



Overview

Introduction to the SAP Web Application Server

**March, 2005
Version 2.3**

Matt Kangas

Product Manager

SAP Web Application Server



© Copyright 2005 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.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corp. in USA and/or other countries.

ORACLE® is a registered trademark of ORACLE Corporation.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.

LINUX is a registered trademark of Linus Torvalds and others.

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.

J2EE™ is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP, mySAP.com, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. MarketSet and Enterprise Buyer are jointly owned trademarks of SAP Markets and Commerce One. All other product and service names mentioned are the trademarks of their respective owners.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Contents

1 Introduction.....	5
1.1 About This Documentation	5
2 The SAP Web Application Server	6
2.1 Description	6
2.2 Two Personalities in One Server	7
2.3 Architecture.....	7
2.3.1 The Evolution of Web Enabling at SAP.....	7
SAP Basis 4.6 And below.....	8
SAP Web AS 6.10.....	8
SAP Web AS 6.20.....	8
SAP Web AS 6.30 and 6.40.....	9
2.3.2 J2EE Architecture.....	11
2.4 Installation Options for SAP Web AS 6.40	12
2.4.1 ABAP Only	12
2.4.2 ABAP & J2EE	12
2.4.3 J2EE Only	12
2.5 Patches	13
2.5.1 Support Packages	13
ABAP	14
J2EE	14
2.5.2 SAP Kernel.....	14
3 SAP Internet Transaction Server.....	15
3.1 Integrated Internet Transaction Server	15
3.1.1 Technical Structure of the Integrated ITS	15
3.1.2 SAP Note 709038	16
Advantages	16
Limitations	17
3.1.3 Installation	17
3.1.4 Patching.....	17
3.1.5 Platforms and Maintenance	18
3.2 Coexistence of Standalone ITS and SAP Web AS.....	18
3.2.1 Migration.....	18
3.2.2 IAC Development.....	19
4 Summary.....	20
5 Appendix.....	21
5.1 Terms.....	21

Version History

- 1.0 Initial draft version
- 1.1 First revision ready for release
- 2.0 Updated document including ITS information and formatting
- 2.1 Final formatting and style updates
- 2.2 Final branding updates and clarifications
- 2.3 First public release version

1 Introduction

1.1 About this Documentation

The purpose of this document is to introduce the *SAP Web Application Server*. The documentation covers SAP Web Application Server Release 6.40. Target audiences for this document are people with some technical inclination who wish to get a better understanding of what the SAP Web Application Server in action really means, and what it can do for customers.

This document relies on an understanding of SAP Systems based on 4.6C and lower. It is intended to give the reader a basic awareness of the differences between release 6.40 and prior versions, new features introduced with release 6.40, and issues introduced by the technology of the SAP Web Application Server. However, please note the following: (a) some portions of this document will be fairly technical in nature; (b) this document by no means should be considered as a sufficient knowledge base for positioning or implementing SAP NetWeaver, nor can it be considered as an alternative to the RKT or training courses.

2 The SAP Web Application Server

2.1 Description

To understand what the SAP Web Application Server (SAP Web AS) is as compared to previous products, it is useful to actually look at the name backwards. So, from end to front, we start with "server." A server can be defined as:

A computer or device on a network that manages network resources. For example, a *file server* is a computer and storage device dedicated to storing files. Any user on the network can store files on the server. A *print server* is a computer that manages one or more printers, and a *network server* is a computer that manages network traffic. A database *server* is a computer system that processes database queries.

Servers are often dedicated, meaning that they perform no other tasks besides their server tasks. On multiprocessing operating systems, however, a single computer can execute several programs at once. A server in this case could refer to the program that is managing resources rather than the entire computer.¹

So, in this case the resource that the server is managing is the application. We will discuss more what exactly this means later, but the term "application server" has been used at SAP since the beginning of the SAP R/3 days. In fact, the SAP R/3 application and the expansion of client/server technology helped build SAP into the company it is today. From this standpoint the term is exactly the same; that is, the application server today is the same as the application server it was before, with additional features in every new release. The technology that you knew and used prior to SAP Web AS is still there, with Java and ABAP enhancements added.

Continuing to look at the name backwards, we can see that we have the same application server with the term "Web" added. Web in this case means the World Wide Web, or Internet. So the server is now both an application server *and* a web server.

Finally, we have SAP Web Application Server. This means that the whole package is manufactured by SAP. We are putting our name on it and thus are telling customers that we are bringing strengths that we brought to the application world to the Web.

