ORACLE ENTERPRISE MANAGER GRID CONTROL – COMPOSITE APPLICATION MONITOR AND MODELER (CAMM) V10.2.0.4 DEPLOYMENT GUIDE

The specifications and information in this manual are subject to change without notice. All statements and recommendations in this manual are believed to be accurate but are presented without warranty of any kind, express or implied. Users must take full responsibility for their application of any products.

1 Overview

1.1 Introduction

The introduction covers pre-deployment, deployment and post-deployment procedures for the Oracle Enterprise Manager Grid Control – Composite Application Monitor and Modeler (CAMM) product. It also provides background on the functionality of CAMM 10.2.0.4, which shall substantially minimize the effort required to successfully deploy CAMM 10.2.0.4 in production and staging environments.

Chapter 2 “CAMM 10.2.0.4 Operation and Environment Dependencies” summarizes the environment requirements that must be met in order to guarantee the successful deployment and functioning of CAMM 10.2.0.4. It also covers CAMM 10.2.0.4 operation modes and provides the reasoning behind running CAMM 10.2.0.4 in each of its supported modes, allowing you to choose the mode best suited to your needs.

Chapter 3 “Deployment Process” elaborates on pre-deployment, deployment and post-deployment procedures. A description of the installation process for Windows and UNIX environments is covered in this chapter.

Chapter 4 “Data Export” describes procedures to export performance data in persistence format.

1.2 Audience

The *CAMM 10.2.0.4 Deployment Guide* will be primarily used by the personnel who are responsible for the deployment of CAMM 10.2.0.4 in staging and production environments. These could be administrators, operation support specialists, architects or similar. The reader is not expected to have programming skills but is expected to know the basics of systems administration.

2 CAMM 10.2.0.4 Operation and Environment Dependencies

The installation of CAMM 10.2.0.4 involves a few simple manual steps and the configuration of supporting subsystems including an RDBMS-based CAMM Data Repository.

In order to guarantee the successful deployment and operation of CAMM 10.2.0.4 the reader is expected to:

- Understand CAMM operation modes
- Comply with the pre-deployment and post-deployment requirements
- Understand or have access to someone who understands basic database management in order to setup the repository database

It is important to choose the proper CAMM Operation mode and comply with the environment requirements before starting the actual deployment.

2.1 CAMM Operation Modes

CAMM 10.2.0.4 can operate in two modes: Service Mode and Application Mode. This section provides some insight into these modes of operation.

2.1.1 Service Mode

CAMM will typically run as a headless Java process that monitors your Oracle WebLogic, Oracle SOA Suite, and/or IBM WebSphere environments. In the service mode, monitoring of your applications continues in the background, even when the user interface is not present. The user interface is delivered as a Java applet in a web browser. Deployment of the acseraadmin.war web application in a standard J2EE web container (such as Apache Tomcat) is required to provide a URL for browser access.

The CAMM and acseraadmin.war container processes should be configured to start automatically with the operating system.

The following picture visualizes the **Service Mode CAMM 10.2.0.4** topology:

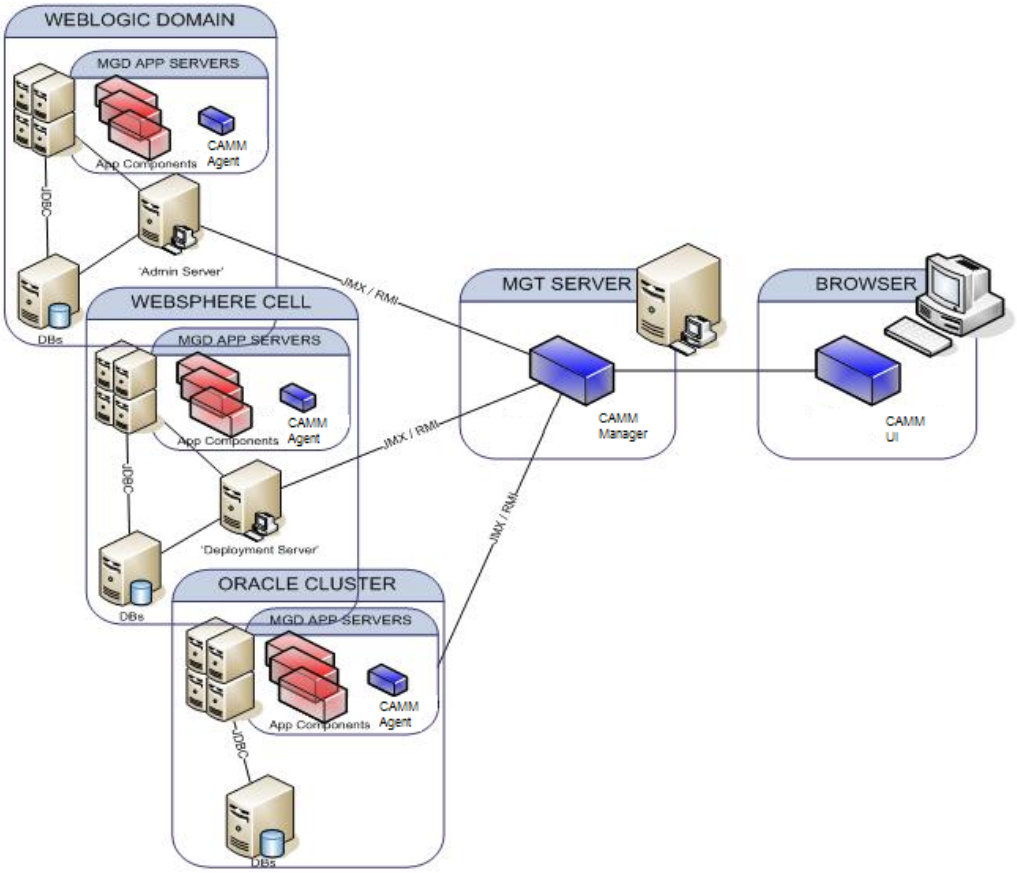


Figure 1. Service Mode CAMM 10.2.0.4 topology

2.1.2 Standalone Application Mode

In the Standalone Application Mode, CAMM runs as a GUI based application rather than a UNIX daemon or Windows service. In contrast to the Service Mode, this mode provides a single-user GUI and does not require a separate web container to host the web application for browser GUI delivery. When the user starts the GUI application, CAMM starts, and when the application is closed, CAMM discontinues operation. In this mode, monitoring and data collection continue only when the application is running. It is useful for occasional debugging and testing of configuration and general connectivity to systems in your environment. However, in most environments where continuous monitoring is required it is advised to run CAMM in Service Mode.

The following figure shows an example of CAMM 10.2.0.4 deployed in **Application Mode** monitoring an Oracle WebLogic and IBM WebSphere application server environment:

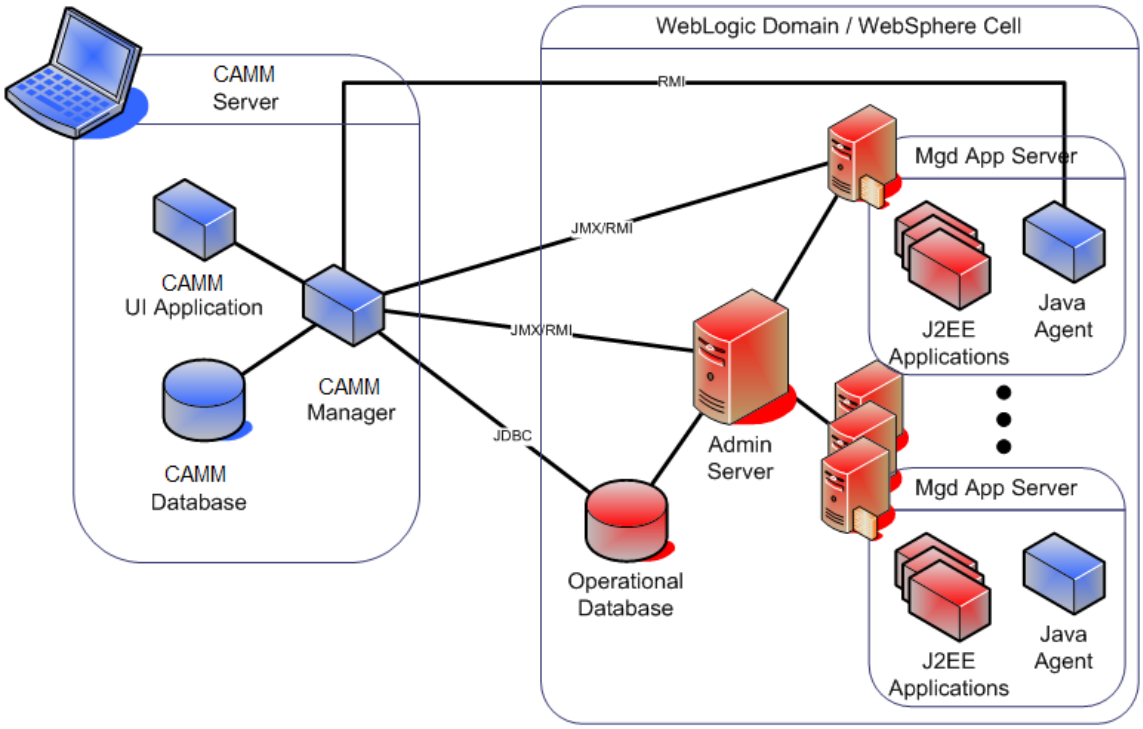


Figure 2. Application Mode CAMM 10.2.0.4 topology

3 Deployment Process

3.1 Pre-deployment

Check compliance with the dependencies (Example of environment dependencies shown below for WebLogic 8.1, 9.1, 9.2, and 10.0)

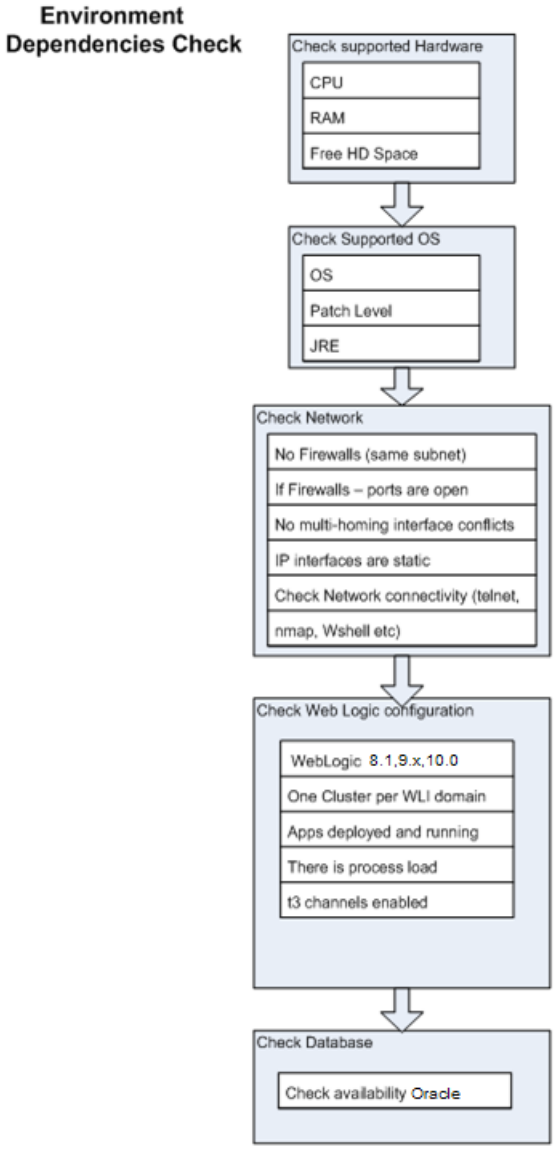


Figure 3. Environment Dependencies Check

3.1.1 Environment Considerations

3.1.1.1 Access and Connectivity

Access to Monitored Application Platform

With respect to the target application platform to be monitored, CAMM requires the following level of access:

- System Access to the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere Application domain environment (e.g. ability to deploy to the domain)
- System Admin capabilities in order to modify WebLogic/WebSphere startup files

Note that CAMM accesses the target application platform as application users created with administrative privileges. Creating an operating system user (e.g. a UNIX user) is not necessary other than securing the CAMM installation and configurations.

The Oracle SOA Suite\Oracle WebLogic\IBM WebSphere Administration domain server and the Managed Servers must be accessible from the machine where CAMM is being installed.

Network Connectivity

Firewalls provide security for networks and applications by preventing certain connectivity between machines, servers, and applications. CAMM is remotely monitoring performance and availability of the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere runtime environment. If firewalls or proxies are present in the network topology they must be configured to allow communication traffic between CAMM and the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere runtime environment.

It is recommended that the CAMM dedicated machine be deployed into the same subnet, within firewall perimeter.

If a firewall is unavoidable ensure that the CAMM port is open and t3, FTP and RMI traffic allowed.

DHCP

It is recommended that the CAMM dedicated machine, the J2EE Application Admin Server, and Managed Servers are assigned static IP addresses. Non-static configurations are more difficult to configure.

