Oracle® Composite Application Monitor and Modeler

Installation and Configuration Guide

Release 10.2.0.5

E14147-02

April 2009

Oracle Composite Application Monitor and Modeler Installation and Configuration Guide, Release 10.2.0.5

E14147-02

Contributing Author: Jacqueline Gosselin

Contributor: Swati Choudhary, Glen Hawkins, Maheshwari Jadamali, Arivalagan Kaliyaperumal, James Kao, Robert Liu, Karthik Somasundaram

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface

This guide provides detailed information and procedures for installing and configuring Oracle Composite Application Modeling and Monitoring (henceforth referrred to as CAMM). CAMM is a production services monitoring and performance reporting product that can dramatically improve your ability to track the performance and efficiency of complex server-side applications deployed on popular J2EE application server platforms. CAMM reduces the amount of time, effort, and errors associated with the typical manual process associated with setting up and maintaining an Model-driven Application Management (MDAM) system.

Audience

This document is intended for personnel responsible for the installation and configuration 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.

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.

Deaf/Hard of Hearing 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. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Related Documents

For more information, see the following documents in the *Oracle Composite Application Modeling and Monitoring (CAMM)* Release 10.2.0.4 documentation set:

- *Oracle Composite Application Modeling and Monitoring Release Notes*
- *Oracle Composite Application Modeling and Monitoring User's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Quick Steps for Installing the CAMM Agent

The following list summarizes the steps used to install the CAMM Agent (Agent). For details of the installation, refer to the chapters in this manual.

1. Collect the credentials for database and application server installation
2. Unpack the zip file
3. Run Disk1/InstData/<platform>/VM/install.bin
4. Select the location to install into when prompted, do not create links
5. Start the configuration console by running em10g/bin/config.sh
6. Connect a web browser to https://<server>:5560/qvadmin
7. Log in as admin/admin (user name/password), you will be prompted to change the password on login
8. Go to the **Configuration** tab and click **Database Configuration**
9. Select the default **MyOracle** connection and click **Edit Configuration**
10. Enter your database access credentials
11. Click **Test Connection** to verify database connection and then click **OK**
12. If you are monitoring a WebLogic server, deploy em10g/deploy/HttpDeployer.ear to your admin server. This step is optional for OC4J.
13. Return to the CAMM UI, click **Resource Configuration**
14. Click **Create New Resource**
15. Click **Configure** and enter your application server access information
16. Click **OK** to return to the previous screen
17. Click **Test Connection**. If you are monitoring a WebLogic server, wait for the em10g/lib/bea/<version>_runtime directory to be created and populated with jar files. This could take a few minutes. If nothing happens, check log/manager-log#.csv for exceptions.
18. Click **Back** to return to the previous screen and then click **Deploy**
19. On the subsequent screen, select **Deploy** from the Command menu and then select the servers/cluster you want to deploy the Agent to. Ensure that the admin server is always selected.
20. Click **Continue**
21. Wait for deployment to complete. Verify that the UI reports **Deployment Successful**; if not, check for exceptions in logs/manager-log#.csv

22. Click **Back** to return to the previous screen and then click **Save**
23. Close the browser-based console
24. Shut down the configuration console by running em10g/bin/acshut.sh
25. Wait about 15 seconds for the shutdown to complete
26. Restart the application servers you deployed the Agent to
27. Start the CAMM Manager by running em10g/bin/acsera.sh (for example, nohup ./acsera.sh &")
28. Wait for discovery to occur. You can check that discovery is happening by looking to see if em10g/darchive/server_archive is created and growing, and then that em10g/darchive/app_archive is created and growing. If you thing something is going wrong, you can check log/manager-log#.csv for exceptions, but this process can take several minutes.
29. Connect a web browser to https://<server>:5560/qvadmin
30. Click the CAMM node in the navigation tree and verify in the Agent Information table that the Agent Status for all servers you deployed the Agent to are REPORTING

1 CAMM Operation and Environment Dependencies

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

To guarantee the successful deployment and operation of CAMM, you are 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.

