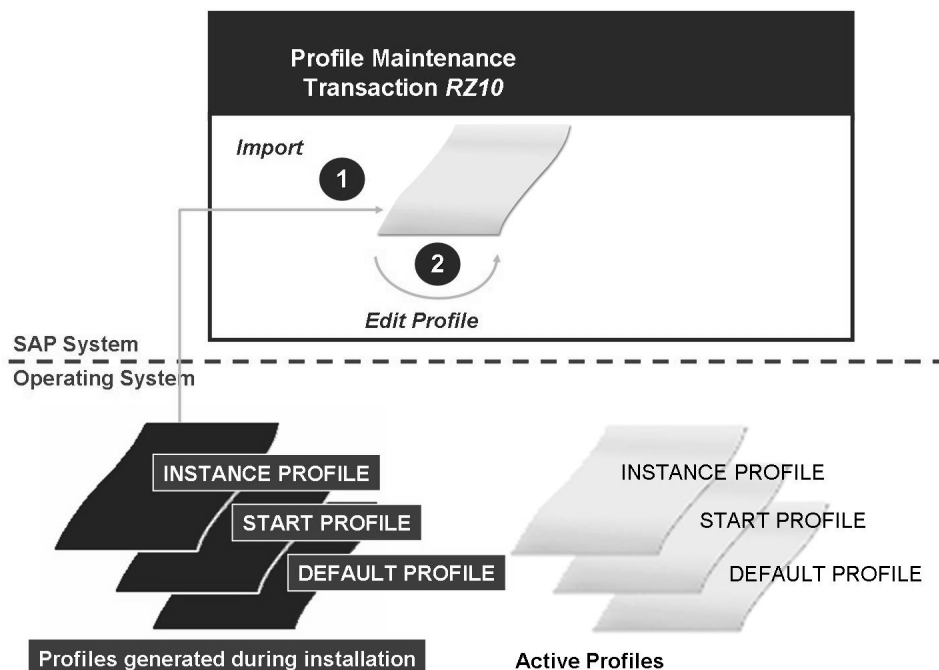


Figure 110: Profile Maintenance 1/2

After installation, the profile parameters are, at first, only present at operating system level. To use the profile administration of the SAP system, the profiles must be imported into the database. During this import, the system performs a consistency check and a check of the way in which the parameters interact. Changes to profile parameters can then be performed in the SAP system, with the previously mentioned advantages. These changes are then stored in the database, and written back to file level. The changes only take effect when they are read by the system; that is, at a restart of the system.

You perform the administration and maintenance of profiles in transaction RZ10. In the first step, you import the profiles into the database by choosing *Utilities* → *Import Profiles* → *Of active servers*. After selecting the profile to edit, you can change individual profile parameters. There are three different levels for maintaining the profiles. The **administration data** contains the type of profile (start, default, or instance profile), a short description, the path of the file, the name of the instance, and the time of the last activation.

You can perform the maintenance of the parameters of the individual profiles through either **basic maintenance** or **extended maintenance**. Basic maintenance allows you to adjust the most important parameters and supports the user through the use of logical descriptions. Extended maintenance displays the unformatted content of the profile; that is, the technical names of the profile parameters. In extended maintenance, you can not only change the values of individual parameters, but also add new parameters and delete existing parameters.

The changes are stored in two steps. In the first step, the changes are temporarily copied when you choose *Copy*. The values are permanently saved to the database in a second step when you choose *Save*. The changes are therefore successfully saved to the database and must now be written at operating system level. This is done either automatically during saving by confirming the relevant query, or manually from the menu by choosing *Profile* → *Activate*.

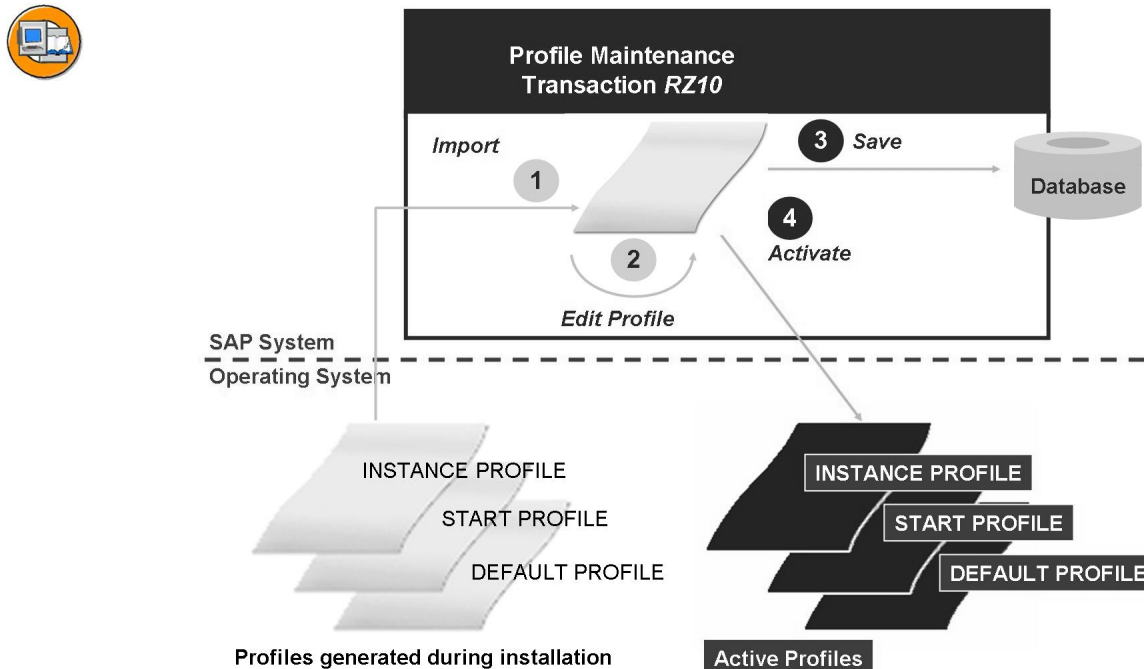


Figure 111: Profile Maintenance 2/2

Changes to instance-specific profiles take effect after a restart of the corresponding instance. Changes to the default profile take effect only after a restart of all instances in the entire system.

Changes to these files are, however, the exception rather than the rule.

Consistency Check

Additional functions available in transaction RZ10 are the consistency check and the comparison of profiles.

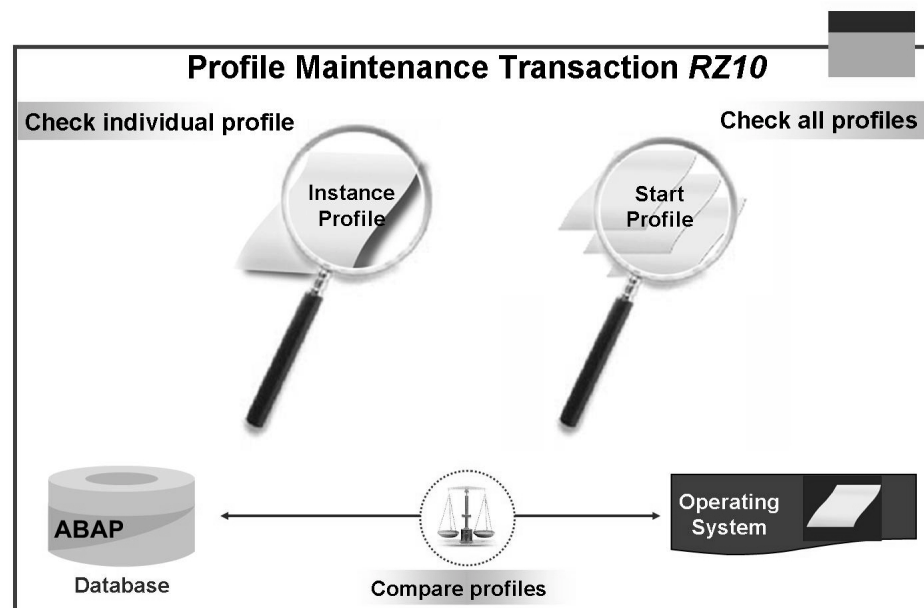


Figure 112: Profile Consistency Checks

During a consistency check, the system checks the syntax and the semantics for individual or for all profiles.

The *Utilities* → *Check all profiles* option also contains a consistency check between the profiles of a type. This means that, for example, the system checks for the start profile whether only one message server is being started for each system.

During the comparison of profiles, the system compares the profile that is in active use with the profile stored in the database. This comparison is automatically performed when an instance is started. If the system identifies a difference between the active profile and the profile stored in the database, the system displays a message in the Alert Monitor.

Exercise 16: Maintaining the System Parameters

Exercise Objectives

After completing this exercise, you will be able to:

- Adjust the system parameters using profiles

Business Example

As an administrator, you are to adjust the configuration of the system parameters.

Task 1: Save Profile Files

First back up the profile files.

1. Save your “runnable” profile files in a backup directory. To do this, create a *Backup<SID>* directory under *C:\Temp* and copy all the profile files to this directory. Replace *<SID>* with the system ID of your SAP system.

Task 2: Import and Maintain Profiles

Use transaction RZ10 (*Tools* → *CCMS* → *Configuration* → *System Profile*) to perform the following tasks.

1. Import all the profiles into your SAP system.
2. Change the values of the following parameters:

Increase the number of dialog work processes for the dialog instance by 2.

Set the value for the maximum runtime of a process step to 750 seconds. To do this, you may need to add a new parameter to a profile.

Consider the profiles in which you are making changes. Save and activate your changes.
3. Check the number of work processes in the process overview and the maximum runtime for a process step.
4. Check your changes in the profile files at operating system level.

What do you still need to do in order that your changes take effect in the system?

Continued on next page

Task 3: Activating the Changed Profile Parameters

Check the changed system parameters after a system restart.

1. Restart the system so that your changes are activated.
2. Check your changes in the system again.

Solution 16: Maintaining the System Parameters

Task 1: Save Profile Files

First back up the profile files.

1. Save your “runnable” profile files in a backup directory. To do this, create a *Backup<SID>* directory under *C:\Temp* and copy all the profile files to this directory. Replace *<SID>* with the system ID of your SAP system.
 - a) Log on to your server. Copy the profile files from the *\usr\sap\<SID>\SYS\profile* directory to the *Backup<SID>* directory.

Task 2: Import and Maintain Profiles

Use transaction RZ10 (*Tools* → *CCMS* → *Configuration* → *System Profile*) to perform the following tasks.

1. Import all the profiles into your SAP system.
 - a) You can import all the profile files together by choosing: *Utilities* → *Imported Profiles* → *Of active servers*. A check log displays the result of the import.
2. Change the values of the following parameters:

Increase the number of dialog work processes for the dialog instance by 2.

Set the value for the maximum runtime of a process step to 750 seconds. To do this, you may need to add a new parameter to a profile.

Continued on next page

Consider the profiles in which you are making changes. Save and activate your changes.

- a) To change the values of individual parameters, select the corresponding profile (default or instance profile) and change this in *Extended Maintenance*.

If no *Change* pushbutton is displayed, choose *Display <-> Change* to switch to change mode. Select the desired parameter and choose *Change* or double click the relevant parameter.

After you have made your changes, choose *Copy* and then *Save*. In the list view, choose *Copy* and then *Back*. *Save* the new version of the profile and confirm the activation of the profile in the dialog box.

To create a new parameter, go to *Extended Maintenance* and choose *Create*.

The required parameters are listed below:

- *rdisp/wp_no_dia* (instance profile)
- *rdisp/max_wprun_time* (default profile)

3. Check the number of work processes in the process overview and the maximum runtime for a process step.
 - a) Check the number of processes by calling the server overview (*Tools → Administration → Monitor → System Monitoring → Servers*, transaction SM51), selecting the dialog instance, and then choosing *Processes*.
 - b) Check the maximum runtime of a process step by displaying the parameter with the transaction RZ11. In the Param. Name field, enter *rdisp/max_wprun_time* and choose the *Display* pushbutton. The *Current Value* field should still display the previous value **600**.
4. Check your changes in the profile files at operating system level.

What do you still need to do in order that your changes take effect in the system?

- a) Open the profile files at operating system level with an editor and use this to view the changes that have been made.
- b) Your changes to profile parameters will only take effect after a restart of the affected instance or the entire system.

Continued on next page

Task 3: Activating the Changed Profile Parameters

Check the changed system parameters after a system restart.

1. Restart the system so that your changes are activated.
 - a) Log on to your server in the same way as you did in the exercises for the Starting and Stopping unit, and restart the system.
2. Check your changes in the system again.
 - a) Check the number of processes by calling the server overview (*Tools → Administration → Monitor → System Monitoring → Servers*, transaction SM51), selecting the dialog instance, and then choosing *Processes*.
 - b) Check the maximum runtime of a process step by displaying the parameter with the transaction RZ11. In the Param. Name field, enter *rdisp/max_wprun_time* and choose the *Display* pushbutton. The *Current Value* field should now display the value **750** that you entered.



Lesson Summary

You should now be able to:

- Adjust the system parameters using profiles

Lesson: Setting up Operation Modes

Lesson Overview

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



Lesson Objectives

After completing this lesson, you will be able to:

- Set up a dynamic switch of work process types using operation modes

Business Example

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

Concept of Operation Modes

The demands of users on the SAP system vary during the course of the day. During the day, a large number of dialog users, who want to be served with high performance by the system, are working. Therefore, a large number of dialog work processes should be available for users during the day. During the night, however, only a small number of these dialog work processes are used, and the system can be used to a larger extent to process background jobs.

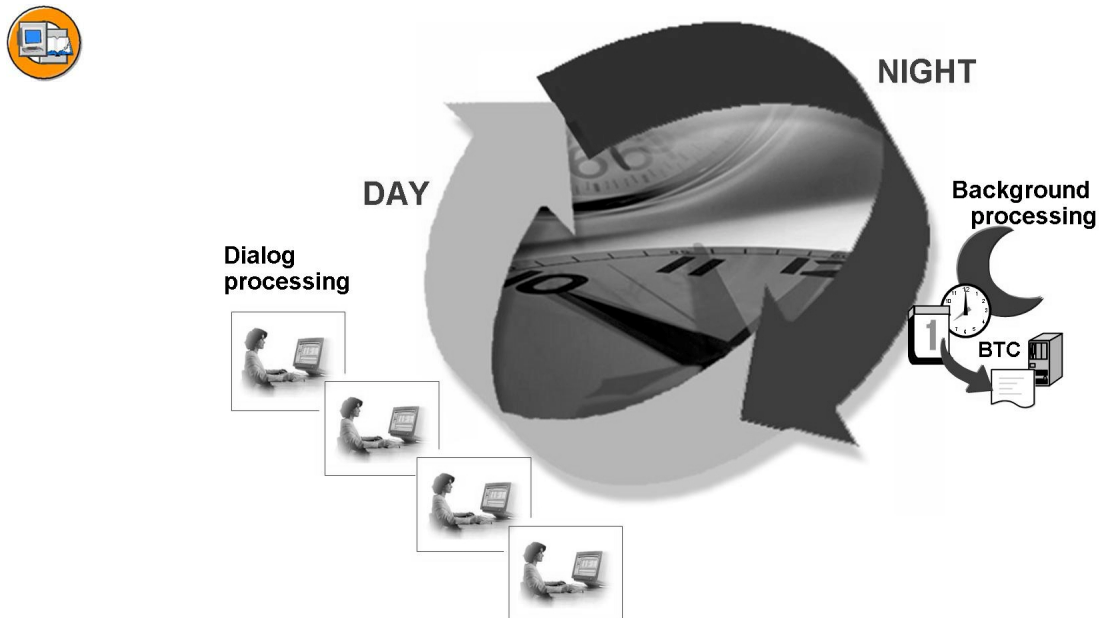


Figure 113: Concept of Operation Modes

The type and number of work processes for each instance is defined in the profiles. The distribution of work processes in the profiles is optimized for fast dialog response times; that is, there are usually lots of dialog work processes and a small number of background work processes. This means that during the night, system resources, such as the main memory, are tied to the dialog work processes, or are not fully utilized by the background processes, such as the CPU. It is therefore practical to define different types and numbers of work processes for these different demands on the SAP system. This is realized through the concept of operation modes.

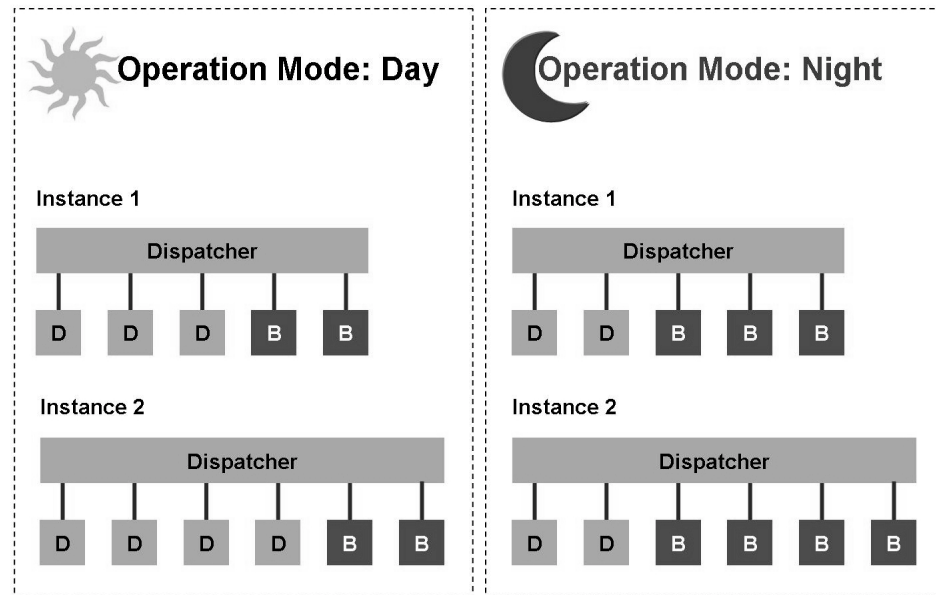


Figure 114: Adjusting Instances to the Load Distribution

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

During the switch of the operation modes, neither the instance nor the affected work processes needs to be restarted. This means that the quality of the buffer of the SAP system is retained during an operation mode switch, and that the request that is currently being processed by a work process is completed. The individual work processes retain their process ID after the switch. You can observe this in the process overview (SM50).

Setting Up Operation Modes

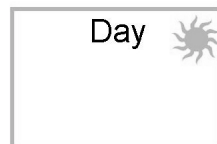
The operation modes are set up in a number of steps.

Steps to Configure Operation Modes

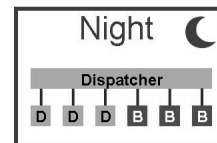
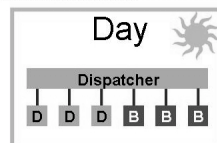
- First, the operation modes are created as empty containers in transaction RZ04.
- Next, all active instances of the system are detected and the work processes defined in the instance profile are assigned to the operation modes as default values.
- You can now make allocations for the individual operation modes in the total number of work processes taken from the instance profile. The allocation should be made primarily between the dialog and background work processes.
- You then specify the periods for which the operation modes are valid and when the switch between the operation modes should occur in the time table (SM63).



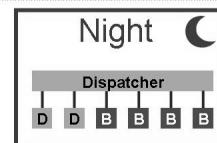
Create operation mode:



Set up distribution of the work processes using instance definition:



Adjust distribution of the work processes:



Define time plan:

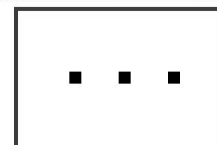
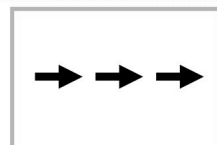


Figure 115: Setting Up Operation Modes

You cannot arbitrarily change the work process distribution.

Work Process Type	Type of Changeability (+ = increase number, - = reduce number)
Dialog	only indirectly changeable
Background	+ : possible, reduces the number of D-WPs accordingly - : possible, increases the number of D-WPs accordingly

Class A	determines the subset of the B-WPs that are only to process class A background jobs
Update	+: possible, if at least 1 V-WP exists, reduces the number of D-WP accordingly -: possible, if at least 1 V-WP remains, increases the number of D-WPs accordingly
V2 Update	+: possible, if at least 1 V2-WP exists, reduces the number of D-WPs accordingly -: possible, if at least 1 V2-WP remains, increases the number of D-WPs accordingly
Enqueue	+: possible, if at least 1 V2-WP exists, reduces the number of D-WPs accordingly -: possible, if at least 1 E-WP remains, increases the number of D-WPs accordingly You should only increase the number of E-WPs if this procedure is suggested by SAP Support.
Spool	cannot be changed



Note: Note the following: A change in the allocation should be made primarily between the dialog and background work processes.

The time table (SM63) differentiates between normal operation and exception operation. A general time definition that applies almost all of the time is set using normal operation. In rare cases, in which a special distribution of the work processes for unusual system loads is required, they are defined using exception operation.

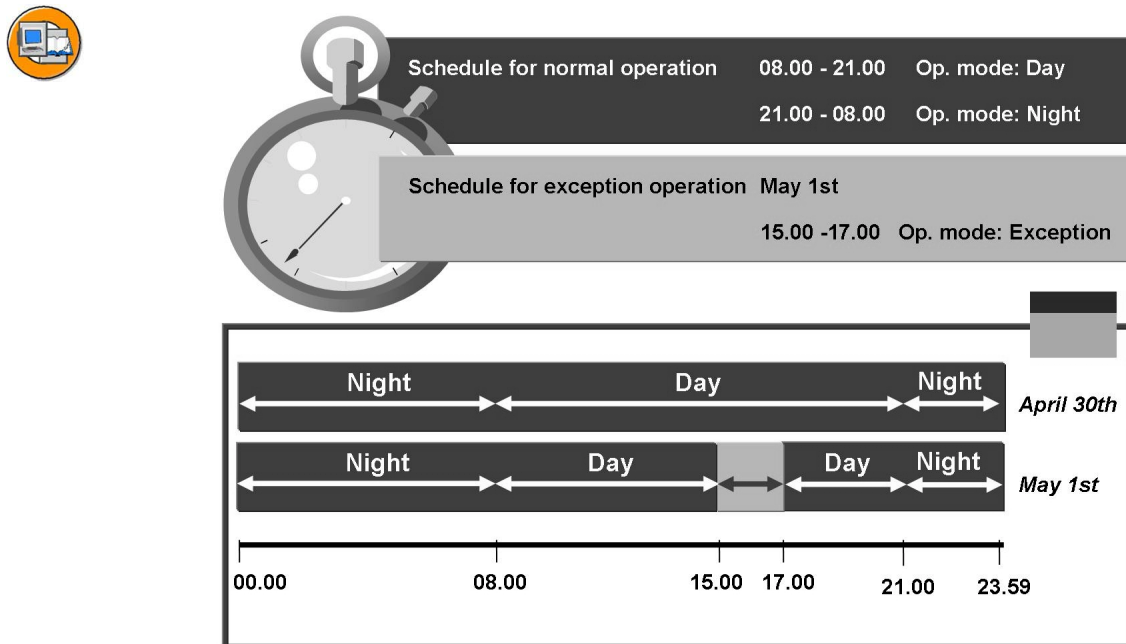


Figure 116: Scheduling Operation Modes

If you do not define a timetable for normal operation, no switch of operation modes is made. The configuration in the instance profile remains active.

The exception operation can only be defined as a unique event.

You can trigger an operation mode switch from a program that you have written using a function module (RZL_PERFORM_BA_SWITCH).

Monitoring and Consistency Check

The Control Panel (RZ03) allows you to monitor the instances and the operation modes and provides functions to:

- Check the status of all instances and of the operation modes
- Start and stop the instances
- Manually switch operation mode
- Display an overview of the work processes
- Switch to the Alert Monitor



Figure 117: Control Panel (RZ03)

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

You can first simulate the switch of operation modes (*Control* → *Switch operation mode* → *Simulation*). The system checks for which instances a switch can be performed.

You can display a detailed analysis of the status of the individual instances by choosing *Monitoring* → *Status Details*.

If it is not possible to switch between operation modes, this is usually due to inconsistencies in the SAP system. These inconsistencies can occur if the number of work processes is defined differently in different places in the system. These are the instance profiles at operating system level, the instance profile in the database, and the definition of the operation modes themselves.

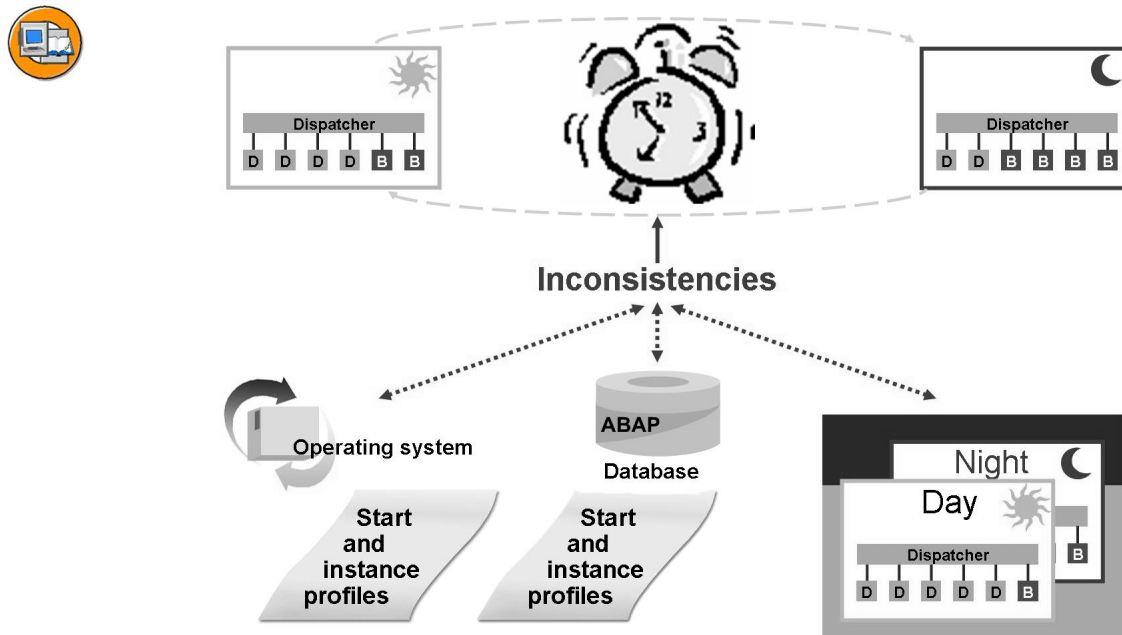


Figure 118: Profile Consistency Checks

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

Exercise 17: Setting up Operation Modes

Exercise Objectives

After completing this exercise, you will be able to:

- Set up a dynamic switch of work process types using operation modes

Business Example

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

Task 1: Create Operation Modes

Create operation modes using transaction RZ04 to perform the following steps.

1. Create two operation modes, *Day* and *Night*.
2. Now create a definition for the work process distribution for all instances of your system.
3. Change the distribution of the work processes of your instances for each of the operation modes.

Note that there should be at least two dialog and two background work processes in all operation modes.

Task 2: Schedule Operation Modes

Schedule the operation modes that you have created.

1. In the timetable, schedule the operation modes that you have created for normal operation.

Task 3: Manual Switch of Operation Modes

1. To observe the change in work process distribution when the operation mode is switched, switch to the *Night* operation mode manually in the Control Panel.
2. How can you observe the change of the work process distribution?

Solution 17: Setting up Operation Modes

Task 1: Create Operation Modes

Create operation modes using transaction RZ04 to perform the following steps.

1. Create two operation modes, *Day* and *Night*.
 - a) You maintain operation modes in transaction RZ04 (*Tools* → *CCMS* → *Configuration* → *Operation Modes/Instances*).

You can create operation modes by choosing *Create operation mode*. Enter the name of the operation mode and a short description and choose *Save*.
2. Now create a definition for the work process distribution for all instances of your system.
 - a) Switch to the *Instances/Operation Modes* view by choosing *Instances/Operation Modes*.

You define work process distribution for all instances of your system by choosing *Settings* → *Based on current status* → *New instances* → *Generate*. Choose *Save*.

3. Change the distribution of the work processes of your instances for each of the operation modes.

Note that there should be at least two dialog and two background work processes in all operation modes.

- a) To change the distribution of the work processes for the operation modes of the instances, double click the operation modes entries. You change and then save the distribution of the work processes in the dialog box: *CCMS: Maintain Work Process Distribution*.

To do this, place the cursor in the column of the work process type that you want to change. You can change the number by choosing + and -. Then choose *Save*.

After you have changed the distribution for all operation modes and instances, choose *Save*.

Continued on next page

Task 2: Schedule Operation Modes

Schedule the operation modes that you have created.

1. In the timetable, schedule the operation modes that you have created for normal operation.

- a) You can schedule operation modes using the operation mode calendar (*Tools → CCMS → Configuration → Operation Mode Calendar*, transaction SM63).

Select *Normal Operation* and then choose *Change*.

Double click to select the start and end of the period for operation mode *Day*. Choose the *Assign* button and use the F4 help to select the operation mode *Day*. Follow the same procedure to assign the operation mode *Night*. After you have assigned the entire 24-hour period to your operation modes, choose *Save*.

Task 3: Manual Switch of Operation Modes

1. To observe the change in work process distribution when the operation mode is switched, switch to the *Night* operation mode manually in the Control Panel.

- a) You can switch the operation modes manually in the Control Panel (*Tools → CCMS → Control/Monitoring → Control Panel*, transaction RZ03).

As a first step, select the operation mode where you want to switch to, by choosing *Choose operation mode*. Select the operation mode and then *Choose*.

The switch is performed in a second step, by choosing *Control → Switch Operation Mode → All Servers*.

2. How can you observe the change of the work process distribution?

- a) Check the changed distribution of the work processes in the process overview (*Tools → Administration → Monitor → System Monitoring → Process Overview*, transaction SM50).



Lesson Summary

You should now be able to:

- Set up a dynamic switch of work process types using operation modes

Lesson: Administration and Configuration Tools of SAP NetWeaver AS Java

Lesson Overview

This lesson provides an overview of the different configuration and administration tools for SAP NetWeaver AS Java. During the course of this lesson, you learn more about the most important of these tools and their primary usage areas.



Lesson Objectives

After completing this lesson, you will be able to:

- Name various configuration and administration tools for the SAP NetWeaver AS Java
- Describe the primary usage areas of configuration and administration tools

Business Example

After an SAP NetWeaver AS Java has been installed, you need to configure the cluster of the SAP NetWeaver AS Java. You do this using the various administration tools of SAP NetWeaver AS Java. This lesson provides an overview of these various administration tools.

Usage Areas of the Tools

This section provides you with an overview of the various tools and their usage areas. Some tools are particularly suitable for a usage area, or are the only tool that can be used for a usage area. Several tools are suitable for other usage areas, on the other hand. This is explained in more detail in this section. The following tools are available for the administration and configuration of SAP NetWeaver AS Java:

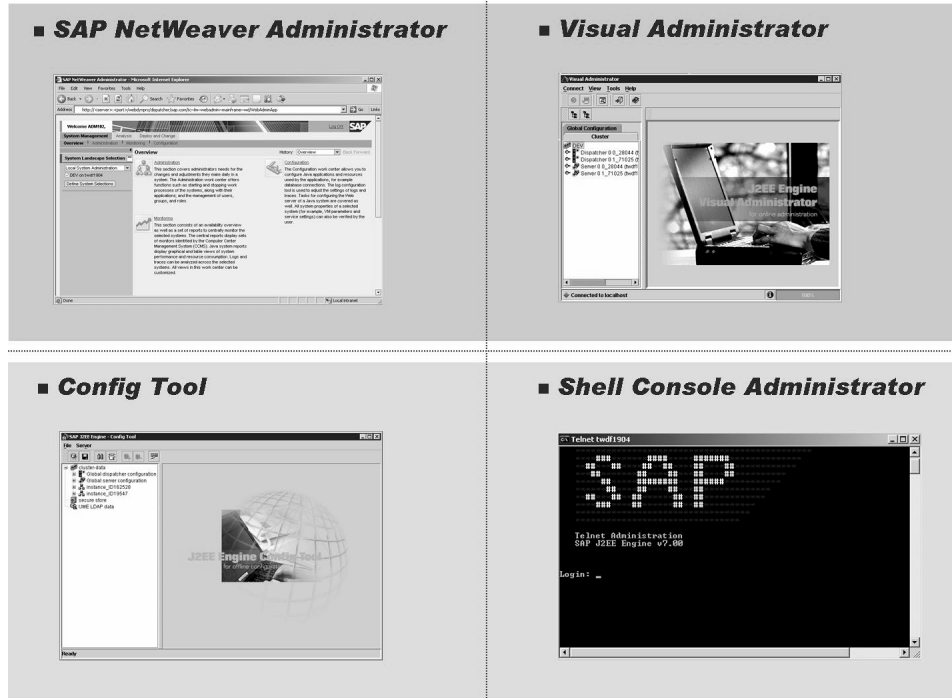


Figure 119: Tool Overview

Config Tool

You use the Config Tool to maintain the settings for SAP NetWeaver AS Java in the database. It is therefore necessary that the database is started, so that you can change the settings with the Config Tool.

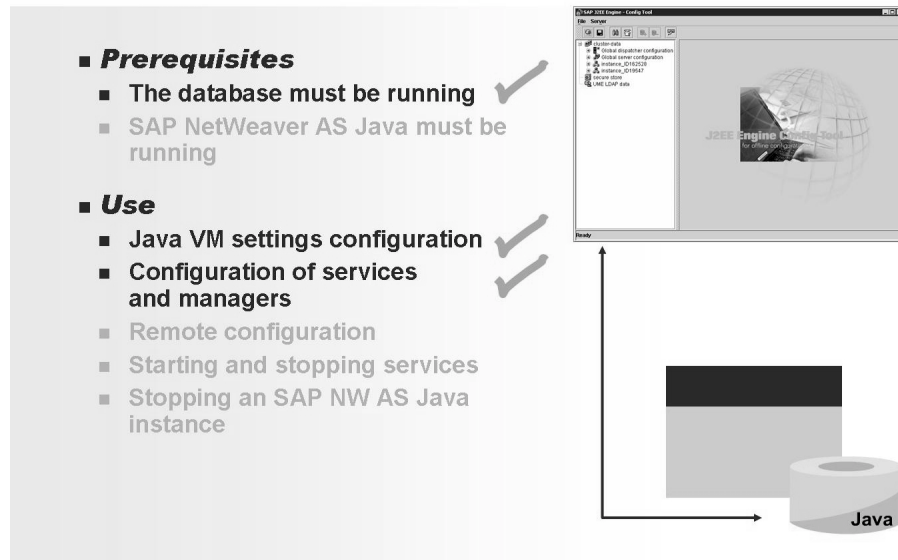


Figure 120: Config Tool 1

The **Java VM Settings** (the settings of the Java Virtual Machine) of SAP NetWeaver AS Java can **only be maintained with the Config Tool**. You can use the Config Tool to configure the managers and services of SAP NetWeaver AS Java. For information about this, see the figure “Config Tool 1”. The figure “Config Tool 2” shows the special features of the tool.

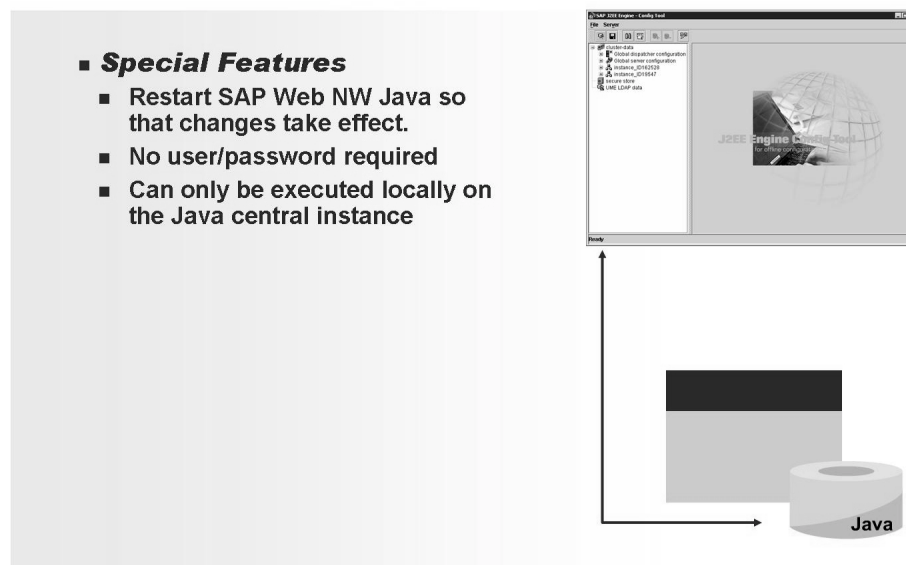


Figure 121: Config Tool 2

The settings that you make with the Config Tool only take effect when the SAP NetWeaver AS Java is started; that is, you must restart the SAP NetWeaver AS Java after maintaining settings.

The Config Tool is called on the host on which the central instance of the SAP NetWeaver AS Java is running. Authentication is performed using the logon to the operating system of the host.

SAP NetWeaver Administrator

The SAP NetWeaver Administrator (NWA) combines the most important administration and monitoring tools for Java and ABAP systems in a new, browser-based user interface. The most important benefits offered by NWA are:

- You no longer need to switch between different tools for the administration, troubleshooting and problem analysis of your entire SAP NetWeaver system landscape.
- Instead, you have for Java, and increasingly for ABAP systems, a central, cross-landscape administration tool which enables you to start and stop instances, check configuration settings and logs and monitor the correct functioning of the components.
- NWA displays all the monitoring infrastructure data, i.e. alerts together with the current and historical values.
- The GUI adheres to the current guidelines concerning interface design, is easy to use, task-oriented and complete. If you use Web Dynpro, it runs in the browser.

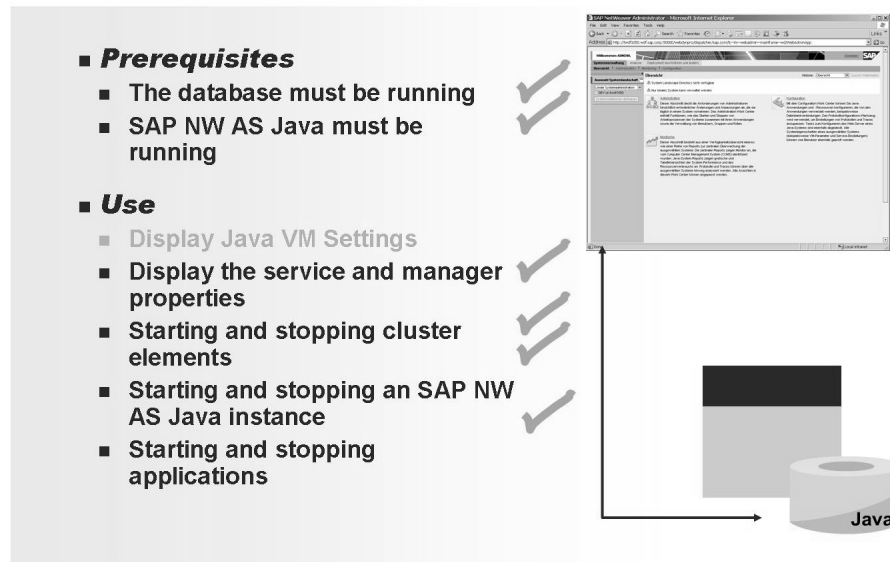


Figure 122: SAP NetWeaver Administrator 1

With the SAP NetWeaver Administrator, you log on to the http port of a SAP NetWeaver AS Java instance. It is therefore necessary that the SAP NetWeaver AS Java instance is started. The SAP NetWeaver Administrator can be called using the following URL: <http://<Rechnername>:<http-Port>/nwa>. In this way, it can be used for remote administration without it being necessary to install additional tools locally. Please note figure “SAP NetWeaver Administrator 1”. The figure “SAP NetWeaver Administrator 2” shows the special features of the tool.



■ **Special Features**

- The NWA will be continuously further developed in subsequent releases and new administration and monitoring functions will be added.

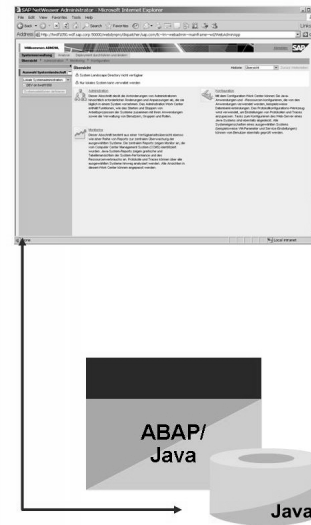


Figure 123: SAP NetWeaver Administrator 2

NWA was first shipped for SAP NetWeaver 04 SP Stack 12. An enhanced version was supplied for SAP NetWeaver 7.0. NWA will also be continuously further developed in subsequent releases and new administration and monitoring functions will be added.

Visual Administrator



■ **Prerequisites**

- The database must be running ✓✓
- SAP NW AS Java must be running ✓✓

■ **Use**

- Java VM settings configuration ✓✓✓
- Configuration of services and managers ✓✓✓
- Remote configuration ✓✓✓
- Starting and stopping services ✓✓✓
- Stopping an SAP NW AS Java instance ✓✓✓

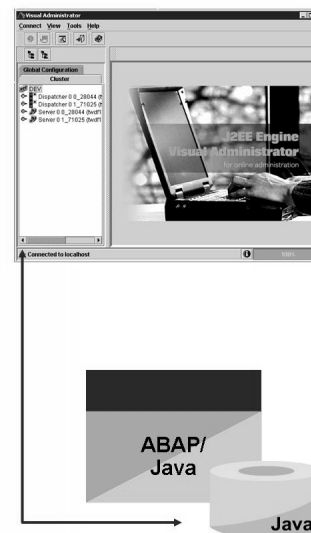


Figure 124: Visual Administrator 1

With the Visual Administrator, you log on to the P4 port of an SAP NetWeaver AS Java instance. It is therefore necessary that the SAP NetWeaver AS Java instance is started. The Visual Administrator can also be installed on a host on which no SAP NetWeaver AS Java instance is installed. In this way, it can be used for remote administration. You can use the Visual Administrator to configure the services and managers of all SAP NetWeaver AS Java instances in the Cluster. The changes to the selected parameters take effect during runtime. You can use the Visual Administrator to start or stop services. You cannot change the Virtual Machine parameters with the Visual Administrator. For information about this, see the figure “Visual Administrator 1”. The figure “Visual Administrator 2” shows the special features of the tool.