The SAP Web Application Server is the reliable, open standards-based application server from SAP. It supports both J2EE and ABAP, and serves as the underlying infrastructure for all of SAP NetWeaver and all new and upcoming SAP solutions, like SAP R/3 Enterprise, mySAP ERP, SAP Enterprise Portal (SAP EP), SAP Exchange Infrastructure (SAP XI), and all other SAP components. The SAP Web Application Server is *not* a new product; it is the natural

¹ <http://www.pcwebopedia.com/TERM/s/server.html>

evolution of proven SAP application server technology formerly known as SAP Basis. It provides the platform to develop, execute, and operate web applications and web services, as well as traditional SAP GUI applications.

2.2 One Server, Two Personalities

One of the key differentiators of SAP Web AS is that it contains both an ABAP personality and a Java personality. These are often referred to as the ABAP stack and the Java, or J2EE², stack; or the ABAP engine and the J2EE engine. Both the ABAP stack and the J2EE stack exist in a single, unified system

This has many advantages. First, a customer's existing investment in software and skills is preserved. For example, the users, developers and administrators will all be familiar with the ABAP stack. There are certainly new features and enhancements, but this happens with every release. They will not have to be completely retrained to begin using the SAP Web AS (and components based upon it); they will just have to understand the deltas from the prior release.

Now, they will have business-strength access to the J2EE engine. SAP Web AS also embraces native web technologies while providing all the benefits of SAP's knowledge and experience through an evolutionary, standards-based approach. With the SAP Web AS, SAP delivers a homogeneous infrastructure for J2EE-based and ABAP-based applications. All existing business objects and interfaces can be used with both the J2EE environment and the ABAP environment. This approach gives companies a single infrastructure that takes best advantage of both environments. This helps reduce deployment costs since now a separate web server is not necessary to begin the web enabling of business processes. At the same time, the customer is not *forced* to begin using Java until they are ready. This allows them to get training in Java at their own pace—either aggressively or conservatively.

Also, all of the application server benefits that SAP applications are recognized for in the ABAP world are now being brought by SAP into the Java world. That means reliability standards such as high availability, scalability, and security, and benefits that help lower TCO, such as software lifecycle management, simple design and deployment strategies, and ease of administration. Finally, SAP supports openness by offering DB/OS platform independence for Java applications, support of web services and participation in open Java standards.

2.3 Architecture

2.3.1 The Evolution of Web Enabling at SAP

To help understand the current web technology at SAP, it is helpful to look back at the beginning of web enabling at SAP. That is, where did we come from?

² Java 2 Enterprise Edition. Throughout this paper the terms Java and J2EE are used interchangeably.

SAP Basis 4.6 and Lower

Figure 1 shows how SAP applications were web enabled in releases prior to SAP Web AS 6.10. User access was based upon a web browser or certain mobile devices, and access to the system was via the Internet Transaction Server (ITS). To provide this access, the ITS required a connection to a web server. Also, the ITS was not platform-independent, as it had services that were tied to Microsoft Windows technology. The only programming language available was ABAP. User interfaces were programmed with a front-end modeling approach called Dynpro, or dynamic program. Connections to external systems or other third-party and legacy systems were made via custom connectors, gateways, or a Remote Function Call (RFC) to another SAP system. These connectors could be expensive to develop and maintain as they were specific to each system. A change or an upgrade to the connected system could require a complete re-development cycle of the connector.

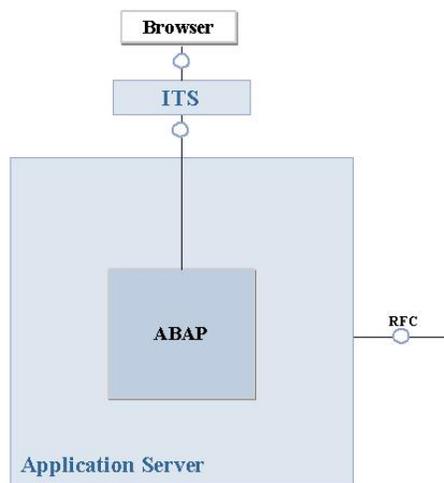


Figure 1

for web connections is realized. Business Server Pages (BSP), a mixture of ABAP and HTML for user interface programming, are also introduced in this release. The SAP Web AS 6.10 did not have a Java engine.

