Monitoring SAP Web Application Server

eG Innovations Product Documentation

www.eginnovations.com

eG
Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

'The SAP Web Application Server (Web AS) is the web services infrastructure for all current versions of SAP ABAP Enterprise, and SAP xApps, mySAP solutions and any SAP J2EE-based application. It is also the underlying technology for SAP Enterprise Portal, SAP Business Information Warehouse, and SAP Exchange Infrastructure, and a key component in the SAP Enterprise Services Architecture and SAP NetWeaver. Therefore, managing the SAP Web AS is crucial for ensuring business continuity in the SAP environment. This can be easily achieved using eG Enterprise.

The graphic below shows the components of the Web Application Server (see Figure 1.1).

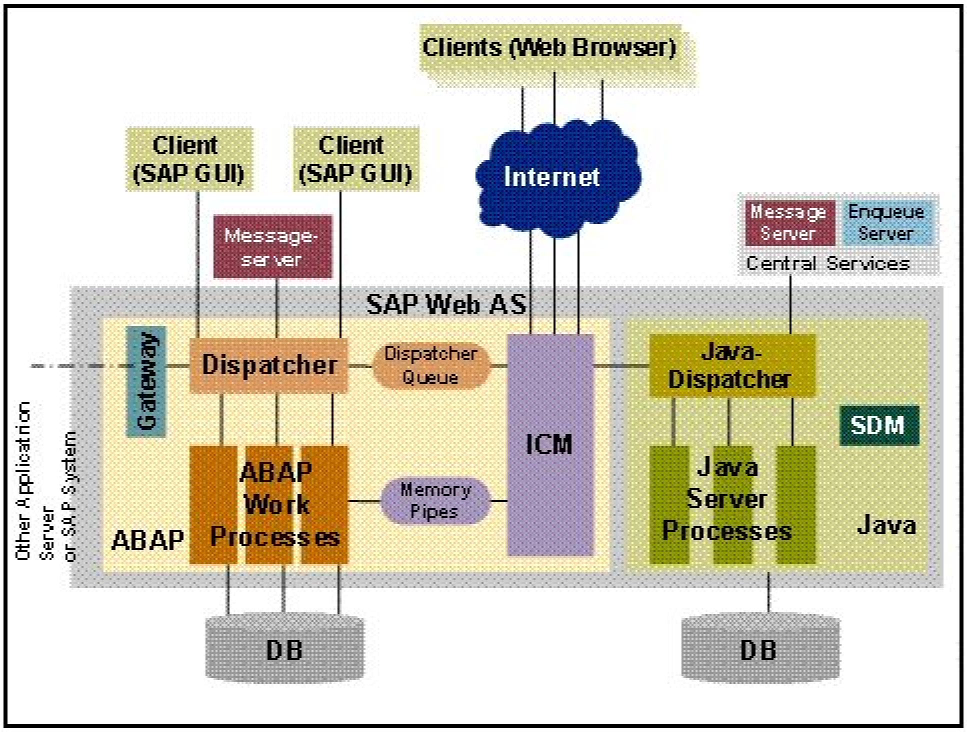


Figure 1.1: The SAP Web AS Architecture

The components and their tasks are described below:

- The Internet Communication Manager (ICM) sets up the connection to the Internet. It can process both server and client Web requests. It supports the protocols HTTP, HTTPS, and SMTP. The SAP Web AS can behave as a Web server or as a Web client .

- The dispatcher distributes the requests to the work processes. If all the processes are occupied the requests are stored in the dispatcher queue.
- The ABAP work process executes the ABAP code.
- The SAP Gateway makes the RFC interface between the SAP instances available (within an SAP System and beyond system boundaries).
- The message server exchanges messages and balances the load in the SAP System.
- In the Java component of the SAP Web AS there are the components Java Dispatcher, Server Process, and Software Deployment Manager. The Java dispatcher receives the client request and forwards it to the server process with the lowest capacity usage. If there is already a connection to the client, the request goes to the server process that processes this client. The server processes actually execute the J2EE application. The Software Deployment Manager (SDM) is a tool with which you can manage and deploy software packages (Software Deployment Archives (SDAs) and Software Component Archives (SCAs)) that you receive from SAP.

This document discusses the monitoring model that eG Enterprise prescribes for the SAP Web Application server in a typical SAP infrastructure.

Chapter 2: How to Monitor SAP Web Application Server Using eG Enterprise?

eG Enterprise is capable of monitoring the SAP Web Application server in both agent-based and agentless manners. To make the eG agent to communicate with the SAP WAS, a set of pre-requisites should be fulfilled before starting monitoring the server. These requirements are given below.

2.1 Pre-requisites for Monitoring a SAP Web AS

Before attempting to monitor a SAP Web AS, make sure that the following requirements are in place:

- Make sure that four specific '.jar' files are available in the install folder for SAP WAS. To know which are these jar files and how to make them available in the install folder, refer to Section **2.2**.
- Starting from SAP WAS v7.5, the four '.jar' files used for monitoring are compatible with Java 1.8 only. The eG agent on the other hand is bundled with JRE 1.7. Because of this JRE mismatch, the eG agent may not be able to monitor SAP WAS v7.5 (and above). To avoid this, make sure that you replace the eG agent's JRE folder with the corresponding folder from a JRE 1.8 installation.
- Make sure you know the name of the SAP Web AS instance that needs to be monitored. To know how to figure out the instance name, refer to Section **2.3.2.1**.
- The eG agent connects to the SAP Web AS instance using the P4 protocol. Find the port at which the P4 protocol listens, so that the eG agent can be configured with the same. To know where to look for the P4 protocol port, see Section **2.3.2.2**.
- Make sure you have the valid credentials of a user who has the right to log on to the SAP Web AS instance to be monitored. The eG agent has to be configured with the credentials of this user, so that it is able to connect to the SAP Web AS instance.
- To perform eG JVM monitoring of the SAP Web Application Server and collect additional metrics, the eG agent uses SAP JMX. To know how to enable SAP JMX and collect the required metrics, refer to the Section **2.3** topic. If the SAP WAS instance to be monitored is in a 'multi-server' environment, then follow the steps detailed in Section **2.3.1** , to know how to enable JVM monitoring for that instance.

2.2 Configuring SAP Web Application Server

The following sections provide you the details about the files to be installed and how to make them available to the eG agent based on the versions of the SAP WAS.

2.2.1 Configuring SAP Web AS Versions Prior to v7.3

Before attempting to monitor a SAP Web AS that is of a version prior to v7.3, ensure that the following files are copied to the /opt/egurkha/lib directory (in Unix; on Windows, this will be the <EG_ INSTALL_DIR>\lib directory):

- com_sap_pj_jmx.jar
- exception.jar
- logging.jar
- sapj2eeclient.jar

While the **sapj2eeclient.jar** is available in the {SAP_ WAS_ HOME_ DIR}\usr\sap\TPP\DVEBMGS00\j2ee\j2eeclient directory, the other 3 jar files reside in the {SAP_ WAS_HOME_DIR}\usr\sap\TPP\DVEBMGS00\j2ee\admin\lib directory.

Once the files are copied, restart the eG agent on the SAP Web AS.

2.2.2 Configuring SAP Web AS v7.3 (and above)

To monitor a SAP Web AS v7.3 (or above), do the following:

1. Copy the **sap.com~tc~bl~pj_ jmx~Impl.jar** file from the <SAP_ WAS_ HOME_ DIR>\<SID>\<instance name>\j2ee\cluster\bin\ext\tc~jmx\lib\privatedirectory to the /opt/egurkha/lib directory (in Unix; on Windows, this will be the <EG_INSTALL_DIR>\lib directory) on the eG agent host.
2. Likewise, copy the following files from the <SAP_WAS_HOME_DIR>\<SID>\<instance ID>\j2ee\j2eeclient\directory to the opt/egurkha/lib directory (in Unix; on Windows, this will be the <EG_INSTALL_DIR>\lib directory) on the eG agent host:

    - sap.com~tc~je~clientlib~impl.jar
    - sap.com~tc~exception~impl.jar
    - sap.com~tc~logging~java~impl.jar

3. Then, rename all the four files so copied as indicated below:

File Name	Rename as
sap.com~tc~bl~pj_jmx~Impl.jar	com_sap_pj_jmx.jar
sap.com~tc~je~clientlib~impl.jar	sapj2eeclient.jar
sap.com~tc~exception~impl.jar	exception.jar
sap.com~tc~logging~java~impl.jar	logging.jar

4. Finally , restart the eG agent on the SAP Web AS.

2.3 Enabling SAP JMX Support

To enable the eG agent to connect to the SAP WAS server and pull metrics, you need to enable SAP JMX support.

Note:

Ensure that the JDK version of the J2EE Engine VM is 1.5 or above.

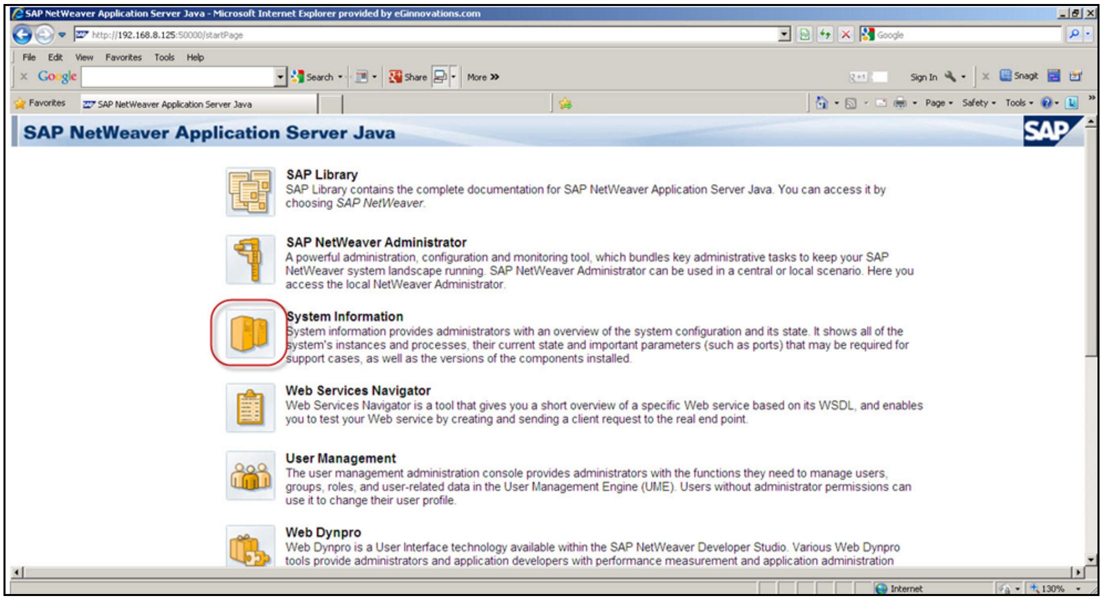
For enabling SAP JMX support, you can use either of the approaches listed below:

- Through the System Information web application using a browser
- Using a config tool

2.3.0.1 Enabling JMX Support Through the System Information Web Application

This approach involves the following steps:

1. Open the default Netweaver AS JAVA landing page (see Figure 2.1). Please click on the System Information icon in Figure 2.1, as indicated.


SAP NetWeaver Application Server Java - Microsoft Internet Explorer provided by eGinnovations.com
http://192.168.8.125:50000/startPage
File Edit View Favorites Tools Help
Google
Search
Share
More
Sign In
Snagit
Favorites
SAP NetWeaver Application Server Java
Page
Safety
Tools
Internet
130%

SAP NetWeaver Application Server Java

SAP Library
SAP Library contains the complete documentation for SAP NetWeaver Application Server Java. You can access it by choosing *SAP NetWeaver*.

SAP NetWeaver Administrator
A powerful administration, configuration and monitoring tool, which bundles key administrative tasks to keep your SAP NetWeaver system landscape running. SAP NetWeaver Administrator can be used in a central or local scenario. Here you access the local NetWeaver Administrator.

System Information
System information provides administrators with an overview of the system configuration and its state. It shows all of the system's instances and processes, their current state and important parameters (such as ports) that may be required for support cases, as well as the versions of the components installed.

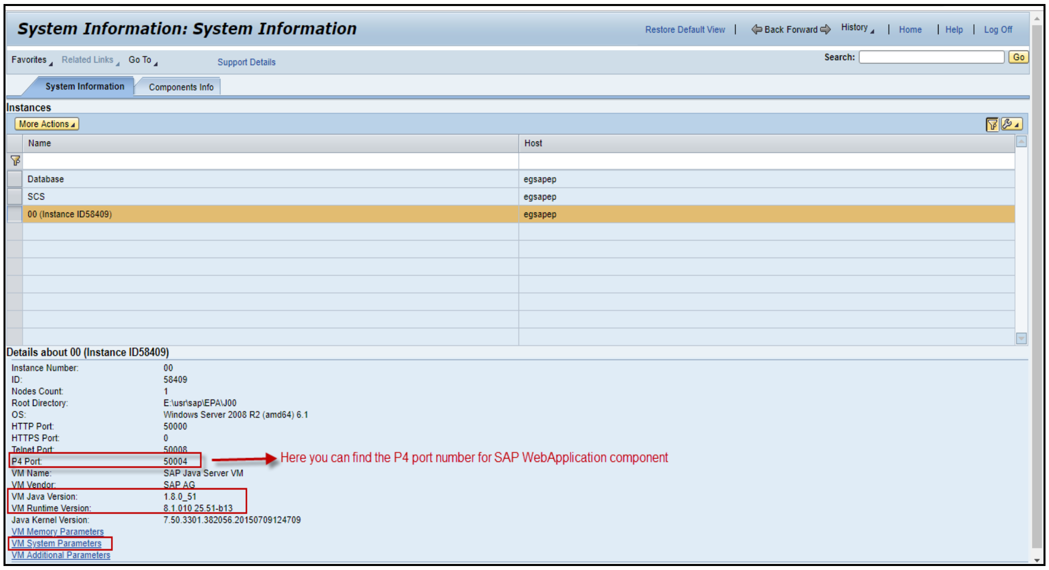
Web Services Navigator
Web Services Navigator is a tool that gives you a short overview of a specific Web service based on its WSDL, and enables you to test your Web service by creating and sending a client request to the real end point.

User Management
The user management administration console provides administrators with the functions they need to manage users, groups, roles, and user-related data in the User Management Engine (UME). Users without administrator permissions can use it to change their user profile.

Web Dynpro
Web Dynpro is a User Interface technology available within the SAP NetWeaver Developer Studio. Various Web Dynpro tools provide administrators and application developers with performance measurement and application administration

Figure 2.1: The SAP NetWeaver Application Server Java landing page

2. A login screen then appears. Login to the system information application using administrator or j2ee_admin credentials. Upon successful login, Figure 2.2 will appear.