3.1.1.2 Databases

CAMM Database

CAMM stores application metadata, performance and availability metrics in a repository database. This dedicated database must be accessible by CAMM. Currently, Oracle CAMM supports Oracle 10g and MySQL database platforms for the repository. For an Oracle or MySQL database installation, a dedicated user database and schema should be created with adequate access rights and connectivity restrictions. Setup and configuration for the MySQL database are covered in chapter 3.2.5.2.

Application Platform Databases
Examples of an application platform's databases include the WebLogic Portal database which stores streaming portal information. Application platform databases are used by CAMM as a source of metadata during the one-time modeling phase of the Manager's lifecycle. This allows CAMM to dynamically discover the structure of the monitored applications. In general, CAMM's monitoring technology provides maximum visibility without the need for querying against an application platform's databases at runtime.

WebLogic Integration (WLI) Database
All WebLogic Integration runtime configurations consist of an RDBMS configuration for tracking system level events, process state, and runtime metadata. CAMM will capture this environment metadata from the WebLogic Integration runtime database as part of its comprehensive analysis of the WebLogic runtime environment.

However, at runtime, CAMM can reliably monitor the WLI application without querying against this datastore for performance metrics. For this reason, the WLI feature of process tracking can be set to "None" for maximum runtime performance of the WLI applications.

WebLogic Portal (WLP)/WebSphere Portal Database/Oracle SOA Suite Dehydration Store
The Oracle SOA Suite/Oracle WebLogic/IBM WebSphere runtime configuration consists of an RDBMS configuration for tracking system level events, application events, and runtime metadata. CAMM will capture this environment metadata from the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere runtime database as part of its comprehensive analysis of the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere runtime environment.

3.1.1.3 System Locale

CAMM supports 4 locale: English (en, US) , Spanish (es, ES) , Chinese Simplified (zh ,CN), and Chinese Traditional (zh,TW). By default, the system locale will be picked.

For running standalone.bat/sh, acsera.bat/sh or client.bat/sh:
You can do following at command line before running the batch:

1. Spanish:

   ```
   Set / export ACSERA_LANG=es
   Set / export ACSREA_COUNTRY=ES
   ```

2. Chinese Simplified:

   ```
   Set / export  ACSERA_LANG=zh
   Set / export ACSREA_COUNTRY=CN
   ```

3. Chinese Traditional:

   ```
   Set / export  ACSERA_LANG=zh
   Set / export  ACSREA_COUNTRY=TW
   ```

For running applet from browser: you need to create environment variable since the command line var is not picked.

1. Spanish:

   ```
   ACSERA_LANG=es
   ACSREA_COUNTRY=ES
   ```

2. Chinese Simplified:

   ```
   ACSERA_LANG=zh
   ACSREA_COUNTRY=CN
   ```

3. Chinese Traditional:

   ```
   ACSERA_LANG=zh
   ACSREA_COUNTRY=TW
   ```

Steps:

1. Open *Control Panel*
2. Click the *System* icon and the window pops up
3. Go to the *Advanced* pane
4. Click the *Environment Variables* button

Note: Once you define the environment variable, you don't need to set the variable at command line for standalone or client or acsera batch files.

3.1.2 Clustering and Application Domains

A single instance of CAMM can be used to monitor multiple application "domains". An application domain is a logical/administrative context for a collection of resources, such as a WebLogic cluster (or WebSphere cell) and/or standalone server instances. A single Manager instance can also monitor mixed application domains of different vendor platforms. Thus, with a single Manager instance, a human operator can have a single, consistent view into a large, heterogeneous environment where, for example, WebLogic domains and WebSphere cells co-exist.

Because of this flexibility, a single instance of CAMM is best dedicated to a single "environment" – as in production, or QA. Within each environment, application server

platforms may be heterogeneous, based on different vendors (WebLogic/WebSphere) or different versions (WLS 9.2 vs. 10.0), and have diverse deployment configurations (e.g. standalone servers, Oracle SOA Suite, WLS clusters and/or WebSphere cells).

Mixing "environments" within a single CAMM instance (such as domains from production and from QA) is also technically feasible but not recommended, since traffic patterns from the different environments tend to be different (live vs. load testing) and can complicate advanced data analyses. In this case deploying multiple instances of CAMM would be the right solution, where a dedicated CAMM instance monitors each specific environment (Production, Staging, QA etc.). You can find details about multiple CAMM instances support in Chapter 3.2.5.3.

3.1.2.1 WebLogic Clustering

Clusters

CAMM supports the monitoring of performance and availability across any number of servers, clusters, and machine configurations across multiple WebLogic domains. It will automatically detect application clustered deployments and dependencies and build metrics that relate to this deployment architecture. There are no limitations for the number of domains, clusters, machines, and servers a single CAMM instance can monitor.

Node Manager

Node Manager is used by the WebLogic Admin server to remotely start and stop WebLogic Managed Server instances. The CAMM Agent Deployment makes some changes to startup parameters of a Managed Server. When using Node Manager, the CAMM Agent Deployment will detect that Node Manager is running and make those changes to the Node Manager server JVM startup parameters.

3.1.2.2 WebLogic Integration (WLI)

Clusters

Oracle requires that a WebLogic Integration domain consist of only **ONE** WebLogic Integration cluster within a WebLogic Integration domain. This means that any other cluster defined in the WebLogic Integration domain shall not have **integration** elements or depend on the WebLogic Integration cluster.

Check WebLogic Platform Environment configured correctly

Are clustered resources configured according to load balancing and failover requirements?

Table	Description
JMS Destinations	WLI and application destination are configured as distributed destinations. And are working correctly.

Event Generators	Are deployed and targeted correctly in the cluster
Connectors	Adapters, Application Views, Connectors are targeted correctly to the cluster (*this will be defined in the adapter documentation*)
Domain Configuration	There should only be one cluster configured for any WLI domain

For more information on WLI cluster configurations:
http://e-docs.bea.com/wli/docs81/deploy/cluster.html

3.2 Deployment

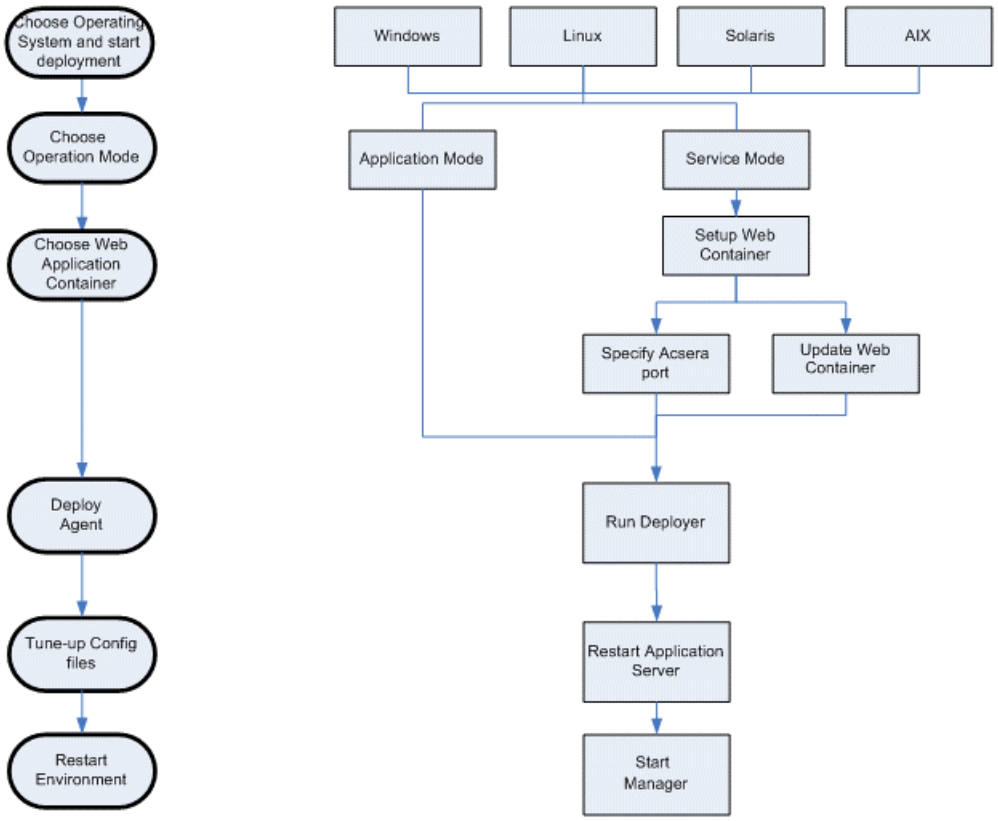


Figure 4. Deployment Workflow

When installing CAMM 10.2.0.4 you are given 3 options:

CAMM 10.2.0.4 (Tomcat J2EE container included): This option is available only on Windows and will install two Windows Services: the CAMM service and the web

container service. As Windows Services, CAMM and the Administration Server (the Tomcat web container) can be controlled via the standard Windows Services administration console (accessible via Settings/Control Panel/Administrative Tools) and be configured to start automatically when the host restarts. The use of this installation option is deprecated and not recommended.

CAMM 10.2.0.4 only: choosing this option will install CAMM 10.2.0.4 as a Windows service without installing a separate web container. On UNIX systems, startup and shutdown scripts are provided. The CAMM manager contains an in-process servlet container (Tomcat) that will host the web-based applet console. This built-in container is enabled by default and its use is highly recommended to simplify the number of running processes.

CAMM 10.2.0.4 as an Application: choosing this option will install CAMM 10.2.0.4 as a stand-alone application. A UI console window will pop up when the CAMM manager is started. Closing this window will shut down the CAMM manager. Typically, this mode is chosen if the installation is on a machine which will monitor Application servers on an ad-hoc basis, for example, as a consultant. Often this mode is chosen to install on a laptop computer.

On the Windows platform, choosing this installation option does not preclude running CAMM as a Windows Service at a later time. A Windows batch file, `createmanagerservice.bat` is included so that the CAMM can be installed as a Windows Service later.

3.2.1 Install CAMM 10.2.0.4 – Windows installation

Insert CAMM 10.2.0.4 CD-ROM into your CD-ROM drive. The installation will automatically start.

At any time during the installation, you can click the **Cancel** button to cancel the installation, or click the **Previous** button to go back a screen.

The Introduction screen will be shown.

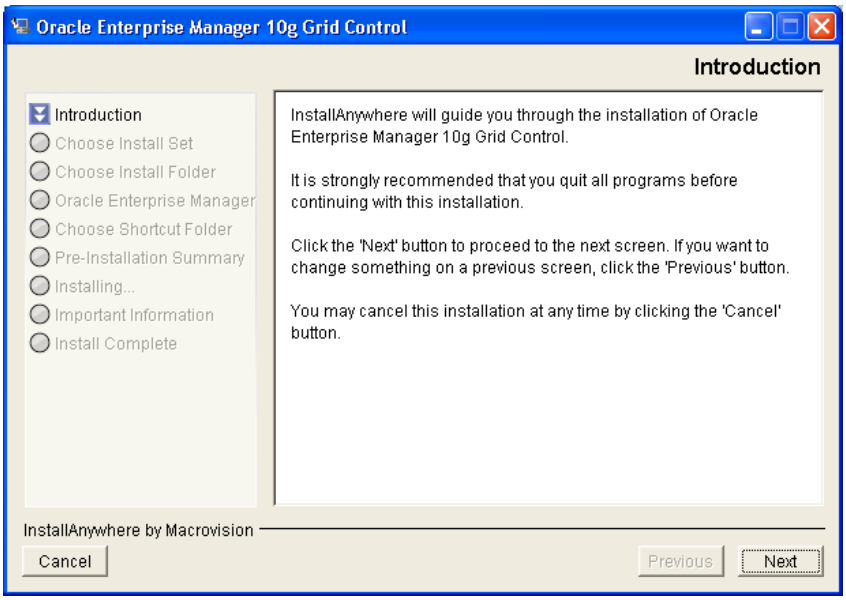


Figure 5. Installation Introduction Screen

Read the introduction and click the **Next** button. The License Agreement screen will be shown.

You will be asked which mode you wish to use CAMM 10.2.0.4 in.

Figure 6. Installation Choose Install Set Screen

You must decide if you want to install CAMM 10.2.0.4 as a Service or as an Application, and if as a Service, if you want to use the included J2EE web application container, or your own. See the **CAMM Operation Modes Section** for more details.

Choose the selection most appropriate for your environment, and click the **Next** button.

You will be asked to choose a folder to install CAMM 10.2.0.4 into.