SAP Web AS 6.20

The first version of the SAP J2EE engine was released with SAP Web AS 6.20 as shown in Figure 3. This means that the engine is fully compliant with the 1.2 J2EE standards, making it a J2EE-certified application server. This enables the development and deployment of ABAP-based or J2EE-based applications on the same instance. With Java now available as a programming language, rich and dynamic web

SAP Web AS 6.10

With release 6.10 of SAP Web AS (the successor to SAP Basis 4.6), a major architecture change occurred for web connectivity. The ITS was replaced with the Internet Communication Manager (ICM), as shown in Figure 2. While Internet connections with the ITS require an external component, the ICM consists of new functionality that is part of the core kernel. It manages the communication between browsers (and other external requests) and the SAP Web AS similar to the way that the dispatcher in the system handles requests from the SAPGUI. The ICM also handles connections made over various protocols such as SMTP for mail, HTTP(S), and SOAP/XML. Since the ICM is integrated into the SAP Web AS architecture, platform independence

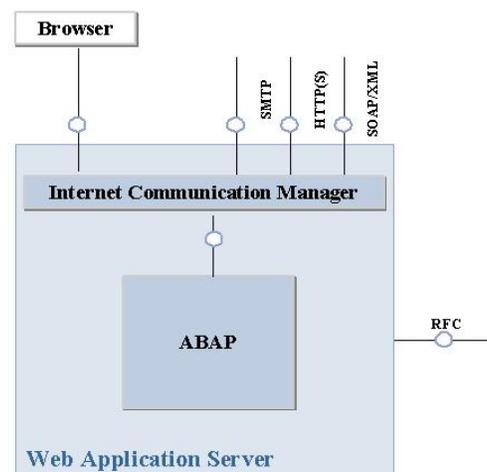


Figure 2

pages can be built with Java Server Pages (JSP). In addition, customers can create their own J2EE web applications to run on SAP Web AS.

SAP Web AS 6.30 and 6.40

With SAP Web AS 6.30, major changes to the architecture of the J2EE engine were introduced (more about J2EE architecture later). The overall architecture of the SAP Web AS is no different than in release 6.20 (Figure 3). The changes were immediately released as version 6.30 so the productivity gains from the new architecture could begin to be realized. So, SAP Web AS 6.30 only contains changes to Java functionality—the ABAP is at the 6.20 level. 6.30 is also a ramp-up only release that was never released to mass shipment.

SAP Web AS 6.40 is the mass ship release and is where the ABAP side of the house catches up with Java development in terms of adding new features. There are no overall architecture changes since SAP Web AS 6.30. The main feature introduced with 6.30/6.40 is Web Dynpro, a runtime and development environment for creating easily adaptable web-based user interfaces. Also included is the SAP NetWeaver Developer Studio, an Integrated Development Environment (IDE) based upon the open source Eclipse framework for J2EE. Java lifecycle management was added as well with the Design Time Repository (DTR), a global System Landscape Directory (SLD), automatic Component Build Service (CBS), and Change Management Services (CMS) similar to the ABAP transport system. This enables team development for Java, just like we have always had for ABAP. Also, the SAP Web AS is now fully J2EE 1.3-compliant and offers limited J2EE 1.4 support (web services and JCA (J2EE Connector Architecture) 1.5). SAP Web AS 6.40 was released for mass shipment as part of SAP NetWeaver '04 on October 1, 2004.

On the long stable ABAP side, the following are the new features:

ABAP Core

- Shared Objects
 - A new kind of ABAP object that allows objects to be kept and managed in shared memory. The results are applications with reduced memory consumption.
- Dynamic Assertions & Breakpoints
 - Both help to augment the supportability of applications by enhancing the coding with dynamic checkpoints that can be left in delivered code and activated later. Breakpoints help give pre-delivered debugging support.

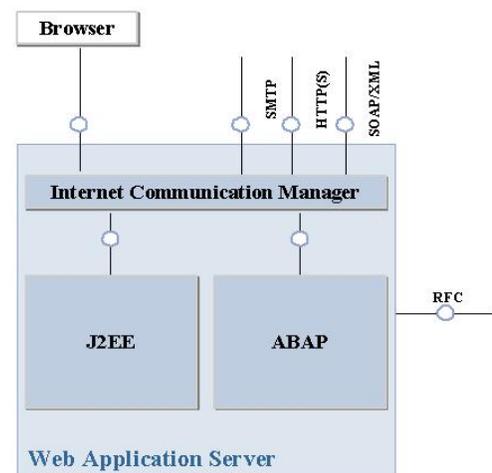


Figure 3

Assertions allow explicit declaration of program correctness to check the state of an application during normal program flow.

- Dynamic Type Creation
 - Permits the creation of type definitions dynamically at runtime. This allows a program to manage and handle data structures that are not known at design time.

ABAP Tools

- Memory Inspector
 - The Memory Inspector is a new tool for analyzing the memory consumption of ABAP programs. The memory use of various objects, from references to other objects up to complex object graphs, may be examined. Comparing memory snapshots allows the identification memory leaks, which might otherwise be hard to find and could cause infinitely growing memory consumption.
- New ABAP Debugger
 - The new ABAP Debugger runs separately from the main application. This means that the limitations of the old debugger no longer apply. Running in a different process, it thus eliminates any side interactions between the debugger and the debugged application, and also enables debugging for all ABAP coding. An improved UI and personalization capabilities, in addition, increases the efficiency of the debugger.
- ABAP Unit
 - ABAP Unit provides a test environment that supports testing of small software units (for example, classes) during the development process according to the Extreme Programming (XP) paradigm. By encouraging the development of tests right from the beginning, it reduces more complex test efforts at later stages.