Instances

Name	Host
Database	egsapep
SCS	egsapep
00 (Instance ID58409)	egsapep

Details about 00 (Instance ID58409)

Instance Number:	00
ID:	58409
Nodes Count:	1
Root Directory:	E:\usr\sap\EPA\J00
OS:	Windows Server 2008 R2 (amd64) 6.1
HTTP Port:	50000
HTTPS Port:	0
Telnet Port:	50008
P4 Port:	50004
VM Name:	SAP Java Server VM
VM Vendor:	SAP AG
VM Java Version:	1.8.0_51
VM Runtime Version:	8.1.010 25.51-b13
Java Kernel Version:	7.50.3301.382056.20150709124709

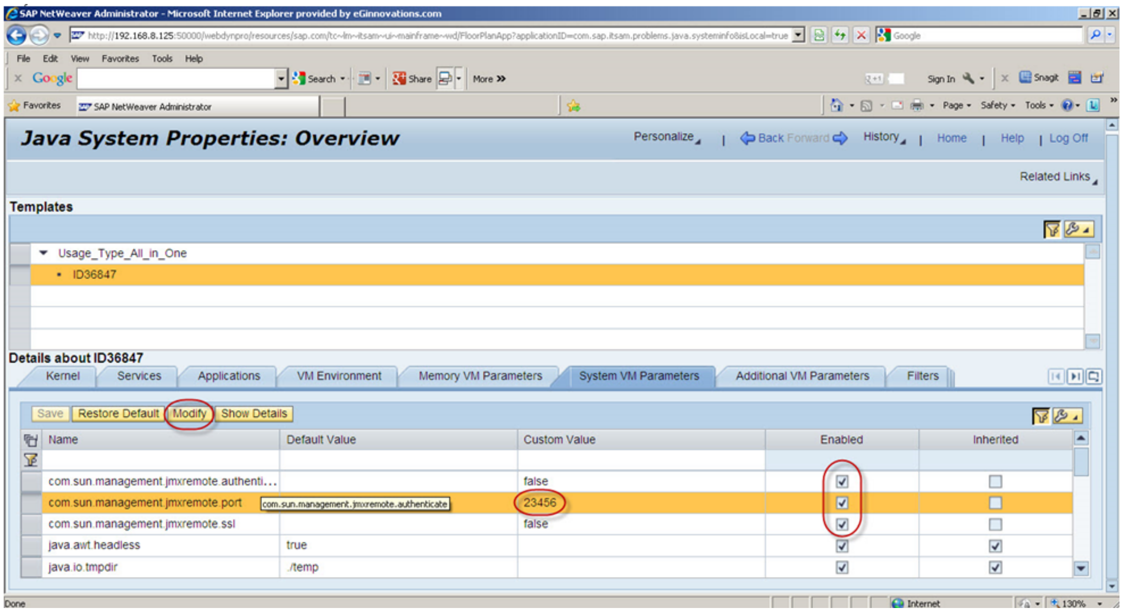
VM Memory Parameters
VM System Parameters
VM Additional Parameters

Here you can find the P4 port number for SAP WebApplication component

Figure 2.2: Selecting instance and checking Java version

3. From Figure 2.2, select the Netweaver AS JAVA instance - i.e., the J2EE Engine Instance - to be monitored (as indicated). Please verify that the version is at least 1.5 and above by checking the value of the **VM Version** parameter in the *Details about <instance>* section. Note down the P4 protocol port from Figure 2.2. Click on **VM System Parameters** link in the same section to

display/modify VM system parameters for this instance. This will invoke Figure 2.3.


SAP NetWeaver Administrator - Microsoft Internet Explorer provided by eGinnovations.com
http://192.168.8.125:50000/webdynpro/resources/sap.com/tc~lm~itsam~ui~mainframe~wd/FloorPlanApp?applicationID=com.sap.itsam.problems.java.systeminfo&isLocal=true
Java System Properties: Overview
Personalize
Back Forward
History
Home
Help
Log Off
Related Links
Templates
Usage_Type_All_in_One
ID36847
Details about ID36847
Kernel
Services
Applications
VM Environment
Memory VM Parameters
System VM Parameters
Additional VM Parameters
Filters
Save
Restore Default
Modify
Show Details
com.sun.management.jmxremote.authenticate
Done
Internet
130%

Name	Default Value	Custom Value	Enabled	Inherited
com.sun.management.jmxremote.authenti...		false	☑	□
com.sun.management.jmxremote.port		23456	☑	□
com.sun.management.jmxremote.ssl		false	☑	□
java.awt.headless	true		☑	☑
java.io.tmpdir	./temp		☑	☑

Figure 2.3: Details of the chosen instance displayed

4. Verify whether the following VM parameters in Figure 2.3 are **Enabled**:

```
com.sun.management.jmxremote.port
```

```
com.sun.management.jmxremote.ssl com.sun.management.jmxremote.authenticate
```

5. If not, enable them by selecting the check box in the **Enabled** column corresponding to each parameter (see Figure 2.3).

6. You may also want to change the **Default Value** of the com.sun.management.jmxremote.port parameter to an appropriate and available port number. To make this change, select the row containing the com.sun.management.jmxremote.port parameter and click the **Modify** button indicated by Figure 2.3.

7. After making the port change, scroll down the list of VM parameters in Figure 2.3 and look for the javax.management.builder.initial parameter. If you find it, then check whether that parameter is **Enabled**. If it is enabled, then ensure that you disable the parameter by unchecking the **Enabled** check box corresponding to it.

8. Then, click the **Save** button in Figure 2.3 to register the changes.

9. Finally, restart the J2EE engine.

2.3.0.2 Enabling JMX Support Through Config Tool

The Config Tool provides offline configuration of the AS Java. It allows you to modify the properties of all services, managers, and applications on a template and an instance level. It also enables you to manage log configuration, add filters, and edit the JVM parameters.

To enable JMX using the Config tool, do the following:

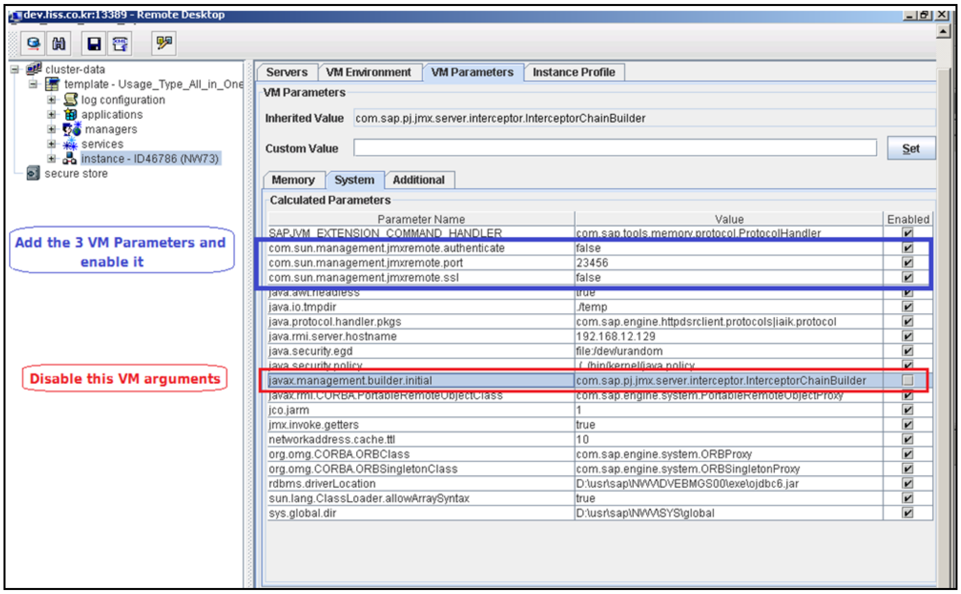
1. Login to the SAP WAS server and open its command prompt.
2. To start the config tool on a Windows installation of the SAP WAS server, follow the steps below:
   - Switch to the <SAP_ WAS_ INSTALL_ DIR>\<SYSTEM_ NAME>\<INSTANCE_ NAME>\j2ee\configtool directory.
   - Run the following command:

     configtool.bat

   To start the config tool on a Unix/Linux installation of the SAP WAS server, follow the steps below:

   - Switch to the <SAP_ WAS_ INSTALL_ DIR>/<SYSTEM_ NAME>/<INSTANCE_ NAME>\j2ee\configtool directory.
   - Run the following command:

     ./configtool.sh
3. Figure 2.4 will then appear.



Inherited Value: com.sap.pj.jmx.server.interceptor.InterceptorChainBuilder

Custom Value:

Calculated Parameters

Parameter Name	Value	Enabled
SAPJVM_EXTENSION_COMMAND_HANDLER	com.sap.tools.memory.protocol.ProtocolHandler	☑
com.sun.management.jmxremote.authenticate	false	☑
com.sun.management.jmxremote.port	23456	☑
com.sun.management.jmxremote.ssl	false	☑
java.awt.headless	true	☑
java.io.tmpdir	./temp	☑
java.protocol.handler.pkgs	com.sap.engine.httpdsrclient.protocols|iaik.protocol	☑
java.rmi.server.hostname	192.168.12.129	☑
java.security.egd	file:/dev/urandom	☑
java.security.policy	./../bin/kernel/java.policy	☑
javax.management.builder.initial	com.sap.pj.jmx.server.interceptor.InterceptorChainBuilder	□
javax.rmi.CORBA.PortableRemoteObjectClass	com.sap.engine.system.PortableRemoteObjectProxy	☑
jco.jarm	1	☑
jmx.invoke.getters	true	☑
networkaddress.cache.ttl	10	☑
org.omg.CORBA.ORBClass	com.sap.engine.system.ORBProxy	☑
org.omg.CORBA.ORBSingletonClass	com.sap.engine.system.ORBSingletonProxy	☑
rdbms.driverLocation	D:\usr\sap\NW\DVEBMGS00\exe\ojdbc6.jar	☑
sun.lang.ClassLoader.allowArraySyntax	true	☑
sys.global.dir	D:\usr\sap\NW\SYS\global	☑

Add the 3 VM Parameters and enable it

Disable this VM arguments

Figure 2.4: Disabling VM arguments in the Config Tool

4. Add the following VM options in the corresponding instance, as highlighted by Figure 2.4.

```
com.sun.management.jmxremote.port=<<JMX_PORT>>
```

```
com.sun.management.jmxremote.ssl=false
```

```
com.sun.management.jmxremote.authenticate=false
```

5. Disable the following VM option, if it exists:

```
Javax.management.builder.initial=com.sap.pj.jmx.server.interceptor.InterceptorChainB
uilder
```

6. Save the changes and close the Config tool.
7. Start the SAP WAS server.

2.3.1 Enabling JVM Monitoring for a Multi-server SAP Web Application Server Instance

In a multi-server environment, two/more Java server processes run within a single SAP WAS instance. For the eG agent to perform JVM monitoring of the SAP WAS multi-server instance, you need to enable JMX support for each Java server process in the instance, separately.

Note:

Ensure that the JDK version of the J2EE Engine VM is version 1.5 and above.

The steps for enabling JMX support for a multi-server SAP WAS instance are as follows:

1. Connect to the SAP Netweaver administrator tool by providing the URL: http://<IP_address_of_tool>:<Port_number_of_tool>/nwa/sysinfo

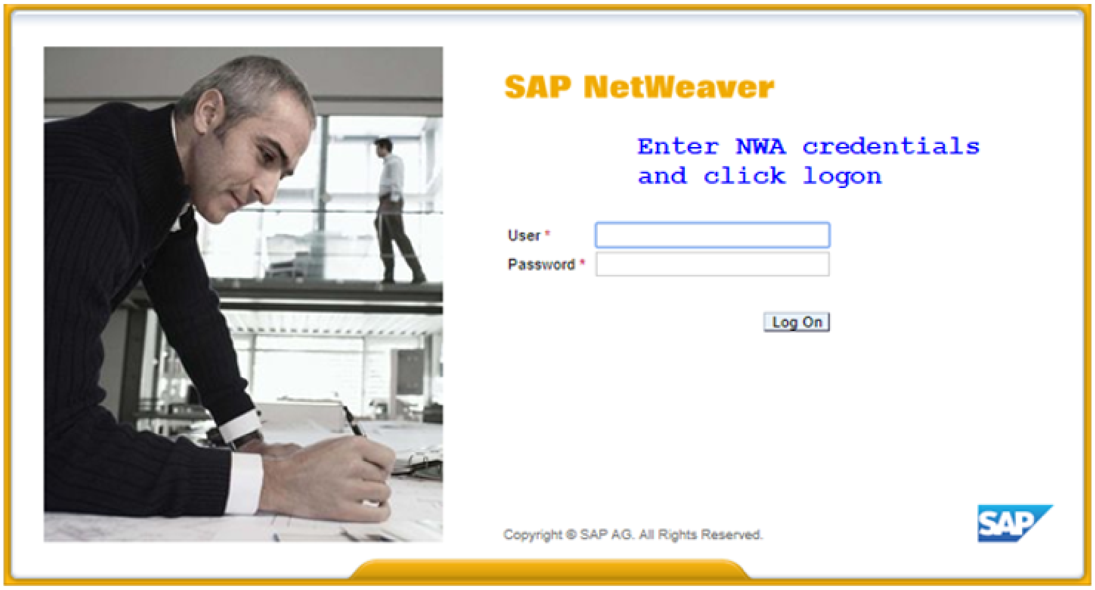
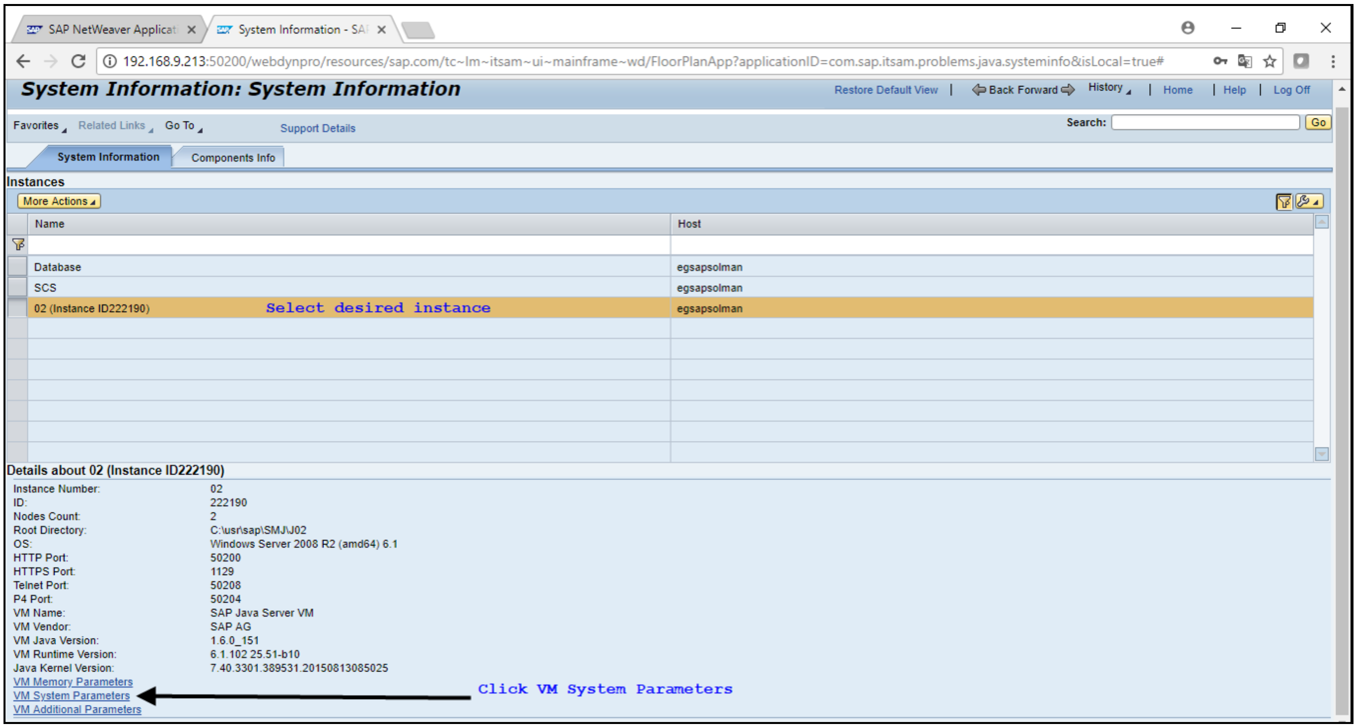

SAP NetWeaver
Enter NWA credentials and click logon
User *
Password *
Log On
Copyright © SAP AG. All Rights Reserved.
SAP

Figure 2.5: The login page of the SAP Netweaver administrator tool

2. A login screen then appears (see Figure 2.5). Login to the system information application using administrator or j2ee_admin credentials. Upon successful login, Figure 2.6 will appear.



