Oracle® Enterprise Manager

Oracle Application Diagnostics for Java (Oracle AD4J)
Installation and Administration Guide

10*g* Release 5 (10.2.0.5)

E11085-03

August 2009

Oracle Enterprise Manager Oracle Application Diagnostics for Java (Oracle AD4J) Installation and Administration Guide, 10*g* Release 5 (10.2.0.5)

E11085-03

Contents

Preface vii

Intended Audience vii
Documentation Accessibility vii

1 Introduction to Application Diagnostics for Java

1.1 Overview 1-1
1.1.1 Low Overhead Monitoring with Deep Diagnostics 1-2
1.1.2 Easy Installation and Deployment 1-2
1.1.3 Real-Time JVM Visibility 1-2
1.1.4 Cross-Tier Transaction Tracing 1-2
1.1.5 Powerful Heap Analysis Tools 1-2
1.1.6 JVM Pooling 1-2
1.1.7 Real-time and Historical Diagnosis 1-3
1.1.8 JVM Tracing 1-3
1.2 System Architecture 1-3
1.3 Software and Hardware Requirements 1-4

2 Installing the Oracle AD4J Console

2.1 Installing the AD4J Console on Unix 2-1
2.1.1 Prerequisites 2-1
2.1.2 Installation Procedure 2-2
2.1.2.1 Collecting Database Repository Details 2-2
2.1.2.2 Extracting Files 2-2
2.1.2.3 Running the Configuration Script 2-3
2.1.2.4 Understanding Console Parameters 2-5
2.1.2.5 Starting Up and Initializing the Console 2-5
2.1.2.6 Specifying the Database Connection Details 2-5
2.1.2.7 Creating the Database Tables 2-6
2.1.2.8 Viewing JVM Console Diagnostics 2-7
2.1.3 Uninstalling the AD4J Console on Unix 2-8
2.2 Installing the AD4J Console on Windows 2-8

2.2.1 Prerequisites 2-8
2.2.2 Installation Procedure 2-9
2.2.2.1 Running the jamserv-win.msi File 2-9
2.2.2.2 Selecting the Installation Location 2-9
2.2.2.3 Setting the Console Parameters 2-9
2.2.2.4 Specifying the Port Information 2-9
2.2.3 Uninstalling the AD4J Console on Windows 2-11
2.3 Post-Installation Tasks 2-11
2.3.1 Accessing the AD4J Console for the First Time 2-12
2.3.2 Setting Up the Console and Diagnostics 2-12
2.3.3 Deploying the Agent on Monitored JVMs and Databases 2-12

3 Installing the AD4J Agent

3.1 Before you Start 3-1
3.1.1 Customizing the Oracle AD4J Agent 3-1
3.1.1.1 Web.XML Contents 3-2
3.2 Installing the AD4J Agent on Oracle Application Server 10g 3-3
3.2.1 Deploying the Agent WAR 3-3
3.3 Installing the AD4J Agent on Oracle WebLogic 3-4
3.3.1 Deploying the Agent WAR 3-4
3.4 Installing the AD4J Agent on Apache Tomcat 3-7
3.4.1 Deploying the Agent WAR 3-7
3.5 Installing the Oracle AD4J Agent on Apache JServ 3-8
3.5.1 Setting Up the CLASSPATH - Automatic Mode 3-8
3.5.2 Setting Up the CLASSPATH - Manual Mode 3-8
3.5.3 Loading the Servlet at Startup Time 3-8
3.6 Installing the Oracle AD4J Agent on IBM WebSphere 3-9
3.6.1 Deploying the Agent WAR 3-9
3.7 Installing the Oracle AD4J Agent on Standalone Java Applications 3-10
3.7.1 Original Java Call 3-11
3.7.2 Modified Java Call with Oracle AD4J Agent 3-11
3.7.3 Sample Configuration 3-11
3.7.4 Adding the Oracle AD4J Agent and Libraries, Classes to Classpath 3-12
3.7.5 Target Parameters 3-12
3.8 Installing the Database Agent 3-12
3.8.1 Prerequisites 3-12
3.8.2 Register the Database in the Setup Tab 3-12
3.8.3 Download the DB Agent for Your Platform from the Console 3-13
3.8.4 Download the Agent for Your Database Platform 3-14

4 Setting Up and Using the AD4J Console

4.1 Oracle AD4J Console Setup 4-2
4.2 Assumptions 4-2
4.3 JVM Console Diagnostics 4-2
4.4 Console Login Screen 4-2
4.5 DB Connection Information 4-3
4.6 Repository DB Page 4-4

4.7 Console Setup Page 4-5
4.8 JVM Pool Info 4-7
4.9 Updating a JVM Pool 4-7
4.10 Update Pool Thresholds 4-8
4.11 Database Information 4-9
4.12 Register New Database/Update Screen 4-10
4.13 JVM Information Page 4-11
4.14 Monitoring 4-11
4.15 Manage Trace Files 4-12
4.16 Manage Users 4-13
4.17 Download Agent 4-13
A.1 Frequently Asked Questions A-1
A.1.1 What is the Meaning of Cost? A-1
A.1.2 In My Console, Why Do I See *No Data Found* in the Java Tab? A-1
A.1.3 How Do I View a Specific Period in the NOC View? A-2
A.1.4 Why are the User, Request, and IP Address fields Empty in the Thread View? A-2
A.1.5 How Much Repository Space is Required for Monitoring and for Heap Dumps A-2
A.1.6 How Do I Install the AD4J Database Agent on Windows? A-2
A.1.7 Can I Still Use AD4J Without a Database Agent? A-2
A.1.8 How Can I Check if my JVM is Supported by AD4J? A-3
A.1.9 Where does the Agent Store Its Log Files? A-3
A.1.10 How do I Prevent Agent from Terminating Program before Thread Execution is Completed? A-3

Preface

This guide provides information about installing the console and the Oracle Application Diagnostics for Java (Oracle AD4J) Agent.

This preface contains the following topics:

- Intended Audience
- Documentation Accessibility

Intended Audience

This guide is meant for system administrators who use Oracle AD4J.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711.

1 Introduction to Application Diagnostics for Java

This chapter provides an introduction to Oracle Enterprise Manager 10g Application Diagnostics for Java. It covers the following:

- Overview
- System Architecture
- Software and Hardware Requirements

1.1 Overview

Mission critical Java applications often suffer from availability and performance problems. Developers and IT administrators spend a lot of time diagnosing the root cause of these problems. Many times, the problems occurring in production environments either cannot be reproduced or may take too long to reproduce in other environments. This can cause severe impact on the business.

Oracle Enterprise Manager 10g Application Diagnostics for Java (Oracle AD4J) is a lightweight Java application monitoring and diagnostics tool that enables administrators to diagnose performance problems in production. By eliminating the need to reproduce problems, it reduces the time required to resolve these problems. This improves application availability and performance. Using Oracle AD4J, administrators will be able identify the root cause of performance problems in the production environment without having to reproduce them in the test or development environment. It does not require complex instrumentation or restarting of the application to get in-depth application details. Application administrators will be able to identify Java problems or Database issues that are causing application downtime without any detailed application knowledge. The key features of Oracle AD4J are:

- Low Overhead Monitoring with Deep Diagnostics
- Easy Installation and Deployment
- Real-Time JVM Visibility
- Cross-Tier Transaction Tracing
- Powerful Heap Analysis Tools
- JVM Pooling
- Real-time and Historical Diagnosis
- JVM Tracing

1.1.1 Low Overhead Monitoring with Deep Diagnostics

Oracle AD4J provides in-depth monitoring of Java applications with very low overheads. Hence it does not slow down your application. It helps you to identify the slowest requests, slowest methods, requests waiting on I/O, requests using a lot of CPU cycles, and requests waiting on database calls. It also identifies the end-user requests that have been impacted by resource bottlenecks. Application resources that are causing the performance bottleneck are also visible.

1.1.2 Easy Installation and Deployment

Oracle AD4J is easy to install, deploy, and use. To install the product, you do not need to modify the source code / byte code of your application, or restart any application server. The AD4J Agent can be deployed easily without modifying the application or changing any configuration files. Once deployed, the Monitoring feature can be turned on so that it is always monitoring your production JVMs and databases. This saves time in reproducing the problems.

1.1.3 Real-Time JVM Visibility

Oracle AD4J provides immediate visibility into the Java stack. You can monitor thread states and Java method/line numbers in real time and you can proactively identify issues rather than diagnosing issues like application crashes, memory leaks, and application hangs after they occur.

If a particular request is hanging or if the entire application is slow, administrators can perform a real-time transaction trace to view current Java application activity. You can see the offending threads and their execution call stacks. You can also analyze various bottleneck resources such as how much time a thread spent in waiting for a database lock. Complex problems such as activity in one thread (or request) affecting the activity in the other thread or rest of the JVM can be found very quickly.

1.1.4 Cross-Tier Transaction Tracing

Oracle AD4J facilitates tracing of Java requests to the associated database sessions and vice-versa enabling rapid resolution of problems that span different tiers. It highlights the slowest SQL queries and helps administrators to tune SQL and the database to improve the performance application. This facilitates smooth communication between the database administrators and application administrators by isolating the problems to the database or the Java tier.

1.1.5 Powerful Heap Analysis Tools

Memory leaks lead to application slowdowns and eventually cause applications to crash. Oracle AD4J alerts administrators on abnormalities in Java memory consumption. Administrators can use Oracle's Java diagnostics console and take heap dumps in production applications without disturbing the application. They can take multiple heap dumps over a period of time, analyze the differences between the heap dumps and identify the object causing the memory leak. Heap analysis can be performed even across different application versions. Differential Heap Analysis with multiple heap dumps makes it easy to identify memory leaks.

1.1.6 JVM Pooling

Oracle AD4J allows administrators to group sets of JVMs together into JVM pools. This provides the console user with a single view across all related JVMs. Hence all

JVM's that make up a single application or a single cluster may be grouped together in an application. This allows administrators to visualize problems naturally and intuitively.

1.1.7 Real-time and Historical Diagnosis

With Oracle AD4J, you can perform real-time and historical diagnostics on your Java applications. This provides you with detailed insight on the root causes of production problems without having to reproduce the same problem in a Test or QA environment. You can play back transactions interactively from the browser and view the time spent in the network and the server. You can also view the response time breakdown by Servlet, JSP, EJB, JDBC, and SQL layers.

Apart from the real-time data, you can also analyze historical data to diagnose problems that occurred in the past. You can view historical data that shows the time taken by end-user requests and the breakdown by Servlet, JSP, EJB, JDBC, and SQL layers.

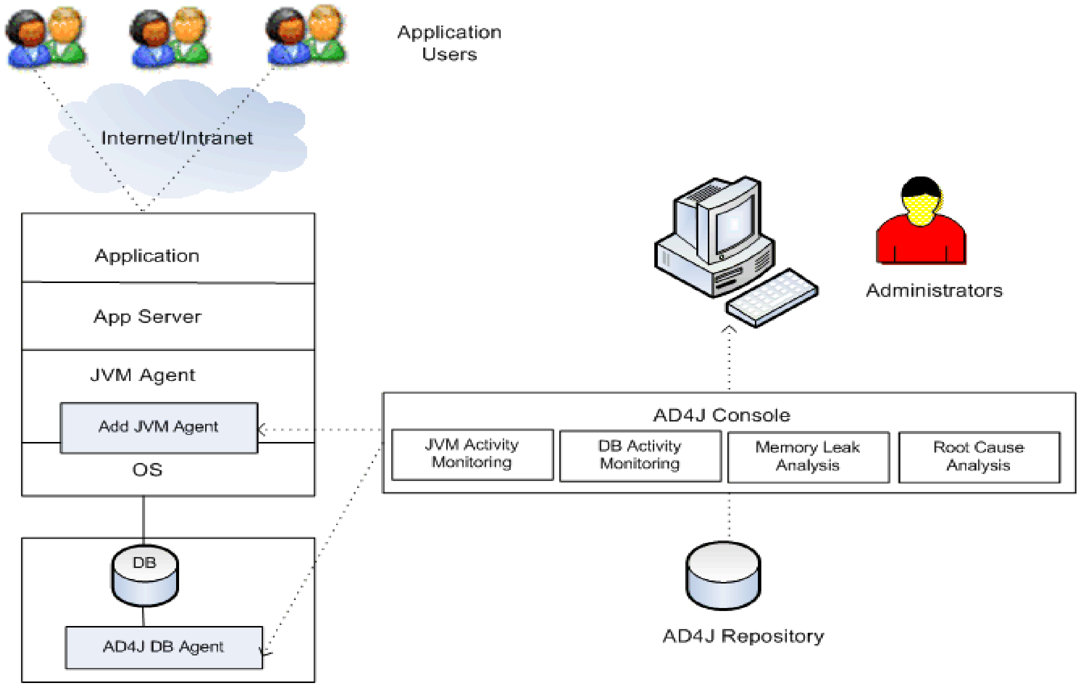
1.1.8 JVM Tracing

Sometimes the monitoring interval (normally 1 second) that is in use is too coarse grained. The Java thread of interest may be too short lived or the amount of monitoring data collected may be insufficient. In such cases, you can run a JVM Trace to get fine-grained details of the JVM activity. This feature of AD4J allows you to monitor your Java application at a very high frequency (default of once every 50ms) for a short period of time. This allows you to identify interdependency of threads, bottleneck resources (DB, I/O, CPU, Locks, Network) & top methods.

1.2 System Architecture

The following figure shows the system architecture of Oracle’s Java Diagnostics application.

Figure 1–1 AD4J Architecture



In this diagram, two AD4J Agents are being used to monitor a standard three tier Java based application. The JVM agent is used to monitor the mid-tier and the database agent is used to monitor the back end. The AD4J Console is a web-based console and does not require any client utility to be installed. In response to specific user actions on the console, the console makes various requests for monitoring data to the AD4J JVM Agent. The Agent takes snapshots of the state of the running JVM and sends a response back to the Console which is stored in the repository. If additional database correlation information is required, the Console sends a request to the AD4J DB Agent which in turn sends a response back to the Console. All the computation and post processing activity takes place only in the Console and its repository thereby ensuring optimal performance from the application itself.

1.3 Software and Hardware Requirements

- **Console JVM Requirements**: Any 32-bit JDK.
- **AD4J Console**: Windows (2000 or later), Windows XE, Linux (RHEL 3 or later), Solaris SPARC (8 or later), 1 CPU, 2GHz (Intel equivalent), 2GB RAM, 10GB Hard Disk.
- **Supported Console Repository Versions**: Oracle 9i, 10g, and 11g either local or remotely installed. On Windows only, you can use the PostgreSQL starter database that is shipped with the product. This database must be changed to an Oracle database before production deployment.

  To run the AD4J Console with a repository on Oracle XE, you need to increase the Max DB Sessions by entering the following commands in SQL*Plus:

  ```
  SQL> show parameters sessions
  SQL> alter system set sessions=500 scope=spfile;
  SQL> show parameters processes
  SQL> alter system set processes=500 scope=spfile;
  ```

Note: When you load heap dumps, ensure that there is at least 5 times the amount of space as the file being loaded. E.g.: If you are loading a 1GB heap dump, you must ensure that there is 5GB available in the database.

2 Installing the Oracle AD4J Console

This chapter provides procedures for installing the Oracle AD4J Console. It covers the following:

- Installing the AD4J Console on Unix
- Installing the AD4J Console on Windows
- Post-Installation Tasks

2.1 Installing the AD4J Console on Unix

This section covers the procedure to install the AD4J Console on the supported Unix platforms (i.e. Linux and Solaris). It describes the following:

- Prerequisites
- Installation Procedure
- Uninstalling the AD4J Console on Unix

2.1.1 Prerequisites

Before you begin the installation process, do the following:

- Determine the host system on which you want to deploy the console. Oracle recommends the use of a dedicated host for the console.
- Install JDK 1.4.2 or later on that host. It must be a 32 bit JVM.

  Note: You can a 32 bit JVM on a 64 bit operating system.
- Have access to Oracle 9i (or later) database for use as a repository. This database may be local or remote.
- Create a database user for the Oracle AD4J repository. Assign appropriate default and temporary tablespace and permissions to the user so that database objects can be created in these tablespaces.
- Assign an appropriate table space for user (temp and default).
- Have adequate space for the heap analysis features. You need about 5 times the heap size space for every heap snapshot.
- Install the Oracle client on the console machine. This is required only for heap analysis features.
- Depending on the platform, download `jamserv-lnx.zip` or `jamserv-sun.zip.`

- Determine the http port you will use to access the console (the default is 3500).
- Determine the TCP/IP port that agents will use to establish a connection with the console (the default is 3600).
- Collect the requisite information for the DB Repository. See Collecting Database Repository Details for details.
- Decide the location at which the console is to be installed. After installation, you must use the following URL to access the console: `http://consolehost:<port_no>`.You must download the agent to be deployed on the monitored JVMs. To log in, you must use the default console setup credentials , `admin` as the user name and `welcome` as the password. After you log in, you must change the default password. You can perform operations on the console only the password has been changed.

2.1.2 Installation Procedure

The installation procedure includes the following steps:

- Collecting Database Repository Details
- Extracting Files
- Running the Configuration Script
- Understanding Console Parameters
- Starting Up and Initializing the Console
- Specifying the Database Connection Details
- Creating the Database Tables
- Viewing JVM Console Diagnostics
- Running the jamserv-win.msi File
- Selecting the Installation Location
- Setting the Console Parameters
- Specifying the Port Information

2.1.2.1 Collecting Database Repository Details

You will need to collect the following information about the database:

- DB server name (E.g. `localhost`)
- Database name (E.g. `orcl`)
- DB Listener Port (E.g. `1521`)
- Repository User Name
- Repository User Password

2.1.2.2 Extracting Files

Unzip the archive file you have downloaded. Run the following command to extract the files at any location where the AD4J console is to be installed:

```
unzip jamserv-<platform>.zip
```

where `<platform>` can be `lnx` or `sun`. This creates a *jamserv* directory with the following sub-directories:

Table 2–1 File install and configuration

Console application files	jam
Configuration files	conf
Binaries and Shell scripts	bin
Sun binaries	bin/sun
Linux binaries	bin/lnx
Log files	logs
Library files	lib
Cache for JSP pages	cache
Trace files	trace

2.1.2.3 Running the Configuration Script

Now, run the following command to execute the configuration script from the *jamserv* directory:

```
./configure.sh <JDK_Home> [Agent_Port] [HTTP_Port]
```

The following table describes the parameters used in the command mentioned above.

Table 2–2 Configuration

Parameter	Default Value	Description
JDK_Home	None	The directory where Java is installed
HTTP_Port	3500	The HTTP port to access Oracle AD4J
Agent_Port	3600	The port that Oracle AD4J agents (this is not an HTTP port) use to communicate with the console.

> **Note:** Only the first parameter `JDK_Home` is mandatory. The rest are optional. So, if you choose 3500 as the default HTTP port and 3600 as the console port, and your `JDK_Home` is in `/opt/jdk142`, then run the following command:
>
> ```
> ./configure.sh /opt/jdk142
> ```

The `configure.sh` file creates self-signed certificates to enable secure access (HTTPS) to the console on port 3443. When the certificates and the certification authority (CA) are created, it prompts you with the following questions twice, once when creating the certification authority (CA) and again when creating the certificates.