ABAP XML

- Simple Transformations
 - Mapping between XML and other data structures is gaining more and more importance. As of SAP Web AS 6.20, this can be accomplished for ABAP data structures using the XSL Transformations language (XSLT). However, handling mass data—a typical scenario in most of today's businesses—requires a very high throughput level, not the resource-demanding flexibility of XSLT. Therefore, *Simple Transformations* is introduced with SAP Web AS 6.40, a simpler transformation language that certainly beats XSLT in runtime and memory efficiency.

Integrated ITS

- The ITS is now integrated into the SAP kernel and basis layer, so a separate installation of a stand-alone ITS is not always necessary. See Section 3.

Adobe Forms

- To optimize forms-based business processes, the new release of SAP Web Application Server includes a new solution for interactive forms. This solution

is based on the Portable Data Format (PDF) and software provided by Adobe Inc. Interactive forms can be created that allow a user to fill in a form in PDF format and save the entries inside the form in XML format. The SAP system extracts the data saved in the form for further processing. A form template can be merged with current system data to produce a PDF form for printing or emailing.

2.3.2 J2EE Architecture

As stated above, the architecture of the J2EE stack was significantly changed between releases SAP Web AS 6.20 and 6.30³. The main reasons

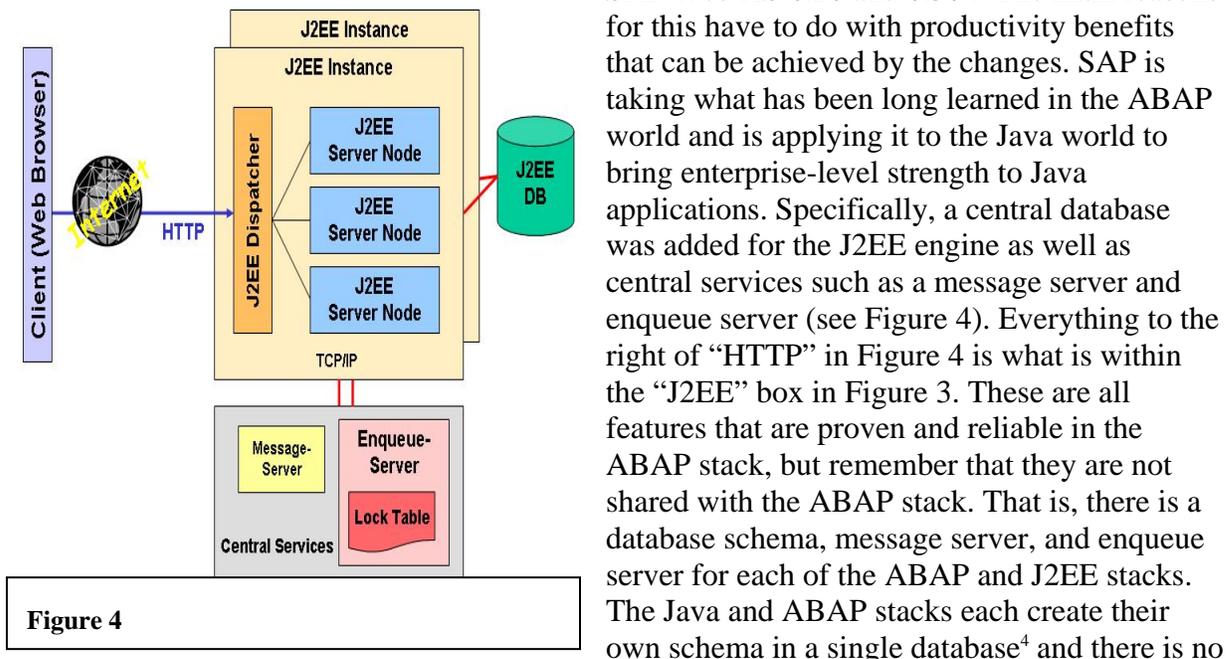


Figure 4

for this have to do with productivity benefits that can be achieved by the changes. SAP is taking what has been long learned in the ABAP world and is applying it to the Java world to bring enterprise-level strength to Java applications. Specifically, a central database was added for the J2EE engine as well as central services such as a message server and enqueue server (see Figure 4). Everything to the right of “HTTP” in Figure 4 is what is within the “J2EE” box in Figure 3. These are all features that are proven and reliable in the ABAP stack, but remember that they are not shared with the ABAP stack. That is, there is a database schema, message server, and enqueue server for each of the ABAP and J2EE stacks. The Java and ABAP stacks each create their own schema in a single database⁴ and there is no direct, cross-stack access of data in the other schema. This is most important in database access. The ABAP engine cannot directly access the J2EE database schema and vice-versa. Communication can occur between the stacks at the application level through interfaces such as the Java Connector (JCo), web services, etc.