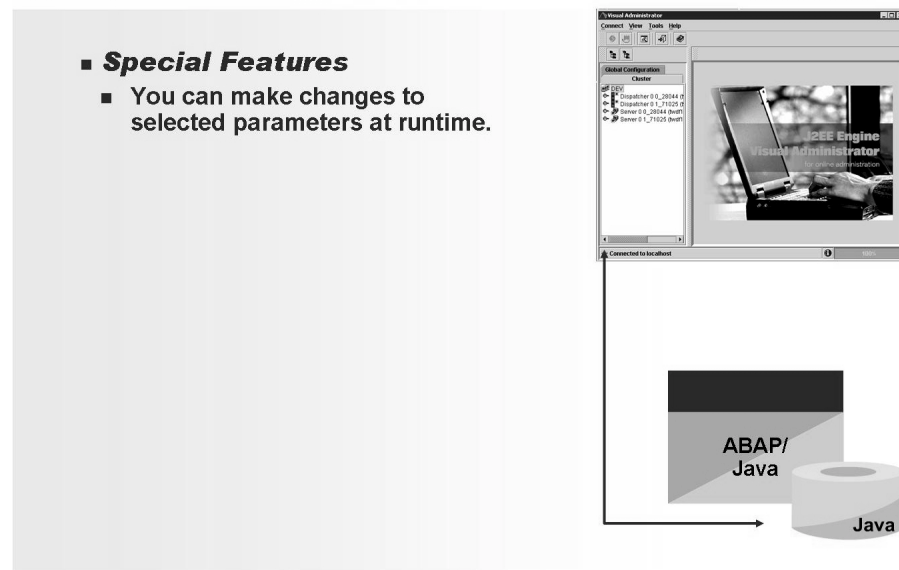


Figure 125: Visual Administrator 2

If necessary, you can use the Visual Administrator to shut down an SAP NetWeaver AS Java instance by stopping the Java dispatcher and the server processes. For an add-in installation you can use the dynamic profile parameter `rdisp/j2ee_start` of the SAP NetWeaver AS ABAP instance to deactivate the starting of the SAP NetWeaver AS Java instance.

Shell Console Administrator

With the Shell Console Administrator, you log on to an SAP NetWeaver AS Java instance. It is therefore necessary that the SAP NetWeaver AS Java instance is started. To use Shell Console Administrator, you must log on via Telnet at the SAP NetWeaver AS Java Telnet port. Telnet is available on many operating systems and can therefore be used for remote administration without additional installation.



■ Prerequisites

- The database must be running ✓
- SAP NW AS Java must be running ✓

■ Use

- Java VM settings configuration
- Configuration of services and managers
- Remote configuration ✓
- Starting and stopping services ✓
- Stopping an SAP NW AS Java instance ✓

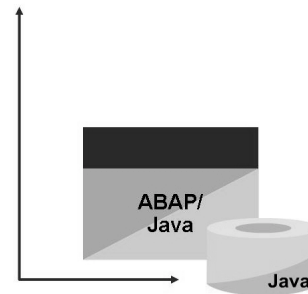


Figure 126: Shell Console Administrator 1

You can use Shell Console Administrator to display basic information for an SAP NetWeaver AS Java system in a simple way. Shell Console Administrator is also suitable for starting and stopping services. You cannot change VM parameters using Shell Console Administrator. For information about this, see the figure “Shell Console Administrator 1”. The figure “Shell Console Administrator 2” shows the special features of the tool.

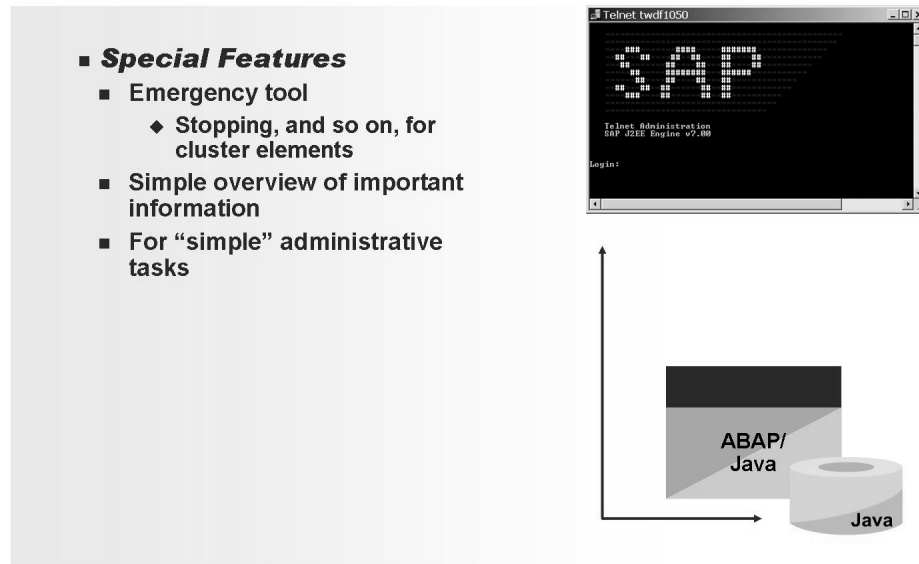


Figure 127: Shell Console Administrator 2

If necessary, you can use the Shell Console Administrator to shut down the SAP NetWeaver AS Java instance by stopping the Java dispatcher and the server processes.

You can use the Shell Console Administrator to configure the services and managers of all Java instances in the Cluster. It is significantly easier to make the changes to managers and services with the Visual Administrator. Therefore, only specialists should make changes to managers and services using Shell Console Administrator.

Calling the Administration and Configuration Tools

This section describes the features of calling the different tools.

If you are using an SAP system on the iSeries platform, please refer to SAP Note 1066038 - iSeries: Special Features of Tool Use in the J2EE environment.

Config Tool

The Config Tool is a tool for administering and configuring SAP NetWeaver AS Java. The Config Tool is started with the call *configtool.bat* at operating system level of the host on which the central instance of SAP NetWeaver AS Java is running. The file *configtool.bat* is stored under the following path at operating system level: `\usr\sap\<SID>\<central instance>\j2ee\configtool`. No user or password are required to call the Config Tool. You must still confirm whether you want to use the default database settings or whether you want to change them. The Config Tool is available to you to edit the parameters (see the figure “Config Tool 3”).

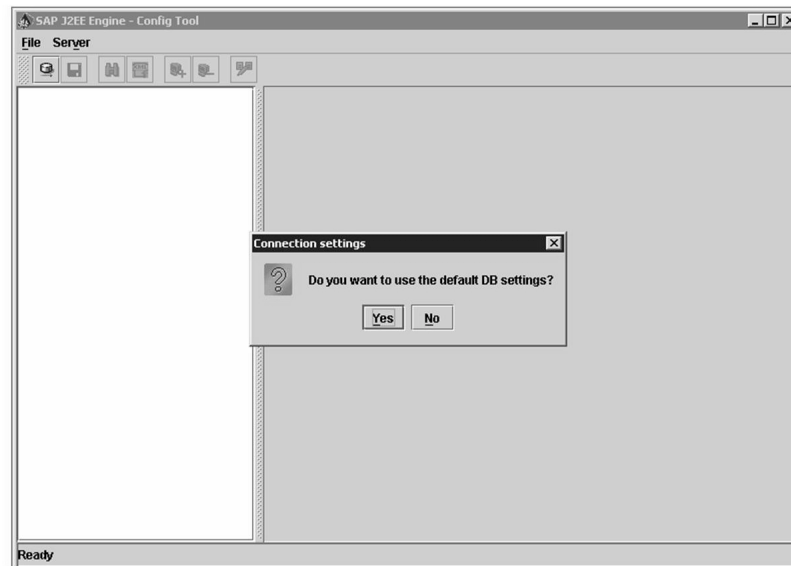


Figure 128: Config Tool 3

During the start process of the SAP NetWeaver AS Java instance, the parameters for the start and running operation are read from the database and copied to the file system. With the Config Tool, these parameters are maintained in the database. It is therefore necessary that the database is running so that the Config Tool can read the current parameters from the database. When the Config Tool is opened, you can see and change the default database settings by choosing the *No* button. The window contains the connection data for the database. This is maintained during the installation process. You can read in the parameters currently saved in the database at any time with the *connect to DB* button.



Connect

Secure Store File: G:\usr\sap\DEV\SYStglobal\security\data\SecStore.properties

Secure Store Key File: G:\usr\sap\DEV\SYStglobal\security\data\SecStore.key

System Name: DEV

Secure Store Lib: G:\usr\sap\DEV\SYStglobal\security\lib

Rdbms connections: jdbc/pool/DEV

Rdbms driverLocation: G:\sapdb\programs\runtime\jar\sapdbc.jar,

Rdbms initial connections: 1

Rdbms max connections: 5

☐ Overwrite rdbms settings

Load Cancel

Figure 129: Config Tool: Connect to DB

SAP NetWeaver Administrator

With the SAP NetWeaver Administrator, you log on to the http port of an SAP NetWeaver AS Java instance. It is therefore necessary that the SAP NetWeaver AS Java instance is started. The SAP NetWeaver Administrator can be called using the following URL: <http://<Rechnername>:<http-Port>/nwa>. The SAP NetWeaver Administrator start page displays the three areas Administration, Configuration and Monitoring.

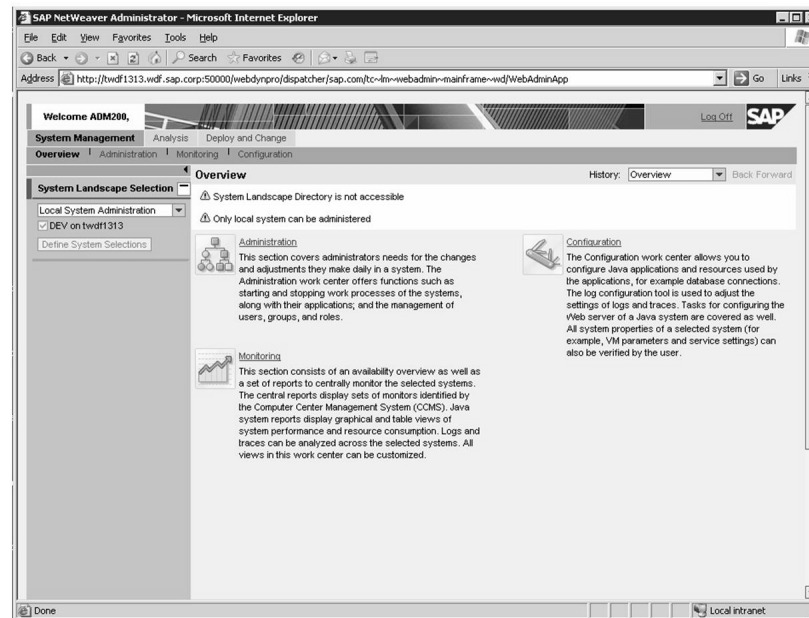


Figure 130: SAP NetWeaver Administrator: Start Page

Visual Administrator

The Visual Administrator is a tool for administering and configuring SAP NetWeaver AS Java. With the Visual Administrator, you log on to an SAP NetWeaver AS Java instance. This must therefore already be running - otherwise the logon process is unsuccessful. The Visual Administrator is started with the call *go.bat* at operating system level. The file *go.bat* is stored under the following path at operating system level: `\usr\sap\<SID>\<central instance>\j2ee\admin`. Once the Visual Administrator has been started, a selection window appears, in which you select the system to which you want to log on. If you are starting the Visual Administrator for the first time, you must first create a system entry in the selection window. Choose the button *New*. A new window, in which you define a connection to your system, appears (see the figure “Visual Administrator: Logon”). You enter, for example, the name of your system here and choose, for example *Direct Connection To a Dispatcher Node*.

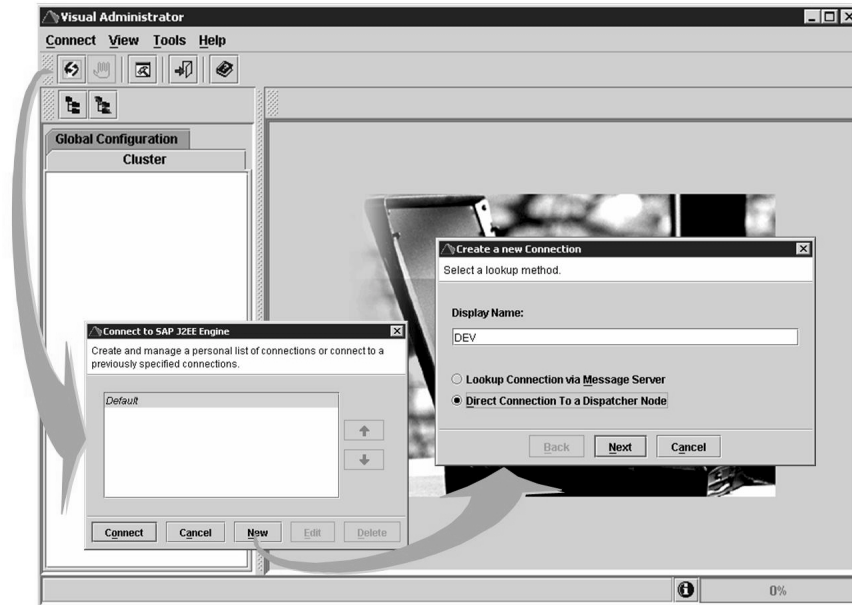


Figure 131: Visual Administrator: Logon

A window now appears in which you specify the logon parameters such as the user name of the administrator, the host name, and the P4 port for the Visual Administrator (see also “Visual Administrator: Logon new Connection”). You have now created a system entry in the logon window.



Figure 132: Visual Administrator: Logon new Connection

You can now logon to the system by choosing the *Connect* button, entering the password on the logon window, and choosing *Connect*. Your logon is successful as soon as a green diamond appears in the status bar and the value 100% is reached in the status bar. See also the figure “Visual Administrator: Logged On”.

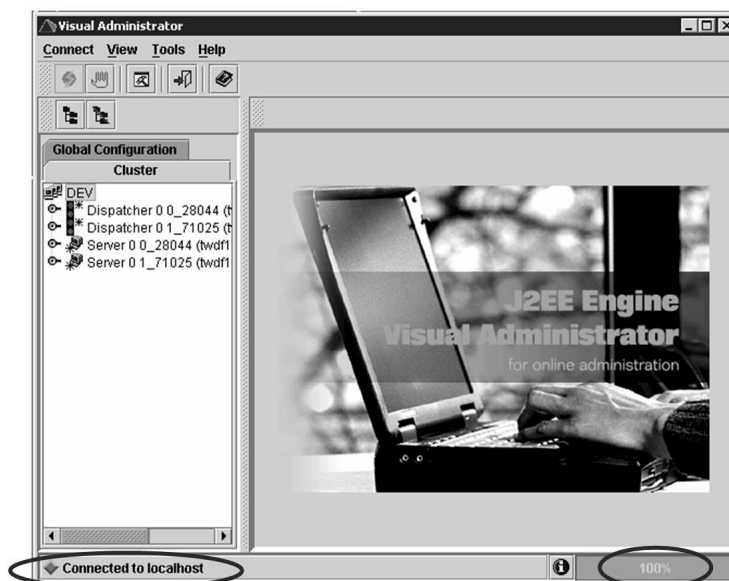


Figure 133: Visual Administrator: Logged On.

Shell Console Administrator

You can perform some administration tasks for SAP NetWeaver AS Java with Shell Console Administrator. To do this, you must specify the host name and the Telnet port of the SAP NetWeaver AS Java instance when you call Shell Console Administrator. You are then prompted to log on to the SAP NetWeaver AS Java, as shown in the figure “Shell Console Administrator”. You use the same user and password to do this as for the Visual Administrator. Since you are logging on to the SAP NetWeaver AS Java instance, this must already be running.



Figure 134: Shell Console Administrator

Exercise 18: Administration and Configuration Tools of SAP NetWeaver AS Java

Exercise Objectives

After completing this exercise, you will be able to:

- Call the most important SAP Web NetWeaver Java configuration tools

Business Example

The company management have decided to implement an SAP product that uses SAP NetWeaver AS Java. Many components of SAP NetWeaver, for example, use the Java application server. The following knowledge is required so that the IT department can ensure the smooth processing of business processes.

Task 1: Starting the Config Tool

Start the Config Tool

1. Log on to the server on which your SAP system is running.
2. Start the Config Tool for your SAP NetWeaver AS Java
3. Check the connection data for the database.

Task 2: Calling and Logging On to SAP NetWeaver Administrator

Log on to your SAP system as SAP NetWeaver Administrator (NWA).

1. Open a browser window at your front-end machine and call NWA..

Task 3: Starting the Visual Administrator and Logging On

Log on to your SAP system with the Visual Administrator.

1. Log on to the server on which your SAP system is running.
2. Start Visual Administrator.
3. Create a new entry for connecting to your SAP NetWeaver AS Java.
4. Log on to your SAP NetWeaver AS Java with the newly created entry.

Continued on next page

Task 4:

Optional: Log on to your SAP NetWeaver AS Java with Telnet.

1. Log on to the server on which your SAP system is running.
2. Open a “Command Shell”.
3. Log on to your SAP NetWeaver AS Java with Telnet.
4. Log on using your user and password from the previous task.
5. Exit Telnet with the *exit* command.

Solution 18: Administration and Configuration Tools of SAP NetWeaver AS Java

Task 1: Starting the Config Tool

Start the Config Tool

1. Log on to the server on which your SAP system is running.
2. Start the Config Tool for your SAP NetWeaver AS Java
 - a) Call the Explorer and navigate to the directory *G:\sap\<SID>\<central instance>\j2ee\configtool*, where *<SID>* stands for the name of your SAP system.
 - b) Double-click the file *configtool.bat*.
 - c) The Config Tool starts. This can take a moment.
3. Check the connection data for the database.
 - a) Confirm the prompt *Do you want to use the default DB settings?* with *No*.**Caution:** Do not change the database settings.
 - b) Use the *Load* button to start the Config Tool with these settings.

Continued on next page

Task 2: Calling and Logging On to SAP NetWeaver Administrator

Log on to your SAP system as SAP NetWeaver Administrator (NWA).

1. Open a browser window at your front-end machine and call NWA..
 - a) Call the following URL in the browser:
For the DEV system: *http://<host name>.wdf.sap.corp:50000*
For the QAS system: *http://<host name>.wdf.sap.corp:51000*
 - b) In the window, select the link *SAP NetWeaver Administrator*.
Alternatively, call the following URL:
For the DEV system: *http://<host name>.wdf.sap.corp:50000/nwa*
For the QAS system: *http://<host name>.wdf.sap.corp:51000/nwa*
 - c) Log on with your user and password (such as ADM200-##).

Task 3: Starting the Visual Administrator and Logging On

Log on to your SAP system with the Visual Administrator.

1. Log on to the server on which your SAP system is running.
 - a) Log on to the host on which your SAP system is running with the user *<sid>adm* and the password *<sid>adm*.
2. Start Visual Administrator.
 - a) Call the Explorer and navigate to the directory *G:\usr\sap\<SID>\<central instance>\j2ee\admin*, where *<SID>* stands for the name of your SAP system.
 - b) Double-click the file *go.bat*.
 - c) The Visual Administrator starts and displays a window with the name *Connect to SAP J2EE Engine*.

Continued on next page

3. Create a new entry for connecting to your SAP NetWeaver AS Java.
 - a) Choose the *New* button, and enter the name of your SAP system in the *DisplayName* field in the new window.
Select *Direct Connection to Dispatcher Node* and choose the *Next* button.
 - b) In the *User Name* field, enter your Java user (such as ADM200-##).
 - c) In the *Host* field, enter “localhost”.
 - d) In the *Port* field, enter “50004”, if your SAP system is called “DEV” and “51004”, if your SAP system is called “QAS”. Save your settings.
4. Log on to your SAP NetWeaver AS Java with the newly created entry.
 - a) Select your entry, choose *connect*, enter your password, then choose *connect* again, and change your password.

Task 4:

Optional: Log on to your SAP NetWeaver AS Java with Telnet.

1. Log on to the server on which your SAP system is running.
 - a) Log on to the host on which your SAP system is running with the user *<sid>adm* and the password *<sid>adm*.
2. Open a “Command Shell”.
 - a) Choose *Start* → *Run*, enter *cmd* and press *Return*.
3. Log on to your SAP NetWeaver AS Java with Telnet.
 - a) For a DEV system: In the *Command Shell*, type *telnet localhost 50008*.
For a QAS system: In the *Command Shell*, type *telnet localhost 51008*.
4. Log on using your user and password from the previous task.
5. Exit Telnet with the *exit* command.



Lesson Summary

You should now be able to:

- Name various configuration and administration tools for the SAP NetWeaver AS Java
- Describe the primary usage areas of configuration and administration tools

Lesson: General Configuration of the SAP NetWeaver AS Java Cluster with the Config Tool

Lesson Overview

This lesson provides you with a brief, simplified introduction to Java Hotspot VM memory management. You will learn the VM Settings configuration as well as how to perform global and local maintenance using the Config Tool.



Lesson Objectives

After completing this lesson, you will be able to:

- Maintain settings for an SAP NetWeaver AS Java with the Config Tool

Business Example

After the installation of an SAP NetWeaver Application Server, fundamental settings need to be made for the cluster of an SAP NetWeaver AS Java. Knowledge about the available tools is required to do this. One of these tools is the Config Tool, which is introduced in this lesson.

Brief Introduction to Memory Management of the Java HotSpot VM

A number of terms for the memory management of a Java Virtual Machine are explained in this section (in a simplified form).

Memory Allocation Terms

The memory area of a Java Virtual Machine (JVM or VM) is mainly divided into three areas, which are called the *young generation*, the *tenured generation*, and the *permanent generation*. The differences between the “generations” are discussed in more detail later. We will first consider the commonalities, however. A “generation” reserves space in the address area of the host.

On start-up, the Virtual Machine allocates operating system memory for each “generation”. This initially allocated memory area is called “initial” or *initial size*. This occupancy of this space is administered internally by the VM.

Once the initial space has been used, the VM allocates further operating system memory space in stages up to a maximum amount. This maximum value is defined in “max size”.

The VM automatically takes care of the allocation of memory space for Java applications. The memory space is implicitly assigned when an object is created. Even if a large amount of memory is required, this does not mean that the system is at risk. The VM determines which objects are no longer used and releases the memory areas which they currently occupy. This task is performed by the Special Java Agent names Garbage Collector (GC) which forms part of the VM. Its job is to prevent the occurrence of situations in which there is a danger of memory bottlenecks.

The memory space that is available is called the *available memory* or *allocated memory*. Since this space is reserved by the operating system, it is also referred to as “reserved” space since the entire space up to the maximum size is “available”. The space that is not yet reserved is called *virtual memory*. However, this should not be confused with the “virtual memory” of the operating system. If less space is required, the memory is returned to the operating system, also in stages. See also the figure “Space Management Terms”.

The reserved memory space (*available memory*) is potentially available to the VM. However, it does not have to be used in full. The memory space that is actually used by Java applications is referred to as *used memory*.

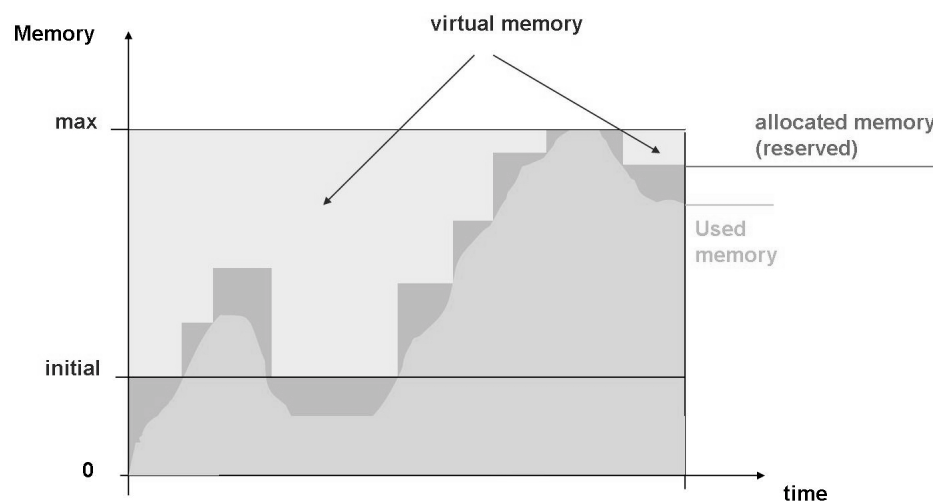


Figure 135: Space Management Terms

Memory Allocation of the Java VM in a Simplified Form

The three main memory areas of the VM, the “young, tenured”, and “permanent generations” differ from one another due to the data stored in them. The objects that have been newly created by the applications are stored in the *young generation*. Objects that have been required for a longer period of time by an application are automatically moved to the *tenured generation*. The newer objects are in the “young

generation” and the older objects are in the “tenured generation”. Objects that are permanently required by the VM, such as classes and methods, are stored in the *permanent generation*. Objects that are no longer required by the applications are automatically removed from the “generation”. This process is known as *garbage collection*.

As you already know from the subsection “Memory Allocation Terms”, the “generations” have an initial and a maximum size. For the “young generation”, you can define the “initial size” with the parameter `-XX:NewSize`, and the “max size” with the parameter `-XX:MaxNewSize`. You can define the corresponding values for the “permanent generation” (also abbreviated to the “perm generation”) with the parameters `-XX:PermSize` and `-XX:MaxPermSize`. You cannot directly define the initial and maximum sizes of the “tenured generation”. These are calculated from the parameters for the “young generation” and the parameters `-Xmx` and `-Xms`. The parameter `-Xmx` is called the “max heap size” and defines the total size of the “young” and “tenured generations”. The parameter `-Xms` is called the “start heap size” or “initial heap size” and defines the total initial size of the “young” and “tenured generations”. See also the figure “Memory Allocation of the Java VM (Simplified)”.

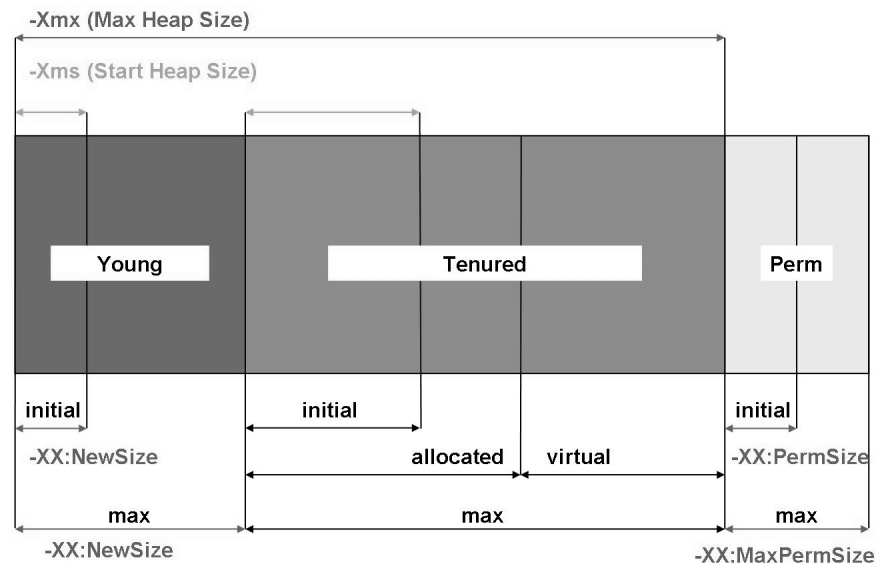


Figure 136: Memory Allocation of the Java VM (Simplified)

In addition to the memory area for the “generations”, the VM also reserves space for its processes and threads.

Config Tool: Overview

The window on the left in the Config Tool displays the structure of the cluster in “global settings” and in settings for the individual cluster elements. The cluster elements are always assigned to an instance with an instance number. This instance number is also contained in the name of the dispatcher or server processes. See also the figure “Window Sections in the Config Tool”. As a guideline, the following should be of use: if there is no instance number next to the dispatcher or server, you are in the global settings. The parameters of the selected elements are displayed and maintained in the window on the right of the Config Tool.



Caution: You can only use the Config Tool to change parameters if all instances of a SAP NetWeaver AS Java are stopped.

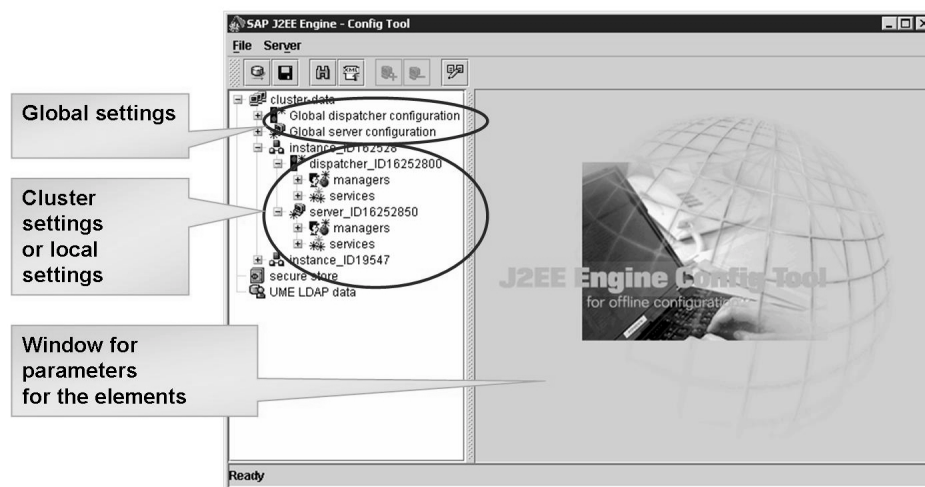


Figure 137: Window Sections in the Config Tool

Configuration of the Java VM Settings with the Config Tool

The settings for the Virtual Machine (VM) are maintained with the Config Tool. You maintain the VM Settings by selecting a dispatcher or server node of the instance in the window of the left of the Config Tool. The VM parameters with their settings then appear in the window on the right. The runtime parameters are on the *General* tab page. See also the figure “VM Settings”. The *Bootstrap* tab page contains the settings for the start process. The parameter *Java Home* sets where in the file system the SDK resides. You can also change the parameter *Max heap size* with the arrow keys. Max heap size is referred to in SAP Notes or the Sun documentation as “-Xmx”. The other Java parameters illustrated in the figure “VM Settings” are familiar from the

previous section. You can edit or delete them in the Config Tool. You can also add additional parameters, such as “-XX:MaxPermSize=128m” here. After you have completed your maintenance work, you must save the settings in the database by choosing *Apply changes*. When the SAP NetWeaver AS Java instance is restarted, it is started with the new values.

If you have changed settings but not yet saved them with *Apply changes*, you can reload the older settings with “Connect to DB”. If you saved settings (*Applied*), you cannot any longer retrieve the old settings.

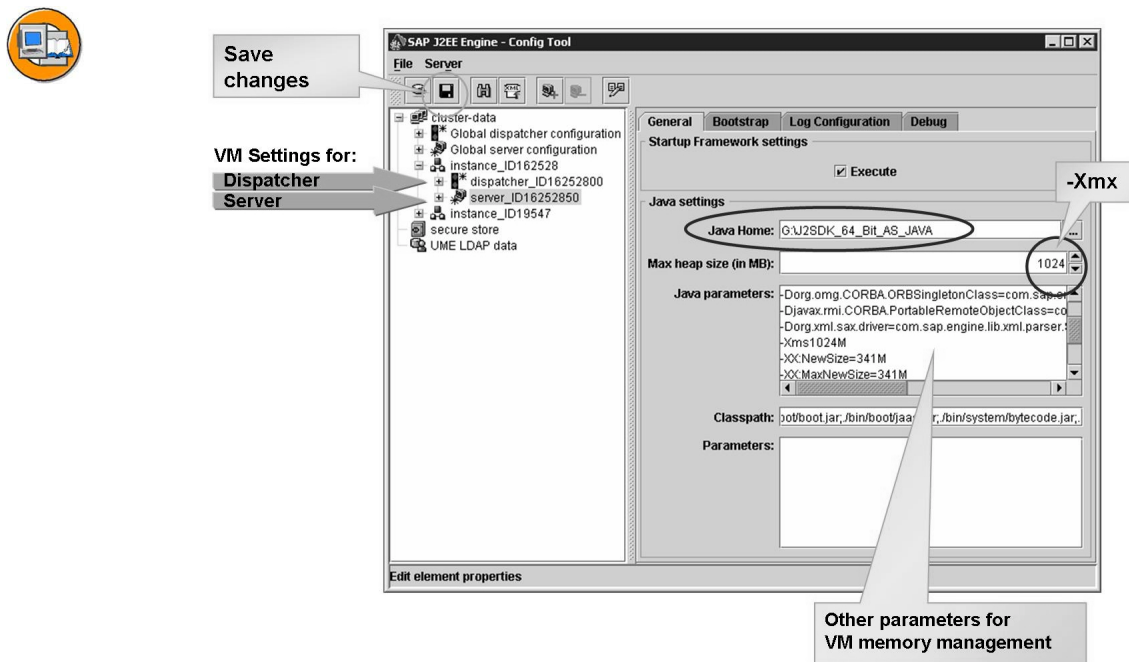


Figure 138: VM settings

Configuration of Cluster Elements with the Config Tool

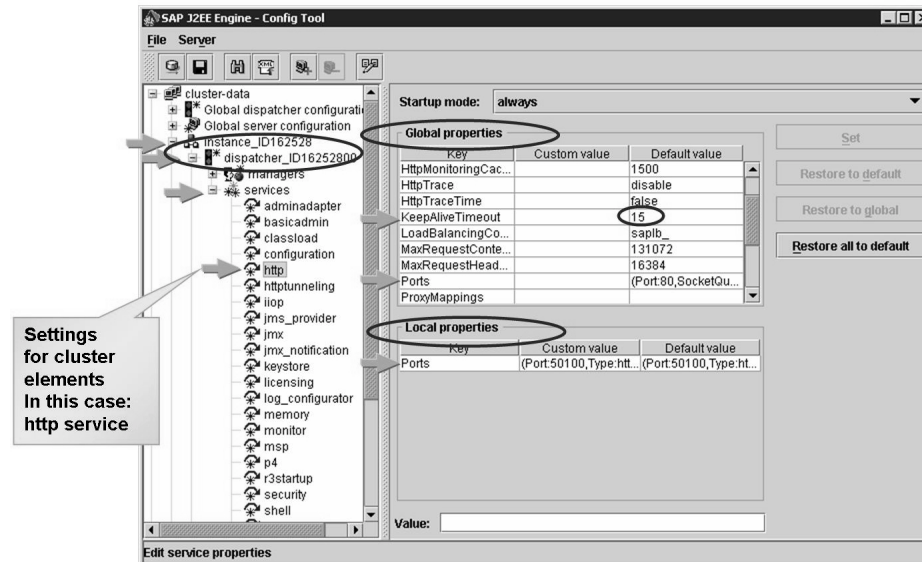


Figure 139: Maintaining Parameters of the Cluster Elements

The figure “Maintaining Parameters of the Cluster Elements” shows some parameters of the http service. As you can see in the window on the left, these are settings for the cluster elements and not for the global elements. The window on the right is subdivided into global settings (which cannot be maintained here) and local settings, which are maintained in this window. There are no local settings for the parameter *KeepAliveTimeout*. The global settings are therefore used. In this case, this is the value “15”. Global and local settings are maintained for the parameter *Ports*. Local settings override the global settings. Therefore, only the local settings are relevant for *Ports*. There are “Custom values” and “Default values” for both local and global settings. The custom values override the default values. Therefore, the relevant value for *Ports* in this figure can be seen under “Local properties, Custom Value”. The evaluation sequence is clarified by the figure “Use of the Cluster Settings”.



Setting used:
Local before Global; Custom before Default

Key	Local properties		Global properties	
	Custom	Default	Custom	Default
KeepAliveTime				15
Ports	(Port:50000,...	(Port:50000,...		(Port:80,SocketQueue:20..

Figure 140: Use of the Cluster Settings

Now that the evaluation sequence and the difference between the global and local settings is clear, we will consider the maintenance of global settings.

Global Settings in the Config Tool

You can find the global parameter settings under the dispatcher or server nodes that are not assigned to any instance (you can identify these by the lack of ID number). If you select the http service under the global dispatcher node, a screen similar to that shown in the figure “Global Parameter Settings” appears. To change a parameter value, select the parameter. The *Value* field now appears, ready for input and with the default value (or the custom value, if there is one) as a proposal. You can now edit this value. You need to confirm the maintained values with the *Set* button. It then appears in the global settings under *Custom value*. For these values to take effect, you need to save the settings in the database with the *Apply changes* button (as is already familiar from maintaining the VM settings) and restart the SAP NetWeaver AS Java instance.

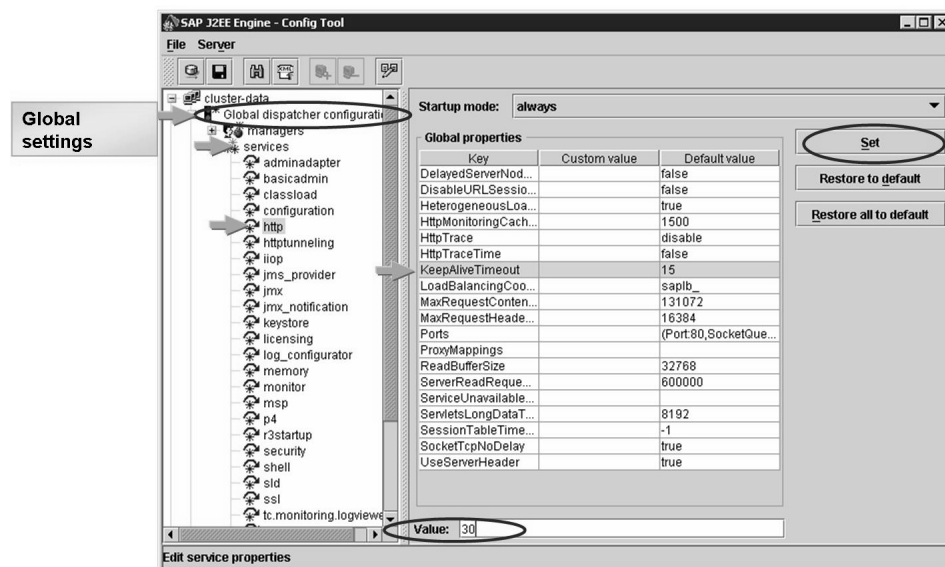


Figure 141: Global Parameter Settings

Now that you are familiar with maintaining the global settings, we will turn to maintaining local settings.

Local Settings in the Config Tool

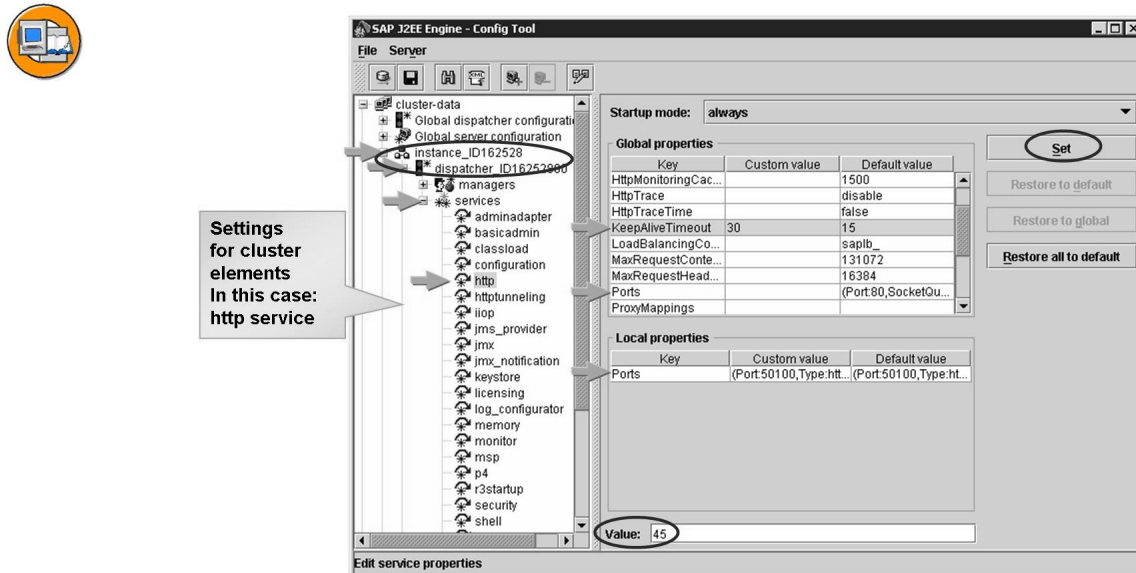


Figure 142: Local Custom Settings

The figure “Local Custom Settings” again shows the local maintenance for the http service. You can now see the previous changed custom value in the global setting for the “KeepAliveTimeout” parameter. You can override this for the selected dispatcher node by maintaining the parameter locally. To do this, select the parameter from the global settings. The *Value* field now appears, ready for input and preassigned with the value from “Global properties, Custom value”. Edit this value to the value you desire, and confirm this with the *Set* button. The parameter now appears under the local settings with the maintained value under *Custom value*. See also the figure “Maintained Custom Settings”.