1.1 CAMM Operation Modes

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

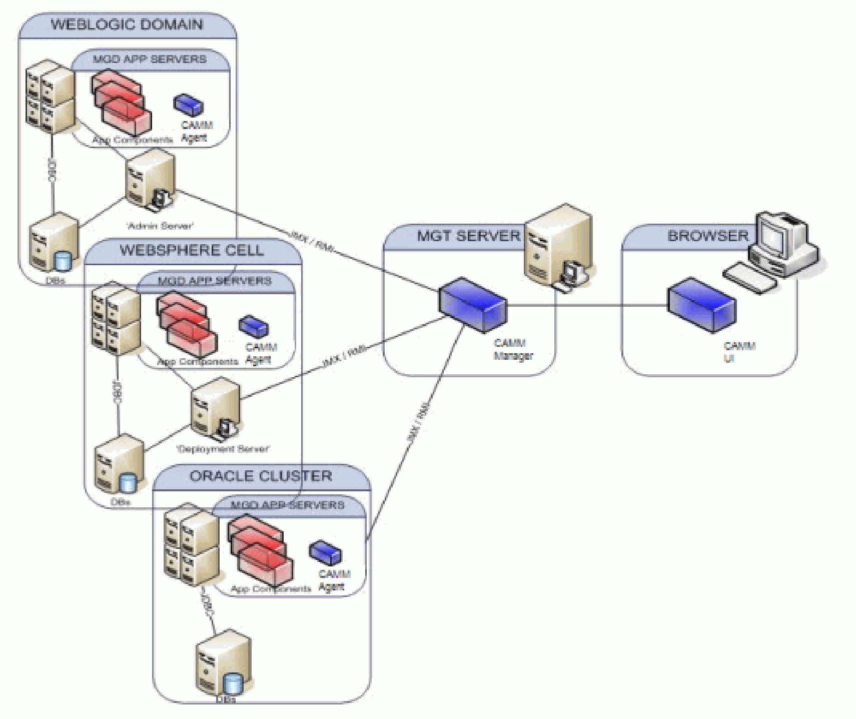
1.1.1 Service Mode

CAMM typically runs as a headless Java process that monitors your Oracle WebLogic, Oracle SOA Suite, and/or IBM WebSphere environments. In service mode, monitoring of your applications continues in the background, even when the user interface is not present. Service Mode is the default mode.

The user interface is delivered as a Java applet in a web browser.

Figure 1–1 visualizes the **Service Mode CAMM** topology:

Figure 1–1 Service Mode CAMM Topology



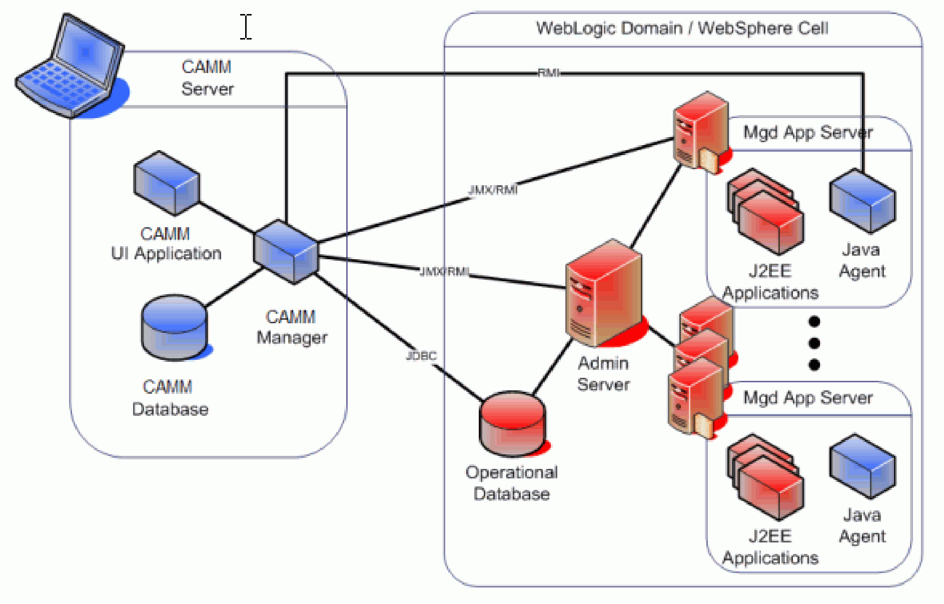
1.1.2 Standalone Application Mode

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. The Standalone Application Mode is used primarily for quality assurance and demonstration purposes.

When you start 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 deployed in **Standalone Application Mode** monitoring an Oracle WebLogic and IBM WebSphere application server environment:

Figure 1–2 Application Mode CAMM Topology

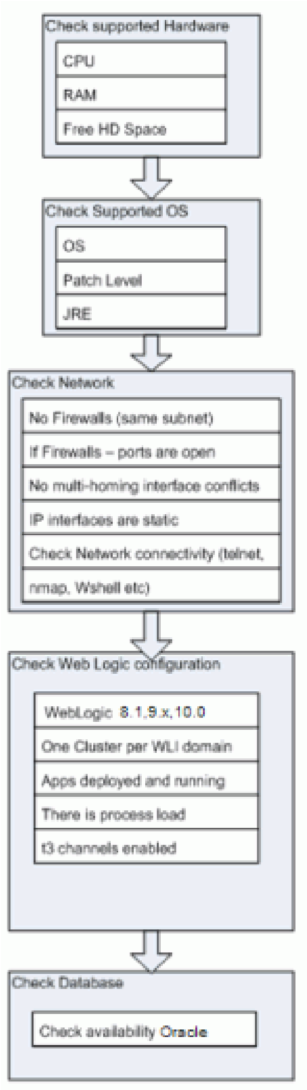


2

Pre-Installation Requirements

To save yourself time during the installation and configuration processes, check compliance with the following dependencies. The following text uses examples of environment dependencies for WebLogic 8.1, 9.2, and 10.0.