System Information: System Information

Instances

Name	Host
Database	egsapsolman
SCS	egsapsolman
02 (Instance ID222190)	egsapsolman

Details about 02 (Instance ID222190)

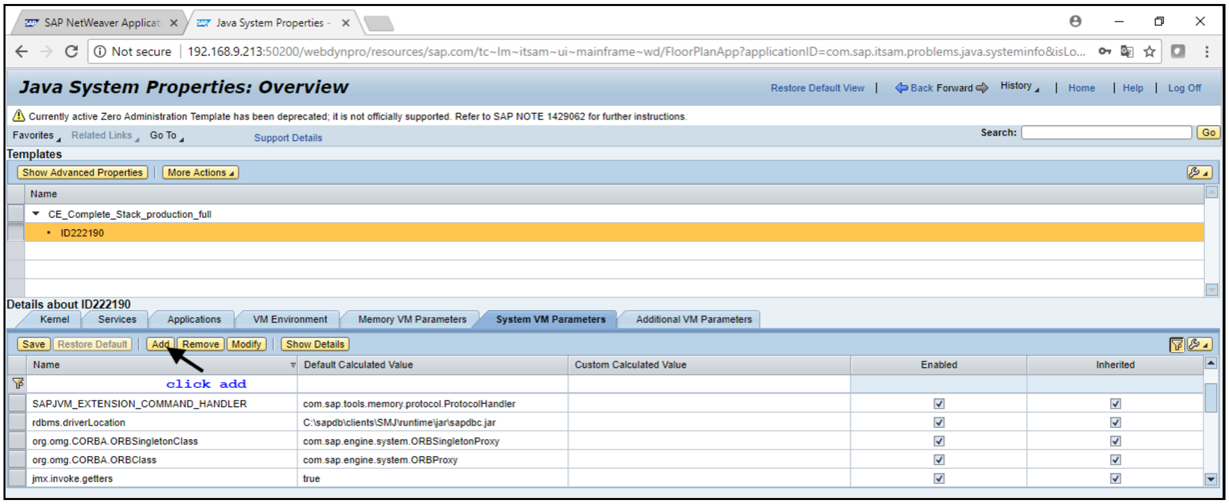
Instance Number:	02
ID:	222190
Nodes Count:	2
Root Directory:	C:\usr\sap\SMJ\J02
OS:	Windows Server 2008 R2 (amd64) 6.1
HTTP Port:	50200
HTTPS Port:	1129
Telnet Port:	50208
P4 Port:	50204
VM Name:	SAP Java Server VM
VM Vendor:	SAP AG
VM Java Version:	1.6.0_151
VM Runtime Version:	6.1.102 25.51-b10
Java Kernel Version:	7.40.3301.389531.20150813085025

VM Memory Parameters
VM System Parameters
VM Additional Parameters

Figure 2.6: Selecting the SA WAS instance in the cluster for which JMX support needs to be enabled

3. From the list of Instances in the **System Information** tab page (of Figure 2.6), select the instance for which JMX support is to be enabled.

4. Then, click the **VM System Parameters** link at the bottom of the **System Information** tab page. Figure 2.7 will then appear.



Java System Properties: Overview

⚠ Currently active Zero Administration Template has been deprecated; it is not officially supported. Refer to SAP NOTE 1429062 for further instructions.

Templates

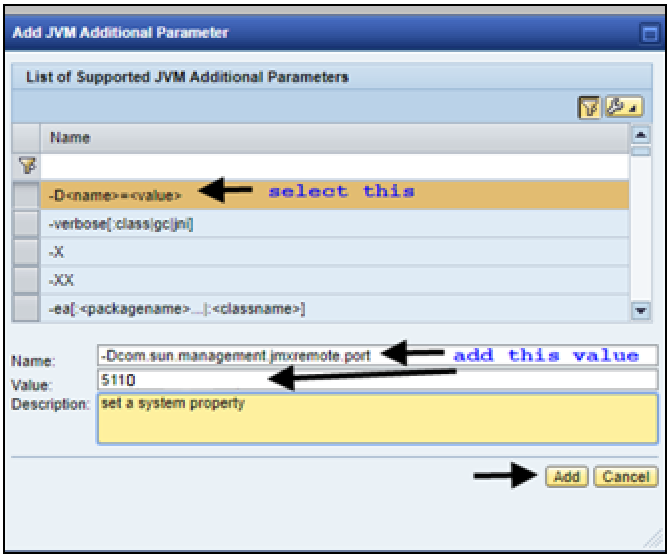
Name
CE_Complete_Stack_production_full
ID222190

Details about ID222190

Name	Default Calculated Value	Custom Calculated Value	Enabled	Inherited
SAPJVM_EXTENSION_COMMAND_HANDLER	com.sap.tools.memory.protocol.ProtocolHandler		☑	☑
rdbms.driverLocation	C:\sapdb\clients\SMJ\runtime\jar\sapdbc.jar		☑	☑
org.omg.CORBA.ORBSingletonClass	com.sap.engine.system.ORBSingletonProxy		☑	☑
org.omg.CORBA.ORBClass	com.sap.engine.system.ORBProxy		☑	☑
jmx.invoke.getters	true		☑	☑

Figure 2.7: The Java System Properties page

5. Click the **System VM Parameters** tab page in the **Java System Properties** page (see Figure 2.7). Then, click the **Add** button therein to add a new Java parameter. This will open Figure 2.8.



Add JVM Additional Parameter

List of Supported JVM Additional Parameters

Name
-D<name>=<value>
-verbose[:class|gc|jni]
-X
-XX
-ea[:<packagename>...|:<classname>]

select this

Name: -Dcom.sun.management.jmxremote.port

add this value

Value: 5110

Description: set a system property

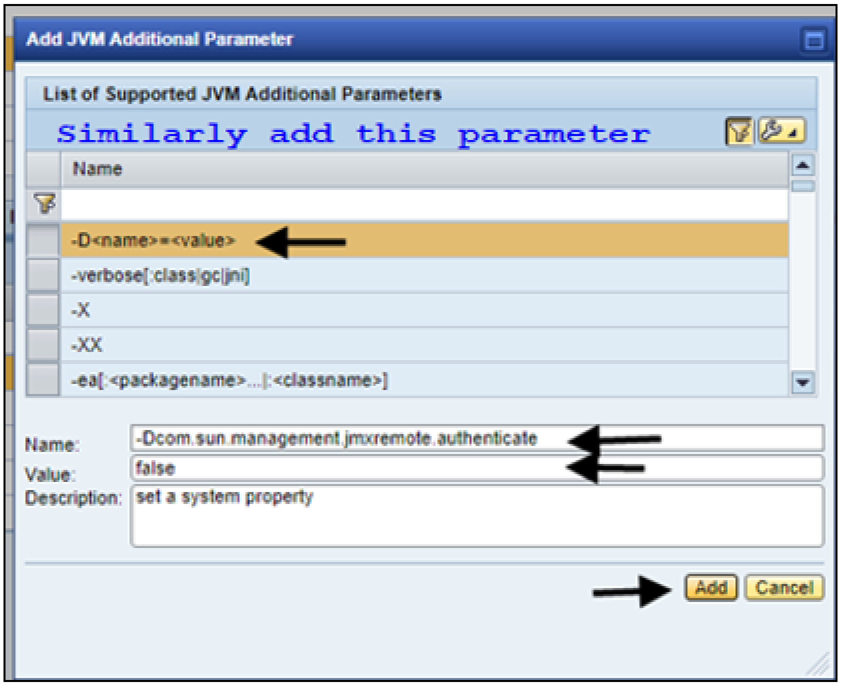
Figure 2.8: Adding the JMX port parameter

6. Start by adding the JMX port JVM parameter. This parameter specifies the port at which JMX listens. To add the parameter, first select *-D<name>=<value>*, and specify the following in the **Name** text box:

   -Dcom.sun.management.jmxremote.port

7. Next, in the **Value** text box, set any random number as the JMX port. This serves just as a 'placeholder' for the actual JMX port. Typically, if multiple Java server processes operate within a single instance, you will have to configure a separate JMX port for each server process. However, the port number specified in the **Value** text box applies to the entire instance - not to specific Java server processes. This is why, its best to configure a random number here. After enabling JMX support for the instance, you should make configuration changes to the instance offline using the Config Tool. In the process, you should configure a 'custom port format' for use by the individual Java server processes. SAP WAS auto-assigns a port number to each Java server process, based on the configured format. Steps 16-25 of this procedure detail how to use this tool to specify the format. For now however, click the **Add** button in Figure 2.8 to add the JMX port parameter with the random number.

8. Now, proceed to add the JMX authentication parameter. This parameter dictates whether/not JMX requires authentication. To add this parameter, once again click the **Add** button in Figure 2.7. Figure 2.9 will appear:


Add JVM Additional Parameter

List of Supported JVM Additional Parameters

Similarly add this parameter

Name
-D<name>=<value>
-verbose[:class|gc|jni]
-X
-XX
-ea[:<packagename>...|:<classname>]

Name: -Dcom.sun.management.jmxremote.authenticate

Value: false

Description: set a system property

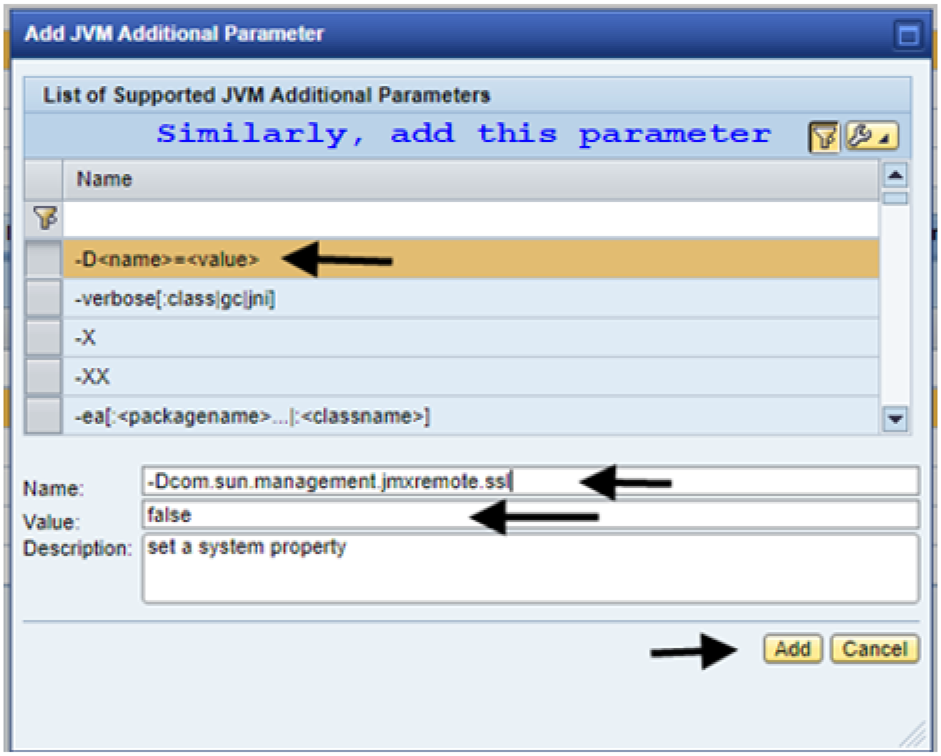
Figure 2.9: Adding JMX authentication parameter

9. Select *-D<name>=<value>* from the list box in Figure 5 and specify the following in the **Name** text box:

   -Dcom.sun.management.jmxremote.authenticate

10. Next, in the **Value** text box, type *false*. This implies that JMX does not require authentication. Finally, click the **Add** button in Figure 2.9 to add the new parameter.

11. Now, proceed to add the JMX SSL parameter. This parameter governs whether JMX is SSL-enabled or not. To add this parameter, once again click the **Add** button in Figure 2.7. Figure 2.10 will appear:



Add JVM Additional Parameter
List of Supported JVM Additional Parameters
Similarly, add this parameter

Name
-D<name>=<value>
-verbose[:class|gc|jni]
-X
-XX
-ea[:<packagename>...|:<classname>]

Name: -Dcom.sun.management.jmxremote.ssl

Value: false

Description: set a system property

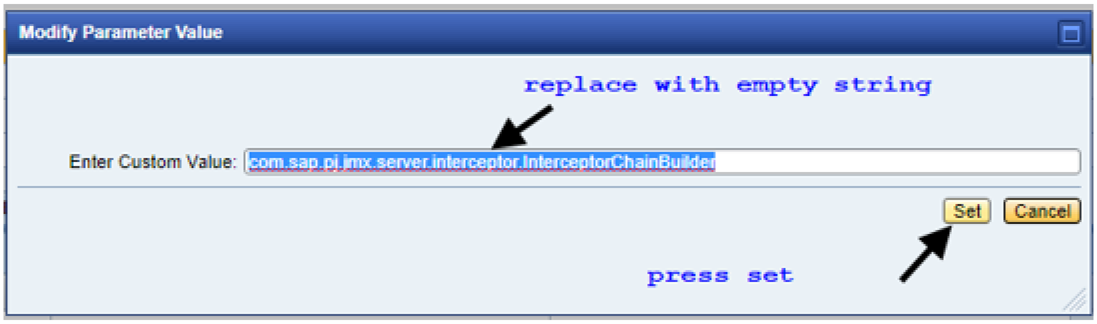
Figure 2.10: Adding JMX SSL parameter

12. From the list box in Figure 2.10, select *-D<name>=<value>* and specify the following in the **Name** text box

    -Dcom.sun.management.jmxremote.ssl

13. Next, in the **Value** text box, type *false*. This implies that JMX is not SSL-enabled. Finally, click the **Add** button in Figure 2.10 to add the new parameter.

14. All the parameters you added will then appear in Figure 2.7 with their default values. Now, browse the parameters list in the **System VM Parameters** tab page for the *javax.management.builder.initial* parameter. To modify the value of this parameter, first select the parameter and click the **Modify** button in Figure 2.7. Figure 2.11 will then appear. Here, replace the default value of the parameter with an empty string. Then, when prompted for confirmation to set an empty value for the parameter (see Figure 2.12), click **Yes** .

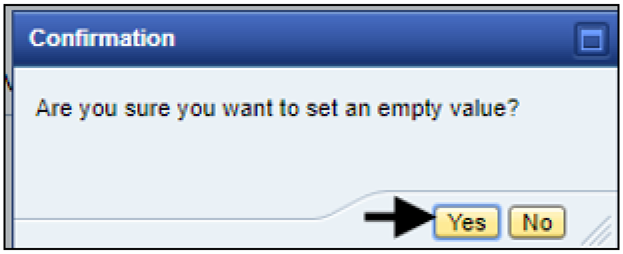


replace with empty string

Enter Custom Value: com.sap.pi.jmx.server.interceptor.InterceptorChainBuilder

press set

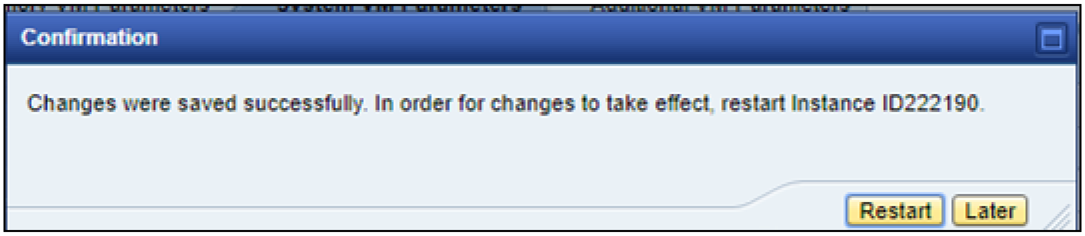
Figure 2.11: Modifying the value of a parameter



Are you sure you want to set an empty value?

Figure 2.12: Confirming the empty value setting

15. Clicking **Yes** in Figure 2.12 will take you back to Figure 2.7. Click the **Save** button in Figure 2.7 to save the changes. Figure 2.13 will then appear confirming the successful application of the changes. For the changes to take effect, click **Later** to restart the instance later.



Changes were saved successfully. In order for changes to take effect, restart Instance ID222190.

Figure 2.13: Choosing to restart the instance later

16. Next, proceed to start the **Config Tool**. For that, first open the command prompt of the server host.

17. To start the config tool on a Windows installation of the SAP WAS server, follow the steps below:

    - Switch to the <SAP_ WAS_ INSTALL_ DIR>\<SYSTEM_ NAME>\<INSTANCE_ NAME>\j2ee\configtool directory.

