

# JBoss Cache on SAP NetWeaver



**SAP Platform Ecosystem**



## Copyright

© 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, Outlook, and PowerPoint 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, and Informix are trademarks or registered trademarks of IBM Corporation in the United States 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.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C 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.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

## Icons in Body Text

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

## Typographic Conventions

Type Style	Description
<i>Example text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.  Cross-references to other documentation.
<b>Example text</b>	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
<b>Example text</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

## Table of Contents

<b>Introduction .....</b>	<b>5</b>
Prerequisites.....	5
<b>Procedure .....</b>	<b>6</b>
Importing the initial projects into the Developer Studio.....	6
Providing the Transaction Manager.....	6
Configuration of JBossCache .....	7
Deploying JBossCache on SAP Web AS .....	8
Log4j configuration for JBossCache .....	9
Using the Cache.....	9
Adding a new local J2EE Server Instance .....	9
<b>The demo project .....</b>	<b>11</b>
Running the sample .....	11
Prerequisites .....	11
Procedure .....	11
Result.....	12
Further Information .....	13
<b>About the author .....</b>	<b>13</b>

## Introduction

Software caches are a common instrument to increase performance and scalability of an application. They can be found in every layer of multi-tiered Client/Server architectures. May it be to minimize the communication to a database server, to connect the server on the client side etc. Therefore caches have a wide acceptance on J2EE applications. The JBoss Application Server comes with its cache simply called JBossCache. Since the JBossCache project is independent to the JBoss server, it can be used standalone and on different application servers.

JBossCache is a replicated cache. Which means that the cache's content can be replicated in a distributed environment. You may run the cache in different JVMs on different host and synchronously the caches if needed. For J2EE applications running in a clustered environment this means, that JBossCache is able to replicate the cache on every server node in a cluster. Naturally the replications take some time and may produce heavy network traffic, which shouldn't be forgotten when planning to optimize performance of applications by caching.

JBossCache supports asynchronous and synchronous replication. Where synchronous replication will block other cache consumers until a replication is done and asynchronous won't. Therefore asynchronous replication is faster, but on the other side synchronous replication guarantees all consumers at every time receive the same content. The cache is transactional. You can make your cache operations transactional by configure the cache to use a JTA Transaction Manager.

The cache consists of two components: the TreeCache and the TreeCacheAop (This document focuses on the TreeCache component). As the name implies the cache is tree-structured. The cached data is stored in hash maps in the nodes of the tree. The API for the tree operations is quite simply, as you will see later on.

This tutorial will show you what it takes to use JBossCache on the SAP NetWeaver Web Application Server. After the reading you'll be enabled to use JBossCache on the SAP WAS – no matter if you want to develop a new application with JBossCache or you want to port an existing application using JBossCache to the SAP WAS.

## Prerequisites

### Systems, installed applications, and authorizations

- SAP WebAS – Java 6.40

## Procedure

### Importing the initial projects into the Developer Studio

1. Call the SAP NetWeaver Developer Network using the URL <http://sdn.sap.com> and log on with your user ID and the corresponding password. If you do not have a user ID, you must register before you can log on.
2. Navigate to *Web Application Server* area and then to the *Samples and Tutorials* section.
3. Download the ZIP file *TutJBossCache.zip*
4. Unzip the contents of the ZIP file into the work area of the SAP NetWeaver Developer Studio or in local directory.
5. Call the SAP NetWeaver Developer Studio.
  - a. To do this, choose *File* → *Import* in new menu.
  - b. In the next window choose *Multiple Existing Projects into Workspace* and choose *Next* to confirm.
  - c. Choose *Browse*, open the folder in which you unzipped the projects *TutJBossCache.zip*
  - d. Select the all projects of the ZIP-File, check the the *open* checkbox and choose *Finish* to confirm.

### Providing the Transaction Manager

As mentioned in the introduction JBossCache is transactional, therefore you have to provide a JTA Transaction Manager. Providing the transaction manager is done by setting a fully qualified name to class that provides a Transaction Manager in the configuration file of the cache (see [Configuration of JBossCache](#)). This class must implement the interface *org.jboss.cache.TransactionManagerLookup*. Since JBossCache doesn't have this implementation for SAP Web AS. You have to implement the interface yourself, which is pretty easy as you can see in the source code.