Figure 2–1 Environment Dependencies Check


Check supported Hardware
CPU
RAM
Free HD Space
Check Supported OS
OS
Patch Level
JRE
Check Network
No Firewalls (same subnet)
If Firewalls – ports are open
No multi-homing interface conflicts
IP interfaces are static
Check Network connectivity (telnet,
nmap, Wshell etc)
Check Web Logic configuration
WebLogic 8.1,9.x,10.0
One Cluster per WLI domain
Apps deployed and running
There is process load
t3 channels enabled
Check Database
Check availability Oracle

2.1 Environment Considerations

Consider the following environment considerations:

- Access and Connectivity
- CAMM Database
- System Locale

2.1.1 Access and Connectivity

Determine whether you have access and connections to the monitored application platform, network, and DHCP.

2.1.1.1 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 (such as ability to deploy to the domain)
- System Administration capabilities 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 (such as 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.

2.1.1.2 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.

Oracle recommends that the CAMM dedicated machine be deployed into the same subnet, within the firewall perimeter.

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

> **Note:** CAMM does not support IPv6 addressing.

2.1.1.3 DHCP

Oracle recommends 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.

2.1.2 CAMM Database

Ensure that the CAMM database is properly installed and configured.

CAMM stores application metadata, performance and availability metrics in a repository database. This dedicated database must be accessible by CAMM. Currently, CAMM supports Oracle 10*g* 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 Section 3.6.2, "Configuring MySQL DBMS for CAMM".

2.1.3 System Locale

CAMM supports 4 locales: 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 type the following at the command line before running the batch job:

1. Spanish:

   Set / export ACSERA_LANG=es

   Set / export ACSERA_COUNTRY=ES

2. Chinese Simplified:

   Set / export ACSERA_LANG=zh

   Set / export ACSERA_COUNTRY=CN

3. Chinese Traditional:

   Set / export ACSERA_LANG=zh

   Set / export ACSERA_COUNTRY=TW

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

1. Spanish:

   ACSERA_LANG=es

   ACSERA_COUNTRY=ES

2. Chinese Simplified:

   ACSERA_LANG=zh

   ACSERA_COUNTRY=CN

3. Chinese Traditional:

   ACSERA_LANG=zh

   ACSERA_COUNTRY=TW

The steps for changing the environment variables on Windows are as follows:

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 do not need to set the variable at the command line for standalone or client or acsera batch files.

2.2 Clustering and Application Domains

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 CAMM 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 (such as 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 technically feasible but not recommended, since traffic patterns from the different environments tend to be different (live versus load testing) and can complicate advanced data analyses. In this case, deploying multiple instances of CAMM would be the correct solution, where a dedicated CAMM instance monitors each specific environment (Production, Staging, QA, and so on). You can find details about multiple CAMM instances support in Section 3.5, "Deploying CAMM Components".

2.2.1 WebLogic Clustering

WebLogic clustering includes clusters and the Node Manager.

2.2.1.1 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.

2.2.1.2 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 detects that Node Manager is running and make those changes to the Node Manager server JVM startup parameters.

2.3 Statistics and Privileges To Obtain From Various Administrators

To save time during the installation and configuration of the CAMM Manager and CAMM Agent, get the privileges and field information you need *before* you start the process.

2.3.1 Database Administrator

Contact the Database Administrator (DBA) for the information needed to fill in the fields in the Resource section of the installation. You need the following values for the Oracle database instance:

- Database SID (service identifier)
- Database host and port
- Database user with CONNECT and RESOURCE roles

2.3.2 WebLogic Administrator

Contact the WebLogic Administrator (WebLogic Admin) for the following values:

- Admin Console host and port
- WebLogic administrator username and password
- If using the Oracle SOA Suite: BPEL and ESB console username, password, host, and port

2.3.3 WebSphere Administrator

Contact the WebSphere Administrator (WebSphere Admin) for the following values:

- Network Deployment Manager (dmgr) SOAP port
- WebSphere administrator username and password
- Client certificates (if global security is enabled)

2.3.4 OC4J Administrator

Contact the OC4J Administrator (OC4J Admin) for the following values:

- Application Server Control host and port
- OC4J administrator username and password
- OPMN request port
- If using the Oracle SOA Suite: BPEL and ESB console username, password, host, and port

3 Installation Process

This chapter describes the CAMM installation process in detail. Topics include:

- Types of Installations
- Installing CAMM - Windows Installation
- Installing CAMM - Solaris, Linux, and AIX Installations
- Setting Up CAMM Data Repository
- Deploying CAMM Components
- Installing and Configuring CAMM Data Repository
- Backing Up the CAMM Database

3.1 Types of Installations

When installing CAMM, you have the following options:

- CAMM as a Service

  Choosing this option installs CAMM as a Service. For Windows, CAMM creates two Windows Services: the CAMM service and the web container service. As Windows Services, CAMM and the Administration Server can be controlled by the standard Windows Services administration console (accessible by the Settings/Control Panel/Administrative Tools) and be configured to start automatically when the host restarts. This is the default mode and is used on a machine dedicated to monitoring and managing one or more application servers on a continuous basis.

- CAMM as a Standalone Application

  Choosing this option installs CAMM as a standalone Application. The CAMM application starts and stops when you start it or exit from it. 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 Manager can be installed as a Windows Service later.