Table 2–3 Certification Authority

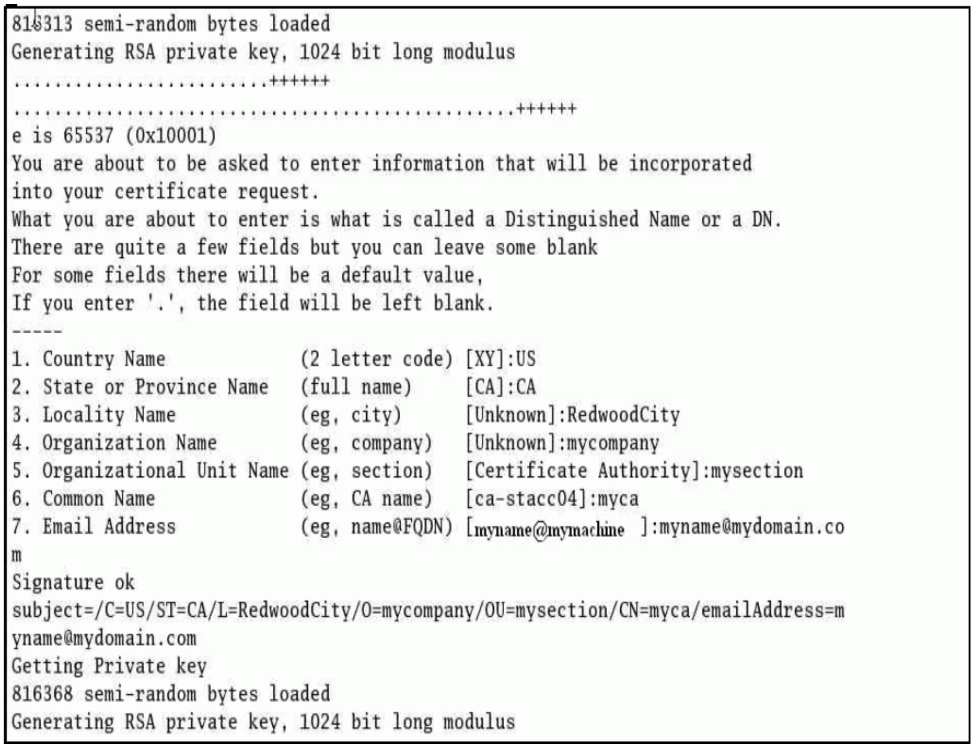
Country Name	(2 letter code) [XY]:
State or Province Name	(full name) [CA]:
Locality Name	(For example, city) [Unknown]:
Organization Name	(For example, company) [Unknown]:
Organizational Unit Name	(For example, section) [Certificate Authority]:
Common Name	(For example, CA name) [mymachine01]:

Table 2–3 (Cont.) Certification Authority

Email Address	(For example, name@FQDN) [myname@aupt01]:

The following illustrates the `configure.sh` command output. Note that `configure.sh` prompts you for these questions twice.

Figure 2–1 configure.sh Command Output



```
816313 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.........................++++++
.................................................++++++
e is 65537 (0x10001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
1. Country Name            (2 letter code) [XY]:US
2. State or Province Name  (full name)     [CA]:CA
3. Locality Name           (eg, city)      [Unknown]:RedwoodCity
4. Organization Name       (eg, company)   [Unknown]:mycompany
5. Organizational Unit Name (eg, section)  [Certificate Authority]:mysection
6. Common Name             (eg, CA name)   [ca-stacc04]:myca
7. Email Address           (eg, name@FQDN) [myname@mymachine ]:myname@mydomain.co
m
Signature ok
subject=/C=US/ST=CA/L=RedwoodCity/O=mycompany/OU=mysection/CN=myca/emailAddress=m
yname@mydomain.com
Getting Private key
816368 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
```

It is important that when creating the certificate (second set of prompts), the Common Name and prompt #5 reflect the server name you will use in the URL to access. While the install script tries to determine the host name, you must ensure that the domain is correct.

For example, for HTTPS port, if the server name is `console01`, then you must specify `https://console01:XXXX`.

For HTTPS port, you cannot specify `https://console01.mydomain.com:XXXX` or `http://124.25.113.10:XXXX` where XXXX is the HTTPS port for the Apache listener.

However, for HTTP port, you can specify all the above names.

The HTTPS port (default 3443) can, however, be changed by changing the following files:

- `jamserv/conf/httpd.conf`
- `jamserv/conf/ssl.conf`

It is strongly recommended that you replace this out-of-box certificate with a certificate issued by a well-known certificate authority.

> **Note:** To disable HTTP access and ensure secure access to the console, remove the `Listen 80` command in the `jamserv/conf/httpd.conf` file.

2.1.2.4 Understanding Console Parameters

Before you initialize the console, you must understand the console parameters specified in the `jamserv/conf/zone.properties` file. The console parameters are described in the following table:

Table 2–4 Console Parameters

Parameter	Default	Description
jamport	None	Port where console listens for connections from agents.
jamloglevel	3	How much information should console log. High value provides greater debug information but can adversely affect the performance.
jamtimeout	20	Max seconds to wait for JVM to respond.
jamlooptimeout	15	Max seconds to wait for JVM to respond for multiple JVM requests.
jampropfilepath	None	When the console starts up for the first time, it creates a property file adminDB.properties where it stores information about the repository DB. This parameter specifies the directory where the file should be created. This file contains sensitive information including username, password. It should be a secure location.
jamtracefilepath	None	Directory where to put the trace files.

2.1.2.5 Starting Up and Initializing the Console

Run the following command to start the console:

```
jamserv/bin/apachectl start
```

> **Note:** If you see a `500 HTTP Error` in your web browser after you run this command, it indicates that the `Jserv` process was not automatically started. To start it manually, enter the following command:
>
> ```
> jamserv/bin/jservctl
> ```

To view details about the error messages, access the `jamserv/logs/error.log` file. The file provides details about the following:

- Class path used by the Oracle AD4J Console
- Parameters (default and non-default) in use
- Build version of the console
- Console is ready for connections from agents

2.1.2.6 Specifying the Database Connection Details

The adminDB.properties file stores connection details about the repository database. The file resides in the location specified by the start up parameter - *jampropfilepath*.

To modify the file, do the following:

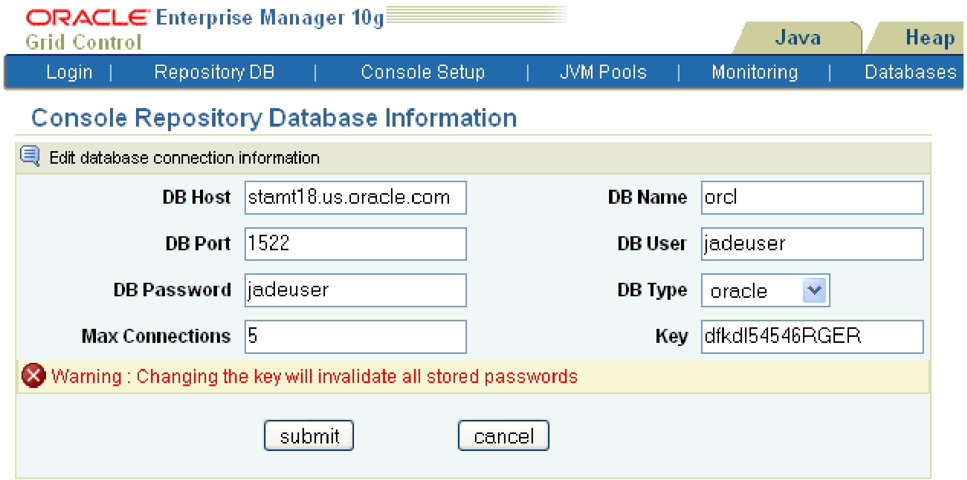
1. Click **Setup**.

   The console displays the JAM Console Diagnostics page.

2. Click the adminDB.properties link.

   The console displays the Console Repository Database Information page (Figure 2–2).

Figure 2–2 Console Repository Database Information


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases
Console Repository Database Information
Edit database connection information
submit
cancel

DB Host	stamt18.us.oracle.com	DB Name	orcl
DB Port	1522	DB User	jadeuser
DB Password	jadeuser	DB Type	oracle
Max Connections	5	Key	dfkdl54546RGER

Warning : Changing the key will invalidate all stored passwords

3. Modify the settings.
4. Restart the console to see the changed settings in effect.

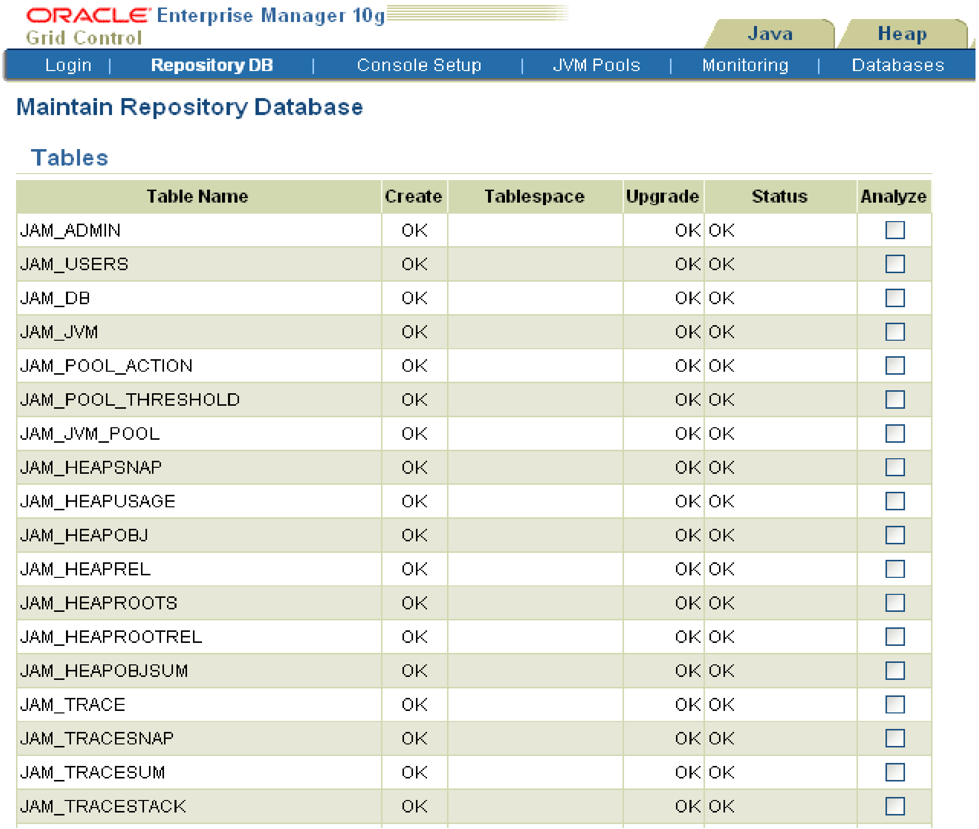
2.1.2.7 Creating the Database Tables

After you specify the database connection details, create the database objects in the repository by following these steps:

1. Click **Setup**.
2. Click **Repository DB**.

   The console displays the Maintain Repository Database page (Figure 2–3).

Figure 2–3 Maintain Repository Database


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases

Maintain Repository Database

Tables

Table Name	Create	Tablespace	Upgrade	Status	Analyze
JAM_ADMIN	OK		OK	OK	□
JAM_USERS	OK		OK	OK	□
JAM_DB	OK		OK	OK	□
JAM_JVM	OK		OK	OK	□
JAM_POOL_ACTION	OK		OK	OK	□
JAM_POOL_THRESHOLD	OK		OK	OK	□
JAM_JVM_POOL	OK		OK	OK	□
JAM_HEAPSNAP	OK		OK	OK	□
JAM_HEAPUSAGE	OK		OK	OK	□
JAM_HEAPOBJ	OK		OK	OK	□
JAM_HEAPREL	OK		OK	OK	□
JAM_HEAPROOTS	OK		OK	OK	□
JAM_HEAPROOTREL	OK		OK	OK	□
JAM_HEAPOBJSUM	OK		OK	OK	□
JAM_TRACE	OK		OK	OK	□
JAM_TRACESNAP	OK		OK	OK	□
JAM_TRACESUM	OK		OK	OK	□
JAM_TRACESTACK	OK		OK	OK	□

3. To create the repository objects, click **Go**.
4. To run the diagnostics, click **Setup**.

   The console is now configured and ready to accept any incoming JVM Connections.

2.1.2.8 Viewing JVM Console Diagnostics

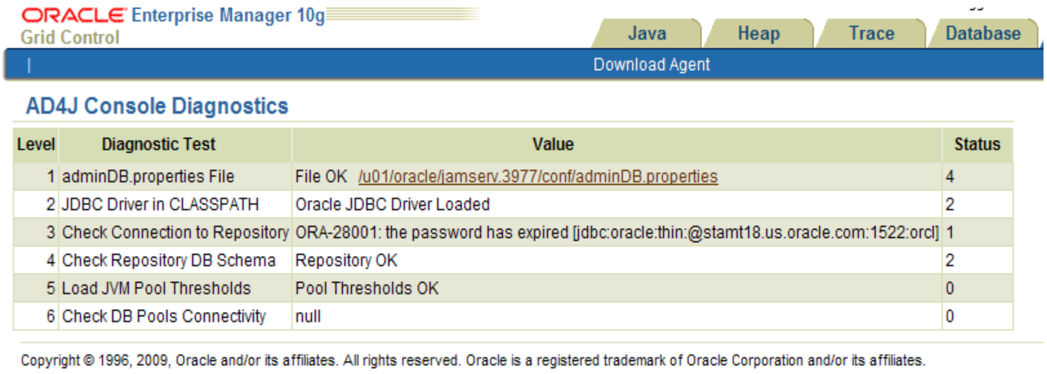
Whenever you click **Setup**, the console runs the diagnostics. Here, you can view details about the configuration of the console and also the errors (if any).

For example, if a database connection cannot be established for some reason, it would show up here along with some debug information.

To view JVM console diagnostics, click **Setup**.

The console displays the following page.

Figure 2–4 AD4J Console Diagnostics



AD4J Console Diagnostics

Level	Diagnostic Test	Value	Status
1	adminDB.properties File	File OK /u01/oracle/jamserv.3977/conf/adminDB.properties	4
2	JDBC Driver in CLASSPATH	Oracle JDBC Driver Loaded	2
3	Check Connection to Repository	ORA-28001: the password has expired [jdbc:oracle:thin:@stamt18.us.oracle.com:1522:orcl]	1
4	Check Repository DB Schema	Repository OK	2
5	Load JVM Pool Thresholds	Pool Thresholds OK	0
6	Check DB Pools Connectivity	null	0

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

2.1.3 Uninstalling the AD4J Console on Unix

To deinstall the Oracle AD4J console from UNIX platforms, go to the location where you extracted the contents of the jamserv.zip file, and delete the jamserv directory completely.

2.2 Installing the AD4J Console on Windows

This section provides installation procedures for installing the Oracle AD4J console on Microsoft Windows platforms, such as Microsoft Windows Itanium, and Microsoft Windows x86_64. It covers the following:

- Prerequisites
- Installation Procedure
- Uninstalling the AD4J Console on Windows

2.2.1 Prerequisites

Before you start the installation process, do the following:

- You must be running Microsoft Windows 2000, XP, or 2003.
- You must download `jamserv-win.msi`. This is the Microsoft Windows Installer package along with the web listener, client database drivers and utilities, JDK HOME, and postgreSQL database.
- You must decide on the location where you want to install the console.
- After installation, you must use the following URL to access the console:

  `http://consolehost:<port_no>`
- You must download the agent to be deployed on the monitored JVMs.
- To log in, you must use the default console setup credentials , *admin* as the user name and *welcome* as the password. After you log in, you must change the default password. You can perform operations on the console only the password has been changed.

2.2.2 Installation Procedure

The installation procedure includes the following steps:

- Running the jamserv-win.msi File
- Selecting the Installation Location
- Setting the Console Parameters
- Specifying the Port Information

2.2.2.1 Running the jamserv-win.msi File

As a first step, run the `jamserv-win.msi` file to invoke the installation wizard. The Welcome screen is displayed. Click **Next** to continue with the installation.

> **Note:** Run the `upgrade-win.zip` if you are upgrading an existing installation.

2.2.2.2 Selecting the Installation Location

On the Select Installation Folder screen, select a directory where you can install the console. The installer creates a subdirectory called *jamserv*. To continue, click **Next**.

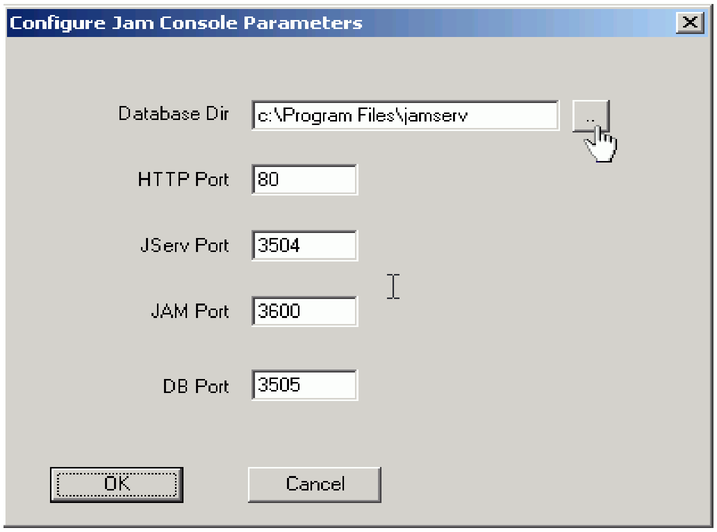
2.2.2.3 Setting the Console Parameters

In the Database Dir field in the Configure JAM Console Parameters screen, select an alternate drive or different location with sufficient space where the database files can be placed. You can specify the path manually or click **Browse** to select the folder.

If you are using the local built-in database, we recommend that you to select an alternate drive because the database can grow to several gigabytes.

Inside this directory, the installer automatically creates a sub-directory titled *pgdata*.

Figure 2–5 Set Console Parameters



Configure Jam Console Parameters

Database Dir: c:\Program Files\jamserv

HTTP Port: 80

JServ Port: 3504

JAM Port: 3600

DB Port: 3505

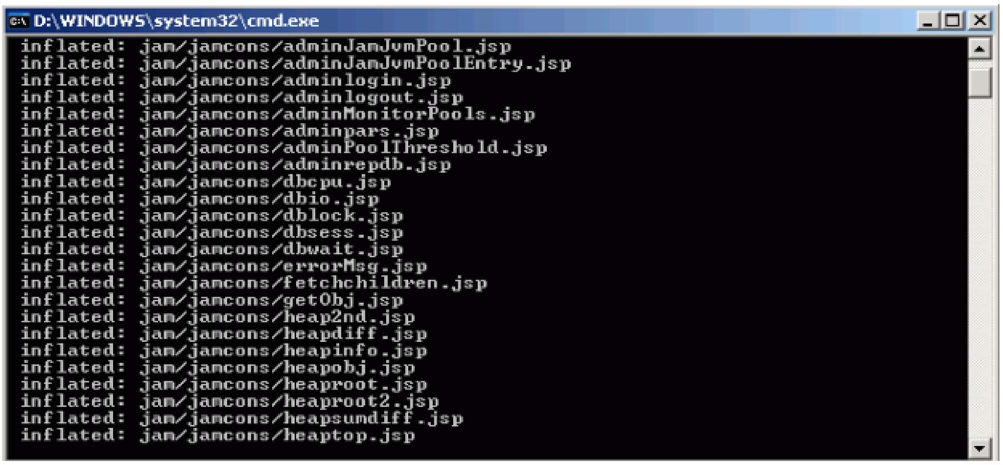
2.2.2.4 Specifying the Port Information

After selecting the database directory, you can accept the default values for other parameters. The http port is the port you use in the URL to access the console. The

Oracle AD4J Port is what you will provide the Oracle AD4J Agents along with the hostname so that the agents can connect to the console. The Jserv port is used internally by the Console and need not be changed.