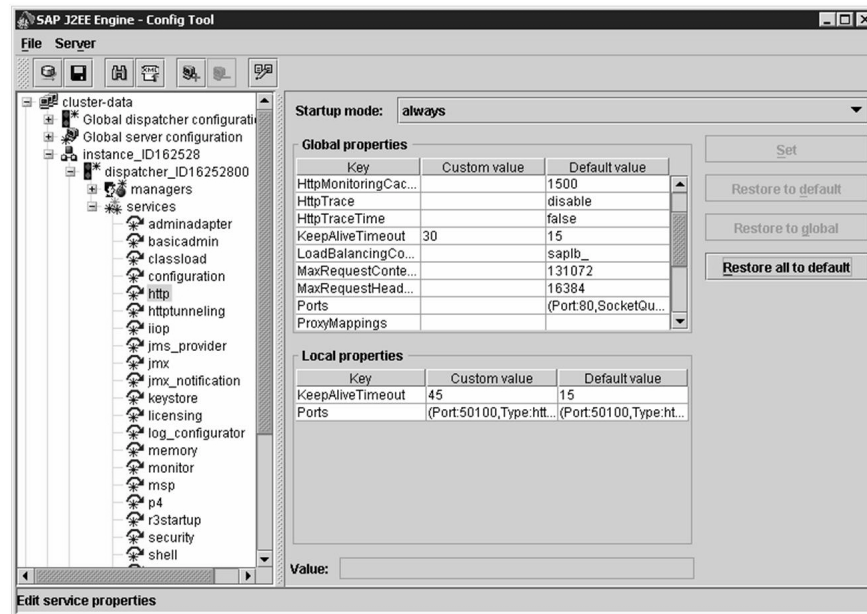


Figure 143: Maintained Custom Settings

For these values to take effect, you need to save the settings in the database with the *Apply changes* button and restart the SAP NetWeaver AS Java instance.

Resetting Values

You can also reset the maintained values to the default values or to the global settings. Select a parameter under the local settings (such as the parameter *KeepAliveTimeout* from the previous example). The buttons *Restore to Default* and *Restore to global* are now active. With *Restore to Default*, you delete the entry under *Custom value* for the local settings. *KeepAliveTimeout* now has the value “15” (unlike the global value, for which the *Custom value* “30” is maintained). If you choose the *Restore to global* button, the entire entry is deleted from the local settings. The global settings are now used for this parameter again. In our “example” with the parameter *KeepAliveTimeout*, the value “30” is now maintained for *Global properties, Custom value*.

Export of Configuration Data

You have the option of saving the SAP NetWeaver AS Java configuration data by exporting it to an XML file. The SAP NetWeaver AS Java data can be exported from the database and imported again later to restore the settings for a cluster element.

This procedure allows you to export the data from an entire cluster (all Java instances), individual cluster elements (Java dispatcher/server), or instance.properties files of Java instances.

You can export the system configuration to an XML file using the menu path *File → Export Unsaved Changes to XML* in the Config Tool.

You can export the configuration of a cluster element to an XML file using the menu path *File → Export Data From DB* in the Config Tool. You can also select multiple or all cluster elements. You can select the instance.properties files as well. This file is in the directory */configtool* by default.

The data is imported using the menu path *File → Import Data To DB*. Only the same node ID can be overwritten with the previously exported data.

The Offline Configuration Editor

In addition to the Config Tool, there is also the “Offline Configuration Editor”, which you can call from the Config Tool.



Caution: You must only use the “Offline Configuration Editor” if you SAP specifically recommends that you do so, since inappropriate use can destroy the SAP NetWeaver AS Java.

Exercise 19: General configuration of SAP NetWeaver AS Java with the Config Tool

Exercise Objectives

After completing this exercise, you will be able to:

- Maintain parameters for the SAP NetWeaver AS Java with the Config Tool

Business Example

The company management have decided to implement an SAP product that uses SAP NetWeaver AS Java. Many components of SAP NetWeaver, for example, use the Java application server. The following knowledge is required so that the IT department can ensure the smooth processing of business processes.

Task 1: Stopping SAP NetWeaver AS Java

Stop the SAP NetWeaver AS Java.

1. Stop all instances of SAP NetWeaver AS Java.

Task 2: Checking the Java Home Paths

Check the Java Home directories set for your instances.

1. Start the Config Tool.
2. Check the settings of an instance.
3. Check the dispatcher settings of an instance.
4. Check the settings for the first server process (for example, Server 0).
5. Perform 1 to 4 for all other instances of your cluster.

Task 3: Maintaining the Initial Heap Size

Optional: Check the Initial Heap Size of the server process of your SAP NetWeaver AS Java dialog instance.

1. Identify which cluster nodes belong to your dialog instance.
2. Check that the values for the *Initial Heap Size (Xms)* of the server process and the *Max Heap Size (Xmx)* are the same.

Continued on next page



Caution: Do not set a “=” for *Xms* otherwise the instance will not start.

Task 4: Maintaining Global Settings

Global Settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the http service (for dispatchers) **globally** to the value “55”.

Task 5: Maintaining Local Settings

Optional: Local settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the http service (for the dispatcher) **locally** to the value “60” for your (Java) dialog instance.

Task 6: Starting SAP NetWeaver AS Java

Start the SAP NetWeaver AS Java.

1. Start all instances of SAP NetWeaver AS Java.

Solution 19: General configuration of SAP NetWeaver AS Java with the Config Tool

Task 1: Stopping SAP NetWeaver AS Java

Stop the SAP NetWeaver AS Java.

1. Stop all instances of SAP NetWeaver AS Java.
 - a) Log on to the ABAP stack of the SAP system.
 - b) Start transaction SMICM and stop all the instances of the Java cluster by choosing the following menu path: *Administration* → *J2EE Cluster (global)* → *Send Hard Shutdown* → *Without Restart*.

Task 2: Checking the Java Home Paths

Check the Java Home directories set for your instances.

1. Start the Config Tool.
 - a) Log on to the operating system of your training server.
 - b) Navigate to the subdirectory *g./usr/sap///j2ee/configtool* and start the Config Tool with the file *configtool.bat*.
2. Check the settings of an instance.
 - a) Select an instance.
 - b) Select the tab page *Message Server & Bootstrap*. Check the Bootstrap settings and, if necessary, change the appropriate Java Home, such as “G:/J2SDK_64_Bit_AS_JAVA”. If you have made changes, save them.
 - c) Select the tab page *Servers General*. Check the Java settings and, if necessary, change the appropriate Java Home, such as “G:/J2SDK_64_Bit_AS_JAVA”. If you have made changes, save them.

Continued on next page

3. Check the dispatcher settings of an instance.
 - a) Select the dispatcher of an instance.
 - b) Select the *General* tab page. Check the Java settings and, if necessary, change the appropriate Java Home, such as “G:/J2SDK_64_Bit_AS_JAVA”. If you have made changes, save them.
 - c) Select the *Bootstrap* tab page. Check the Java settings and, if necessary, change the appropriate Java Home, such as “G:/J2SDK_64_Bit_AS_JAVA”. If you have made changes, save them.
4. Check the settings for the first server process (for example, Server 0).
 - a) Follow the same procedure to do this as for the dispatcher settings.
5. Perform 1 to 4 for all other instances of your cluster.
 - a) See 1 to 4.

Task 3: Maintaining the Initial Heap Size

Optional: Check the Initial Heap Size of the server process of your SAP NetWeaver AS Java dialog instance.

1. Identify which cluster nodes belong to your dialog instance.
 - a) Switch to the directory that contains the developer traces for the dialog instance.

In the DEV system, this is: *G:\usr\sap\DEV\D01\work*

In the QAS system, this is: *G:\usr\sap\QAS\D11\work*
 - b) Open the developer trace for the Java dispatcher (dev_dispatcher). Search for “Cluster ID”. You will find the Cluster ID for the dispatcher there. For the dispatcher, this ends with 00. If you leave off the 00 at the end, you have the Cluster ID for the instance.
 - c) Alternatively, you can find the Cluster ID in the *instance.properties* file, which is stored in the subdirectory *j2ee\cluster* of the instance (such as D01).
 - d) Select the instance for which you have found the Cluster ID.
2. Check that the values for the *Initial Heap Size (Xms)* of the server process and the *Max Heap Size (Xmx)* are the same.

Continued on next page



Caution: Do not set a “=” for *Xms* otherwise the instance will not start.

- a) Select the server node.
- b) If necessary, increase the parameter *-Xms* to the value that is specified for *Max Heap Size*.

Task 4: Maintaining Global Settings

Global Settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the http service (for dispatchers) **globally** to the value “55”.
 - a) Select the entry for *Global Dispatcher Configuration*.
 - b) Expand the services.
 - c) Select *http*.
 - d) Under the *Global Settings*, select the parameter *KeepAliveTimeout*.
 - e) Maintain the value “55” for Value.
 - f) Choose the button *Set*.
 - g) Save the setting with the *Apply changes* button.

Task 5: Maintaining Local Settings

Optional: Local settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the http service (for the dispatcher) **locally** to the value “60” for your (Java) dialog instance.
 - a) Select your dialog instance and open the tree structure.
 - b) Select the dispatcher for **this instance**.
 - c) Expand the services.
 - d) Select *http*.
 - e) Under the *Global Settings*, select the parameter *KeepAliveTimeout*.
 - f) Maintain the value “60” for Value.
 - g) Choose the button *Set*. The parameter *KeepAliveTimeout* now appears under *Local Properties*.
 - h) Save the setting with the *Apply changes* button.

Continued on next page

Task 6: Starting SAP NetWeaver AS Java

Start the SAP NetWeaver AS Java.

1. Start all instances of SAP NetWeaver AS Java.
 - a) Log on to the ABAP stack of the SAP system.
 - b) Start transaction SMICM and start all the instances of the Java cluster by choosing the following menu path: *Administration* → *J2EE Clusters (global)* → *Restart* → *Yes*.



Lesson Summary

You should now be able to:

- Maintain settings for an SAP NetWeaver AS Java with the Config Tool

Lesson: General Configuration of the SAP NetWeaver AS Java Cluster with the Visual Administrator

Lesson Overview

This lesson introduces global and local maintenance with the Visual Administrator.



Lesson Objectives

After completing this lesson, you will be able to:

- Maintain settings for an SAP NetWeaver AS Java with the Visual Administrator

Business Example

After the installation of an SAP NetWeaver Application Server, fundamental settings need to be made for the SAP NetWeaver AS Java. There are various tools and processes for making these settings.

Visual Administrator: Overview

There are two different views in the Visual Administrator: the view for global configuration, and the view for local configuration, which is also called “cluster” configuration. For more information, see also figure: “Global vs. Local Configuration”. With “cluster” configuration, all active nodes of the cluster are visible and can be maintained with the Visual Administrator. In the case of global configuration, a distinction is made between maintaining the dispatcher nodes and the server nodes using the corresponding tab pages.

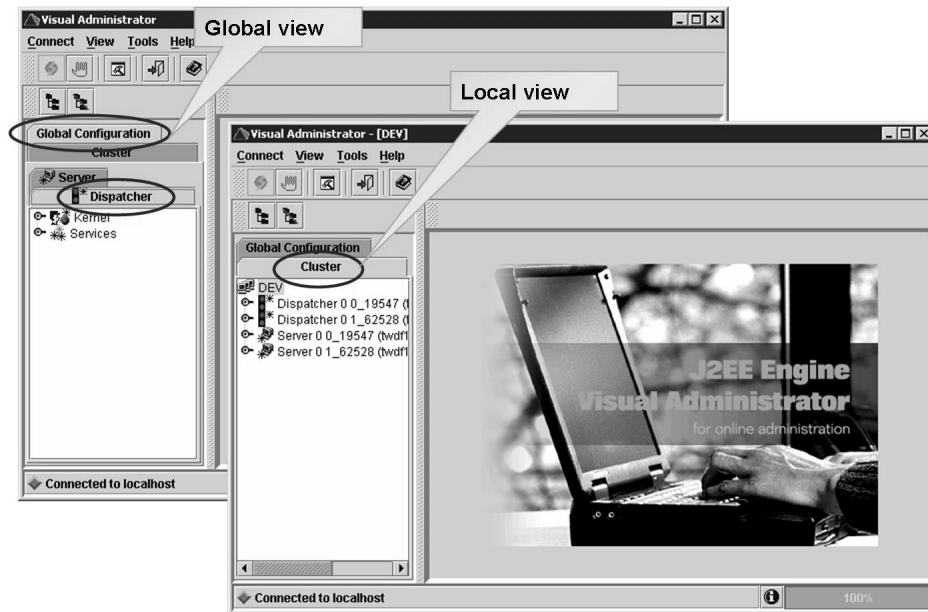


Figure 144: Global vs Local Configuration

Maintenance with the “Visual Administrator” is explained here, using the example of the *KeepAliveTimeout* parameter that is already familiar from the lesson about the Config Tool. The figure “Local Values and Icons” shows the local setting of the parameter *KeepAliveTimeout*, which will be maintained first globally and then locally with the Visual Administrator in the following. It also shows the most important icons for parameter maintenance.

- Update
- Save Properties
- Reset Properties
- Reset Default Properties
- Show Details

The maintained values are transferred to the view using the *Update* button. You must then activate these and save them in the database with the *Save Properties*. As long as the maintained values have not yet been saved, you can undo the changes with the *Reset Properties* button. You can display the “Default Values” with the *Show Details* button. The *Reset Default Properties* button resets the value to the value specified under “Default Value”. You complete the value maintenance with the *Save Properties* button.

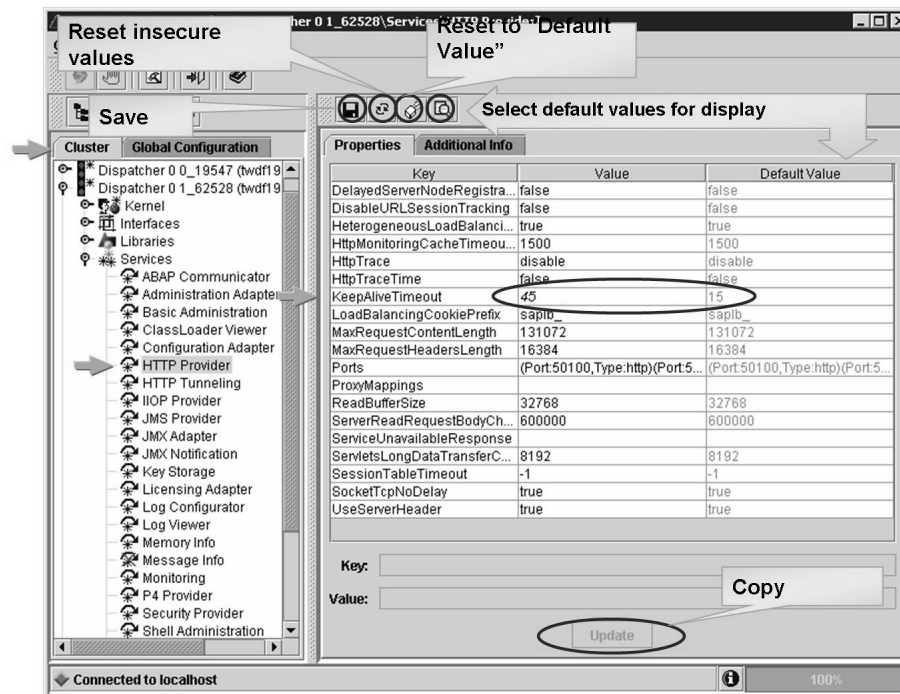


Figure 145: Local Values and Icons

Visual Administrator: Global Configuration

With the global settings, it is also possible to transfer the maintained values as local settings for the parameter. This will be examined in more detail in the subsection “Global Configuration with Local Maintenance”. The next subsection considers the maintenance of global parameters without local maintenance.

Global Configuration Without Local Maintenance

The global settings are maintained using the *Global Configuration* tab page.

The parameter *KeepAliveTimeout* is globally maintained under *Global Configuration* → *Dispatcher* → *HTTP Provider*. If you select the parameter, you can maintain it in the *Value:* field. You transfer the maintained value to the view by choosing the *Update* button. Once you have finished the value maintenance, complete the maintenance with the *Save Properties* button. A new input window now appears, which offers you the option of transferring the settings to the individual nodes in the cluster. In this case, you do not want to use this option, and choose the *No* button. See also the figures “Global Settings 1a” and “Global Settings 1b”.

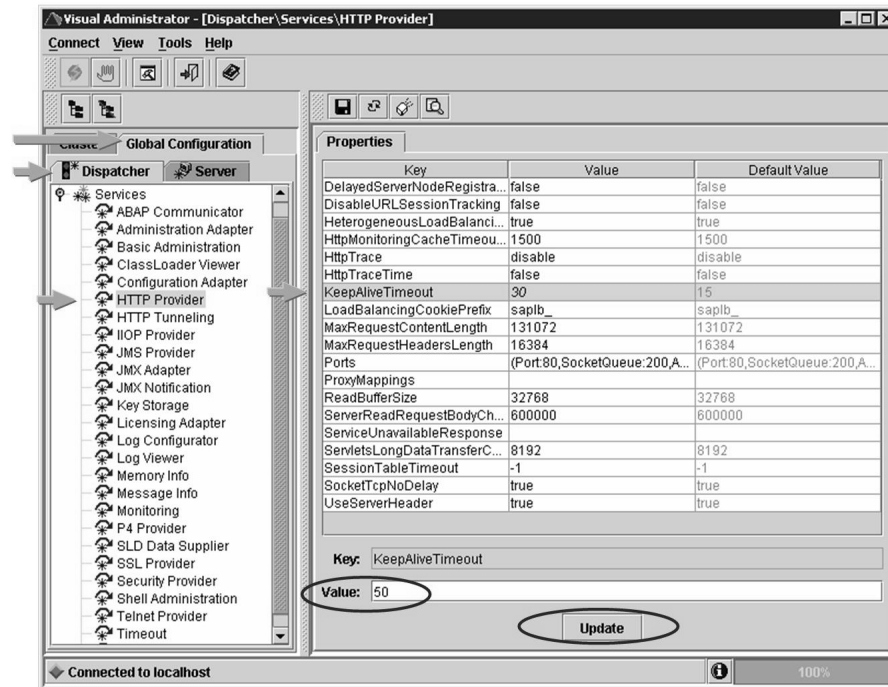


Figure 146: Global Settings 1a

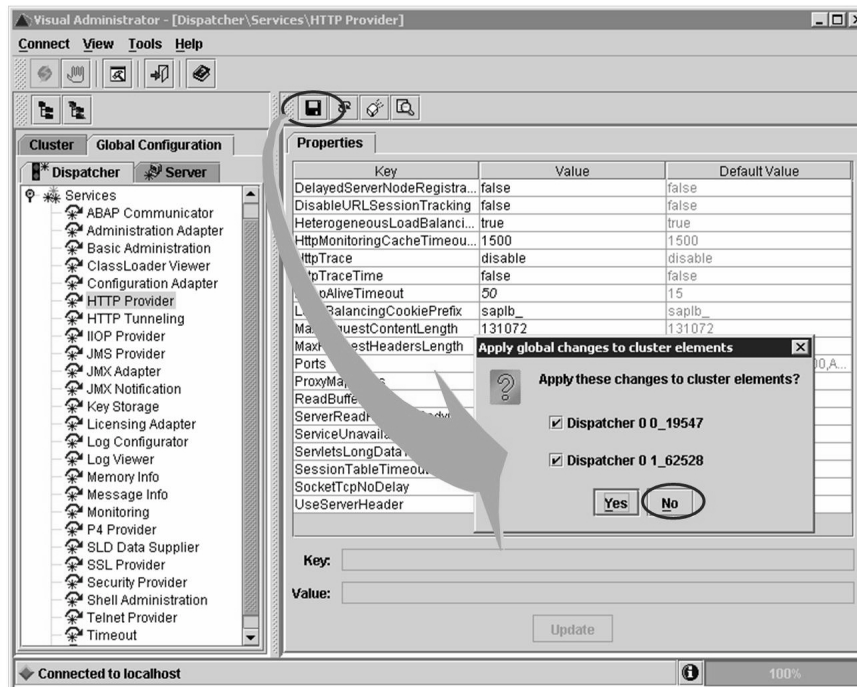


Figure 147: Global Settings 1b

In the figure “Config Tool: Check Settings 1” shows a check of the settings with the Config Tool. The maintained value is displayed as a custom value in the global settings and nothing is overridden locally. We now use the Visual Administrator to check the values currently in effect under the cluster settings, and see that the global maintenance has not yet taken effect. See also the figure “Visual Administrator: Check Current (Local) Values 1a”. For the new global value to take effect, the SAP NetWeaver AS instance must be restarted (as with the Config Tool). After the restart, the Visual Administrator shows in the cluster settings, that the globally maintained value is now active. See also the figure “Visual Administrator: Check Current (Local) Values 1b”. Changed values or values that are different from the default values of the Visual Administrator are displayed in italics.

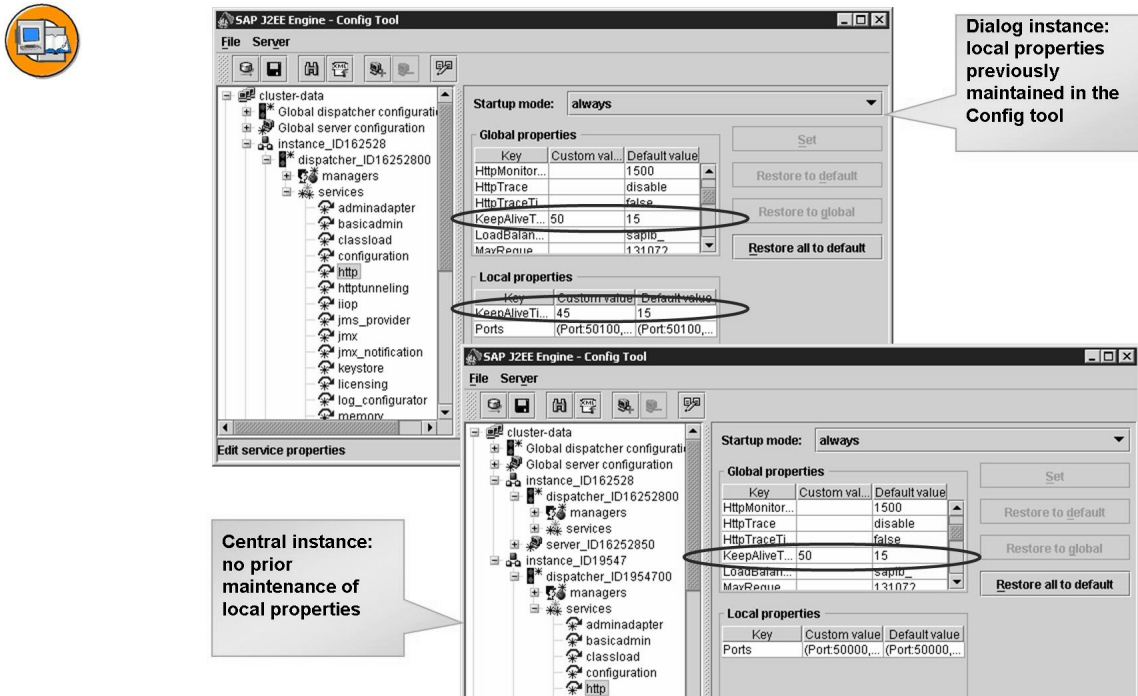


Figure 148: Config Tool: Check Settings 1

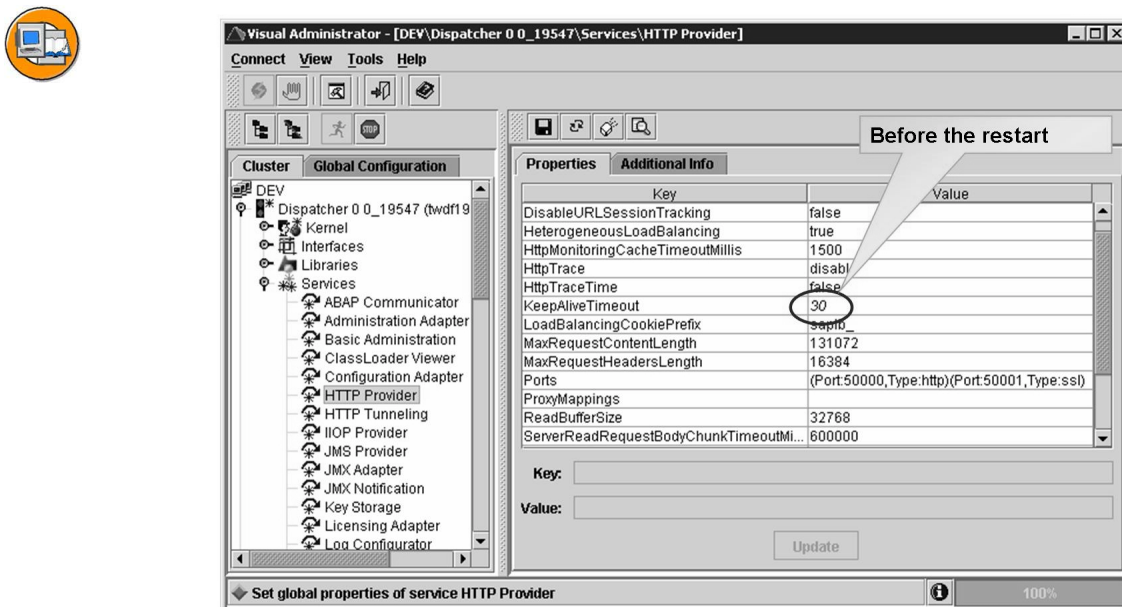


Figure 149: Visual Administrator: Check Current (Local) Values 1a.

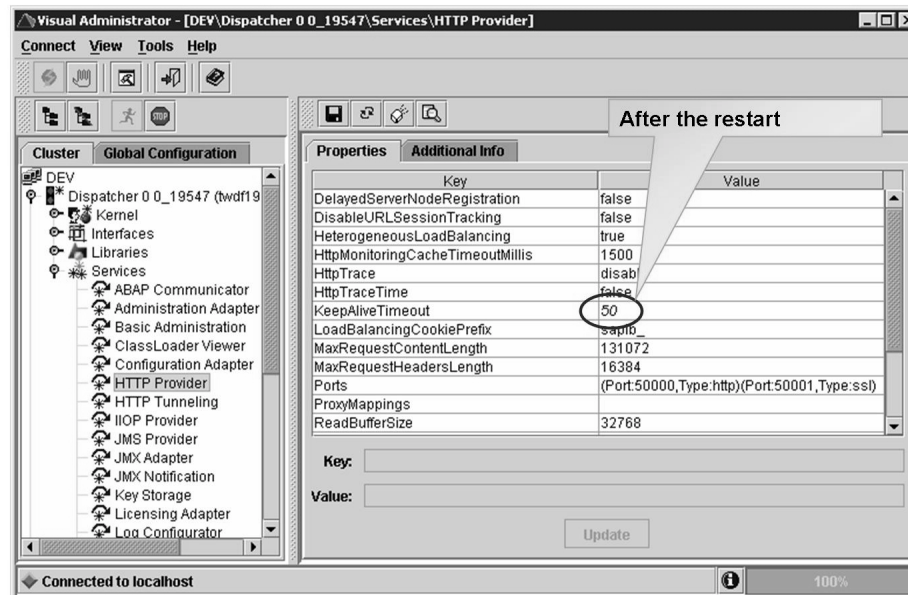


Figure 150: Visual Administrator: Check Current (Local) Values 1b.

Global Configuration with Local Maintenance

In the following, we proceed in a similar way to the previous subsection. See the figures “Global Settings 2a” and “Global Settings 2b”. With the current procedure, we decide to transfer the settings to the nodes in the cluster by choosing the *Yes* button instead of the *No* button.

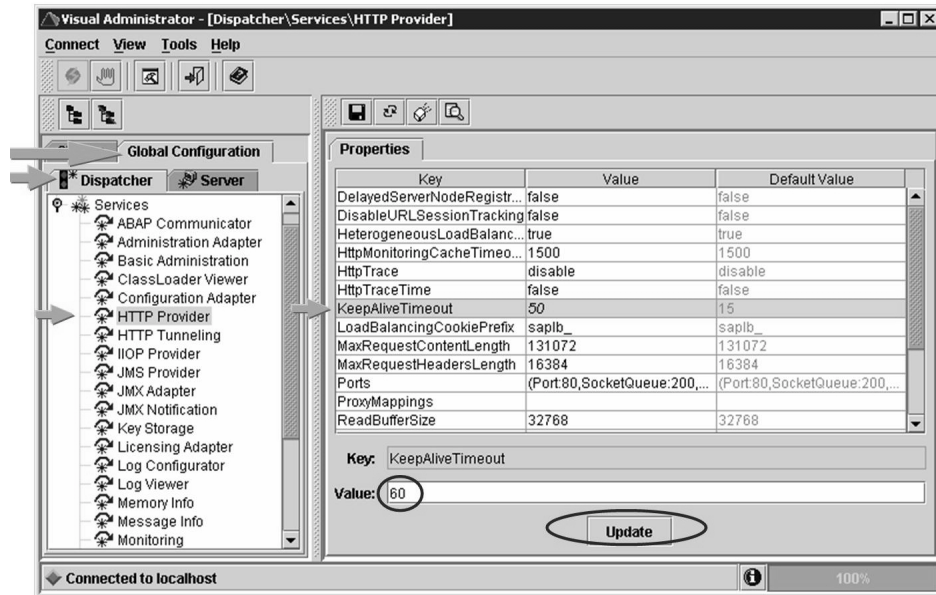


Figure 151: Global Settings 2a

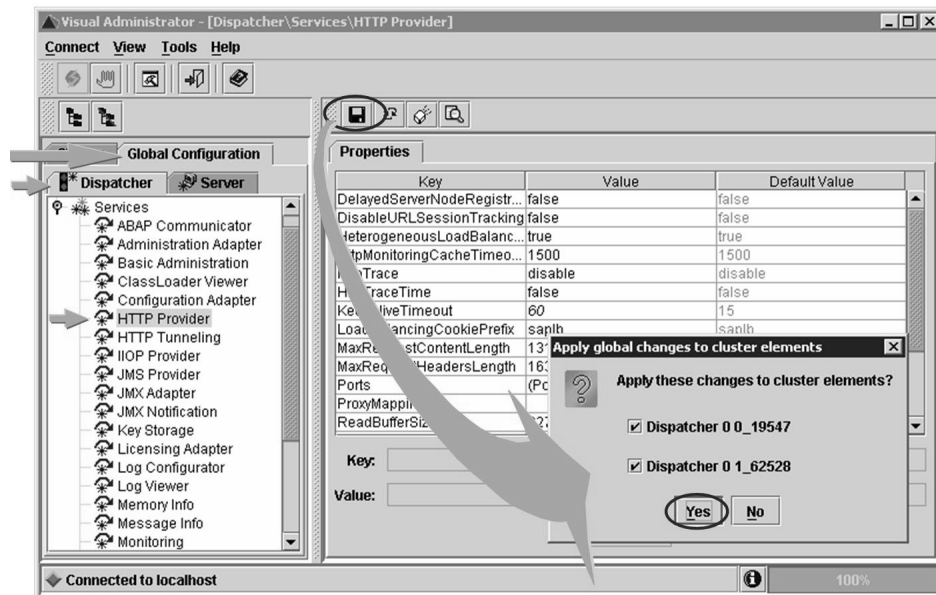


Figure 152: Global Settings 2b

In the Visual Administrator, we can check, as figure “Visual Administrator: Check Current (Local) Values 2” illustrate, the currently active (cluster) value. We see that our maintained value is now immediately active and that it is not necessary to restart

the SAP NetWeaver AS instance to activate the maintained value. A check using the Config Tool (see also figure “Config Tool: Check Settings 2”) shows that the settings have now been set locally.



Caution: For parameters maintained in this way, “global configurations without local maintenance” with the Visual Administrator or “global settings” with the Config Tool only take effect again if the local settings are removed with the Config Tool. It is, of course, necessary to restart the SAP NetWeaver AS instance, so that the changes from the Config Tool take effect.



Visual Administrator - [DEV\Dispatcher 0 1_62528\Services\HTTP Provider]

Key	Value
DelayedServerNodeRegistration	false
DisableURLSessionTracking	false
HeterogeneousLoadBalancing	true
HttpMonitoringCacheTimeoutMillis	1500
HttpTrace	disable
HttpTraceTime	false
KeepAliveTimeout	60
LoadBalancingCookiePrefix	saplo
MaxRequestContentLength	131072
MaxRequestHeadersLength	16384
Ports	(Port 50100,Type:http)(Port 50101,Type:ssl)
ProxyMappings	
ReadBufferSize	32768

Visual Administrator - [DEV\Dispatcher 0 0_19547\Services\HTTP Provider]

Key	Value
DelayedServerNodeRegistration	false
DisableURLSessionTracking	false
HeterogeneousLoadBalancing	true
HttpMonitoringCacheTimeoutMillis	1500
HttpTrace	disable
HttpTraceTime	false
KeepAliveTimeout	60
LoadBalancingCookiePrefix	saplo
MaxRequestContentLength	131072
MaxRequestHeadersLength	16384
Ports	(Port 50000,Type:http)(Port 50001,Type:ssl)
ProxyMappings	
ReadBufferSize	32768

Dialog instance: global configuration with local maintenance in Visual Admin.

Central instance: global configuration with local maintenance in Visual Admin.

Figure 153: Visual Administrator: Check Current (Local) Values 2

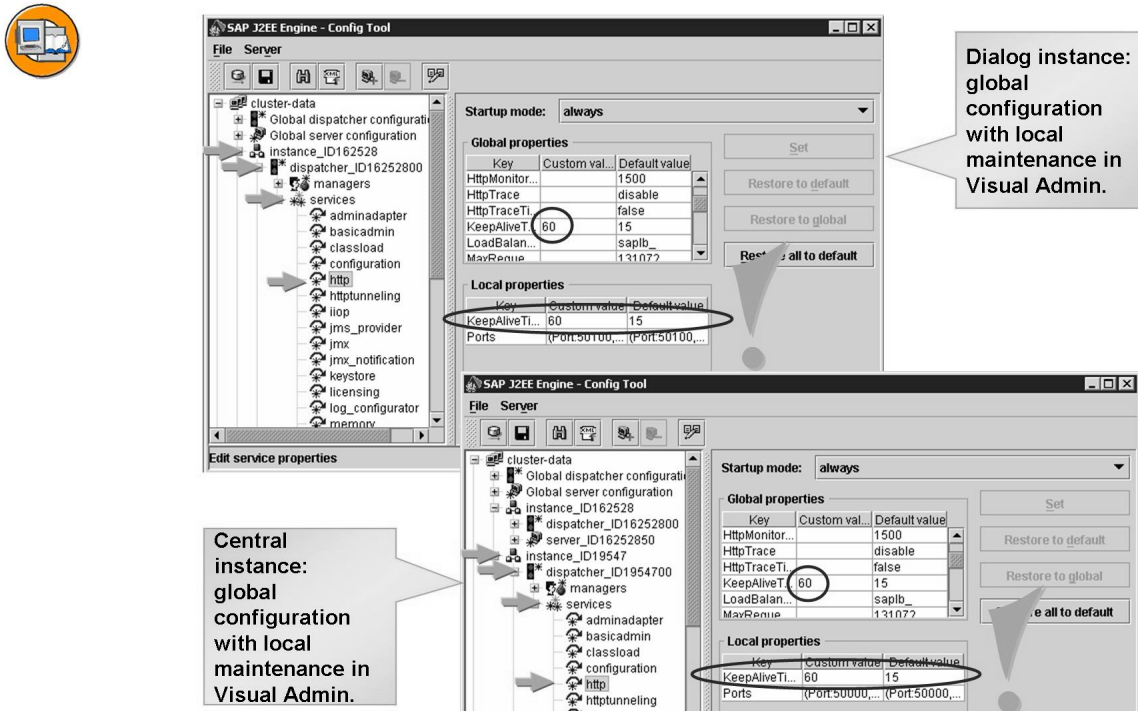


Figure 154: Config Tool: Check Settings 2

Visual Administrator: Cluster (Local) Configuration

Cluster settings are maintained using the *Cluster* tab page. You can select all active nodes here. We now maintain the parameter *KeepAliveTimeout* of the HTTP Provider service of a dispatcher node again. The procedure is only marginally different from the procedure described in the previous section. One difference is the different tab page on which you are maintaining the value. Another is that no additional popup appears after you have saved the value. See also the figures “Visual Administrator: Local Settings 1a” and “Visual Administrator: Local Settings 1b”. Changes that are made take effect immediately, as long as no message appears to say otherwise. Messages saying otherwise could, for example, be that the service must first be restarted, and so on. As shown in the figure “Config Tool: Check Local Settings” to check that the modified value is entered as a local custom value as expected. If no local setting was maintained for the parameter before your maintenance, a local setting is now created.

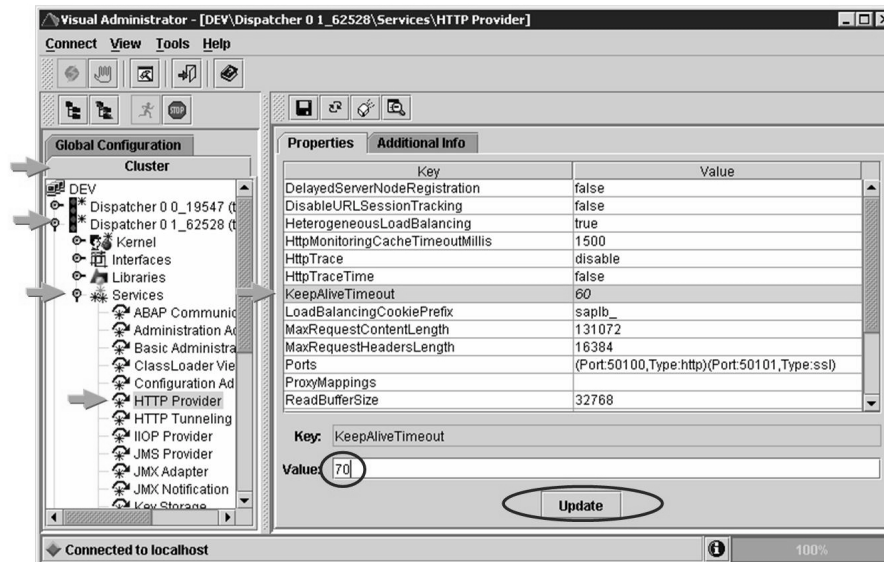


Figure 155: Visual Administrator: Local Settings 1a.

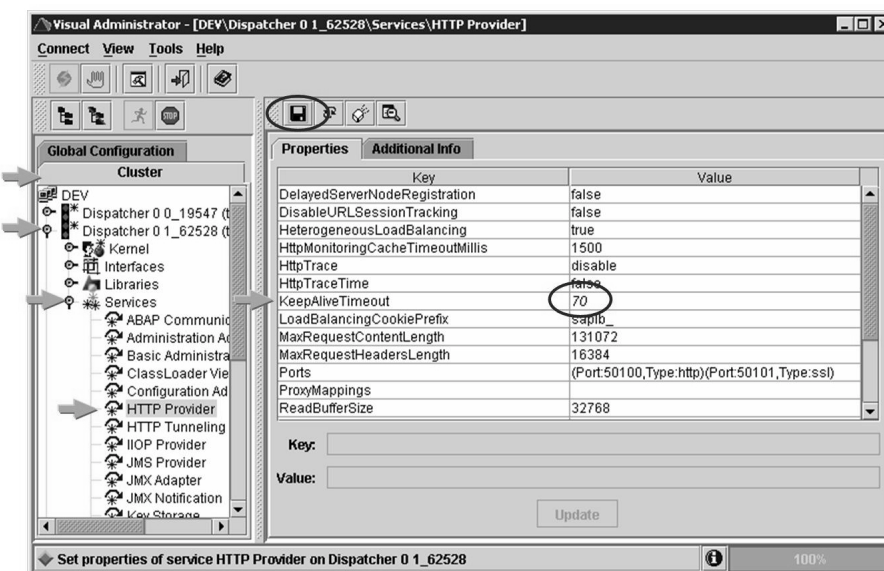


Figure 156: Visual Administrator: Local Settings 1b

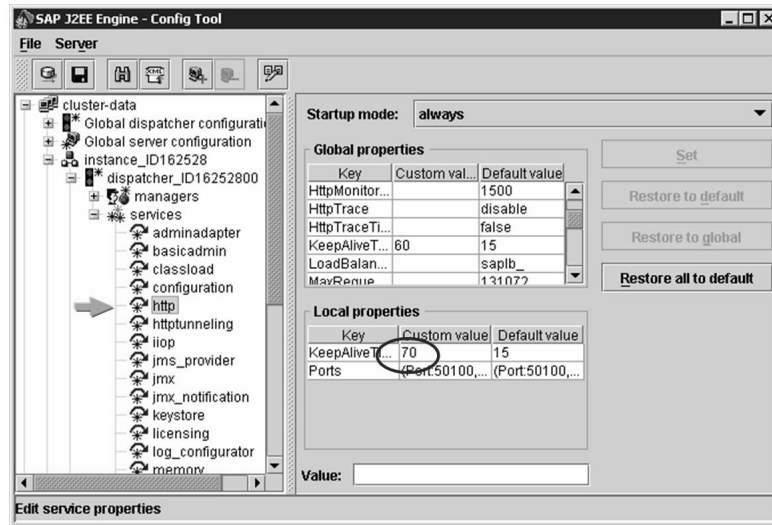


Figure 157: Config Tool: Check Local Settings

Exercise 20: General Configuration of the SAP NetWeaver AS Java Cluster with the Visual Administrator

Exercise Objectives

After completing this exercise, you will be able to:

- Maintain parameters for the SAP NetWeaver AS Java with the Visual Administrator

Business Example

The company management have decided to implement an SAP product that uses SAP NetWeaver AS Java. Many components of SAP NetWeaver, for example, use the Java application server. The following knowledge is required so that the IT department can ensure the smooth processing of business processes.