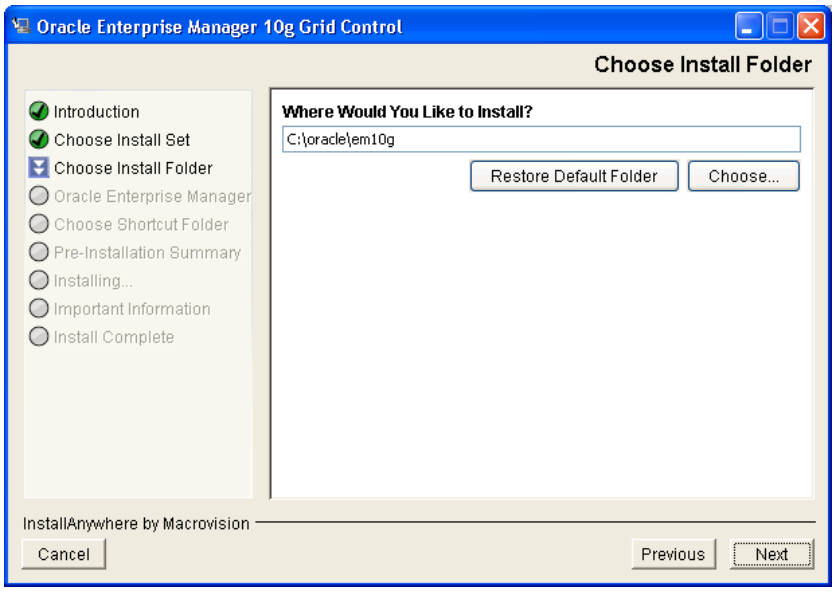


Figure 7. Installation Choose Install Folder Screen

Choose a folder, or accept the default C:\CAMM folder. Click the **Next** button.

If you chose to install CAMM 10.2.0.4 as a service and install its own web container, you will be asked the port to use for the administration web server. If you chose to use your own web container, you will not be asked this, and will need to know the correct port. This does not apply if you chose to install CAMM 10.2.0.4 as an Application.

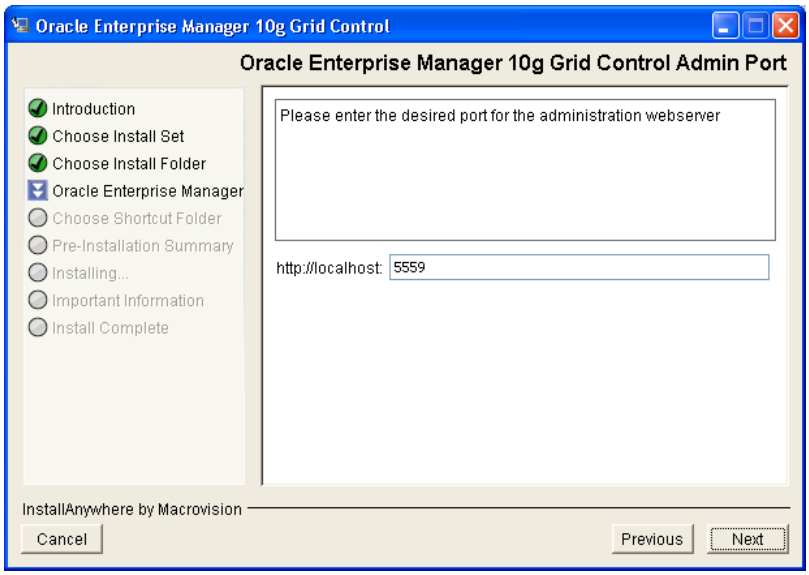


Figure 8. Installation Specify CAMM Administration Port Screen

Select a port to use to access CAMM 10.2.0.4 or accept the default 5557. Click the **Next** button.

You will be asked to choose a Shortcut Folder to create CAMM 10.2.0.4 product icons in.

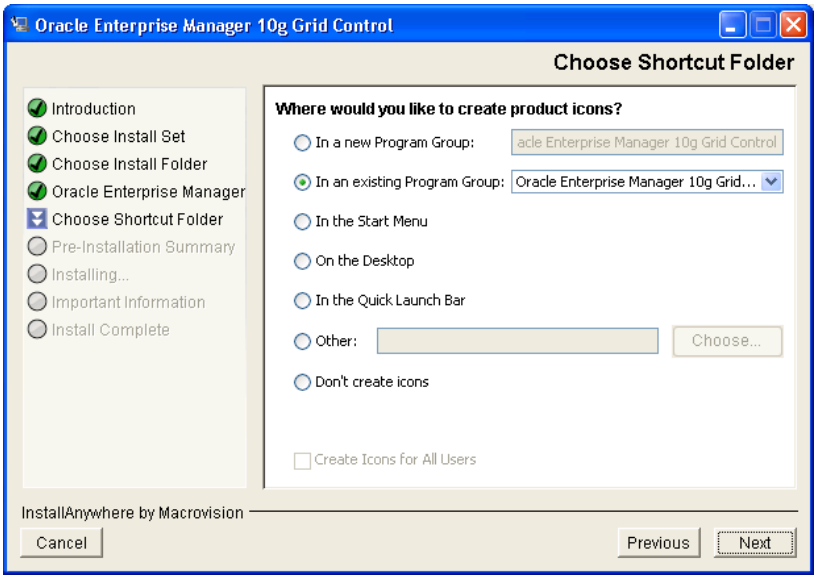


Figure 9. Installation Choose Shortcut Folder Screen

Choose a Shortcut folder, or accept the default new Program Group CAMM. If desired, select the **Create Icons for All Users** box. Click the **Next** button.

You will be shown a summary of what is about to be installed.

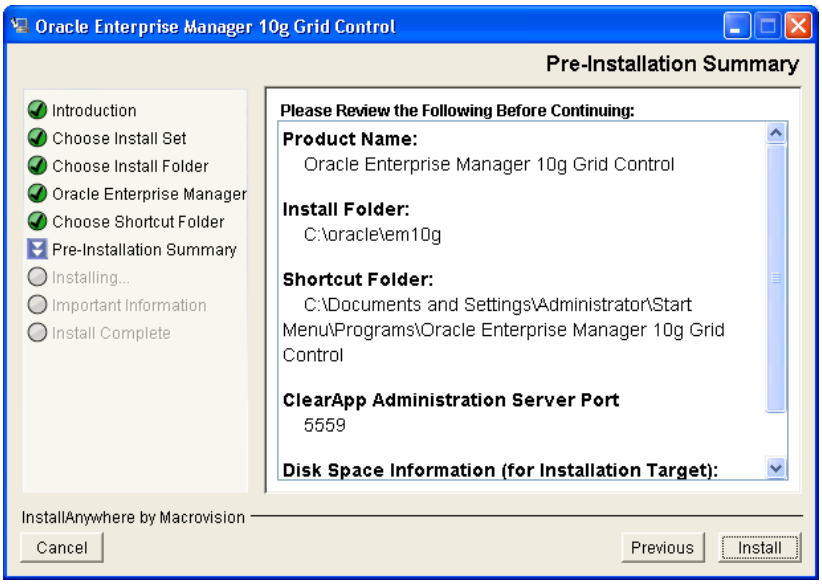


Figure 10. Pre-Installation Summary Screen

If something is not correct, press the **Previous** button to move back and correct it. When the summary is correct, click the **Next** button.

CAMM 10.2.0.4 will begin installing.

Figure 11. Installation Complete Screen

When CAMM 10.2.0.4 installation is complete, you may be asked to reboot your system. (In most cases, you won't be).

If you are asked to restart your system, you may have the installation restart it for you by selecting the **Yes, restart my system** option. Or you may choose to restart it later by selecting the **No, I will restart my system myself**. After making your selection, click the **Done** button.

3.2.2 Install CAMM 10.2.0.4 – Solaris, Linux and AIX installation

Insert CAMM 10.2.0.4 CD-ROM into your CD-ROM drive. Mount your CD-ROM if necessary.

Change directory to where the CD-ROM is mounted.

Change directory to the **Disk1/InstData/Linux/VM** directory and execute the *install.bin* program for Linux:

```
% cd Disk1/InstData/Linux/VM
% ./install.bin
```

For Solaris change directory to the **Disk1/InstData/Solaris/VM** directory and execute the *install.bin* program:

```
% cd Disk1/InstData/Solaris/VM
% ./install.bin
```

For AIX change directory to the **Disk1/InstData/AIX/VM** directory and execute the install.bin program:

```
% cd Disk1/InstData/AIX/VM
% ./install.bin
```

Note: make sure that *install.bin* has execution privileges and if necessary use *chmod a+x install.bin* command to make it executable

The Introduction screen will be shown.

```
===============================================================================
Introduction
------------

InstallAnywhere will guide you through the installation of CAMM 10.2.0.4.

It is strongly recommended that you quit all programs before continuing
with
this installation.

Respond to each prompt to proceed to the next step in the installation.  If
you
want to change something on a previous step, type 'back'.
```

```
You may cancel this installation at any time by typing 'quit'.

PRESS <ENTER> TO CONTINUE:
```

As noted, you can type ‘back’ at any time to go back, and may cancel the installation by typing ‘quit’. Press the <ENTER> key to continue. The License Agreement will be displayed:

```
================================================================================
==
License Agreement
-----------------

Installation and use of CAMM 10.2.0.4 requires acceptance of the following
License Agreement:
      <License>
IF YOU AGREE TO THE FOREGOING TERMS AND CONDITIONS AND DESIRE TO INSTALL
AND
USE THE SOFTWARE, PLEASE RESPOND ‘Y’ TO THE QUESTION BELOW. OTHERWISE,
PLEASE RESPOND ‘N’ AND THE INSTALLATION PROCESS WILL STOP.

DO YOU ACCEPT THE TERMS OF THIS LICENSE AGREEMENT? (Y/N):
```

Read the License Agreement, type ‘Y’ indicating you accept the terms, and press the <ENTER> key.

You will be asked which mode you wish to use CAMM 10.2.0.4 in.

```
================================================================================
==
Choose Install Set
------------------

Please choose the Install Set to be installed by this installer.

  ->1- CAMM 10.2.0.4 only
    2- CAMM 10.2.0.4 as Application

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE
DEFAULT
   :
```

You must decide if you with to install CAMM 10.2.0.4 as a Service or as an Application. See the **CAMM Operation Modes** section of this Guide for more information. Choose whether to install as a service (option 1) or as an application, then press the <ENTER> key. You will be asked where to install:

```
================================================================================
==
```

```
Choose Install Folder
---------------------

Where would you like to install?

  Default Install Folder: /root/CAMM

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
      :
```

Choose the directory you'd like to install CAMM 10.2.0.4 into and press the <ENTER> key. You'll be asked where to install CAMM 10.2.0.4 product icons.

```
===============================================================================
==
Choose Link Location
--------------------

Where would you like to create links?

  ->1- Default: /root
    2- In your home folder
    3- Choose another location...

    4- Don't create links

ENTER THE NUMBER OF AN OPTION ABOVE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
      :
```

Choose the directory you'd like and press the <ENTER> key. You'll be shown a summary of your installation choices:

```
===============================================================================
==
Pre-Installation Summary
------------------------

Please Review the Following Before Continuing:

Product Name:
    CAMM 10.2.0.4

Install Folder:
    /root/oracle/CAMM

Link Folder:
    /root

Disk Space Information (for Installation Target):
    Required:  145,432,962 bytes
    Available: 17,651,695,616 bytes
```

```
PRESS <ENTER> TO CONTINUE:
```

Verify that your installation choices are correct. If they are not, enter 'back' to go back and make choices. If you are satisfied, press the <ENTER> key.

```
===============================================================================
==
Ready To Install
----------------

InstallAnywhere is now ready to install CAMM 10.2.0.4 onto your system at
the
following location:

   /root/oracle/CAMM

PRESS <ENTER> TO INSTALL:
```

Press the <ENTER> key to begin installing. When the installation is complete, you'll be notified:

```
===============================================================================
==
Installation Complete
---------------------

Congratulations. CAMM 10.2.0.4 has been successfully installed to:

   /root/oracle/CAMM

PRESS <ENTER> TO EXIT THE INSTALLER:
```

Press <ENTER> to finish.

3.2.3 Setup CAMM Data Repository

CAMM requires a RDBMS be setup as the data repository for runtime metrics collection. Both Oracle 10g and MySQL RDBMS are supported for this purpose. Details for manual installation and configuration of Data Repository are described in Chapter 3.2.6.

3.2.4 Deploying CAMM Components

Deploying CAMM components involves a few simple steps:

- Configuring CAMM with information about the target application platform. This can optionally include configuration of the embedded MySQL database in cases

of manually configuring multiple instances of CAMM. For each monitoring environment, a single instance of CAMM can monitor multiple application servers or clusters.

- Deploying the CAMM Agent components to the target application server instances or clusters (e.g. managed servers in a cluster of a domain, etc).

3.2.4.1 Configuring CAMM

In CAMM, a monitored target application platform, whether it is an individual application server instance or a cluster in a management domain, is called a "Resource."