These enhancements integrate the well-known and robust SAP Basis features into Java. For example, the central database helps bring stability to the Java world. This happens in two ways. First, the database is a centralized location for configuration information. Second, the database stores application information. Java applications at runtime end up as files at the operating system level. Special backup strategies must be in place to ensure that they can be recovered in case of deletion. However, SAP Web AS stores a copy of these files in the J2EE database as well, so if they are deleted from the file system they are retrieved from the database and recreated. Now a separate file system backup is not necessary, as the database can just be inserted into the customer’s existing database backup strategy.

³ 6.20 J2EE engine architecture is not discussed in this document.

⁴ Although it is technically possible to use separate databases.

The new architecture offers other proven benefits as well. The enqueue server is used as a reliable lock handler, offering stability. The message server acts as both a message hub and monitor. This offers scalability and high availability since the message server can recognize when an application server is down (planned or unplanned) and can then re-direct users to other application servers. Additional application servers can also easily be added. Plus, SAP supports platform independence by offering the J2EE engine and database on multiple operating systems, database vendors and web browsers. See the [Product Availability Matrix](#) on the SAP Service Marketplace for supported platform combinations.

2.4 Installation Options for SAP Web AS

6.40

With both an ABAP stack and a Java stack, SAP Web AS now has many different installation options depending on the used components and customer scenarios. Although SAP Web AS is part of every technical SAP NetWeaver scenario, not all of these options may be available for every SAP component—the individual component usage of Java and resulting installation requirements is outside the scope of this document.

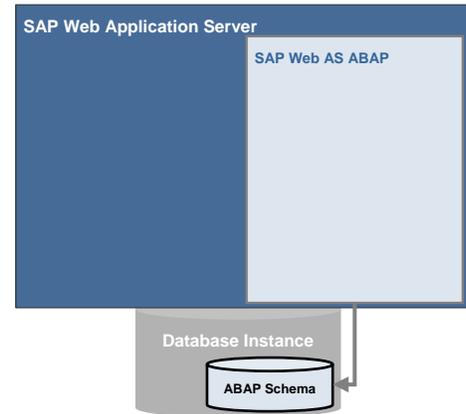


Figure 5

2.4.1 ABAP Only

This is the classic installation where only the ABAP stack and services are installed (see Figure 5). No Java services would be available and web enabling would require the addition of the Java stack or an ITS. BSP is possible as well as all ABAP enhancements

2.4.2 ABAP & J2EE

In this installation option both the ABAP and J2EE stacks are installed (see Figure 6). The J2EE engine is installed as a second schema in the existing ABAP database. The J2EE portion of the installation is known as a J2EE add-in installation. A J2EE add-in installation can be performed on any SAP Web AS system 6.20 and higher.

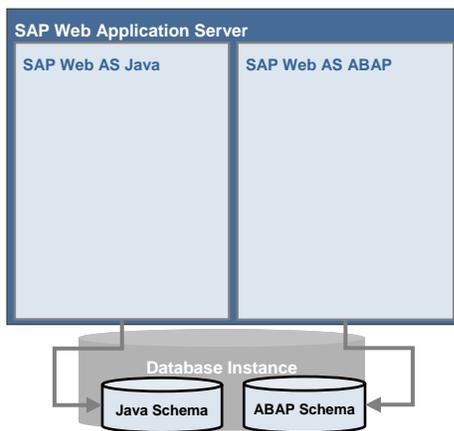


Figure 6

2.4.3 J2EE Only

Only the J2EE stack is installed (see Figure 7). This would be equivalent to other web server installations (such as WebSphere from IBM or WebLogic from BEA) for Java projects. An ABAP schema cannot be added into an existing J2EE database, so a J2EE-only installation cannot be converted to an ABAP and J2EE installation.

2.5 Patches

With two different stacks, ABAP and J2EE, there are many different patch types that must be kept clear. As will be seen below, the application of patches can have varying implications for the release level of the server(s).

This document does not address patch types at the application component level.

2.5.1 Support Packages

The application of support packages to each stack has very significant implications to the release level and also to the applications based upon them.