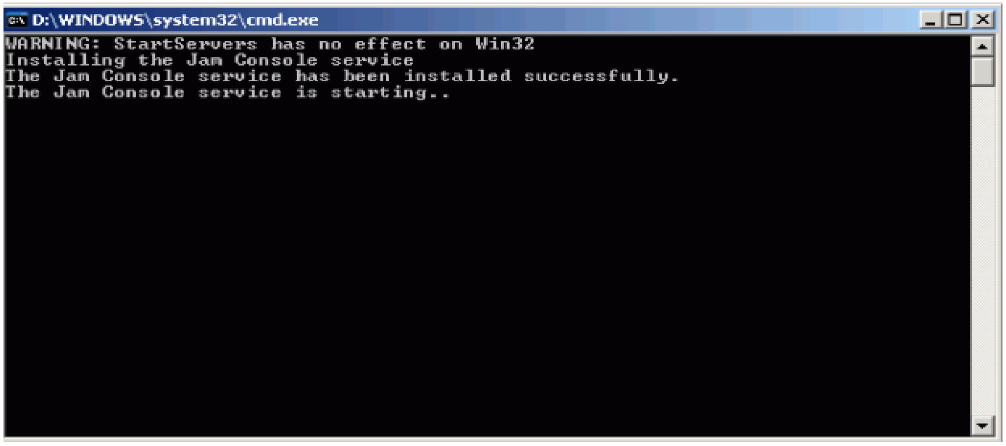
You will see the screen scroll as all the files are extracted and the Oracle AD4J Console service installed and started.

Figure 2–6 Installing Oracle AD4J Console Service 1



```
 inflated: jam/jamcons/adminJamJvmPool.jsp
 inflated: jam/jamcons/adminJamJvmPoolEntry.jsp
 inflated: jam/jamcons/adminlogin.jsp
 inflated: jam/jamcons/adminlogout.jsp
 inflated: jam/jamcons/adminMonitorPools.jsp
 inflated: jam/jamcons/adminpars.jsp
 inflated: jam/jamcons/adminPoolThreshold.jsp
 inflated: jam/jamcons/adminrepdb.jsp
 inflated: jam/jamcons/dbcpu.jsp
 inflated: jam/jamcons/dbio.jsp
 inflated: jam/jamcons/dblock.jsp
 inflated: jam/jamcons/dbsess.jsp
 inflated: jam/jamcons/dbwait.jsp
 inflated: jam/jamcons/errorMsg.jsp
 inflated: jam/jamcons/fetchchildren.jsp
 inflated: jam/jamcons/getObj.jsp
 inflated: jam/jamcons/heap2nd.jsp
 inflated: jam/jamcons/heapdiff.jsp
 inflated: jam/jamcons/heapinfo.jsp
 inflated: jam/jamcons/heapobj.jsp
 inflated: jam/jamcons/heaproot.jsp
 inflated: jam/jamcons/heaproot2.jsp
 inflated: jam/jamcons/heapsumdiff.jsp
 inflated: jam/jamcons/heaptop.jsp
```

Figure 2–7 Installing Oracle AD4J Console Service 2



```
WARNING: StartServers has no effect on Win32
Installing the Jam Console service
The Jam Console service has been installed successfully.
The Jam Console service is starting..
```

The Installer creates self-signed certificates to enable secure http (https) access on port 443 to the console. When the certificates and the certification authority (CA) are created, you are prompted with the following questions twice. Once when creating the certification authority (CA) and again when creating the certificates.

Table 2–5 Prompted Questions

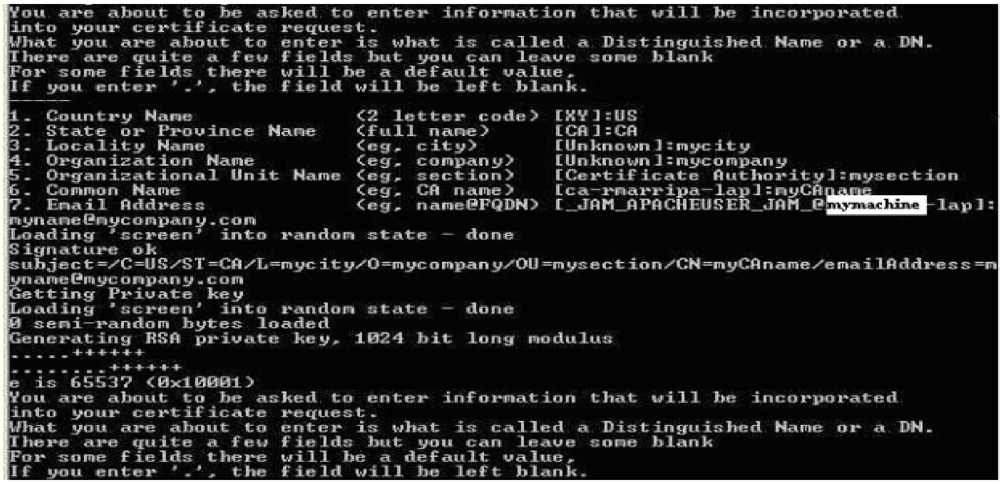
Country Name	(2 letter code) [XY]:
State or Province Name	(full name) [CA]:
Locality Name	(e.g., city) [Unknown]:
Organization Name	(e.g., company) [Unknown]:
Organizational Unit Name	(e.g., section) [Certificate Authority]:
Common Name	(e.g., CA name) [mymachine01]:

Table 2–5 (Cont.) Prompted Questions

Email Address	(e.g., name@FQDN) [myname@mycompany.com]:

The following is the install output. Note that install prompts you for these questions twice.

Figure 2–8 Install Output



```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
1. Country Name                 <2 letter code> [XY]:US
2. State or Province Name       <full name>     [CA]:CA
3. Locality Name                <eg, city>      [Unknown]:mycity
4. Organization Name            <eg, company>   [Unknown]:mycompany
5. Organizational Unit Name     <eg, section>   [Certificate Authority]:mysection
6. Common Name                  <eg, CA name>   [ca-rmarripa-lap]:myCAname
7. Email Address                <eg, name@FQDN> [_JAM_APACHEUSER_JAM_@mymachine-lap]:
myname@mycompany.com
Loading 'screen' into random state - done
Signature ok
subject=/C=US/ST=CA/L=mycity/O=mycompany/OU=mysection/CN=myCAname/emailAddress=m
yname@mycompany.com
Getting Private key
Loading 'screen' into random state - done
0 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....++++++
.........++++++
e is 65537 (0x10001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

It is important that when creating the certificate (second set of prompts), the Common Name and prompt #5 reflect the server name you use in the URL to access. While the install script tries to determine the host name, you must ensure that the domain is accurate.

For example: If the server name is `console01`, you can only specify `https://console01:XXXX`.

You cannot specify `https://console01.mydomain.com:XXXX` or `http://123.123.123.123:XXXX` where XXXX is the https port for the Apache listener. You can, however, specify all the above names on the http port

The HTTPS port (default 443) can be changed by changing the following files:

```
jamserv/conf/httpd.conf
```

```
jamserv/conf/ssl.conf
```

It is strongly recommended that you replace this out-of-box certificate with a certificates issued by a well-known certificate authority.

The installation is now complete and shortcuts to stop and start the console service, load the heaps, and uninstall the product are also available.

2.2.3 Uninstalling the AD4J Console on Windows

To uninstall the AD4J Console on Windows, from the **Start** menu, click **Programs**, and **JAM Console**. Then click **Uninstall JADE** to uninstall the console.

2.3 Post-Installation Tasks

This section covers the following:

- Accessing the AD4J Console for the First Time

- Setting Up the Console and Diagnostics
- Deploying the Agent on Monitored JVMs and Databases

2.3.1 Accessing the AD4J Console for the First Time

If you want to access the console from the host where you installed it, then use the following URL:

`http://localhost:3500 or http://localhost:80` (on Windows)

If you want to access the console from a different host, then use the following URL:

`http://hostname:3500 or http://hostname:80` (on Windows)

Here, *hostname* is the host on which you installed the console.

When you access the Oracle AD4J console for the first time, it does the following:

- Runs diagnostics
- Prompts you for the database connection details for the repository and stores it in the admin.DBProperties file
- Creates the necessary DB objects in the repository

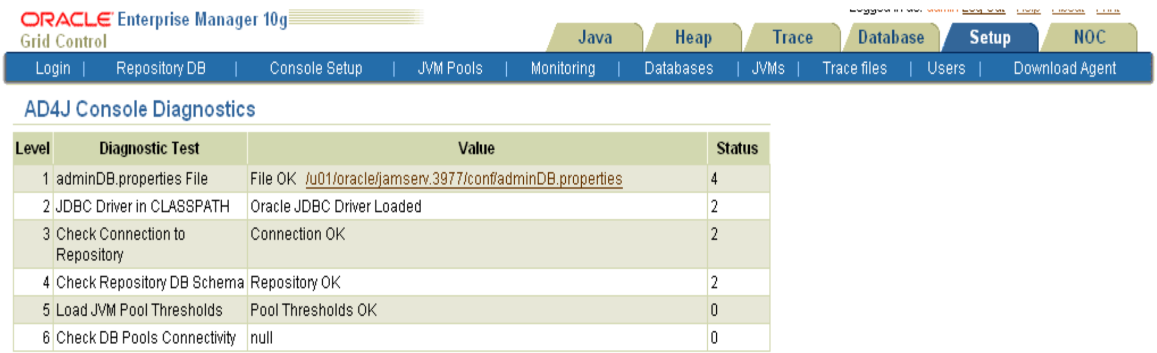
2.3.2 Setting Up the Console and Diagnostics

The console must be accessible from the URL `http://localhost` if accessing from the machine where it was installed. From another machine, it must be accessible from `http://hostname` where the host name is the machine on which your console is installed. The default password for the admin user is `welcome`.

When you access the Oracle AD4J console for the first time, it will run Diagnostics.

Whenever you click on the Setup tab, it will run Diagnostics. Here, you can see information about the configuration of the console and the errors (if any). For example, if a database connection cannot be established for some reason, it would show up here along with some debug information.

Figure 2–9 AD4J Console Diagnostics



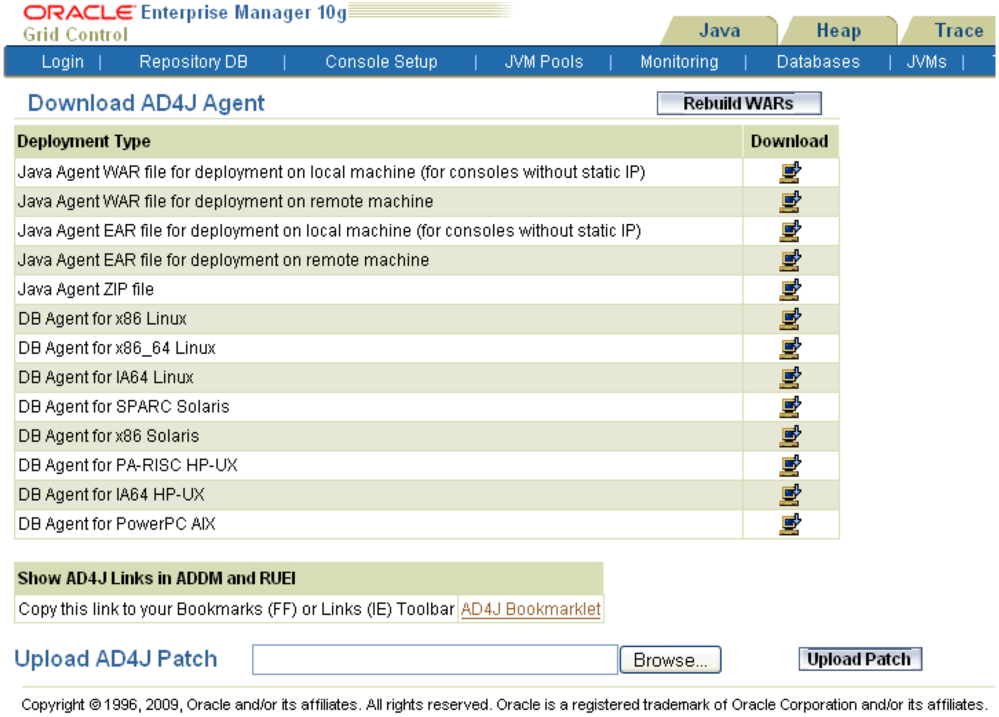
AD4J Console Diagnostics

Level	Diagnostic Test	Value	Status
1	adminDB.properties File	File OK /u01/oracle/jamserv.3977/conf/adminDB.properties	4
2	JDBC Driver in CLASSPATH	Oracle JDBC Driver Loaded	2
3	Check Connection to Repository	Connection OK	2
4	Check Repository DB Schema	Repository OK	2
5	Load JVM Pool Thresholds	Pool Thresholds OK	0
6	Check DB Pools Connectivity	null	0

2.3.3 Deploying the Agent on Monitored JVMs and Databases

The agents can be downloaded from the Download Agent screen. To display this screen, select the **Setup** tab from the console and click **Download Agent**.

Figure 2–10 Downloading Agents


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs |
Rebuild WARs
Browse...
Upload Patch

Download AD4J Agent

Deployment Type	Download
Java Agent WAR file for deployment on local machine (for consoles without static IP)	⬇
Java Agent WAR file for deployment on remote machine	⬇
Java Agent EAR file for deployment on local machine (for consoles without static IP)	⬇
Java Agent EAR file for deployment on remote machine	⬇
Java Agent ZIP file	⬇
DB Agent for x86 Linux	⬇
DB Agent for x86_64 Linux	⬇
DB Agent for IA64 Linux	⬇
DB Agent for SPARC Solaris	⬇
DB Agent for x86 Solaris	⬇
DB Agent for PA-RISC HP-UX	⬇
DB Agent for IA64 HP-UX	⬇
DB Agent for PowerPC AIX	⬇

Show AD4J Links in ADDM and RUEI
Copy this link to your Bookmarks (FF) or Links (IE) Toolbar AD4J Bookmarklet

Upload AD4J Patch

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

If you are running the console on a machine with DHCP, and the agent is running on the same machine, you can use the WAR file for deployment on local machine. Otherwise, use the WAR file for deployment on remote machine.

3

Installing the AD4J Agent

This chapter describes the procedure to install the AD4J Agent. It covers the following:

- Before you Start
- Installing the AD4J Agent on Oracle Application Server 10g
- Installing the AD4J Agent on Oracle WebLogic
- Installing the AD4J Agent on Apache Tomcat
- Installing the Oracle AD4J Agent on Apache JServ
- Installing the Oracle AD4J Agent on IBM WebSphere
- Installing the Oracle AD4J Agent on Standalone Java Applications
- Installing the Database Agent

3.1 Before you Start

To install the AD4J Agent, you must do the following:

- Login to your Application Server console.
- Download the Java agent from the Oracle AD4J Console and deploy it to the JVM using the standard deployment techniques for your Application Server.
- If necessary, customize the AD4J Agent and specify the parameters as described in Customizing the Oracle AD4J Agent.
- Download the DB Agent from the Oracle AD4J Console and deploy it to your database.

3.1.1 Customizing the Oracle AD4J Agent

The AD4J Agent WAR file contains an embedded `web.xml` deployment descriptor. This file contains the default values of the Oracle AD4J Agent input parameters. You can change the parameters if you want. To do so, extract the `web.xml` file from the WAR archive and then update it.

To explode or update the WAR file, do the following:

- To extract the `web.xml` file from the WAR archive, run the following command:

  ```
  jar xvf jamagent.war WEB-INF/web.xml
  ```

- Edit the `web.xml` file with your custom values for input parameters in any text editor. You may only change the values within the <param-name> fields of this file. No other changes are supported.

- Then run the following command to place the new `web.xml` file back into the archive:

  ```
  jar uvf jamagent.war WEB-INF/web.xml
  ```

- To remove the newly created directory, run the following command:

  ```
  rm -rf WEB-INF
  ```

The following are the Oracle AD4J Agent parameters.

Table 3–1 Oracle AD4J Agent Parameters

Parameter	Default	Description
jamconshost	localhost	The server where console is running.
jamconsport	3600	The port where the console is listening for the agents.
jamjvmid	Application Server Port or 5555	Identifies the specific JVM on the console when you have multiple JVMs on the same machine. For most app servers, this identifier is the port which the server is listening on (Web Server Port for Weblogic, Jserv port for Apache). If the application server port cannot be discovered then this value is used.
jamconsretr	90	If the console goes down, the agent will keep trying to reconnect. This parameter specifies the duration in seconds between each attempt. The default value is 90 seconds (15 minutes) between each try. If this parameter is set to 0, then the agent will not try to reconnect.
jamtimeout	300	Maximum time duration for a request. If a request takes longer time than this timeout, it is terminated.
jamloglevel	3	Level of logging. Valid values range from 1 to 5.
jammaxbackoff	10	Some times we wait for other operations to finish (like GC or main thread initialization). The amount of sleep time between the retries increases exponentially till this number. When this number is reached, we give up and return with failure.
jamdelaystartup	0	This parameter is only for standalone programs. If you want to analyze the startup behavior of a monitored program, you can use this parameter. This specifies the number of seconds to wait before starting the target program.
jamisdaemon	false	This parameter is only for standalone programs. When using jamrun with standalone programs, the agent will normally exit when the main method in the target program completes. In some programs the threads might still be active after main completes.In such cases specify jamisdaemon=true to prevent the agent from exiting.

3.1.1.1 Web.XML Contents

```
<?xml version="1.0" ?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN""http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
<servlet>
<servlet-name>jamagent</servlet-name>
<servlet-class>jamagent.jaminit</servlet-class>
<init-param>
<param-name>jamconshost</param-name>
<param-value>localhost</param-value>
<description>Default Jam Console host</description>
</init-param>
<init-param>
<param-name>jamconsport</param-name>
<param-value>3600</param-value>
<description>Jam console port</description>
```