#### com.example.SapWasTransactionManagerLookup

```
package com.example;

import javax.naming.InitialContext;
import javax.transaction.TransactionManager;
import org.jboss.cache.TransactionManagerLookup;

/**
 * Implementation of JBossCache's TransactionManagerLookup-Interface.
 * The class provides the SAP Web AS JTA TransactionManager by a JNDI
lookup.
 */
public class SapWasTransactionManagerLookup
    implements TransactionManagerLookup {

    /**
     * Returns the JTA Transaction Manager of the SAP Web AS.
     */
    public TransactionManager getTransactionManager() throws Exception {
        return (TransactionManager) new InitialContext().lookup("ts");
    }
}
```

## Configuration of JBossCache

JBossCache can be configured by the API or – which is recommended - by loading a XML configuration file. The configuration for the example looks as follows. (The bold line shows the reference to our SapWasTransactionManagerLookup class, which naturally has to be in the classpath.)

```

cache-conf.xml
<?xml version="1.0" encoding="UTF-8"?>
<server>
  <classpath codebase="./lib" archives="jboss-cache.jar, jgroups.jar"/>
  <mbean code="org.jboss.cache.TreeCache"
name="jboss.cache:service=TreeCache">
  <depends>jboss:service=Naming</depends>
  <depends>jboss:service=TransactionManager</depends>
  <attribute name="TransactionManagerLookupClass">
    com.example.SapWasTransactionManagerLookup
  </attribute>
  <attribute name="IsolationLevel">REPEATABLE_READ</attribute>
  <attribute name="CacheMode">REPL_ASYNC</attribute>
  <attribute name="UseReplQueue">>false</attribute>
  <attribute name="ReplQueueInterval">20000</attribute>
  <attribute name="ReplQueueMaxElements">5</attribute>
  <attribute name="ClusterName">SAP Web AS - Cluster</attribute>
  <attribute name="ClusterConfig">
    <config>
      <UDP mcast_addr="230.8.8.8" mcast_port="56677" ip_ttl="32"
ip_mcast="true" mcast_send_buf_size="80000" mcast_rcv_buf_size="150000"
ucast_send_buf_size="80000" ucast_rcv_buf_size="150000" loopback="true"/>
      <PING timeout="2000" num_initial_members="3" up_thread="false"
down_thread="false"/>
      <MERGE2 min_interval="10000" max_interval="20000"/>
      <FD shun="true" up_thread="true" down_thread="true"/>
      <VERIFY_SUSPECT timeout="1500" up_thread="false"
down_thread="false"/>
      <pbcast.NAKACK gc_lag="50"
retransmit_timeout="600,1200,2400,4800" max_xmit_size="8192"
up_thread="false" down_thread="false"/>
      <UNICAST timeout="600,1200,2400" window_size="100"
min_threshold="10" down_thread="false"/>
      <pbcast.STABLE desired_avg_gossip="20000" up_thread="false"
down_thread="false"/>
      <FRAG frag_size="8192" down_thread="false" up_thread="false"/>
      <pbcast.GMS join_timeout="5000" join_retry_timeout="2000"
shun="true" print_local_addr="true"/>
      <pbcast.STATE_TRANSFER up_thread="false" down_thread="false"/>
    </config>
  </attribute>
  <attribute name="InitialStateRetrievalTimeout">10000</attribute>
  <attribute name="SyncReplTimeout">10000</attribute>
  <attribute name="LockAcquisitionTimeout">15000</attribute>
  <attribute name="EvictionPolicyClass"/>
  <attribute name="FetchStateOnStartup">>true</attribute>
</mbean>
</server>

```

Explaining every node is beyond the scope of this document. A detailed description can be found in the [Admin Guide of JBoss](#). Nevertheless it's worth mentioning that everything under the *config*-node will be relevant for the configuration of [JGroups](#) which is used by JBossCache to realize the needed peer-to-peer communication between nodes in a cluster. This example uses UDP as transport protocol. You may use TCP or TUNNEL for communication. Once again you will find detailed information about the possibilities and configurations that's needed in the [Admin Guide of JBoss](#).