Task 1: Starting SAP NetWeaver AS Java

Start the SAP NetWeaver AS Java (if it has not already been started).

1. Start all the SAP NetWeaver AS Java instances.

Task 2: Logon with the Visual Administrator

Log on to your SAP NetWeaver AS Java with the Visual Administrator. You have already learnt the details for this in the lesson “Administration and Configuration Tools”.

Task 3: Maintaining Local Settings

Local Settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the HTTP Provider service (for the dispatcher) **locally** to the value “75” for your (Java) dialog instance.

Task 4: Maintaining Global Settings

Optional: Global Settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the HTTP Provider service (for dispatchers) to the value “90” **globally** for all (Java) instances.

Continued on next page

Task 5: Checking the Settings

Optional: Check the values set in the database with the Config Tool.

1. Check the globally set values for the KeepAliveTimeout parameter.
2. Check the locally set values for the KeepAliveTimeout parameter for all instances.

Solution 20: General Configuration of the SAP NetWeaver AS Java Cluster with the Visual Administrator

Task 1: Starting SAP NetWeaver AS Java

Start the SAP NetWeaver AS Java (if it has not already been started).

1. Start all the SAP NetWeaver AS Java instances.
 - a) Log on to the ABAP stack of the SAP system.
 - b) Start transaction SMICM and start all the instances of the Java cluster by choosing the following menu path: *Administration* → *J2EE Clusters (global)* → *Restart* → *Yes*.

Continued on next page

Task 2: Logon with the Visual Administrator

Log on to your SAP NetWeaver AS Java with the Visual Administrator. You have already learnt the details for this in the lesson “Administration and Configuration Tools”.

Task 3: Maintaining Local Settings

Local Settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the HTTP Provider service (for the dispatcher) **locally** to the value “75” for your (Java) dialog instance.
 - a) Select the **Cluster** tab page.
 - b) Expand the dispatcher for the dialog instance.
 - c) Expand the services of the dispatcher.
 - d) Select the HTTP Provider service.
 - e) Show the detailed information by choosing the *Show Details*.
 - f) Select the parameter *KeepAliveTimeout*.
 - g) Maintain the new value “75” in the *Value* field.
 - h) Choose the *Update* button.
 - i)



Hint: Remember to save your settings.

Continued on next page

Task 4: Maintaining Global Settings

Optional: Global Settings: Set *KeepAliveTimeout*.

1. Set the *KeepAliveTimeout* parameter of the HTTP Provider service (for dispatchers) to the value “90” **globally** for all (Java) instances.
 - a) Choose the **Global Configuration** tab page.
 - b) Now choose the *Dispatcher* tab page.
 - c) Expand the services of the dispatcher.
 - d) Select the HTTP Provider service.
 - e) Show the detailed information by choosing the *Show Details*.
 - f) Select the parameter *KeepAliveTimeout*.
 - g) Maintain the new value “90” in the *Value* field.
 - h) Choose the *Update* button.
 - i)



Hint: Remember to save your settings.

- j) Select the dispatcher nodes that are to be set locally with the value.

Task 5: Checking the Settings

Optional: Check the values set in the database with the Config Tool.

1. Check the globally set values for the *KeepAliveTimeout* parameter.
 - a) For the procedure, see the lesson: General Configuration of the J2EE cluster with the Config Tool.
2. Check the locally set values for the *KeepAliveTimeout* parameter for all instances.
 - a) For the procedure, see the lesson: General Configuration of the J2EE cluster with the Config Tool.



Lesson Summary

You should now be able to:

- Maintain settings for an SAP NetWeaver AS Java with the Visual Administrator

Lesson: Other Administration Tools

Lesson Overview

Other tools with which you can obtain an overview of the most important information for an SAP NetWeaver AS Java are introduced.



Lesson Objectives

After completing this lesson, you will be able to:

- Obtain an overview of the most important information for an SAP NetWeaver AS Java with the System Information and SAP NetWeaver Administrator or Telnet Console

Business Example

You need to obtain an overview of the most important settings of the SAP NetWeaver AS Java.

System Information

The *System Information* is used to display the most important administrative information of an SAP NetWeaver AS Java. *System Information* is called using a browser. It can therefore be called from any front end on which a browser is installed. It is called using the HTTP port of the SAP NetWeaver AS Java. On the start page of the SAP NetWeaver AS Java, there is a link to the *System Information*. Authentication as a system administrator is required to view this information. *System Information* displays general system information for the SAP NetWeaver AS Java and information about the individual instances of the SAP NetWeaver AS Java (see the figure “System Information”).



System DEV **SAP**

Find other systems in SLD...

Message Serv...	Enqueue Serv...	Database
Host: twdf1903	Host: twdf1903	Name: DEV
Port: 3602	Port: 3202	Host: twdf1903
		Type: SAP DB (KERNEL 7.6.01 BUILD 015-121-147-649)

Software Components

Name	Version	Applied	Licenses
sap.com/SAP-JEECOR	7.00 SP11 (1000.7.00.11.0.20070201154700)	20070511121327	Installation Number *** not yet assigned ***
sap.com/SAP-JEE	7.00 SP11 (1000.7.00.11.0.20070201154700)	20070511121259	System Number DEV Software Product Days Until Expiry

Download system configuration

Instance DVEBMGS00 All processes running

Host: twdf... OS: Windows 2003 (amd64) 5.2

dispatcher Running

VM	system properties...	Cluster
PID: 11200		Node ID: 1954700
Name: Java HotSpot(TM) 64-Bit Server VM		Kernel Version: 7.00 PatchLevel
Vendor: Sun Microsystems Inc.		HTTP Port: 50000
Version: 1.4.2_13-b06		HTTPS Port: 50001
VM Parameters		P4 Port: 50004
		Telnet Port: 50008

server0 Running

VM	system properties...	Cluster	Debug
PID: 11208		Node ID: 1954750	Debug Port: 50021
Name: Java HotSpot(TM) 64-Bit Server VM		Kernel Version: 7.00 PatchLevel	Debug Mode: false
Vendor: Sun Microsystems Inc.			Load Balance Restricted: false
Version: 1.4.2_13-b06			
VM Parameters			

SDM Running

VM
PID: 11216
SDM Port: 50010

Figure 158: System Information

The general system information for the SAP NetWeaver AS Java includes the host on which the message and enqueue servers are running, and at which ports they are listening. Message and enqueue servers form the SAP Central Services. Therefore, the host name provides information about the host on which the Central Service instance is running. The database type and the host on which the DB is running are also displayed. The Support Package level of the SAP NetWeaver AS Java is displayed in the lower area of the general system information. System Information also shows the license data and the days remaining until the license expires.



Instance DVEBMGS00 All processes running

Host: twdf... OS: Windows 2003 (amd64) 5.2

dispatcher Running

VM	system properties...	Cluster
PID: 11200		Node ID: 1954700
Name: Java HotSpot(TM) 64-Bit Server VM		Kernel Version: 7.00 PatchLevel
Vendor: Sun Microsystems Inc.		HTTP Port: 50000
Version: 1.4.2_13-b06		HTTPS Port: 50001
VM Parameters		P4 Port: 50004
		Telnet Port: 50008

server0 Running

VM	system properties...	Cluster	Debug
PID: 11208		Node ID: 1954750	Debug Port: 50021
Name: Java HotSpot(TM) 64-Bit Server VM		Kernel Version: 7.00 PatchLevel	Debug Mode: false
Vendor: Sun Microsystems Inc.			Load Balance Restricted: false
Version: 1.4.2_13-b06			
VM Parameters			

SDM Running

VM
PID: 11216
SDM Port: 50018

Figure 159: System Information: Instances

The figure “System Information: Instances” shows the information that is displayed for an instance. You can see from the figure that this is the Java central instance, since the SDM belongs to this instance. The host on which the instance is running is displayed. All nodes (dispatcher or server processes) that belong to the instance are also displayed with additional detail and status information. Take the dispatcher node as an example. PID shows the process ID under which the process is running on the operating system. The ports are also displayed; here, for example, the Telnet port 50008, through which you can log on to the SAP NetWeaver AS Java by Telnet. You can use the link *VM Parameters* to display information about the VM parameters with which this node of the SAP NetWeaver AS Java instance was started.



Instance D01

Starting the processes

Host: twdf1903 OS: null (null) null

dispatcher

Waiting for start

VM

PID: 10260

Name: null

Vendor: null

Version: null

Cluster

Node ID: null

Kernel Version: null

HTTP Port: null

HTTPS Port: null

P4 Port: null

Telnet Port: null

server0

Waiting for start

VM

PID: 10252

Name: null

Vendor: null

Version: null

Cluster

Node ID: null

Kernel Version: null

Debug

Debug Port: 50121

Debug Mode: false

Load Balance Restricted: false

Instance D01

Starting the processes

Host: twdf1903 OS: null (null) null

dispatcher

Starting

VM

PID: 10856

Name: null

Vendor: null

Version: null

Cluster

Node ID: 16252800

Kernel Version: null

HTTP Port: null

HTTPS Port: null

P4 Port: null

Telnet Port: null

server0

Starting

VM

PID: 12212

Name: null

Vendor: null

Version: null

Cluster

Node ID: 16252850

Kernel Version: null

Debug

Debug Port: 50121

Debug Mode: false

Load Balance Restricted: false

Figure 160: System Information: Start Process 1

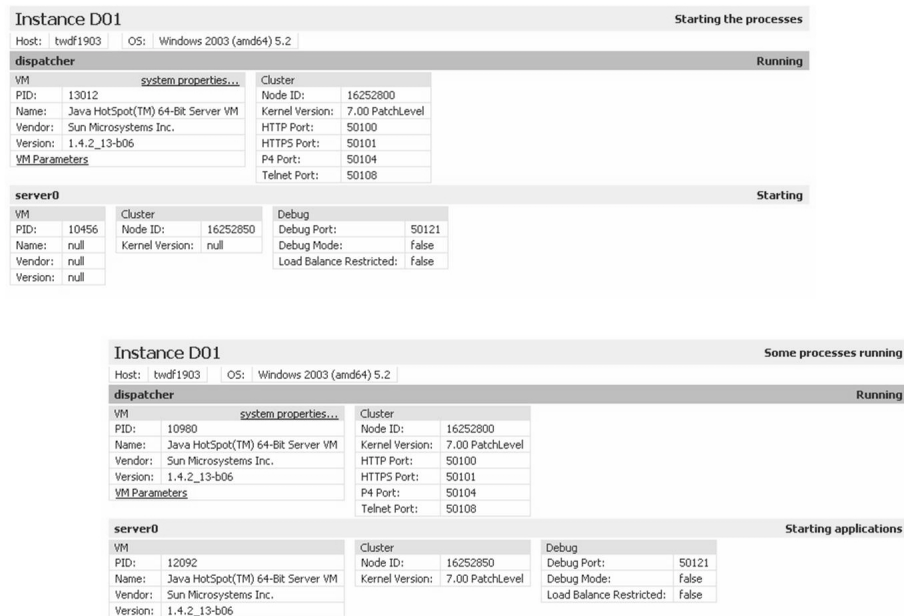


Figure 161: System Information: Start Process 2

The figures “System Information: Start Process 1+2” show how the start process of an instance is displayed by System Information. It is, of course, necessary to use the Refresh button of the browser for this display. The dispatcher node runs through the statuses “Initial” and “Starting”, before it reaches the status “Running”. A server node, on the other hand, runs through the statuses “Initial”, “Starting”, and “Starting applications” before it reaches the status “Running”.

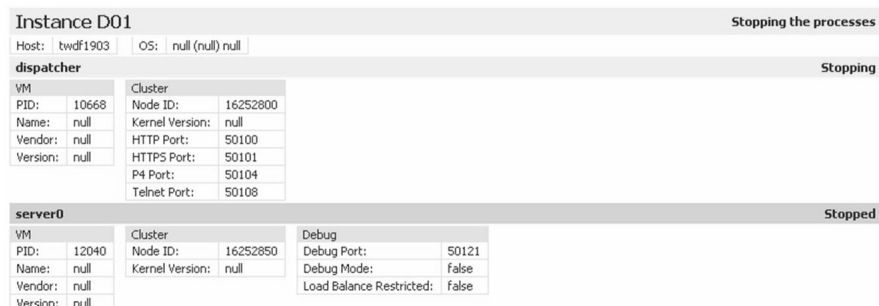


Figure 162: System Information: Stop Process

The figure “System Information: Stop Process” shows the statuses that a node runs through when stopping (“Stopping”, “Stopped”). The figure “System Information: Instance Problems”, on the other hand, shows that it was not possible to start server

process 1. It is easy to see in System Information whether an instance in the cluster has a problem, and facilitates troubleshooting, since in this case, for example, it also shows which server process could not be started.



Instance D01				Some processes running	
Host:	twdf1903	OS:	Windows 2003 (amd64) 5.2		
dispatcher				Running	
VM	system properties...	Cluster			
PID:	13872	Node ID:	16252800		
Name:	Java HotSpot(TM) 64-Bit Server VM	Kernel Version:	7.00 PatchLevel		
Vendor:	Sun Microsystems Inc.	HTTP Port:	50100		
Version:	1.4.2_13-b06	HTTPS Port:	50101		
VM Parameters		P4 Port:	50104		
		Telnet Port:	50108		
server0				Running	
VM	system properties...	Cluster		Debug	
PID:	7156	Node ID:	16252850	Debug Port:	50121
Name:	Java HotSpot(TM) 64-Bit Server VM	Kernel Version:	7.00 PatchLevel	Debug Mode:	false
Vendor:	Sun Microsystems Inc.			Load Balance Restricted:	false
Version:	1.4.2_13-b06				
VM Parameters					
server1				Stopped	
VM		Cluster		Debug	
PID:		Node ID:	16252851	Debug Port:	50126
Name:	null	Kernel Version:	null	Debug Mode:	false
Vendor:	null			Load Balance Restricted:	false
Version:	null				

Figure 163: System Information: Instance Problems

Telnet Console of SAP NetWeaver AS Java

With Telnet, you can log on to SAP NetWeaver AS Java and perform “simple” administrative tasks. The easiest way to see the options that Telnet offers here is by logging on to an SAP NetWeaver AS Java instance and, as in the figure “Telnet: . man” execute the *man* command. All currently possible commands are displayed. If you enter the command *man add*, information about calling the *add* command is displayed. You can use the *add* command to include additional command groups and call them using Telnet.



```

Telnet Administration
SAP J2EE Engine v7.00

Login: adm200
Password:
>man
***** ADMIN *****
CIC          GC          INFO          LISTMP        LISTSP
LL           LLR         LP           LSC           LSI
LSL          LSM         LSP          LSS           LSTOL
SETMP        SETSP       SHUTDOWN    STARTSERVICE STOPSERVICE
VERSION

***** SYSTEM *****
ADD          CAT          CLS          ECHO          EXIT
GET          GREP         KILL         LOADENU       MAN
MORE         PS           QUIT        REMOVE        RUN
SAVEENU      SET

***** TELNET *****
JUMP
>

```

Figure 164: Telnet: man Command

One of the most important commands is *lsc*. This stands for “List Cluster” and displays all active nodes in the cluster. See also the figure “Telnet: lsc Command”.



```

>lsc

```

NAME	ID	GROUP	HOST	PORT	TYPE	STATE
Dispatcher 0 0_19547	1954700	19547	10.22.16.204	N/A	dispatcher	running
Dispatcher 0 1_62528	16252800	62528	10.22.16.204	N/A	dispatcher	running
Server 0 0_19547	1954750	19547	10.22.16.204	50020	server	running
Server 0 1_62528	16252850	62528	10.22.16.204	50120	server	running

Figure 165: Telnet: lsc Command

You can use the *jump* command to switch to another node (it does not matter whether it is a dispatcher or a server). To do this, you must specify the node name. Please refer to the figure “Telnet: jump Command”. You can use the *shutdown* command to stop a node in the cluster. With *shutdown* as with *jump*, you need to specify the node name.



```
>jump 16252850
You jumped on node 16252850.
>lsc
```

	NAME	ID	GROUP	HOST	PORT	TYPE	STATE
Server 0 1	62528	16252850	62528	10.22.16.204	50120	server	running
Dispatcher 0 0	19547	1954700	19547	10.22.16.204	N/A	dispatcher	running
Dispatcher 0 1	62528	16252800	62528	10.22.16.204	N/A	dispatcher	running
Server 0 0	19547	1954750	19547	10.22.16.204	50020	server	running

Figure 166: Telnet: jump Command

SAP NetWeaver Administrator (SAP NWA): Administration and Configuration

SAP NetWeaver Administrator (SAP NWA) is a new tool for administration and monitoring. The SAP NWA combines the most important administration and monitoring tools for Java in a new, browser-based user interface.

It provides functions in a wide variety of areas (monitoring, configuration, administration, performance analysis, ...). In the administration area, you will find plug-ins for user administration, for applications and for systems.



Hint: The SAP NWA can only show data in the configuration area at the moment. The display function will be expanded with a change function in following releases/support packs.

You can call the SAP NWA using the following URL: <http://<Rechner>:<Port>/nwa>, e.g. <http://P12345:50000/nwa>

Administration in the NWA:

The “Administration” area in the NWA contains areas for system and application administration as well as for user management.

- Starting and stopping instances (see figure: SAP NWA: System Administration).
- Starting and stopping cluster elements (dispatcher, server, SDM)
- Starting and stopping applications

If it is necessary to start or stop instances or cluster nodes then authentication via the operating system user `<sid>adm` is required.

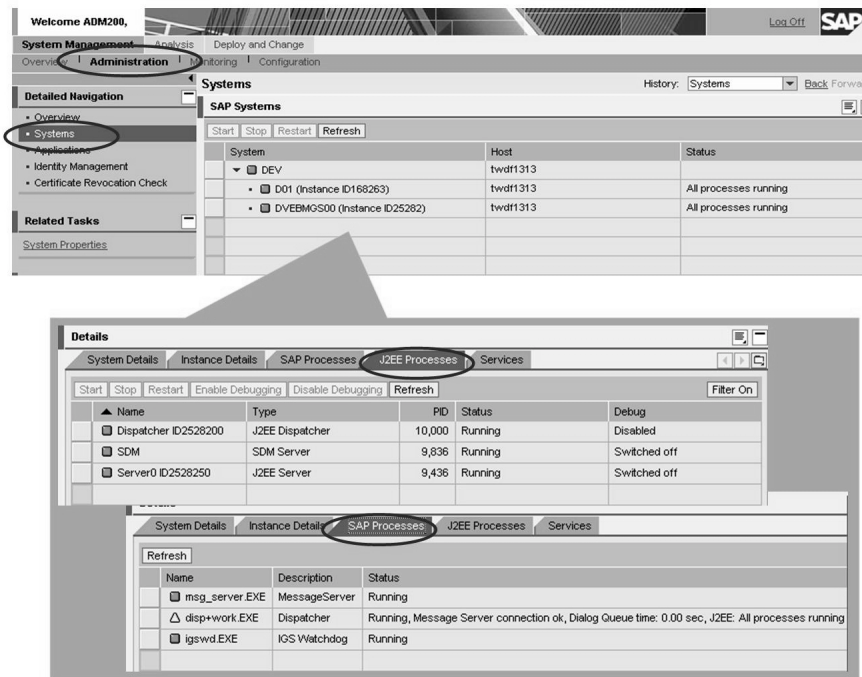


Figure 167: SAP NWA: System Administration

Configuration in the NWA

The configuration area contains, among other things, a display of the system properties such as the properties of the JVM and the service and manager configuration. By way of an example, the next two figures depict the display of the JVM properties (SAP NWA: JVM Properties) and the display of the properties of the http service (SAP NWA: Properties of Services).



Welcome ADM200, Log Off SAP

System Management Analysis Deploy and Change

Overview Administration Monitoring Configuration

System Properties History: System Properties Back Forward

Configurations

Refresh

Name
DEV
Global dispatcher configuration
Global server configuration
Instance ID168263
Dispatcher ID16826300
Server0 ID16826350
Instance ID25282

Details

JVM Properties Kernel Services

Display All Values Filter On

Name	Value
debuggable	no
directory	dispatcher
java.class.path	.bin\boot\boot.jar;.bin\system\bytecode.jar;.
java.main.class	com.sap.engine.boot.Start
java.main.class.parameters	
java.max.heap.size	203
java.parameters	-Djava.security.policy=-:java.policy -Djava.security.egd=file:/dev/urandom -Xms203M -XX:NewSize=68M -XX:MaxNewSize=68M -XX:+DisableExplicitGC
java.path	G:\V2SDK_64_BW_AS_JAVA
os.bit.length	64
os.name	ntintel

Row 1 of 14

Figure 168: SAP NWA: JVM Properties

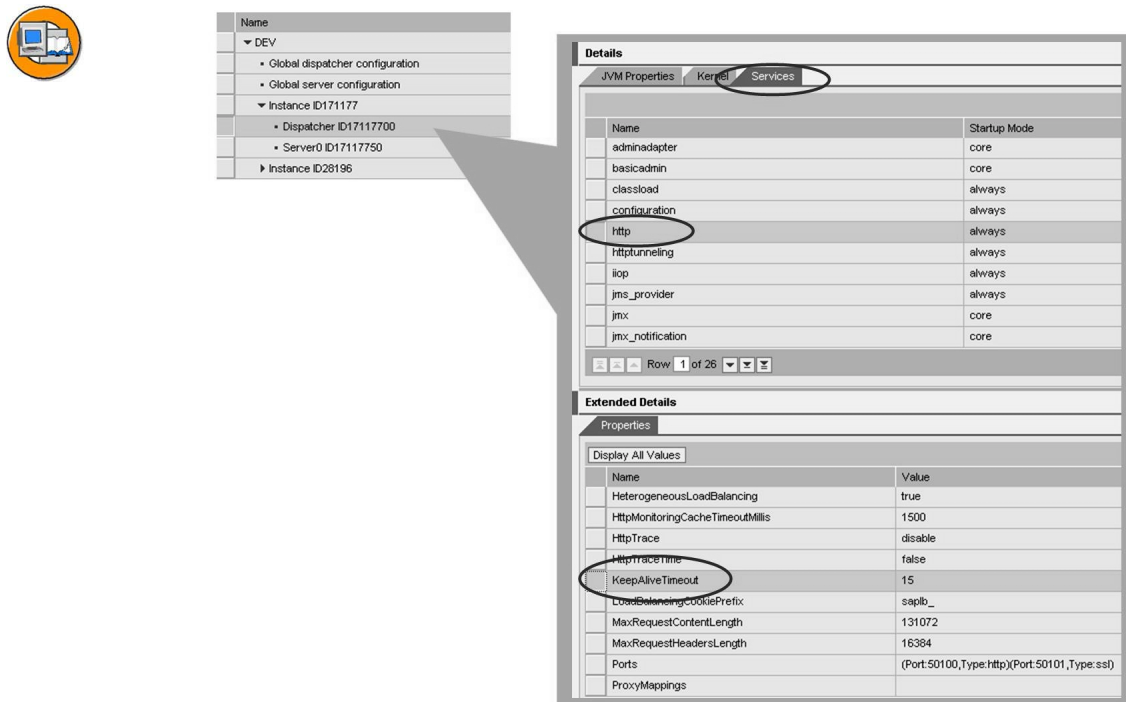


Figure 169: SAP NWA: Properties of Services

J2EE Configuration Browser

Display of the information in the same way as in the Config Tool's Offline or Editor mode: *J2EE Configuration Browser* (see figure: SAP NWA: J2EE Configuration Browser).

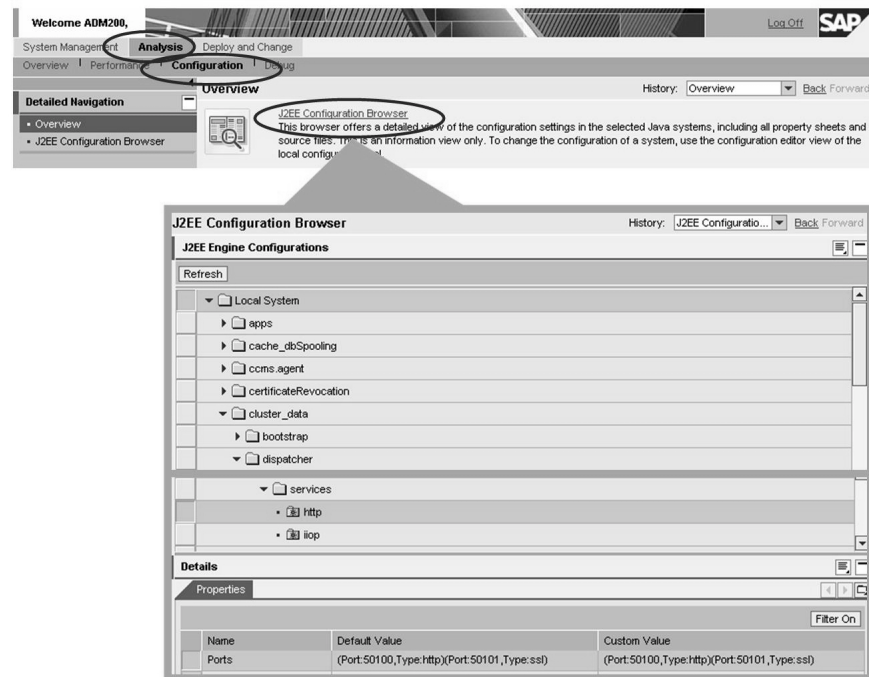


Figure 170: SAP NWA: J2EE Configuration Browser

The other functions of the NetWeaver Administrator such as monitoring and user management are illustrated in the units devoted to these subjects.

Exercise 21: Other Administration Tools

Exercise Objectives

After completing this exercise, you will be able to:

- Use System Information to obtain an overview of basic administration information for an SAP NetWeaver AS Java

Business Example

The company management have decided to implement an SAP product that uses SAP NetWeaver AS Java. Many components of SAP NetWeaver, for example, use the Java application server. The following knowledge is required so that the IT department can ensure the smooth processing of business processes.

Task: Checking SAP NetWeaver AS Java Using “System Information”

Optional: Use System Information to check the most important information for your SAP NetWeaver AS Java.

1. Call System Information.
2. Check the SP status of your SAP Web AS Java.
3. Are all of your instances and their nodes running?
4. Display the VM parameters for the dispatcher of the dialog instance.

Solution 21: Other Administration Tools

Task: Checking SAP NetWeaver AS Java Using “System Information”

Optional: Use System Information to check the most important information for your SAP NetWeaver AS Java.

1. Call System Information.
 - a) Call the following URL in the browser:
For the DEV system: `http://<host name>.wdf.sap.corp:50000`
For the QAS system: `http://<host name>.wdf.sap.corp:51000`
 - b) In the browser window, select the link *System Information*.
 - c) Log on with your Java user and password (such as ADM200-##).
2. Check the SP status of your SAP Web AS Java.
 - a) Under *Software Components*, you can find entries of the type “7.00 SP...”. .
3. Are all of your instances and their nodes running?
 - a) You should see all instances and their nodes.
4. Display the VM parameters for the dispatcher of the dialog instance.
 - a) You can easily identify the dialog instance using the instance number and the host name.
 - b) The dialog instance in the DEV system is “D01”. In the QAS system, it is “D11”.
 - c) Display the parameters by choosing the link *VM Parameters*. Xmx and Xms should be identical, if you have performed the relevant exercise with the Config Tool.



Lesson Summary

You should now be able to:

- Obtain an overview of the most important information for an SAP NetWeaver AS Java with the System Information and SAP NetWeaver Administrator or Telnet Console

Lesson: Selected Configurations

Lesson Overview

In this lesson, you learn how to add additional server processes to an instance. Other services and managers are also presented, especially those that are responsible for communication with the Central Services instance.



Lesson Objectives

After completing this lesson, you will be able to:

- Make selected configurations

Business Example

After the installation of an SAP NetWeaver Application Server, fundamental settings need to be made for the elements of the Java cluster. Various services can be configured with an SAP NetWeaver Application Server.

Creating Server Processes

An instance must have at least one server process. However, with sufficiently large hardware additional server processes can be added. An instance can consist of a maximum of one dispatcher and 16 server processes. You add an additional server process to an instance with the **Config Tool**. If you select an instance or a node of an instance, the *Add server* button, with which you can add an additional server process, becomes active. See the figure “Server Processes: Add”

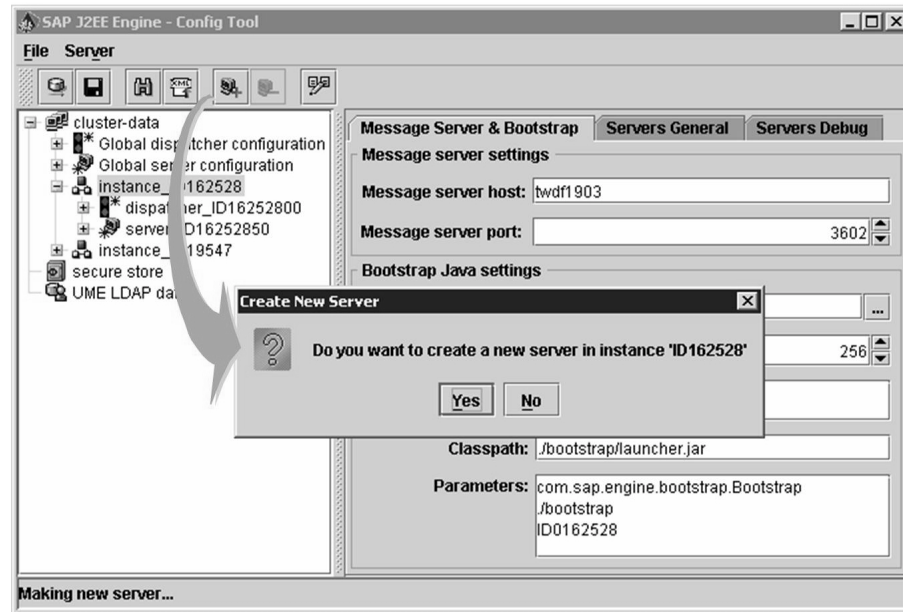


Figure 171: Server Processes: Add:

Before the new server process is added, a popup appears with a confirmation prompt. You can check again whether you are adding the server process to the correct instance ID. After you have confirmed the pop-up, a confirmation pop-up is displayed (figure “New Server Process: Information”). Among other things, this pop-up contains information about the *join-Port* (this is the port which the server process uses to exchange data in the cluster) and the path in the operating system directory in which the data for the new server process is stored. The *join-Port* is usually composed of $5 \times \text{instance number} + 20 \times 5 \times (\text{number of the server process})$. In the example, this is $5 \times 21 + 20 \times 5 \times 1 = 52125$.

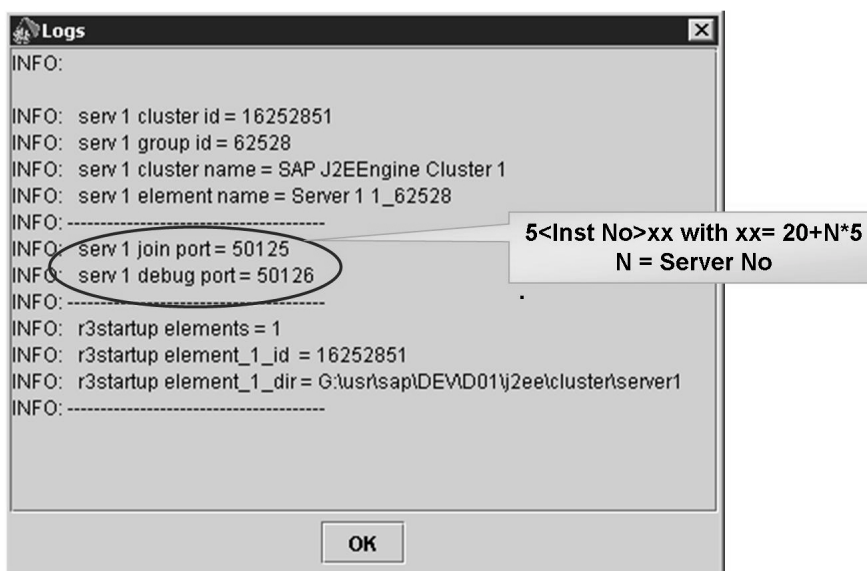


Figure 172: New Server Process: Information

Now that you have added an additional server process to the instance, you should check the VM parameters, and adjust them if necessary.

Central Services and Managers

This section deals with some central services and managers. The SAP Central Services, which are covered in the following subsection, are an important part.

Central Services and their Connection to the Other Instances

The Central Services consist of the message server and the enqueue server. We will first consider the interaction with the message server. As shown in the “Message Server” figure, each node (dispatcher and servers) of each instance is connected to the message server.

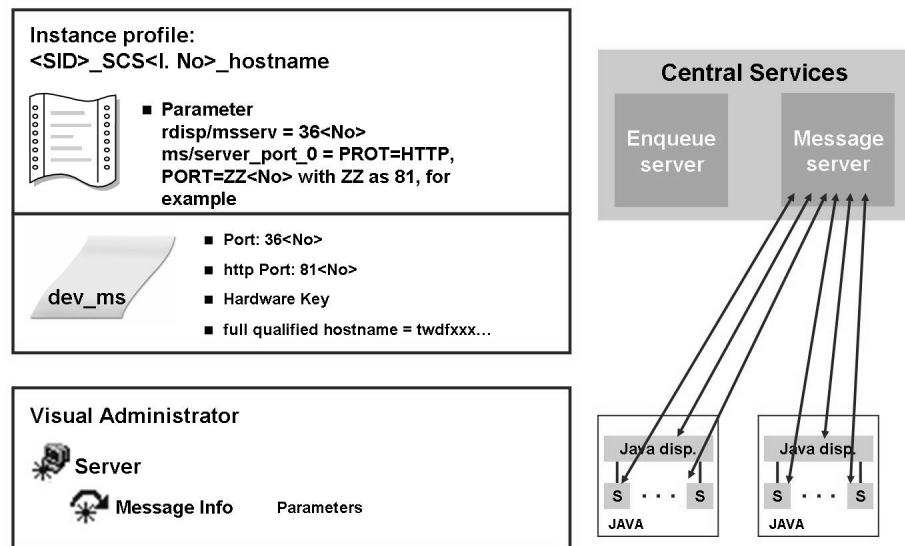


Figure 173: Message Server

When the Central Services are starting, the profile parameters for the message server are read. The internal communication port for the message server is defined using the parameter *rdisp/msserv*. This is usually the port number 36<instance number>. The HTTP(S) port of the message server is specified with the profile parameter *ms/server_port_0*. The instance number is also often used for the last two digits in this case. The HTTP(S) port is required by the Web dispatcher, among other things. You can obtain the port numbers with which the message server was started from the developer trace (*dev_ms*). The developer trace also contains the hardware key (which is required for requesting a license) and the host on which the message server is running. You can use the **Visual Administrator** to display the parameters with which the message server is running. You can find the parameters in the “Message Info” service of a server process on the tab page *Runtime*. You need to select the *Parameters* tab page in the lower part of the screen here. You may also have to start the service. However, the parameters can only be changed in the above instance profile.

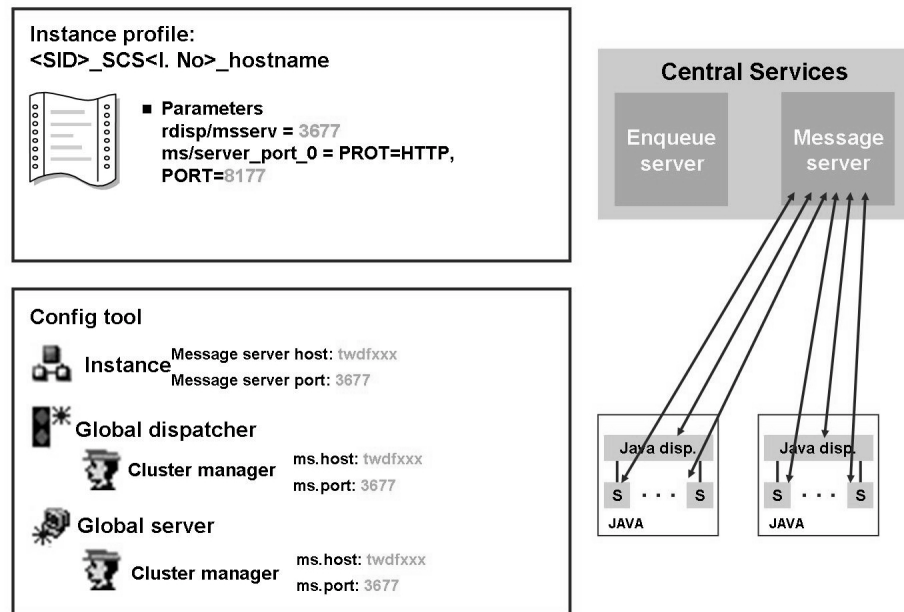


Figure 174: Message Server: Changing Ports/Hosts

In the figure “Message Server: Changing Ports/Hosts” you can see which areas need to be adjusted if the port or the name of the host on which the message server is running changes. Ports are, of course, changed in the instance profile of the Central Services instance. In the example from the figure, the internal communication port 3677 and the http port 8177 have been set, and the new host is called twdfxxx. You need to use the **Config Tool** to set these values. For each instance, you need to maintain the corresponding values for the *Message server host* and *Message server port* on the *Message Server and Bootstrap* tab page. The connection details for dispatchers or servers to the message server are maintained in the Cluster Manager. As shown in the figure, the parameters *ms.host* and *ms.port* are maintained, respectively, in the Cluster Manager with **Global Dispatcher** and the **Global Server**. After you have maintained all of the parameters, you can start the instances.



Caution: If these parameters are overridden locally, you also need to enter the correct values for *ms.host* and *ms.port* at these locations.

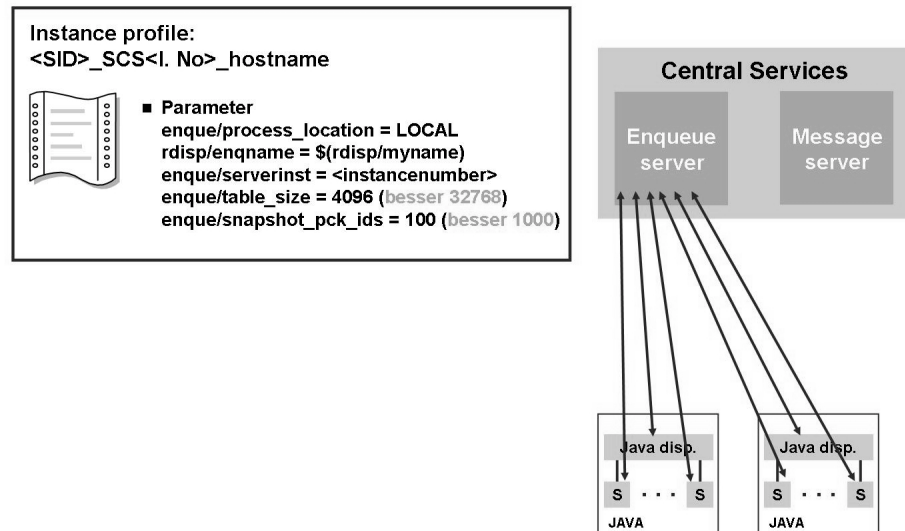


Figure 175: Enqueue Server

Now that you have seen the settings for the message server, we will consider the enqueue server. The “Enqueue Server” figure shows that each dispatcher/server node of each instance also has a connection to the enqueue server. The figure shows the profile parameters that are relevant for the enqueue server in the instance profile of the Central Services instance. The port is usually set to 32<instance number> and the instance number is defined by means of the parameter *enque/serverinst*. The parameter *enque/encni/port* can be used to select the port for the enqueue server explicitly (including outside of the range 32xx). The default values of the parameters *enque/table_size* and *enque/snapshot_pck_ids* are minimum values and should be adjusted to the values shown in the figure. These are, however, not maximum values, and it may be necessary to further increase the values.

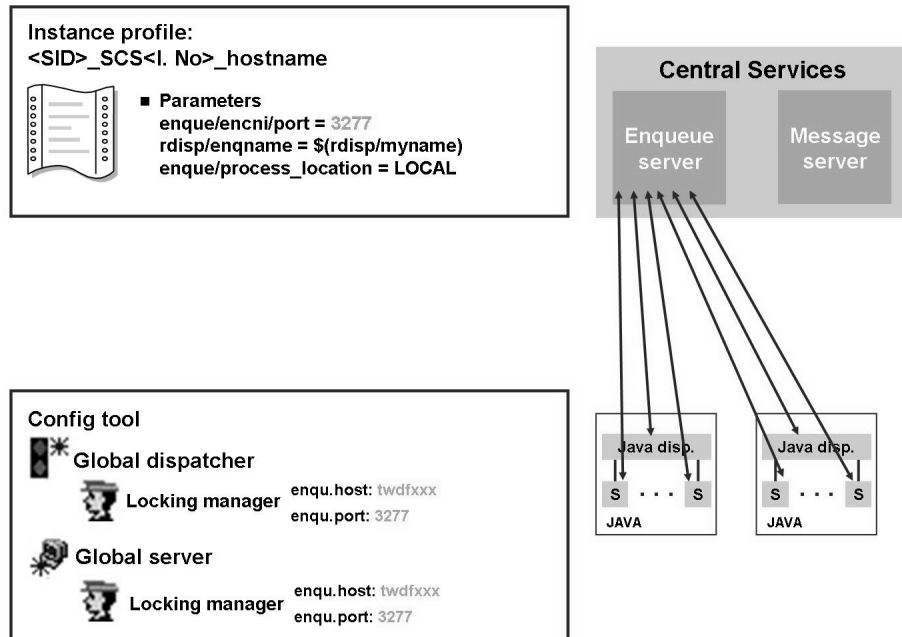


Figure 176: Enqueue Server: Changing Ports/Hosts

In the figure “Enqueue Server: Changing Ports/Hosts” you can see which areas need to be adjusted if the port or host of the enqueue server changes. Ports are, of course, changed in the instance profile of the Central Services instance. In the example from the figure, the port 3277 has been set, and the new host is called twdfxxx. You need to use the **Config Tool** to set these values. The connection details for dispatchers or servers to the message server are maintained in the Locking Manager. As shown in the figure, the parameters *enqu.host* and *enqu.port* are maintained, respectively, in the Locking Manager with **Global Dispatcher** and the **Global Server**. After you have maintained all of the parameters, you can start the instances.