3.2 Installing CAMM - Windows Installation

Perform the following steps to install CAMM on Windows. Note that at any time during the installation, you can click **Cancel** to cancel the installation, or click **Previous** to go back a screen.

1. Insert the CAMM CD-ROM into your CD-ROM drive. The installation automatically starts. Choose the language in which you want the installation instructions.

   The Introduction screen appears.

2. Read the introduction text and click **Next**.

   The Choose Install Set screen appears.

3. On the Install Set screen, choose to install either the Service Mode or the Standalone Application Mode. The default is the Service Mode.

   The Install Folder screen appears.

4. Click **Next**. On the Choose Install Folder screen, indicate which folder will contain the installation. Either type the folder's location or accept the default C:\oracle\em10g folder.

5. Click **Next**. On the Choose Shortcut Folder screen, choose a Shortcut Folder in which to copy the CAMM product icons. If desired, select the **Create Icons for All Users** box.

6. Click **Next**. The Pre-Installation Summary screen lists the options to be installed. If an option is not correct, click **Previous** to move back and correct the option. When the summary is correct, click **Next**.

3.3 Installing CAMM - Solaris, Linux, and AIX Installations

Perform the following steps to install CAMM on your UNIX environment. Note that at any time during the installation, you can type **back** to go back, and can cancel the installation by typing **quit**.

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

2. Change directory to where the CD-ROM is mounted.

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

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

   For Solaris, change the 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: Ensure that *install.bin* has execution privileges and, if necessary, use *chmod a+x install.bin* command to make it executable.

The Introduction screen appears. Choose the language in which you want the installation instructions.

3. Read the introduction text and press the <ENTER> key to continue.

   The Install Set text appears.

4. From the Install Set text, choose to install either the Service Mode or the Standalone Application Mode. The default is the Service Mode.

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

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

  ->1- "Oracle CAMM"
    2- "Orache CAMM as Application"

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

5. Click **Next**. From the **Install Folder** text, indicate which folder will contain the installation. Either type the folder's location or accept the default $HOME/oracle/em10g.

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

Where would you like to install?

  Default Install Folder: /root/oracle/em10g

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

   Choose the directory in which to install CAMM and press the <ENTER> key. You will be prompted of where to install the CAMM product icons.

6. Click **Next**. Indicate which directory will contain the Link locations.

```
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 desired directory and press the <ENTER> key. You will be shown a summary of your installation choices

Choose option 4 - **Don't create links**. Run startup scripts out of the bin directory as the install location.

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

8. Press <ENTER> to finish after the installation completes.

3.4 Setting Up CAMM Data Repository

CAMM requires an RDBMS be setup as the data repository for runtime metrics collection. Both Oracle 10*g* and MySQL RDBMS are supported for this purpose. Details for manual installation and configuration of the Data Repository are described in Section 3.6, "Installing and Configuring CAMM Data Repository".

3.5 Deploying CAMM Components

Deploying CAMM components involves the following steps:

- Configuring CAMM with information about the target application platform. 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 (for example, managed servers in a cluster of a domain, and so on).

3.5.1 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*.

Oracle highly recommends that you register and update the configuration of your server resources through the CAMM Manager.

1. Start the CAMM Manager.
   - Windows: Start Oracle CAMM.
   - UNIX: In the /bin directory, invoke the following command:

     nohup ./acsera.sh &
2. Click the **Configure** tab on the left pane (which contains the navigation tree).
3. Click the **Resource Configuration** node in the navigation tree.
4. Click the **Create New Resource** button in the main pane.
5. Name your resource and select your application server product and version.
6. Click **Continue**.
7. Click **Configure** in the middle of the main pane.
8. Type in your application server details. Ignore the BPEL and ESB related options if you do not have the Oracle SOA Suite.

> **Important:** DO NOT modify the fields for Agent Keystore Password and Agent Truststore Password.

9. Click **OK**.

10. Click **Save**. (If you omit this step, your configuration will be lost.)

3.5.2 Deploying CAMM Agents on the WebLogic Platform

The following sections describe how to deploy CAMM Agents on the WebLogic platform.

3.5.2.1 HttpDeployer.ear Procedure

Use the WebLogic Admin console to deploy the em10g/deployHttpDeployer.ear application.

3.5.2.2 Deploying the CAMM Java Agent

Using the GUI-based deployer as documented in the Quick Steps for Installing the CAMM Agent at the beginning of this manual is recommended. There exists a command-line tool with equivalent behavior. The documentation follows.

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

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

- **Usage:**

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

Where:

Attribute	Description
`command`	Is either `-deploy, -ejbdeploy, -copy, -enable, -status, -systemprops, -disable, -remove, or -ejbundeploy`
`-version`	Specifies WebLogic version. For -version, use: 8.1.3 for WebLogic 8.1.3 8.1.4 for WebLogic 8.1.4 8.1.5 for WebLogic 8.1.5 and higher 9.2.0 for WebLogic 9.2.x 10.0.0 for WebLogic 10.0.x 10.2.0 for WebLogic 10.2.x 10.3.0 for WebLogic 10.3.0
`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.

- **Commands Supported:**

Command	Description
-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).

Command	Description
-copy	Deploys the CAMM EJB only (typically only used for debugging purposes).
-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.

- **Actions:**
  - **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 -version 8.1.3 -status <targets> -resource 'resource name'
    ```
  - **Disabling a Running Agent**

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

    ```
    bin/oracleDeployer.bat -version 8.1.3 -disable -resource <resource name>
    -targets
    ```

    After un-deploying the agent, you must reboot the target application server. This procedure is valid for all application servers.

    Note: If upgrading the Deployer EAR, make sure you undeploy the older version from admin/managed servers.
  - **Explicit Un-Deployment of the CAMM Enterprise JavaBean (EJB)**

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

    ```
    bin/deployer.bat -version 8.1.3 -ejbundeploy -resource 'resource
    name'-targets <targets>
    ```

    This will undeploy the CAMM EJB from the administration 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 -version 8.1.3 -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 -version 8.1.3 -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 -version 8.1.3 -copy -resource 'resource name'-targets C1
    -admindir CAMMAgentUpgrade

    bin/deployer.bat -version 8.1.3 -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 -version 8.1.3 -status -resource 'resource name'-targets
    C1
    ```
  - To undeploy the CAMM Agent:

    ```
    bin/deployer.bat -version 8.1.3 -disable -resource 'resource name'-targets
    C1
    ```

    Restart the C1 servers. This will unlock the CAMM Agent files.
  - To remove the CAMM Agent:

    ```
    bin/deployer.bat -version 8.1.3 -remove -resource 'resource name'-targets
    C1
    ```
  - To undeploy the CAMM EJB from all servers:

    ```
    bin/deployer.bat -version 8.1.3 -ejbundeploy -resource 'resource name'
    ```

3.5.2.3 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 -version <version> -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 `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 the JDK used for Managed Servers is based on JRockit you do not need to install OS Metric. CAMM uses JMX services of JRockit to retrieve OS Metrics.

- **Actions:**
  - 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 osmetric -version 8.1.3 -disable -resource 'resource
    name'-targets <targets>
    ```

  - Removing 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 -version 8.1.3 -remove -resource 'resource
    name'-targets <targets>
    ```

  - Explicit un-deployment of the CAMM OS Agent

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

    ```
    bin/deployer.bat osmetric -version 8.1.3 -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.

3.5.3 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

3.5.3.1 Required IBM WebSphere Libraries

Once WebSphere is registered via the CAMM Administration UI, the libraries required by CAMM to connect to WebSphere must be defined in the actual classpath. This is done through the Resource Configuration UI.

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>
```