- Run the following command:

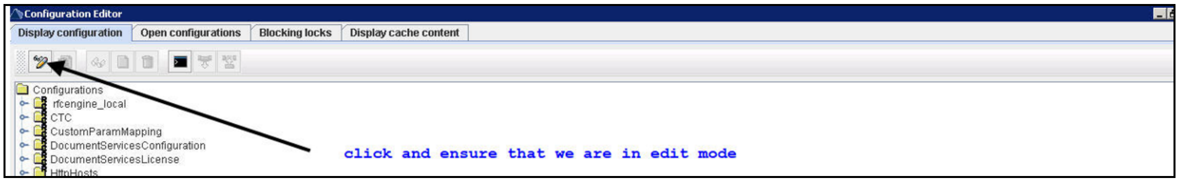
  configtool.bat

18. To start the config tool on a Unix installation of the SAP WAS server, follow the steps below:

    - Switch to the <SAP_ WAS_ INSTALL_ DIR>/<SYSTEM_ NAME>/<INSTANCE_ NAME>\j2ee\configtool directory.
    - Run the following command:

      ./configtool.sh

19. Figure 10 will then appear. Click the ✎ button at the right, top corner of Figure 2.14 to switch to the Edit mode.



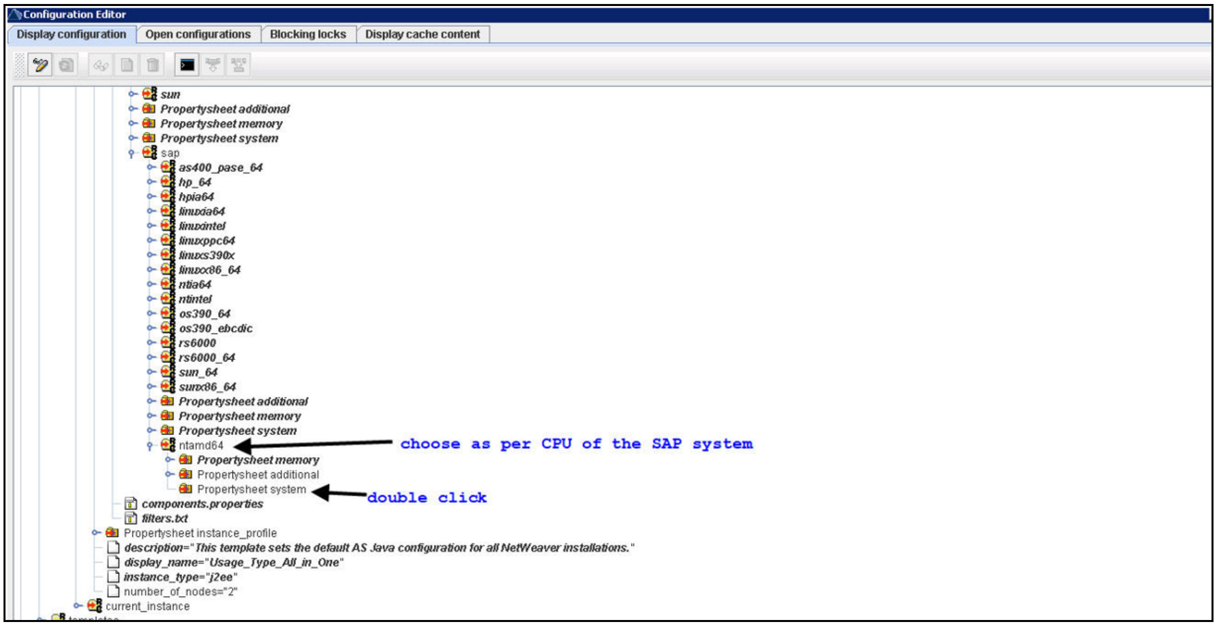
click and ensure that we are in edit mode

Figure 2.14: Checking whether the Edit mode is enabled

20. Next, follow this node-sequence on the tree-structure in Figure 2.14:

    Configurations -> cluster_config->instances -> <Name of instance to be modified> -> cfg -> jvm_params -> sap

    Under the **sap** node, expand the node that corresponds to the CPU of the SAP system being monitored. Then, double-click on **Propertysheet system** node within.



Configuration Editor

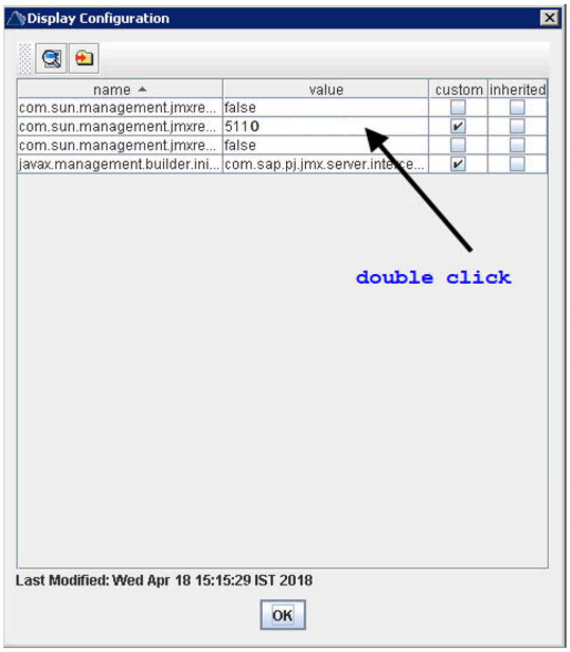
```
sun
Propertysheet additional
Propertysheet memory
Propertysheet system
sap
  as400_pase_64
  hp_64
  hpia64
  linuxia64
  linuxintel
  linuxppc64
  linuxs390x
  linuxx86_64
  ntia64
  ntintel
  os390_64
  os390_ebcdic
  rs6000
  rs6000_64
  sun_64
  sunx86_64
  Propertysheet additional
  Propertysheet memory
  Propertysheet system
  ntamd64
    Propertysheet memory
    Propertysheet additional
    Propertysheet system
components.properties
filters.txt
Propertysheet instance_profile
description="This template sets the default AS Java configuration for all NetWeaver installations."
display_name="Usage_Type_All_in_One"
instance_type="j2ee"
number_of_nodes="2"
current_instance
```

choose as per CPU of the SAP system

double click

Figure 2.15: Clicking on the Propertysheet system node

21. Figure 2.16 will then appear.



Display Configuration

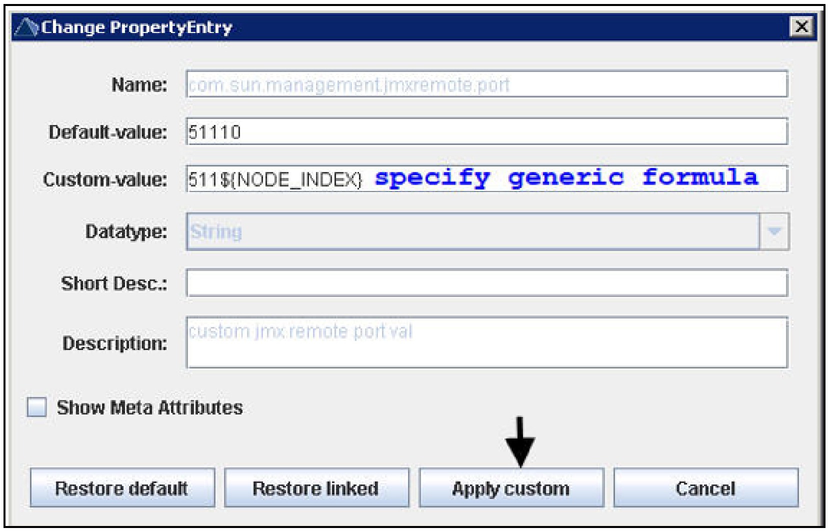
name ▲	value	custom	inherited
com.sun.management.jmxre...	false	□	□
com.sun.management.jmxre...	5110	☑	□
com.sun.management.jmxre...	false	□	□
javax.management.builder.ini...	com.sap.pj.jmx.server.interce...	☑	□

double click

Last Modified: Wed Apr 18 15:15:29 IST 2018

Figure 2.16: The Configuration of the chosen SAP system

22. From the list of JVM parameters in Figure 2.16, double-click on the JMX port parameter to edit its value. This will invoke Figure 2.17.



Name: com.sun.management.jmxremote.port

Default-value: 51110

Custom-value: 511${NODE_INDEX}

Datatype: String

Short Desc.:

Description: custom jmx remote port val

□ Show Meta Attributes

Figure 2.17: Providing a custom value for the JMX port parameter

23. In Figure 2.17, specify a 'custom port format' against **Custom Value**. SAP WAS uses this format specification to auto-generate the JMX port numbers and distribute them to each Java server process in the cluster.

Typically, each Java server process in a multi-server SAP WAS instance is assigned a node index - eg., the node index for server0 is 0, for server1 is 1, for server2 is 2, and so on. When configuring a JMX port for a server process, its recommended that the node index of that server process be set as the last digit of the JMX port number. It is also recommended that a static value, common to all server processes, precede this last digit, to form the complete port number. The format of your JMX port specification should therefore be: *<Static_value>${NODE_INDEX}*, where *<Static_value>* is an 'absolute' value, which will not change with Java server process. The JMX port number of every Java server process in the instance will begin with this *<Static_value>* only. *${NODE_INDEX}* is the 'variable' representing the node index of each Java server process. If this format is specified against **Custom Value**, then SAP WAS will auto-generate JMX ports according to this format and will distribute them to the Java server processes in the instance. For instance, assume that the **Custom Value** specification is *511${NODE_INDEX}*. This specification implies that the JMX port number of every server process in the cluster will begin with *'511'*. This value will be followed by the node index of the corresponding Java server process. Based on this format, the JMX port number of *server0* will be *5110*, *server1* will be *5111*, *server2* will be *5112*, and so on.

24. Then, click the **Apply custom** button in Figure 2.17 to apply the custom port format.
25. Finally, restart the SAP WAS server instance.

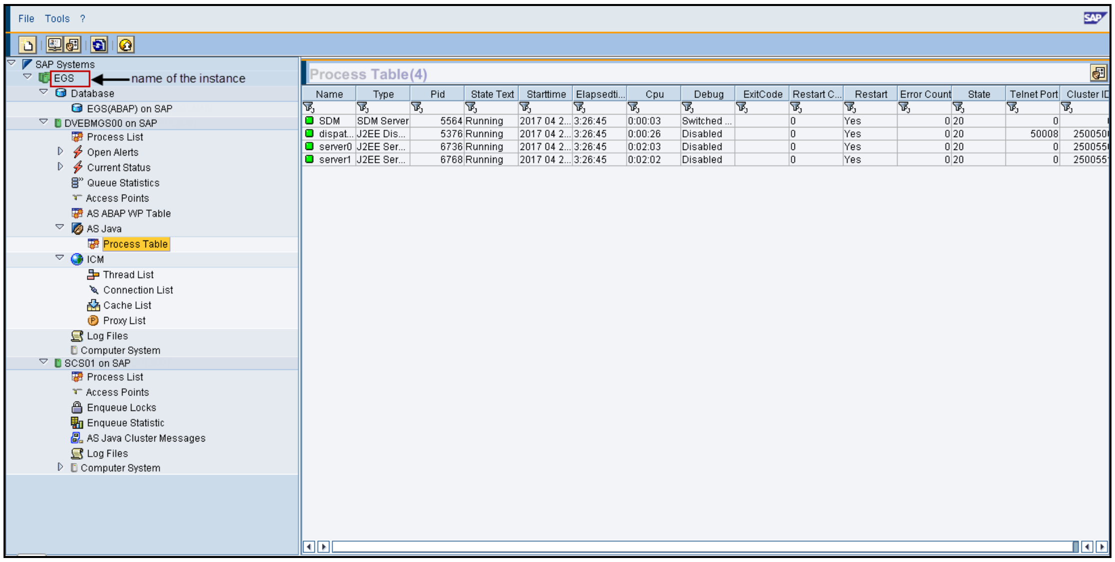
To monitor a multi-server SAP WAS instance, each server process in the instance should be added as a separate *Java Application Server* component in eG Enterprise. Then, when configuring the JVM tests for each such component, you need to make sure that the JMX port of the corresponding server process is correctly configured. Knowing the custom port format will help you determine the JMX port that you need to configure for a server process. You can use the Config Tool to view the custom port format specification and then figure out what JMX port you need to configure the JMX tests with. For instance, if the **Custom Value** specification as per Figure 2.17 is *'458${NODE_ INDEX}'*, and your are configuring a JVM test for the server process *server9*, then the JMX port specification will be *4589*.

2.3.2 kishtooDetermining the IDs and Ports Required for Monitoring

The following section explain you in great detail about how to determine the name of the SAP WAS instance to be monitored and the P4 protocol port.

2.3.2.1 Determining the Name of the SAP Web AS Instance

By default, the name of the instance will be the SID of the SAP Web Application Server. To know the available server instances, use the Programs -> SAP Management Console menu sequence on the application server host, and open the SAP Management Console. In the tree-structure in the left pane of the console (see Figure 2.18), you will find a **SAP Systems** node. When you expand the **SAP Systems** node, the available server instances will appear as its sub-nodes. Make note of the name of the instance that needs to be monitored. In Figure 2.18, the instance that is to be monitored is EGS. When configuring the eG tests for SAP Web AS, this name should be provided as the value of the **InstanceName** parameter.



Process Table(4)

Name	Type	Pid	State Text	Starttime	Elapsedti...	Cpu	Debug	ExitCode	Restart C...	Restart	Error Count	State	Telnet Port	Cluster ID
🟩 SDM	SDM Server	5564	Running	2017 04 2...	3:26:45	0:00:03	Switched ...		0	Yes	0	20	0	
🟩 dispat...	J2EE Dis...	5376	Running	2017 04 2...	3:26:45	0:00:26	Disabled		0	Yes	0	20	50008	250050
🟩 server0	J2EE Ser...	6736	Running	2017 04 2...	3:26:45	0:02:03	Disabled		0	Yes	0	20	0	250055
🟩 server1	J2EE Ser...	6768	Running	2017 04 2...	3:26:45	0:02:02	Disabled		0	Yes	0	20	0	250055

Figure 2.18: Determining the name of the SAP web Application server instance

2.3.2.2 Determining the P4 Port

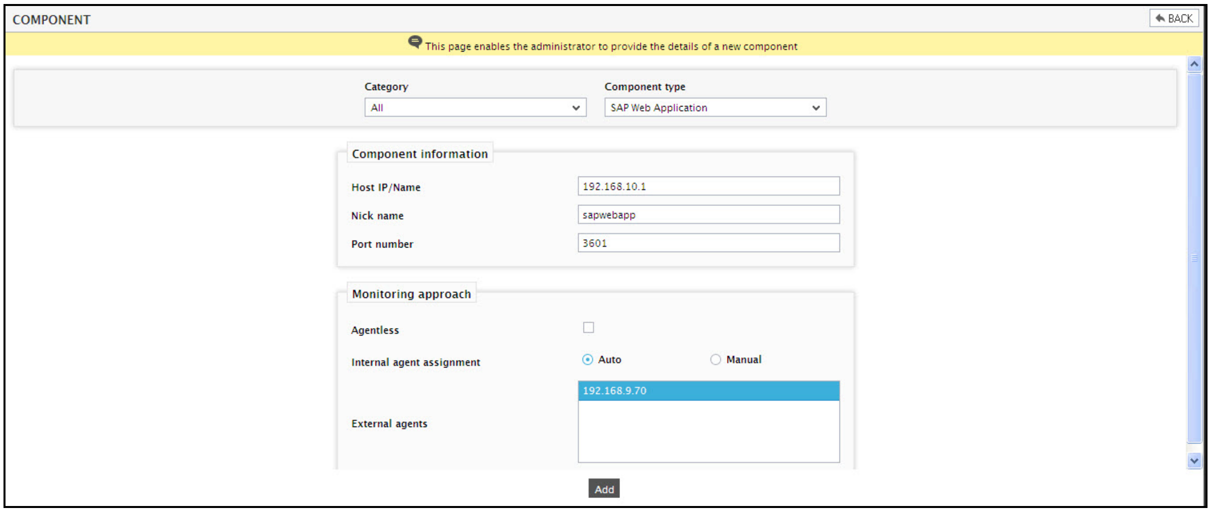
To know the P4 protocol port, refer to step 2 of the Section **2.3**.

2.4 Managing the SAP Web Application Server

eG Enterprise can automatically discover the SAP Web Application Server in the environment and also lets you to add the SAP Web Application Server component if the server is not auto-discovered. The following steps explain you how to manage the server that is auto-discovered and how to manually add the SAP Web Application Server using the eG administrative interface.

1. Log into the eG administrative interface.
2. If a SAP Web Application Server is already discovered, then directly proceed towards managing it using the **COMPONENTS – MANAGE/UNMANAGE** page (Infrastructure -> Components -> Manage/Unmanage).
3. However, if it is yet to be discovered, then run discovery (Infrastructure -> Components -> Discover) to get it discovered or add the component manually using the **COMPONENTS**page (Infrastructure -> Components -> Add/Modify). Remember that components manually added are managed automatically. Discovered components, however, are managed using the **COMPONENTS – MANAGE / UNMANAGE** page.
4. To add the component manually, follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu.