```
</init-param>
<init-param>
<param-name>jamconsretr</param-name>
<param-value>90</param-value>
<description>Jam console number of retries</description>
</init-param>
<init-param>
<param-name>jamtimeout</param-name>
<param-value>900</param-value>
<description>Jam console timeout</description>
</init-param>
<init-param>
<param-name>jamloglevel</param-name>
<param-value>3</param-value>
<description>Jam log level</description>
</init-param>
<init-param>
<param-name>jammaxbackoff</param-name>
<param-value>10</param-value>
<description>Max time to wait for long operations</description>
</init-param>
<init-param>
<param-name>jamjvmid</param-name>
<param-value>Aplication Server Port</param-value>
<description> Unique Identifier for JVM. It will detect and use the WLS port by
default.</description>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
</web-app>
```

3.2 Installing the AD4J Agent on Oracle Application Server 10g

This section describes the procedure to install the AD4J Agent on Oracle Application Server 10g. To install the Agent, you must do the following:

- Login to the Enterprise Manager 10g console to administer Application Server 10g. Enterprise Manager 10g is installed by default when you install iAS 10g. Make sure you can log on to Enterprise Manager as the administrative user, `ias_admin`.
- Deploy the `jamgent.war` file from the Oracle AD4J Console as described in Deploying the Agent WAR.
- Customize the Oracle AD4J Agent and specify the parameters. The `jamgent.war` file contains the java classes and native libraries that constitute the Oracle AD4J Agent. It also contains the default Oracle AD4J Agent input parameters in the packaged `web.xml` file. See Customizing the Oracle AD4J Agent for details.

3.2.1 Deploying the Agent WAR

The following steps assume that the Oracle AD4J Console is running on the host where the .war file is deployed. The steps also assume that Oracle AD4J Console is using the default port, that is, port 3600. If not, then change the default settings. For information about the default settings and the procedure to change them, refer to Customizing the Oracle AD4J Agent.

To deploy the agent WAR file:

1. Log on to Enterprise Manager iAS 10g as `ias_admin.`

Enterprise Manager displays the Home page by default.

2. On the Home page, in the **System Components** section, you will find an OC4J instance called *home*. This OC4J instance is available by default. Click the *home* instance name to open the administration page of this OC4J instance.
3. Click the **Applications** tab.

   Enterprise Manager displays the page for managing applications deployed in the *home* OC4J instance.
4. Click **Deploy WAR file** and provide the following details:
   - **Web Application** - Specify the path (along with the file name) for the `jamgent.war` file. You can click **Browse** to locate the file on your computer.
   - **Application Name** - Specify `jamagent`.
   - **Map to URL** - Specify `/jamagent.`
5. Click **Deploy** to deploy the `jamagent.war` file.
6. To test the deployment, log on to the Oracle AD4J Console and view the JVM running the OC4J instance on the main page.

3.3 Installing the AD4J Agent on Oracle WebLogic

This section describes the procedure to install the AD4J Agent on Oracle WebLogic. Before you install the AD4J Agent, you must have installed and configured the Oracle AD4J Console. You must also ensure that the Weblogic Administration Console is up and running. To install the AD4J Agent, you must do the following:

- Login to the Oracle WebLogic Admin Console.
- Deploy the `jamgent.war` containing the agent classes and the native libraries as described in Deploying the Agent WAR.
- Customize the Oracle AD4J Agent and specify the parameters as described in Customizing the Oracle AD4J Agent. The `jamagent.war` also contains the default Oracle AD4J Agent input parameters in the packaged `web.xml` file.

3.3.1 Deploying the Agent WAR

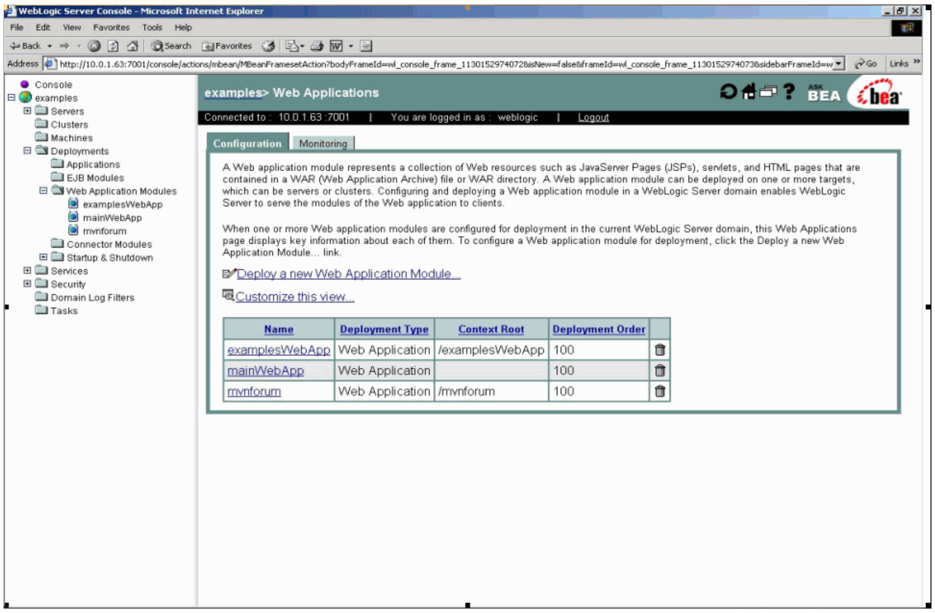
To deploy the Agent WAR file, do the following:

1. Log on to the Administration Console using the following URL:

   `http://weblogichost:ADMINPORT/console`
2. From the tree view, select **Deployments** and then **Web Application Modules**.

   The Web Applications page displays the web applications currently deployed on the Weblogic Application Server.

Figure 3–1 Web Application Modules


WebLogic Server Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.0.1.63:7001/console/actions/mbean/MBeanFramesetAction?bodyFrameId=wl_console_frame_1130152974072&isNew=false&frameId=wl_console_frame_1130152974073&sidebarFrameId=w
Go Links
Console
examples
Servers
Clusters
Machines
Deployments
Applications
EJB Modules
Web Application Modules
examplesWebApp
mainWebApp
mvnforum
Connector Modules
Startup & Shutdown
Services
Security
Domain Log Filters
Tasks
ASK BEA
bea
Configuration
Monitoring

examples> Web Applications

Connected to : 10.0.1.63 :7001 | You are logged in as : weblogic | Logout

A Web application module represents a collection of Web resources such as JavaServer Pages (JSPs), servlets, and HTML pages that are contained in a WAR (Web Application Archive) file or WAR directory. A Web application module can be deployed on one or more targets, which can be servers or clusters. Configuring and deploying a Web application module in a WebLogic Server domain enables WebLogic Server to serve the modules of the Web application to clients.

When one or more Web application modules are configured for deployment in the current WebLogic Server domain, this Web Applications page displays key information about each of them. To configure a Web application module for deployment, click the Deploy a new Web Application Module... link.

Deploy a new Web Application Module...

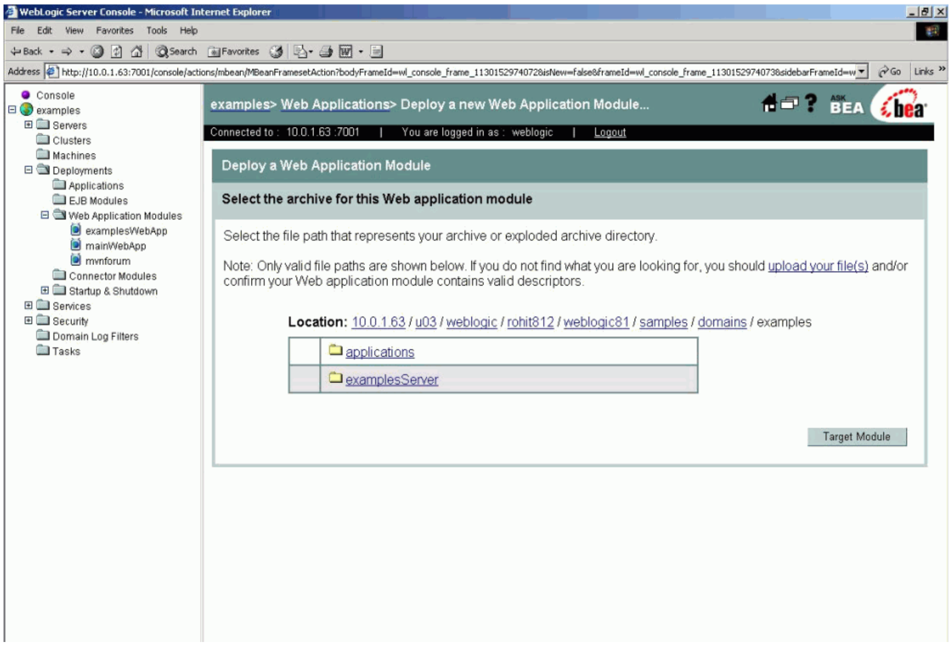
Customize this view...

Name	Deployment Type	Context Root	Deployment Order	
examplesWebApp	Web Application	/examplesWebApp	100	🗑
mainWebApp	Web Application		100	🗑
mvnforum	Web Application	/mvnforum	100	🗑

3. On the Web Applications page, click **Deploy a New Web Application Module**.

   The Deploy a Web Application Module page displays.

Figure 3–2 Uploading the WAR Archive


WebLogic Server Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.0.1.63:7001/console/actions/mbean/MBeanFramesetAction?bodyFrameId=wl_console_frame_1130152974072&isNew=false&frameId=wl_console_frame_1130152974073&sidebarFrameId=w
Go Links
Console
examples
Servers
Clusters
Machines
Deployments
Applications
EJB Modules
Web Application Modules
examplesWebApp
mainWebApp
mvnforum
Connector Modules
Startup & Shutdown
Services
Security
Domain Log Filters
Tasks
ASK BEA
bea
Target Module

examples> Web Applications> Deploy a new Web Application Module...

Connected to : 10.0.1.63 :7001 | You are logged in as : weblogic | Logout

Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your Web application module contains valid descriptors.

Location: 10.0.1.63 / u03 / weblogic / rohit812 / weblogic81 / samples / domains / examples

	applications
	examplesServer

4. On the Deploy a Web Application Module page, do the following:

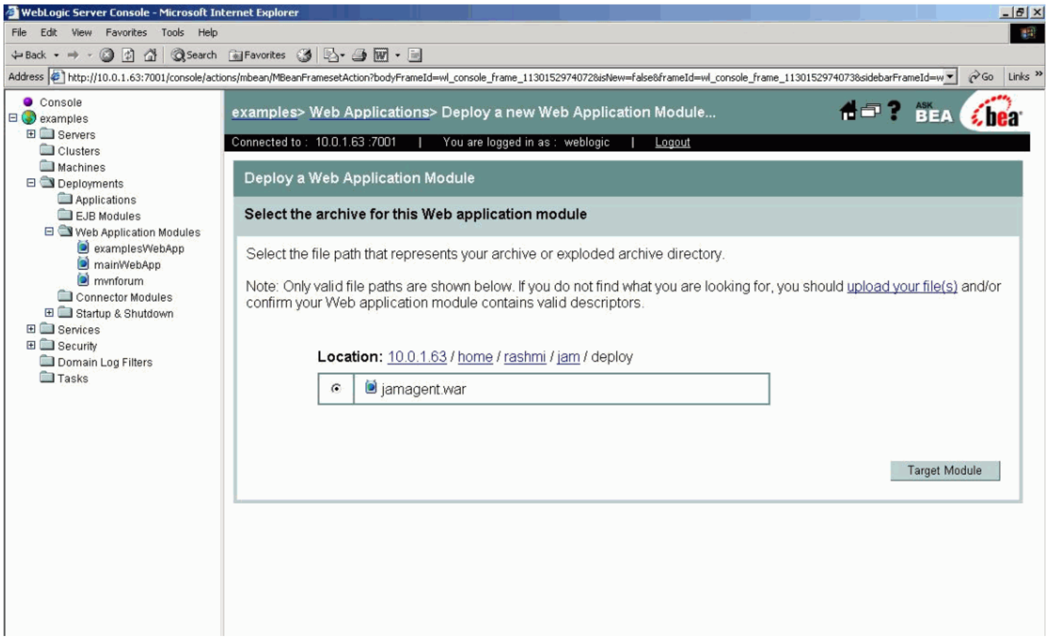
   - Upload the WAR archive directly from the local file system.
   - Choose the WAR archive placed on the server.

- Use the exploded WAR on the server

Explore all three approaches.

5. Select the jamagent.WAR file.

Figure 3–3 Choosing the jamagent.WAR file


WebLogic Server Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.0.1.63:7001/console/actions/mbean/MBeanFramesetAction?bodyFrameId=wl_console_frame_1130152974072&isNew=false&frameId=wl_console_frame_1130152974073&sidebarFrameId=w
Go Links
Console
examples
Servers
Clusters
Machines
Deployments
Applications
EJB Modules
Web Application Modules
examplesWebApp
mainWebApp
mvnforum
Connector Modules
Startup & Shutdown
Services
Security
Domain Log Filters
Tasks
ASK BEA
bea

examples> Web Applications> Deploy a new Web Application Module...

Connected to : 10.0.1.63 :7001 | You are logged in as : weblogic | Logout

Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your Web application module contains valid descriptors.

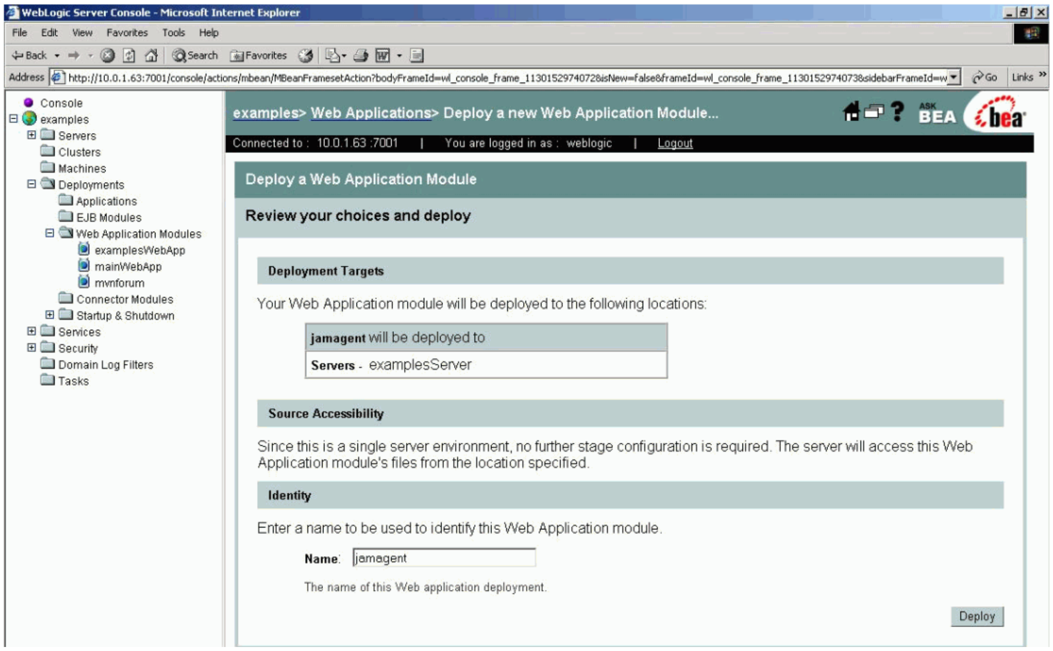
Location: 10.0.1.63 / home / rashmi / jam / deploy

● jamagent.war

Target Module

6. Go to the next screen and click **Deploy**.

Figure 3–4 Deploy a Web Application Module


WebLogic Server Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.0.1.63:7001/console/actions/mbean/MBeanFramesetAction?bodyFrameId=wl_console_frame_1130152974072&isNew=false&frameId=wl_console_frame_1130152974073&sidebarFrameId=w
Go Links
Console
examples
Servers
Clusters
Machines
Deployments
Applications
EJB Modules
Web Application Modules
examplesWebApp
mainWebApp
mvnforum
Connector Modules
Startup & Shutdown
Services
Security
Domain Log Filters
Tasks
ASK BEA
bea

examples> Web Applications> Deploy a new Web Application Module...

Connected to : 10.0.1.63 :7001 | You are logged in as : weblogic | Logout

Deploy a Web Application Module

Review your choices and deploy

Deployment Targets

Your Web Application module will be deployed to the following locations:

jamagent will be deployed to

Servers - examplesServer

Source Accessibility

Since this is a single server environment, no further stage configuration is required. The server will access this Web Application module's files from the location specified.

Identity

Enter a name to be used to identify this Web Application module.

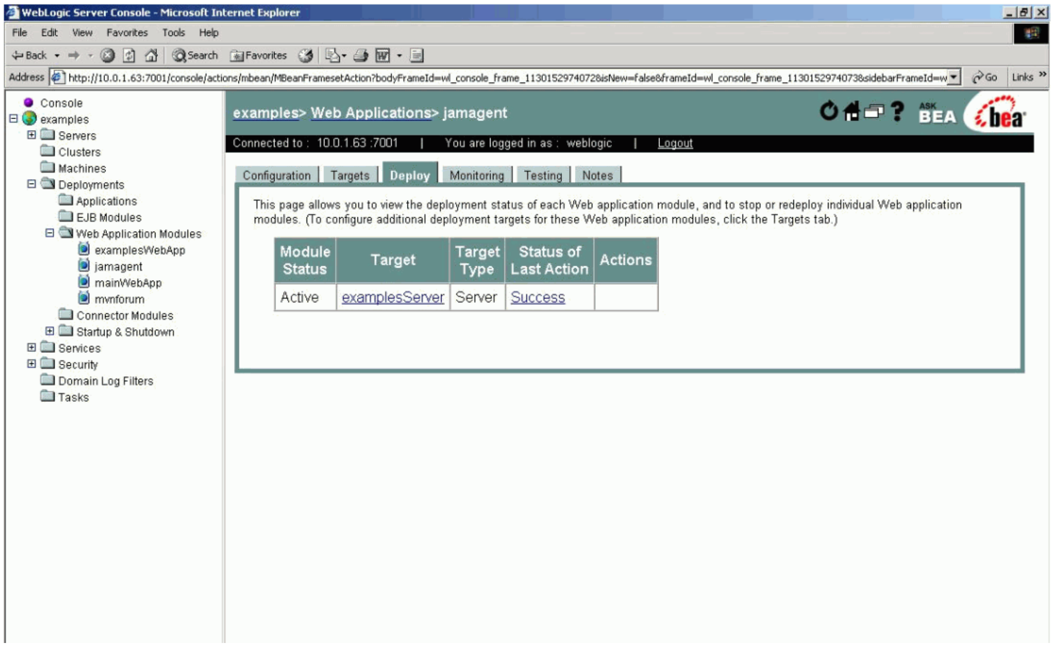
Name: jamagent

The name of this Web application deployment.

Deploy

A successful deployment takes you to the following screen:

Figure 3–5 Successful Deployment


WebLogic Server Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.0.1.63:7001/console/actions/mbean/MBeanFramesetAction?bodyFrameId=wl_console_frame_1130152974072&isNew=false&frameId=wl_console_frame_1130152974073&sidebarFrameId=w
Go Links
Console
examples
Servers
Clusters
Machines
Deployments
Applications
EJB Modules
Web Application Modules
examplesWebApp
jamagent
mainWebApp
mvnforum
Connector Modules
Startup & Shutdown
Services
Security
Domain Log Filters
Tasks
examples> Web Applications> jamagent
ASK BEA
bea
Connected to : 10.0.1.63 :7001 | You are logged in as : weblogic | Logout
Configuration
Targets
Deploy
Monitoring
Testing
Notes

This page allows you to view the deployment status of each Web application module, and to stop or redeploy individual Web application modules. (To configure additional deployment targets for these Web application modules, click the Targets tab.)

Module Status	Target	Target Type	Status of Last Action	Actions
Active	examplesServer	Server	Success	

3.4 Installing the AD4J Agent on Apache Tomcat

This section describes the procedure to install the AD4J Agent on Apache Tomcat. Before you install the agent, you must ensure that Tomcat has been correctly installed. You can test this by starting Tomcat and bringing up the default page - `http://<host>:8080`. If you have not installed on the default port, replace 8080 with your port number. To install the agent, you must do the following:

- Copy `jamagent.war` to `$CATALINA_HOME/webapps` directory.
- Deploy the `jamagent.war` file containing the java classes and native libraries that constitute the Oracle AD4J Agent. See Deploying the Agent WAR for details.
- Customize the Oracle AD4J Agent and specify the parameters as described in Customizing the Oracle AD4J Agent. The jamagent.war also contains the default Oracle AD4J Agent input parameters in the packaged `web.xml` file.
- Restart Tomcat.

3.4.1 Deploying the Agent WAR

These steps assume that the Oracle AD4J Console is running on the same host where the war file is being deployed. It also assumes that Oracle AD4J Console is using the default port - 3600. If not, refer to Customizing the Oracle AD4J Agent to change these default settings.

1. Copy the `jamagent.war` to the `$CATALINA_HOME/webapps` directory.
2. Shutdown Tomcat using `$CATALINA_HOME/bin/shutdown.sh`
3. Startup Tomcat using `$CATALINA_HOME/bin/startup.sh`

4. Tomcat automatically explodes the war and deploys it.
5. To test, log on to the Oracle AD4J Console. The jvm running Tomcat now appears on the main page.

3.5 Installing the Oracle AD4J Agent on Apache JServ

This section describes the procedure to install the Oracle AD4J Agent on Apache JServ. To install the agent, you must do the following:

- Include the file `jamagent.zip` in the `CLASS_PATH`. This is described in:
  - Setting Up the CLASSPATH - Automatic Mode
  - Setting Up the CLASSPATH - Manual Mode
- Edit the `Jserv` configuration files to load the servlet `jamagent.jaminit` as describe in Loading the Servlet at Startup Time.

> **Note:** You need a copy of `jamagent.zip` for your platform and a working console for the agent to connect to. (The console must be up and listening for JVM connections on localhost port 3600).

- Customize the Oracle AD4J Agent and specify the parameters as described in Customizing the Oracle AD4J Agent.

3.5.1 Setting Up the CLASSPATH - Automatic Mode

If you are running Apache in automatic mode (jserv.conf: ApJServManual is off), apache automatically starts the JVMs. The CLASSPATH is specified in jserv.properties by the parameter wrapper.classpath. If your jamagent.zip lives in `/u01/app/jam/jamagent,` then you'll add the following line to jserv.properties `wrapper.classpath=/u01/app/jam/jamagent/jamagent.zip`.

3.5.2 Setting Up the CLASSPATH - Manual Mode

When ApJServManual is on, you must use a separate script (typically called jservctl) to start the JVM. You have to add jamagent.zip to the CLASS_PATH in this file. You can do this in either of these methods:

- CLASSPATH is defined as an environment variable
- CLASSPATH is passed with -classpath to the java command

If your `jamagent.zip` is located in the `/u01/app/jam/jamagent` directory, you must add `/u01/app/jam/jamgent/jamagent.zip` to the CLASSPATH.

3.5.3 Loading the Servlet at Startup Time

To load the servlet at startup time, add the following lines to the `zone.properties` file for Apache Jserv:

```
servlets.startup=jamagent.jaminit

servlet.jamagent.jaminit.initArgs=jamconshost=localhost

servlet.jamagent.jaminit.initArgs=jamconsport=3600
```

The first line instructs the servlet container to start the servlet `jamagent.jaminit`. The other one passes the startup parameters `jamconshost=localhost` and `jamconsport=3600` to specify the location of the console.

3.6 Installing the Oracle AD4J Agent on IBM WebSphere

This section describes the procedure to install the Oracle AD4J Agent on IBM WebSphere. Before you install the agent, the WebSphere Administrative Console should be up and running. You must also have installed and configured Oracle AD4J Console. For WebSphere installation, you require a `jamagent.war` file that contains the java classes and native libraries that constitute the Oracle AD4J Agent. To install the agent, you must do the following:

- Log on to the IBM WebSphere Application Server Express Administrative Console.
- Deploy `jamagent.war` containing the agent classes and native libraries as described in Deploying the Agent WAR.
- Customize the Oracle AD4J Agent and specify the parameters as described in Customizing the Oracle AD4J Agent. The `jamagent.war` also contains the default Oracle AD4J Agent input parameters in the packaged `web.xml` file.

3.6.1 Deploying the Agent WAR

To deploy the Agent WAR file, do the following:

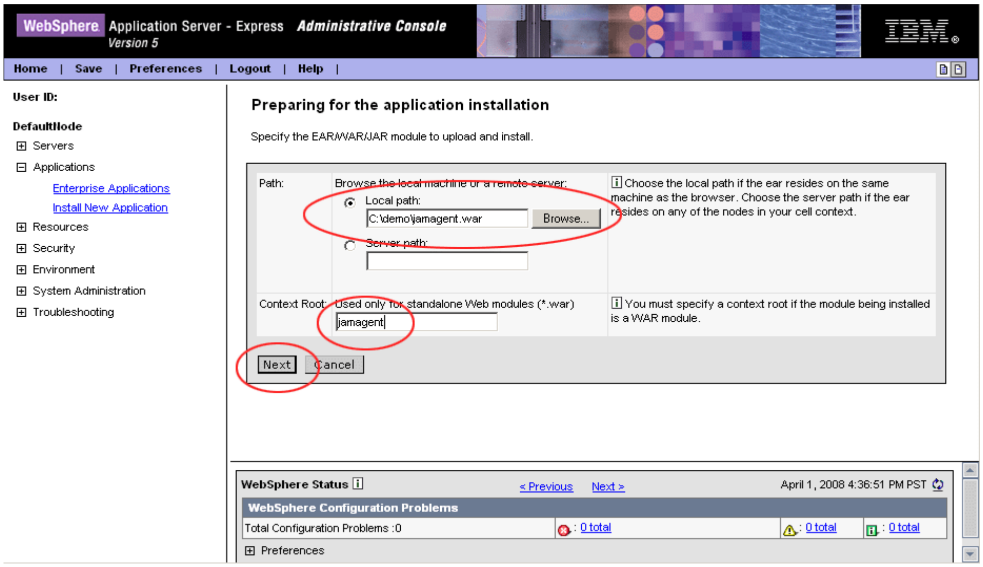
1. Log on to the Administrative Console using the following URL:

   `http://hostname:7090/admin`

2. From the tree view, select **Applications** and then **Install New Application**.

   The Preparing for Application Installation page opens where you can select the module to upload.

Figure 3–6 Preparing for Application Installation



Preparing for the application installation

Specify the EAR/WAR/JAR module to upload and install.

Path: Browse the local machine or a remote server:
● Local path: C:\demo\jamagent.war
○ Server path:

Choose the local path if the ear resides on the same machine as the browser. Choose the server path if the ear resides on any of the nodes in your cell context.

Context Root: Used only for standalone Web modules (*.war)
jamagent

You must specify a context root if the module being installed is a WAR module.

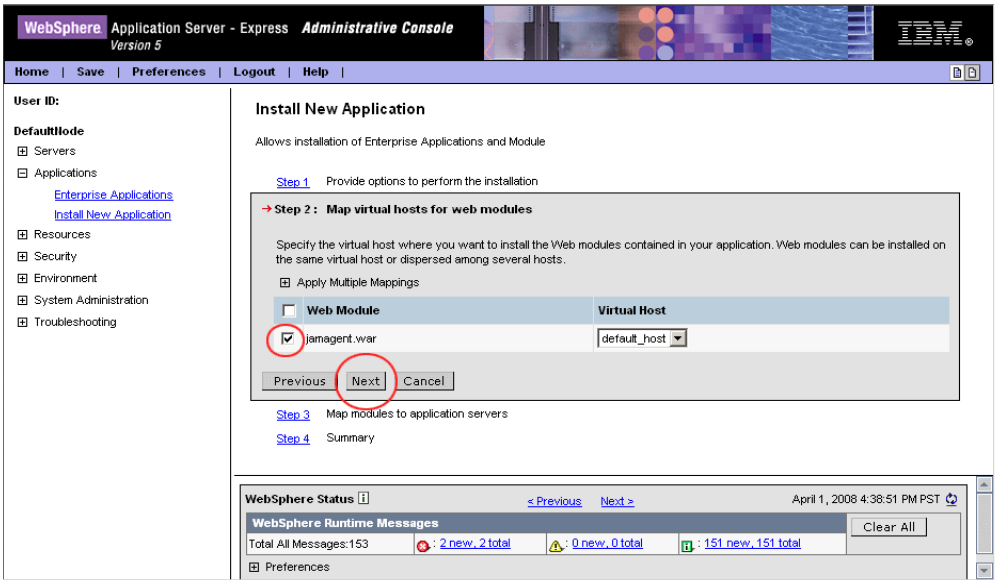
April 1, 2008 4:36:51 PM PST

WebSphere Configuration Problems

Total Configuration Problems :0 : 0 total : 0 total : 0 total

3. Click **Browse** to choose the downloaded jamagent.war file from your desktop.
4. Provide a context root of jamagent in the **Context Root** text box.
5. Select the **Generate Default Bindings** and click **Next**.
6. Review warning if any and click **Continue**.
7. Select the **Enable Class Reloading** check box from the AppDeployment Options section and click **Next**. (figure)
8. Select **jamagent.war** from the Step:2 Map virtual hosts for Web Module section of the Install New Application page and click **Next**.

Figure 3–7 Install New Application - Step 2



Install New Application

Allows installation of Enterprise Applications and Module

Step 1 Provide options to perform the installation

→ Step 2 : Map virtual hosts for web modules

Specify the virtual host where you want to install the Web modules contained in your application. Web modules can be installed on the same virtual host or dispersed among several hosts.

⊞ Apply Multiple Mappings

□	Web Module	Virtual Host
☑	jamagent.war	default_host

Step 3 Map modules to application servers

Step 4 Summary

WebSphere Status April 1, 2008 4:38:51 PM PST

WebSphere Runtime Messages

Total All Messages:153 | ⊗ : 2 new, 2 total | ⚠ : 0 new, 0 total | ⓘ : 151 new, 151 total

9. Select **jamagent.war** for Step 3: Map modules to application Servers as well and click **Finish**.
10. Click Save to **Master Configuration** and then click **Save**.
11. Click the **Enterprise Application** from the tree view on the left side.
12. Select jamagent.war and click **Start**.

    AD4J agent is deployed and started.

3.7 Installing the Oracle AD4J Agent on Standalone Java Applications

This section describes the procedure to install the Oracle AD4J Agent on Standalone Java Applications. Before you install the agent, you must have installed and configured the Oracle AD4J Console. To install the agent, you must do the following:

- Include the `jamagent.zip` in the `CLASSPATH`. See Adding the Oracle AD4J Agent and Libraries, Classes to Classpath.
- Change the java call to call the `jamagent.jamrun` wrapper class. The wrapper class is required to monitor and diagnose Java programs that do not have a way of loading a servlet or another class. The following sections describe the original and modified java call:

  - Original Java Call
  - Modified Java Call with Oracle AD4J Agent
- Customize the Oracle AD4J Agent and specify the parameters as described in Customizing the Oracle AD4J Agent.

3.7.1 Original Java Call

```
java $JVM_OPTIONS $TARGET_CLASS $TARGET_CLASS_PARAMS
```

Here, JVM_OPTIONS are the JVM properties and options.

For example:

```
 -Xmx512M -Dweblogic.name
```

TARGET_CLASS is the program which being examined

TARGET_CLASS_PARAMS are parameters passed to the class

The idea is to run the target program through `jamagent.jamrun`. This is done by calling the `jamagent.jamrun`.

3.7.2 Modified Java Call with Oracle AD4J Agent

```
java $JVM_OPTIONS jamagent.jamrun [$JAMAGENT_PARAMS_LIST]
$TARGET_CLASS $TARGET_CLASS_PARAMS
```

Oracle AD4J uses default parameters. If you want to change any parameter, you can specify them as name=value pairs. To specify a different console and port, enter the following command:

```
jamconshost=<console hostname>, jamconsport=<console port>
```

Thus, to run the Oracle AD4J Agent, you just need to edit the script which starts your program.

It is not required to do anything with the target program parameters. They are picked up and passed along to the program by jamrun.

3.7.3 Sample Configuration

In this example, the MainClass is being called and the steps to edit the Java Call are shown below:

```
"$JAVA_HOME/bin/java" ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
com.example.MainClass
```

The class being called is `com.example.MainClass`. There are no parameters. Insert the class and parameters before the class as follows:

```
"$JAVA_HOME/bin/java" ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
jamagent.jamrun com.example.MainClass
```

If you want to change some Oracle AD4J parameters, you can specify them as:

```
"$JAVA_HOME/bin/java" ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
jamagent.jamrun  jamconshost=myconsole01 jamisdaemon=true
jamjvmid=3001 com.example.MainClass
```

Refer to the Frequently Asked Questions appendix for more details on setting the `jamisdemon` parameter.

3.7.4 Adding the Oracle AD4J Agent and Libraries, Classes to Classpath

We also need to add the jamagent.zip directory to the CLASSPATH.

It is best to do this just before the call to java.

Here we find a couple of lines above the call to Java.

```
CLASSPATH="example1.jar:example2.jar
```

We can add the jamagent.zip file to the CLASSPATH by adding the following line before the original Java call.

```
CLASSPATH=${CLASSPATH} : /opt/jamagent/jamagent.zip
```

3.7.5 Target Parameters

About the -Dweblogic and other properties, you must insert `jamagent.jamrun` just before the calling program and its parameters. This should happen after all the JVM options and properties have been specified.

3.8 Installing the Database Agent

To track database bottlenecks to Java and Java bottlenecks into the database, you need to run a DB Agent on the DB Server and register the database with the console. This section outlines the steps to accomplish this and contains the following steps:

- Prerequisites
- Register the Database in the Setup Tab
- Download the DB Agent for Your Platform from the Console
- Download the Agent for Your Database Platform

3.8.1 Prerequisites

- Oracle AD4J Console needs to be installed, configured and accessible.
- You need to have the admin password on Oracle AD4J Console to register a new database
- You will then need the DB username and password on the target database with select privileges on `GV_$SESSION`, `GV_$SESSION_WAIT`, `GV_$PROCESS`, `GV_$SQLTEXT`, `GV_$SQLAREA`, `GV_$LOCK`, and `GV_$LATCHNAME` fixed views.
- You will also need to be able to run a program on the database server as the OS user who owns the database (in most cases this is the user name `oracle`)

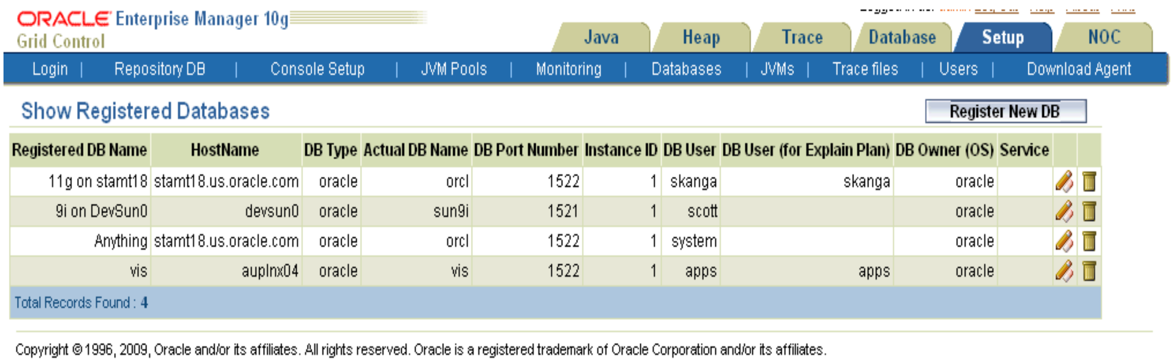
Note: In this release, databases running on Windows cannot be monitored.

3.8.2 Register the Database in the Setup Tab

Follow these steps to register the database:

1. Click the **Setup** tab.
2. The Setup options are displayed. Click **Databases**. The Show Registered Databases screen is displayed.

Figure 3–8 Show Registered Databases


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Database
Setup
NOC
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | Trace files | Users | Download Agent
Register New DB

Show Registered Databases

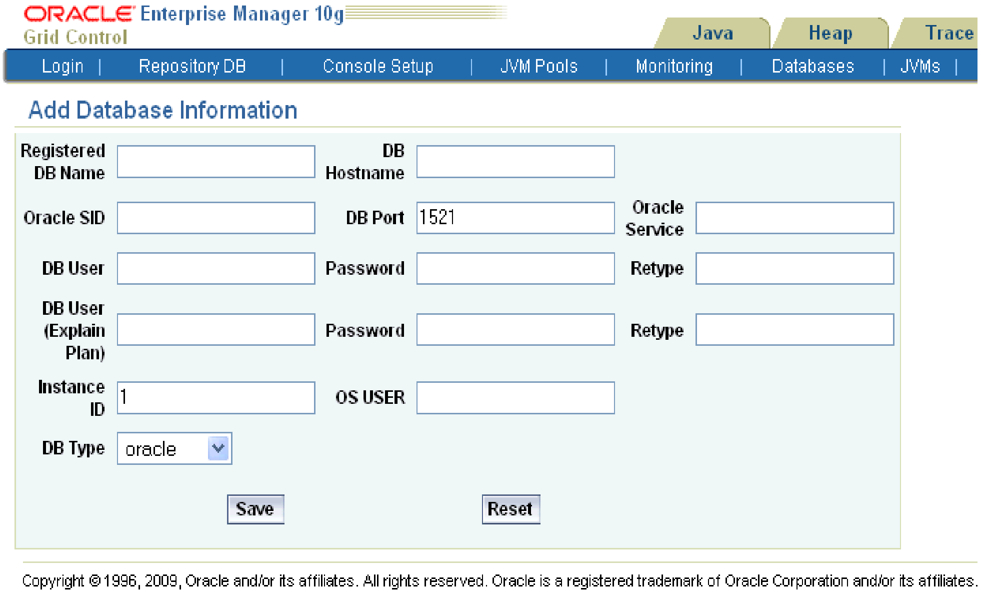
Registered DB Name	HostName	DB Type	Actual DB Name	DB Port Number	Instance ID	DB User	DB User (for Explain Plan)	DB Owner (OS)	Service		
11g on stamt18	stamt18.us.oracle.com	oracle	orcl	1522	1	skanga	skanga	oracle		✎	🗑
9i on DevSun0	devsun0	oracle	sun9i	1521	1	scott		oracle		✎	🗑
Anything	stamt18.us.oracle.com	oracle	orcl	1522	1	system		oracle		✎	🗑
vis	auplnx04	oracle	vis	1522	1	apps	apps	oracle		✎	🗑

Total Records Found : 4

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

3. Click **Register New DB**. The Add Database Information screen is displayed.

Figure 3–9 Add Database Information


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs |
Save
Reset

Add Database Information

Registered DB Name		DB Hostname			
Oracle SID		DB Port	1521	Oracle Service	
DB User		Password		Retype	
DB User (Explain Plan)		Password		Retype	
Instance ID	1	OS USER			
DB Type	oracle				

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

4. Enter the database information and click **Save** to register the new database.

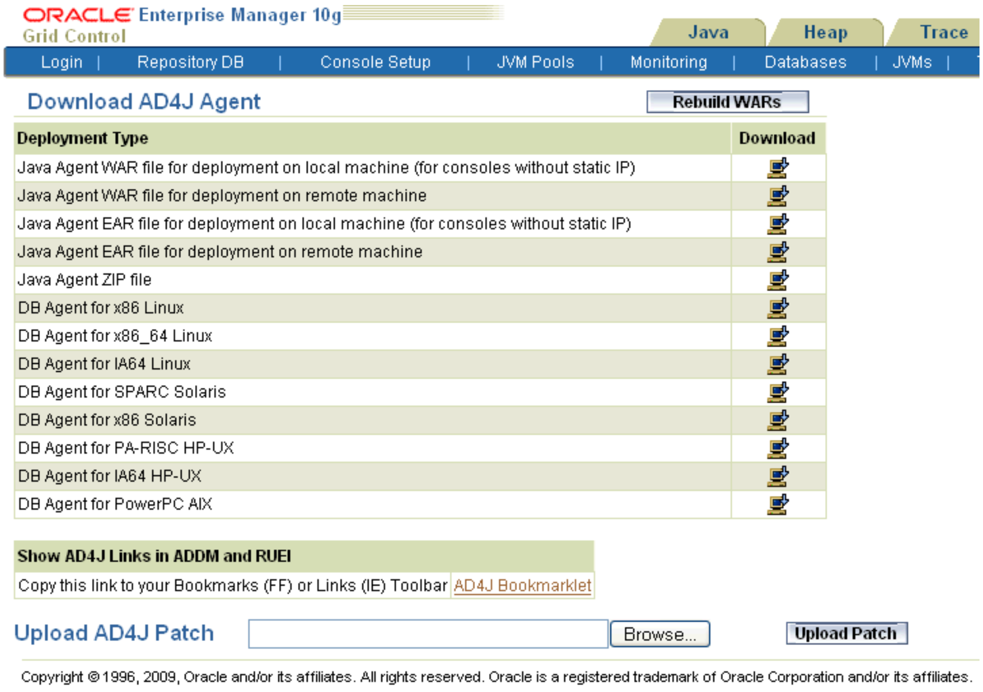
> **Note:** Before you can monitor a RAC database, you need to register all the nodes in the database and run the Agent on each node.

3.8.3 Download the DB Agent for Your Platform from the Console

Follow these steps to download the DB Agent for your platform:

1. Click the **Setup** tab.
2. The Setup options are displayed. Click **Download Agent**. The Download AD4J Agent is displayed.

Figure 3–10 Download AD4J Agent



Download AD4J Agent

Deployment Type	Download
Java Agent WAR file for deployment on local machine (for consoles without static IP)	⬇
Java Agent WAR file for deployment on remote machine	⬇
Java Agent EAR file for deployment on local machine (for consoles without static IP)	⬇
Java Agent EAR file for deployment on remote machine	⬇
Java Agent ZIP file	⬇
DB Agent for x86 Linux	⬇
DB Agent for x86_64 Linux	⬇
DB Agent for IA64 Linux	⬇
DB Agent for SPARC Solaris	⬇
DB Agent for x86 Solaris	⬇
DB Agent for PA-RISC HP-UX	⬇
DB Agent for IA64 HP-UX	⬇
DB Agent for PowerPC AIX	⬇

Show AD4J Links in ADDM and RUEI
Copy this link to your Bookmarks (FF) or Links (IE) Toolbar AD4J Bookmarklet

Upload AD4J Patch

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

3. Click on the Download icon to download the agent for your database platform.

3.8.4 Download the Agent for Your Database Platform

Depending on your method of upload to the database server, you might need to change the permissions of the agent.