3.5.3.2 Automatic CAMM Agent Deployment for WebSphere Platform

The following sections explain how to automatically deploy the CAMM Agent for the WebSphere platform.

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), ensure 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 CAMM Agent is being installed manually, the file transfer application is not required.

Deploying CAMM Java Agent for WebSphere Platform

Using the GUI-based deployer as documented in the Quick Steps for Installing the CAMM Agent at the beginning of this manual is recommended. There exists a command-line tool with equivalent behavior. The documentation follows.

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

- **Usage:**

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

  Where:

Attribute	Description
`command`	Either `-deploy, -ejbdeploy, -copy, -enable, -status, -systemprops, -disable, -remove,` or `-ejbundeploy`
version	Indicates version of WebSphere to use ■ 5.1.0 for WebSphere 5.1.x ■ 6.0.0 for WebSphere 6.0.x ■ 6.1.0 for WebSphere 6.1.x
`resource`	Name of the resource that was configured in the CAMM Manager. The name must be exact and is case sensitive.
`targets`	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`	Name of the agent dir to use (no paths allowed, just a file name). For example: CAMM_Agent

- **Commands Supported:**

Command	Description
`-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

Command	Description
`-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.

- **Actions:**
  - **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 the Deployed Agent**

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

    ```
    bin/websphereDeployer.bat -version <version> -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 -version <version> -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 -version <version> -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.

- **Example:**

  To deploy to the admin server on localhost (to upgrade the Agent, use the same command):

  ```
  bin/websphereDeployer.bat -version 5.1.0 -deploy -resource
  'resource name' -targets WebSphere_Portal -agentdir
  AcseraAgent
  ```

  Restart WebSphere_Portal for this to take effect.

3.5.3.3 Deploying CAMM Java Agent for Oracle Platform

Using the GUI-based deployer as documented in the Quick Steps for Installing the CAMM Agent at the beginning of this manual is recommended. There exists a command-line tool with equivalent behavior. The documentation follows.

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

- **Usage:**

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

Where:

Attribute	Description
`command`	Either `-deploy, -ejbdeploy, -copy, -enable, -status, -systemprops, -disable, -remove`, or `-ejbundeploy`.
version	Indicates version of Oracle AS to use ■ 10.1.3.1 for Oracle AS 10.1.3.1 ■ 10.1.3.3 for Oracle AS 10.1.3.3 ■ 10.1.3.4 for Oracle AS 10.1.3.4
`resource`	Name of the resource that was configured in the CAMM Manager. The name must be exact and is case sensitive.
`targets`	Comma-delimited list of group and instance names (no spaces allowed). Instance names should be preceded by their group name. For example: `default_group/oc4j_soa` for an instance, or `default_group` for all instances in the group.
`agentdir`	Name of the agent dir to use (no paths allowed, just a file name). For example: CAMM_Agent.