5. In the **COMPONENT** page that appears next, select *SAP Web Application Server* as the **Component type**. Then, click the **Add New Component** button. This will invoke Figure 2.19.



COMPONENT

This page enables the administrator to provide the details of a new component

Category: All

Component type: SAP Web Application

Component information

Host IP/Name: 192.168.10.1

Nick name: sapwebapp

Port number: 3601

Monitoring approach

Agentless: □

Internal agent assignment: ● Auto ○ Manual

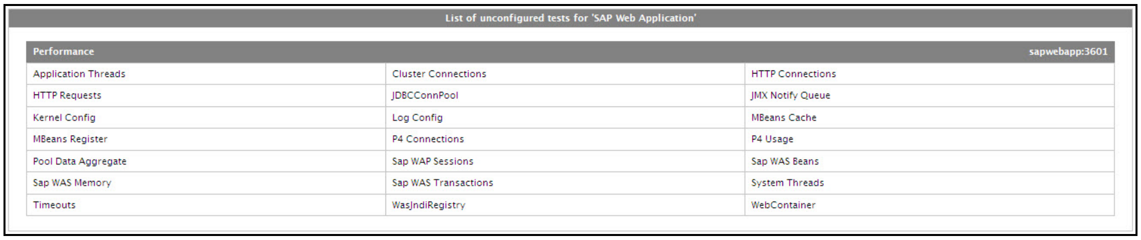
External agents: 192.168.9.70

Figure 2.19: Adding a SAP Web Application server

Note:

A SAP WAS cluster typically has one/more server processes running within a single SAP WAS instance. To monitor a SAP WAS cluster, each server process in the cluster should be added as a separate SAP Web Application Server component in eG Enterprise.

6. Then, try to sign out of the eG administrative interface. Upon doing so, Figure 2.20 will appear listing the unconfigured tests for the SAP Web Application server.



List of unconfigured tests for 'SAP Web Application'

Performance		sapwebapp:3601
Application Threads	Cluster Connections	HTTP Connections
HTTP Requests	JDBCConnPool	JMX Notify Queue
Kernel Config	Log Config	MBeans Cache
MBeans Register	P4 Connections	P4 Usage
Pool Data Aggregate	Sap WAP Sessions	Sap WAS Beans
Sap WAS Memory	Sap WAS Transactions	System Threads
Timeouts	WasJndiRegistry	WebContainer

Figure 2.20: The list of unconfigured tests for the SAP Web Application server

7. Click on any test in the list of unconfigured tests to configure. Refer to **Monitoring the SAP Web Application Server** chapter to know more details on configuring these tests.

8. Finally, signout of the eG administrative interface.

Chapter 3: Monitoring the SAP Web Application Server

The eG Enterprise suite embeds a specialized monitoring model for the SAP Web AS (see Figure 3.1), using which the performance of the critical services and components of the server can be tracked, issues affecting server-performance captured at their infancy, and the root-cause of the issues promptly traced and treated before it adversely impacts the transaction of business in the SAP environment.

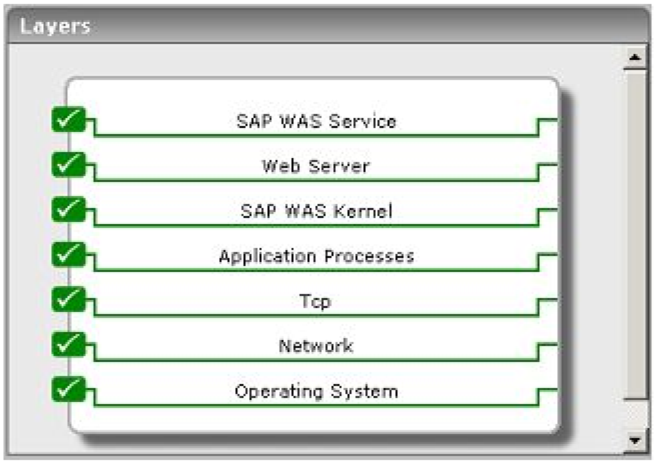


Figure 3.1: The layer model of the SAP Web AS

Every layer depicted by Figure 3.1 is associated with a series of tests, each of which seeks to answer the following questions related to the performance of the SAP Web AS:

- Are the thread managers of the SAP Web AS making optimum use of the threads in the pool?
- Are direct database accesses kept at a minimum?
- What type of connection requests are received by the SAP Web AS? How well does the server handle these requests?
- How is the memory usage of the Java objects managed by the SAP Web AS?
- Is the server cache been utilized effectively?
- Were any HTTP connections terminated abnormally by the SAP Web AS?
- Has the SAP Web AS been sized adequately to handle both the present and future logs?
- Has sufficient memory been allocated to the JVM?
- Have any new bean instances been added/removed from a bean pool? What are they?

- Are there any invalid user sessions on the SAP Web AS?
- How often do transaction rollbacks occur on the SAP Web AS?
- Are too many transactions/sessions getting timed out?
- Is the P4 connection to the server available? How quickly was the connection established?
- Is the web server component on the SAP Web AS accessible? What is its response time?

Since the last 4 layers of the layer model of the SAP Web AS have been dealt with to a great extent in the *Monitoring Unix and Windows Servers* document, the sections to come will discuss the first 3 layers only.

3.1 The Application Processes Layer

For a SAP Netweaver server, this layer is mapped to a **Processes** test, which will report on the status and resource usage of process patterns that have been explicitly configured for monitoring. In addition to the **Processes** test, a **SAP WAS Process Status** test also runs on this layer. The sub-section that follows discusses this test at length.

3.1.1 SAP WAS Process Status Test

A Java instance is a unit in the AS Java cluster which is identified by its instance number.

A Java Instance consists of:

- Internet Communication Manager (ICM)
- One or several server processes

The ICM is an element of the Java instance which handles requests coming from clients and dispatches them to the available server processes. The server processes of AS Java actually execute the Java application. They are responsible for processing incoming requests which are assigned to them by the ICM. Each server process is multi-threaded, and can therefore process a large number of requests simultaneously.

The proper functioning of a Java instance therefore depends upon the availability and error-free functioning of the ICM and server processes. If the ICM or any of the server processes experience errors or reboot often, then the Java instance may not be able to process requests from clients. To ensure the smooth transaction of business, administrators should periodically run the **SAP WAS Process Status** test on the target Java instance, check on the health state (whether Critical, Normal, Offline, or Online) and operational state (whether running or not running) of the ICM and each of the server processes in that instance, and capture process failures before they impact the critical business operations.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the ICM and each of the server processes in a target Java instance.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
WSDL Port	This test uses the SAPControl web service to pull metrics on application and service status. To enable the test to communicate with the web service, you need to configure the test with the port number of the web service. Therefore, specify the port number of the SAPControl web service against WSDL Port. To determine the exact port number of the SAPControl web service, you can look up the **etc/services** file on the SAP WAS being monitored. If the port number is not declared in the **etc/services** file, you can specify the default port number of the web service against WSDL Port. If the web service is not SSL-enabled, then the default port number of the web service will be: 5<NR>13. Similarly, if the web service is SSL-enabled, then the default port number of the web service will be: 5<NR>14. <NR> in the port number refers to the system number of the SAP server being monitored. The system number is an indicator of the TCP/IP port at which the SAP server listens. For example, for a server that listens at port 3200, the system number will be '00'. Similarly, if the SAP server port is 3201, the system number will have to be specified as '01'. Accordingly, the default port number of an SSL-enabled SAPControl web service will be 50014 , if the system number is 00, or 50114, if the system number is 01.
OS Username and OS Password	To enable the test to access the SAPControl web service and determine component state, you need to configure the OS Username and OS Password parameters of the test with the credentials (i.e., name and password) of an OS user, who fulfills one of the following conditions: • The name of the OS user should be declared using the service\admin_users profile parameter; (OR)

Parameter	Description
	- The name of the user group to which the OS user belongs should be declared using the service\admin_groups profile parameter; (OR) - The OS user should have the permission to execute the **sapstartsrv** executable
Confirm Password	Confirm the OS Password by retyping it here.
SSL	Set this flag to **Yes**, if the SAPControl web service is SSL-enabled. Set this flag to **No**, if the SAPControl web service is not SSL-enabled.
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD Frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: - The eG manager license should allow the detailed diagnosis capability - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Process status	Indicates the current health state of this process.		The values that this measure can report and their corresponding numeric values are discussed in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Normal</td><td>1</td></tr><tr><td>Critical</td><td>2</td></tr><tr><td>Error</td><td>3</td></tr><tr><td>Offline</td><td>4</td></tr></table>
Note:
By default, the test reports the **Measure Value**s in the table above to indicate process state. In the graph of this measure however, the same is represented using the numeric equivalents only.
If the value of this measure is Critical, Error, or Offline, you can use the detailed diagnosis of the measure to determine the probable cause of the abnormal process status. The detailed diagnosis will also reveal when the process was started and the elapsed time of the process. From this information, you can figure out when the problem could have occurred and how long the problem may have persisted.</td></tr>
<tr><td>Is process running?</td><td>Indicates whether/not this process is running currently.</td><td></td><td>The values that this measure can report and their corresponding numeric values are discussed in the table below:
<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>2</td></tr></table>
Note:</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>By default, the test reports the <b>Measure Value</b>s in the table above to indicate whether a process is running or not. In the graph of this measure however, the same is represented using the numeric equivalents only.</td></tr>
<tr><td>Has process restarted?</td><td>Indicates whether/not the process was restarted in the last measurement period.</td><td></td><td>The values that this measure can report and their corresponding numeric values are discussed in the table below:
<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>No</td><td>1</td></tr><tr><td>Yes</td><td>2</td></tr></table>
<b>Note:</b>
By default, the test reports the <b>Measure Value</b>s in the table above to indicate whether a process was restarted or not. In the graph of this measure however, the same is represented using the numeric equivalents only.</td></tr>
</table>

3.2 The SAP WAS Kernel Layer

The eG agents use JMX to obtain critical performance metrics from the SAP Web AS. At the **SAP WAS Kernel** layer, the agents execute a wide range of Kernel tests (see Figure 3.2) that extract performance statistics pertaining to the J2EE Engine managers such as:

- the Application Thread manager
- the Configuration manager
- the Connections Manipulator Manager
- the Pool manager
- the Threads manager

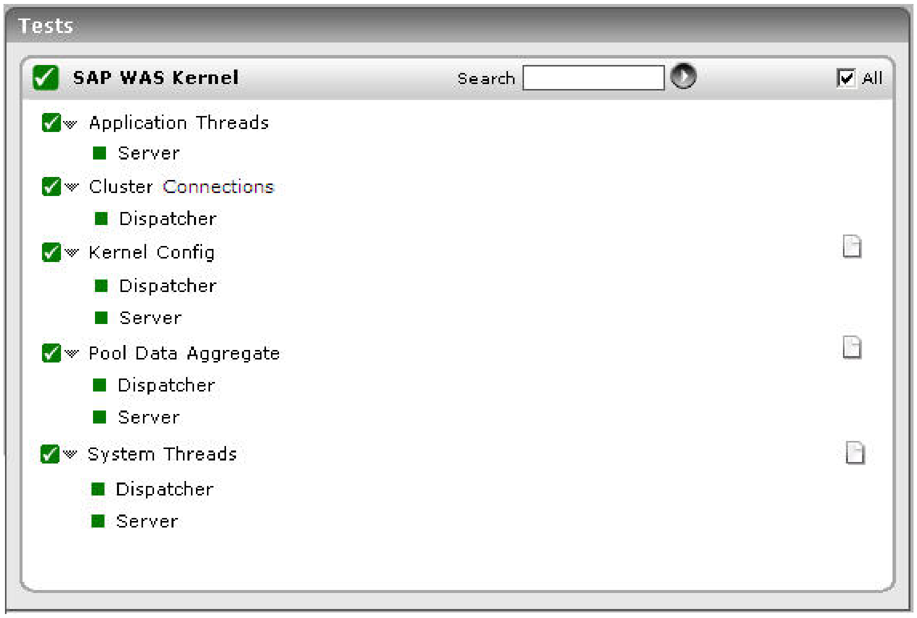


Figure 3.2: The tests associated with the SAP WAS Kernel layer

3.2.1 Kernel Config Test

This test monitors the Configuration Manager's interactions with the database. The Configuration Manager enables J2EE Engine modules to store and access data from a relational database management system (RDBMS). It provides properties for configuring a database connection.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.

Parameter	Description
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Commit duration	Indicates the time taken for a commit to take place	Secs	

Measurement	Description	Measurement Unit	Interpretation
	for data storage in the last measurement period.		
Locked exception ratio	Indicates the number of database lock exceptions that occurred for every database access in the last measurement period.	Exceptions/Db_tries	
Cache hit rate	Indicates the percentage of time data was retrieved from the database cache in the last measurement period.	Percent	Ideally, the ratio of cache hits to database accesses should be high, as direct database accesses are expensive operations.
Db read ratio	Indicates the number of database reads that occurred for every database access in the last measurement period.	Reads/Db_tries	
Db write ratio	Indicates the number of database writes that occurred for every database access in the last measurement period.	Writes/Db_tries	

3.2.2 Application Threads Test

The J2EE Engine thread system is responsible for handling system and client threads. It comprises of two managers – Thread Manager and Application Thread Manager. The **Application Threads** test monitors the Application Thread Manager of the SAP Web Application server, which supplies the threads in which the client applications' source code is executed. This manager provides a set of properties for starting and managing client threads in the Java Virtual Machine. When a client request comes, the system tries to find a free thread in the Application Thread Manager and to start the execution of the request. If no free thread is available, the thread system buffers the request in a request queue. By buffering the threads and using them again, the system achieves better performance than using normal Java thread system without buffering. The Application Thread Manager runs only on server processes.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results each for the Dispatcher and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs.

Parameter	Description
	Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process.* For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Active threads	Indicates the number of threads from the thread pool which are currently executing a runnable task.	Number	This measure serves as a good indicator of the server workload.
Current pool size	Indicates the number of threads that are currently in the thread pool.	Number	
Threadpool usage	Indicates the percentage of threads in the thread pool that are being currently utilized.	Percent	If the value of this measure is high, it could indicate a heavy server workload.
Waiting tasks	Indicates the number of tasks waiting for threads, so as to begin execution.	Number	
Tasks queue size	Indicates the capacity of the request queue where the tasks waiting for execution are stored.	Number	
Task queue overflow	Indicates the number of tasks waiting to be placed in the request queue, in the event that the request queue is full.	Number	If the value of this measure increases consistently, it could indicate a processing bottleneck.

3.2.3 Cluster Connections Test

The Cluster Connections test monitors the SAP J2EE engine's Connections Manipulator Manager, which manages the client connections to the cluster by providing a set of properties for managing the pools where the TCP connection objects are stored. The Connections Manipulator Manager runs on dispatchers only.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results each for the Dispatcher and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to

Parameter	Description
	monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Current pool size	Indicates the current size of the TCP connections pool.	Percent	
HTTP connections	Indicates the number of TCP connections in the pool that are currently been utilized for servicing HTTP requests.	Number	
P4 connections	Indicates the number of TCP connections in the pool that are currently been utilized for servicing P4 requests.	Number	
IIOP connections	Indicates the number of TCP connections in the pool that are currently been utilized for servicing IIOP	Number	

Measurement	Description	Measurement Unit	Interpretation
	requests.		
Unrecognized connections	Indicates the number of TCP connections in the pool that are currently being utilized for servicing requests of an unknown kind.	Number	

3.2.4 Pool Data Aggregate Test

The Pool Data Aggregate test reports metrics pertaining to the J2EE Engine Pool Manager, which facilitates the centralized creation and reuse of byte arrays.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results each for the Dispatcher and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the

Parameter	Description
	Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Pool memory allocated	Indicates the total memory that is allocated by the Pool Manager to the Java objects that it manages.	MB	
Pool memory used	Indicates the total memory that is used by the Java objects managed by the Pool Manager.	MB	

3.2.5 System Threads Test

The J2EE Engine thread system is responsible for handling system and client threads. It comprises two managers – Thread Manager and Application Thread Manager. The System Threads test monitors the health of the Thread Manager, which supplies the threads in which SAP J2EE Engine system operations are executed. This manager provides a set of properties for starting and managing system threads. This thread pool is for system activities such as making backup, background optimizations for load/store data, and so on. The logic in the system thread manager is similar to the application logic – the system uses a queue for system requests if a free thread is not available.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results each for the Dispatcher and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in

Parameter	Description
	Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Active threads	Indicates the number of threads from the thread pool which are currently executing a runnable task.	Number	This measure serves as a good indicator of the server workload.
Current pool size	Indicates the number of threads that are currently in the thread pool.	Number	
Threadpool usage	Indicates the percentage of threads in the thread pool that are being currently utilized.	Percent	If the value of this measure is high, it could indicate a heavy server workload.
Tasks waiting	Indicates the number of	Number	

Measurement	Description	Measurement Unit	Interpretation
	tasks waiting for threads, so as to begin execution.		
Tasks queue size	Indicates the capacity of the request queue where the tasks waiting for execution are stored.	Number	
Tasks queue overflow	Indicates the number of tasks waiting to be placed in the request queue, in the event that the request queue is full.	Number	If the value of this measure increases consistently, it could indicate a processing bottleneck.