```
chmod +x dbagent
```

Note: On some operating systems (such as AIX), due to system level restrictions, you must run the `dbagent` as the `root` user.

The dbagent needs two parameters to be passed to it, which are:

- the host where the console is running.
- the port which is being used to accept connections from the agents (default value 3600).

To pass these parameters, enter the following command:

```
nohup dbagent jamconshost=console01 jamconsport=3600 >
dbagent.log 2>&1 &
```

You should see the following message in the console error log.

```
JAM Console: Agent connection from 192.168.1.31:59269,
[Hostname] jamdb.us.oracle.com
```

```
JAM Console: Received AJDBOracle|Oracle-Sun9i|oracle|SunOS
5.8|oracle|3|0|0 Oracle AD4J Console: New DB 0
```

The number next to **New DB** changes everytime a new DB agent is pointed to the AD4J Console.

Note: If you want to stop the database agent, then manually kill the process. For UNIX platforms, run the `ps` command to view the database agent process and kill it using the `kill` command.

4

Setting Up and Using the AD4J Console

This chapter provides information about the administration tasks that you can perform with Oracle Application Diagnostics For Java (Oracle AD4J). In particular, the chapter covers the following:

- Oracle AD4J Console Setup
- Assumptions
- JVM Console Diagnostics
- Console Login Screen
- DB Connection Information
- Repository DB Page
- Console Setup Page
- JVM Pool Info
- Updating a JVM Pool
- Update Pool Thresholds
- Database Information
- Register New Database/Update Screen
- JVM Information Page
- Monitoring
- Manage Trace Files
- Manage Users
- Download Agent

Note: By default, Oracle AD4J Agent will write the following files to /tmp (C:\ on Windows):

1. Heapdump file

   Change for specific JVM from the Setup -> JVMs -> Edit JVM -> Heapdump Dir.

2. Extract and load native libraries

   Change the parameter jamlibdir in the file jamserv/jam/jamagent/WEB-INF and rebuild wars.

3. Log file (only on Windows)

   This is always written to C:\

4.1 Oracle AD4J Console Setup

If you have followed the default installation, the console should be accessible from the URL

`http://console01.mycompany.com:3500`

Where `console01.mycompany.com` is the name of the server you have installed the console on. You can also access the console using the IP address zzz.zzz.zzz.zzz for the server.

`http://zzz.zzz.zzz.zzz:3500`

> **Note:** The default port for Linux is 3500 and for Windows, it is 80. If you are using HTTPS, the default port number will be different.

4.2 Assumptions

This guide assumes the following:

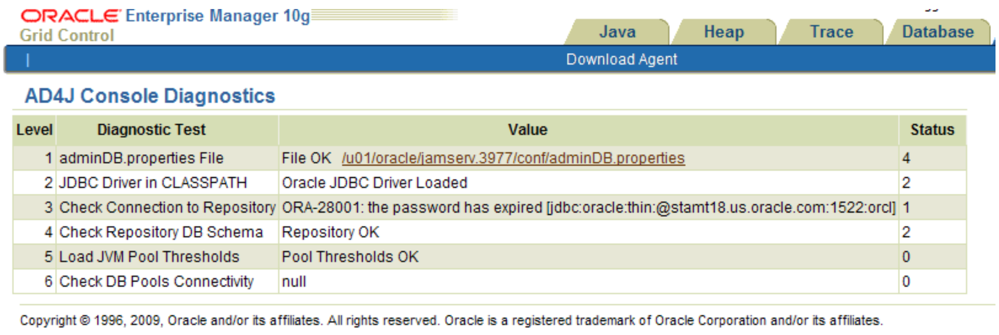
- Console is installed and running
- Console has been configured correctly with a repository database
- Database agents are installed and running
- Java agents are installed and running

For details on how to set up the console and agents, refer to the *Oracle AD4J Installation Notes*.

4.3 JVM Console Diagnostics

When you first access the console, it will take you to the **Setup** tab and run Diagnostics. Here, you can see the information about the configuration of the console and the errors (if any). For example, if a connection to the console repository cannot be established for any reason, it would show up here along with some debug information.

Figure 4–1 AD4J Console DIagnostics



AD4J Console Diagnostics

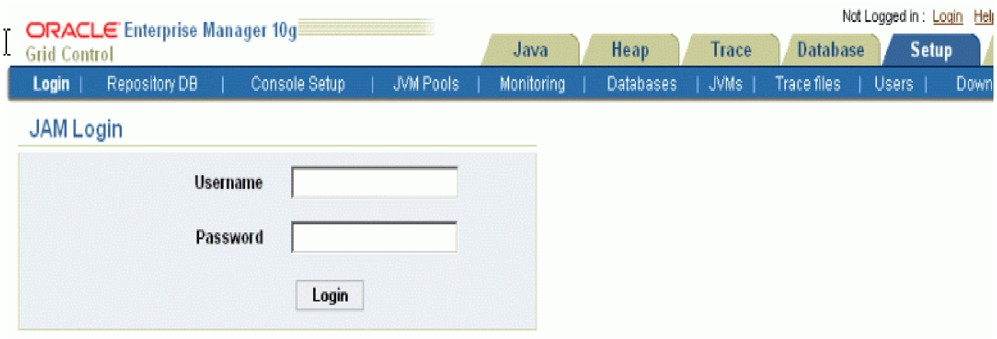
Level	Diagnostic Test	Value	Status
1	adminDB.properties File	File OK /u01/oracle/jamserv.3977/conf/adminDB.properties	4
2	JDBC Driver in CLASSPATH	Oracle JDBC Driver Loaded	2
3	Check Connection to Repository	ORA-28001: the password has expired [jdbc:oracle:thin:@stamt18.us.oracle.com:1522:orcl]	1
4	Check Repository DB Schema	Repository OK	2
5	Load JVM Pool Thresholds	Pool Thresholds OK	0
6	Check DB Pools Connectivity	null	0

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

4.4 Console Login Screen

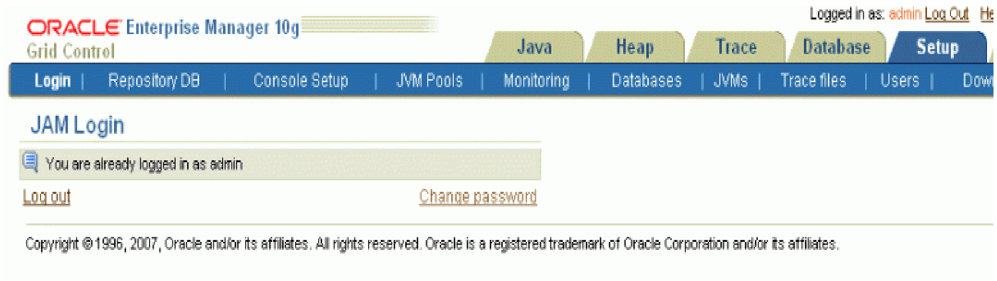
You are taken to the Login screen if you want to modify any admin information on the Oracle AD4J console. Login as the `administrator` to continue. The default administrator username and password is admin and welcome. You will need to change the password after your first login.

Figure 4–2 Console Login Screen


ORACLE Enterprise Manager 10g
Grid Control
Not Logged in : Login Hel
Java
Heap
Trace
Database
Setup
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | Trace files | Users | Down
JAM Login
Username
Password
Login

You get the following screen if you try logging in again before timeout.

Figure 4–3 Login Screen Before Timeout


ORACLE Enterprise Manager 10g
Grid Control
Logged in as: admin Log Out He
Java
Heap
Trace
Database
Setup
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | Trace files | Users | Dow
JAM Login
Log out
Change password

You are already logged in as admin

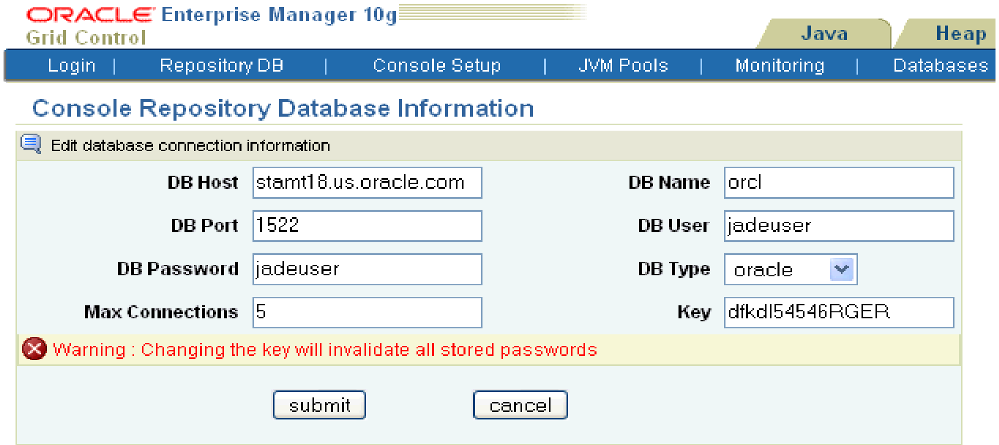
Copyright ©1996, 2007, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

4.5 DB Connection Information

The adminDB.properties file stores connection information about the repository database. The file resides in the location specified by the start up parameter jampropfilepath.

Click on the adminDB.properties link to create the file:

Figure 4–4 Console Repository Database Information



ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases

Console Repository Database Information

Edit database connection information

DB Host	stamt18.us.oracle.com	DB Name	orcl
DB Port	1522	DB User	jadeuser
DB Password	jadeuser	DB Type	oracle
Max Connections	5	Key	dfkdl54546RGER

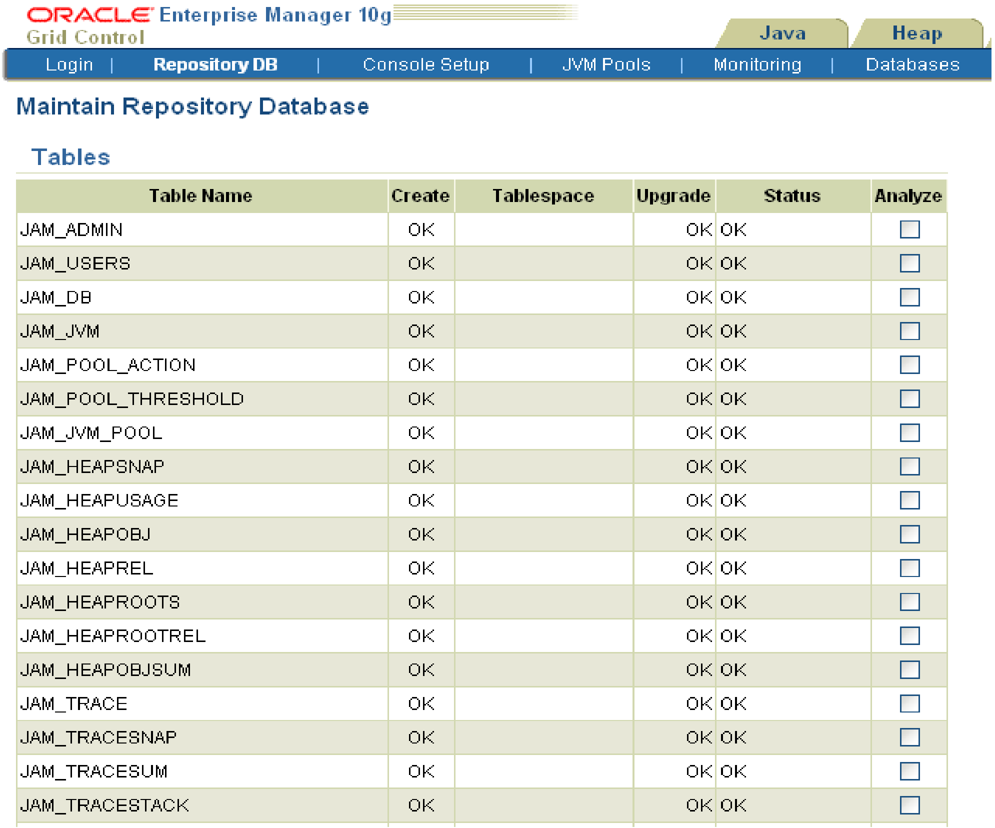
Warning : Changing the key will invalidate all stored passwords

Enter the fields with a valid DB Connection Information. Refer to the *Oracle AD4J Console Install Note* for details.

4.6 Repository DB Page

From this page you can create and/or view the tables and indexes in the database that are being used by the Oracle AD4J console.

Figure 4–5 Maintain Repository Database


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases

Maintain Repository Database

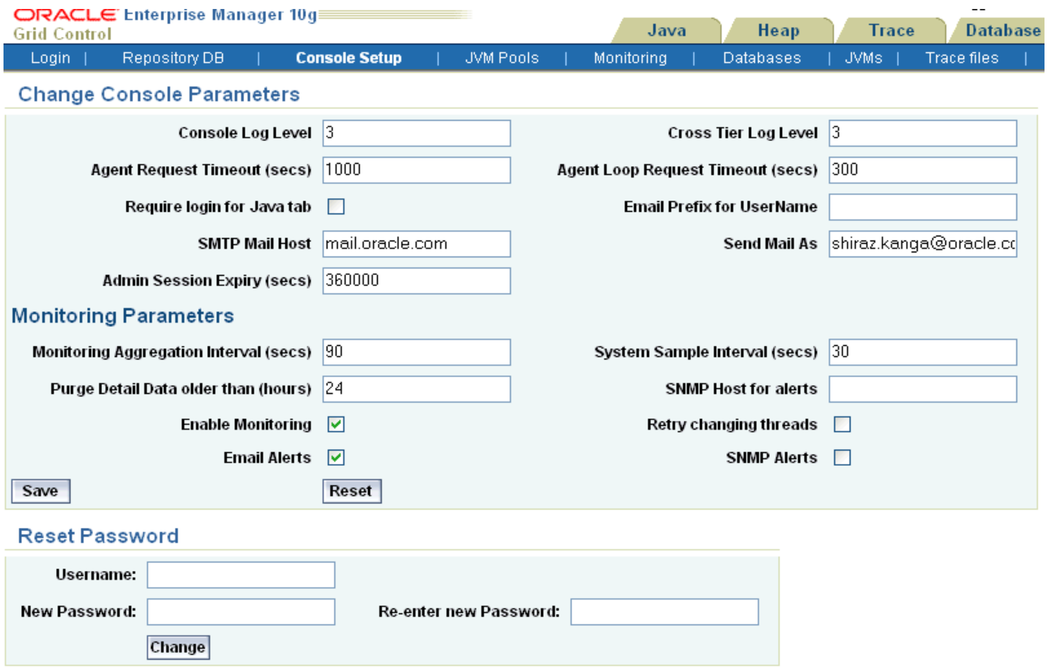
Tables

Table Name	Create	Tablespace	Upgrade	Status	Analyze
JAM_ADMIN	OK		OK	OK	□
JAM_USERS	OK		OK	OK	□
JAM_DB	OK		OK	OK	□
JAM_JVM	OK		OK	OK	□
JAM_POOL_ACTION	OK		OK	OK	□
JAM_POOL_THRESHOLD	OK		OK	OK	□
JAM_JVM_POOL	OK		OK	OK	□
JAM_HEAPSNAP	OK		OK	OK	□
JAM_HEAPUSAGE	OK		OK	OK	□
JAM_HEAPOBJ	OK		OK	OK	□
JAM_HEAPREL	OK		OK	OK	□
JAM_HEAPROOTS	OK		OK	OK	□
JAM_HEAPROOTREL	OK		OK	OK	□
JAM_HEAPOBJSUM	OK		OK	OK	□
JAM_TRACE	OK		OK	OK	□
JAM_TRACESNAP	OK		OK	OK	□
JAM_TRACESUM	OK		OK	OK	□
JAM_TRACESTACK	OK		OK	OK	□

4.7 Console Setup Page

This page is used to set console parameters and reset lost passwords.

Figure 4–6 Change Console Parameters


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Database
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | Trace files |
Save
Reset
Change

Change Console Parameters

Console Log Level	3	Cross Tier Log Level	3
Agent Request Timeout (secs)	1000	Agent Loop Request Timeout (secs)	300
Require login for Java tab	□	Email Prefix for UserName	
SMTP Mail Host	mail.oracle.com	Send Mail As	shiraz.kanga@oracle.cc
Admin Session Expiry (secs)	360000		

Monitoring Parameters

Monitoring Aggregation Interval (secs)	90	System Sample Interval (secs)	30
Purge Detail Data older than (hours)	24	SNMP Host for alerts	
Enable Monitoring	☑	Retry changing threads	□
Email Alerts	☑	SNMP Alerts	□

Reset Password

Username:			
New Password:		Re-enter new Password:	

Console Log Level, Cross Tier Log Level: Verbosity of console log messages. Increase only when advised by Oracle Support.

Agent Request Timeout, Agent Loop Request Timeout: Increase if the monitored JVMs are extremely busy and the console times out and disconnects waiting for their response.

Require login for Java tab: Enable this checkbox to restrict access to other (non-setup Tabs). Users will need to login with valid accounts (created from the Users screen) and will have access to all non-setup screens. The setup screens can only be accessed by the ADMIN user.

Enable Monitoring: With this checkbox enabled, JVMs belonging to selected pools will be polled for active requests periodically. This data is available on the NOC and in the last 90 seconds/24 hrs charts.

Email Alerts: Email alerts based upon threshold violations (requires monitoring to be enabled).

Email Prefix for UserName, SMTP Mail Host, Send Mail As: Parameters used for emailing alerts.

SNMP Alerts: Send SNMP traps for threshold violations (requires monitoring to be enabled).

SNMP Host for Alerts: SNMP Host to send alerts to.

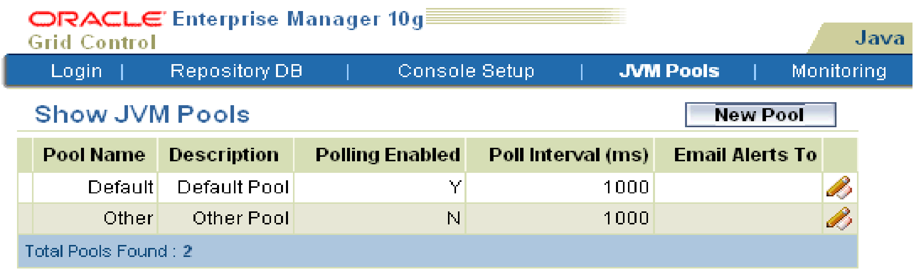
Reset Password Screen can be used to reset lost passwords and does not require old password.

4.8 JVM Pool Info

You can group sets of JVMs into JVM pools that provide monitoring information across all related JVMs in a single view. This table displays all the JVM pools in the system. You can view all available pools, delete existing pools or add new pools. The **Default** and **Other** pools cannot be deleted. Any user defined pools can be deleted.

Each row displays whether polling is currently enabled for the pool, the polling interval, and the email notification recipient. If the Polling Enabled flag is set to Y, JVMs belonging to this pool will be polled for active requests periodically based on the Poll Interval. You can click on the **Edit** icon to edit the JVM pool or click on **New Pool** to add a new JVM Pool.

Figure 4–7 Show JVM Pools


ORACLE Enterprise Manager 10g
Grid Control
Java
Login | Repository DB | Console Setup | JVM Pools | Monitoring
New Pool

Show JVM Pools

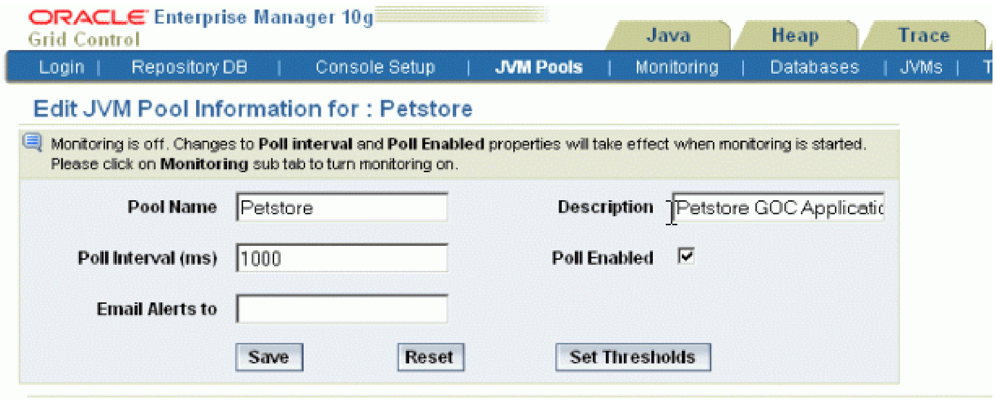