## Deploying JBossCache on SAP Web AS

In the environment of JBoss the cache can be easily deployed as a service by defining d deploying a Mbean as service. Since on the [SAP Web AS custom services](#) are not supported, you're in the need to look for alternatives. The SAP Web AS allows to deploy [J2EE Libraries](#) that are basically a collection of a Jar archive, that can be used by different applications. The library project will package the JBossCache Jars archives and a archive, that is contains the Classes we provide for the usage of the cache, which is the [SapWasTransactionManagerLookup](#) and the JBossCache class that takes care of he lifecycle of the cache and makes sure only one instance of the TreeCache is used by using the Singleton Pattern.

```
com.example.JBossCache
package com.example;

import org.jboss.cache.CacheException;
import org.jboss.cache.PropertyConfigurator;
import org.jboss.cache.TreeCache;

public class JBossCache {
    /** TreeCache Object */
    private static TreeCache tree;

    /** Private construtor to prevent instating the Class */
    private JBossCache() {

    }
    /**
     * Singelton Pattern for Tree.
     */
    public static TreeCache getInstance() {
        if (tree == null)
            init();
        return tree;
    }
    /**
     * Initliazation of the Cache.
     */
    public static void init() {
        try {

            tree = new TreeCache();
            PropertyConfigurator configurator = new PropertyConfigurator();
            configurator.configure(tree, "cache-conf.xml");
            tree.createService();
            tree.startService();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * Destroying the cache.
     */
    public static void destroy() {
        tree.stopService();
        tree.destroyService();
    }

    ...
}
```

}

## Log4j configuration for JBossCache

Apache Log4j is used by JBossCache to log everything happening concerning the cache. The information provided might be helpful for you, when problem occurs or just to make sure everything works fine. Log4j looks for a configuration file in the classpath to initialize itself. Therefore you have to provide a configuration file called log4j.xml. In the *CacheUser1\_web*-project you will find an example of for this file in the *WEB-INF* folder. For detailed information on c Log4j visit the [Log4j site](#).

## Using the Cache

The following example shows the usage of the cache which is pretty much self-explaining. In short terms the example does the following.

1. Initialize the Cache
2. Put a key/value pair on the Cache under /node1/node2/node3 in the tree
3. Get the just added entry and print it to the standard out.
4. Shutdown the Cache

### Using the TreeCache

```
TreeCache tree = new TreeCache();
tree.setClusterName("MyCluster");
tree.setClusterProperties("myConfig.xml");
tree.setCacheMode(TreeCache.REPL_SYNC);
tree.createService();
tree.startService();
tree.put("/node1/node2/node3", "key", "value");
System.out.println(tree.get("/node1/node2/node3", "key"));
tree.remove("/node1/node2/node3");
tree.stopService();
tree.destroyService();
```

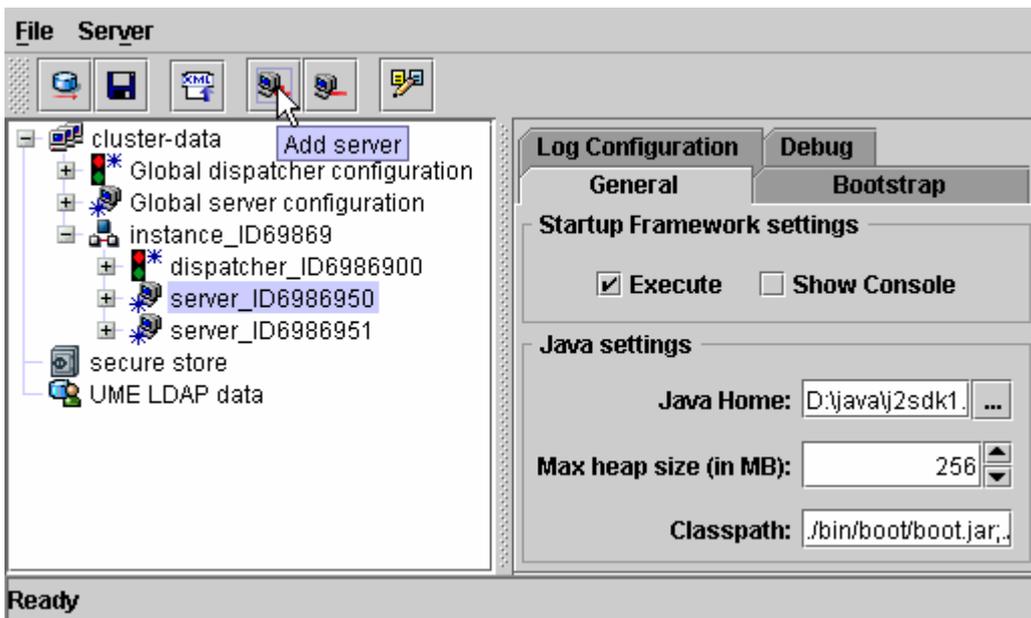


*Note that for using the TreeCache with replication the keys and values have to implement the Serializable interface. If this doesn't conform your requirements take a look at [TreeCacheAOP](#).*