- **Commands Supported:**

Command	Description
`-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.

- **Actions:**
  - **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 -version <version> -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 -version <version> -undeploy
    -resource 'resource name' -targets <targets>
    ```

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

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

  ```
  bin/oracleDeployer.bat -version 10.1.3.1 -deploy -resource
  'resource name' -targets default_group/oc4j_soa -agentdir
  AcseraAgent
  ```

  Restart Oracle SOA Suite for this to take effect.

3.5.4 Running Multiple CAMM Instances

There will always be a *default* instance of the CAMM (the installed instance). For each additional Manager instance needed, first create an instance directory underneath ACSERA_HOME. (ACSERA_HOME is the directory that CAMM was installed into, for example, C:\oracle\em10g.) Copy the installed config, mccconfig, and schema directories under the new instance directory.

In INSTANCE_DIR/config/Acsera.properties, set the following properties to unique, instance-specific values:

- RMI.Registry.Port = other than the original one (51099 by default)
- RMI.RemoteServiceController.ServerPort= other than the original one (55000 by default)
- RMI.JavaProvider.ServerPort= other than the original one (55003 by default)
- RMI.Registry.OSMetrics.Port= other than the original one (51099 by default)
- Tomcat.Hosted=false (for original and instances)

The default value for RMI.Registry.Port is 51099. It is recommended that additional Manager instances use values counting down from the default (51098, 51097, and so on.)

Update dbconfig.xml for the target database (use a different database schema for the instance).

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.

The following commands may be used to control non-default Manager instances:

Windows	UNIX
bin/acsera.bat	bin/acsera.sh
bin/acshut.bat	bin/acshut.sh
bin/deployer.bat	bin/deployer.sh
bin/standalone.bat	bin/standalone.sh

The acsera.*, acshut.* and standalone.* scripts take an instance name as the first argument. Leaving out the instance name causes the script to target the default instance.

3.5.4.1 Front-End Configuration

Create a new qvadmin directory for instance on \ACSERA_HOME\apache-tomcat-5.5.20\webapps. For example:

```
\ACSERA_HOME\apache-tomcat-5.5.20\webapps\qvadmininstance1
```

Copy contents of webapps/qvadmin to the new directory.

In each of the previous manager instances, set a unique RMI address/port. You will also need to set up each of the qvadmin directory's web.xml to match.

For each qvadmin directory in apache-tomcat-5.5.20/webapps, edit the WEB-INF/web.xml file as follows:

- Edit the Acsera.RMIRegistry.HOST value to reflect the address of this instance of the Manager. The default is localhost, and this is the only supported configuration.
- Edit the Acsera.RMIRegistry.Port value to reflect the port of this instance of the Manager. Note that each qvadmin/Manager pair must have a unique port if the Manager exists on the same machine with other Managers.

Each of these qvadmin deployments can be in the same container or in separate containers. If they are in the same container, you eed to give each a unique name, such as qvadmin1 (for an exploded directory) qvadmin1.war (for a war file), so they do not conflict. In Tomcat, this is done by copying qvadmin to its new name, and restarting Tomcat.

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):

```
https://1.2.3.4:5560/qvadmin1    Manager 1 (Domain 1)
https://1.2.3.4:5560/qvadmin2    Manager 2 (Domain 2)
and so on
```

Deploying in multiple containers could result in URLs (note different ports):

```
https://1.2.3.4:5560/qvadmin
https://1.2.3.4:5561/qvadmin
and so on
```

3.6 Installing and Configuring CAMM Data Repository

CAMM maintains a database (can be Oracle or MySQL) of information collected on the system it is monitoring. This database can be contained on the same machine on which CAMM is running. 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.

3.6.1 Configuring Oracle DBMS for CAMM

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 "CAMM" or something distinct)
3. Set the System Global Area to have at least 1 GB (1275068416 bytes)
4. Increase the number of processes in database from 150 to 300 (optional; this is primarily for an OracleXE database).

   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 number of processes. You can set it to a higher number 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)

Use of the GUI-based configuration tool described in the Quick Steps for Installing the CAMM Agent at the beginning of this manual is recommended. For manual database configuration, fields in dbconfig.xml may be modified.

3.6.2 Configuring MySQL DBMS for CAMM

Oracle CAMM currently supports MySQL 4.1 database and higher for its runtime repository. The following steps are necessary to setup and configure the Oracle DBMS for Oracle CAMM. Oracle CAMM initializes the database upon connecting to it and generates the required tables.