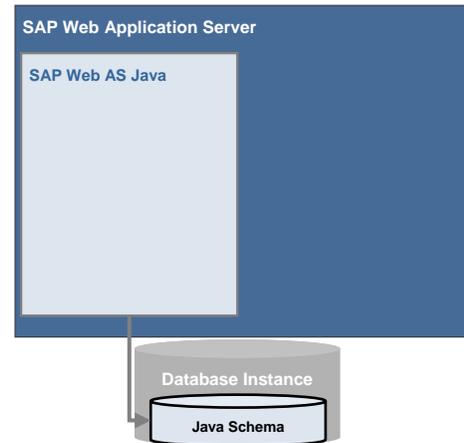


Figure 7

- There are ABAP support packages for both the 6.20 release and the 6.40 release. An ABAP system with SP25 or greater is considered 6.30. However, no amount of support packages applied to an ABAP 6.20 or 6.30 system will ever make it an ABAP 6.40 system; only an upgrade can do this.
- Because of the differences in the architecture between J2EE 6.20 and 6.30/6.40, it is not possible to upgrade the J2EE Engine 6.20 to 6.30/6.40. A new J2EE Engine 6.40 must be installed and all applications that were running on J2EE Engine 6.20 must be redeployed. See the *6.30/6.40 Migration Manual* [Ext].
- Each and every Java installation of SAP Web AS 6.30 automatically becomes a Java installation of SAP Web AS 6.40 system by applying Java SP 4. This is true for the standalone case as well as the add-in case. SAP Web AS 6.30 Java Ramp-up customers have committed to apply the support packages in their acceptance letter. So over time, there will be no Java 6.30 systems left. SAP Web AS 6.30 was never be released to mass shipment.
- SAP has committed to the SAP R/3 Enterprise customers that they can get Java on SAP Web AS 6.30/6.40 as an add-on as soon as it becomes available on their platform combination. SAP R/3 Enterprise customers can not do an upgrade to ABAP on SAP Web AS 6.40 without upgrading the application as well (for example, to mySAP ERP).
- We do not force the upgrade to ABAP on SAP Web AS 6.40 from our customers that have ABAP on SAP Web AS 6.20 + Java on SAP Web AS 6.40 today. Upgrading to SAP Web AS 6.40 releases of ABAP + Java is a future OPTION for them.

This means that J2EE 6.30/6.40 with ABAP 6.20 is a fully supported combination (and may be quite common).

ABAP

All releases of SAP Web AS still have the traditional ABA and BASIS Support Packs and they are still applied in the same way (transaction SPAM). They can be downloaded at SAP Service Marketplace alias /patches → SAP WEB AS → SAP WEB AS <Release Level> → Support Packages → <SP Type>.

J2EE

The Java stack has two types of Support Packs: those for the J2EE Engine and those for J2EE applications.

J2EE Engine support packages modify the actual engine executables and are similar to a kernel patch. SAP Note 657763 describes how to apply the patches and they can be downloaded at SAP Service Marketplace alias /patches → SAP WEB AS → SAP WEB AS <Release Level> → Binary Patches → SAP J2EE Engine 6.XX. The operating system-dependent files can be found in the folder named after the operating system and the operating system-independent file in folder "OS independent."

For information on how to deploy J2EE application support packages, see SAP Note 656711.

2.5.2 SAP Kernel

The SAP kernel (for the ABAP stack) still has its own patches as well. And, just as before, the SAP Web AS 6.40 kernel is downward compatible to 6.20 and will replace SAP Web AS 6.20 kernel at a later point in time (see SAP Note 662523). The downward compatible kernel is SAP's way to deliver kernel patches to its customers with different releases of the component SAP_BASIS, without forcing them to upgrade the rest of SAP_BASIS (ABAP programs etc).

This reduces SAP's development costs by helping SAP reduce the number of correction tracks and reduce the TCO for the customer by providing enhanced kernel features without a complete release upgrade. Note that applying a higher level kernel patch does not change the system to a higher release level. Release levels can only be changed through an upgrade⁵. SAP kernel patches can be downloaded at SAP Service Marketplace alias /patches → SAP WEB AS <Release Level> → Binary Patches → <kernel type> → <operating system>. The operating system-dependent files can be found in the folder named after the operating system and the operating system-independent file in folder "OS independent."

⁵ Except for J2EE SP4 that upgrades the J2EE engine from 6.30 to 6.40.

3 SAP Internet Transaction Server

3.1 Integrated Internet Transaction Server

With both the Internet Transaction Server and SAP Web AS, SAP offers differing Internet-enabling functionality, customers (who are always interested in lowering TCO) asked for the two products to be integrated. ITS development worked long and hard to integrate the ITS into the SAP Web Application Server and the vision finally became reality with release 6.40 of SAP Web Application Server. The new ITS version is officially called "Integrated ITS."