Caution: If these parameters are overridden locally, you also need to enter the correct values for *enqu.host* and *enqu.port* at these locations.

Other Selected Managers and Services

The figure “Other Managers and Services” shows other parameters of managers and services. Dispatcher and server processes use threads to process tasks. Threads are dynamically generated and closed by the processes. You can use parameters to specify, among other things, the maximum and minimum number of threads. The Thread Managers of dispatchers and servers manage the threads. The Application Thread Manager only exists for the server processes, and the threads that it manages are reserved for applications. The threads managed by the Thread Managers are

responsible for processing internal SAP NetWeaver AS Java requests. In this way, it is possible to make the settings for the application and the actual SAP NetWeaver AS Java instance individually. You can use the *Ports* parameter of the HTTP Provider service of dispatchers to change the HTTP port. After a standard installation, the HTTP port is set to 5<Instance number>00.

You will usually change parameters because corresponding SAP Notes prompt you to do so, or relevant Tuning Guides advise it. Discussing tuning measures here would go beyond the scope of this lesson. For more information about the topic of tuning, see the SAP Developer Network (SDN): www.sdn.sap.com. Use the search string *+Tuning +J2EE* to obtain additional information.

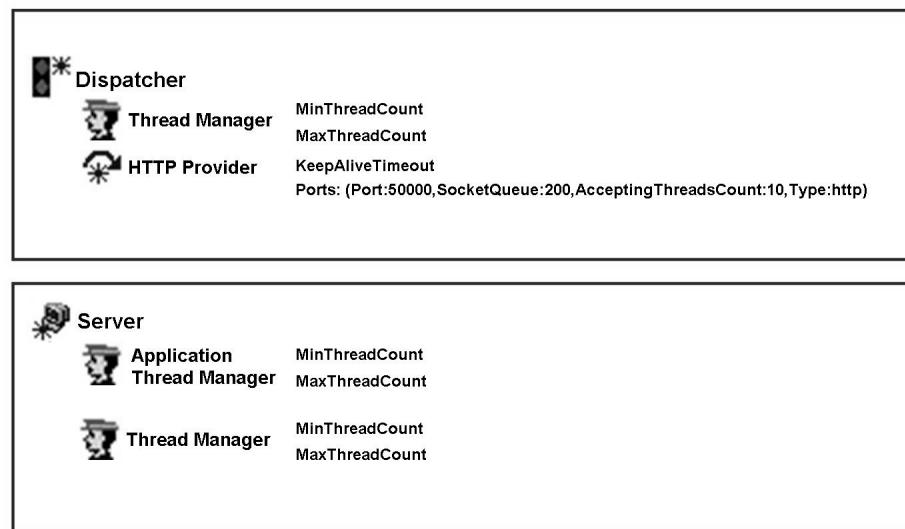


Figure 177: Other Managers and Services

Exercise 22: Selected Configuration

Exercise Objectives

After completing this exercise, you will be able to:

- Add additional server processes to an instance

Business Example

The company management have decided to implement an SAP product that uses SAP NetWeaver AS Java. Many components of SAP NetWeaver, for example, use the Java application server. The following knowledge is required so that the IT department can ensure the smooth processing of business processes.

Task 1: Adding a Server Process

Add an additional server process for your Java dialog instance.

1. Use the System Information to determine the ID of your Java dialog instance.
2. Stop all dialog instances of SAP NetWeaver AS Java.
3. In the Config Tool, select the node of your dialog instance and create an additional server process.
4. Start your dialog instance.

Task 2: Maintaining the Central Services

Optional: Maintaining the Central Services instance

1. Stop the Central Services on your system using the Microsoft Management Console (MMC).
2. Maintain the profile parameters *enqueue/table_size* to “32768” and *enqueue/snapshot_pck_ids* to “1000”.
3. Start the Central Services instance of your system using the Microsoft Management Console (MMC).

Solution 22: Selected Configuration

Task 1: Adding a Server Process

Add an additional server process for your Java dialog instance.

1. Use the System Information to determine the ID of your Java dialog instance.
 - a) The instance name ends with the instance number. For the DEV systems, this is “01”, and for the QAS systems, it is “11”. The instance ID can be derived from the dispatcher's node ID. The dispatcher's node ID is the instance ID followed by 00. You can find the node ID under *Dispatcher* in the field *Node-ID*.
2. Stop all dialog instances of SAP NetWeaver AS Java.
 - a) Log on to the ABAP stack of the SAP system.
 - b) Use transaction SM51 to go to a dialog instance .
 - c) Start transaction SMICM and stop the associated Java instance by choosing the following menu path: *Administration* → *J2EE Server (local)* → *Send Hard Shutdown* → *Without Restart*.
 - d) Follow this procedure for all other dialog instance.
3. In the Config Tool, select the node of your dialog instance and create an additional server process.
 - a) Choose the instance with the ID found under 2. (or one of its nodes).
 - b) Choose the *Add server* button, and confirm all subsequent popups.
 - c)



Hint: Remember to save the settings.

4. Start your dialog instance.
 - a) Log on to the ABAP stack of the SAP system.
 - b) Use transaction SM51 to go to a dialog instance .
 - c) Start transaction SMICM and start the associated Java instance by choosing the following menu path: *Administration* → *J2EE Server (local)* → *Restart* → *Yes*.

Continued on next page

Task 2: Maintaining the Central Services

Optional: Maintaining the Central Services instance

1. Stop the Central Services on your system using the Microsoft Management Console (MMC).
 - a) Select your Central Services instance in the MMC.
 - b) Right-click on *stop*
2. Maintain the profile parameters *enqueue/table_size* to “32768” and *enqueue/snapshot_pck_ids* to “1000”.
 - a) Switch to the profile directory of your system, *G:\usr\sap\<SID>\SYS\profile*.
 - b) Create a backup copy of the *profile* directory.
 - c) Maintain the above specified parameters in the profile DEV_SCS02_<host name> for the DEV system or QAS_SCS12_<host name> for the QAS system.
3. Start the Central Services instance of your system using the Microsoft Management Console (MMC).
 - a) Select your Central Services instance in the MMC.
 - b) Right-click on *start*



Lesson Summary

You should now be able to:

- Make selected configurations



Unit Summary

You should now be able to:

- Determine the configuration of system parameters
- Adjust the system parameters using profiles
- Set up a dynamic switch of work process types using operation modes
- Name various configuration and administration tools for the SAP NetWeaver AS Java
- Describe the primary usage areas of configuration and administration tools
- Maintain settings for an SAP NetWeaver AS Java with the Config Tool
- Maintain settings for an SAP NetWeaver AS Java with the Visual Administrator
- Obtain an overview of the most important information for an SAP NetWeaver AS Java with the System Information and SAP NetWeaver Administrator or Telnet Console
- Make selected configurations



Test Your Knowledge

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

Choose the correct answer(s).

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

2. In what profile can you change the number of background work processes?

Choose the correct answer(s).

- ☐ A Start profile
- ☐ B Default profile
- ☐ C Background profile
- ☐ D Instance profile

3. You can use transaction _____ to check the consistency of individual profiles.

Fill in the blanks to complete the sentence.

4. In what sequence should you perform the listed steps when setting up operation modes?

Choose the correct answer(s).

- ☐ A 1. Create operation modes 2. Assign instances 3. Distribute work processes 4. Maintain time table
- ☐ B 1. Create operation modes 2. Maintain time table 3. Distribute work processes 4. Assign instances
- ☐ C 1. Create operation modes 2. Distribute work processes 3. Maintain time table 4. Assign instances

5. You can maintain parameter settings of a SAP NetWeaver AS Java with the Visual Administrator when no SAP NetWeaver AS Java instance has been started.

Determine whether this statement is true or false.

- ☐ True
- ☐ False

6. You can use the Config Tool to maintain Java VM parameters for a SAP NetWeaver AS Java instance.
Determine whether this statement is true or false.
- ☐ True
 - ☐ False
7. The parameters that are maintained with the Visual Administrator only become active after the SAP NetWeaver AS Java has been restarted.
Determine whether this statement is true or false.
- ☐ True
 - ☐ False
8. If you want to use the System Information tool, you need to install it on your front end.
Determine whether this statement is true or false.
- ☐ True
 - ☐ False
9. You can use the Config Tool to add additional server processes to an SAP NetWeaver AS Java instance.
Determine whether this statement is true or false.
- ☐ True
 - ☐ False



Answers

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

Answer: C

There is no file called start.ini involved in this context; it has been made up.

2. In what profile can you change the number of background work processes?

Answer: D

The instance profile is the correct profile in this case, as settings in this profile only apply to the affected instance, and the number of work processes for an instance is defined here. The start profile contains information about the processes to be started during the system start. In the default profile, you maintain parameters that apply to all instances or for the entire SAP system. The background profile does not exist.

3. You can use transaction RZ10 to check the consistency of individual profiles.

Answer: RZ10

In addition to the consistency of the individual profiles, you can use transaction RZ10 to check *all profiles*. In this case, the system also checks the consistency between the profiles.

4. In what sequence should you perform the listed steps when setting up operation modes?

Answer: A

You must follow the correct sequence when first setting up the operation modes.

5. You can maintain parameter settings of a SAP NetWeaver AS Java with the Visual Administrator when no SAP NetWeaver AS Java instance has been started.

Answer: False

With the Visual Administrator, you must log on to a Java instance. You can only maintain parameters of started instances.

6. You can use the Config Tool to maintain Java VM parameters for a SAP NetWeaver AS Java instance.

Answer: True

You can maintain parameters such as -Xms with the Config Tool. -Xms is a Java VM parameter.

7. The parameters that are maintained with the Visual Administrator only become active after the SAP NetWeaver AS Java has been restarted.

Answer: False

Most of the parameters changed with the Visual Administrator become active immediately.

8. If you want to use the System Information tool, you need to install it on your front end.

Answer: False

System Information is a Java application that runs in a browser.

9. You can use the Config Tool to add additional server processes to an SAP NetWeaver AS Java instance.

Answer: True

You can use the Config Tool to add additional server processes to an SAP NetWeaver AS Java instance.

Unit 6

Implementation of SSL for SAP NetWeaver AS Java

Unit Overview

SAP NetWeaver AS Java forms the technical basis for SAP Java applications. For example, a portal landscape consists of many different components which are distributed across intranets, extranets or the public web. This gives rise to questions such as “How safe is my data?” and “Can I protect the communication between these components against attacks and infiltration?” This topic discusses the issue “Ensuring secure, encrypted communication between the different components”.



Unit Objectives

After completing this unit, you will be able to:

- Describe different encryption processes
- Point out the coherence between authentication and digital signatures
- Explain the server authentication mechanism used within SSL
- Setup SSL for the SAP NetWeaver AS Java

Unit Contents

Lesson: Setting up SSL	382
Exercise 23: Setting up SSL	399

Lesson: Setting up SSL

Lesson Overview

This lesson gives you a brief introduction to cryptography and its adoption in the communication between different communication partners. In the second part you will learn how to setup a secure http-communication (SSL).



Lesson Objectives

After completing this lesson, you will be able to:

- Describe different encryption processes
- Point out the coherence between authentication and digital signatures
- Explain the server authentication mechanism used within SSL
- Setup SSL for the SAP NetWeaver AS Java

Business Example

Your corporation wants to provide access to applications on SAP NetWeaver AS Java for its business partners. Since sensitive data is transferred between the AS Java server and the client (a web browser) a secure communication should be established.

Introducing Cryptography

Cryptography is the science of encrypting information. Why is this a big subject of today's IT world? The standard protocol used for transporting http requests, TCP/IP, is a insecure transport mechanism. Everyone connected to a specific network is able, with more or less effort and knowledge, to listen to the packages and its content transferred with the IP protocol in that network. This vulnerable protocol makes it necessary to encrypt the transferred data itself. For a better understanding we describe here a possible attack against the TCP/IP protocol and the data transferred with this protocol.

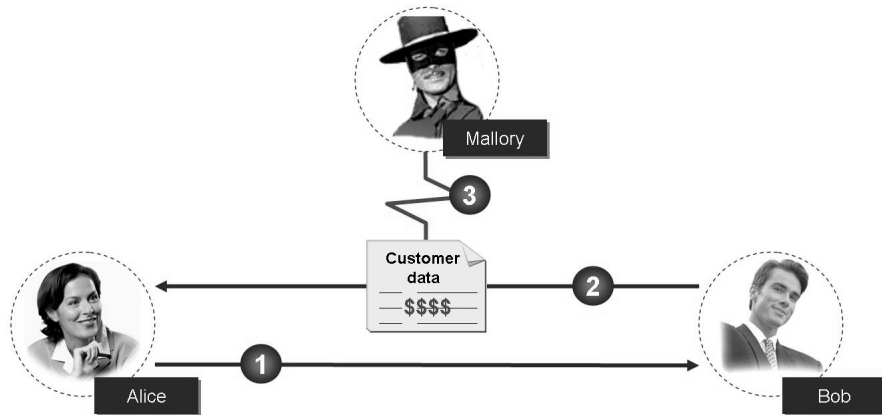


Figure 178: Threat: Eavesdropping

In above example Alice (1) initiate a communication with Bob and requests some data about customers from him. Bob gathers the requested data and responds to Alice's request (2). The entire exchange is eavesdropped by Mallory. He now knows about the discussed information (3).

Transferring that in the TCP/IP world means, Alice, which is a web browser for example requests some data via an http-request transferred with the TCP/IP protocol. The server (here represented through Bob) responds and transfers some sensitive customer data from the server to the client via the TCP/IP protocol. Mallory, an attacker is on the same network and therefore is able to eavesdrop that TCP/IP communication.

The solution for securing that communication is the **encryption** of the transferred data, in means making the conversation not understandable to the attacker but understandable for exactly the participants involved in that conversation.

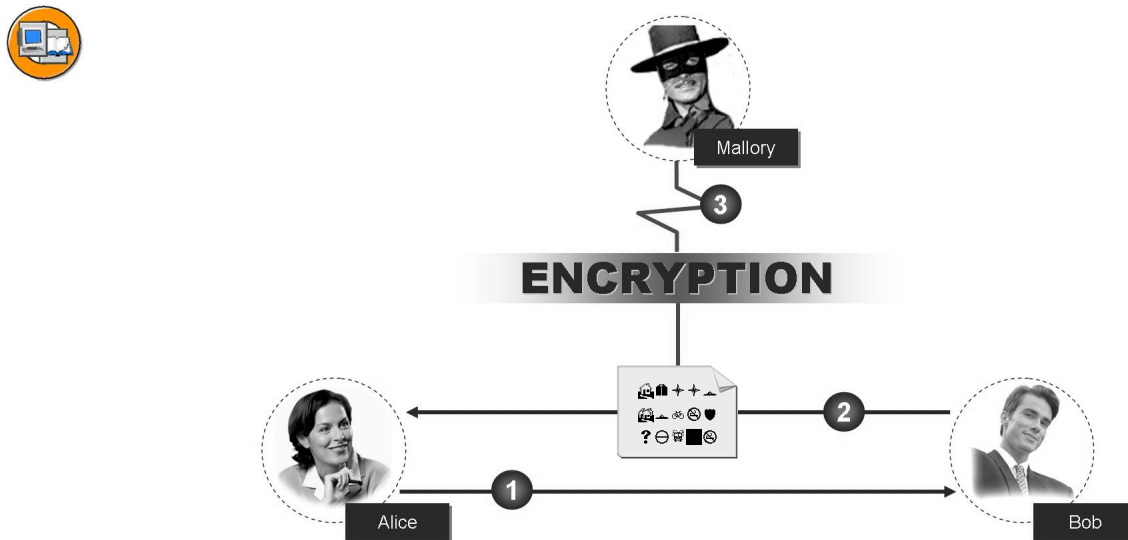


Figure 179: Safeguard: Cryptography

Encryption Methods

Encryption itself is based on mathematical operations. A key therefore has to be exchanged between the communication partners in order to have a computable basis for encrypting and decrypting information. There are three different methods for exchanging these keys.

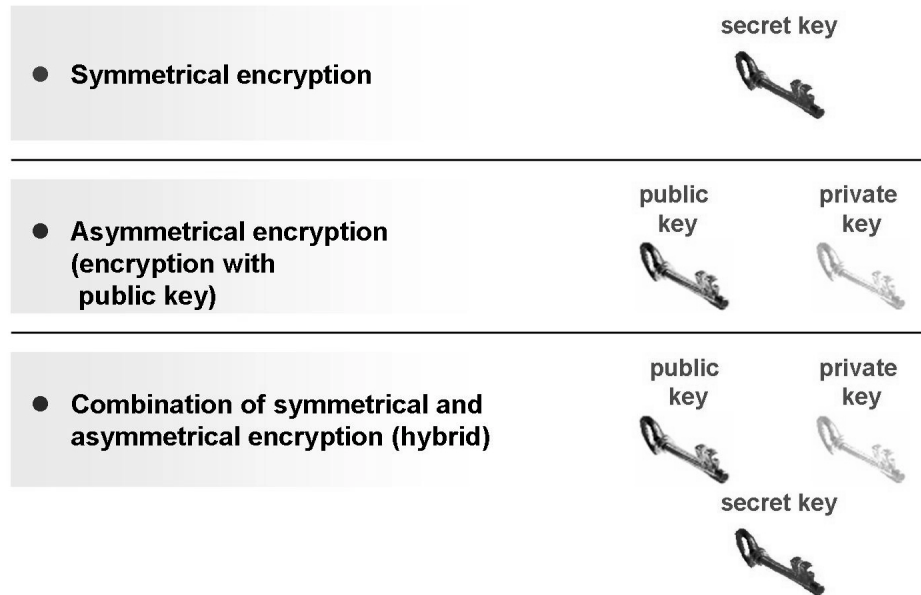


Figure 180: Encryption Methods

Symmetric Key Encryption is the classical cryptography method for encrypting and decrypting messages. In this case, both the sender and receiver of a message share a “secret” called a secret key. The sender uses this key to encrypt the message. The receiver also uses this key to decrypt the message.

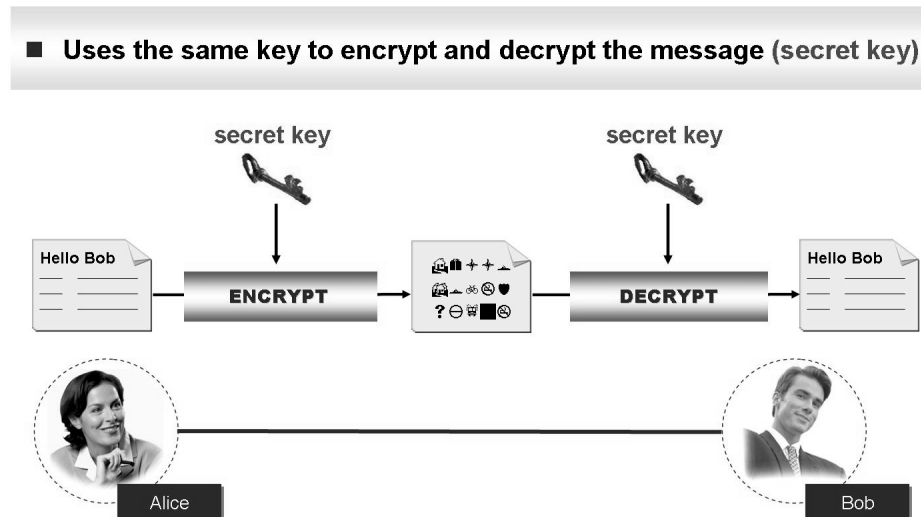


Figure 181: Symmetrical Encryption

The shared secret is called a secret key. It consists of a value of a certain length – 128 bits is the standard in the most recent technology. These encryption algorithms are in widespread use and are employed in most Web browsers and Web servers. Typical Symmetric Key Encryption Algorithms include:

- Digital Encryption Standard (DES)
- Triple DES
- Advanced Encryption Standard (AES)
- International Data Encryption Algorithm (IDEA)
- RC4
- RC5
- Blowfish

Asymmetric Key Encryption uses a different algorithm than Symmetric Key Encryption. Asymmetric Key Encryption uses a key pair that consists of a private and a public key. These keys belong to each other. A message that is encrypted with the public key can only be decrypted with the matching private key. The public key can be made public. The owner of the key pair “publishes” the public key and can distribute it as required. The private key must be kept secret.

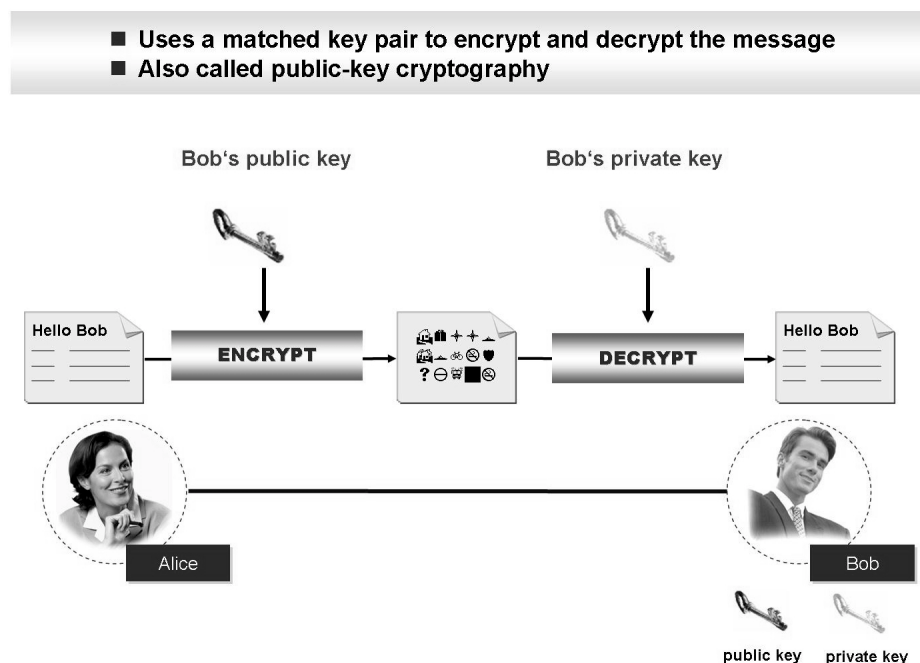


Figure 182: Asymmetrical Encryption

The person who is sending a confidential message uses the recipient's public key to encrypt the message. Only the recipient can then decrypt the message using his or her private key. Typical public key encryption algorithms are:

- RSA (Rivest, Shamir, Adleman), Diffie-Hellman

Disadvantages of Public Key Encryption:

- It is slower than Symmetrical Key Encryption.
- Encryption is only possible in one direction with a single key pair. **Alice** can encrypt a message to send to Bob, but not vice versa.
- Yes, if Alice also has a key pair, then Bob can send her an encrypted message. However, there is an easier way.

Hybrid Encryption Process is the combination of both above explained encryption processes. The Hybrid Encryption Process make use of the advantages of both process types. For the better understanding we describe this process in the following example.

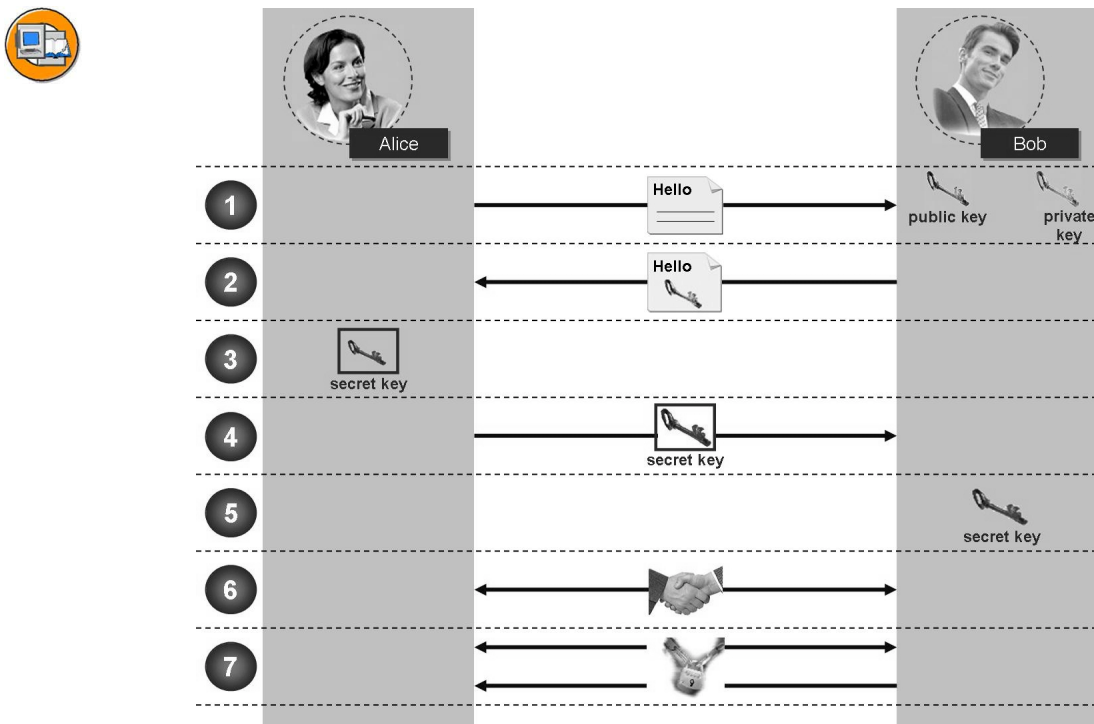


Figure 183: Hybrid Encryption

Process:

1. The client (browser) contacts the SAP NetWeaver Application Server Java
2. The Application Server responds and sends its Public Key
3. Client-side a Secret Key is created and encrypted with the Public Key the server sent before
4. The client sends back the encrypted Secret Key
5. On the server the Secret Key is decrypted using the Private Key. Only the server can decrypt the received Secret Key cause its holding the Private Key which is necessary for the decrypting.
6. The communication partners perform a “Handshake”.
7. Further communication between the client and the server is encrypted using the Secret Key

Authentication and Digital Signatures

In the first part of this lesson we described a possible attack to the transport protocol and what can be done to secure this communication. But what happens if Mallory interferes with the communication and pretends to be Bob? He may even provide Alice a public key, saying that is Bob's key. The question here is now, how can we make sure that Alice is really communicating with Bob and therefore the public key she received is really Bob's public key?



- How does Alice know that the public and private key pair really belongs to Bob and not to Mallory?

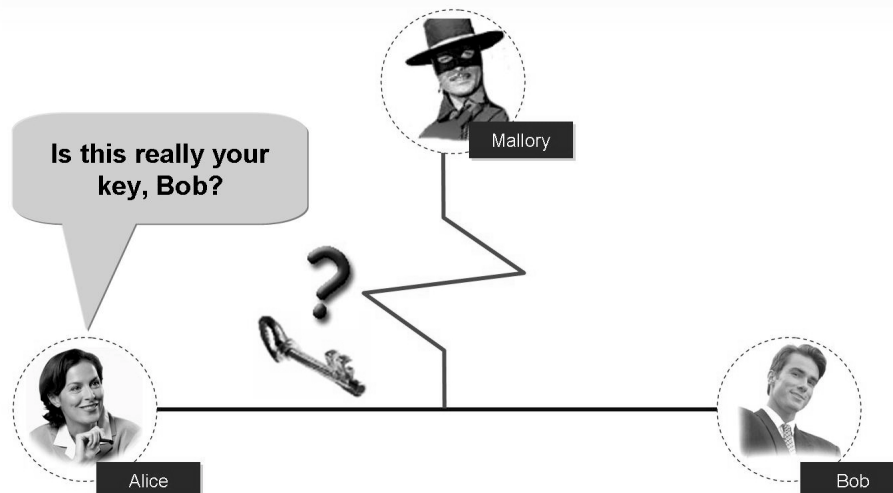


Figure 184: Threat: Masquerading

The problem is also covered by cryptography and is called **Authentication**. Authentication normally takes place using the user ID and password. But with cryptographic mechanisms it is possible to authenticate communication partners, in means of verifying that the communication partner is the one she or he pretends to be. Basis for the authentication of communication partners are **Digital Certificates**.

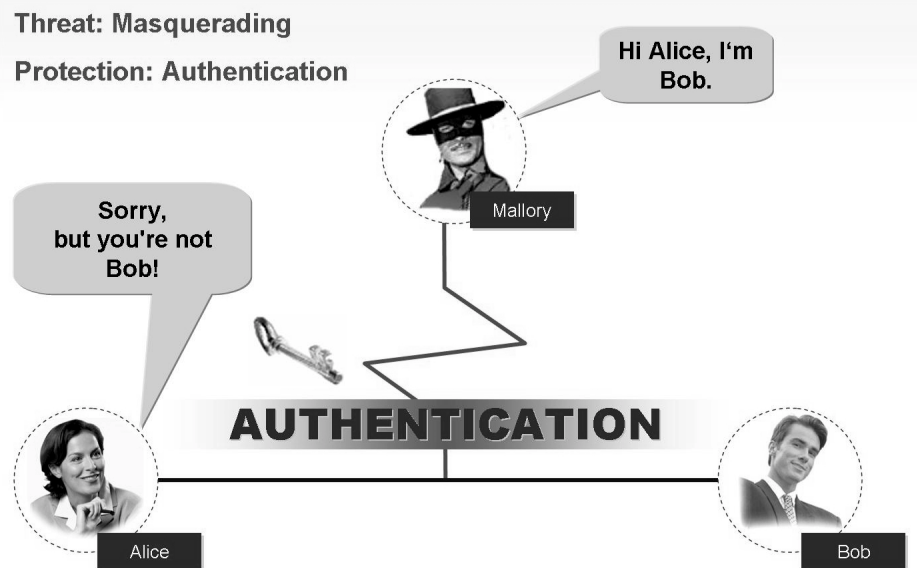


Figure 185: Safeguard: Authentication

Understanding Digital Certificates and Digital Signatures

The digital certificate is the individual's "digital identity card" on the Internet. Compared to the "real world" digital certificates can be compared to a passport which contains information about owner, issuer, serial number, and validity period. The format of the certificate is specified by the X.509 standard for digital certificates.



Contains information about:

Holder
 Public Key Info
 Issuer (CA)
 Validity
 Version
 Serial Number
 Extended Attributes

- e.g. Email,
- Address,
- Position

 CA Digital Signature



Your “Digital Identity Card” on the Web

- Defines the relationship between an identity and a unique public key
- Belongs to an individual or a system
- Digitally signed by Certification Authority (CA)
- Unique with respect to CA and serial number
- Managed within global Public-Key Infrastructure (PKI)
- Contains public part of the cryptographic key pair
- The private key is NOT included and has to be stored in a secure place

Figure 186: X.509 Digital Certificates

Beneath some general information the certificate contains also the public part of the key pair whereas the private key is not included in the certificate. This one must be kept on a safe place. The certificate is issued to a person or server by an authorized entity called a **Certification Authority (CA)**. The CA ensures by digitally signing the certificate that the public key, which matches to a private key, belongs to a specific person or server. Thus, the CA ensures that the certificate cannot be "faked". The complete infrastructure that manages the issue and verification of certificates is called the **Public Key Infrastructure (PKI)**.



- Certificate is issued by an authorized entity called a Certification Authority (CA)
- The issued certificate has been digitally signed by the certification authority (official stamp)



Figure 187: Certification Authority

Examples of well-known Certification Authorities:

- Verisign Inc.
- TC Trust Center

SAP also runs a CA that issues digital certificates to customers. Follow the Quick Link */tcs* (Trust Center Services) on the SAP Service Marketplace.

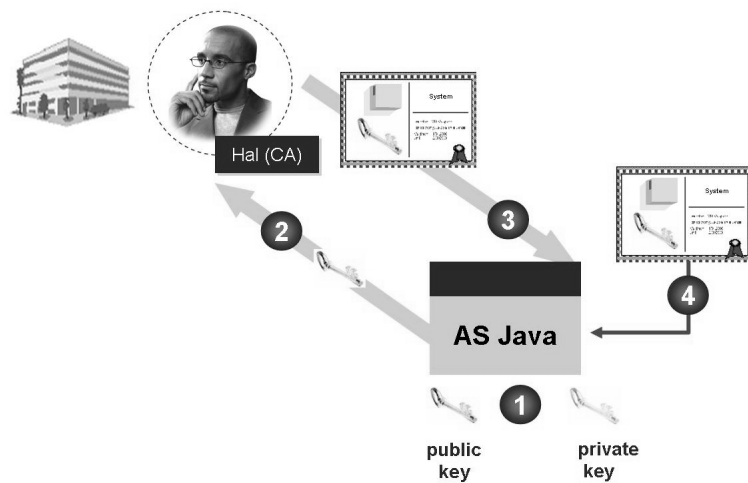


Figure 188: Certificate Enrollment

The certification of digital certificates is performed, for example, as follows:

1. A public and private key pair is generated on the server
2. The public key is sent to the CA (it is called a Certificate Signing Request - short CSR)
3. The CA digitally signs the server's public key and sends it back to the requestor
4. Import of the CSR response, the digitally signed certificate, into the server

Different CA's use different policies, on how to check the identity of a person or system, before issuing a digital certificate. SAP's process for applying for a digitally signed certificate is as follows:

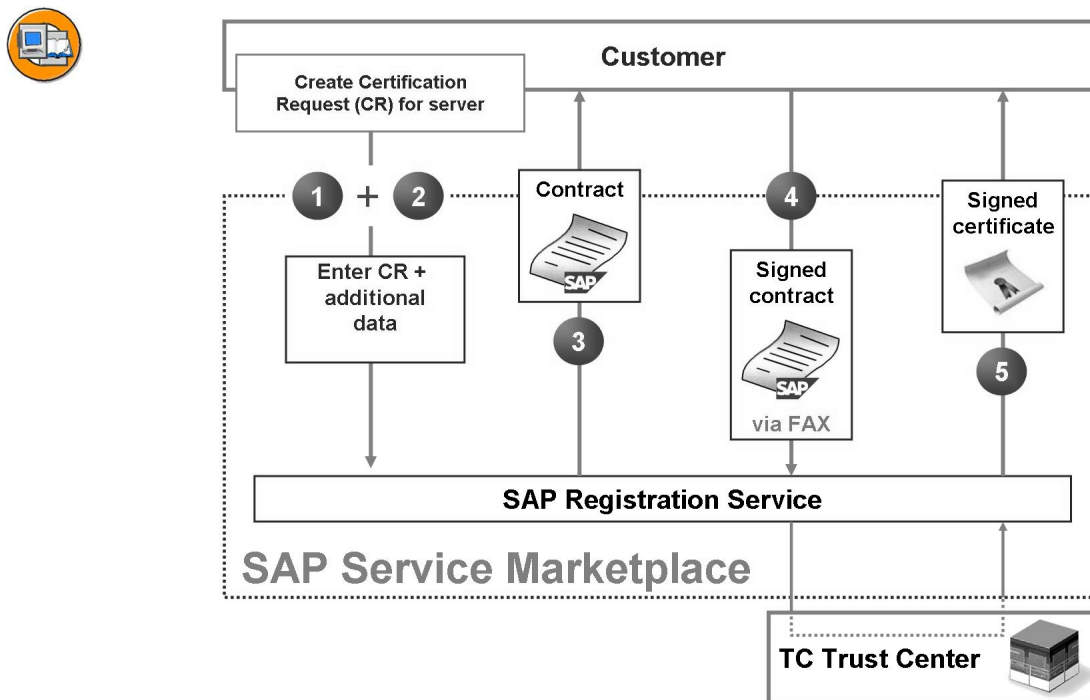


Figure 189: Certificate Order Process via SAP TCS

1. Create CSR and send it to SAP via the SAP Service Marketplace
2. Enter some additional data
3. You receive a contract. Check the details entered before, print it out and sign it
4. Fax the signed contract back to SAP
5. SAP checks your data and has TC TrustCenter issue a certificate

The server is now sending the digitally signed certificate, which includes the public key, to the communication partner. This kind of authentication is called Server Authentication. But how can the communication partner ensure, that the digitally signed certificate is signed from a trusted CA? The communication partner has to have a trust relationship to the CA which issued the certificate. Technically this can be achieved by importing a digital certificate of the institution (CA) issued the certificate for the server. This is the so-called **root certificate**. The most common root certificates are pre-installed in most Web browsers.



■ How does Alice trust the CA technically?

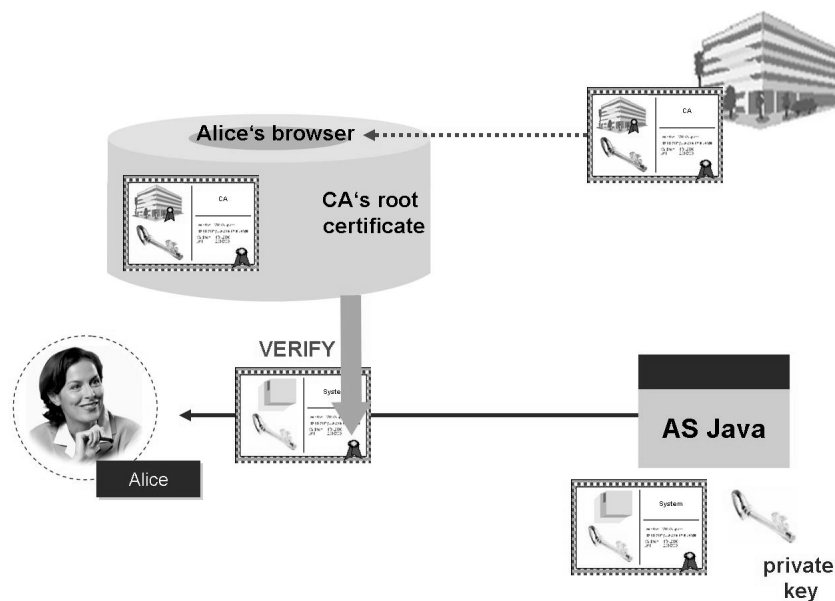


Figure 190: Trusted Relationship

Securing HTTP communication using Secure Socket Layer (SSL)

In the previous sections you learned the fundamentals of Cryptography, Authentication and Digital Certificates. These technologies are also the fundamental of securing the HTTP communication. Secure Socket Layer (SSL) is a transparent protocol enhancing other protocols having no security functionalities. SSL is not an HTTP-specific protocol but a protocol used between the TCP layer and application protocols like LDAP, SMTP, HTTP and so on. An HTTP application protocol that has been extended by SSL has the protocol identification HTTPS in the URL.

SSL uses a Hybrid Encryption method and provides besides **data encryption** the following authentication mechanisms:

- Server authentication
- Client authentication
- Mutual authentication

To use SSL for server authentication, the SAP NetWeaver AS Java possesses a private and public key pair.

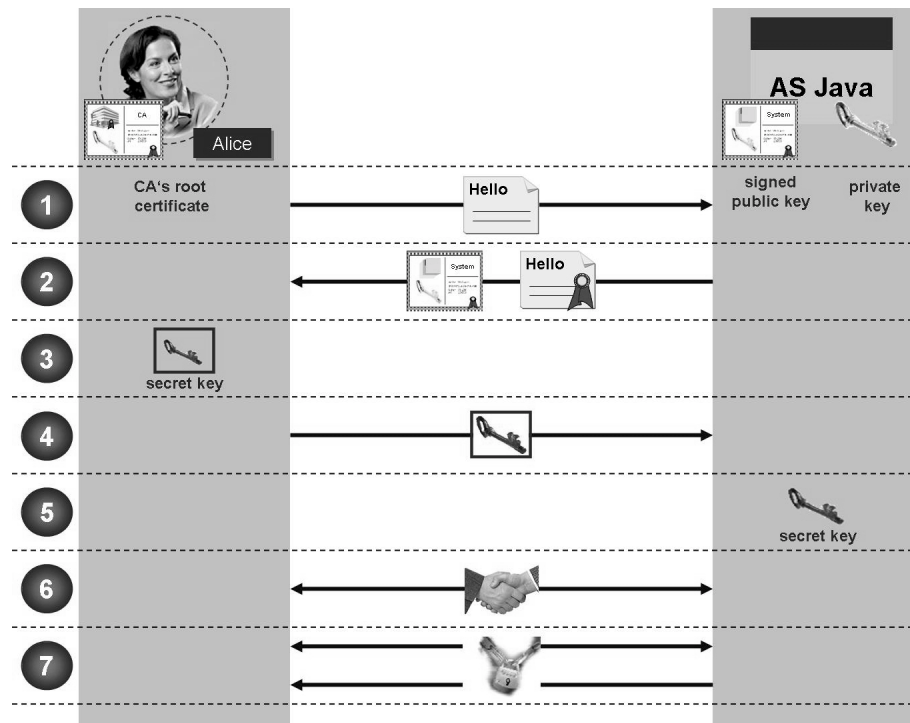


Figure 191: SSL - Server Authentication

1. Alice contacts the SAP NetWeaver Application Server Java using a browser
2. The Application Server responds and sends its Public Key with a digitally-signed message. The client-side server's identity is verified by checking the validity of the certificate. The certificate is only accepted if the client trusts the CA that issued that certificate to the SAP NetWeaver AS Java. This is done with the CA root certificate.
3. The Secret Key is created and encrypted with the Public Key the server sent before
4. The client sends back the encrypted Secret Key
5. On the server the Secret Key is decrypted using the Private Key. Only the server can decrypt the received Secret Key cause its holding the Private Key which is necessary for the decrypting.
6. The communication partners perform a "Handshake".
7. Further communication between the client and the server is encrypted using the Secret Key

Setting up HTTPS for the SAP NetWeaver Application Server Java

As the basis of SSL and therefore HTTPS is cryptography, SAP NetWeaver Application Server Java has to be enabled in order to support this feature. Cryptographic software is needed to support the different mathematical algorithms. This cryptographic software is already delivered with SAP NetWeaver AS Java installation. To make use of strong encryption algorithms additionally so-called JCE-policy files have to be downloaded from your J2SE vendor and installed as described there.