Although it is highly recommended that you simply register and update the configuration of your server resources through the CAMM Admin UI. To do this:

1. Click on the "Configure" tab on the left pane (which contains the navtree)
2. Click on the "Resource Configuration" node in the navtree
3. Click on the "Create New Resource" button in the main pane
4. Name your resource and select your application server product and version
5. Click "Continue"
6. Click the "Configure" button in the middle of the main pane
7. Enter in your application server details. Ignore the BPEL and ESB related options if you do not have the Oracle SOA Suite. (**Important**: If they appear, DO NOT modify the fields for "Agent Keystore Password" and "Agent Truststore Password".)
8. Click "OK"
9. Click "Save" (if you omit this step, your configuration will be lost)

It is possible to directly edit the `config/configuration.xml` file if necessary. The snippet below provides some detail in relation to the various elements within the configuration file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:configuration verbose="true"
xmlns:ns1="http://www.acsera.com/ns/configuration">
  <ns1:infrastructures name="default">
    <ns1:infrastructure name="Oracle Enterprise Manager" enabled="true">
      <ns1:resource name="medrec" enabled="true">
        <ns1:resourceDefinition>WebLogic_10.0.0</ns1:resourceDefinition>
        <ns1:mipProfiles>
          <ns1:mipProfile enabled="true">
            <ns1:mip>BEAJMXMIP9</ns1:mip>
            <ns1:configParameters>
              <ns1:configParameter>
```

```
                <ns1:key>username</ns1:key>
                <ns1:value>weblogic</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>password</ns1:key>

<ns1:value>${UuaSdejp2tYxNj0aG6pk6qSJeigeg1LGpIl6KB6DUsakiXooHoNSxqSJeigeg1
LGpIl6KB6DUsakiXooHoNSxkIr/GW49EHb}</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>bpelJNDIFactory</ns1:key>

<ns1:value>com.evermind.server.rmi.RMIInitialContextFactory</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>bpelJNDIURL</ns1:key>

<ns1:value>opmn:ormi://localhost:6003:oc4j_soa/orabpel</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>bpelusername</ns1:key>
                <ns1:value>oc4jadmin</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>bpelpassword</ns1:key>

<ns1:value>${pr+sWrsFVM06oo1mLJUqvKSJeigeg1LGpIl6KB6DUsakiXooHoNSxqSJeigeg1
LGpIl6KB6DUsakiXooHoNSxkIr/GW49EHb}</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>esbhost</ns1:key>
                <ns1:value>localhost</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>esbport</ns1:key>
                <ns1:value>80</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>esbusername</ns1:key>
                <ns1:value>oc4jadmin</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>esbpassword</ns1:key>

<ns1:value>${pr+sWrsFVM06oo1mLJUqvKSJeigeg1LGpIl6KB6DUsakiXooHoNSxqSJeigeg1
LGpIl6KB6DUsakiXooHoNSxkIr/GW49EHb}</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>protocol</ns1:key>
                <ns1:value>t3</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
```

```
                <ns1:key>host</ns1:key>
                <ns1:value>localhost</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>port</ns1:key>
                <ns1:value>7101</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>httpHost</ns1:key>
                <ns1:value>localhost</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>httpPort</ns1:key>
                <ns1:value>7101</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>agentKeystorePassword</ns1:key>

<ns1:value>${2PYLgzdp/sQgkFze7IX5jqSJeigeg1LGpIl6KB6DUsakiXooHoNSxqSJeigeg1
LGpIl6KB6DUsakiXooHoNSxkIr/GW49EHb}</ns1:value>
              </ns1:configParameter>
              <ns1:configParameter>
                <ns1:key>agentTruststorePassword</ns1:key>

<ns1:value>${2PYLgzdp/sQgkFze7IX5jqSJeigeg1LGpIl6KB6DUsakiXooHoNSxqSJeigeg1
LGpIl6KB6DUsakiXooHoNSxkIr/GW49EHb}</ns1:value>
              </ns1:configParameter>
            </ns1:configParameters>
          </ns1:mipProfile>
          <ns1:mipProfile enabled="true">
            <ns1:mip>OS</ns1:mip>
          </ns1:mipProfile>
          <ns1:mipProfile enabled="true">
            <ns1:mip>BEAJava9</ns1:mip>
          </ns1:mipProfile>
        </ns1:mipProfiles>
      </ns1:resource>
    </ns1:infrastructure>
  </ns1:infrastructures>
</ns1:configuration>
```

The parameters to access your WebLogic admin server are contained inside the <mipProfile> element whose <mip> element has value: BEAJMXMIP9. Change the necessary <value> parameters as appropriate. Parameters:

Username

This is the User Login name used to access the Admin Server.
Default: weblogic

Password
This is the User's password to access the Admin Server.
Default: weblogic

Protocol
This is the protocol used to access the admin server.
Default: t3

Host
This is the IP address or network name of the admin server.
Default: localhost

Port
This is the port the WebLogic Admin Server is using.
Default: 7001

These are the same username, password, host, and port used to access the WebLogic admin console. Make sure also that you have specified the right target version for Web Logic Service Pack (e.g. 8.1.3)

Much of the configuration of the CAMM 10.2.0.4 is done through editing the `config/Acsera.properties` file.

3.2.4.2 Deploying CAMM Agents on the WebLogic Platform

3.2.4.2.1 Deploying CAMM Java Agent

The *deployer* utility is used to install the CAMM *Deployer.ear* file and Java Agent on any application servers managed by CAMM 10.2.0.4.

Usage:

```
C:\CAMM\bin\deployer.bat [-version] <command> [-targets targets] [-resource
'configured resource name'] [-agentdir agentdir]
```

Where:

`command` is either `-deploy, -ejbdeploy, -copy, -enable, -status, -systemprops, -disable, -remove, or -ejbundeploy`.

`-version` specify WebLogic version, e.g. 8.1.3

`targets` is a comma-delimited list of server and/or cluster names (no spaces allowed). Default server name is `cgServer`.

`agentdir` is the name of the agent dir to use (no paths allowed, just a file name)

The following commands are supported:

`-deploy` - Initial deployment or upgrade of the agent (with automatic deployment of the CAMM EJB if it's not already deployed).

`-ejbdeploy` – Deploys the CAMM EJB only (typically only used for debugging purposes).
`-copy` – Copies the agent to the App Server without enabling it.
`-enable` – Enables an agent which is already installed on the App Server.
`-status` – Displays status of the deployed agent, including whether it is actually deployed.
`-systemprops` – Displays the System Props of the remote machine.
`-disable` – Removes references to the agent from the App Server start parameters. The App Server must be restarted for this to take effect.
`-remove` – Deletes the agent directories from the App Server. This will only succeed if the agent is already disabled and the App Server is restarted.
`-ejbundeploy` – Removes the CAMM EJB from the Admin Server and all Managed Servers.

`-resource` –The resource name entered for the managed server when it was originally configured in the CAMM Manager. The name must be exact and is case sensitive.

Deployer.ear is deployed on every managed server and admin server.

Deploying or Upgrading the Agent

To initially deploy the agent (and to redeploy it to an upgraded version), use the following command:

```
bin/deployer.bat –version 8.1.3 –deploy –targets <targets> -resource
‘resource name’
```

After deploying the agent, you must reboot the target application server.

Checking the status of a deployed Agent

To view the status of the deployed agent(s):

```
bin/deployer.bat -status <targets> -resource ‘resource name’
```

Disabling a running Agent

To disable the agent, use the undeploy command. Be sure to use the correct agent directory name:

```
bin/deployer.bat -undeploy -resource ‘resource name’-targets <targets>
```

After un-deploying the Agent, you must reboot the target application server.

Note: If upgrading the Deployer EAR, make sure you undeploy the older version from admin/managed servers.

Explicit un-deployment of the CAMM EJB

To remove the CAMM EAR (which is automatically deployed if the deployer utility requires it), issue the following command:

```
bin/deployer.bat -ejbundeploy -resource ‘resource name’-targets <targets>
```

This will undeploy the CAMM EJB from the admin server and all managed servers. No restart is required after this step.

Examples

To deploy to the admin server on `localhost` (to upgrade the agent use the same command):

```
bin/deployer.bat –deploy -resource ‘resource name’-targets cgServer
```

Restart cgServer for this to take effect.

To deploy to all servers in cluster C1 (not including the admin server):

```
bin/deployer.bat -deploy -resource ‘resource name’-targets C1
```

To upgrade all servers in cluster C1 (assume default CAMMAgent directory is in use and cannot be replaced)

```
bin/deployer.bat -copy -resource ‘resource name’-targets C1 -admindir
CAMMAgentUpgrade

bin/deployer.bat –enable -resource ‘resource name’-targets C1 -admindir
CAMMAgentUpgrade
```

RestartC1 servers for this to take effect.

To get status of all servers in C1:

```
bin/deployer.bat –status -resource ‘resource name’-targets C1
```

To undeploy the CAMM agent:

```
bin/deployer.bat –disable -resource ‘resource name’-targets C1 // remove
references to CAMMAgent
```

Restart the C1 servers. This will unlock the CAMM Agent files

```
bin/deployer.bat –remove -resource ‘resource name’-targets C1
```

To undeploy the CAMM EJB from all servers:

```
bin/deployer.bat –ejbundeploy -resource ‘resource name’
```

3.2.4.2.2 Deploying CAMM OS Agent

To initially deploy the OS Agent (and to redeploy it to an upgraded version), use the following command:

```
bin/deployer.bat osmetric -deploy -resource ‘resource name’–targets
<targets>
```

If you are running the OS Agent on the same machine as CAMM, set the `RMI.Registry.OSMetrics.LocalRegistry` property in `ORACLE_WEBLOGIC_HOME\weblogic81\CAMMOSMetricAgent\config\OSMetrics.properties` to “false”.

Change element in `CAMM_HOME\config\configuration.xml`

From:

```
<ns1:mip enabled="false" name="OS">
```

To:

```
<ns1:mip enabled="true" name="OS">
```

Note: If JDK used for Managed Servers is based on JRockit you don’t need to install OS Metric. CAMM uses JMX services of JRockit to retrieve OS Metrics

Disabling a running CAMM OS Agent

To disable the CAMM OS Agent, use the `-disable` command. Be sure to use the correct agent directory name:

```
bin/deployer.bat –disable -resource ‘resource name’-targets <targets>
```

Remove a running CAMM OS Agent

To remove the CAMM Agent, use the `-remove` command. Be sure to use the correct agent directory name:

```
bin/deployer.bat osmetric –remove -resource ‘resource name’-targets
<targets>
```

Explicit un-deployment of the CAMM OS EJB

To remove the CAMM OS EJB (which is automatically deployed if the deployer utility requires it), issue the following command:

```
bin/deployer.bat osmetric –ejbundeploy -resource ‘resource name’-targets
<targets>
```

This will undeploy the CAMM OS EJB from the admin server and all managed servers. No restart of Application Server is required after this step.

3.2.4.2.3 Manual CAMM Agent deployment for WebLogic platform

Manual CAMM Agent deployment is a two staged process:

1. Deployment of the CAMM EJB (*Deployer.ejb)* on application server,
2. Deployment of the CAMM Agent libraries with supporting application server configuration.

This chapter is dedicated to detailed description of these two steps.

CAMM EJB deployment:
Deploy and activate $*CAMM_HOME/deploy/Deployer.ear* on every application server to be monitored.

Deploy CAMM Agent:
Copy files under *$CAMM_HOME/deploy/bea* to the home directory of the application server *$WEBLOGIC_HOME/weblogic81/AcseraAgent,* retaining original file structure. If you're using the nodemanager, you will need to put the following in to the Configuration->Remote Start (WebLogic admin console) for each managed server:

<u>For JRockit:</u>

Java Options:
-Xbootclasspath/p:${ACSERA_AGENT_HOME}lib/jagent1.4.2_05.jar -Xmanagement:class=com.acsera.javaagent.JRockitMultiManagement -Dmultimanagement1=class=com.acsera.javaagent.BEAJavaAgent -Dacsera.config=${ACSERA_AGENT_HOME}/config/BEAJAgent.properties

Classpath:
${ACSERA_AGENT_HOME}/lib/beaaj.jar

Replace ${ACSERA_AGENT_HOME} with the path to wherever you uploaded the acsera agent files to.

<u>For Sun JDK</u>

Java Options:
-Xbootclasspath/p:${ACSERA_AGENT_HOME}lib/jagent1.4.2_05.jar -Dacsera.config=${ACSERA_AGENT_HOME}/config/BEAJAgent.properties

Classpath:
${ACSERA_AGENT_HOME}/lib/beaaj.jar

Make sure to provide proper separator when listing libraries in the classpath. For UNIX it is colon ':' and Windows semicolon ';'.

Once the entire configuration is done, restart the managed servers so that they pick up the CAMM Agents.

3.2.4.3 Deploying CAMM Agents on the WebSphere Platform

Deployment of CAMM Agent for WebSphere platform is a two-phase process. First you need to install CAMM IBM Deployer application responsible for CAMM Agent libraries deployment initial handshake between CAMM and CAMM Agent, then deploy the Agent libraries on the target system. There are two options to achieve this:

- Automatic deployment using CAMM *websphereDeployer* script
- Manually installing all the supporting artifacts

3.2.4.3.1 Required IBM WebSphere Libraries

Once WebSphere is registered via the CAMM administration UI, the required libraries for CAMM to connect to WebSphere must be provided in the actual classpath by modifying the configuration.xml file directly in the CAMM Manager config sub-directory (i.e. CAMM_HOME/config/configuration.xml).

Note that an installation must be available directly on the machine or via a NFS/SMB mount. The following property would need to be modified to mirror the absolute path to the WebSphere Application Server (WAS) home directory. The necessary libraries will then be loaded on the classpath accordingly.

Example of wsHome setting:

```
<ns1:configParameter>
    <ns1:key>wsHome</ns1:key>
    <ns1:value>C:/Progra~1/IBM/WebSphere/AppServer</ns1:value>
</ns1:configParameter>
```

3.2.4.3.2 Automatic CAMM Agent deployment for WebSphere platform

3.2.4.3.2.1 Deploying WebSphere File Transfer Application

If you are using a network deployment manager (dmgr), you should skip this step. Otherwise (if running a standalone WAS server), check to make sure the WebSphere File Transfer Application is installed and running. If not, deploy `filetransfer.ear` to the WebSphere application server. This is required to enable the CAMM Agent automatic deployment process. If the Agent is being installed manually, the file transfer application is not required.

3.2.4.3.2.2 Deploying CAMM Java Agent for WebSphere platform

The deployer utility is used to install the CAMM Java Agent on any application servers managed by CAMM 10.2.0.4.

Usage:

```
C:\CAMM\bin\websphereDeployer.bat [-version] <command> [-targets targets]
[-resource 'configured resource name'] [-agentdir agentdir]
```

Where:

`command` is either `-deploy, -ejbdeploy, -copy, -enable, -status, -systemprops, -disable, -remove, or -ejbundeploy`.

`resource` is the name of the resource that was configured in the CAMM Manager. The name must be exact and is case sensitive.

`targets` is a comma-delimited list of server and/or cluster names (no spaces allowed). For example: `websphere_portal`. In a clustered environment this should be a cluster name.

`agentdir` is the name of the agent dir to use (no paths allowed, just a file name). For example: `CAMM_Agent`.

The following commands are supported:

`-deploy` - Initial deployment or upgrade of the agent (with automatic deployment of the CAMM EJB if it's not already deployed).

`-ejbdeploy` – Deploys the CAMM EJB only (typically only used for debugging purposes).
`-copy` – Copies the agent to the App Server without enabling it.
`-enable` – Enables an agent which is already installed on the App Server.
`-status` – Displays status of the deployed agent, including whether it is actually deployed.
`-systemprops` – Displays the System Props of the remote machine.
`-disable` – Removes references to the agent from the App Server start parameters. An The App Server must be restarted for this to take effect.
`-remove` – Deletes the agent directories from the App Server. This will only succeed if the agent is already disabled and the App Server is restarted.
`-ejbundeploy` – Removes the CAMM EJB from the Admin Server and all Managed Servers.

Deploying or Upgrading the Agent

To initially deploy the agent (and to redeploy it to an upgraded version), use the following command:

```
bin/websphereDeployer.bat –version <version> -deploy -resource ‘resource name’ –targets <targets> -agentdir <agentdir>
```

After deploying the agent, you must reboot the target application server.

Checking the status of a deployed Agent

To view the status of the deployed agent(s):

```
bin/websphereDeployer.bat -status -resource ‘resource name’<targets>
```

Disabling a running Agent

To disable the agent, use the undeploy command. Be sure to use the correct agent directory name:

```
bin/websphereDeployer.bat -undeploy -resource ‘resource name’-targets <targets>
```

After un-deploying the Agent, you must reboot the target application server.

Explicit un-deployment of the CAMM EJB

To remove the CAMM EJB (which is automatically deployed if the deployer utility requires it), issue the following command:

```
bin/websphereDeployer.bat -ejbundeploy -resource ‘resource name’-targets <targets>
```

This will undeploy the CAMM EJB from the admin server and all managed servers. No restart is required after this step.

Examples

To deploy to the admin server on `localhost` (to upgrade the agent use the same command):

```
bin/websphereDeployer.bat -deploy -resource ‘resource name’ -targets WebSphere_Portal -agentdir AcseraAgent
```

Restart WebSphere_Portal for this to take effect.

3.2.4.3.3 Manual CAMM Agent deployment for WebSphere platform

Manual CAMM Agent deployment is a three staged process:

1. Deployment of the CAMM EJB (*IBMDeployer.ejb* in case of Server and *IBMPortalDeployer.ejb* in case of Portal) on application server,
2. Configuration of the Applications servers to report JMX performance data, and

3. Deployment of the CAMM Agent libraries with supporting application server configuration.

This section is dedicated to detailed description of all three steps.

CAMM EJB deployment:
Deploy and activate $*CAMM_HOME/deploy/IBMDeployer.ear* in case of Server and $*CAMM_HOME/deploy/IBMPortalDeployer.ear* in case of Portal on every application server to be monitored.

Application Server configuration:
Configure Application server to report JMX metrics (e.g. sessions, JVM related metrics). For this you need to configure Performance Monitoring Infrastructure service on each Application server:

- Check the Startup Check Box
- Set Initial Specification Level to Standard

Deploy CAMM Agent:
Copy files under *$CAMM_HOME/deploy/ibm* to home directory of the application server *$WEBSPHERE_HOME/AppServer/AcseraAgent* retaining original file structure.

Configure Application Server Java Virtual Machine classpath by adding the following values:
Classpath: *$WEBSPHERE_HOME/AppServer/AcseraAgent/lib/ibmaj.jar*, and
Boot Classpath: $*WEBSPHERE_HOME/AppServer/AcseraAgent/lib/jagentIBMx.jar (x could be 1.4.2, 1.4.2.3.1 etc* depending on JDK and OS)

This can be done from Admin console: Application Server -> <instance_name>->ProcessDefintion -> Java Virtual Machine

Make sure to provide proper separator when listing libraries in the classpath. For UNIX it is colon ':' and Windows semicolon ';'.

Finally add new JVM Custom properties to each Application Server. This can be done from Admin console: Application Server -> <instance_name>->ProcessDefintion -> Java Virtual Machine. Once you are there choose "Custom Properties" and add the following properties:

1. *acsera.config* with value of *$WEBSPHERE_HOME/AppServer/CAMMAgent/config/IBMJAgent.properties,*
2. *acsera.server.name* with value of <*application server instance name*> (e.g. WebSphere_Portal_2).

Once all the configuration is done, restart the managed servers so that they pick up the CAMM Agents.

3.2.4.3.4 Automatic CAMM Agent deployment for Oracle platform

3.2.4.3.4.1 Deploying CAMM Java Agent for Oracle Platform

The deployer utility is utilized to install the CAMM Java Agent on any application servers managed by CAMM 10.2.0.4.

Usage:

```
C:\CAMM\bin\oracleDeployer.bat [-version] <command> -[resource 'configured resource name'] [-targets targets] [-agentdir agentdir]
```

Where:

`command` is either `-deploy, -ejbdeploy, -copy, -enable, -status, -systemprops, -disable, -remove, or -ejbundeploy`.

`resource` is the name of the resource that was configured in the CAMM Manager. The name must be exact and is case sensitive.

`targets` is a comma-delimited list of server and/or cluster names (no spaces allowed). For example: `oc4j_soa`. In a clustered environment this should be a cluster name.

`agentdir` is the name of the agent dir to use (no paths allowed, just a file name). For example: `CAMM_Agent`.

The following commands are supported:

`-deploy` - Initial deployment or upgrade of the agent (with automatic deployment of the CAMM EJB if it's not already deployed).

`-ejbdeploy` – Deploys the CAMM EJB only (typically only used for debugging purposes).
`-copy` – Copies the agent to the App Server without enabling it.
`-enable` – Enables an agent which is already installed on the App Server.
`-status` – Displays status of the deployed agent, including whether it is actually deployed.
`-systemprops` – Displays the System Props of the remote machine.
`-disable` – Removes references to the agent from the App Server start parameters. An The App Server must be restarted for this to take effect.
`-remove` – Deletes the agent directories from the App Server. This will only succeed if the agent is already disabled and the App Server is restarted.
`-ejbundeploy` – Removes the CAMM EJB from the Admin Server and all Managed Servers.

Deploying or Upgrading the Agent

To initially deploy the agent (and to redeploy it to an upgraded version), use the following command:

```
bin/oracleDeployer.bat –version <version> -deploy -resource ‘resource name’
–targets <targets> -agentdir <agentdir>
```

After deploying the agent, you must reboot the target application server.

Checking the status of a deployed Agent

To view the status of the deployed agent(s):

```
bin/oracleDeployer.bat -status –adminurl -resource ‘resource name’
<targets>
```

Disabling a running Agent

To disable the agent, use the undeploy command. Be sure to use the correct agent directory name:

```
bin/oracleDeployer.bat -undeploy -resource ‘resource name’ -targets
<targets>
```

After un-deploying the Agent, you must reboot the target application server.

Examples

To deploy to the admin server on `localhost` (to upgrade the agent use the same command):

```
bin/oracleDeployer.bat -deploy -resource ‘resource name’ -targets oc4j_soa
-agentdir AcseraAgent
```

Restart Oracle SOA Suite for this to take effect.

3.2.4.4 Running Multiple CAMM Instances

When is it necessary to configure Multiple CAMM Instances?
CAMM offers flexibility in monitoring diverse, heterogeneous application server environments. A single instance of CAMM can be configured to monitor:

- multiple management domains (e.g.WebLogic domains)
- clusters (WebLogic clusters or WebSphere cells)
- standalone application server instances
- a mix of the above.

This flexibility allows a single instance of the CAMM to oversee a single *environment*, even if the environment spans multiple vendors and platforms. Given this capability, the "monitoring context" of a single instance of CAMM should be defined carefully, with considerations to:

- the purpose of the environment
- the traffic / load pattern seen in a given environment
- network topology, organizational boundaries of different environments.

The criteria listed above often suggest that a monitoring environment be defined along the lines of the software lifecycle: development, performance tuning, quality assurance, staging, and production. For example, the QA environment typically mirrors the Production environment in terms of the software build, platform and systems configuration. However, despite the similarity in configuration, the traffic/load characteristics between these environments are clearly different, while they could also be separated by network topology (different subnets, firewalls, etc.) and organizationally (engineering vs. operations). For this reason, deploying multiple instances of CAMM, one for each environment, can simplify configuration and streamline operations of the tool, while ensuring consistent interpretation of measured performance metrics.

In general, each instance of CAMM has its own unique configurations:

- A unique `INSTANCE_NAME` directory which contains set of directories:
  - `config` – for instance-specific configuration files.
  - `schema` – contains required XML schemas
  - `deploy` – contains required binary files for agent, webapp deployment
  - `log` – directory for storing manager log files
  - `slo` – service level objectives definition specific for this instance
- Instance-defining properties in the `config/Acsera.properties` file:
  - `RMI.Registry.Port`
  - `DataManager.MySQL.DbLocalName`
  - `ServiceController.ManagerID`
- Depending on the resources being monitored, entries in the `config/configschema.xml` file that contains path names to application platform-specific jar files.

The steps for deploying multiple instances of CAMM are outlined as follows:

1. Create instance directory

There will always be a "default" instance of CAMM (the installed instance). For each additional Manager instance needed, first create an instance directory underneath *CAMM_HOME*. For each instance directory, do:

1. Copy the installed `config`, `schema` and `deploy` directories under the new instance directory.
2. Manually create the `log` directory (optional).

2. Update INSTANCE_DIR/config/Acsera.properties file.

The following properties need to be set to unique, instance-specific values:

- RMI.Registry.Port
- DataManager.MySQL.DbLocalName
- RMI.RemoteServiceController.ServerPort
- RMI.JavaProvider.ServerPort

The default value for RMI.Registry.Port is 51099. It's recommended that additional Manager instances use values counting down from the default (51098, 51097, etc.). RMI.RemoteServiceController.ServerPort and RMI.JavaProvider.ServerPort and shall be incremented by 2 for every CAMM instance.

For DbLocalName, use a name that incorporates the instance name, for example: qv6_PRODUCTION or qv6_QA. Pre-pending the instance name (the qv6_ prefix) is useful in cases where CAMM is sharing a MySQL database instance with other applications which may have their own schemas.

In order to be able to tailor Custom Views per CAMM instance the following parameter needs to be set:

- ServiceController.ManagerID

You can set it to any string. CAMM GUI prefixes the custom view file name used to store the custom view definitions on the client. By default, if this property is not set the ServiceController.ManagerID becomes its hostname which may be an IP address if it cannot find the symbolic one.

Here is an example of this property:

```
ServiceController.ManagerID = 192.168.8.24
```

On the client side, files which define custom views are located under "Documents and Settings" on the client machine. Here is an example:

```
C:\Documents and Settings\user_acsera\.acsera_preferences
```

In our example, you will see a custom view definition file in this folder:

```
192.168.3.24_userpref.ser
```

3. Configure the CAMM instance to monitor resources (application servers) of interest.

Resources can be defined in the CAMM UI. Start the CAMM and configure each monitored resource with correct values for the following properties:

- Version
- Username
- Password
- Host
- Port

Do this for each resource to be monitored. A single instance of CAMM can monitor multiple domains and application servers of different vendor platforms.

4. Deploy the CAMM Agent to the target application platform

This includes:

- deploying `Deployer.ear` into WebLogic application server,
- copying CAMM Agent class files to the WebLogic application server, and
- updating the WebLogic start-up scripts to load the CAMM Agent.

The CAMM Agent can be installed automatically by running `deployer` from `CAMM_HOME/bin` or manually.

Automated Agent installation:
The `deployer` scripts require that the instance name be defined in the environment before being invoked:

Windows:

```
set INSTANCE_NAME=<name>
bin/deployer.bat  <deployer-parameters>
```

UNIX:

```
INSTANCE_NAME=<name>
export INSTANCE_NAME
bin/deployer.sh <deployer-parameters>
```

or

```
INSTANCE_NAME=<name> bin/deployer.sh <deployer-parameters>
```

Manual Agent installation:
Sometimes the WebLogic startup files need to be manually updated to enable the loading of the CAMM Agent, especially when custom startup scripts are being used. Also, security policies may require manual deployment of the CAMM Agent files across the target application servers.
Deploy `Deployer.jar` (located under `CAMM_HOME/deploy`) as CAMM EJB Module on the target Application Server and copy the `agent` directory (located under `ACSERA_HOME/deploy`) to the WebLogic server domain directory `CAMMAgent`. Update the WebLogic startup script in a way so that it invokes the `acseraoptions` script found in `CAMMAgent/bin` folder. For a Windows system this will look like:

```
REM BEGIN CAMM Agent install
REM DO NOT MODIFY!
REM Use CAMM Administration to remove
if "%DOMAIN_HOME%" == "" goto continueacsera
```

```
For /F %%i in ('Findstr "*"
C:\bea\WEBLOG~1\CAMMAgent\config\acseradomains.txt') do (goto
continueacsera)
For /F %%i in ('Findstr "%DOMAIN_HOME%"
C:\bea\WEBLOG~1\CAMMAgent\config\acseradomains.txt') do (goto
continueacsera)
goto End

:continueacsera
if EXIST C:\bea\WEBLOG~1\CAMMAgent.acs_commit (
echo C:\bea\WEBLOG~1\CAMMAgent Updated.  Committing changes
    rmdir /q /s C:\bea\WEBLOG~1\CAMMAgent.acs_rollback
    ren C:\bea\WEBLOG~1\CAMMAgent CAMMAgent.acs_rollback
    ren C:\bea\WEBLOG~1\CAMMAgent.acs_commit CAMMAgent
)

set ACSERA_AGENT_HOME=C:\bea\WEBLOG~1\CAMMAgent
call C:\bea\WEBLOG~1\CAMMAgent\bin\agentoptions.bat

:End
REM END CAMM Agent install
```

For UNIX – similar to the following:

```
# BEGIN CAMM Agent install
# DO NOT MODIFY!
# Use CAMM Administration to remove
count=`grep -c '*' /bea/weblogic81/CAMMAgent/config/acseradomains.txt`
if [ \( "$count" -ne 0 \) -o -z "$DOMAIN_HOME" ]
then

 ACSERA_AGENT_HOME=/bea/weblogic81/CAMMAgent
export ACSERA_AGENT_HOME
. /bea/weblogic81/CAMMAgent/bin/agentoptions.sh
else
            if [ -n "$DOMAIN_HOME" ]
            then
            name=`grep "$DOMAIN_HOME"
/bea/weblogic81/CAMMAgent/config/acseradomains.txt`
            if [ -n "$name" -a \( "$name" = "$DOMAIN_HOME" \) ]
            then

 ACSERA_AGENT_HOME=/bea/weblogic81/CAMMAgent
export ACSERA_AGENT_HOME
. /bea/weblogic81/CAMMAgent/bin/agentoptions.sh
            fi
            fi
fi
# END CAMM Agent install
```

5. CAMM Administrator application configuration

Each CAMM instance is administered by a separate web application instance. In each of the manager instances above, you set a unique RMI address/port. You'll also need to set up each of the `acseraadmin.war's` to match.

Deploy one `acseraadmin.war` for each of the CAMM instances.

The .war file can be unjarred. After unjarring the .war file in a temporary directory, edit the `WEB-INF/web.xml` file found inside:

- Edit the `Acsera.RMIRegistry.HOST` value to reflect the address of this instance of the Manager. The default is localhost, and this is correct if the Manager is on the same machine as the container.
- Edit the `Acsera.RMIRegistery.Port` value to reflect the port of this instance of the Manager. Note that each `acseraadmin.war`/Manager pair must have a unique port if the Manager exists on the same machine with other Managers.

After editing the `web.xml` file, jar the contents of the temp directory and name the jar file `qv6admin_ENVIRONMENT.war`, for example, `qv6admin_QA.war` or `qv6admin_PROD.war`. Each of these `acseraadmin.war` deployments can be in the same container or in separate containers. If they're in the same container, you'll need to give each a unique name, such as `acseraadmin_QA.war`, so they don't conflict. In Tomcat, the root path in the URL corresponds to the name of the war file.

Note that this will impact the URL used for the Browser to access the Manager. Deploying each in the same container could result in URLs (note different filepath). For example, for files `acseraadmin_QA.war` and `acseraadmin_PRODUCTION.war`, the URLs would look like the following, respectively:
http://1.2.3.4:5557/qv6admin_QA Manager 1 (Domain 1)
http://1.2.3.4:5557/qv6admin_PRODUCTION Manager 2 (Domain 2)
etc.

Each URL above will launch a dedicated CAMM GUI for the given environment. In general, having a single container with different URL paths for different environments is preferable to multiple containers listening at different ports.

6. Running multiple CAMM instances.

`CAMM_HOME/bin/acsera <instance_dir_name>`
Runs the CAMM instance

`CAMM _HOME/bin/acshut <instance_dir_name>`
Stops the CAMM instance

Notice that scripts take an instance name as the first argument. Leaving out the instance name will cause the script to target the default instance.

Generally, to use the CAMM scripts like `acsera` and `acshut`, the following variables MUST be set in the environment:

```
CAMM_HOME
```

`ACSERA_HOME` should be set to the directory into which the CAMM was originally installed.

The following variables MAY need to be set in the environment:

```
JAVA_HOME
PATH_SEPARATOR
```

If there is a JDK installed under the `ACSERA_HOME` directory, then the scripts will use that JDK. Otherwise, `JAVA_HOME` must be set to the directory of the installed JDK.

PATH_SEPARATOR is used only by the *.sh versions of the scripts, and should be set to ':' (colon) for native UNIX (Linux, etc.) systems and to ';' (semi-colon) for shell environments on Windows. If not set in the environment, it defaults to native UNIX mode.

3.2.5 Installing and Configuring CAMM Data Repository

CAMM 10.2.0.4 maintains a database (can be Oracle or MySQL as indicated in the sections below) of information collected on the system it is monitoring. This database can be contained on the same machine that CAMM 10.2.0.4 is running on. It can also be hosted on a remote machine. The following paragraphs describe how to configure the CAMM Data Repository. By default the CAMM installation utility installs the database under *$ CAMM _HOME/database* directory.

The following sections describe procedures that may be performed by the CAMM administrator to customize the database configuration.

3.2.5.1 Configuring Oracle DBMS for CAMM 10.2.0.4

Oracle CAMM currently supports the Oracle 10g database for its runtime repository. In order to set this up, the following steps are necessary in order to setup and configure the Oracle DBMS for Oracle CAMM. Oracle CAMM will initialize the database upon connecting to it and generate the required tables.

1. Install Oracle DB: 10*g* on a separate machine
2. Create a new user in database (preferably "Oracle" or something distinct)

3. Set the System Global Area to have at least 1275068416 bytes (1GB)

4. Increase the number of processes in database from 150 to 300 (Optional)

Execute the following commands using Oracle SQL *Plus

a. connect / as sysdba; (this will connect to the database as DBA)
b. show parameter processes;

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- --------
aq_tm_processes                      integer     0
db_writer_processes                  integer     1
gcs_server_processes                 integer     0
job_queue_processes                  integer     10
log_archive_max_processes            integer     2
processes                            integer     150
```

c. alter system set processes=300 scope=spfile; (run this to increase the no of processes .. you can set it higher no if you want)
d. shutdown immediate (for shutdown the server)
e. startup (startup the server)
f. show parameter processes; (run this to verify the changes took effect)

5. In the Configuration file set the following properties

- Metric table prefix :

  MetricTableDescriptor.Prefix = metric

- JDBC setting :

  Values for properties DataManager.Oracle.DbHost , DataManager.Oracle.DbPort DataManager.Oracle.DbName ,DataManager.Oracle.DbUser DataManager.Oracle.DbPassword depends on installation of DB and other properties have the default values.

```
##############################
# Oracle DBMS jdbc setting #
##############################
DataManager.Oracle.DbDriverName = oracle.jdbc.OracleDriver
DataManager.Oracle.DbUrlFormat = {0}:{1}:@{2}:{3}:{4}
DataManager.Oracle.DbConnectorType = jdbc
DataManager.Oracle.DbConnectorDbms = oracle:thin
DataManager.Oracle.DbHost = <your host name>
DataManager.Oracle.DbPort = <your port number>
DataManager.Oracle.DbName = <your db name>
DataManager.Oracle.DbUser = <your db user>
DataManager.Oracle.DbPassword = <your db password>
DataManager.Oracle.DbLocalName = <your db sid or local name>
DataManager.Oracle.InListLimit = 1000
```

3.2.5.2 Configuring MySQL DBMS for CAMM 10.2.0.4

Oracle CAMM currently supports MySQL 4.1 database and higher for its runtime repository. In order to set this up, the following steps are necessary in order to setup and configure the Oracle DBMS for Oracle CAMM. Oracle CAMM will initialize the database upon connecting to it and generate the required tables.

1. Install MySQL 4.1 or higher on a separate machine
2. Create a new user in database (preferably “Oracle” or something distinct) and grant the user the appropriate privileges
3. Tune memory for performance as indicated below:

On a 1 GB server shared by CAMM and MySQL, the following parameter should be set in the my.ini file to increase database subsystem performance:

```
set-variable=key_buffer=128M
```

On a 2 GB server shared by CAMM and MySQL or on a dedicated to MySQL 1GB server, the parameter should be set as:

```
set-variable=key_buffer=256M
```

4. In the Configuration file set the following properties

- Metric table prefix :

  MetricTableDescriptor.Prefix = metric

- JDBC setting :

  Values for properties DataManager.MySQL.DbHost , DataManager. MySQL.DbPort DataManager. MySQL.DbName ,DataManager. MySQL.DbUser DataManager. MySQL.DbPassword depends on installation of DB and other properties have the default values.

  ```
  ##############################
  # MySQL DBMS jdbc setting #
  ##############################
  DataManager.MySQL.DbDriverName = com.mysql.jdbc.Driver
  DataManager. MySQL.DbConnectorType = jdbc
  DataManager. MySQL.DbConnectorDbms = mysql
  DataManager. MySQL.DbHost = <your host name>
  DataManager. MySQL.DbPort = <your port number>
  DataManager. MySQL.DbName = <your db name>
  DataManager. MySQL.DbUser = <your db user>
  DataManager. MySQL.DbPassword = <your db password>
  DataManager. MySQL.DbLocalName = <your db sid or local name>
  DataManager. MySQL.InListLimit = 1000
  ```

3.2.6 Backing Up the CAMM 10.2.0.4 Database

It is highly recommended that regular back-ups be made of the database utilized as a runtime repository for Oracle CAMM. The Oracle database provides a back-up utility that can be utilized for this purpose. These convenient tools allow users to quickly back-up and restore the database if necessary. The most convenient method for backing up the MySQL database is using the mysqldump command. By default, this is found in c:\mysql\bin on Windows, or /mysql/bin on UNIX.

3.2.6.1 Tables created by CAMM 10.2.0.4

For reference, the tables created by CAMM 10.2.0.4 and their contents are listed below. The product generates these on its own when it connects to the database initially, so there is no need to run this script against the database directly. These are detailed in CREATE statements, in the case the Administrator wants to create these tables manually. When the CAMM 10.2.0.4 is executed and does not find a “Oracle” database, it will attempt to create it with the structure defined below:

```
CREATE DATABASE acsera;

USE acsera;

CREATE TABLE config_ids (
  element_name text NOT NULL,
  element_id int(11) NOT NULL,
  UNIQUE KEY element_id (element_id),
  KEY element_name (element_name(255))
)

CREATE TABLE derived_ids (
  element_name text NOT NULL,
  element_id int(11) NOT NULL,
  UNIQUE KEY element_id (element_id),
  KEY element_name (element_name(255))
)

CREATE TABLE event_j2ee (
  ev_me_id int(11) NOT NULL,
  ev_start bigint(20) NOT NULL,
  ev_complete bigint(20) NOT NULL,
  ev_duration int(11) NOT NULL,
  ev_expiration bigint(20) NOT NULL,
  ev_type varchar(255) binary NOT NULL,
  ev_attributes text NOT NULL
)

CREATE TABLE managed_entities (
  me_id int(11) NOT NULL,
  key_name varchar(255) binary NOT NULL,
  key_value varchar(255) binary NOT NULL,
  ref_ts bigint(20) NOT NULL
```

```
)

CREATE TABLE metric_j2ee (
  end_ts bigint(20) NOT NULL,
  me_id int(11) NOT NULL,
  metric_abstract_id int(11) NOT NULL,
  metric_concrete_id int(11) NOT NULL,
  metric_value int(11) NOT NULL,
  KEY end_ts (end_ts),
  KEY me_id (me_id),
  KEY metric_abstract_id (metric_abstract_id)
)
```

3.3 Post- deployment

Oracle CAMM requires that some open source libraries and other utilities be installed or made available after completing the installation and deployment of the product. In some cases, the requirements are only necessary when specific platforms are being monitored such as IBM WebSphere.