The integrated ITS is the successor of the standalone ITS releases up to and including ITS 6.20. With the integrated ITS it is no longer required to install the ITS on a dedicated server or use a separate web server in order to access the SAP system via SAP GUI for HTML or IACs⁶. The integrated ITS runs "out-of-the-box" if you have installed an application server based on SAP Web AS 6.40 or higher and is fully integrated into the infrastructure of the SAP System (this especially means that the communication from the SAP system to the browser is not managed by the integrated ITS, but by the Internet Communication Manager "ICMan").

3.1.1 Technical Structure of the Integrated ITS

Similar to the standalone ITS, the integrated ITS consists of two components:

- A single file called `dw_gui` that is part of the SAP Web AS kernel.
- A set of HTML templates, JavaScript files, and MIME objects that are part of the database of the SAP System.

Figure 8 shows how the integrated ITS is embedded into the SAP Web Application Server:

⁶ In some circumstances customers may want or need to install stand-alone ITS servers.

SICF (configuration of the ITS services), ST11 (traces) and "SE38 -> SITSPMON" (status of the integrated ITS).

- Publication of Internet Application Components (IACs) to external servers is no longer necessary. After a "pseudo publishing operationl," the integrated ITS loads the templates, MIMES, and other elements of the IAC directly from the database.

Limitations

Limitations in comparison with the "standalone" versions of the ITS:

- ITS Flow Logic, WebRFC, and Web Reporting are not supported. Due to this restriction, customers will have to continue to execute applications that are using these programming models in SAP ITS 6.20.
- The integrated ITS can only connect to a system in which it is contained. In particular, this means that you have to continue to use a "standalone" ITS for all systems that are based on SAP Basis or SAP Web Application Server 6.30 or earlier.
- It is not sufficient to upgrade the SAP Web Application Server kernel to Release 6.40 to be able to use the "Integrated ITS." Although the Integrated ITS is a subcomponent of the SAP Web Application Server kernel, it also requires database contents that are only delivered with release 6.40 of the SAP Web Application Server.

3.1.3 Installation

The integrated ITS does not need to be installed separately. It is automatically installed with the installation of the SAP Web Application Server 6.40, and only needs to be activated as an ICF service. Please refer to the online documentation for information on how to configure the integrated ITS.

3.1.4 Patching

Since it is now part of the SAP Web Application Server kernel, separate patches are no longer delivered for the Integrated ITS. Corrections are delivered as part of the relevant SAP Web Application Server kernel patch (Disp+Work package). Refer to SAP notes 19466 and 138704 for more information on kernel patches. Due to the fact that the Integrated ITS is not only part of the SAP Web Application Server kernel but also requires components in the database, all corrections cannot be delivered via a kernel patch (services, templates and MIMES previously delivered in SAR archives), therefore Integrated ITS corrections may also be delivered as corrections inside of SAP Basis Support Packages. Customers may, therefore, have to install a kernel patch and a Support Package to correct certain problems.

The patches can be found on the SAP Service Marketplace → SAP Web AS → SAP Web AS 6.40 → Binary Patches → SAP Kernel <version> → <Operating System> → Database independent for ITS Kernel patches and SAP Web AS → SAP Web AS 6.40 → Support Packages → SAP Basis 6.40 for corrections to ITS templates.

Note: Currently it is not possible to apply corrections to ITS templates using the Note Assistant (SNOTE) - therefore customers have to apply Support Packages in order to get corrections for ITS templates (a better integration into the Correction Workbench is currently being implemented). All corrections for ITS in SAP Web AS release 6.40 are maintained in SAP Note 721993.

3.1.5 Platforms and Maintenance

The Integrated ITS is part of the SAP Web Application Server kernel. Thus it is supported on all platforms that the kernel of the SAP Web Application Server is supported on. Information on these platforms can be found on the SAP Service Marketplace under the "platforms" or "pam" aliases. The maintenance deadline for SAP Kernel 6.40 is currently defined to be 31 March 2009 - the end of maintenance for the integrated ITS will occur at the same time.

3.2 Coexistence of Standalone ITS and SAP Web AS

SAP ITS 6.10 reached its end-of-life on 31 December 2004. SAP ITS 6.20 will be supported long-term (to be more precise: until the last application that might require the external ITS is no longer supported - most likely this means that this will be the day SAP R/3 Enterprise is no longer supported). There will be no additional releases of the standalone ITS and most enhancements to ITS will only be implemented in the SAP Web AS delivered Integrated ITS.

This does not mean that ITS 6.20 is no longer recommended to be used and it should be made clear to customers that ITS 6.20 is currently our primary ITS release.