3.3 The Web Server Layer

An external **Http** test executes on this layer (see Figure 3.3), which emulates a user access to the web server component of the SAP Web AS, and reports the web server's availability and responsiveness.

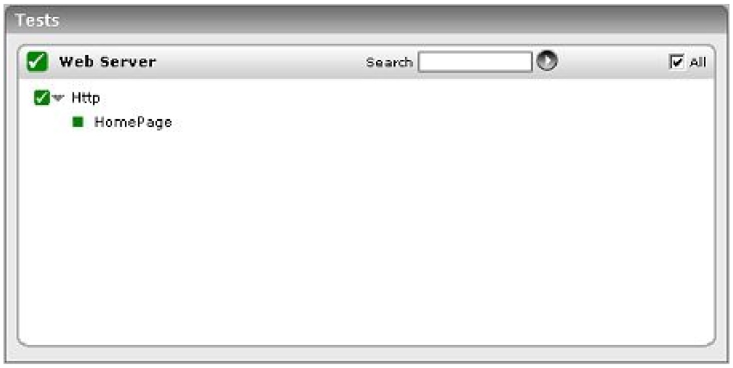
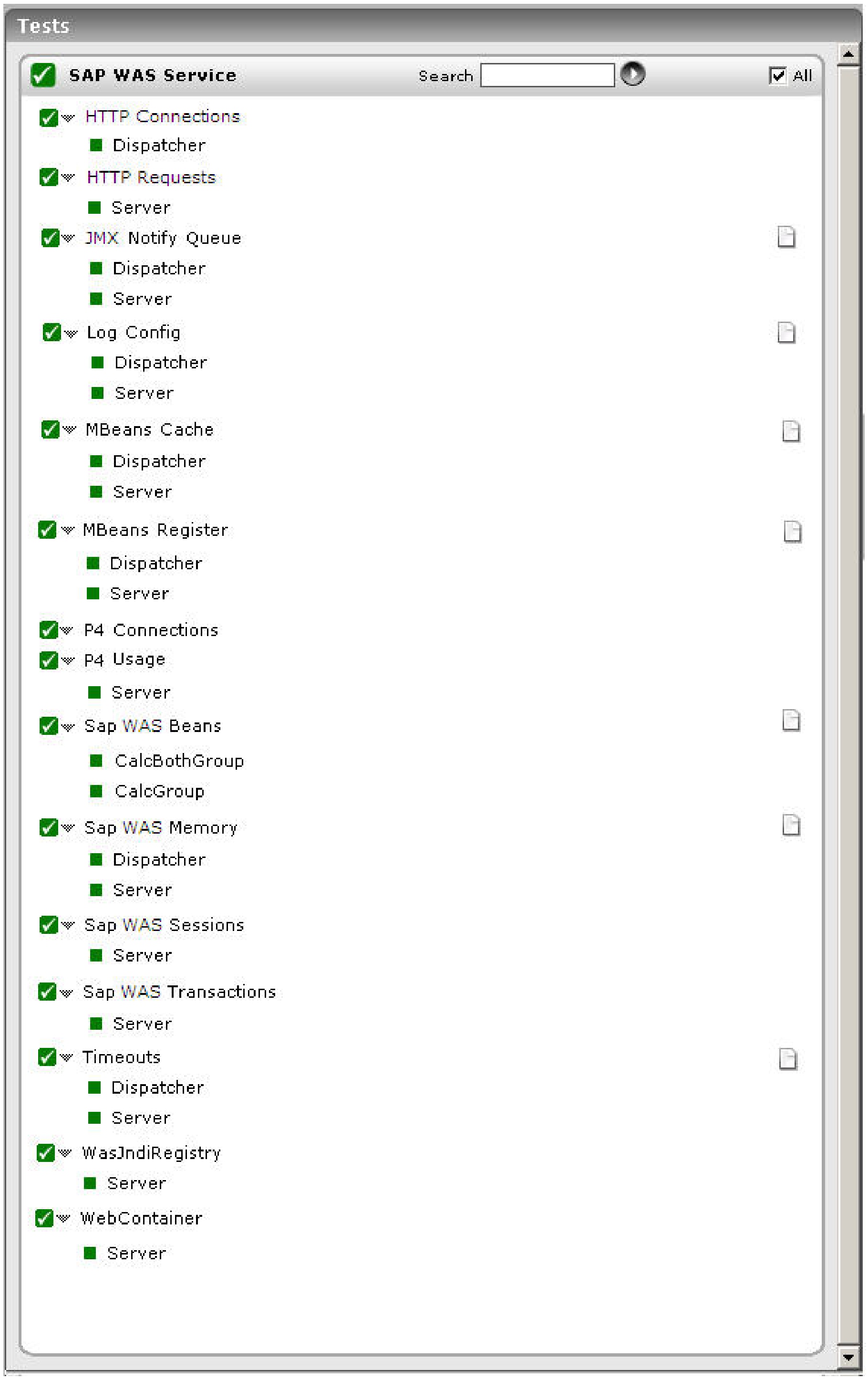


Figure 3.3: The test executing on the Web Server layer

Since the Http test has been dealt with in *Monitoring Apache Web Servers* document, let us focus on the sap was Service layer.

3.4 The SAP WAS Service Layer

The tests mapped to the **SAP WAS Service** layer (see Figure 3.4) extract critical performance metrics relating to the services running on a SAP Web AS, where each service performs an application function.



- ☑ HTTP Connections
  - 🟩 Dispatcher
- ☑ HTTP Requests
  - 🟩 Server
- ☑ JMX Notify Queue
  - 🟩 Dispatcher
  - 🟩 Server
- ☑ Log Config
  - 🟩 Dispatcher
  - 🟩 Server
- ☑ MBeans Cache
  - 🟩 Dispatcher
  - 🟩 Server
- ☑ MBeans Register
  - 🟩 Dispatcher
  - 🟩 Server
- ☑ P4 Connections
- ☑ P4 Usage
  - 🟩 Server
- ☑ Sap WAS Beans
  - 🟩 CalcBothGroup
  - 🟩 CalcGroup
- ☑ Sap WAS Memory
  - 🟩 Dispatcher
  - 🟩 Server
- ☑ Sap WAS Sessions
  - 🟩 Server
- ☑ Sap WAS Transactions
  - 🟩 Server
- ☑ Timeouts
  - 🟩 Dispatcher
  - 🟩 Server
- ☑ WasJndiRegistry
  - 🟩 Server
- ☑ WebContainer
  - 🟩 Server

Figure 3.4: The tests associated with the SAP WAS Service layer

3.4.1 MBeans Cache Test

This test monitors the accesses to the MBeans cache, where the MBeans are created and administered by the JMX Adapter Service. The JMX Adapter Service manages the configuration and lifecycle of the MBeanServer and provides access to it for applications, services, and libraries.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results each of the Dispatcher and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this

Parameter	Description
	parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Cache hit rate	Indicates the ratio of the number of cache hits to the total number of accesses to the MBeanServer.	Hits/tries	Ideally, this ratio should be high. Direct server accesses are expensive operations, and hence need to be kept at the minimum.

3.4.2 HTTP Connections Test

The HTTP Provider Service represents a server socket that listens for client HTTP connections on the J2EE Engine. It takes care of parsing the URL of the incoming HTTP requests, dispatching them to the correct J2EE Engine's module for processing, and returning the generated responses back to the client. The HTTP Connections test monitors the HTTP Provider Service and reports key statistics pertaining to client HTTP connections on the J2EE engine.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these

Parameter	Description
	cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
New requests	Indicates the number of HTTP requests newly received from client since the last measurement period.	Number	
Connections keep alive	Indicates the number of TCP connections that are currently open.	Number	Under HTTP 1.0, if the browser supports keep-alive, it adds an additional header to the request: “Connection: Keep-Alive”. Then, when the server receives this request and generates a response, it also adds a header to the response: “Connection: Keep-Alive”. Following this, the connection is NOT dropped, but is instead kept open. When the client sends another request, it uses the same connection. This will continue until either the client or the server decides that the conversation is over, and one of them drops the connection. Under HTTP 1.1, all connections are kept alive by default, unless stated otherwise with the following header: “Connection: close”. The “Connection: Keep-Alive” header no longer has any meaning because of this.
Connections closed by client	Indicates the number of TCP connections that	Number	

Measurement	Description	Measurement Unit	Interpretation
	have been closed by the client since the last measurement period.		
Connections closed by server	Indicates the number of TCP connections closed by the server since the last measurement period.	Number	When the server closes a TCP connection, it could be a normal connection timeout or an abnormal termination. In case of the latter, the reasons for the abnormal connection loss will have to be investigated.

3.4.3 HTTP Requests Test

The HTTP Provider Service represents a server socket that listens for client HTTP connections on the J2EE Engine. It takes care of parsing the URL of the incoming HTTP requests, dispatching them to the correct J2EE Engine's module for processing, and returning the generated responses back to the client. The HTTP Requests test monitors how well the HTTP Provider Service responds to the HTTP connection requests received from clients.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.

Parameter	Description
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
New HTTP requests	Indicates the number of HTTP requests received by the server since the last	Number	

Measurement	Description	Measurement Unit	Interpretation
	measurement period.		
Responses from cache	Indicates the number of times the server has retrieved information from the cache to respond to HTTP requests from clients, since the last measurement period.	Number	Ideally, this value should be high.

3.4.4 JMX Notify Queue Test

The JMX Notification Service is responsible for the distribution of the MBeanServer notifications throughout the cluster. MBeanServer notifications inform all the clients within the cluster about recently registered MBeans and the removal of MBeans from the MBeanServer. This test reports metrics that indicate how well the JMX Notification Service performs.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect.

Parameter	Description
	For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Active threads	Indicates the number of threads that are currently processing JMX notifications.	Number	
Queue size	Indicates the number of	Number	If the value of this measure increases

Measurement	Description	Measurement Unit	Interpretation
	notifications currently in queue, waiting to be sent to clients.		consistently, it could indicate a processing/delivery bottleneck.

3.4.5 Log Config Test

Using the Log Configurator service runtime available in the Visual Administrator, one can manage the logging and tracing configurations of the J2EE Engine components and of the deployed applications. The LogConfig test monitors this service to indicate whether the SAP web application server has been adequately configured to handle both its present and future logs.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server.

Parameter	Description
	Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Archives size	Indicates the total size of all log and trace archives.	MB	
Log files size	Indicates the total size of all log and trace files excluding archives.	MB	

3.4.6 MBeans Register Test

This test monitors the process of the registration of MBeans with the MBeanServer.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs.

Parameter	Description
	Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process.* For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Notify listeners	Indicates the number of notification listeners registered with the MBeanServer since the last measurement period.	Number	Adding a listener for MBeanServerNotifications causes the MBeanServer to send a broadcast message for each MBean that is (un)registered. Therefore, avoid it whenever possible and do not forget to unregister the listener if you no longer require notification.
Mbeans registered	Indicates the number of Mbeans registered with the MBeanServer since the last measurement period.	Number	

3.4.7 P4 Connection Status Test

This test reveals the availability and responsiveness of the P4 connection to the SAP Web AS.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results each for every SAP Web AS monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server*

Parameter	Description
	process name:Cluster ID of the server process. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Availability	Indicates whether the P4 connection is available or not.	Percent	If the value of this measure is 100, it indicates that the P4 connection is available. The value 0, on the other hand, is indicative of the non-availability of the P4 connection to the server.
Response time	Indicates the responsiveness (in seconds) of the P4 protocol.	Secs	

3.4.8 P4 Usage Test

The P4Usage test measures the capability of the P4 Provider Service in handling P4 requests. The P4 Provider Service provides functions for the communication of remote objects over the P4 protocol on the J2EE Engine. It also provides functions for communication support generation. The P4 Provider Service consists of two parts. One of the parts runs on the Java dispatcher. It accepts requests from the remote clients and has the responsibility to dispatch them to the appropriate cluster element, where the implementation of the remote object resides. For the purpose, the P4 Provider Service uses the J2EE Engine Load Balancing System. The second part of the P4 Provider Service runs on the server processes. This part contains the implementation of the P4RemoteObject broker and is responsible for executing methods on the implementation of the remote object and returning the results.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server*

Parameter	Description
	process name:Cluster ID of the server process. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Remote objects	Indicates the number of remote objects handled by the P4 Provider Service.	Number	
P4 requests	Indicates the number of P4 requests received from remote clients since the last measurement period.	Number	
Failed requests	Indicates the number of P4 requests which have failed since the last measurement period.	Number	

3.4.9 SAP WAS Beans Test

This test measures the efficiency with which the EJB Container Service manages the enterprise bean instances deployed on the server. The EJB Container provides all the services that are required by an EJB application, such as transaction and security management, clustering, persistence, network distribution of remote clients, scalable management of resources, and so on.

If too many EJBs have been deployed on the server, then managing the individual EJBs could become a cumbersome task. In such a case, you can use the eG administrative interface to group EJBs and manage the groups, instead. To create an EJB group, you will need to click on the **Click here** hyperlink displayed above the parameters of the **SAP WAS Beans** test. **By default, eG Enterprise system monitors only those beans that are part of a group.**

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every configured EJB group.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server*

Parameter	Description
	process name:Cluster ID of the server process. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.
Autodiscovery	By default, the eG Enterprise suite allows administrators to configure bean groups using the eG administrative interface, and reports metrics pertaining to every group so created. Accordingly, by default, Autodiscovery is set to **No**. If you want beans to be discovered and monitored automatically, then select the **Yes** option against Autodiscovery. When this is done, the eG agent automatically discovers all the beans on the SAP web application server, and reports one set of measures for every bean hosted on the server.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Bean creations	Indicates the number of times a bean has been created.	Number	The detailed diagnosis of this measure, if enabled, provides an individual bean-wise breakup of the current pool size, and the number of additions and removals.
Bean removals	Indicates the number of times a bean has been removed.	Number	
Current pool size	Indicates the number of bean instances provided	Number	

Measurement	Description	Measurement Unit	Interpretation
	by this pool, which are currently used by the application or are stored in the pool.		
Bean activations	Indicates the number of times a bean instance has been activated since the last measurement period.	Number	
Bean passivations	Indicates the number of times a bean instance has been passivated or deactivated since the last measurement period.	Number	
Active sessions	Indicates the number of bean sessions that are currently active.	Number	
Passive sessions	Indicates the number of bean sessions that are currently passive.	Number	
Completed sessions	Indicates the number of bean sessions that have been completed since the last measurement period.	Number	
Bean stores	Indicates the number of entity bean instances that were stored in the database since the last measurement period.	Number	
Bean loads	Indicates the number of entity bean instances that were loaded to the EJB container from the database since the last measurement period.	Number	

3.4.10 SAP WAS Memory Test

This reports the memory usage metrics revealed by the Memory Info Service. This service is responsible for keeping track of the memory that is used internally by the JVM of the owner cluster element. The service provides a set of properties for managing memory usage levels.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this

Parameter	Description
	parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Memory used	Indicates the memory that has been currently utilized by the JVM.	MB	
Memory allocated	Indicates the amount of memory that has been allocated to the JVM.	MB	
Memory available	Indicates the amount of memory that is currently available.	MB	If this value is very low or is decreasing consistently, it could be a cause for concern. You might then consider allocating more memory to the JVM of the owner cluster element.