## Adding a new local J2EE Server Instance

To test the replication of JBossCache on the SAP Web AS, you need to have at least two J2EE Server nodes. Adding a node can be done with the Config Tool, as shown below.

1. Start Config Tool
2. Mark a server
3. Click the plus button



Picture 1 Adding a new local server

The new server will be created during the next startup of your SAP Web AS.



*In case you want to use only one server instance after the tutorial, just hit the minus button while the additional node is marked.*

## The demo project

To demonstrate the usage of JBossCache we need an application that uses the cache and naturally the cache itself must be available on the server. The cache is packaged as J2EE library. The library consists of the Jar archives needed by JBossCache and a Jar containing the classes that are needed to run the JBossCache on the SAP Web AS (see [Deploying JBossCache on SAP Web AS](#)). To initialize and shutdown the cache, there is a web application that has one servlet, that is loaded on startup. In the init-method and the destroy method of that servlet the according method on the provided JBossCacheManager in the Library are called. Last but not least we have two applications that use the cache in simply way for demo purposes.



*It is not guaranteed that the destroy method of a servlet is only called when the container shuts down. It may be called for instance to conserve memory.*

In a short overview there are the following projects in the listed categories.

- The JBoosCache server J2EE Library that packages the JBossCache
  - *JBossCache* project contains JBossCache Jars and classes for running the cache on the SAP Web AS
  - *JBossCacheLib* project, that makes the *JBossCache* project a deployable unit, since pure Java Project can't be deployed
- A Web application that only serves for initialization and shutdown of the cache by using the init and destroy method of a servlet, that is loaded on startup
  - *JBossCacheManager\_ear* and *JBossCacheManager\_web*
- 2 Web applications that demonstrate the usage of JBossCache. There are two projects, just to show a communication between different applications using the cache.
  - *CacheUser1\_ear* and *CacheUser1\_web*
  - *CacheUser2\_ear* and *CacheUser2\_web*

## Running the sample

### Prerequisites

- A Running SAP Web AS – Java 6.40 with at least two nodes (see [Adding a new local ...](#) )

### Procedure

1. Deploy the JBossCacheLib's SDA
  - a. Start the SAP NetWeaver Application Server
  - b. Start the SAP NetWeaver Developer Studio
  - c. Open the *J2EE Development Perspective* and chose the *J2EE Explorer View*
  - d. Chose in the context menu of the JBossCacheLib project Build Application Archive
  - e. Chose in the context menu of the JBossCacheLib.sda Deploy to J2EE Engine
2. Deploy the JBossCacheManager\_ear EAR
  - a. Start the SAP NetWeaver Application Server
  - b. Start the SAP NetWeaver Developer Studio
  - c. Open the *J2EE Development Perspective* and chose the *J2EE Explorer View*
  - d. Chose in the context menu of the JBossCacheManager\_ear project Build Application Archive
  - e. Chose in the context menu of the JBossCacheManager\_ear.ear Deploy to J2EE Engine