1. Install MySQL 4.1 or higher on a separate machine
2. Create a new user in the 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
   ```

3.7 Backing Up the CAMM Database

Oracle recommends that regular backups be made of the database utilized as a runtime repository for Oracle CAMM. The Oracle database provides the EXP and IMP tools that allow you to quickly backup 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.

4

Post-Installation Requirements

This chapter explains the following post-installation requirements:

- IBM WebSphere Post-Deployment Requirements
- Configuring Oracle SOA Suite for Secure Connectivity
- Configuring Oracle WebLogic Server or Oracle WebLogic Portal (WLP) for Secure Connectivity
- Configuring WebSphere 5.1 for Secure Connectivity
- Importing a Certificate into the Manager's Keystore
- Configuring the CAMM Agent When WebLogic Is Installed As a Windows Service

4.1 IBM WebSphere Post-Deployment Requirements

The following post-deployment requirements are specific to CAMM deployments on IBM WebSphere Application Server.

4.1.1 Configuring CAMM for WebSphere Application Server 6.1 Secured Connections

The main goal is to add the signer certificate of each administration server to CAMM's truststore, which is needed by each resource to connect to the server. This allows CAMM to trust the server when making secured (SSL) connections to the server. Without this trust, the SSL handshake will fail.

When using the default CAMM 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 administration server's signer certificate for resource.

   If the administration server is the deployment manager in a WebSphere Application Server ND, export signer certificate from trust.p12 of the deployment manager located at path

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

   If the administration server is a standalone server, export the signer certificate from trust.p12 of the standalone server located on the following path:

   ```
   <WAS_
   HOME>\profiles\AppSrv01\config\cells\<CellName>\nodes\<NodeNa
   me>\trust.p12
   ```

   To export, run the following command:

```
JAVA_HOME/bin/keytool -export -keystore <trust path>
-storepass WebAS -storetype PKCS12 -alias default_signer
-file servercert
```

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

2. Import the administration server's signer certificate into the CAMM truststore.

   Import the exported certification according to the information in Section 4.4, "Importing a Certificate into the Manager's Keystore".

4.1.2 Configuring WebSphere 5.1 for Secure Connectivity

To run CAMM against WebSphere with enabled Global Security, perform the following steps:

1. Identify the com.ibm.ssl.keyStore and com.ibm.ssl.trustStore files in soap.client.props and sas.client.props in [WAS_HOME]/properties as follows:

   a. Copy the indicated keystore and truststore files to the CAMM Manager.

   b. Import the files following the instructions in Section 4.4, "Importing a Certificate into the Manager's Keystore".

Example 4–1 soap.client.props

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

com.ibm.ssl.keyStore=
com.ibm.ssl.keyStorePassword=acserajava

com.ibm.ssl.trustStore=
com.ibm.ssl.trustStorePassword=acserajava
```

Example 4–2 sas.client.props

```
com.ibm.CORBA.securityEnabled=true

com.ibm.ssl.keyStoreType=JKS
com.ibm.ssl.keyStore=
com.ibm.ssl.keyStorePassword=acserajava

com.ibm.ssl.trustStoreType=JKS
com.ibm.ssl.trustStore=
com.ibm.ssl.trustStorePassword=acserajava
```

2. If you encounter security exceptions in the CAMM EJB when the application server starts, you may need to update the [WAS_HOME]/properties/server.policy file and append the configuration that follows.

Example 4–3 server.policy

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

// Allow the Acsera Deployer EJBs all permissions
```

```
grant codeBase "file:${was.install.root}/installedApps/[node]/[Acsera 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:

```
Acsera_<node name>_<server name>
```

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

```
Acsera_a6-7_WebSphere_Portal
```

4.2 Configuring Oracle SOA Suite for Secure Connectivity

The Oracle SOA Suite may be configured to support RMIS (RMI over SSL) connectivity. In this case, CAMM can be configured to use this secure connection. To configure CAMM to do this, perform the following steps:

1. On the Oracle SOA Suite install, look at ORACLE_HOME/j2ee/<instance>/config/rmi.xml, locate the <ssl-config> element, and identify the path in the keystore attribute.
2. Copy the keystore file indicated to CAMM manager's config directory (for example, em10/config)
3. Import this keystore file following the instructions in Section 4.4, "Importing a Certificate into the Manager's Keystore".

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

To configure Oracle WebLogic Server 10.0 to handle connectivity using t3s, the location of the keystore files needs to be updated through the console.

1. Log in to the WebLogic Server console and select the servers under the 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 **Load & Edit** to update the Keystore
4. Make the following changes. Identify the keystore and truststore file paths from the following properties:

   Identity

   Custom Identity Keystore

   Trust

   Custom Trust Keystore: location of the trust file
5. Repeat steps 2 through 4 for additional server instances that will be managed.
6. Copy the identified keystore and truststore files to the CAMM manager.
7. Copy the BEA_HOME/license.bea to the CAMM manager's config directory (for example, em10g/config)

8. Import the keystore and truststore files following the instructions in Section 4.4, "Importing a Certificate into the Manager's Keystore"

9. Locate the following properties in the Acsera.properties file and set them as follows:

```
weblogic.security.TrustKeyStore=CustomTrust
weblogic.security.CustomTrustKeyStoreFileName=AcseraManagerTrust.jks
weblogic.security.CustomTrustKeyStorePassPhrase=acseramanager
```

4.4 Importing a Certificate into the Manager's Keystore

To import entries from a keystore or truststore, perform the following steps, replacing *ServerStoreFile.jks* with the keystore or truststore from your application server. (Skip steps 1 and 2 if you are importing certificate files from WAS 6.1 as described in Section 4.1.1, "Configuring CAMM for WebSphere Application Server 6.1 Secured Connections".) You will generally need to complete these steps twice, once for the keystore and once for the truststore.