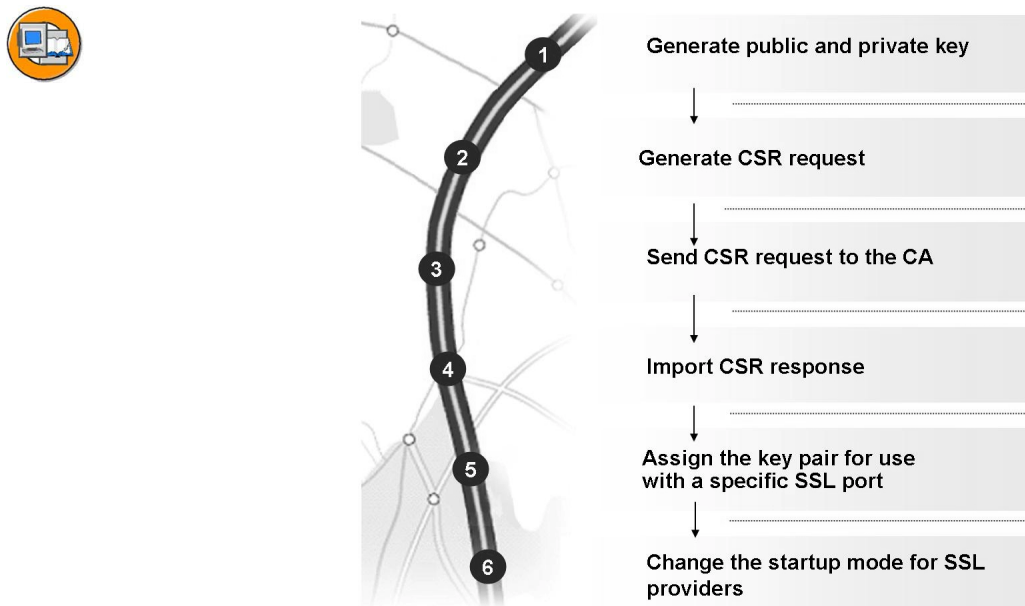


Figure 192: Roadmap for Configuring SSL

Using SSL with an Intermediary Server

You can also use SSL for connections where an intermediary server is used. An intermediary server may be a Web proxy or the standalone SAP Web Dispatcher. A typical scenario is to place the intermediary server in the DMZ and the SAP NetWeaver AS Java in the intranet zone. The servers that are supported for use with the SAP NetWeaver AS Java are:

- SAP Web Dispatcher
- Microsoft Internet Information Server (IIS) with an IIS proxy module from SAP
- Other servers (for example, the Apache Web Server)

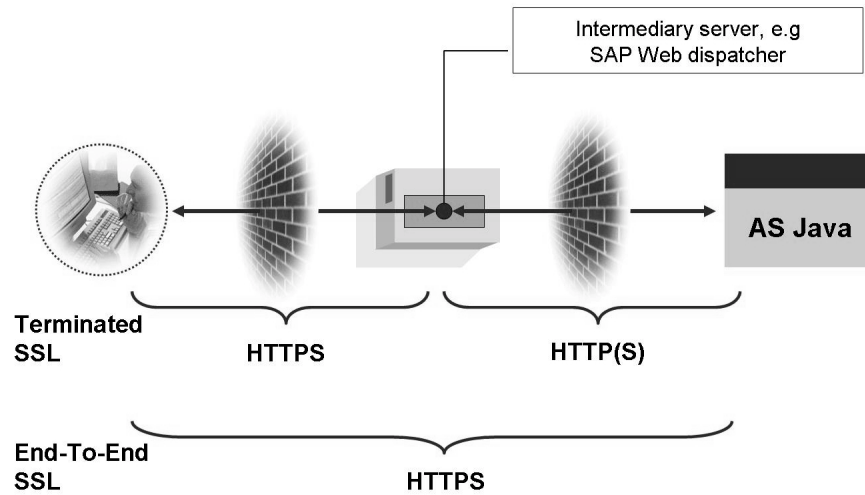


Figure 193: SSL with an Intermediary Server

Depending on the intermediary server used, options exist to use either an end-to-end SSL connection or to terminate the connection on the intermediary server and establish a new connection to the backend system (terminated SSL). See the figure below.

Outlook: Mutual Authentication

Beside the server authentication mechanism and the data encryption described in the above sections, SSL can also be used for mutual authentication.

In case of **Mutual Authentication** both, the user and the server acknowledge their authenticity by providing a digitally signed certificate to the other communication partner. The important fact is, Alice also authenticates herself to the server. Therefore you can use this authentication to integrate the SAP NetWeaver AS Java in a Single Sign-On environment.

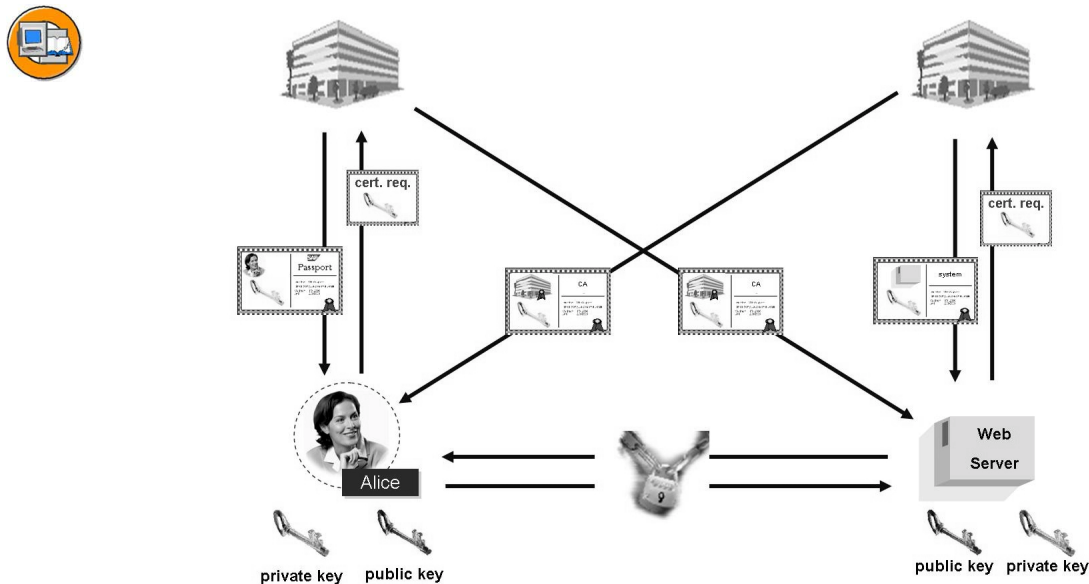


Figure 194: Mutual Authentication

Server authentication is performed using the same process as described within the SSL scenario. Let's focus on the client part of this authentication. Alice obtains a certificate, as shown in the figure:

- Alice creates a key pair and a certificate request
- Alice sends the request to a CA, such as the SAP CA
- Alice imports the certificate request response

The Web server must also trust Alice's issuing CA by importing its CA root certificate into its trusted CA store. When communicating with the server, both parties are authenticated and the data communication is encrypted.

Exercise 23: Setting up SSL

Exercise Objectives

After completing this exercise, you will be able to:

- Generate a key pair for SAP NetWeaver AS Java
- Import a digitally signed certificate
- Assign the certificate for use with a specific HTTP port

Business Example

Task 1: Key Generation

Generate a private-public key pair and create a certificate signing request (CSR).

1. Launch the *Visual Administrator* and navigate to the *Key Storage* service of a server node.
2. Create a new public and private key under the *service_ssl* entry. Use the following data:

Attribute	Value
Country Name (2-letter code)	<country code> for example, DE for Germany, US for United States.
State/Province (full name)	<optional>
Locality Name (e.g. city)	<your training location> e.g. Walldorf or Sydney.
Organization Name	SAP-AG
Organization Unit Name	Education
Common Name	<hostname>.<fully-qualified domain name of your training server> e.g. twdfSSSS.wdf.sap.corp
Entry Name	SSL-ADM200
Store Certificate	true

3. Generate the CSR data and save the information on a local drive.

Continued on next page

Result

Congratulations! You successfully generated a public-private key pair and a CSR.

Task 2: Signing and Importing the Certificate

Apply for a SSL server certificate and import the CSR response in your SAP NetWeaver AS Java.

1. Open a Web browser and call the SAP Service Marketplace with the Quick Link `/ssltest`.
2. Access the SSL order test application and enter your certificate request.
3. Import the certificate into your SAP NetWeaver AS Java.

Result

Congratulations! You successfully applied for a SSL server certificate and imported it to your SAP NetWeaver AS Java.

Task 3: Assigning the Certificate to a Port

Assign the imported certificate to the https port of the SAP NetWeaver AS Java and test the https connection.

1. Determine the https port used by the Dispatcher of your SAP NetWeaver AS Java installation.
2. Assign the previously imported certificate to the https port of your SAP NetWeaver AS Java.
3. Test the SSL connection to your DEV or QAS training system.

Result

You successfully setup https on your SAP NetWeaver AS Java.

Task 4: Importing the Root Certificates

If a security alert came up while accessing the https port of your SAP NetWeaver AS Java import the Root CA certificate to your Web browser.

1. Identify the issuer of your server certificate.
2. Download the certification authority's root certificate.
3. Import the certification authority's root certificate.

Continued on next page

4. Again, test the SSL connection to your SAP NetWeaver AS Java.

Result

Congratulations again! You did it! The server certificate process is now complete!

Solution 23: Setting up SSL

Task 1: Key Generation

Generate a private-public key pair and create a certificate signing request (CSR).

1. Launch the *Visual Administrator* and navigate to the *Key Storage* service of a server node.
 - a) Navigate to the Visual Administrator with *Start → Programs → SAP J2EE Engine → Visual Administrator*.
 - b) You will be prompted to choose a connection. Choose your previously created connection, click the *Connect* button and enter your password on the following screen. Another time, click the *Connect* button.
 - c) Navigate to *Server 0 → Services → Key Storage*.
2. Create a new public and private key under the *service_ssl* entry. Use the following data:

Attribute	Value
Country Name (2-letter code)	<country code> for example, DE for Germany, US for United States.
State/Province (full name)	<optional>
Locality Name (e.g. city)	<your training location> e.g. Walldorf or Sydney.
Organization Name	SAP-AG
Organization Unit Name	Education
Common Name	<hostname>.<fully-qualified domain name of your training server> e.g. twdfSSSS.wdf.sap.corp
Entry Name	SSL-ADM200
Store Certificate	true

- a) In the *Runtime* tab, select the *service_ssl* entry in the *Views* pane of the *Key Storage* service.
- b) Select the *Create* button in the *Entry* area. Fill the next screen with the data provided in the above table.
- c) Press the *Generate* button.

Continued on next page

3. Generate the CSR data and save the information on a local drive.
 - a) In the *Views* pane, select *service_ssl* and in the *Entries* pane, select *SSL-ADM200*.
 - b) On the *CSR* area click the *Generate CSR Request* button.
 - c) In the following dialog screen select the folder *G:\usr\sap\<SID>* and enter your servers' name in the *File Name* field (e.g. twdfSSSS) and click on *OK*. *<SID>* can be either DEV or QAS.

Result

Congratulations! You successfully generated a public-private key pair and a CSR.

Task 2: Signing and Importing the Certificate

Apply for a SSL server certificate and import the CSR response in your SAP NetWeaver AS Java.

1. Open a Web browser and call the SAP Service Marketplace with the Quick Link */ssltest*.
 - a) Open the Internet Explorer on your training server (twdfssss).
 - b) Enter the following URL in the address bar. <http://service.sap.com/ssltest>.
 - c) If present, use your personal S-User-ID to logon to the SAP Service Marketplace or alternatively use the S-User-ID provided by your instructor.
 - d) In the *Choose a digital certificate* screen select the S-User from the list and press *OK*.

Continued on next page

2. Access the SSL order test application and enter your certificate request.
 - a) Click *Test it Now!* in the "SSL Test Server Certificate" box.
 - b) Open the folder *G:\usr\sap\<SID>* on your training server (or the location you saved your CSR in the previous task).
 - c) Right-click on the file you created in the previous task and choose *Send to* → *Sappad.exe* in the context menu.
 - d) Select the complete content of the file via *Edit* → *Select all* and copy it to the clipboard using *Edit* → *Copy*.
 - e) Go to the SSL order test application you opened before and paste the public certificate in the input field *Enter data for public key* with *CTRL+V*.
 - f) Choose *SAP J2EE Server* from the drop-down list located in the *Choose server type* area.
 - g) Press *Continue*.
3. Import the certificate into your SAP NetWeaver AS Java.

- a) Mark the certificate including the *BEGIN CERTIFICATE* and *END CERTIFICATE* statement and copy it to the clipboard. The following is an example of a certificate request response:

```
-----BEGIN CERTIFICATE-----MIICgTCCAeqgAwIBAgICR-
RwwDQYJKoZIhvcNAQEFBQAwUDELMAkGA1UEBhMCREUx
HDAaBgNVBAoTE1NBUCBUcnVzdCBDb21tdW5pdHkxDzAN-
BgNVBAsTBINlcnZlcjES MBAGA1UEAxMJU2Vyd-
mVybIENBMB4XDTA2MDYyNzEzMDY0M1oXDTA2MDgyNjEzMDY0
M1owMTELMakGA1UEBhMCREUxDzANBgNVBAoTBINBUC1BRz-
ERMA8GA1UEAxMIdHdk ZjAyOTcwZ8wDQYJKoZIhvc-
NAQEBBQADgY0AMIGJAoGBALgkp0vq9pUjTP+IJpBN KphK-
wEt6+4ziWh4NaFKvO4omnOhx73/B3pvQM9XQGVeETxRQaVaH-
bYUrvrbGmDNu 8G6EHO5TA1x+uyPnneL+7PekF-
JEwVZSsJEwJx1pNkmhY3E7nSIJK4fYzmTE6ZTM+
pweBOBQCf173BRoshWmwG5eFagMBAAGjgYgwYUwDAY-
DVR0TAQH/BAIwADAIBgNV HRIEHjAChhpodHRwOi8vc2Vy-
dmljZS5zYXAuY29tL1RDUzAOBgNVHQ8BAf8EBAMC
BPAwHQYDVR0OBBYEFKoancevrVPRC0gxnxjr3wacD-
KNSMB8GA1UdIwQYMBaAFIHb g/NK+zUYCLkBvbcdW51zN-
VtJMA0GCSqGSib3DQEBBQUAA4GBABB9TY31rzsWoF1+ 3Q9woag-
FYyf891HZhlBebB24zQvLc0jfUtd/fZdi+3l5rom+O9peAaldc3lgC4RX
```

Continued on next page

```

dwwKuWd9PYkfcP6K2ltNiqbHmeyK9YkVgzkS0DPCYRnX-
aUGKGxShQqVmw9ag8lRp l9yJJuE4tPsaHziXV0vojt+/G2de-----END
CERTIFICATE-----

```

- b) Create a text-file named **<your server hostname>.txt** on the training server in the directory *G:\usr\sap\<SID>*.
- c) Open the file and paste the certificate into it. Save the file and close it.
- d) Rename the file to **<your server hostname>.crt**.
- e) Locate your public-private key in the *Key Storage* service using the Visual Administrator and select the *SSL-ADM200* entry.
- f) In the *CSR* area click on the *Import CSR Response* and select your previously created certificate file in the file selection box and press *OK*.
- g) Check your *SSL-ADM200* entry. This should now contain a section that starts with *[signedWith]*: instead of *[selfSigned]*: and the following line should contain: *[issuerDN]: CN=Server CA,OU=Server,O=SAP Trust Community,C=DE*

Result

Congratulations! You successfully applied for a SSL server certificate and imported it to your SAP NetWeaver AS Java.

Task 3: Assigning the Certificate to a Port

Assign the imported certificate to the https port of the SAP NetWeaver AS Java and test the https connection.

1. Determine the https port used by the Dispatcher of your SAP NetWeaver AS Java installation.
 - a) Open the Visual Administrator and navigate to *<SID> → Dispatcher <ID> → Services → HTTP Provider*.
 - b) In the *Properties* pane a entry named *Ports* shows you the port used for SSL, for example: *(Port:50000,Type:http)(Port:50001,Type:ssl)*.

Continued on next page

2. Assign the previously imported certificate to the https port of your SAP NetWeaver AS Java.
 - a) On the left hand side select the *SSL Provider* service under *<SID> → Dispatcher → Services*.
 - b) In the *Configuration* area of the *Runtime* pane select *Active Sockets* and mark the port you determined before which will be used for the SSL connection by clicking once on it.
 - c) Select the *Server Identity* tab and press the *Add* button. From the *Available Credentials* Screen select your *SSL-ADM200* credentials and press *OK*.
3. Test the SSL connection to your DEV or QAS training system.
 - a) Start a Web browser and enter the following URL **https://<your training server>.<domain>:<https port>**. Example: *https://twdfxxxx.wdf.sap.corp:50001*.
 - b) In case of an security alert read the message marked with an exclamation point and afterwards click on *Yes*.

Result

You successfully setup https on your SAP NetWeaver AS Java.

Task 4: Importing the Root Certificates

If a security alert came up while accessing the https port of your SAP NetWeaver AS Java import the Root CA certificate to your Web browser.

1. Identify the issuer of your server certificate.
 - a) In the previously opened Web browser (where you called the https port) double click on the lock symbol in the lower right area of the Web browser.
 - b) The *Certificates* window appears. Navigate to the *Details* tab and locate the *Issuer* field. The issuer of your certificate is a Root CA with the common name *Server CA* of the organization *SAP Trust Community*.

Continued on next page

2. Download the certification authority's root certificate.
 - a) Open the URL <http://service.sap.com/TCS> and navigate to *Download Area* → *Root Certificates*.
 - b) In the following screen click on *SAP Server CA Certificate*. The file download dialog opens.
 - c) Click the *Save* button and choose the storage location for the certificate e.g. *G:\usr\sap\<SID>*.
3. Import the certification authority's root certificate.
 - a) In any Internet Explorer window on your training server, click on *Tools* → *Internet Options* and navigate to the *Content* tab. Here, you click on *Certificates....*. The *Certificates* window appears.
 - b) Choose the *Trusted Root Certification Authorities* tab and click on *Import....*. Then click on *Next*.
 - c) In the *File to Import* dialog, click on *Browse...* and select the certificate you downloaded in the previous step. Click twice on *Next* and click *Finish* to finalize the import.
 - d) Confirm the *Security Warning* dialog with *Yes*. Now close the *Certificates* Window and the *Internet Options* dialog.
4. Again, test the SSL connection to your SAP NetWeaver AS Java.
 - a) Start a Web browser and enter the following URL **https://<your training server>.<domain>:<https port>**. Example: *https://twdfSSSS.wdf.sap.corp:50001/*.
 - b) Note that no security alert is displayed. Click on the lock symbol in the lower right area of your Web browser and navigate to the *Certification Path* tab of the *Certificate* window. The server certificate of your server now has its issuer certificate as parent in the tree.

Result

Congratulations again! You did it! The server certificate process is now complete!



Lesson Summary

You should now be able to:

- Describe different encryption processes
- Point out the coherence between authentication and digital signatures
- Explain the server authentication mechanism used within SSL
- Setup SSL for the SAP NetWeaver AS Java

Related Information

- Training class ADM960 - Security in SAP System Environments
- SAP Service Marketplace Quick Link */security*



Unit Summary

You should now be able to:

- Describe different encryption processes
- Point out the coherence between authentication and digital signatures
- Explain the server authentication mechanism used within SSL
- Setup SSL for the SAP NetWeaver AS Java

Unit 7

Software Development in SAP Systems

Unit Overview

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

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

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



Unit Objectives

After completing this unit, you will be able to:

- Describe the data structure of the ABAP Stack in an SAP system
- Describe the three-system landscape as recommended by SAP
- Explain the importance of transports
- Describe the procedure for transports
- Name important tools in the *ABAP Workbench*
- Access ABAP Repository objects
- Write a simple ABAP program
- Explain the significance of the *ABAP Dictionary*

- Outline the two-level domain concept
- Name the individual components of the SAP NetWeaver Development Infrastructure
- List the main steps in terms of software development with SAP NetWeaver Development Infrastructure

Unit Contents

Lesson: Data Structure of an SAP System and Transports between SAP Systems (ABAP Stack).....	413
Lesson: Accessing and Editing ABAP Repository Objects.....	423
Exercise 24: Accessing and Editing ABAP Repository Objects	437
Lesson: Introduction to the SAP NetWeaver Development Infrastructure (NWDI)	446

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

Lesson Overview

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



Lesson Objectives

After completing this lesson, you will be able to:

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

Business Example

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

Data Structure of SAP Systems

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



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

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

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

A **client** is a unit within an SAP system that is self-contained in terms of business, organization and data. A client is characterized by the fact that it has its own business data environment, its own master and transaction data, and its own user data. This data in a client is known as **client-specific data**. Client-dependent data types are closely interdependent. For example, business application data is checked against the Customizing settings for the client when this data is entered, and rejected if this data is inconsistent.

When an SAP system is implemented, **Customizing** is set up for each client. You use Customizing settings to define the customer's organizational structures, such as distribution channels and company codes, and to set parameters for SAP transactions so that they reflect customer-specific processes. In addition to the client-specific Customizing settings, there are also settings that only need to be made once to be valid for all clients in that SAP system. These cross-client Customizing settings include the public holiday calendar, for example, and settings for the system change option.

The **Repository**, the central store for all ABAP Workbench development objects, is also cross-client. It contains all Dictionary objects (tables, data elements, domains), and also all ABAP programs, menus, and screens. Because of the Repository's cross-client nature, any Repository objects developed or changed in any one client are used in exactly the same form in every other client in that system.

Repository objects are grouped together to form packages. Packages are containers for semantically linked development objects (programs, tables, and so on). Packages are characterized by the properties nesting, interfaces, visibility and use accesses. The ABAP Repository contains packages from all software components.

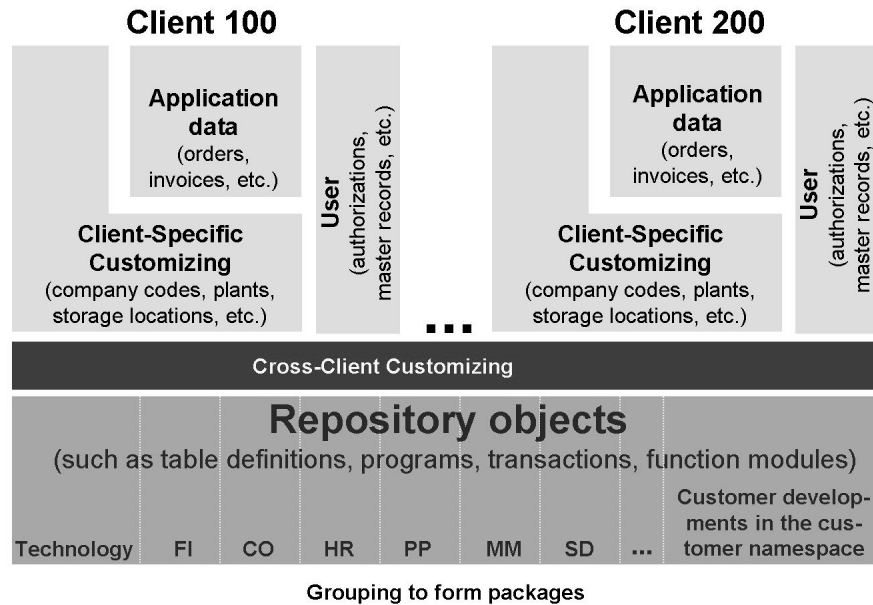


Figure 195: The Data Structure of the ABAP Stack in an SAP system

Changing and Adapting the Data Structure

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

The SAP system software is standard business software that needs to be adapted to company-specific requirements when it is implemented. The process of adapting the software is known as Customizing. It includes both client-specific and cross-client settings. You have a special introductory guide for this in the SAP system: the SAP Reference IMG (to be accessed using transaction SPRO). You may need to repeat Customizing on a much smaller scale when you upgrade your SAP system.

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

- Customer developments
- Customer enhancements
- Modifications

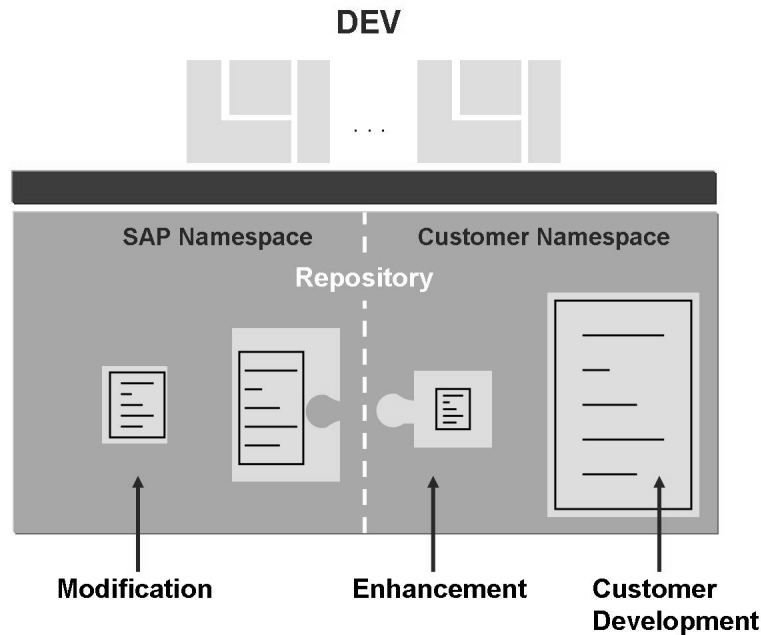


Figure 196: Adjustments to the ABAP Repository

Expanding the Repository through **Customer developments**. In the SAP system, you can create your own Repository objects, such as tables, programs, transactions, and so on. All customer development normally takes place in the customer namespace, that is, all the objects created by the customer have names from a specified namespace; these names generally start with the letters Y or Z. This applies to ABAP programs, tables and so on. SAP has also provided its customers with an additional method of globally unique assignment of individual namespaces for several years. Such a customer namespace could be, for example, /<companyname>/.

Changing the Repository through **enhancements**: In this type of change, you add customer-specific objects to the Repository. There are specific places in the coding, known as **customer exits**, where you can supplement the SAP standard system with your own objects. There is also another modern method for the customer-specific enhancement of the standard: the use of so-called **Business Add-Ins** (BAdIs).

Modifications to the standard SAP system: Changes to SAP objects such as programs and table definitions are called modifications. The Repository delivered by SAP is not only extended, but changed. When you next upgrade your system or when importing Support Packages, you will therefore need to check these modifications against the new Repository. This modification adjustment takes time, although using the **Modification Assistant** speeds up the process substantially.

The Three-System Landscape

There are consequences from the data structure of the ABAP Stack in an SAP system described above - especially when changing data. Since the ABAP Repository objects are not client-specific, you must not use the same system for development and production. The risk of data inconsistency or data loss is too high. Also taking into consideration security factors (and perhaps performance aspects, if the developers have to debug programs), developers must not work on a productive system.

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

A three-system landscape facilitates the following recommended process:

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

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

- Development
- Quality assurance
- Production

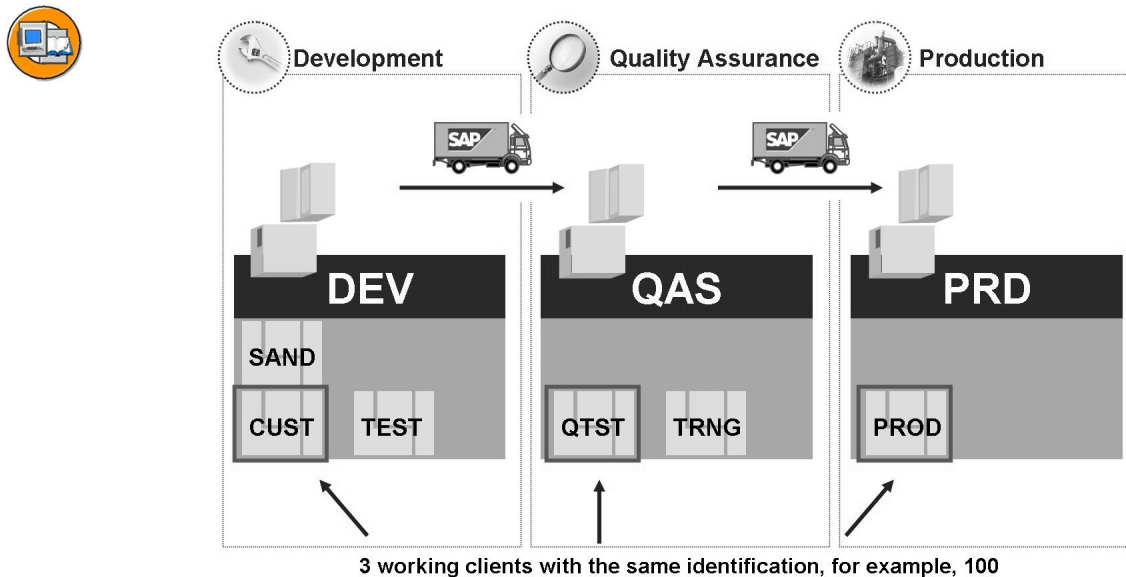


Figure 197: Three System Landscape in the ABAP Environment

Transports in the ABAP Environment

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

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

The following procedure is selected for the transport of objects: When a development project starts, the person responsible for the development project creates a change request. In doing so, he or she assigns the team members to the change request. The Transport Organizer SE09 assigns a number to the change request (in the <SID>K9<nnnn> format, so for example, DEVK900050). A request should contain objects that are logically related and can only be constructively executed together. A request therefore enables the transport and administration of complete, constructive finished developments.

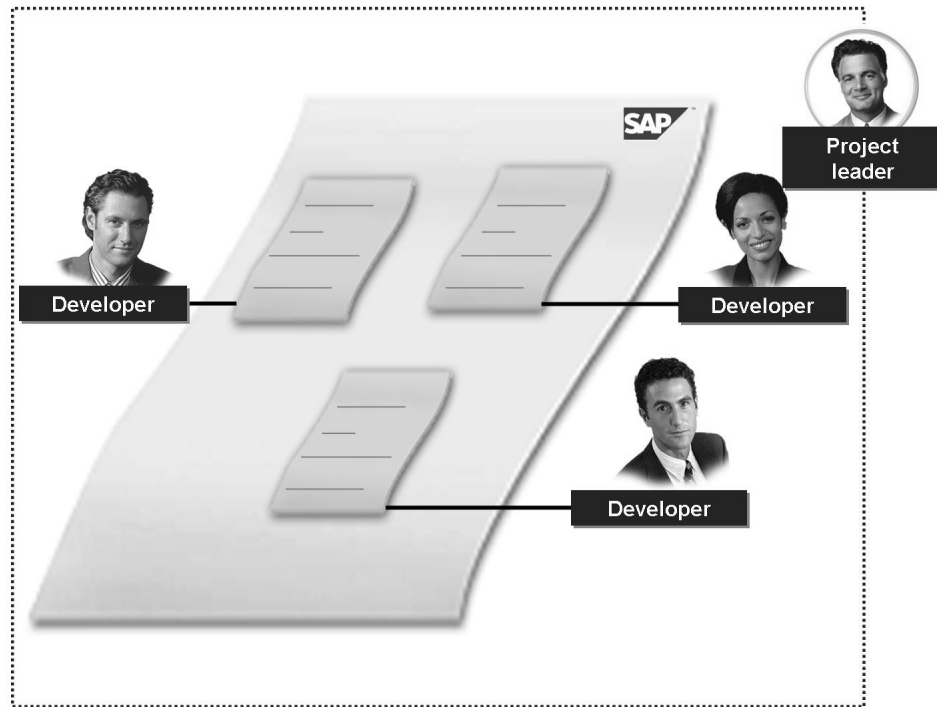


Figure 198: Change Request and Assigned Tasks

The Transport Organizer automatically creates a task for this change request for every employee who is assigned to the change request. If an employee assigns a Repository object to the change request, the Repository object is logged in the task of that employee. The task collects all of the Repository objects that this team member has worked on as part of the change request.

Actions at Close of Development

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

Change requests may be transportable or local. The Transport Organizer classifies the request automatically on the basis of the objects contained in the change request. Data export from the source system is only triggered if the change request is transportable.

Once a request is released, the Repository objects are copied from the source database to a directory at operating system level. The “Release” and “Export” steps are noted in the transport protocol for the change request. You can display this log even after the transport has taken place.

Import into the target system is in general not automatic, but is triggered by the transport administrator in the Transport Management System (TMS). The Repository objects are then copied from the central transport directory at operating system level into the target system database. The import logs can then be checked.

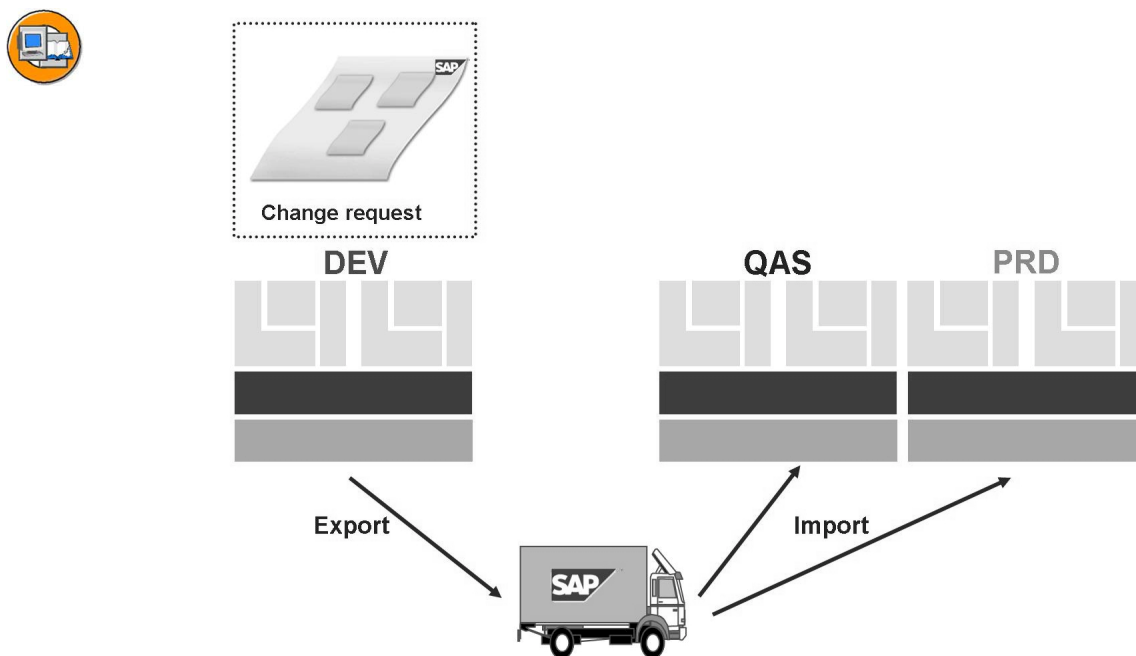


Figure 199: Export and Import into Other Systems

The Software Development Cycle

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

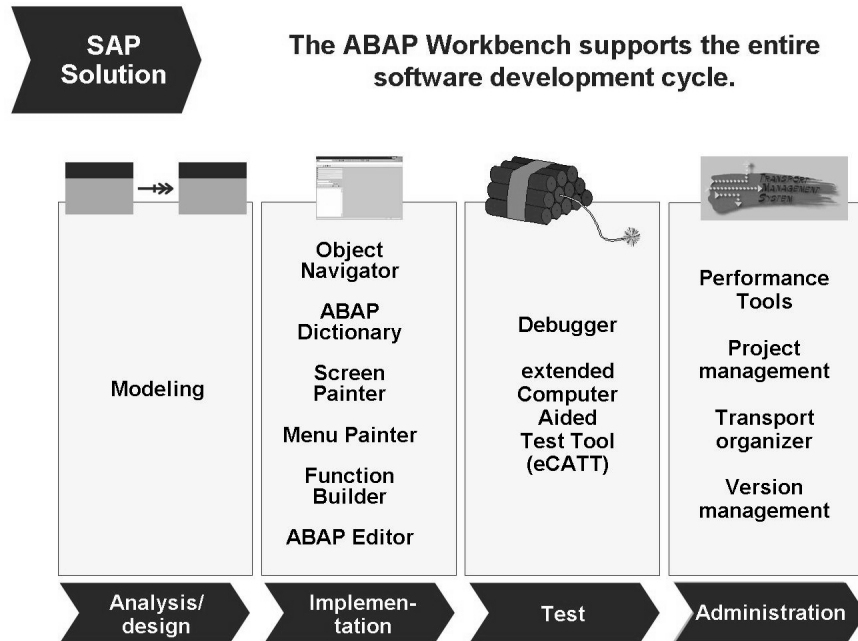


Figure 200: The Software Development Cycle

SAP's ABAP Development Workbench is a programming environment for developing enterprise wide client/server solutions for business purposes. SAP supports the entire software development cycle with tools for modeling, programming using ABAP, defining data and table structures, and for designing graphical user interfaces. An extensive array of tools for testing, tuning, and maintaining software, as well as supporting development in large teams, are also available.

In the concept phase of a project, you enter the results of your analyses into the SAP data model. Then the models are converted into tables and programs. Next you develop the individual program components, such as user interface, ABAP code, and so on, in a sequence of your choice in the separate development tools designed for each functionality. You only need combine the different elements into a single application when you want to run it. The development cycle concludes with program tests and transport into the production system.

As a supplement to the development tools, SAP also provides a library of business and utility software components that you can easily incorporate into your own programs.



Lesson Summary

You should now be able to:

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

Related Information

- For more, very detailed information on this topic, attend the training course ADM325 - *Software Logistics*.
- You can find further information about enhancements and how to deal with modifications in the training course BC425 – *Enhancements and Modifications*.

Lesson: Accessing and Editing ABAP Repository Objects

Lesson Overview

The *ABAP Workbench* is the SAP system's integrated graphical development environment. It supports, among other things, the development, testing, and administration of applications written in ABAP. This lesson introduces various ABAP Workbench tools and the connections between them.

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

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



Lesson Objectives

After completing this lesson, you will be able to:

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

Business Example

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

The Language ABAP

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

The example program

```
REPORT first report.
```

```
WRITE 'My first ABAP report!'.
```

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

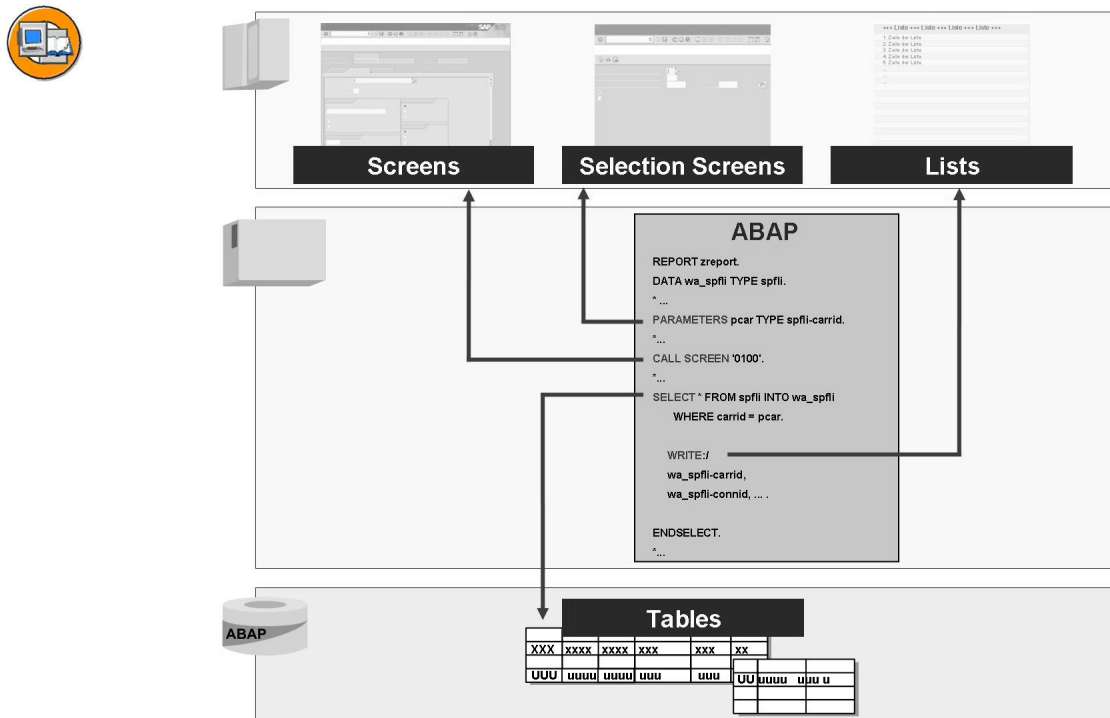


Figure 201: The ABAP Language

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

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

Some characteristics of the ABAP programming language are:

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

ABAP Standards

The economic core functions of an SAP ERP system (for example, ECC 6.0) are programmed in the ABAP language. The client-server architecture of the SAP system (which is often three-level) is reflected in the structure of traditional ABAP applications. The standards provided by SAP for development will be described in the following sections.