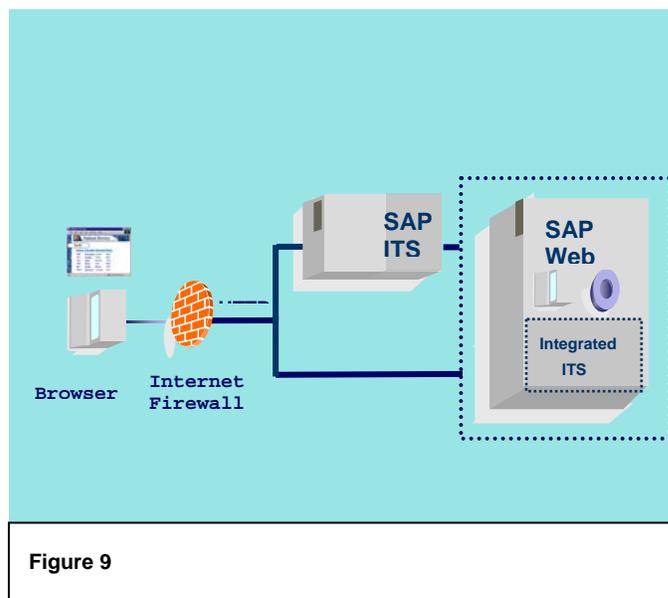


Figure 9

3.2.1 Migration

For migration of existing ITS processes that are not supported on the Integrated ITS, one or more standalone SAP ITS implementations can be used in conjunction with SAP Web AS 6.40 or above (see Figure 9). However, this may cause problems if functionality that runs through the ITS 6.20 should also attempt to use functionality that is only available

in higher SAP Web AS releases (6.40 or greater)—it may or may not function correctly and it is not supported by SAP. This should not be an issue for SAP delivered functionality, but customers should be aware and take care with their own developments using ITS functionality.

3.2.2 IAC Development

The release of the Integrated ITS resulted in several important changes for development of Internet Application Components (IAC). These changes are described in the guide *Migration of Existing ITS Services*. This guide is available at the SAP Service Marketplace alias /sap-its → Media Library → Literature → Developing IACs with the SAP Integrated ITS.

4 Summary

The SAP Web Application Server is the technical platform for SAP NetWeaver, providing the complete infrastructure to develop, deploy, and run all SAP NetWeaver components, the mySAP Business Suite, customer-developed applications, and third-party J2EE 1.3-compliant applications. The SAP Web AS brings together the benefits of a proven, scalable, and reliable infrastructure with the interoperability and flexibility of web service infrastructure.

SAP Web AS fully supports both the proven ABAP technology and the open source Internet-driven technologies—Java and Java 2 Enterprise Edition, also called J2EE—offering best-in-class, comprehensive security solutions that protect data and ensure the confidentiality of business transactions.

5 Appendix

5.1 Terms

We use the following terms to refer to components of the SAP Web AS system configuration:

Term	Explanation
SAP Web AS	The whole system consisting of one or more SAP Web AS instances and the underlying database management system (DBMS) seen as a single entity. Often <i>SAP system</i> is used as a synonym.
Database system (DB, DBMS)	Database software or database management system that supports the SAP Web AS
Central instance (CI)	SAP Web AS instance (that is, the underlying database management system is connected, but on other hardware), but including a dispatcher process with at least the two system-critical components enqueue and message service . In addition to this minimal configuration, the CI normally contains additional work processes (dialog, update, batch, spool) and services such as gateway or sapcomm. A CI is (except for the database) a standalone SAP Web AS unit. When using the ABAP+J2EE installation of SAP Web AS , the CI contains also the J2EE CI.
J2EE Central Instance (J2EE CI)	SAP Web AS instance without a DBMS (that is, DBMS is on other hardware), but including a J2EE dispatcher, J2EE servers and the Software Delivery Manager SDM.
SAP Central Services (SCS)	SAP Web AS instance, containing the enqueue and message service for the J2EE Engine.
Central system	Full SAP Web AS instance including DBMS installed on the same hardware. A central system is a standalone SAP Web AS unit.
Application server (AS)	SAP Web AS instance, without a DBMS, consisting of a dispatcher process and several kinds of work processes (dialog, update, batch, spool) and additional services such as gateway or sapcomm. An AS does not contain the central, system-critical enqueue and message service components.
SAP Web AS instance	General expression for a single SAP Web AS client-server unit that runs on a specified host machine. An instance can be a central system or an application server. It is started as a unit that is configured in a common instance profile.

The definition of a central instance (CI) used in this documentation is restricted and might differ from that used in other SAP documentation. We use it to mean that only the enqueue and message service have to run on the CI instance. Therefore, an AS instance becomes a CI simply by the addition of the enqueue and message services.

Please note that besides the CI there might be the J2EE CI which is a J2EE instance with SDM.