3.3.1 Post-deployment requirements for all platforms

The following libraries and/or utilities need to be made available prior to running Oracle CAMM.

3.3.2 IBM Post-deployment Requirements

The following post-deployment requirements pertain directly to WebSphere.

3.3.2.1 Configuring QV for WAS 6.1 secured connections

The main goal is to add the signer certificate of each admin server to QV's truststore that QV needs to connect for each resource. This allows QV to trust the admin server when making secured (SSL) connections to server and without this trust the SSL handshake will fail. In the case when using the default QV truststore, the server's signer certificate would be added to AcseraManagerTrust.jks. This procedure assumes that the customer is using the default key.p12 and trust.p12 keystores for their security support. If a different trust store is being used, refer to that trust store instead.

1) Exporting the admin server's signer certificate for resource.

If admin server is the deployment manager in a WAS ND, export signer certificate from trust.p12 of the deployment manager located at path

<WAS_HOME>\profiles\Dmgr01\config\cells\<CellName>\trust.p12

If admin server is a standalone server, export signer certificate from trust.p12 of the standalone server located at path

<WAS_HOME>\profiles\AppSrv01\config\cells\<CellName>\nodes\<NodeName>\trust.p12

To export, run the following command:

keytool –export –keystore <trust path> -storepass WebAS –storetype PKCS12 –alias default_signer –file servercert

Note: when exporting a PKCS12 store type, you need to run keytool from an IBM JDK since it has support for this format type.

2) Importing the server's signer certificate into QV's truststore

Copy over the exported certificate file, servercert, from server's host to QV host. To import the server's signer certificate, run the following command:

keytool –import –keystore <ACSERA_HOME>/config/AcseraManagerTrust.jks –storepass acseramanager –storetype JKS –alias default_signer –file servercert

Assign a new alias name, if necessary, especially if the alias name conflicts with an existing entry in the trust store.

The keytool command prior to updating the truststore will prompt for confirmation in which case "yes" should be entered.

For secured connection to WAS 5.1 or 6.0, the signer certificate from the DummyClientTrustFile.jks can be exported and imported into AcseraManagerTrust.jks in similar fashion. For WAS 5.1 or 6.0, the signer certificate is same for all installations since they are not dynamically created for each server profile like in WAS 6.1. This is assuming the customer has turned on global security but is relying on the default WAS Dummy key/trust stores.

At present with QV running with a bundled Sun HotSpot JDK, it is not possible for QV to configure with PKCS12 type key/trust stores for secured connections. IBM JDK has built-in enhancements that allow it to work with PKCS12 key/trust stores, such as WAS 6.1's default key.p12 and trust.p12 stores. Also, there is a WAS 6.1 automatic function that is enabled with the property com.ibm.ssl.enableSignerExchangePrompt=true that allows a client connecting to a secure WAS port that allows automatic download of server's signer certificate and update of client's truststore. However, this automatic function is only available when QV is running with an IBM JDK which is not the case at present. This is reason why we need to follow the above procedure to connect with a secured WAS 6.1.

3.3.3 Configuring Oracle SOA Suite for Secure Connectivity

In Oracle SOA Suite CAMM will connect to the Oracle application server using the RMIS protocol. The SSL handshake can be configured by performing the following tasks for each Oracle SOA Suite instance that is to be managed by CAMM.

1. Update the <ssl-config/> element in rmi.xml for each SOA instance in the SOA suite manually.
2. This xml resides in %ORACLE_HOME%/j2ee/<instance>/config/rmi.xml. Please navigate to that sub-directory.
3. Open rmi.xml with any text or XML editing tool.
4. If the rmi.xml does have <ssl-config/> element add the below element
   It should be the following:

<ssl-config keystore="C:\product\10.1.3.1\OracleAS_1\AcseraAgent\config\AcseraJavaAgentKey.jks" keystore-password="acserajava" />

5. Restart the Oracle SOA Suite server

RMIs communication should now be configured enabling the SSL handshake to take place between CAMM and the managed Oracle SOA Suite server.

3.3.4 Configuring Oracle WebLogic Server or Oracle WebLogic Portal (WLP) for Secure Connectivity

In order to configure Oracle WLP to handle connectivity via RMIS, the following configuration changes are necessary.

In WLP10 the location of Keystore files needs to be updated through console via the following steps.

1. Login to the WebLogic Server console and select the servers under Environment . Servers list that is displayed which you plan to manage with CAMM
2. Select a server from the server list.
3. Select the keystores tab click on “Load & Edit” to update the Keystore
4. Make the following changes:

Identity

Custom Identity Keystore: location of the identity file (i.e. C:\beaPortal10\wlserver_10.0\AcseraAgent\config\AcseraJavaAgentKey.jks)

Custom Identity Keystore Type: JKS

Custom Identity Keystore Passphrase:acserajava

Confirm Custom Identity Keystore Passphrase:acserajava

Trust

Custom Trust Keystore: location of the trust file (i.e.C:\beaPortal10\wlserver_10.0\AcseraAgent\config\AcseraJavaAgentTrust.jks)

Custom Trust Keystore Type: JKS

Custom Trust Keystore Passphrase:acserajava
Confirm Custom Trust Keystore Passphrase:acserajava

5. Click on save button and then Resource Configuration button .Finally restart the server.
6. Repeat steps 2 through 5 for additional server instances that will be managed.

3.3.5 Configuring CAMM to Monitor WebSphere 5.1

For configuring CAMM to monitor WebSphere 5.1, set the following properties to false: Tomcat.Hosted=true

Start the Container manually using <CAMM_HOME>\apache-tomcat-5.5.20\

3.4 Data Export

There are 3 different modes to export performance data collected by CAMM to external databases and other persistence formats. These modes give users flexibility to choose the best way to extract performance data from CAMM. Both the metrics and events can be exported.

Export to File
In this mode, CAMM exports its raw performance data as several CSV (comma separated value) files.

Export to Database
In this mode, CAMM exports its raw performance data as several ANSI SQL statements. These SQL statements will allow users to create tables and insert data.

Aggregation Export to File
In this mode, CAMM exports its aggregated performance data after its daily aggregation operation as several CSV files.

How to Setup Export
To Export to Files or Export to Database, you need to create an xml file that describes the data you want and which way you want it (to file or database). There are sample files in *config/metric-export.xml* and *config/event-export.xml*. Here is what can be configured:

- In the **export** tag (second line), there is a parameter, **dataGrain**. This is the size of the sample that will be exported. It cannot be lower than the lowest data grain the manager stores (default 15secs)
- In the **output** tag (third line), the **type** is set to either **file** or **jdbc**. Each of these has different parameters:
    - If set to **file**, an **arguments** parameter is set to a filename prefix and the type of metric to export (**event** or **metric**). For example, in windows, it could read: **arguments=”c:/CAMM/export,metric”**

    - If set to **jdbc**, an **arguments** paramenter is set to the type of metric to export (**event** or **metric**), the driver name, and the url. For example, it could read **arguments="metric,com.mysql.jdbc.Driver,jdbc:mysql://localhost/acsera export"**
- A set of entityType tags need to be set, one for each table to be exported. For example, **<entityType name="J2EE.Servlet"/>**
- A set of filters can be setup to filter based on containerID or nodeID for example. These are setup in the filters section. For example, to filter on cgServer, you could enter a line **<filter key="containerID" values="cgServer"/>**

To execute this export, run the command **$CAMM_HOME/bin/runExportMetric.sh** or **$CAMM_HOME/bin/runExportEvent.sh**, with the parameters config-xml file, start time, and end time. For example,

$CAMM_HOME/bin/runExportMetric.sh ../config/metric-export.xml "12/15/05 12:00:00" "12/16/05 10:00:00".

If you are creating files, the files will appear in the **$CAMM_HOME/export** directory. If you are exporting to a database, new tables will appear in the database.

To export all tables, (also to see a list of all tables and their data), edit $CAMM_HOME/bin/runExportMetric.sh, and change the parameter **acsera.debugexport=false** to **true**.

To set up the Aggregation process to automatically do an export when aggregation happens, edit $CAMM_HOME/config/Acsera.properties, and change the line **AggregationManager.ExportMetrics = false** to **true**. Setting this will cause metrics to be exported every 24 hours with a granularity of 3 minutes. When using Aggregation Export, these variables are fixed.

What is Exported

Files (or tables) are created with names similar to this: **metric_j2ee_1_3m_1d_050921_1700J2EE_Servlet.csv**. "**3m**" means that the data in the table represents 3 minutes worth of data (the value that **dataGrain** was set to). "**1d**" means that this table contains one day's worth of data (i.e. 1 day was exported). "**050921_1700**" is the date and time of the export. "**Servlet**" is the type of data.

The files (or tables) created contain different data formats depending on the type of data exported. In the case of files, headers are inserted which make things more obvious. In some instances, multiple metrics are represented on a single line in a file or table. For example, the **JVM** file contains **ActiveThreads**, **HeapSizeCurrent**, **PhysMemFree** (among others) as entries on each row. On other files, multiple rows in the file (or table) may represent data from a single sample. For example, in the **ProcessNode** file, for each sample, there will be several rows, one for each process. And each of these Process rows will contain multiple metrics.

3.4.1 Enable/Disable Secure Communication

The default protocol for all communication is secure which should be fine in most cases. In very rare cases, a user may need to modify or disable the security settings. The sections below provide some details into the various settings that would need to be modified between the CAMM Manager, CAMM Admin UI (applet), and the CAMM agents. Please be warned that these should only be changed if absolutely necessary with the exception of the JMX communication which can easily and safely be modified via the resource configuration in the CAMM Admin UI. The exact communication protocol depends on the actual application server as can be seen below.

3.4.1.1 JMX Communication:

Oracle WebLogic: t3s – secure protocol, t3 – non-secure
Oracle SOA Suite: rmis – secure protocol, RMI – non-secure
IBM WebSphere: Enable security from Admin console, then SOAP uses security.

3.4.1.2 Enable/Disable Security for CAMM Manager – JavaAgent/OS Agent Communication:

The following property files enable and disable security between the CAMM Manager and the agents.

In Acsera.properties,
RMI.SSL.enabled=true – secure communication, set it to false for non-secure.

In XXXJAgent.properties
RMI.SSL.enabled=true – secure communication, set it to false for non-secure.

3.4.1.3 Enable/Disable Security for Manager – Client communication:

The following XML file enables and disables security between the CAMM Manager and the client.

In web.xml in
<QVHOME>apache-tomcat-5.5.20\webapps\qvadmin\WEB-INF

```
<init-param>
    <param-name>RMI.SSL.enabled</param-name>
    <param-value>true</param-value>
  </init-param>
```