Pool Name	Description	Polling Enabled	Poll Interval (ms)	Email Alerts To	
Default	Default Pool	Y	1000		✎
Other	Other Pool	N	1000		✎

Total Pools Found : 2

4.9 Updating a JVM Pool

To update a JVM Pool, click the **Edit** icon in the Show JVM Pools page. The Edit JVM Pool page is displayed:

Figure 4–8 Edit JVM Pool Information


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | T
Save
Reset
Set Thresholds

Edit JVM Pool Information for : Petstore

Monitoring is off. Changes to **Poll interval** and **Poll Enabled** properties will take effect when monitoring is started. Please click on **Monitoring** sub tab to turn monitoring on.

Pool Name	Petstore	Description	Petstore GOC Applicatic
Poll Interval (ms)	1000	Poll Enabled	☑
Email Alerts to			

Poll interval (ms): Sample interval for JVMs belonging to this pool when monitoring (polling) is enabled.

Poll Enabled: Enable monitoring for this pool.

Email Alerts to: Email Alias/list to email this pool's alerts to.

4.10 Update Pool Thresholds

Threshold violations can have a **Level** of **R** (Red) or **Y** (Yellow). The **Metric** is the attribute being monitored and the **Threshold** is the value against which that metric is being compared. A violation occurs when the threshold is exceeded after a minimum number of samples have been monitored. Threshold units are % CPU Utilization for Machine CPU (CPU), JVM CPU (CPU), % Heap Utilization for Heap, number of threads in that state for others. 3 to 5 threads active in DB for more than 50% of the samples will cause an amber (Y) alert for DB, while 6+ threads active in DB wait state for more than 50% of the samples will cause a red alert for DB.

Action URL is the URL to be invoked when a threshold violation occurs. This includes internal URLs into the AD4J console and external URLs. The Action URL can be any valid URL on a remote system. It can also accept URLs on the local AD4J Console. It can be used to trace a particular thread, all active threads, or dump a heap in response to a threshold violation.

The Action URL should be a valid URL as called from a browser. If not specified, default parameters for the `traceThread` and `heapdump` will be added to the URL. If the Action URL specifies a host that is different from the current console host, you must navigate to the Console Setup page and uncheck the **Require Login for Java Tab** checkbox.

A sample URL template for Tracing Threads is given below:

http://localhost/jvmTraceActive.jsp?traceThread=allactive&JVM_THREAD_ID=&JVM_ID=1&JVM_MACHINE=machine_name&JVM_PORT=8080&pollInterval=50&pollDuration=10&samplesfilename=traceactive&detailsfile=on

where:

- `traceThread` is the name of the thread to be traced. If you want to trace all the active threads, specify the value as `traceThread=allactive`. This is the value in the Thread column in the View All Threads page.
- `JVM_THREAD_ID` is the ID of the thread to be traced. If you want to trace all active threads or if `traceThread=allactive`, the `JVM_THREAD_ID` parameter must be blank. This is the value specified in the `OS PID` column in the View All Threads page.
- `JVM_ID`, `JVM_MACHINE`, `JVM_PORT` are the ID, machine name, and port number of the JVM Agent on which the thread or threads to be trace are running. For example, if the text displayed at the top of the View All Threads page is - `Name: machine_name:8080 ID: 1: Linux-2.6.9-55.0.0.0.2.EL: Sun Microsystems Inc.: 1.6.0_07`
  - `machine_name` is the `JVM_MACHINE`
  - `8080` is the `JVM_PORT`
  - `1` is the `JVM_ID` (If the `JVM_ID` parameter is not supplied in the URL, it is automatically appended by the Console)
- `PollInterval` is the time interval (in milliseconds) between successive samples. The default value is 50.
- `PollDuration` is the duration (in seconds) for the trace. The default value is 10.
- `Samplesfilename` is the prefix of the trace sample file name created on the console machine. The default value is `traceactive` and an unique identifier is added as a suffix to each trace file that is created.

- `Detailsfile` is the sample thread stack in addition to the state. You must set this to `On` if you require complete thread details. The default is On.

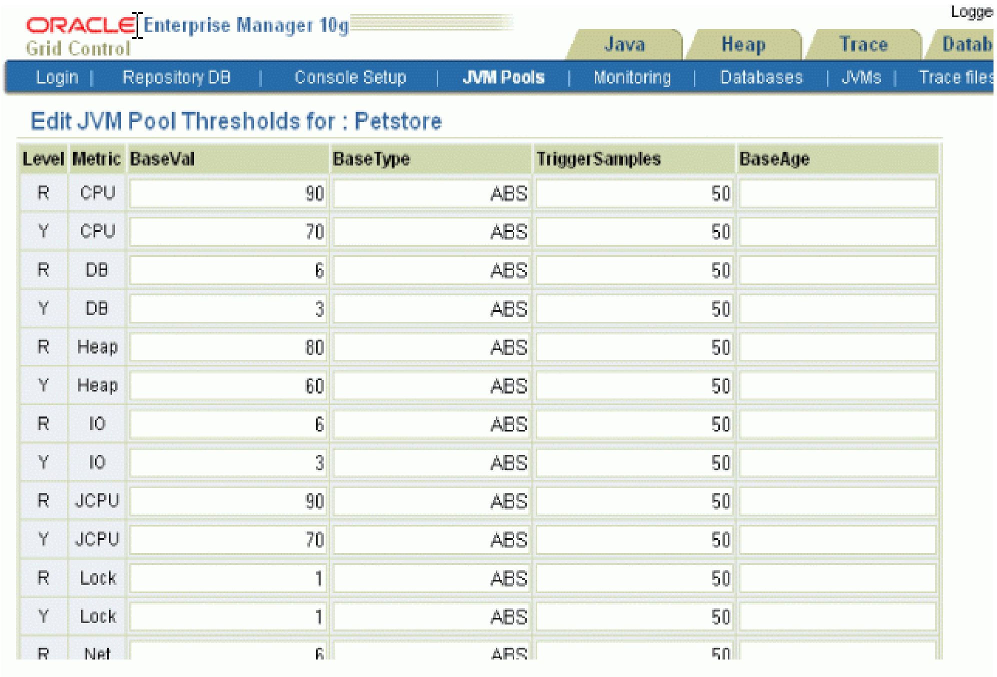
A sample URL for heapdump is given below:

http://localhost/jvmHeapDump.jsp?JVM_ID=1&JVM_MACHINE=machine_name&JVM_PORT=8080&JVM_DUMP_FILE=heapdump <http://localhost/jvmHeapDump.jsp?JVM_ID=1&JVM_MACHINE=machine_name&JVM_PORT=8080&JVM_DUMP_FILE=heapdump>

The following parameters are required in the URL for **heapdump**.

- `JVM_ID`, `JVM_MACHINE`, and `JVM_PORT` are the same parameters as specified for the Tracing Threads template.
- `JVM_DUMP_FILE` is the prefix of the heap dump file name on the agent. An unique identifier is added as a suffix to each heap dump file.

Figure 4–9 Edit JVM Pool Thresholds


ORACLE Enterprise Manager 10g
Grid Control
Logge
Java
Heap
Trace
Datab
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | Trace files

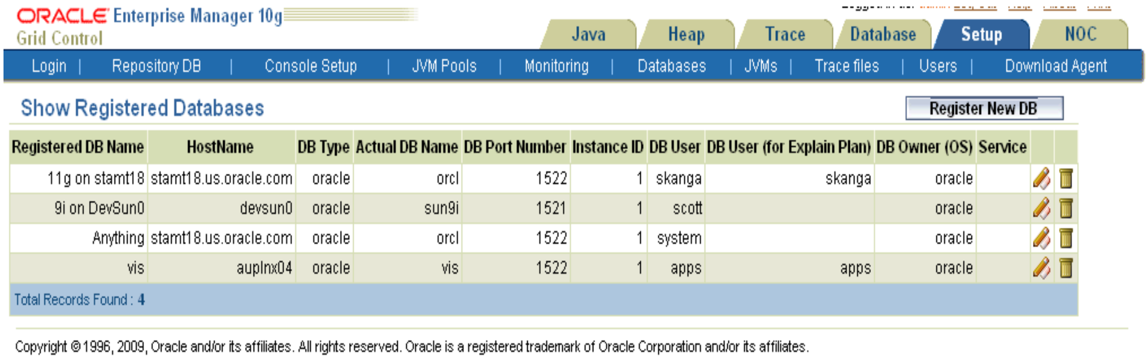
Edit JVM Pool Thresholds for : Petstore

Level	Metric	BaseVal	BaseType	TriggerSamples	BaseAge
R	CPU	90	ABS	50	
Y	CPU	70	ABS	50	
R	DB	6	ABS	50	
Y	DB	3	ABS	50	
R	Heap	80	ABS	50	
Y	Heap	60	ABS	50	
R	IO	6	ABS	50	
Y	IO	3	ABS	50	
R	JCPU	90	ABS	50	
Y	JCPU	70	ABS	50	
R	Lock	1	ABS	50	
Y	Lock	1	ABS	50	
R	Net	6	ABS	50	

4.11 Database Information

Use this screen to view information about the databases being monitored/profiled currently or to add additional databases for the same.

Figure 4–10 Show Registered Databases


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Database
Setup
NOC
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | Trace files | Users | Download Agent
Register New DB

Show Registered Databases

Registered DB Name	HostName	DB Type	Actual DB Name	DB Port Number	Instance ID	DB User	DB User (for Explain Plan)	DB Owner (OS)	Service	
11g on stamt18	stamt18.us.oracle.com	oracle	orcl	1522	1	skanga	skanga	oracle		✎ 🗑
9i on DevSun0	devsun0	oracle	sun9i	1521	1	scott		oracle		✎ 🗑
Anything	stamt18.us.oracle.com	oracle	orcl	1522	1	system		oracle		✎ 🗑
vis	auplnx04	oracle	vis	1522	1	apps	apps	oracle		✎ 🗑

Total Records Found : 4

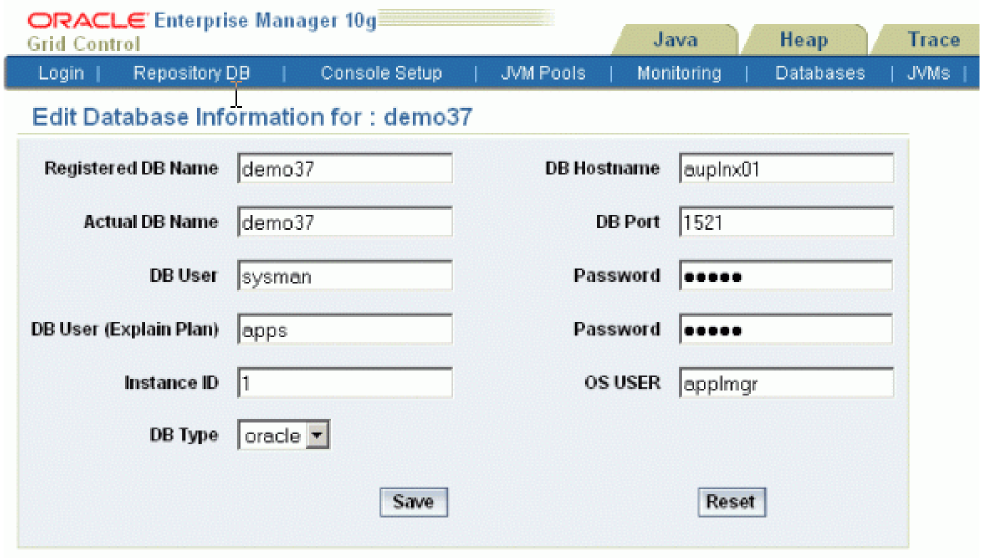
Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Click the Edit icon to edit an existing database or click **Register NewDB** to register a new database.

4.12 Register New Database/Update Screen

Enter the database information in the New

Figure 4–11 Register New Database


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs |
Save
Reset

Edit Database Information for : demo37

Registered DB Name	demo37	DB Hostname	auplnx01
Actual DB Name	demo37	DB Port	1521
DB User	sysman	Password	●●●●●
DB User (Explain Plan)	apps	Password	●●●●●
Instance ID	1	OS USER	applmgr
DB Type	oracle		

Registered DB Name: Provide a user friendly name for Oracle AD4J to display.

DB Hostname: Database server for this monitored instance.

Actual DB Name: Actual Oracle SID for the monitored DB.

DB Port: Oracle Net Listener Port on the DB Server.

DB User: DB User to use for querying database information.

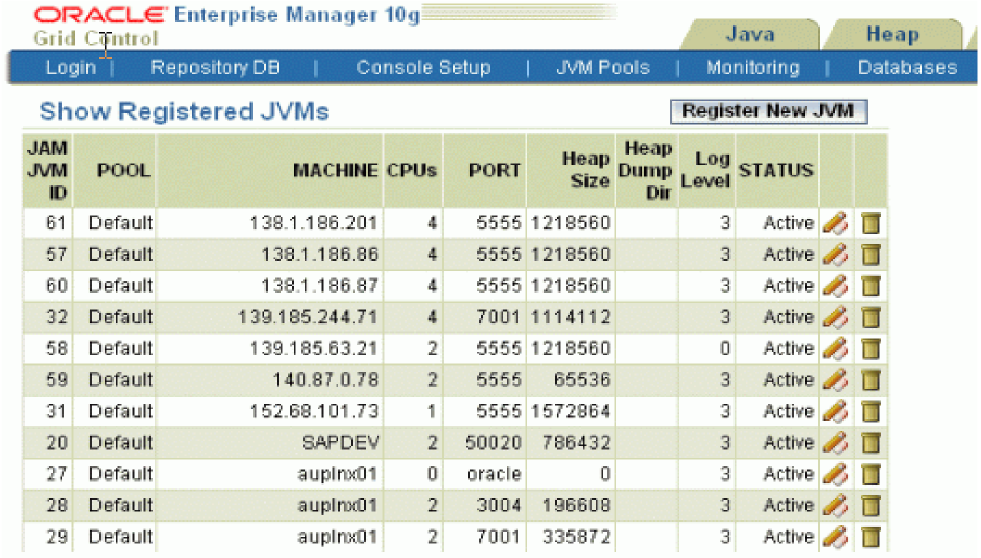
Password: (only in update/new mode) : The password for the DB User.

DB User (Explain Plan): Application DB User account to use for explain plan analysis.

DB OS User: Operating System User which owns the database.

4.13 JVM Information Page

Figure 4–12 Show Registered JVMs

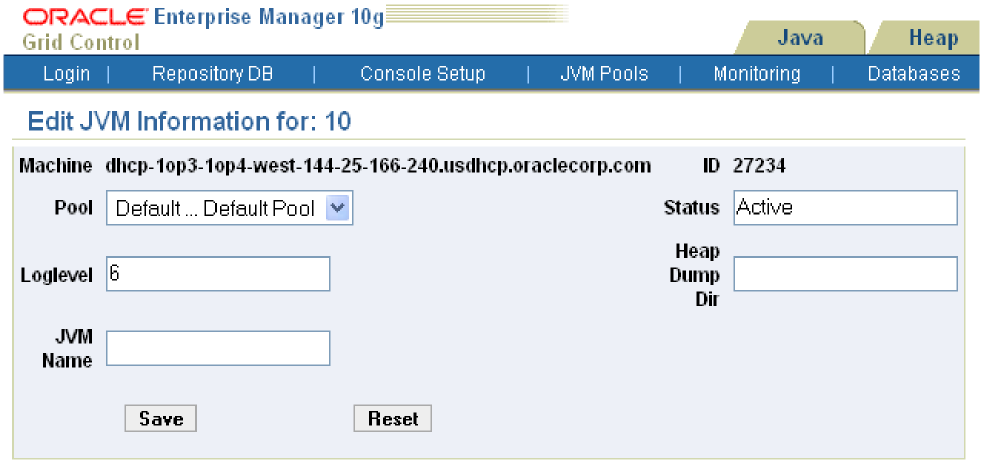

ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases
Register New JVM

Show Registered JVMs

JAM JVM ID	POOL	MACHINE	CPUs	PORT	Heap Size	Heap Dump Dir	Log Level	STATUS		
61	Default	138.1.186.201	4	5555	1218560		3	Active	✎	🗑
57	Default	138.1.186.86	4	5555	1218560		3	Active	✎	🗑
60	Default	138.1.186.87	4	5555	1218560		3	Active	✎	🗑
32	Default	139.185.244.71	4	7001	1114112		3	Active	✎	🗑
58	Default	139.185.63.21	2	5555	1218560		0	Active	✎	🗑
59	Default	140.87.0.78	2	5555	65536		3	Active	✎	🗑
31	Default	152.68.101.73	1	5555	1572864		3	Active	✎	🗑
20	Default	SAPDEV	2	50020	786432		3	Active	✎	🗑
27	Default	auplnx01	0	oracle	0		3	Active	✎	🗑
28	Default	auplnx01	2	3004	196608		3	Active	✎	🗑
29	Default	auplnx01	2	7001	335872		3	Active	✎	🗑

Lists all the JVMs that have connected to the console. You can edit the JVM parameters using the edit icon, or delete old JVMs that no longer connect to the console.

Figure 4–13 Edit JVM Information


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases
Save
Reset

Edit JVM Information for: 10

Machine dhcp-1op3-1op4-west-144-25-166-240.usdhcp.oraclecorp.com

ID 27234

Pool: Default ... Default Pool

Status: Active

Loglevel: 6

Heap Dump Dir:

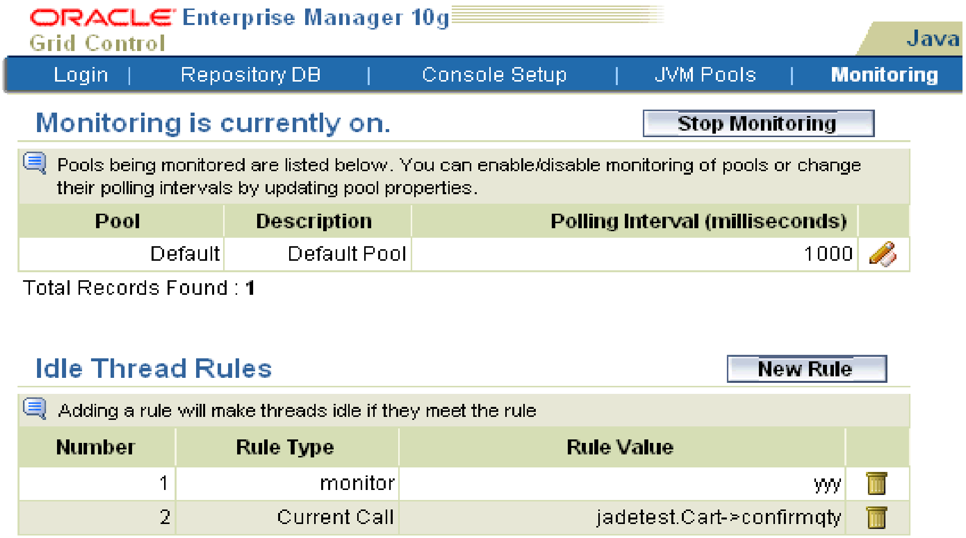
JVM Name:

You can assign a JVM to a different pool, change it's log level (if recommended by Support) or change the temporary directory where the heap snapshots are dumped.

4.14 Monitoring

Once you have updated a JVM pool with polling enabled, you can monitor the JVMs in that pool.

Figure 4–14 Monitoring Status Page


ORACLE Enterprise Manager 10g
Grid Control
Java
Login | Repository DB | Console Setup | JVM Pools | Monitoring
Stop Monitoring
New Rule

Monitoring is currently on.

Pools being monitored are listed below. You can enable/disable monitoring of pools or change their polling intervals by updating pool properties.

Pool	Description	Polling Interval (milliseconds)	
Default	Default Pool	1000	✎

Total Records Found : **1**

Idle Thread Rules

Adding a rule will make threads idle if they meet the rule

Number	Rule Type	Rule Value	
1	monitor	yyy	🗑
2	Current Call	jadetest.Cart->confirmqty	🗑

Click on the **Start Monitoring** link above to start monitoring. This can also be done from the Console Setup Screen. You can enable or disable monitoring of pools or change their polling intervals by updating the pool properties.