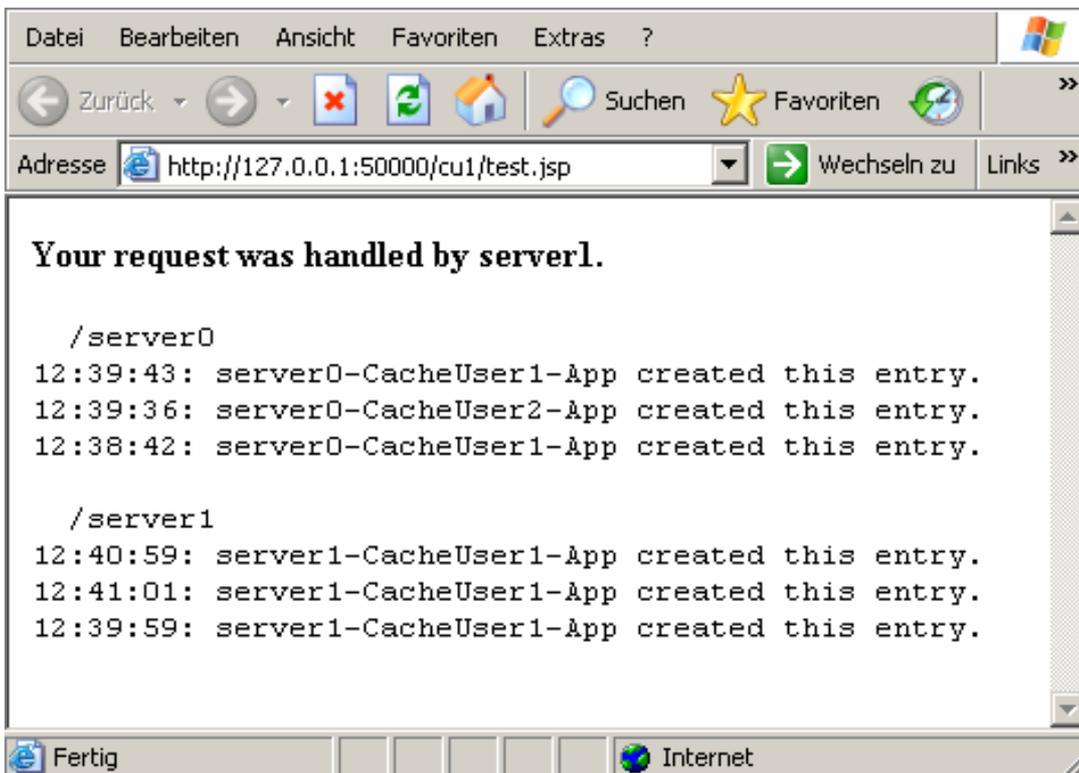
3. Deploy the CacheUser1\_ear EAR
  - a. Start the SAP NetWeaver Application Server
  - b. Start the SAP NetWeaver Developer Studio
  - c. Open the *J2EE Development Perspective* and chose the *J2EE Explorer View*
  - d. Chose in the context menu of the CacheUser1\_ear project Build Application Archive
  - e. Chose in the context menu of the CacheUser1\_ear.ear Deploy to J2EE Engine
4. Deploy the CacheUser2\_ear EAR
  - a. Start the SAP NetWeaver Application Server
  - b. Start the SAP NetWeaver Developer Studio
  - c. Open the *J2EE Development Perspective* and chose the *J2EE Explorer View*
  - d. Chose in the context menu of the CacheUser2\_ear project Build Application Archive
  - e. Chose in the context menu of the CacheUser2\_ear.ear Deploy to J2EE Engine
5. Go to <http://<host>:<port>/cu1/test.jsp> in your browser
6. Go to <http://<host>:<port>/cu2/test.jsp> in your browser
7. Go to <http://<host>:<port>/cu1/test.jsp> in your browser
8. Repeat step 7 in a **new browser** (a new user session is required!) until the server, that handles the requests changes

## Result

The applications demonstrate two scenarios.

1. Communication between different applications
2. Replication on different server nodes

The web pages show which server handles the request and prints the content of the cache on the screen. Implicitly every loading of a side will put a new entry on the cache. Regardless of the server which handles the request and regardless of the application, that handles the request you will always see the same content in the browser. This proves that the communication between applications and the replication between server nodes works successfully.



Picture 2 Sample of demo application output



*Note that the name of the server in the web pages is parsed by the folder in which the application runs on the server, which only works in scenarios, that use different server nodes on one machine. In an environment, where the nodes are located on different machines the information may be misleading.*

## Further Information

- [JBossCache](#)
- [Admin Guide of JBoss](#)
- [JBossCache FAQ](#)
- [TreeCache](#)
- [JGroups](#)

## About the author

Johannes Hamel is a technology consultant with main interest on J2EE. He is experienced on different J2EE platforms and different Open Source products. Currently he is focused on the SAP NetWeaver platform as a team member of the Platform Ecosystem – Market Development Engineering for SAP in Walldorf.