3.4.11 SAP WAS Sessions Test

This test reports metrics pertaining to user sessions on the SAP web application server as revealed by the Security Provider service. This service enables the management of the security policy and the authentication and authorization mechanisms on the system, monitors user sessions, and restricts access to the resources or the applications deployed on the J2EE Engine.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to

Parameter	Description
	the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Failed logon attempts	Indicates the number of unsuccessful logon attempts since the last measurement period.	Number	
New sessions	Indicates the number of sessions that have newly opened since the last measurement period.	Number	
Active sessions	Indicates the number of currently active sessions.	Number	
Invalid sessions	Indicates the number of invalid sessions since the last measurement period.	Number	
Logged off sessions	Indicates the number of sessions users logged out of since the last measurement period.	Number	
Timed out sessions	Indicates the number of sessions that timed out.	Number	If the value of this measure is very high, then consider resetting the Timeout period for user sessions.

Measurement	Description	Measurement Unit	Interpretation
Logged in users	Indicates the number of users logged in since the last measurement period.	Number	
Opened sessions	Indicates the number of web sessions that were opened during the last measurement period.	Number	
Active web sessions	Indicates the number of web sessions that were active during the last measurement period.	Number	
Security sessions	Indicates the number of security sessions opened during the last measurement period.	Number	Security sessions are objects that store authentication information for users for their various HTTP sessions. Each security session corresponds to at least one HTTP session. If a security session is invalidated, so are the associated HTTP sessions.
EJB sessions	Indicates the number of EJB sessions opened during the last measurement period.	Number	EJB sessions are Enterprise JavaBean sessions which may be stateful (bean state is preserved and stored across method invocations) or stateless (bean state is not preserved across method invocations).

3.4.12 SAP WAS Transactions Test

This test monitors the transactions to the SAP web application server, and reports key statistics pertaining to the transactions.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server*

Parameter	Description
	process name:Cluster ID of the server process. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Active transactions	Indicates the number of transactions that are currently active.	Number	
Commit transactions	Indicates the number of transactions that have been committed since the last measurement period.	Number	
Rollback transactions	Indicates the number of transactions that have been rolled back since the last measurement period.	Number	Ideally, there should be fewer rollbacks happening, as rollbacks are costly operations on the database.
Suspend transactions	Indicates the number of suspended transactions since the last measurement period.	Number	
Timedout transactions	Indicates the number of transactions that have timed out since the last measurement period.	Number	If the value of this measure is very high, then consider resetting the Timeout period for transactions.

3.4.13 Timeouts Test

The Timeouts test monitors the Timeout service, which provides an open structure for registering numerous listeners willing to perform time-base actions. This service is a non-distributed system for scheduling tasks for future execution in a background thread. A special Timeout object represents each task. If you want to reuse a particular thread, you can specify a timeout after which the same thread can be reused. The Timeout Service is used by other J2EE Engine services that need to receive events at particular intervals. The service can be used for accomplishing regular operations – checking the module status, logging information about the current load, and so on.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	**Description**
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of

Parameter	Description
	these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Timeout rate	Indicates the count of timeout events received by a J2EE Engine service per minute.	Timeouts/min	If the timeout value set for the Timeout objects is high, then the value of this measure will be low. Since the Timeout Service optimizes thread utilization by running multiple application tasks simultaneously, it would be good practice to create more Timeout objects and set a low timeout value for the objects.

3.4.14 SAP WAS JNDI Registry Test

This monitors the JNDI Registry Service, which provides a way by which names are associated with objects, and objects are found based on their names. It provides a set of properties for specifying the number of trials for locking an object in a database, for assigning communication protocols to be used by this service, and for specifying the method of the lookup process. The JNDI Registry runs on server processes only.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server*

Parameter	Description
	process name:Cluster ID of the server process. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Bound objects	Indicates the total number of objects that are bound in the naming tree, including the serializable and non-serializable objects.	Number	This measure gives an idea about the application runtime.
Byte array cache size	Indicates the size of the byte array cache.	MB	

3.4.15 WebContainer Test

This test monitors the functions of the Web Container Service, which manages J2EE Web components across a cluster environment, generates dynamic responses, and so on. This service enables the life cycle of Web applications to be managed. It also helps developing and running session- and security-aware Web applications.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each of the Dispatchers and each of the configured server processes of the SAP Web AS.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts

Parameter	Description
	critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process.* For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
New requests	Indicates the number of requests newly received by the Web Container Service since the last measurement period.	Number	
Current http sessions	Indicates the number of HTTP sessions that are currently valid.	Number	
Current secure sessions	Indicates the number of valid security sessions currently created for HTTP clients.	Number	
Invalid http sessions	Indicates the number of HTTP sessions invalidated by application since the last measurement period.	Number	
Invalid secure sessions	Indicates the number of security sessions which have been invalidated by application since the last measurement period.	Number	
Timeout http sessions	Indicates the number of HTTP sessions which have timed out since the last measurement period.	Number	
Timeout secure sessions	Indicates the number of secure sessions which have timed out since the last measurement period.	Number	

3.4.16 JDBC Connection Pools Test

Sufficient free connections should be available in the JDBC connections pools on a SAP web application server, so that applications deployed on the server are able to communicate with the database server without any interruption. If a connection pool runs out of free connections, then application- database interactions will be adversely impacted, thereby significantly degrading application performance. This is why, it is imperative that administrators continuously track the usage of JDBC connection pools on the SAP web application server, and proactively detect the shortage of free connections in any connection pool. The **JDBC Connection Pools** test helps administrators in this mission! This test monitors the usage of the standalone and application portal JDBC connection pools, and accurately pinpoints the connection pool that is being over-utilized and is hence running out of free connections. The test also accurately brings out how the absence of free connections is impacting application server performance, and provides useful pointers to how this contention can be cleared.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results each for the standalone and application portal JDBC connection pools.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
UserName	This test connects to a specific SAP web application server instance, and extracts critical metrics from it. Therefore, in the UserName text box, provide a valid **j2ee admin** user name which the test should use for connecting to the server instance.
Password	Provide the Password that corresponds to the specified UserName.
Confirm Password	Confirm the OS Password by retyping it here.
InstanceName	Provide the name of the application server instance to which the test should connect. For example, if you specify *tpp* here, then the test will use the login credentials (UserName and Password) provided here to connect to an instance named *TPP*. To know the available server instances, use the instructions provided in Determining the Name of the SAP Web AS Instance.
ConnectorPort	This test uses the P4 protocol for connecting to the SAP web application server. Therefore, the port at which the P4 protocol listens needs to be specified as the ConnectorPort. To know how to determine the P4 port, use the procedure detailed in Determining the P4 Port.

Parameter	Description
DispatcherID	This test reports a set of metrics for the Java dispatcher that receives client requests. By default, the cluster ID of the dispatcher is auto-discovered by the eG Enterprise. Therefore, *default* option is specified against this parameter. Alternately, if you wish to monitor a specific cluster ID, then you can mention that cluster ID against this parameter. To know the cluster ID of the dispatcher, follow the steps discussed in Determining the Cluster ID of the J2EE Dispatcher .
ServerIDs	Typically, the Java dispatcher or the ICM distributes the client requests it receives to the server processes executing on the Java component of the SAP Web AS for processing. By default, the ServerIDs are auto-discovered by the eG Enterprise. To monitor specific server processes, specify a comma-separated list of the cluster IDs of these server processes in the ServerIDs text box. To know the cluster IDs of the server processes, follow the procedure detailed in Determining the Server IDs. Providing a comma-separated list of server process cluster IDs will ensure that these cluster IDs alone appear as the descriptors of the test. If need be, you can have server process name-cluster ID pairs appear as the test descriptors. To achieve this, the specification in the ServerIDs text box should be of the following format: *Server process name:Cluster ID of the server process*. For example, if the cluster ID of a server process named *Server0* is *12621850*, then you can specify the ServerID in the format: *Server0:12621850*.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Used connections	Indicates the number of connections in this pool that are currently in use.	Number	
Free connections	Indicates the number of unused managed connections in this pool.	Number	Ideally, the value of this measure should be high. A consistent drop in the value of this measure is indicative of excessive usage of the pool, which may result in a serious contention for free managed connections. Many connection requests
Maximum connections	Indicates the maximum number of connections from a single datasource	Number	If the value of this measure is very low, it means that the Maximum

Measurement	Description	Measurement Unit	Interpretation
	that this pool can have.		Connections parameter of the connection pool has been set very low. A very low value for this measure may cause the pool to rapidly run out of connections. Many connection requests will then end up waiting for a connection to be released to use it. This in turn may result in long-winding connection queues on the application server, which may eventually choke the server. This is why, it is important that the Maximum Connections parameter is set after careful thought. Set a higher value for this parameter if the applications need to connect multiple times to the database server, or if there are many clients working with the database server simultaneously.
Open connections	Indicates the number of connections opened per application.	Number	
Waiting requests	Indicates the number of connection requests that are waiting for connections to be released.	Number	A high value for this measure is a cause for concern, as it indicates that many requests are waiting for connections to be released. If the number of waiting requests increases steadily, the application server may choke. The probable causes for a large number of waiting requests are as follows: • A low Maximum connections parameter setting; this may cause the pool to rapidly run out of connections. Many connection

Measurement	Description	Measurement Unit	Interpretation
			requests will then end up waiting for a connection to be released to use it. This in turn may result in long-winding connection queues on the application server, which may eventually choke the server. A high Maximum connections configuration may help reduce the number of waiting requests. • A high Maximum time to wait for connection configuration. This setting indicates the maximum time for which the clients will wait, if the Maximum connections setting is reached and there are no free connections. With a high wait time setting, requests from clients will wait longer, increasing the number of waiting requests. By reducing the Maximum time to wait for connection, you may be able to decrease the number of waiting requests.
Managed connections usage	Indicates what percentage of the maximum connection capacity of this pool, is currently in use.	Percent	A low value is desired for this measure. A value close to 100% indicates that the pool is about to run out of free connections. This is a cause for concern. Under such circumstances, you may want to increase the Maximum connections configurations of the pool, so that the pool can accommodate more connections.

Measurement	Description	Measurement Unit	Interpretation
Timedout connection requests	Indicates the number of failed connection requests in the last measurement period.	Number	

3.4.17 SAP WAS Component Status Test

To quickly identify applications and critical SAP services that stopped suddenly or are in a Critical state, administrators need to closely track the health and operational state of every application and service on SAP WAS. For this purpose, administrators can use the **SAP WAS Component Status** test. This test automatically discovers the applications and services on a target SAP WAS, and reports the current health and operational state of each application and service. This way, the test promptly alerts administrators if any application/service is not running or switches to a Critical state suddenly, and thus enables administrators to swiftly initiate corrective measures.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the ICM and each of the server processes in a target Java instance.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
WSDL Port	This test uses the SAPControl web service to pull metrics on application and service status. To enable the test to communicate with the web service, you need to configure the test with the port number of the web service. Therefore, specify the port number of the SAPControl web service against WSDL Port. To determine the exact port number of the SAPControl web service, you can look up the **etc/services** file on the SAP WAS being monitored. If the port number is not declared in the **etc/services** file, you can specify the default port number of the web service against WSDL Port. If the web service is not SSL-enabled, then the default port number of the web service will be: 5<NR>13. Similarly, if the web service is SSL-enabled, then the default port number of

Parameter	Description
	the web service will be: 5<NR>14. <NR> in the port number refers to the system number of the SAP server being monitored. The system number is an indicator of the TCP/IP port at which the SAP server listens. For example, for a server that listens at port 3200, the system number will be ‘00’. Similarly, if the SAP server port is 3201, the system number will have to be specified as ‘01’. Accordingly, the default port number of an SSL-enabled SAPControl web service will be 50014 , if the system number is 00, or 50114, if the system number is 01.
OS Username and OS Password	To enable the test to access the SAPControl web service and determine component state, you need to configure the OS Username and OS Password parameters of the test with the credentials (i.e., name and password) of an OS user, who fulfills one of the following conditions: • The name of the OS user should be declared using the service\admin_users profile parameter; (OR) • The name of the user group to which the OS user belongs should be declared using the service\admin_groups profile parameter; (OR) • The OS user should have the permission to execute the **sapstartsrv** executable
Confirm Password	Confirm the OS Password by retyping it here.
Startup Mode	Startup modes determine how your application is started on the J2EE Engine. By default, this test monitors only those applications that automatically start whenever the J2EE engine is restarted – i.e., those applications for which the Startup mode is set to **Always**. This is why, this flag is set to **Always** by default. Alternatively, you can set this flag to **Manual**. In this case, the test will monitor only those applications for which the Startup Mode is set to **Manual** – i.e., all applications that will require a manual start whenever the J2EE engine is restarted. You can also set this flag to **Lazy**, so that the test monitors only those applications for which the Startup Mode is set to **Lazy**. These are applications that will be stopped upon restart of the J2EE engine, and which will start only when another application requests a resource from it.
Include Apps/Services	Provide a comma-separated list of applications and/or services that you want the test to monitor. Note that you can provide the whole/part of an application/service name here. When providing the complete application/service name, make sure that the exact application/service name is provided here. If a part of an application/service name is provided, then all applications and services that embed that string will be monitored.
Exclude Apps/Services	Provide a comma-separated list of applications and/or services that you want the test to exclude from monitoring. Note that you can provide the whole/part of an application/service name here. When providing the complete application/service name,

Parameter	Description
	make sure that the exact application/service name is provided here. If a part of an application/service name is provided, then all applications and services that embed that string will be excluded from monitoring.
SSL	Set this flag to **Yes**, if the SAPControl web service is SSL-enabled. Set this flag to **No**, if the SAPControl web service is not SSL-enabled.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Health</td><td>Indicates the current health of this application/service.</td><td></td><td>The values that this measure can report and their corresponding numeric values are discussed in the table below:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Normal</td><td>1</td></tr>
<tr><td>Critical</td><td>2</td></tr>
<tr><td>Error</td><td>3</td></tr>
<tr><td>Offline</td><td>4</td></tr>
</table>
<b>Note:</b>
By default, the test reports the <b>Measure Value</b>s in the table above to</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>indicate application/service state. In the graph of this measure however, the same is represented using the numeric equivalents only.
You can use the detailed diagnosis of the measure to determine the startup mode of each application/service.</td></tr>
<tr><td>Status</td><td>Indicates the current operational status of this application/service.</td><td></td><td>The values that this measure can report and their corresponding numeric values are discussed in the table below:<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Running</td><td>1</td></tr><tr><td>Stopped</td><td>2</td></tr></table><b>Note:</b>
By default, the test reports the <b>Measure Value</b>s in the table above to indicate operational status. In the graph of this measure however, the same is represented using the numeric equivalents only.</td></tr>
<tr><td>Expected status</td><td>Indicates the expected status of this application/service.</td><td></td><td>The values that this measure can report and their corresponding numeric values are discussed in the table below:<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Running</td><td>1</td></tr><tr><td>Stopped</td><td>2</td></tr></table><b>Note:</b>
By default, the test reports the <b>Measure Value</b>s in the table above to</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			indicate expected status. In the graph of this measure however, the same is represented using the numeric equivalents only.

3.4.18 SAP WAS ICM Threads Test

The Internet Communication Manager ensures that communication between the SAP System (SAP Web Application Server) and the outside world via HTTP, HTTPS and SMTP protocols works properly. In its role as a server, the ICM can process requests from the Internet that arrive as URLs with the server/port combination that the ICM can listen to. The ICM then calls the relevant local handler for the URL in question.

The ICM process uses threads to parallel process the load. The following illustration shows a detailed overview of the ICM.