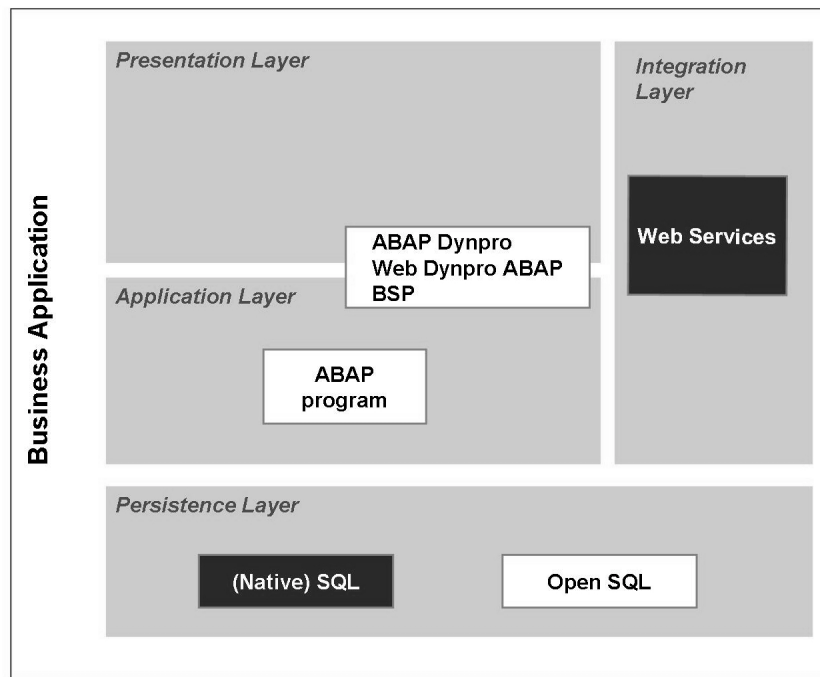


Figure 202: ABAP Standards and Enhancements for SAP Applications

Conventional application programs are programmed in ABAP. ABAP dynpros (screens) are displayed in the conventional SAP GUI; they are only displayed in the SAP GUI for HTML after conversion into HTML.

In the era of browser-based Internet access, you can also access ABAP applications using a web browser and operate applications or even administrate systems in this way. Of course, the ABAP applications do not need to be reprogrammed for this, the output is presented using web dynpros which can be programmed for the ABAP and Java runtime environments.

Free-style Business Server Pages (BSP) can be programmed in ABAP for customer-specific, web-based applications. These are usually dynamic web applications and not static pages. The dynamic behavior is attained by means of the processing of the ABAP coding or runtime Java script. Business Server Pages can contain both process and presentation logic.

The standard applications can be enhanced and supplemented with custom web services. As part of the Enterprise Service Orientated Architecture (ESOA) from SAP Systems, Enterprise and Web Services are being supplied as standard with increasing frequency.

SQL is supported for accessing a database, for example, for table queries from an ABAP application. Open SQL should be used in the ABAP as this ensures that your application is database-independent. The Native SQL, SQL*NET and Open Database Connect (ODBC) technologies are also supported, enabling you to access databases outside of your SAP system.

The ABAP Workbench and its Tools

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

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

- *ABAP Editor* (transactions SE38 or SE80) to write ABAP programs
- *ABAP Dictionary* (transaction SE11) to define and describe tables, data elements, lock objects and so on
- *Screen Painter* (transaction SE51, in the *User Interface* subdirectory) to create interactive user interfaces
- *Function Builder* (transaction SE37), to create and manage function modules (these are encapsulated sections of ABAP code with a defined input/output interface)

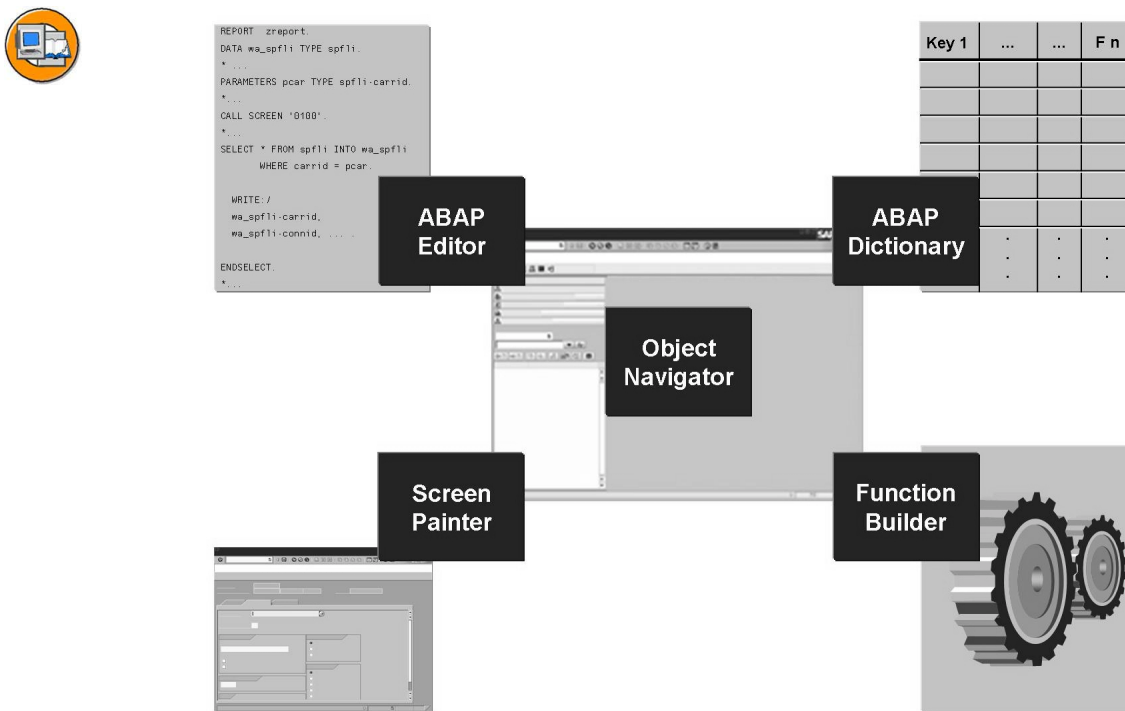


Figure 203: Some ABAP Workbench tools

The individual Workbench tools combine to form an integrated system. If, for example, you are working with program objects in the *ABAP Editor*, then the Editor will also recognize objects created using other tools. By selecting an object and subsequently double-clicking, the tool is started from the workbench in which the object was created. Following that, you can edit this object.

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

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



Hint: In releases before *SAP Web AS 6.10*, packages were still called development classes, which is why, in SAP statements, the data element and the domain *DEVCLASS* still exist (for a definition of data elements and domains, see below).

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

Accessing ABAP Source Code

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

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

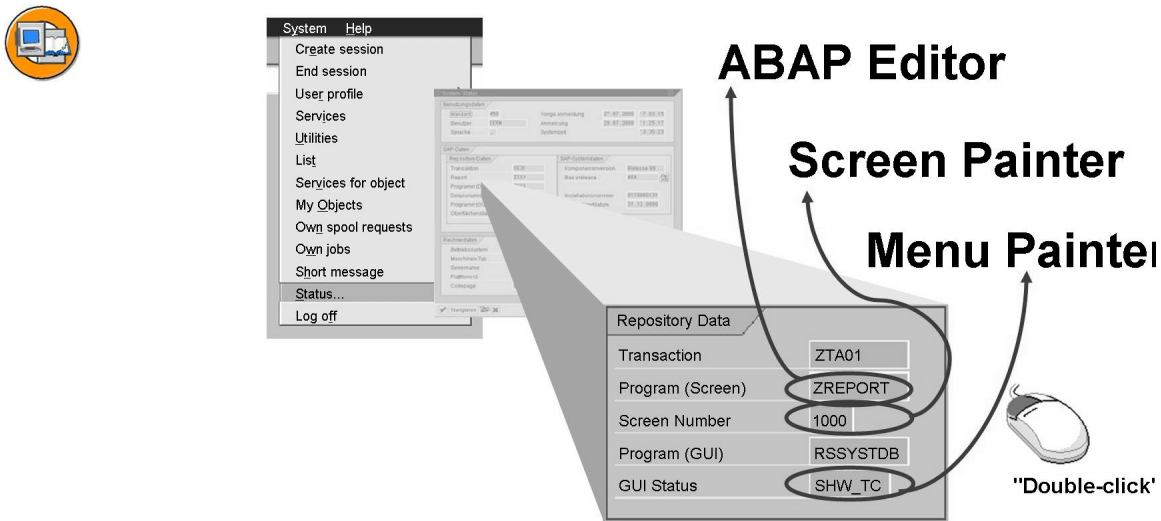


Figure 204: Navigating into the Source Code Editor

Creating ABAP Reports Using the ABAP Editor

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

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

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

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

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

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

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

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



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

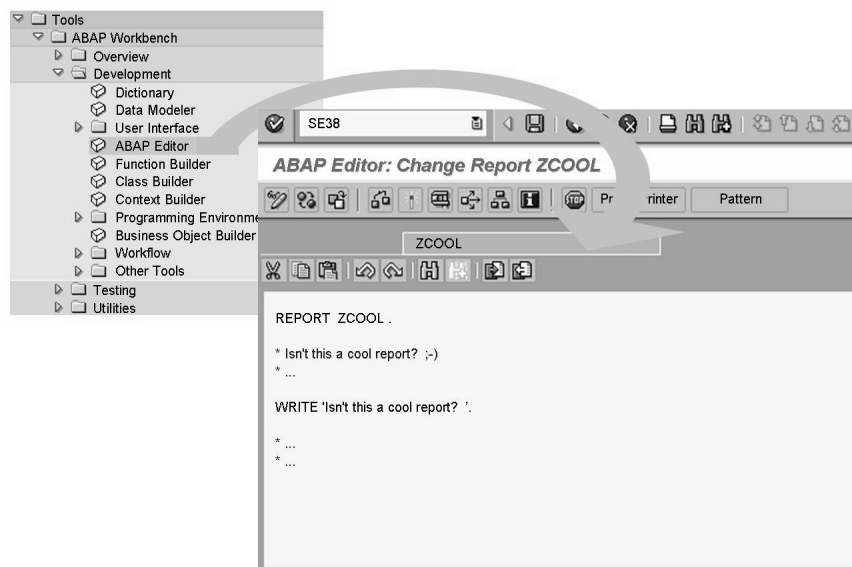


Figure 205: Editing ABAP Reports Using the ABAP Editor

Creating Business Server Pages using the Object Navigator

The *Internet Communication Manager* enables SAP systems to communicate outside the SAP environment using the HTTP, HTTPS and SMTP protocols. The *ICM* can process requests from the Internet that include its server/port combination in their URLs. If database data is required for processing the request, then a connection to a work process is created using memory pipes.

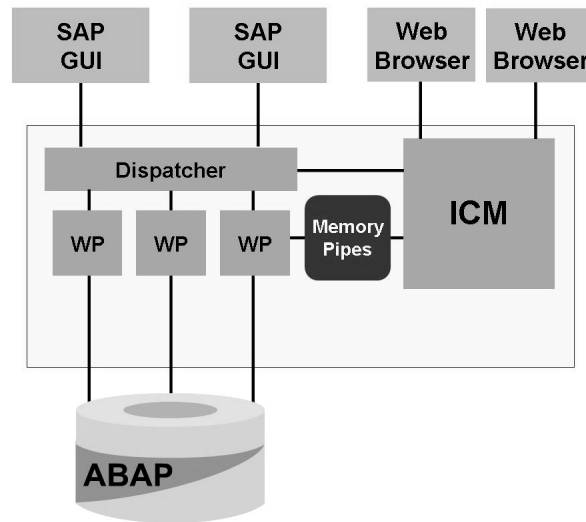


Figure 206: SAP NetWeaver Web Application Server with ICM

As of *SAP Web AS 6.10*, work processes are able to directly create Web-enabled content that the *ICM* then transmits to the browser front end that sent the original request. You can develop this content, called **Business Server Page applications**, in the SAP system using a tool in transaction SE80, the **Web Application Builder**.

Business Server Pages (BSPs) enable you to map complex Internet business processes in the *SAP NetWeaver AS*. You can use the SAP system to create some of the graphic elements required for a corporate website (such as HTML pages, or Web themes) and to manage (and store) those and other elements (such as the MIME objects used). You can use both ABAP and JavaScript as scripting languages. The business logic is created in ABAP Objects. ABAP Objects is an object-oriented extension of the ABAP programming language. It enables you to use the principles of object-oriented programming in ABAP by using concepts such as encapsulation, inheritance, classes, and interfaces.



Hint: You should not confuse Business Server Pages with Java Server Pages (JSP), which are executed in the Java runtime environment of the *SAP NetWeaver AS*.

Often, SAP customers already have tools for creating attractive corporate websites. To enable you to continue using these tools, SAP systems support the WebDAV standard (DAV = Distributed Authoring and Versioning). In other words, you can design pages in the *SAP NetWeaver AS*, although you need not, if you prefer to use other tools.

Business Server Page programming is very much a modern technique. It should be used in projects in which the flow logic and the display in the web browser are linked to each other. Put briefly: Business Server Pages are used for specific customer projects which require free-style programming.

Web Dynpro is an ideal solution if your major priority is a browser-based interface for traditional ERP applications. Web Dynpros can be created for ABAP as of SAP NetWeaver AS 7.00 and for Java runtime environments as of SAP Web AS 6.40.

What Is the ABAP Dictionary?

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

The *ABAP Dictionary* enables all data definitions used in the SAP system to be described and managed centrally. It is an integrated and active dictionary, that is, the *ABAP Dictionary* is completely integrated in the SAP development environment. The Dictionary information is created only once, but is available throughout the system at all times. The *ABAP Dictionary* (transaction SE11) automatically provides all the information that has been created or modified, thus ensuring that runtime objects are up-to-date, and that data is consistent and secure.

The tasks of the *ABAP Dictionary* can be subdivided into:

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

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

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

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

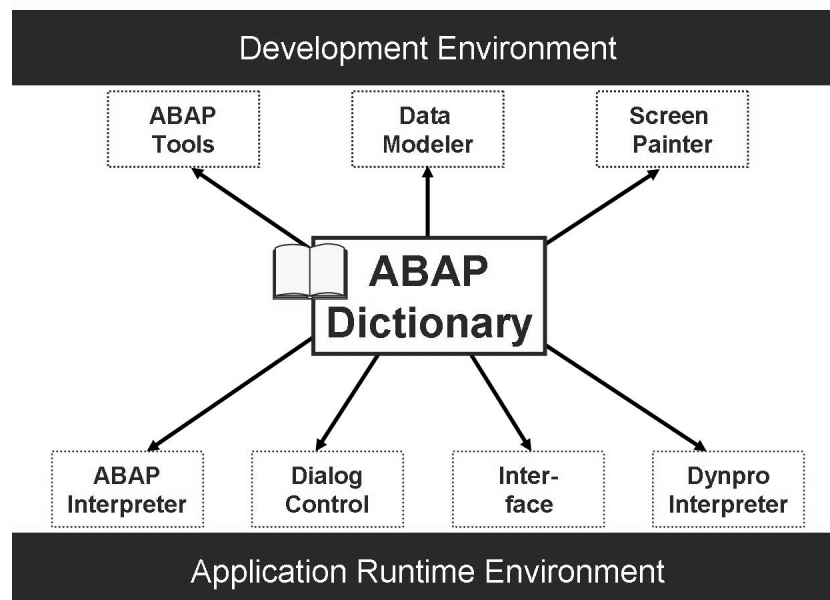


Figure 207: Significance of the ABAP Dictionary

The integration of the *ABAP Dictionary* into the program flow is based on the interpretative method of the *SAP Web AS* runtime environment. Instead of working with the original of an ABAP program, the ABAP processor interprets a runtime

object generated from the program text prior to its first execution. Runtime objects are automatically generated again before execution if a time stamp comparison reveals that they are no longer consistent with the current status of the *ABAP Dictionary*.

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

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



Hint: Every database system also contains a so called dictionary. This is **not** the dictionary referred to in this lesson.

Appendix: Table Definition and the Two-Level Domain Concept

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

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

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

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

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

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



Hint: If you want to check where in the SAP system a particular data definition (data element, domain, table, or similar) is used, then you can look in the Where-used list in transaction SE11 for that data definition.

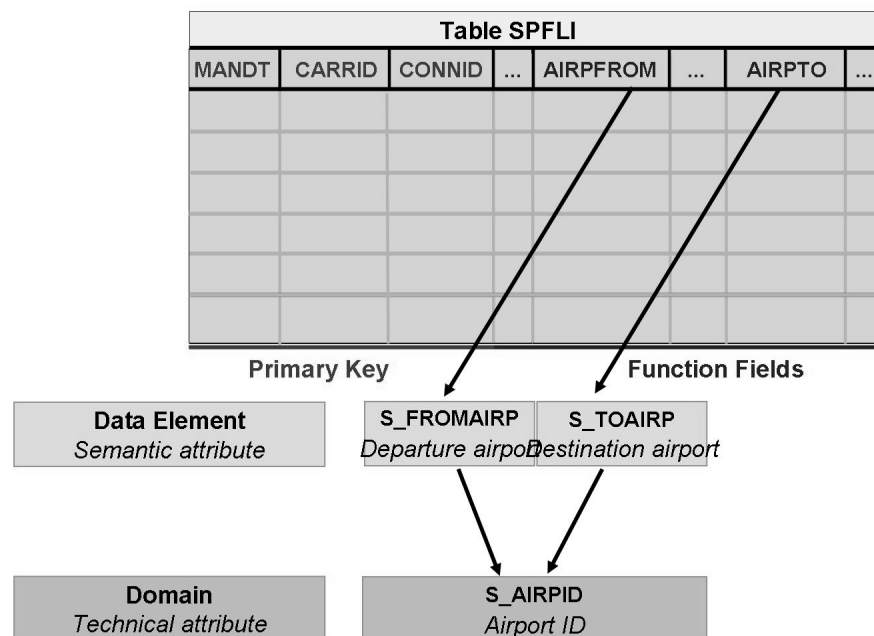


Figure 208: Appendix: Table definition and domain concept

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

Exercise 24: Accessing and Editing ABAP Repository Objects

Exercise Objectives

After completing this exercise, you will be able to:

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

Business Example

As an ABAP developer, you create your own ABAP program and a Business Server Page (BSP). You continue to view the table definitions and contents.

Task 1: Working with Change Requests

Checking your own change requests and tasks

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

Task 2: Writing a Short ABAP Program

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

1. Start the *ABAP Editor* (transaction SE38 or SE80). Name your program **Z<CLNT>PROGRAM_##**, where <CLNT> is the client you are logged on to, and ## is your group number.

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

2. On the selection screen that appears, either choose *Local Object* or enter the name of a package specified by your instructor. If you do not save your program as a “Local object”, you also need to specify a transport request. For this, choose the request you displayed in task 1.
3. This takes you to the *ABAP Editor* where you can write your program text (if necessary, after choosing *Display* ↔ *Change*). Choose *Return* after the final period in the line

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

Continued on next page

then go to the next line and enter, for example,

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

4. Then choose the following pushbuttons in this order: *Save*, *Check*, *Activate* and *Execute (F8)*. Your ABAP program should now run without errors.

Result

Congratulations on creating your first ABAP program!

Task 3: Creating a Simple Business Server Page (BSP)

Display simple output from your SAP system in your browser.

1. Call transaction SE80 and create a BSP application with the following name: **Z<CLIENT>_BSP_##**. <CLNT> is the client you are logged on to and ## is your group number. The application will therefore have the following name, for example: Z100_BSP_01
2. Enter the name of the page **Hello_##.htm**. ## again corresponds to your group number.
3. Optional: On the *Layout* tab page, you can change the source code, for example by replacing the text `Hello World!` (within the quotation marks) with this text `Hello`. Make sure that you text is still in quotation marks.
4. Save and activate all objects you have created. Then carry out a “test” using the right mouse button. You need to enter your user ID and password for the training client. The client is predefined (see below) and the language is the original language of the object called.

Result

Congratulations, you have created your first BSP!

Task 4: Displaying Table Contents

Display the content of the table TADIR.

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

Continued on next page

Task 5: Optional: Displaying Table Definitions

Analyze the definition of table TADIR.

1. Find out which fields make up the primary key of table TADIR.
2. What is the name of the field that has the short text *Package*?
3. To which data element and to which domain is this field assigned?
4. In which data elements is the domain *DEVCLASS* used? (Tip: Select the domain name and then run a where-used list.)

Task 6: Releasing a Request

Release your task of the change request.

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

Solution 24: Accessing and Editing ABAP Repository Objects

Task 1: Working with Change Requests

Checking your own change requests and tasks

1. Display all change requests to which you are assigned. How many tasks are assigned to these requests?
 - a) Call the *Transport Organizer* using *Tools* → *ABAP Workbench* → *Overview* → *Transport Organizer* or use the transaction code SE09: On the initial screen of the *Transport Organizer* ensure that your name is entered in the *user* input field and that the *Workbench Requests* field is selected.
 - b) You can use the *Display* pushbutton to display the change requests to which you are assigned as developer. If your instructor has created one change request for all course participants, you should see an entry for the change request for which there are several tasks (one for each group of participants).

Task 2: Writing a Short ABAP Program

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

1. Start the *ABAP Editor* (transaction SE38 or SE80). Name your program **Z<CLNT>PROGRAM_##**, where <CLNT> is the client you are logged on to, and ## is your group number.

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

 - a) Call *ABAP Editor* using transaction SE38 (menu path *Tools* → *ABAP Workbench* → *Development* → *ABAP Editor*).
 - b) Enter your program name **Z<CLNT>PROGRAMM_##** and choose *Create*. Create the program with a meaningful title and the attributes specified in the task. Choose *Save*.
2. On the selection screen that appears, either choose *Local Object* or enter the name of a package specified by your instructor. If you do not save your program as a “Local object”, you also need to specify a transport request. For this, choose the request you displayed in task 1.
 - a) For the procedure, see the exercise.

Continued on next page

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

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

then go to the next line and enter, for example,

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

- a) For the procedure, see the exercise.
4. Then choose the following pushbuttons in this order: *Save*, *Check*, *Activate* and *Execute (F8)*. Your ABAP program should now run without errors.
 - a) For the procedure, see the exercise. *Save* saves the program in the database. *Check* executed a syntax check. *Activate* generates a runtime object so that the generated program can now also be executed by people outside the ABAP Workbench.

Result

Congratulations on creating your first ABAP program!

Task 3: Creating a Simple Business Server Page (BSP)

Display simple output from your SAP system in your browser.

1. Call transaction SE80 and create a BSP application with the following name: **Z<CLIENT>_BSP_##**. <CLNT> is the client you are logged on to and ## is your group number. The application will therefore have the following name, for example: Z100_BSP_01
 - a) Carry out the following actions in transaction SE80:
 1. Select the object type “BSP Application” from the possible entries help for the “Object Type” field.
 2. In the field beneath it, enter the name of your object following the **Z<CLNT>_BSP_##** pattern. Choose *Return*.
 3. Confirm that you want to create the object.
 4. Enter a short description of your choice for the object and choose *Save*. ..
 5. Either assign the object to a package specified by your instructor, or save it as a “Local object”.

Continued on next page

2. Enter the name of the page **Hello_##.htm**. ## again corresponds to your group number.
 - a) Proceed as follows:
 - Right mouse click the object name and choose *Create* → *Page*. Enter the following name as the *page* characteristic: **Hello_##.htm**. ## again corresponds to your group number.
 - Choose *Page with Flow Logic* as the page type.
 - Enter a meaningful description and choose *Continue*.
3. Optional: On the *Layout* tab page, you can change the source code, for example by replacing the text `Hello World!` (within the quotation marks) with this text `Hello`. Make sure that you text is still in quotation marks.
 - a) See exercise text.
4. Save and activate all objects you have created. Then carry out a “test” using the right mouse button. You need to enter your user ID and password for the training client. The client is predefined (see below) and the language is the original language of the object called.
 - a) Choose *Save*, followed by *Activate*. Select all objects when you are activating.

Navigate to “your” page and choose the entry *Test* from the right mouse-click context menu. Enter “your” user ID and password for the training system if you are prompted to do so. Note that the description you entered (see above) appears in the title bar of the browser.

Result

Congratulations, you have created your first BSP!

Task 4: Displaying Table Contents

Display the content of the table TADIR.

1. Find out how many programs are in your SAP system Repository.
 - a) To do this, call transaction SE16 (*Data Browser, Tools* → *ABAP Workbench* → *Overview* → *Data Browser*), enter the table name TADIR on the screen and choose *Enter*. To display the number of programs, on the following selection screen, enter the value PROG (for programs) in the *OBJECT* field and choose *Number of Entries*.

Continued on next page

2. Find out which of these programs begin with “RSP”.
 - a) Further limit your search on the selection screen by entering the value **RSP*** in the *OBJ_NAME* field and choosing *Execute*.

Task 5: Optional: Displaying Table Definitions

Analyze the definition of table TADIR.

1. Find out which fields make up the primary key of table TADIR.
 - a) To do this, call transaction SE11 (*ABAP Dictionary, Tools → ABAP Workbench → Development → ABAP Dictionary*), choose the *Database table* radio button, enter the table name, and choose *Display*. The primary key consists of the fields *PGMID*, *OBJECT* and *OBJ_NAME*.
2. What is the name of the field that has the short text *Package*?
 - a) You can use *Find* on the standard toolbar to search for the name *Package*. The field name for package is *DEVCLASS*.
3. To which data element and to which domain is this field assigned?
 - a) The *DEVCLASS* field is assigned to the data element *DEVCLASS*. Double-clicking the name of the data element displays the definition of the data element *DEVCLASS*. You can see that this data element refers to the domain *DEVCLASS*. Although the names of the data element and the domain are identical in this case, data elements and domains serve different purposes in the SAP system.
4. In which data elements is the domain *DEVCLASS* used? (Tip: Select the domain name and then run a where-used list.)
 - a) In the definition of the data element *DEVCLASS* from the previous exercise, double-click on the *DEVCLASS* domain name to display the definition of the domain. Here you can use the *Where-used list* pushbutton to run a where-used list in which you are only looking for uses in data elements. The result is a list of data elements that refer to the *DEVCLASS* domain.

Continued on next page

Task 6: Releasing a Request

Release your task of the change request.

1. Use the *Transport Organizer* to release your task for the change request.
 - a) Access the *Transport Organizer* again (either using menu path *Tools* → *ABAP Workbench* → *Overview* → *Transport Organizer* or directly with the transaction code SE09).
 - b) On the initial screen of the *Transport Organizer* choose your change request which is set to “modifiable” and choose *Display*. On the next screen, select the task and then choose *Release directly*.



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



Lesson Summary

You should now be able to:

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

Related Information

More information on the ABAP Workbench is provided in the leading courses for ABAP-Workbench (starting with courses **BC400 - ABAP Workbench Basics** and **BC430 – ABAP Dictionary**) as well as in the SAP Library for SAP NetWeaver 7.0 in the area *SAP Library* → *SAP NetWeaver Library* → *Key Areas of SAP NetWeaver* → *Key Areas of the Application Platform* → *ABAP Technology* → *ABAP Workbench (BC-DWB)*.

Lesson: Introduction to the SAP NetWeaver Development Infrastructure (NWDI)

Lesson Overview

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

This lesson first of all provides an overview of the steps that development has to make to execute a Java application. Following that, typical problems that arise in a team-oriented development in the Java environment are discussed. These problems should largely be avoided with the *SAP NetWeaver Development Infrastructure (NWDI)*, by transferring known and approved concepts from the ABAP world into the Java environment. This lesson will give an overview of development with the NWDI. The lesson concludes with an outlook at enhancing the ABAP transport mechanism with which it is also possible to transport non-ABAP objects.



Lesson Objectives

After completing this lesson, you will be able to:

- Name the individual components of the SAP NetWeaver Development Infrastructure
- List the main steps in terms of software development with SAP NetWeaver Development Infrastructure

Business Example

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

The Java Programming Language

Java is an object-oriented programming language, similar to C++ or C#. However, Java is not fully object-oriented: the basic data types (int, boolean etc.) are not objects.

Java makes it possible to write classes which can be executed in a variety of runtime environments. In this way, it is possible, for example, to run applets in Web browsers which support Java. Java possesses an extensive class library. This provides

programmers with a uniform, operating system-independent interface (application programming interface, API). With the help of Remote Method Invocation (RMI) it is also possible to call up objects on other computers.

Source codes written in Java are first of all translated by the Java compiler *javac* into an intermediary code that is independent of architecture, the so-called byte code. A class file is created from every source file (.java file). This byte code cannot be executed directly, but is either interpreted and executed by a so-called virtual machine or (using HotSpot technology, available as of Java Runtime Environment 1.3) and is compiled and executed as native processor code at runtime. This virtual machine must have been developed for every supported platform, if necessary, separately.

The Java runtime environment consists of three main components:

- Class Loaders for loading all the classes required for the execution of the program. The Class Loaders control the reliable provision of class information to the Java Virtual Machine
- Bytecode Verifier to check whether the loaded classes are compatible with specification of the virtual machine. This ensures that the Java Virtual Machine is not able to execute any invalid byte code.
- The Java Virtual Machine itself

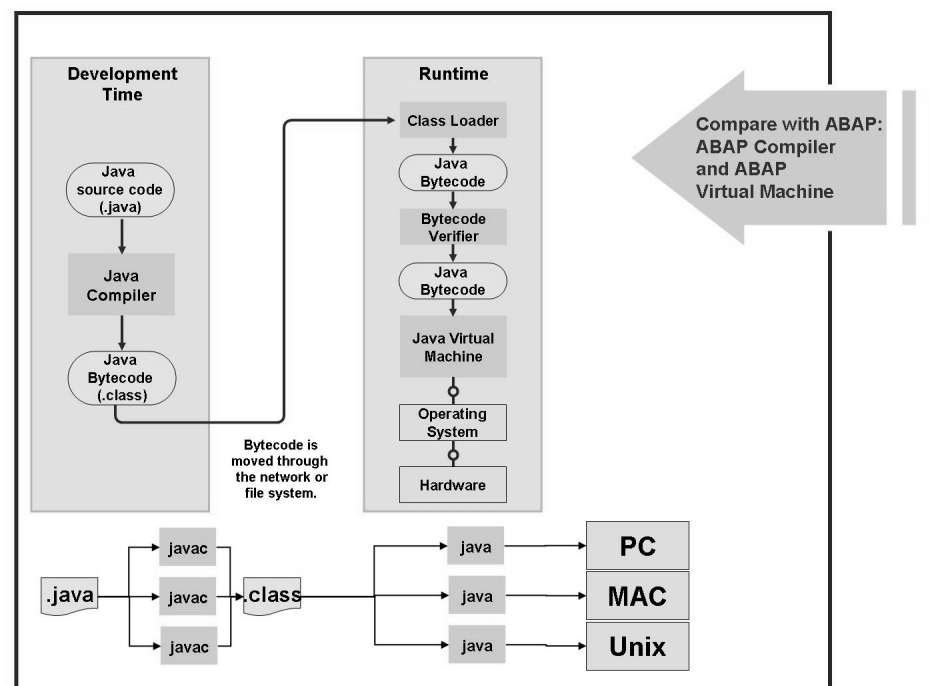


Figure 209: From the development to the execution of a Java application

If a Java program is to be executed, a runtime environment (Java Runtime Environment, JRE) is required. The JRE consists of the Java Virtual Machine, the standard Java interfaces and other components which are required to execute Java applications and applets. The Java Virtual Machine is responsible for the independence of the J2EE platform from the hardware and the operating system.



Hint: J2EE stands for Java 2 Enterprise Edition. This is a set of rules that must be fulfilled by applications if they wish to be regarded as “J2EE compliant” and if they are to be executed on a J2EE Server (which must also comply with these rules).

In the Java 2 Software Development Kit – also known as Java Development Kit (JDK) or Software Development Kit (SDK) are - compared with the JRE - some additional tools with which Java applications can be developed. These tools include the Java Compiler and the Java Debugger.

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

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

Enhancements to the J2EE Standard for SAP Applications

J2EE applications consist of components. A J2EE component is a complete, functional software entity, which is combined with the associated classes and files to form a J2EE application (assembly), and which can communicate with other components. The J2EE specification differentiates between the following components:

- Applets (presentation components that run on the client)
- Servlets and Java Server Pages (JSP): Presentation or J2EE web components that run on the server)
- Enterprise Java Beans (EJBs): Business logic components that run on the server).

The communication between the J2EE server and the frontend is essentially based on Web standards such as HTTP, HTML, or XML. Communication with the database or backend layer is conducted on the JDBC (Java Database Connectivity) basis. JDBC is a database interface of the Java platform that provides a uniform interface to databases from different manufacturers and has been specially designed for relational databases.

To adapt the structures specified by the J2EE standard for larger business processes, SAP has expanded the programming model (Java Web Dynpro) and also made new/additional features available (such as Open SQL for Java or SQLJ). In addition open standards have been integrated (such as Web Services).

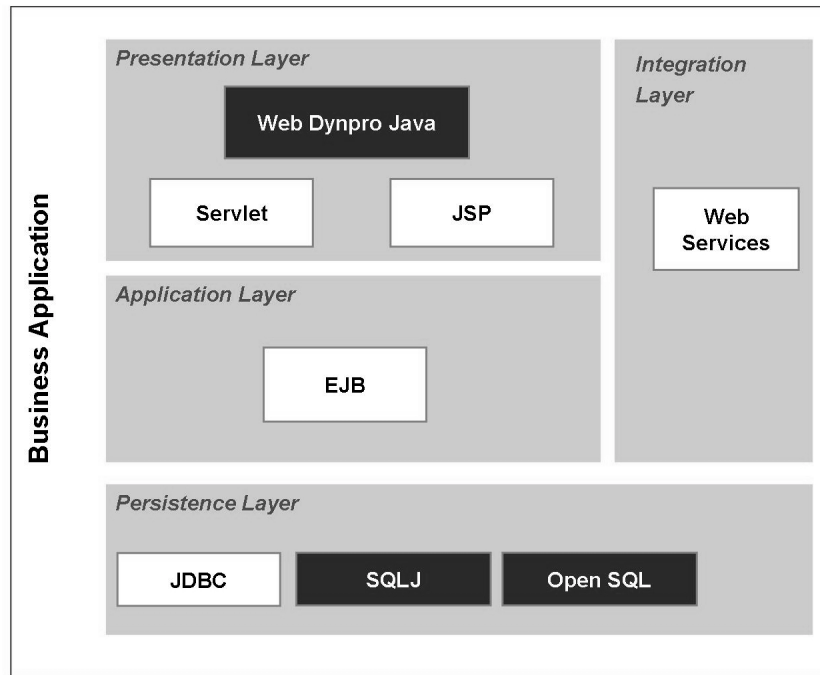


Figure 210: Enhancements to the J2EE Standard for SAP Applications



Hint: Java Web Dynpro applications follow the Model View Controller paradigm. At the least, they contain all logical units that are obligatory for the application: The user interface, a model for the provision of data or business functions and the controller for the data flow between all application units.

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

In the following figure, the component parts of a standard J2EE application are shown separately from the additions developed by SAP (highlighted in color).

Differences Between ABAP and Java Development in the SAP Environment

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

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

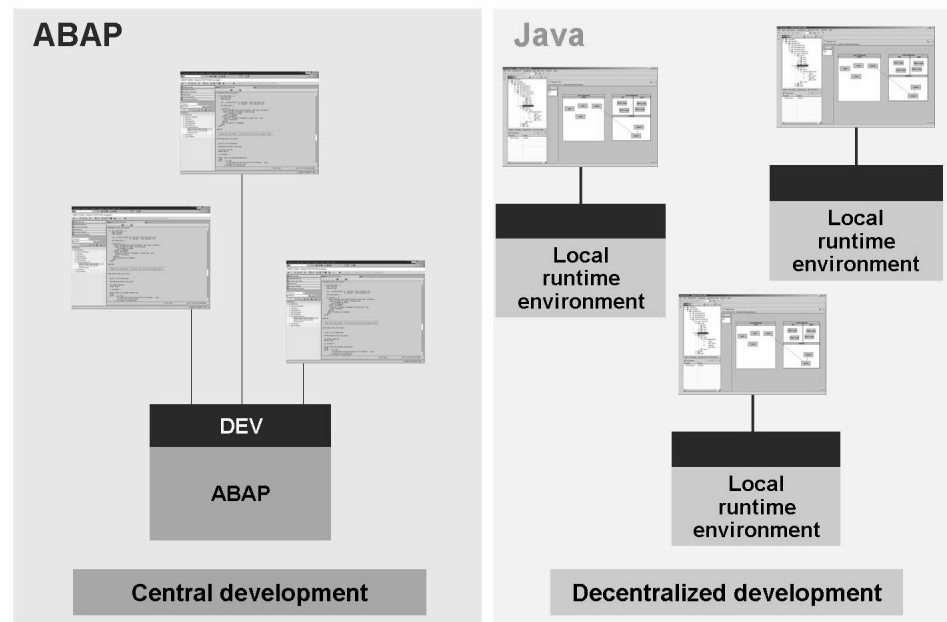


Figure 211: Different development philosophies

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

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

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

Motivation and Aims of the SAP NetWeaver Development Infrastructure (NWDI)

The Java 2 Software Development Kit (JDK / SDK) provides, in principle, all of the tools required to develop smaller Java applications - in addition to the runtime environment, programs such as the Java Compiler or a debugger are also supplied. For more convenient development of applications, there are also various local development environments (Integrated Development Environments, IDE) based on the JDK / SDK.

However, new challenges arise when developing applications in large development teams, which work decentralized (as is usually the case in the Java environment). The following illustration shows the typical development process in the Java environment and shows some of the difficulties that have to be dealt with. These difficulties often have to do with the fact that the development takes place locally in the runtime environment of the individual developer and this development environment does not correspond in every detail to the central test environment.

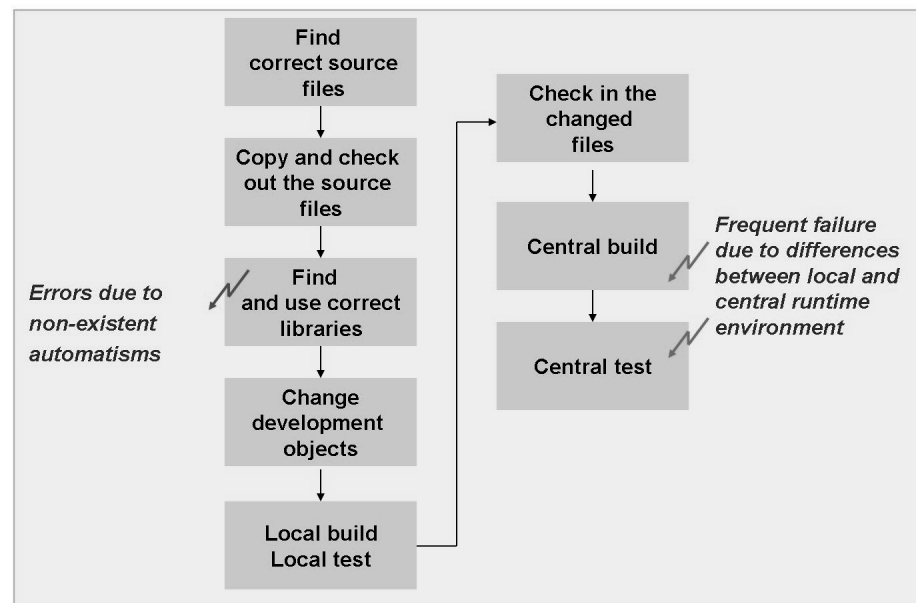


Figure 212: Typical problems in a team-oriented development in the Java environment

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

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

Comparison of the ABAP and Java Development Environments

To develop and execute applications, regardless of the selected programming language, you require at least an editor, a compiler (or interpreter) and a runtime environment. In addition to this you require a transport mechanism (which can consist of a single “Copy & Paste” process) if the applications are to run on other computers at a later stage. Especially for developments in which several developers are participating, a source code administration and versioning system is also desirable.

All of these requirements are met in the AS ABAP area. The Object Navigator (transaction SE80) is available as an editor (as entry point). ABAP is an interpreted (and non-compiled) language. The ABAP Interpreter is a component part of every ABAP work process. Therefore the ABAP kernel (instance) generally functions both as an interpreter and as a runtime environment. The ABAP Transport Management System (transaction STMS) is available for transports between the development environment and the production environment, the ABAP Workbench provides source code versioning in the corresponding development system.

In the Java Software Development Kit (SDK 1.4.2) (for example, as provided by Sun), as it is used for AS Java in SAP NetWeaver 7.0, a compiler (javac) and a runtime environment (the Java Virtual Machine, Java VM) are provided. However, there is no actual editor, source code administration system and transport mechanism. These areas are also not specified exactly in the Java standard (J2SE Standard and J2EE

Standard). The Java developers are therefore required to use commercially available tools (for example for editors) or to develop their own tools / processes (for example for transport mechanisms).

This is where SAP and the SAP NetWeaver Development Infrastructure (NWDI) comes in. NWDI provides an editor, a source code administration system and a transport mechanism.