You can mark a thread as idle by adding it to an Idle Thread Rule. All threads that have been marked as idle will not be monitored. Click **New Rule** to create a new Idle Thread Rule. The Rule Type can be:

- Monitor (Waiting on Lock): Select this type if you want to ignore all threads that are locked.
- Current Call: Select this type if you want to ignore all threads that are making a call to the selected function.

The Rule Value should contain the class name, method, followed by class+method. An example of a Current Call is `weblogic.socket.PosixSocketMuxer->processSockets`. An example of a Monitor (Waiting on Lock) is `weblogic.socket.PosixSocketMuxer$1`.

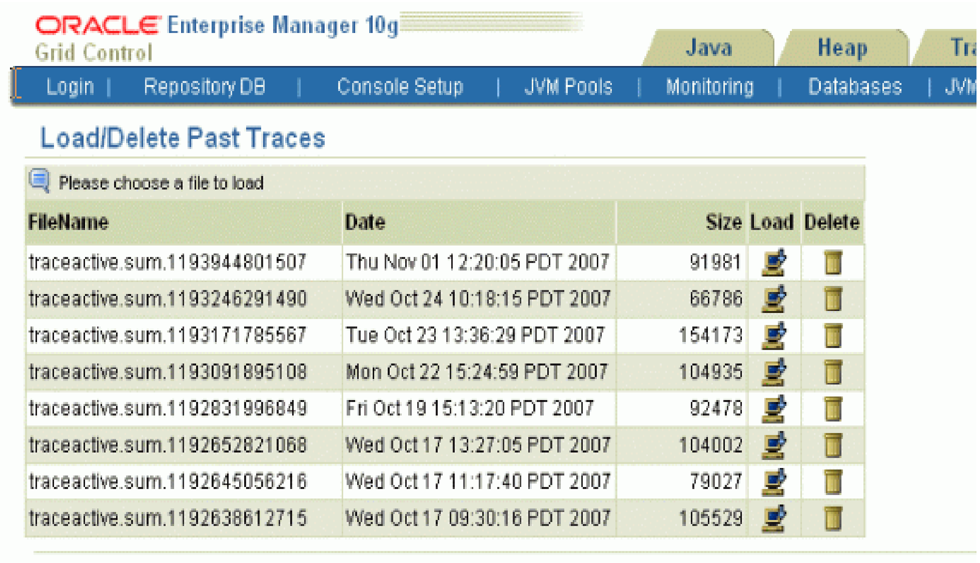
All threads that meet the criteria specified in the Idle Thread Rule will not appear in the View Active Threads screen.

4.15 Manage Trace Files

Use this screen to:

- Delete trace files from the traces you have run.
- Load a trace file from another console. You'll need to move the trace file(s) to the `jamserv/trace` directory for them to visible in this screen.

Figure 4–15 Manage Trace Files


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Tra
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVM

Load/Delete Past Traces

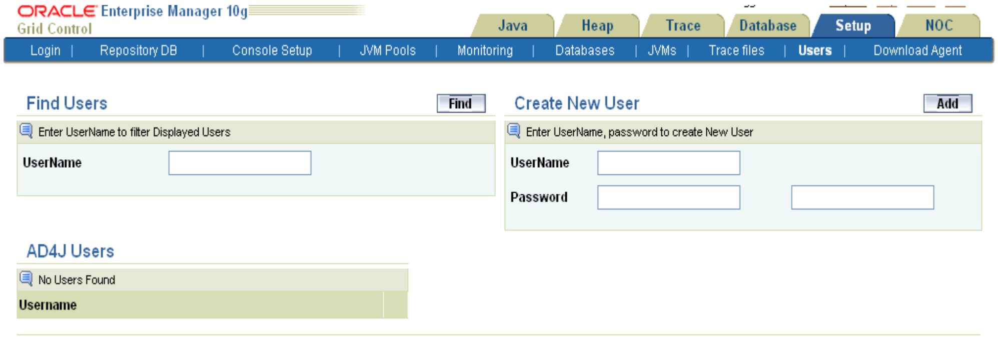
Please choose a file to load

FileName	Date	Size	Load	Delete
traceactive.sum.1193944801507	Thu Nov 01 12:20:05 PDT 2007	91981	✎	🗑
traceactive.sum.1193246291490	Wed Oct 24 10:18:15 PDT 2007	66786	✎	🗑
traceactive.sum.1193171785567	Tue Oct 23 13:36:29 PDT 2007	154173	✎	🗑
traceactive.sum.1193091895108	Mon Oct 22 15:24:59 PDT 2007	104935	✎	🗑
traceactive.sum.1192831996849	Fri Oct 19 15:13:20 PDT 2007	92478	✎	🗑
traceactive.sum.1192652821068	Wed Oct 17 13:27:05 PDT 2007	104002	✎	🗑
traceactive.sum.1192645056216	Wed Oct 17 11:17:40 PDT 2007	79027	✎	🗑
traceactive.sum.1192638612715	Wed Oct 17 09:30:16 PDT 2007	105529	✎	🗑

4.16 Manage Users

From this screen, you can create non-admin user accounts for accessing the Oracle AD4J console.

Figure 4–16 Manage Users


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Database
Setup
NOC
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs | Trace files | Users | Download Agent
Find
Add

Find Users

Enter UserName to filter Displayed Users

UserName

Create New User

Enter UserName, password to create New User

UserName

Password

AD4J Users

No Users Found

Username

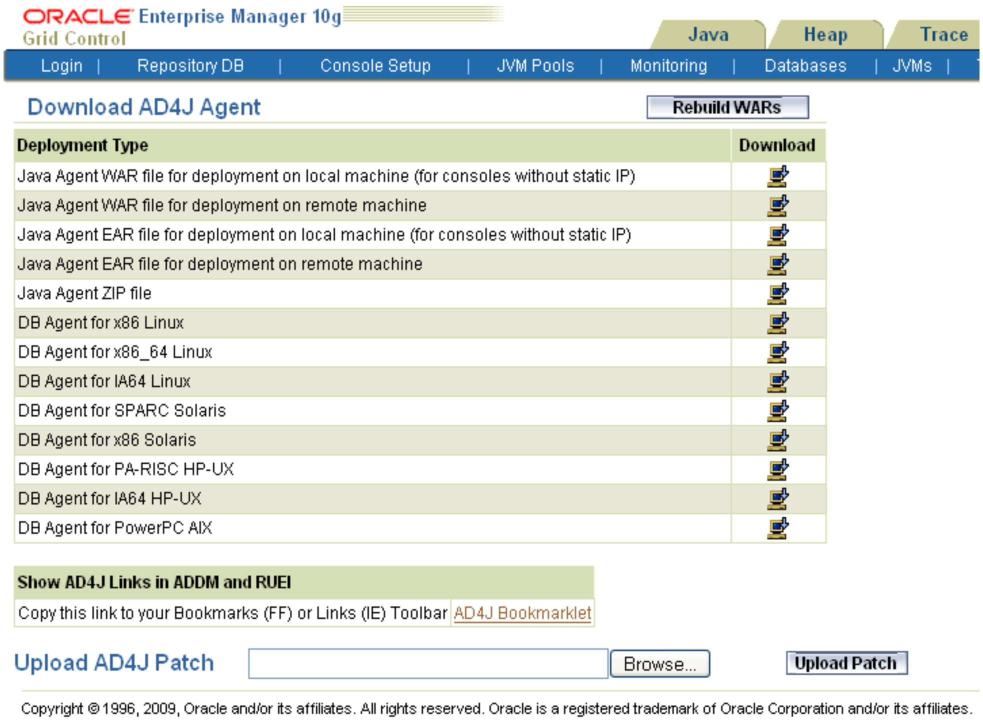
4.17 Download Agent

Use this screen to download Java and database agents for deployment on remotely monitored JVMs or databases. Depending on the format supported by your application server, or if it is remote or local (on the same host as the console), you can select a `war` or `ear` file for the Java agent.

If you are running a standalone Java application then you can select the zip file. For database agents, you can select the correct one for the platform on which you are running the Oracle database to be monitored. The Java agents contain the IP address of the console. If your console IP address changes, you need to click on **Rebuild WARs**, download a new agent and deploy it.

Note: The rest of this screen should be used only with instructions from Oracle support.

Figure 4–17 Download Agent


ORACLE Enterprise Manager 10g
Grid Control
Java
Heap
Trace
Login | Repository DB | Console Setup | JVM Pools | Monitoring | Databases | JVMs |
Rebuild WARs
Browse...
Upload Patch

Download AD4J Agent

Deployment Type	Download
Java Agent WAR file for deployment on local machine (for consoles without static IP)	⬇
Java Agent WAR file for deployment on remote machine	⬇
Java Agent EAR file for deployment on local machine (for consoles without static IP)	⬇
Java Agent EAR file for deployment on remote machine	⬇
Java Agent ZIP file	⬇
DB Agent for x86 Linux	⬇
DB Agent for x86_64 Linux	⬇
DB Agent for IA64 Linux	⬇
DB Agent for SPARC Solaris	⬇
DB Agent for x86 Solaris	⬇
DB Agent for PA-RISC HP-UX	⬇
DB Agent for IA64 HP-UX	⬇
DB Agent for PowerPC AIX	⬇

Show AD4J Links in ADDM and RUEI	
Copy this link to your Bookmarks (FF) or Links (IE) Toolbar	AD4J Bookmarklet

Upload AD4J Patch

Copyright © 1996, 2009, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

A

Frequently Asked Questions

This chapter lists some of the frequently asked questions encountered while using Oracle AD4J.

A.1 Frequently Asked Questions

Some of the frequently asked questions include:

- What is the meaning of "Cost"?
- In my console why do I see "No Data Found" in the Java tab?
- How do I navigate the NOC tab to get to a specific date or time period?
- Why are the User/Request/IP Addr fields empty in the thread view?
- How much repository space do I require for monitoring and for heap dumps?
- How do I install the AD4J Database Agent on Windows?
- Can I still use AD4J without a database agent?
- How can I check if my JVM is supported by AD4J?
- Where does the Agent Store Its Log Files?
- How do I Prevent the Agent from Terminating Program before Thread Execution is Completed?

A.1.1 What is the Meaning of Cost?

Cost represents the number of times the agent has seen a given method across all sampled threads. When the polling interval is set to 1 second, this represents approximately the total number of seconds spent in this method across all active threads seen during the total time.

A.1.2 In My Console, Why Do I See *No Data Found* in the Java Tab?

- You will see No Data Found in the Java tab on the console if monitoring has not been enabled for the pool currently being viewed. To enable monitoring, do the following:

  1. Click the **Setup** tab. Then click **Console Setup** and **Enable Monitoring**.
  2. Click the **Setup** tab. Then click **Monitoring** and **Start Monitoring**.
  3. Click the **Setup** tab. Then click **JVM Pools** and click on the **Edit** icon. Check the **Poll Enabled** checkbox.

- You may also see this error if your browser cache is corrupted. You must ensure that the browser cache is emptied before retrying this operation.

A.1.3 How Do I View a Specific Period in the NOC View?

The three arrow icons on the top left of the NOC view can be used to navigate to different time periods of monitoring. The arrow pointing up is used to zoom out and increase the amount of time being viewed while the left and right arrows allow you to navigate back and forth across different time periods.

A.1.4 Why are the User, Request, and IP Address fields Empty in the Thread View?

Some application servers protect this data and hide it so that is not available to the AD4J Agent. This will cause the fields to appear blank. Additionally, these fields can also appear blank on an application server that is not fully supported by AD4J.

A.1.5 How Much Repository Space is Required for Monitoring and for Heap Dumps

In a standard monitoring setup, at least 25 - 100 MB is required initially with a default 24 hour purge interval. You must check the tablespace growth periodically and if required, you may need to change the space requirements. This will ensure that database growth due to standard monitoring will occur smoothly without sudden spikes.

However, there are 2 things that can drastically affect your tablespace:

- **Heap Dumps**: Analyzing heaps requires a large amount of tablespace. As a standard practice, we recommend that you must have 5 times the size of heap dump file being loaded in your tablespace. Since you know the size of your dump file, make sure that there is adequate space to accommodate the dump file before it is loaded into the database.
- **Thread Traces**: While these are smaller than heaps. they are loaded into the database automatically when a user initiates a trace at the console. The size of these threads can vary dramatically depending on the number of active threads during the trace, the duration of the trace, and the sample interval of the trace. This should usually be under 100MB but if several thread traces have been initiated, it could fill up the database quickly. Before initiating the traces, you must ensure that there is adequate space in the database.

A.1.6 How Do I Install the AD4J Database Agent on Windows?

In this release, the AD4J Database Agent cannot be installed on Windows.

A.1.7 Can I Still Use AD4J Without a Database Agent?

Most AD4J functions for monitoring JVMs and databases will work properly without a database agent. But if you encounter a thread that is in the DB Wait state, it will not be a hyperlink. Therefore, you cannot follow the thread into the database and see its database session information including the SQL that it executes. Since the only function of the database agent is to provide cross-tier correlation, the information in the database tab will not be impacted.

A.1.8 How Can I Check if my JVM is Supported by AD4J?

The supported JVMs are listed in the README.TXT file available at `http://www.oracle.com/technology/software/products/oem/htdocs/jade.html`.

A.1.9 Where does the Agent Store Its Log Files?

All agent logging is done to the STDERR stream which the Application Server redirects to a file. Refer to the *Application Server* documentation for details on the location of the file in which the STDERR messages are logged.

A.1.10 How do I Prevent Agent from Terminating Program before Thread Execution is Completed?

While monitoring a standalone program with the AD4J agent, the program may get terminated before all the threads have been executed. To prevent this from happening, you need to add `jamisdaemon` parameter and set the value as `jamisdaemon=true`. This parameter wil prevent the agent from exiting before all the threads have been completely executed.