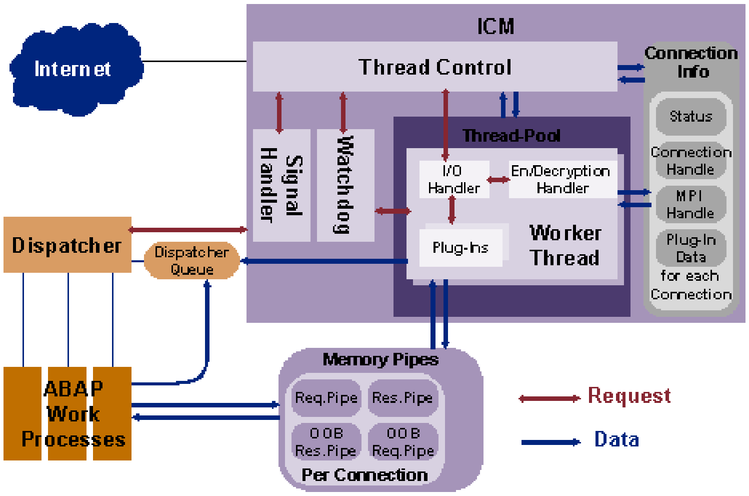


Figure 3.5: Detailed overview of the ICM

Besides the pool of worker threads, which process incoming requests, the following ICM components are also implemented as threads:

- **Thread Control**: This thread accepts the incoming TCP/IP requests and creates (or wakes) a worker thread from the thread pool to process the request. From this point on, thread control initializes the connection info data.

- **Worker Threads**: These threads handle connection requests and responses. A worker thread contains an I/O handler for the network input and output, diverse plugins for the various supported protocols (HTTP, SMTP,…), which are required to be able to decide when the sent packet is finished (depends on the protocol).
- **Watchdog**: Usually, a worker thread waits for the response, regardless of whether the worker thread is a server or a client.If a timeout occurs, the watchdog takes on the task of waiting for the response. This makes the worker thread available for other requests. When the watchdog receives the response, it informs the thread control components, which then call a worker thread.
- **Signal Handler**: This thread processes signals sent from the operating system or from another process (for example, the dispatcher).
- **Connection Info**: This table contains information about the state of the connection, the memory pipes, and the plug-in data for every existing network connection.
- **Memory Pipes**: Memory pipes are memory-based communication objects that handle data transfer between the ICM and the work processes. For each connection there are four pipes: for each request and response one data pipe and an OOB (Out Of Band) pipe. The OOB pipe is used for control data.
- **Internet Server Cache**: The ICM contains another cache to enable repeated requests to be quickly responded to. This cache is not shown in the graphic

If any of these threads is unavailable or is slow, then HTTP/S requests received by the SAP WAS cannot be serviced quickly. Such processing delays can severely hamper the experience of Internet users with the SAP. This is why, it is imperative that SAP administrators proactively detect such slowdowns, accurately identify when the slowdown occurred, and pinpoint the precise thread responsible for it. This is where the **SAP ICM Threads** test helps! This test auto-discovers the threads that are at work in the ICM and reports the current state of each thread, what type of work each thread is performing presently, and how quickly that thread is doing that work. In the process, the test sheds light on processing bottlenecks, pinpoints where the bottleneck lies, and leads administrators to the exact thread causing it. This knowledge helps administrators rapidly troubleshoot slowdowns and avert serious outages.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for every ICM thread.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
WSDL Port	This test uses the SAPControl web service to pull metrics on application and service status. To enable the test to communicate with the web service, you need to configure the test with the port number of the web service. Therefore, specify the port number of the SAPControl web service against WSDL Port. To determine the exact port number of the SAPControl web service, you can look up the **etc/services** file on the SAP WAS being monitored. If the port number is not declared in the **etc/services** file, you can specify the default port number of the web service against WSDL Port. If the web service is not SSL-enabled, then the default port number of the web service will be: 5<NR>13. Similarly, if the web service is SSL-enabled, then the default port number of the web service will be: 5<NR>14. <NR> in the port number refers to the system number of the SAP server being monitored. The system number is an indicator of the TCP/IP port at which the SAP server listens. For example, for a server that listens at port 3200, the system number will be '00'. Similarly, if the SAP server port is 3201, the system number will have to be specified as '01'. Accordingly, the default port number of an SSL-enabled SAPControl web service will be 50014 , if the system number is 00, or 50114, if the system number is 01.
OS Username and OS Password	To enable the test to access the SAPControl web service and determine component state, you need to configure the OS Username and OS Password parameters of the test with the credentials (i.e., name and password) of an OS user, who fulfills one of the following conditions: • The name of the OS user should be declared using the service\admin_users profile parameter; (OR) • The name of the user group to which the OS user belongs should be declared using the service\admin_groups profile parameter; (OR) • The OS user should have the permission to execute the **sapstartsrv** executable
Confirm Password	Confirm the OS Password by retyping it here.
SSL	Set this flag to **Yes**, if the SAPControl web service is SSL-enabled. Set this flag to **No**, if the SAPControl web service is not SSL-enabled.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Status</td><td>Indicates the current status of this thread.</td><td></td><td>The values that this measure can report, what they mean, and their corresponding numeric values are discussed in the table below:
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td>Running</td><td>The thread is processing a request</td><td>1</td></tr>
<tr><td>Idle</td><td>The thread is waiting for a request</td><td>2</td></tr>
<tr><td>Not Available</td><td>There are no requests</td><td>3</td></tr>
</table>
<b>Note:</b>
By default, the test reports the <b>Measure Value</b>s in the table above to indicate thread status. In the graph of this measure however, the same is represented using the numeric equivalents only.</td></tr>
<tr><td>Current operation</td><td>Indicates the operation that this thread is currently performing.</td><td></td><td>The values that this measure can report, what they mean, and their corresponding numeric values are discussed in the table below:
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td>NOP</td><td>No action</td><td>0</td></tr>
<tr><td>Administration</td><td>Action that is triggered from transaction SMICM or program icmon - for example, display list. For</td><td>1</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td></td><td>this reason, at least one thread always displays Administration, if you call SMICM.</td><td></td></tr>
<tr><td>Read request</td><td>Reads the request (server)</td><td>2</td></tr>
<tr><td>Read response</td><td>Reads the response (client)</td><td>3</td></tr>
<tr><td>Write request</td><td>Writes the request (client)</td><td>4</td></tr>
<tr><td>Write response</td><td>Writes the response (server)</td><td>5</td></tr>
<tr><td>Open connection</td><td>Sets up connection to a server</td><td>6</td></tr>
<tr><td>Close connection</td><td>Closes connection to a partner</td><td>7</td></tr>
<tr><td>Accept connection</td><td>Accepts the connection from a client</td><td>8</td></tr>
<tr><td>Time-dependent action</td><td>Executes time-dependent (periodically scheduled) events</td><td>9</td></tr>
<tr><td>Wait for data</td><td>Waits for data from the network or from the application server (SAP Web dispatcher)</td><td>10</td></tr>
<tr><td>Wait for response (SERV)</td><td>Waits for a response from</td><td>11</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Description</th><th>Numeric Value</th></tr>
<tr><td></td><td>the application server (SAP Web AS is a server with active connections)</td><td></td></tr>
<tr><td>Wait for response (CLNT)</td><td>Waits for a request from the application server (SAP Web AS is a server with active connections)</td><td>12</td></tr>
<tr><td>Shutdown</td><td>Ends ICM</td><td>13</td></tr>
</table>
<b>Note:</b>
By default, the test reports the <b>Measure Value</b>s in the table above to indicate operational status. In the graph of this measure however, the same is represented using the numeric equivalents only.
</td></tr>
<tr><td>Processing rate</td><td>Indicates the rate at which this thread is processing requests.</td><td>Requests/Sec</td><td>A consistent drop in the value of this measure could indicate a processing bottleneck.
Compare the value of this measure across threads to know which thread is the slowest.</td></tr>
</table>

3.4.19 Access Points List Test

This test reports the total number of access points available for each protocol and the number of access points that were inactive.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for every *Process:protocol* on the SAP WAS monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
WSDL Port	This test uses the SAPControl web service to pull metrics on application and service status. To enable the test to communicate with the web service, you need to configure the test with the port number of the web service. Therefore, specify the port number of the SAPControl web service against WSDL Port. To determine the exact port number of the SAPControl web service, you can look up the **etc/services** file on the SAP WAS being monitored. If the port number is not declared in the **etc/services** file, you can specify the default port number of the web service against WSDL Port. If the web service is not SSL-enabled, then the default port number of the web service will be: 5<NR>13. Similarly, if the web service is SSL-enabled, then the default port number of the web service will be: 5<NR>14. <NR> in the port number refers to the system number of the SAP server being monitored. The system number is an indicator of the TCP/IP port at which the SAP server listens. For example, for a server that listens at port 3200, the system number will be ‘00’. Similarly, if the SAP server port is 3201, the system number will have to be specified as ‘01’. Accordingly, the default port number of an SSL-enabled SAPControl web service will be 50014 , if the system number is 00, or 50114, if the system number is 01.
OS Username and OS Password	To enable the test to access the SAPControl web service and determine component state, you need to configure the OS Username and OS Password parameters of the test with the credentials (i.e., name and password) of an OS user, who fulfills one of the following conditions: • The name of the OS user should be declared using the service\admin_users profile parameter; (OR) • The name of the user group to which the OS user belongs should be declared using the service\admin_groups profile parameter; (OR) • The OS user should have the permission to execute the **sapstartsrv** executable
Confirm Password	Confirm the OS Password by retyping it here.
SSL	Set this flag to **Yes**, if the SAPControl web service is SSL-enabled. Set this flag to **No**, if the SAPControl web service is not SSL-enabled.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an

Parameter	Description
	optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total access points	Indicates the total number of access points configured for this protocol.	Number	
Inactive access points	Indicates the number of active points of this protocol that were inactive.	Number	

3.4.20 SAP WAS Aliases Test

An application alias is an alternative name for a J2EE Web application running on the J2EE Engine. Each Web application has its own root as defined by the Java Servlet Specification version 2.3. It is also referred to as context path. The context path of a Web application is mapped to a URL pattern. Server users utilize this URL pattern to access the resources of a particular Web application. The application alias actually represents the URL pattern that is mapped to a certain application's context path. If too many aliases are inactive or not available, then the users may experience difficulty in accessing the web applications corresponding to the aliases. Therefore, it is necessary to monitor the application aliases round the clock.

This test auto-discovers the patterns of the aliases and for each pattern, reports the total number of aliases available and the total number of aliases that were inactive. In addition, this test reports the rate at which aliases of each pattern are retrieved upon user request.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for every Alias Pattern on the SAP WAS monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
WSDL Port	This test uses the SAPControl web service to pull metrics on application and service status. To enable the test to communicate with the web service, you need to configure the test with the port number of the web service. Therefore, specify the port number of the SAPControl web service against WSDL Port. To determine the exact port number of the SAPControl web service, you can look up the **etc/services** file on the SAP WAS being monitored. If the port number is not declared in the **etc/services** file, you can specify the default port number of the web service against WSDL Port. If the web service is not SSL-enabled, then the default port number of the web service will be: 5<NR>13. Similarly, if the web service is SSL-enabled, then the default port number of the web service will be: 5<NR>14. <NR> in the port number refers to the system number of the SAP server being monitored. The system number is an indicator of the TCP/IP port at which the SAP server listens. For example, for a server that listens at port 3200, the system number will be ‘00’. Similarly, if the SAP server port is 3201, the system number will have to be specified as ‘01’. Accordingly, the default port number of an SSL-enabled SAPControl web service will be 50014 , if the system number is 00, or 50114, if the system number is 01.
OS Username and OS Password	To enable the test to access the SAPControl web service and determine component state, you need to configure the OS Username and OS Password parameters of the test with the credentials (i.e., name and password) of an OS user, who fulfills one of the following conditions: • The name of the OS user should be declared using the service\admin_users profile parameter; (OR) • The name of the user group to which the OS user belongs should be declared using the service\admin_groups profile parameter; (OR) • The OS user should have the permission to execute the **sapstartsrv** executable

Parameter	Description
Confirm Password	Confirm the OS Password by retyping it here.
Include Alias	Specify a comma-separated list of alias patterns that need to be included in the scope of monitoring.
Exclude Alias	Specify a comma separated list of alias patterns that need to be excluded from the scope of monitoring.
SSL	Set this flag to **Yes**, if the SAPControl web service is SSL-enabled. Set this flag to **No**, if the SAPControl web service is not SSL-enabled.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total aliases	Indicates the total number of aliases available in this pattern.	Number	
Inactive aliases	Indicates the number of aliases in this pattern that were inactive.	Number	The detailed diagnosis of this measure lists the Application name and the status of the Ignore cookie.
Average request rate	Indicates the rate at which aliases of this pattern were retrieved upon a user request.	Requests/sec	A high value is desired for this measure.

3.4.21 SAP WAS Caches Test

This test auto-discovers the caches available in the SAP Web Application server and for each cache, reports the current health and size. In addition, this test reports how well the cache was servicing the user requests and how many times the instances and cluster were invalidated.

Target of the test : A SAP Netweaver Application Server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each *user:cache* available on the target SAP WAS being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	Specify the server for which the test is to be configured.
Port	The port number at which the specified server listens.
WSDL Port	This test uses the SAPControl web service to pull metrics on application and service status. To enable the test to communicate with the web service, you need to configure the test with the port number of the web service. Therefore, specify the port number of the SAPControl web service against WSDL Port. To determine the exact port number of the SAPControl web service, you can look up the **etc/services** file on the SAP WAS being monitored. If the port number is not declared in the **etc/services** file, you can specify the default port number of the web service against WSDL Port. If the web service is not SSL-enabled, then the default port number of the web service will be: 5<NR>13. Similarly, if the web service is SSL-enabled, then the default port number of the web service will be: 5<NR>14. <NR> in the port number refers to the system number of the SAP server being monitored. The system number is an indicator of the TCP/IP port at which the SAP server listens. For example, for a server that listens at port 3200, the system number will be '00'. Similarly, if the SAP server port is 3201, the system number will have to be specified as '01'. Accordingly, the default port number of an SSL-enabled SAPControl web service will be 50014 , if the system number is 00, or 50114, if the system number is 01.
OS Username and OS Password	To enable the test to access the SAPControl web service and determine component state, you need to configure the OS Username and OS Password parameters of the test with the credentials (i.e., name and password) of an OS user, who fulfills one of the following conditions:

Parameter	Description
	- The name of the OS user should be declared using the service\admin_users profile parameter; (OR) - The name of the user group to which the OS user belongs should be declared using the service\admin_groups profile parameter; (OR) - The OS user should have the permission to execute the **sapstartsrv** executable
Confirm Password	Confirm the OS Password by retyping it here.
Include Alias	Specify a comma-separated list of alias patterns that need to be included in the scope of monitoring.
Exclude Alias	Specify a comma separated list of alias patterns that need to be excluded from the scope of monitoring.
SSL	Set this flag to **Yes**, if the SAPControl web service is SSL-enabled. Set this flag to **No**, if the SAPControl web service is not SSL-enabled.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: - The eG manager license should allow the detailed diagnosis capability - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Health	Indicates the current health of this cache.	Number	The values that this measure can report and their corresponding numeric values are discussed in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Normal</td><td>1</td></tr><tr><td>Critical</td><td>2</td></tr><tr><td>Error</td><td>3</td></tr><tr><td>Offline</td><td>4</td></tr></table>
<b>Note:</b>
By default, the test reports the <b>Measure Value</b>s in the table above to indicate the current health of this cache. In the graph of this measure however, the same is represented using the numeric equivalents only.</td></tr>
<tr><td>Size</td><td>Indicates the current size of this cache.</td><td>KB</td><td>The size of the entries is not known for all caches. This is because not every program specifies this entry explicitly when an entry is generated. However, in the case of a Web repository manager, the size is given and can therefore be displayed.</td></tr>
<tr><td>Hit ratio</td><td>Indicates the rate of cache hits to cache gets in this cache.</td><td>Percent</td><td>A high value is desired for this measure. A sudden/gradual decrease in the value of this measure is an indication of the cache being ineffective.</td></tr>
<tr><td>Instance invalidations</td><td>Indicates the number of instance invalidations in this cache.</td><td>Number</td><td>Instance invalidations are mainly used to refresh the cache of a particular user or a particular iFrame.
If an object in the cache has to be invalidated, all servers in the system must be notified by RFC that this object is no longer up to date and should be invalidated. The next time the object is requested, the up-to-date version of it must be re-loaded into</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			each cache. A high value for this measure is a cause of concern as this may affect system performance.
Cluster invalidations	Indicates the number of cluster invalidations in this cache.	Number	Cluster invalidations are mainly used to refresh the cache of all the users on the cluster i.e., the SAP WAS server. Depending on the number of users logged on to the system, invalidating the caches across the entire cluster can affect system performance.

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.