Comparison Between ABAP and Java Development in the SAP Environment

	AS ABAP	AS Java
Editor	Object Navigator (SE80)	SAP NetWeaver Developer Studio
Compiler / Interpreter	Kernel (instance)	Component Build Service (CBS)
Runtime Environment	Kernel (instance)	Kernel (instance)
Source code administration and versioning	Versioning in the development system	Design Time Repository (DTR)
Transport mechanism	Transport Management System (STMS)	Change Management Service (CMS)

SAP NetWeaver Developer Studio is used as an editor, the Design Time Repository (DTR) functions as a central source code administration and versioning system and the Change Management Service (CMS) is used for transport from the development system to other systems. In addition to these components, the Component Build Service (CBS) enables central building. These components will be described in detail in the following sections.

The SAP NetWeaver Developer Studio

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



- ▶ Eclipse provides all of the tools required to develop Java programs. For J2EE projects, additional functionality is required in the SAP environment, however (for example, integration with the application server), that is not provided by Eclipse.

- ▶ The SAP NetWeaver Developer Studio adds new functions to Eclipse using plug-ins.

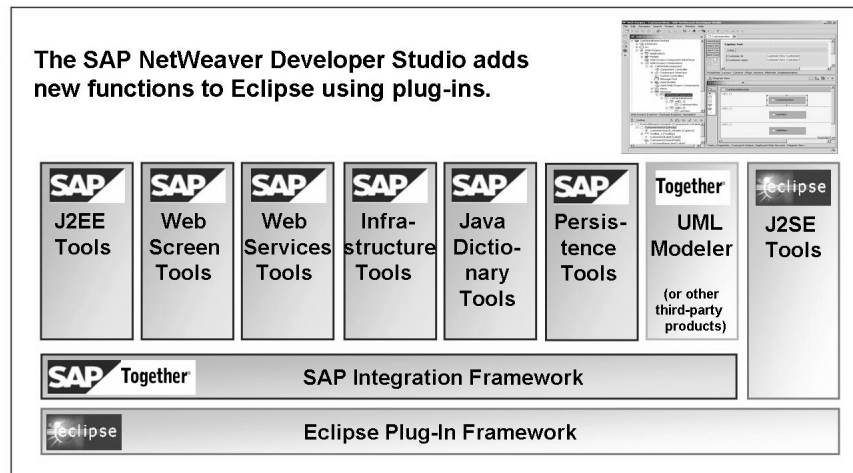


Figure 213: Structure of the SAP NetWeaver Developer Studio

The SAP NetWeaver Developer Studio provides a range of tools for all aspects of application development, a few of which are listed below as examples:

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

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

Aims of the SAP NetWeaver Development Infrastructure (NWDI)

In the ABAP environment, the problems described above do not arise due to the central development environment (all developers work in a central runtime environment and each use the current version of the respective development object). The goal of the SAP NetWeaver Development Infrastructure – NWDI for short below - is therefore to avoid the problems described above by transferring known and approved concepts from the ABAP world into the Java world. Here, SAP builds on

the known standards (J2EE or WebDAV and DeltaV as a Repository standard for the access and the versioning of any development objects). The local development environment is based on the open source development tool *Eclipse*.

Structure of the NWDI

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

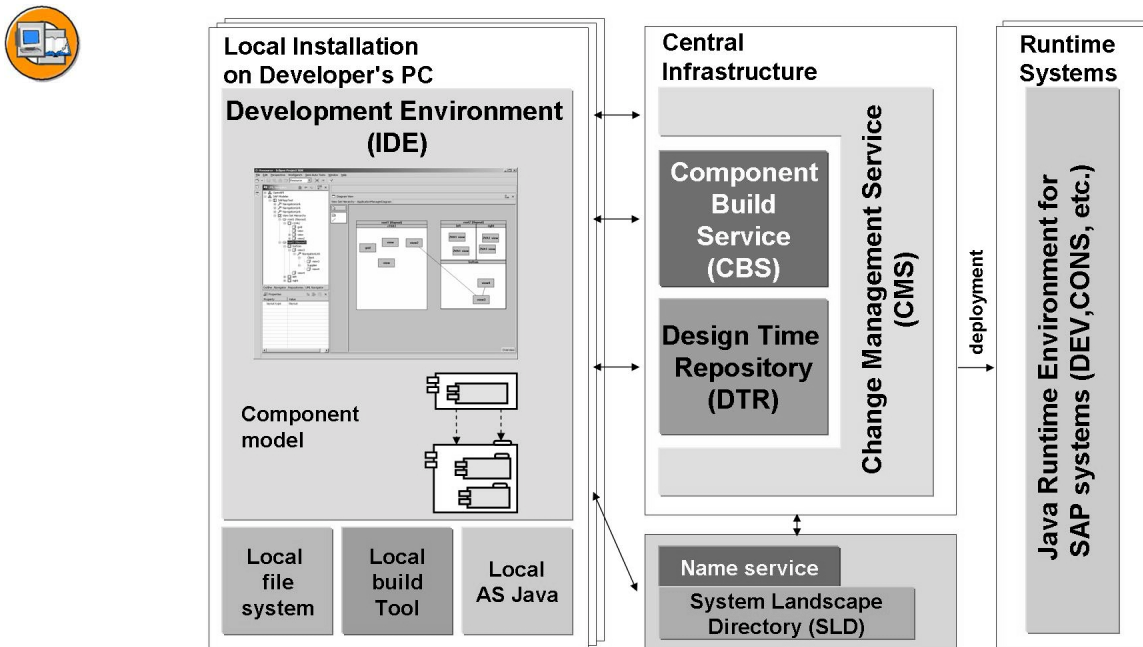


Figure 214: Overview of the SAP NetWeaver Development Infrastructure

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

On the side of the central infrastructure, you can find (among others) the *Design Time Repository*, the *Component Build Service*, and the *Name Server* as part of the *System Landscape Directory*.

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

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

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

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

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

The following list summarizes the components of NWDI and their characteristic properties:



- **Design Time Repository (DTR)**
 - Central source code administration and versioning
- **Component Build Service (CBS)**
 - Build on demand
 - Central storage of Build results and Build tools
 - Activation concept
- **Change Management Service (CMS)**
 - Central administration of the entire transport landscape
 - Transports between the individual development levels (development, consolidation, test, production)
- **System Landscape Directory (SLD)**
 - Overview of systems, installed products and software versions (releases, support packages)
- **Name Service**
 - To avoid naming conflicts.

The SAP Component Model

Using components means that software projects can be systematically structured into manageable and reusable units from the start. The following table lists the elements of the component model for ABAP or Java development:



SAP Component Model for ABAP and Java Development

	Example for AS ABAP	Example for AS Java	Note
Product (version)	SAP NetWeaver (SAP NetWeaver 7.0)		–
Software unit (with version)	SAP NetWeaver AS ABAP 7.0	SAP NetWeaver AS Java 7.0	Smallest installation unit / upgrade unit

Software component (SC) with version	SAP_BASIS 7.00	SAP Java Tech Services 7.00	Maintenance using support packages
Package (ABAP) / Development Component (Java)	STUN	com.sap.lcr	–
Development object	RSUSR000	only supplied as byte code for this SC	Error correction using SAP Notes (ABAP)

A development component (abbreviation: component or DC) is – simply put – a common container for a set of objects that are part of the software. A component has a well-defined external interface and an internal life that is not visible from outside. Components can use each other by referring to the public interfaces of other components. Due to these properties, components are the elementary reusable units of the model.



Hint: In ABAP development, the individual development objects are bundled together to form packages (instead of development components).

A development object is an element of a component that provides part of its function and can be changed or developed in some way. This can be a Java class, a table definition, an ABAP program, and so on. Development objects are essentially stored in a repository.

Software components combine components (DCs or ABAP packages) for delivery and maintenance to larger units. A release is a large step in the development of a software component, which provides new functionality. For the purposes of maintaining software components support packages are provided.

Software Units group software components for the purposes of installation, maintenance or upgrading. They form the smallest installation unit.



Hint: In SAP NetWeaver software units correspond roughly to usage types.

A product consists of one or more software components that represent related business processes. Products are generally installation and upgrade units.

Appendix: Overview of the Development Process

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

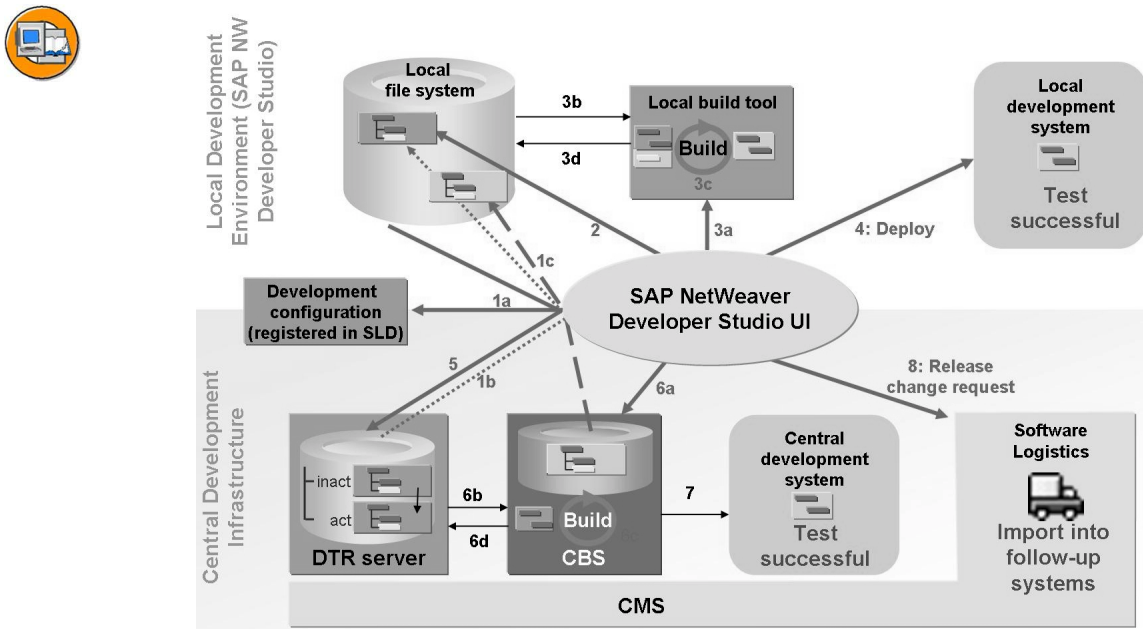


Figure 215: Appendix: Java Development Process using the NWDI

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

At the beginning of development, the developer logs on to a development configuration from the developer studio. Logging on to a development configuration corresponds to logging on to the development system in the ABAP environment, that is, the developer can automatically access the required sources and archives in the correct version (step 1a). Afterwards, the required source files from the *DTR* and the required archives (for example, from other used software components) from the *Component Build Service (CBS)* are stored on the developer's PC (steps 1b and 1c). The local development can then begin. The source files are created or revised here (step 2). The sources can be built locally from time to time (to do this, call the local build process from the studio - step 3). The runtime environment of *SAP NetWeaver AS Java*, which is located on the developer's PC, can also be tested locally (step 4, but a deployment to the corresponding runtime environment is required beforehand).

After completing the local development, the changed objects are checked in to the DTR (under a new version number) - step 5. After that, the developer can trigger a central build on the central development system (DEV) from the studio (step 6). After successfully building the entire application centrally, it can also be tested centrally, that is, in the runtime environment of the central development system (step 7).

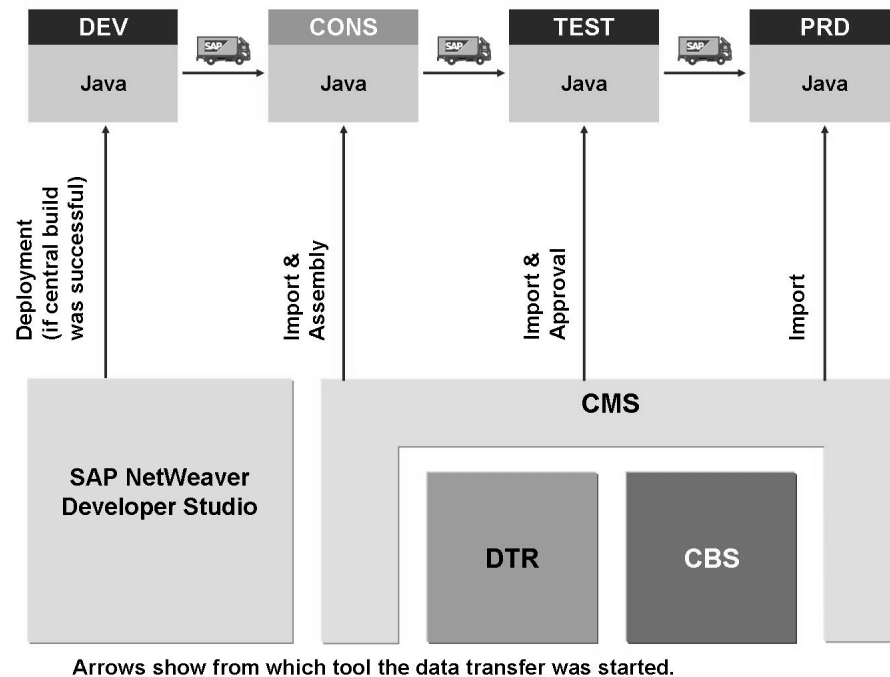
If all tests of the developer were successful, he can release his changes for import into follow-up systems. This corresponds to releasing the task and the change request in the ABAP environment (step 8). The developer's work is now completed. Subsequently, the source files waiting to be imported can be imported centrally into the following systems (development stages) by using the Change Management Service (CMS).

Digression: The Four-Stage Landscape

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



Hint: Note that a physical runtime system does not have to exist in the background for each of these four stages. In the following figure, for example, a runtime system is not assigned to the CONS stage.



Arrows show from which tool the data transfer was started.

Figure 216: Digression: The Four-Stage Landscape for Java Developments

The central development system/development stage (DEV) is used by the individual developers to test the local developments in a larger context, that is, in conjunction with changes made by other developers. The consolidation stage (CONS) and the test stage (TEST) are used to consolidate a certain fixed status of a software component or for the final test (also in the sense of an integration test). Once a test has been successful, the development is finally imported into the productive system/productive stage (PROD).

CMS triggers and controls the transport of data between the individual stages.

Outlook: Enhancing the ABAP Transport System for the Transport of Non-ABAP Objects

As you have seen in this lesson, tools are available to transport both ABAP and Java objects. However, these tools have some restrictions. For example, there is no automatic synchronization for application transports based on both AS ABAP and AS Java (for example, usage type PI). In this case, various tools (ABAP TMS and Java NWDI) are required to transport parts of the same application. Furthermore, portal transports were not controlled centrally.

With SAP NetWeaver 7.0 Support Package Stack 12, the ABAP Change and Transport System (CTS) was enhanced, so that it can now transport non-ABAP objects. As a result, a transport synchronization between ABAP and Java transports is now possible. Furthermore, portal content can also be transported in this way.

Prerequisite for this is centralized AS ABAP + Java with SAP NetWeaver 7.0 SP Stack 13 or higher as a central SAP system, where the enhancement of the Change and Transport is configured.



- Extension of the ABAP Change and Transport System (CTS) with SAP NetWeaver 7.0 SP Stack 13 to make it possible to transport non-ABAP objects
- Transport synchronization between ABAP and Java transport possible
- Portal Content can also be transported in this way.

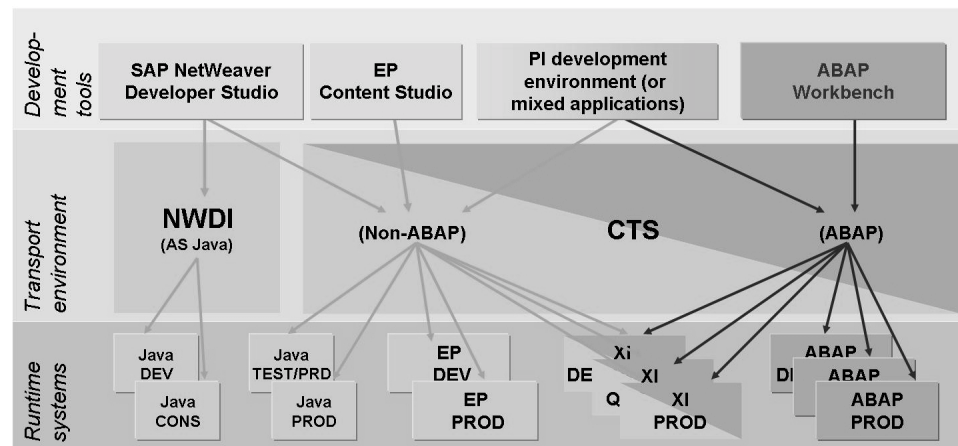


Figure 217: Outlook: Transport synchronization between ABAP and Java transports by enhancing the ABAP transport system.

This enhancement will affect transport management in SAP system landscapes. Therefore, you should take account of this new option when planning transport routes in a large system landscape.



Hint: Note that none of the transport scenarios that already exist become invalidated as a result of enhancing the CTS. Furthermore, the SAP NetWeaver Development Infrastructure (NWDI) will remain the recommended infrastructure for (most) non-ABAP development objects. Nevertheless, we recommend that you use the ABAP transport system for combined ABAP + Java transports in your system landscape. You can also use the NWDI in connection with the ABAP CTS; after the Java consolidation stage, you can transfer the development result (SCA file) to the ABAP transport for onward transport.



Lesson Summary

You should now be able to:

- Name the individual components of the SAP NetWeaver Development Infrastructure
- List the main steps in terms of software development with SAP NetWeaver Development Infrastructure

Related Information

You can find an overview of the SAP NetWeaver Development Infrastructure in the online documentation for *SAP NetWeaver 7.0* under *SAP Library* → *SAP NetWeaver Library* → *SAP NetWeaver Developer's Guide* → *Fundamentals* → *Using Java* → *Working with the Development Infrastructure* (only available in English at the time of creating this training guide).

The ADM200 – *Administration AS Java* and ADM225 – *SAP Software Logistics for Java* training courses provide more information on software development in the AS Java environment.



Unit Summary

You should now be able to:

- Describe the data structure of the ABAP Stack in an SAP system
- Describe the three-system landscape as recommended by SAP
- Explain the importance of transports
- Describe the procedure for transports
- Name important tools in the *ABAP Workbench*
- Access ABAP Repository objects
- Write a simple ABAP program
- Explain the significance of the *ABAP Dictionary*
- Outline the two-level domain concept
- Name the individual components of the SAP NetWeaver Development Infrastructure
- List the main steps in terms of software development with SAP NetWeaver Development Infrastructure



Test Your Knowledge

1. What data is contained in a client?
Choose the correct answer(s).
 - ☐ A Cross-client Customizing, user data, and application data
 - ☐ B Customizing and Repository objects
 - ☐ C Client-specific Customizing, user data, and application data
 - ☐ D Only application data entered in this client

2. The data structure of the ABAP Stack in an SAP system consists of
Choose the correct answer(s).
 - ☐ A Application data and Repository
 - ☐ B Client-specific data, cross-client Customizing, and the Repository
 - ☐ C Client-specific data and the Repository
 - ☐ D Client-specific data and cross-client Customizing

3. Why does SAP recommend a three-system landscape?
Choose the correct answer(s).
 - ☐ A Because of the special data structure in SAP systems
 - ☐ B To enable preparation and testing of upgrades
 - ☐ C To sell more licenses
 - ☐ D Because you can only create one client per system

4. At the end of an (ABAP) development project, the tasks and change requests must be released so that they can be exported. Who releases which objects?
Choose the correct answer(s).
 - ☐ A Developers release their tasks
 - ☐ B The development leader releases the request as a whole
 - ☐ C The request is released automatically

5. In the ABAP environment, you can carry out the entire software development cycle in the SAP system using the ABAP Workbench tools.
Determine whether this statement is true or false.
 - ☐ True
 - ☐ False

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

Choose the correct answer(s).

- ☐ A For defining the technical attributes of a field
- ☐ B For adding foreign key dependencies
- ☐ C For defining the semantic attributes of a field
- ☐ D For entering data in a table

7. Which of the following services are elements of the SAP NetWeaver Development Infrastructure (NWDI)?

Choose the correct answer(s).

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



Answers

1. What data is contained in a client?

Answer: C

A client contains client-specific Customizing, user data, and application data.

2. The data structure of the ABAP Stack in an SAP system consists of

Answer: B

the ABAP Stack of an SAP system consists of clients, cross-client Customizing, and the Repository.

3. Why does SAP recommend a three-system landscape?

Answer: A, B

Recommending the three-system landscape does not sell more licenses, since you can set up several SAP systems on one license, although you can only use one of these systems as the production system. You can always create several clients in an SAP system.

4. At the end of an (ABAP) development project, the tasks and change requests must be released so that they can be exported. Who releases which objects?

Answer: A, B

Every developer releases his or her own subtasks. Then the development leader releases the request as a whole. Requests cannot be released automatically.

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

Answer: True

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

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

Answer: A

You use domains for defining the technical attributes of fields in a table. The semantic attributes of a field are defined in the data element.

7. Which of the following services are elements of the SAP NetWeaver Development Infrastructure (NWDI)?

Answer: A, C, D

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



Course Summary

You should now be able to:

- To process basic tasks within the technology environment of SAP systems

Related Information

- Use a URL or a cross-reference tag to point out additional information that the participants may find useful such as Web sites or White Papers. Delete this if it is not relevant.

Glossary

ABAP

ABAP (Advanced Business Application Programming) is a programming language developed by SAP.

Advanced Business Application Programming. SAP's own programming language for developing application programs.

ABAP Dictionary

Central redundancy-free information store in the SAP system for the logical structures of application development objects (such as tables, views, and data types) and for their representation in the structures of the relational database used.

ABAP Editor

ABAP Workbench program editor. You can use the ABAP Editor to create, test and modify ABAP programs, function modules, screen flow logic and logical databases. You can enter source text and carry out syntax checks.

ABAP Workbench

The SAP system's integrated graphical development environment. The ABAP Workbench supports the development, modification, testing and administration of client/server applications written in ABAP.

Access Method

The access method describes the connection between a spool work process and the operating system spool.

activation

Procedure that makes an object available to the runtime environment. During activation, runtime objects are generated, which the application programs and screen templates access.

Activation

Process that makes a runtime object available. The effect of activation is to generate runtime objects, which are accessed by application programs and screens.

background job

A background job describes steps that are to be executed in the system without user interaction (in the background). Background jobs are processed by background work processes.

Buffer

Area in the main memory of an instance in which data frequently used by applications can be temporarily stored.

Business Add-In

The location in a program defined by the developer at which software recipient layers (industries, partners, customers and so on) can insert coding without modifying the original object.

Business Server Page

A Business Server Page (BSP) is a storage unit (HTML, XML) that may contain server-side scripting and that is developed in the *SAP Web Application Server* development environment, the *Web Application Builder*. A BSP is therefore part of a BSP application and contains the layout part, all event handlers, page attributes, and type definitions.

Central system

The central system is an SAP system where the central instance, central services instance (only for AS Java) and the database are installed on the same computer.

Change Management Service (CMS)

Part of the NetWeaver Development Infrastructure, which is responsible for the management of the software lifecycle. It includes the configuration of development landscapes, the development infrastructure and the transport of software changes.

change request

An information source in the Transport Organizer that records and manages all alterations made to Repository objects and Customizing settings during a development project.

Client

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

Software component that uses the services provided by a server (software-oriented view) or work station for which the server provides resources (hardware-oriented view).

Command field

You can start applications directly by entering the transaction code in the command field.

Component Build Service (CBS)

The *Component Build Service (CBS)* is part of the SAP NetWeaver Development Infrastructure and is used for the central Build of the source texts.

Computing Center Management System (CCMS)

Integrated tools for monitoring and administering SAP systems and independent SAP business components, with which you can, for example, automate operations such as resource distribution and the management of SAP databases.

Customizing

Customizing is the overall procedure for setting up one or more SAP systems. This procedure is directed toward adapting the standard, industry-specific SAP system functions to a company's particular business requirements. Customizing is obligatory both during the first installation and during an upgrade and is performed in the SAP system using the Implementation Guide (IMG).

Data Element

ABAP Dictionary object that describes the data type and semantic meaning of a table field or structure field.

Database Interface

Component of a work process that connects it to the database. The database interface translates ABAP Open SQL into database-specific SQL, thereby enabling communication with the database.

database transaction

Non-divisible sequence of database operations, that must be either executed completely, ending with a database commit, or not at all.

Design Time Repository (DTR)

Central memory for all kinds of source files. The memory is presented logically as a hierarchical file and folder structure. Physically, the contents are stored in a database and are accessible via the open logs WebDAV and DeltaV.

development component

Reusable module of the software component, from a development point of view.

Development Configuration

Collects a set of states of software components required to develop, build, and test a certain software component, providing the information needed to resolve dependencies between development components.

Development object

Component of an ABAP application. The following objects belong to the development objects: programs such as transactions, reports, subroutines and function modules; program components such as event blocks, screens and menus; and objects used by several programs (for example, database fields, field definitions and program messages).

DIAG

Dynamic Information and Action Gateway: Protocol for communication between *SAP GUI* and the application server (also known as the *SAP GUI* protocol).

Domain

An ABAP Dictionary object that describes the technical attributes of a data element, such as data type, length and value range. You can group fields that have similar technical or business purposes under a single domain. All fields based on a domain are updated automatically when you change the domain. This guarantees the consistency of the fields.

enqueue server

Enqueue server is a possible description for the instance of an SAP system on which the enqueue work process manages the lock table in a distributed SAP system.

Enqueue Server

Enqueue server is a possible description for the instance of an SAP system on which the enqueue work process manages the lock table.

Enterprise Service-Oriented Architecture

Enterprise service-oriented architecture: Provision of application functions in the form of enterprise services. These are web services that can be combined and used for business purposes.

F1 Help

You can use the F1 key to display an explanation of fields, menus, functions and system messages. The F1 help also displays technical information on the relevant field.

F4 Help

The F4 help displays input help (possible values) for a field that is ready for input.

favorite

Favorites' lists can be compiled by SAP users and typically contain frequently used transaction codes and web links. A favorite allows you to call up a function directly without additional navigation.

field group

A field group comprises a set of related screen elements, for example, all checkboxes in one selection.

Function Builder

Tool for creating and managing function modules. You can use the Function Builder to create, change, test and document function modules.

gateway

Interface that converts one communication protocol into another communication protocol.

Gateway (Reader)

Interface for communication between SAP systems.

Instance

Administrative unit that combines SAP system components providing one or more services. The offered services of an instances are started, stopped and monitored together. You use a common instance profile to set parameters of all the components of an instance. An instance runs on one physical computer, but there can be multiple instances on one computer. An instance is identified by the system ID (SID) and the instance number.

Internet Communication Manager

The Internet Communication Manager (ICM) is the component of the SAP architecture that enables an SAP system to communicate directly with the Internet.

Java Virtual Machine

Interpreter for the controlled execution of Java bytecode on the respective hardware platform.

JDBC

Java Database Connectivity. Standard Java interface for database integration.

job

Closed chain of steps (such as ABAP programs) that can be controlled successively using certain control commands.

keyword

The first word in an ABAP statement. The keyword determines the meaning of the entire statement.

Lock mode

Status that determines whether a user has exclusive access to a data record, or whether access is shared with other users.

lock object

Object type in the ABAP Dictionary. Activating a lock object generates function modules for setting and releasing locks; you can then use these function modules in ABAP programs.

lock table

The lock table is a table in the main memory of the enqueue server, which contains information about which locks currently exist. For each lock that exists, it shows the owner, the lock mode, the name, and the fields of the locked table.

Message Server

Independent program that maintains a list of all instances in an SAP system. The message server determines which instance a user logs on to and organizes communication between instances.

name server

part of the SAP System Landscape Directory (SLD). The Name Server enables the reservation of globally unique names for Java development objects.

Native SQL (ABAP)

Database language that enables the use of database-specific SQL statements in an ABAP program.

Object Navigator

Navigation tool for managing development objects.

Open SQL (ABAP)

SAP-specific range of SQL statements. ABAP Open SQL allows you to avoid conflicts between database tables and makes ABAP programs independent of the database system used.

Output Request

An output request contains the data from a spool request in a format for a particular printer model.

package

Container for semantically related development objects. A package consists of sub-packages and development objects (programs, tables, screens, function modules, classes, and so on) that are developed and transported together. Packages are characterized by the properties nesting, interfaces, visibility and use accesses. Packages are created and managed with the Package Builder. Packages replace development classes.

Product

is visible to the customer, can be installed and renewed.

Ramp-up

Market introduction process of new products or new releases of SAP software

Release to Customer

Date that marks the initial availability of a new release for customers and the start of the restricted shipment phase.

Repository

Central store for all ABAP Workbench development objects. The development objects stored in the repository of the SAP system include: Program objects, function group objects, Dictionary objects, Business Engineering objects, and other objects.

Restricted Shipment Phase

First delivery phase for a new release. In the phase, the number of customers using the new release is increased in a controlled fashion.

RFC

Remote Function Call (RFC) is an SAP interface protocol based on CPI-C. It simplifies the programming of communication processes between systems.

Role

A role is a collection of activities that a person performs to participate in one or more business scenarios in an organization. You access the transactions, reports, Web-based applications and other objects contained in roles through user menus.

SAP Business Suite

The SAP Business Suite includes a complete package of open enterprise applications that link all parties involved, information and processes, and therefore improve the effectiveness of your business relationships.

SAP Easy Access

SAP Easy Access is the default initial screen in SAP systems. The left side of the screen contains a tree hierarchy of the menus available to you in the SAP system; you can use the right side of the screen to display a graphic, such as your company logo.

SAP GUI

SAP Graphical User Interface; program, which the user can use to exchange information with the ABAP SAP system. You can use the user interface to choose commands, start programs, display files and execute other options by pressing function keys or selecting menu options.

SAP GUI for HTML

SAP GUI that runs in the Web browser and generates HTML pages dynamically on the basis of SAP screens. Requires the Internet Transaction Server as part of AS ABAP.

SAP GUI for Java

Platform-independent *SAP GUI*, requires a Java environment.

SAP GUI for Windows

SAP GUI implementation in a Windows environment

SAP Logon

When you call up the SAP Logon, it displays a list of SAP systems for which you can start the logon process. This list is taken from the information in a file on the front end: *saplogon.ini*. This file is usually centrally preconfigured and provided for end users. During logon, the SAP Logon program also enables logon load balancing using the resources available for the system selected.

SAP NetWeaver

SAP NetWeaver is SAP's integration and application platform. It is the technical basis for almost all SAP applications at the moment.

SAP NetWeaver Business Intelligence

Provides comprehensive analytical functions with optionally installable business content.

SAP NetWeaver Developer Studio

SAP's own development environment for the development of Java-based multi-level business applications. The new development environment is based on the open source product Eclipse.

SAP NetWeaver Development Infrastructure (NWDI)

The SAP NetWeaver Development Infrastructure (NWDI) extends the concept of an integrated development environment (IDE, SAP NetWeaver Developer Studio) with server-side services that provide development teams with a consistent central development environment and support software development during the entire lifecycle of a product.

SAP NetWeaver MDM

SAP Master Data Management provides functionalities for integrated, central master data administration in a company.

SAP NetWeaver Mobile

SAP NetWeaver Mobile serves as the universal platform for company mobility and supports SAP solutions for Mobile Business.

SAP Note

An SAP Note is text information on a specific topic, problem or system message that you may come across when working in the system. All SAP Notes are stored at SAP in an online database, and customers can call them up from there. An example of a possible question could be: "Which database versions of database XY are released in combination with my SAP system release?"

SAP Reference IMG

A complete guideline for introducing SAP systems, which contains all the necessary Customizing activities, sorted according to application components.

SAP system

An SAP system can be, for example, an *SAP ECC 6.0*, an *SAP BW*, or an *SAP CRM* system. SAP systems form the central components of the **SAP Business Suite**.

SAP System Landscape Directory (SLD)

SAP J2EE application, which contains all relevant information about software products and components that can be installed on a system landscape. Furthermore, it contains a description of the current system landscape.

Screen Painter

Tool for creating screens for a dialog transaction. The graphical layout of the screen image and its underlying flow logic are defined in the *Screen Painter*.

Server

Software component that provides a service (software-oriented view) or central computer within a network that provides resources for the individual work-station computers (hardware-oriented view).

shared memory

Main memory area that is accessible for all work processes of an instance.

Shared Memory

Main memory area that is accessible for all work processes of an instance.

Software Component

Quantity of software objects that are grouped in development classes (Java) or packages (ABAP) and are delivered and maintained together.

spool request

A spool request contains information about data to be output, its formatting, and the printer model used.

Startup and Control Framework

The Framework is used to start, stop, and monitor Java instances.

The *Startup and Control Framework* is the infrastructure that SAP provides for starting and stopping Java instances.

Table

Tabular array of data in the *ABAP Dictionary*. A table consists of columns (data values of the same type) and rows (data records). Each record can be identified uniquely by one or more fields.

task handler

The task handler coordinates processes and manages resources within a work process.

TCO

Total cost of ownership

Transaction (AS ABAP)

An ABAP transaction describes a logically complete action in an AS ABAP system. From the user's point of view, a transaction represents a unit (for example, creating a list of a certain type of customer, changing a customer's address, creating a flight reservation for a customer, or executing a program).

transaction code

A transaction code (also known as a TCode) is a sequence of characters that identifies a transaction in the SAP system. A transaction code may contain up to 20 characters and must always begin with a letter. Permitted characters are letters from A to Z, numbers from 0 to 9, and the underscore. To call up a transaction, enter the transaction code into the OK Code field and choose ENTER.

transport

Transfer of SAP system components from one system to another. The components to be transported are specified in the object list of a transport request. Every transport consists of an export and an import process: The export process reads the objects and stores them in the data files at operating system level. The

import process reads objects from the data file and writes them to the database of the target system. The SAP system maintains a transport protocol of all actions during export and import.

Unrestricted Shipment Phase

Second delivery phase for new software releases, in which all customers can request the new software.

User Master Record

The user master record contains the definition of a particular user in the client. Some of the fields are name, first name, initial password and phone number. The user master record is used to create the user context (see: user context) when a user logs on to the system.

VB* tables

Update tables; the data to be changed is stored here until the full set can be collected and written to the final database tables (within a single database transaction).

view

Virtual table that contains no data, but is an application-specific view of one or more tables in the ABAP Dictionary.

WebDAV

DAV = Distributed Authoring and Versioning. WebDAV is an XML-based enhancement of the HTTP protocol for asynchronous document management that is used as a standard for accessing documents using a Web browser.

Index

A

- ABAP, 77, 423
- ABAP Dictionary
 - central administration, 432
- ABAP Dictionary Objects, 433
- ABAP Editor, 427, 429, 482
- ABAP processor, 433
- ABAP Workbench, 423, 426, 428
- Access Method, 114
- ACID, 103
- Application help, 57
- AS ABAP, 38, 77, 87
- AS ABAP + Java System, 87
- AS Java, 78, 87
- Asynchronous Update, 109

B

- Background processing, 114
- Background work process, 84
- Buffer (AS ABAP), 100
- Buffer (AS Java), 145
- Business Server Page
 - applications, 431
- Business Server Pages, 431
- bytecode, 134, 447

C

- central instance, 151, 175
- Central instance, 82
- Central Services, 154, 217
- Central Services (Instance), 82
- Central system, 82
- Change Management Service, 456
- Client, 40

- Client/Server Concept, 78
- Cluster Manager, 145
- Collective run, 111
- command field, 56
- Component Build Service, 454, 456
- Connections Manipulator Manager, 145
- customizing, 58, 414
- Customizing, 415

D

- Database Interface, 99–100
- database transaction, 103
- Database transaction, 105
- default profile, 258
- Defaults and Own Data, 45
- DeltaV, 455
- Design Time Repository, 454, 456
- development component, 458
- Development Configuration, 459
- DIAG, 38
- Dialog work process, 84
- Dispatcher, 100
- Dispatcher (ABAP), 84, 98
- Dispatcher (Java), 86, 144

E

- Eclipse, 455
- EJB, 448
- EJB CMP, 147
- EJB Container Service, 145
- enqueue server, 156
- enqueue server, port, 348

- enqueue service, 156
- Enqueue Service (Java), 86
- Enqueue work process, 84, 105
- Enterprise Java Beans, 137
- Enterprise JavaBeans, 78

F

- F1 Help, 57
- F4 Help, 58
- Find
 - transaction code, 46
- Function Builder, 482

G

- gateway, 85
- Gateway (Reader), 116

H

- HTTP Provider Service, 145

I

- IDE, 455
- Input History, 67
- Instance, 81
- instance profile, 258
- Internet Communication
 - Manager, 85, 117

J

- J2EE, 78
- J2EE Engine Components, 145
- Java applet, 132
- Java application, 132
- Java Beans, 137
- Java cluster, 144, 152
- Java Development Kit, 134
- Java dispatcher, 153
- Java Enterprise Runtime, 145
- Java instance, 151
- Java program, 132
- Java Runtime Environment, 131
- Java Runtime Environment (JRE), 448
- Java Server Pages, 138

- Java Virtual Machine, 132
- Java Web Dynpro, 449
- JCmon, 239
- JDBC, 147, 448
- JDBC Connector Service, 145
- JDK, 448
- JDO (Java Data Objects), 147
- JSP, 145, 448

L

- Layout, Customizing, 67
- license, expiration, 348
- Lock management, 105
- lock table, 105, 110

M

- manager, 160
- Manager (AS Java), 145
- MaxNewSize, 313
- MaxPermSize, 313
- message server, 155, 175, 186
- Message Server, 85
- message server, port, 348
- Message Service, 155
- Message Service (Java), 86
- multiple logons, 41

N

- name server, 456
- Native SQL (ABAP), 100
- NewSize, 313
- NWDI, 454

O

- Object Navigator, 428
- online documentation, 60
- Open SQL, 424
- Open SQL (ABAP), 100
- Open SQL (Java), 147
- operation mode, 278
 - switch, 283
- Output Request, 114

P

package, 414, 429
 Parameter ID, 57, 68
 Performance Assistant, 57
 permanent generation, 312
 PermSize, 313
 Print (AS ABAP), 112
 product, 458
 profile parameter
 rdisp/max_wprun_time, 264
 rdisp/myname, 264
 rdisp/TRACE, 184
 rdisp/wp_no_btc, 264
 rdisp/wp_no_dia, 264
 rdisp/wp_no_enq, 264
 rdisp/wp_no_spo, 264
 rdisp/wp_no_vb, 264
 service/hostname, 177
 service/protectedwebmeth-
 ods, 177
 Profile parameter
 rdisp/max_alt_modes, 41

Q

Quick Info, 67

R

Ramp-up, 17
 Release to Customer, 16
 repository, 414–415
 Request queue, 98
 required entry, 58
 Restricted Shipment Phase, 16
 RMI, 447
 Role-based user menu, 53
 Roll In, 100

S

SAP Business All-in-One, 4
 SAP Business ByDesign, 3
 SAP Business One, 3

SAP Central Process
 Scheduling (SAP CPS),
 115
 SAP Easy Access, 44
 SAP GUI, 37, 39, 98
 SAP GUI for HTML, 38
 SAP GUI for the Java
 environment, 38
 SAP GUI for Windows, 38
 SAP Java Connector (JCo), 87
 SAP Library, 59–60
 SAP Logon, 39
 SAP NetWeaver, 24
 SAP NetWeaver - Development
 Infrastructure, 454–455
 SAP NetWeaver Developer
 Studio, 453, 455
 SAP Note, 40
 SAP Reference IMG, 415
 SAP System Landscape
 Directory, 456
 SAP transaction, 105
 SAP transaction (ABAP), 104
 SAPOSCOL, 175
 Screen elements, 45
 Screen Painter, 427
 Server (processes of AS Java),
 144
 server process, 154
 Server process, 86
 Service (AS Java), 145
 services, 161
 Servlet, 145
 session, 41
 software component, 458
 Software Deployment Manager
 (SDM), 86
 software unit, 458
 Spool Processing, 112
 spool request, 113
 Spool work process, 84
 SQLJ, 147, 449

- start profile, 258
- Startup and Control Framework,
 - 217, 235
- Status bar, 69
- support package, 458
- Support Package, 17
- System Information, 347
- system parameters, 260

T

- table
 - database-independent, 434
 - defining, 434
 - definition, 434
- task handler, 99
- Technical Information, 57
- TemSe, 113
- tenured generation, 312
- three-system landscape,
 - recommended procedure,
 - 417
- time table, 280
- Transaction (AS ABAP), 104
- transaction code
 - ABAPDOCU, 430
 - RSPFPA, 260
 - RZ03, 282
 - RZ04, 280
 - RZ10, 268
 - RZ11, 260
 - SA38, 127–128
 - SE09, 418
 - SE11, 427
 - SE38, 427
 - SE51, 427
 - SE80, 428
 - SM12, 107
 - SM13, 112
 - SM50, 88
 - SM51, 88
 - SM63, 280

- SM66, 120

- SU3, 114

- Transaction code

- SM04, 89

- SM36, 115

- SM37, 115

- SMX, 115

- SP02, 114

U

- Unrestricted shipment phase, 17

- Update, 108

- Update work process, 84

- URL

- global SAP homepage, 13

- SAP Help Portal, 13, 59

- SAP Info, 13

- SAP Insider Online, 13

- SAP Service Marketplace,
 - 13

- usage type, 458

- User Master Record, 41

V

- V1 Updates, 111

- V2 Updates, 111

- VB* tables, 110

- VM settings, 291

W

- Web Application Builder, 431

- Web Container Service, 145

- WebDAV, 431, 455

- Work process, 99, 103

- Work processes, 84

X

- Xms, 313

- Xmx, 313

Y

- young generation, 312

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.