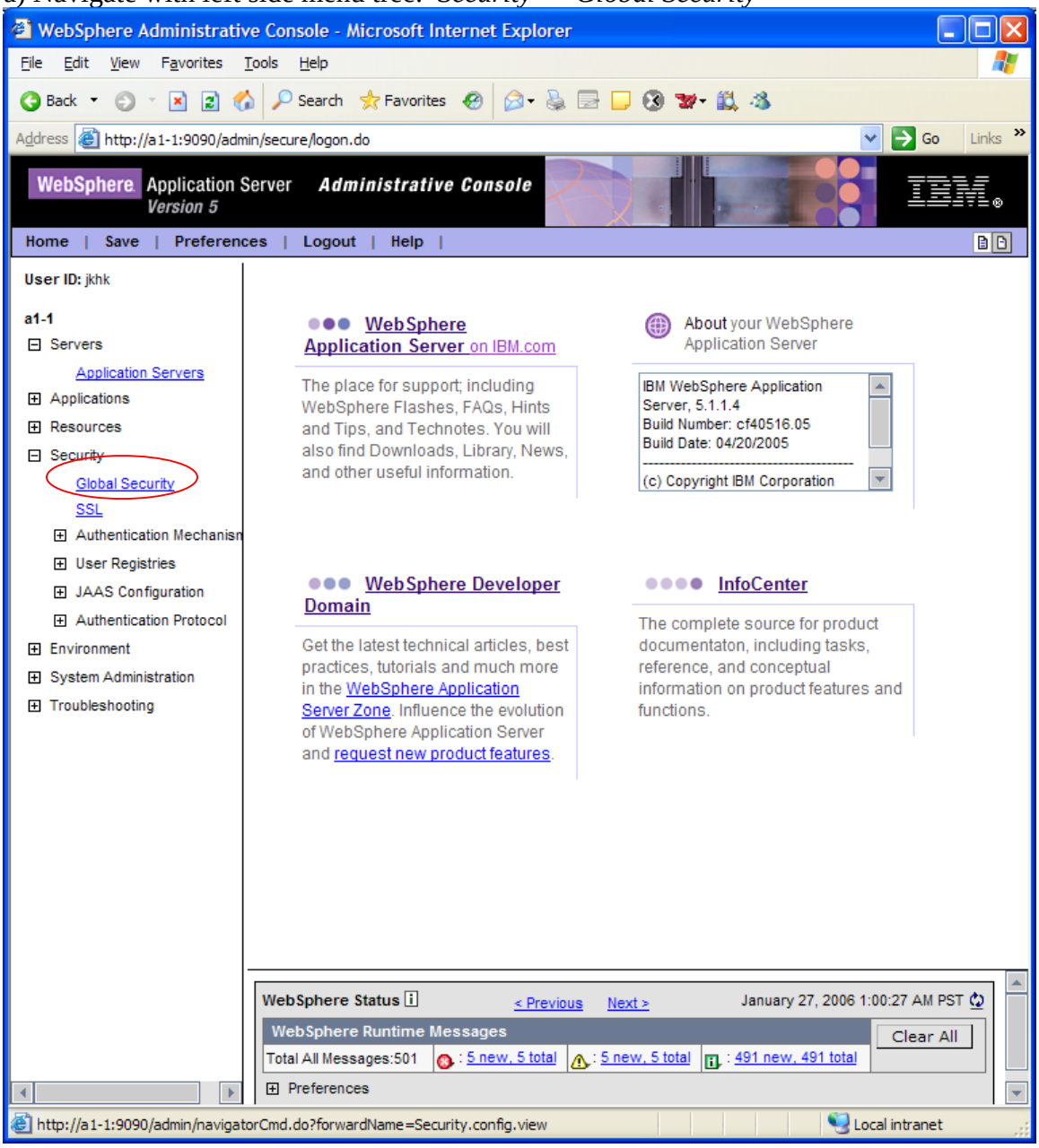
For non-secure, it has to be set to false. Remember the parameter is listed twice once for AcseraDataServlet & AcseraLogonServlet.

3.4.2 CAMM Configuration for WebSphere 5.1 Global Security

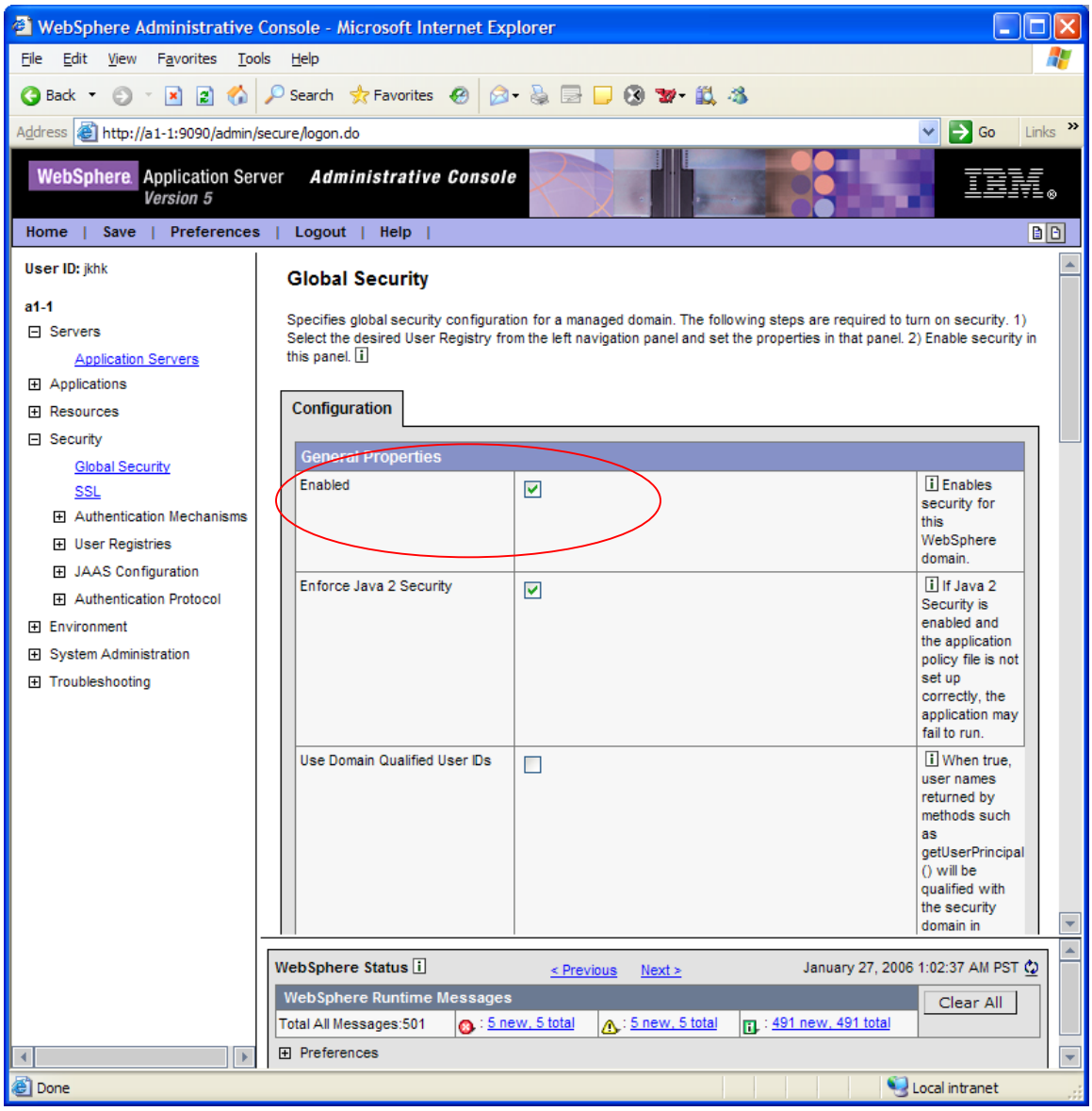
In order to run CAMM against WebSphere with enabled Global Security you need to follow the following procedures:

1) **Enable Global Security** in WebSphere 5.1 using the WebSphere admin console. This will enable all ports to WebSphere and the admin console to run with authentication and encryption.

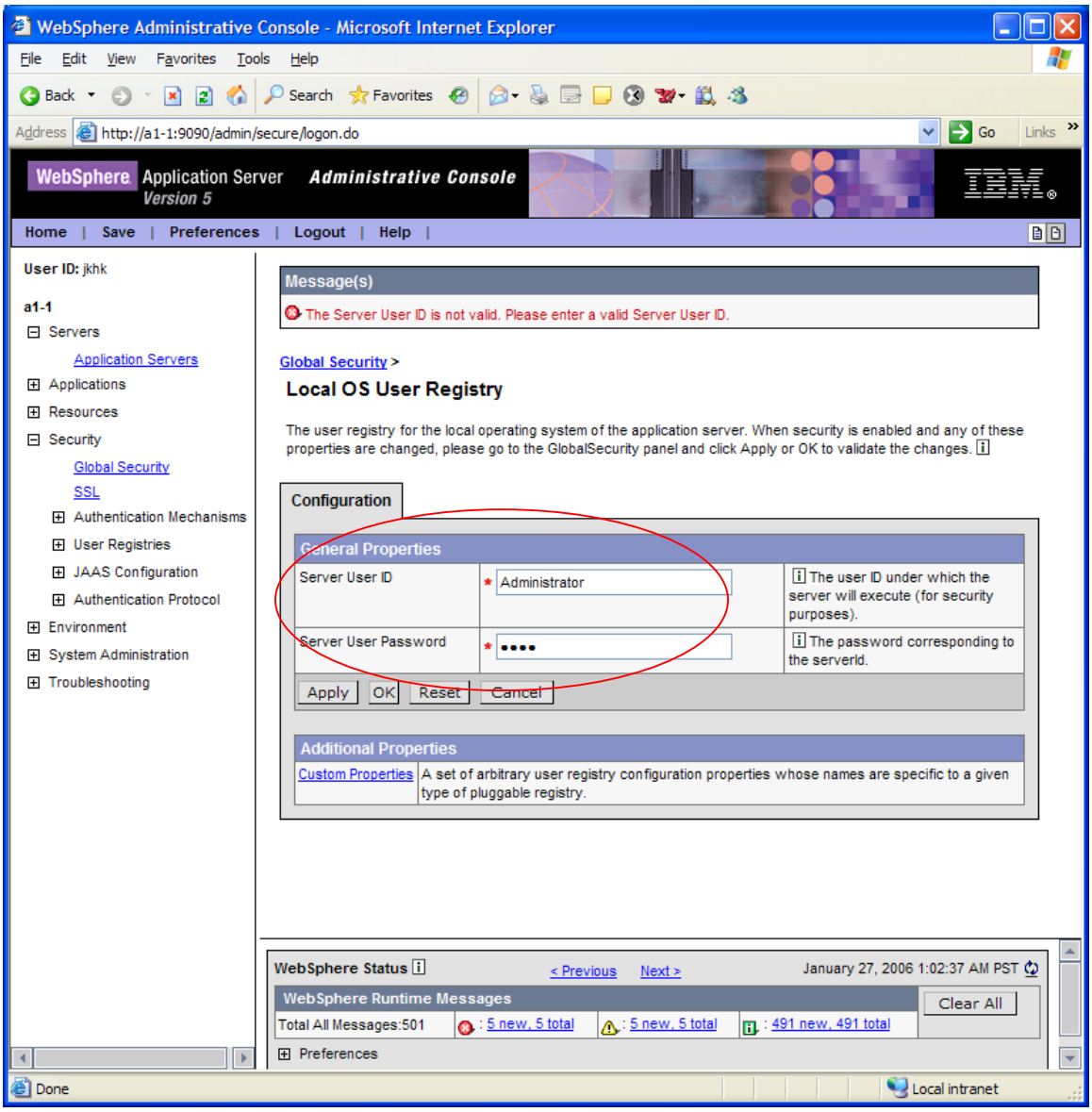
a) Navigate with left side menu tree: *Security -> Global Security*



b) Click on the Enabled checkbox to enable WebSphere global security for node.



c) It may prompt you for the OS root or administrator username/password, if these have not been updated before.



2) **Update server side WebSphere property** files in [WAS_HOME]/properties: **soap.client.props**

```
com.ibm.SOAP.securityEnabled=true
com.ibm.SOAP.loginUserid=admin
com.ibm.SOAP.loginPassword=test

com.ibm.ssl.keyStore=[WAS_HOME]/MDAMAgent/config/MDAMJavaAgentKey.jks
com.ibm.ssl.keyStorePassword=acserajava

com.ibm.ssl.trustStore=[WAS_HOME]/MDAMAgent/config/MDAMJavaAgentTrust.jks
com.ibm.CORBA.securityEnabled=true

com.ibm.ssl.keyStoreType=JKS
com.ibm.ssl.keyStore=[WAS_HOME]/MDAMAgent/config/MDAMJavaAgentKey.jks
com.ibm.ssl.keyStorePassword=acserajava
```

```
// Allow the MDAM Agent all permissions
grant codeBase "file:${was.install.root}/MDAMAgent/lib/-" {
        permission java.security.AllPermission;
};

// Allow the MDAM Deployer EJBs all permissions
grant codeBase "file:${was.install.root}/installedApps/[node]/[MDAM app name].ear/-" {
        permission java.security.AllPermission;
};
```

Normally, using the websphereDeployer command, the CAMM deployer EJBs would be deployed in the WebSphere server environment with the application name of the form:

CAMM_<node name>_<server name>

For example, this is an application name of a *deployer* deployed on node *a6-7* and server *WebSphere_Portal*.

CAMM_a6-7_WebSphere_Portal

Configure CAMM for security:

3) Update configuration on CAMM side.

a) Update *$ CAMM_HOME/config/Acsera.properties*:

The property values shown in the following are set by default. If a WebSphere installation is configured with a custom set of certificates, these properties will need to be updated. The path of the JKS files are specified as relative to the *$CAMM_HOME/config* folder.

```
// Allow the MDAM Agent all permissions
javax.net.ssl.keyStore=MDAMManagerKey.jks
javax.net.ssl.keyStorePassword=acseramanager
javax.net.ssl.keyStoreType=JKS
javax.net.ssl.trustStore=MDAMManagerTrust.jks
javax.net.ssl.trustStorePassword=acseramanager
javax.net.ssl.trustStoreType=JKS
```

Notice that if customer is already using the corporate certificates then you need to copy the key and trust client files to the *$CAMM_HOME/config* file and update the Acsera.properties file accordingly. The KeyStore and TrustStore passwords also need to be supplied.

b) Update *CAMM_HOME/config/configuration.xml*:

The username and password parameters for the WebSphere resource need to be updated since these values will be used to authenticate the manager with WebSphere.