1. List the key aliases in the keystore/trustfile file from the server

```
keytool -list -keystore ServerStoreFile.jks -storepass
DemoIdentityKeyStorePassPhrase
```

Output:

```
Keystore type: jks
Keystore provider: SUN

Your keystore contains 1 entry:

demoidentity, Wed Nov 19 13:34:56 PST 2008, keyEntry, Certificate fingerprint
(MD5): 36:06:C2:44:31:0A:28:FC:06:19:F7:AB:C0:7D:27:6A
```

2. Export a key entry to an intermediate file

```
keytool -export -alias demoidentity -keystore ServerStoreFile.jks -storepass
DemoIdentityKeyStorePassPhrase -file demo103
```

Output:

```
Certificate stored in file <demo103>
```

3. Import the key into the CAMM store file (either AcseraManagerKey.jks or AcseraManagerTrust.jks in the CAMM manager's config directory)

```
keytool -import -alias demoidentity1 -keystore AcseraManagerKey.jks -storepass
acseramanager -file demo103
```

Output:

```
Owner: CN=b91, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState,
C=US
Issuer: CN=CertGenCAB, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown,
ST=MyState, C=US
Serial number: 510fb3d4b2872e3a093d436fcbe9b24b
Valid from: Tue Nov 18 13:34:47 PST 2008 until: Sun Nov 19 13:34:47 PST 2023
Certificate fingerprints:
        MD5:  36:06:C2:44:31:0A:28:FC:06:19:F7:AB:C0:7D:27:6A
        SHA1: BB:85:6D:4C:0B:4A:92:63:CA:5E:E9:A8:54:42:80:2D:0D:BE:7C:91
Trust this certificate? [no]:  yes
```

```
Certificate was added to keystore
```

4. Verify that the key was imported successfully

   ```
   keytool -list -keystore AcseraManagerKey.jks -storepass acseramanager
   ```

   Output:

   ```
   Keystore type: jks
   Keystore provider: SUN

   Your keystore contains 3 entries:

   demoidentity1, Wed Apr 01 13:03:21 PST 2009, trustedCertEntry,
   Certificate fingerprint (MD5): 36:06:C2:44:31:0A:28:FC:06:19:F7:AB:C0:7D:27:6A
   demoidentity, Fri Mar 13 15:15:06 PST 2009, trustedCertEntry,
   Certificate fingerprint (MD5): 0B:11:02:B5:44:0D:2A:CC:7F:C5:30:5C:1A:C9:A1:6C
   mykey, Thu May 19 16:57:36 PDT 2005, keyEntry,
   Certificate fingerprint (MD5): 5D:B0:EC:28:14:33:26:1F:44:F5:BE:DD:A8:50:15:9D
   ```

5. Repeat steps 2 through 4 for each key entry listed in step 1.

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

4.5 Configuring the CAMM Agent When WebLogic Is Installed As a Windows Service

When the monitored WebLogic server is installed as a Windows Service, the automatic startup changes to deploy the CAMM Agent need to be manually applied to the registry entries that control WebLogic startup.

The parameters which need to be changed are in the Windows registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\$ServiceName\Parameters
```

Users should then consult the file on the CAMM Manager:

```
deploy/agent/bea/bin/agentoptions.bat (for WebLogic 8.1.x)
deploy/agent/bea9/bin/agentoptions.bat (for WebLogic 9.x and higher)
```

Inspect this file and resolve the net results of its execution as Parameters in the registry.

Note that the beaaj.jar named on the %EXT_POST_CLASSPATH% variable needs to be placed on the server's classpath.

A
Configuration Directories and Files

This appendix lists and defines the files and directories available in CAMM. Topics include:

- Configuration Directories
- Acsera.properties File
- UrlMap.properties

A.1 Configuration Directories

After CAMM is installed all the components of the application package are located in $CAMM_HOME directory.

A.1.1 Directory Structure

The following is the $CAMM_HOME directory structure once CAMM is installed:

-apache-tomcat-5.5.20
-bin
-config
-darchive (created when manager first discovers monitored application servers)
-deploy
-j2sdk
-lib
-log
-mcconfig
-tmp
-uninstall Oracle Composite Application Monitor and Modeler
-userdata

Table A–1 Directories in $CAMM_HOME Directory

Directory	Description
apache-tomcat-5.5.20	Tomcat server where CAMM Web Application (GUI) is located
bin	Contains all the executable files to start and stop CAMM, run deployer for Agent and CAMM EJB, run export utility.
config	Contains all the CAMM runtime configuration parameters that control execution logic, CAMM schemas enablement, CAMM GUI functionality, Service Level Objectives definition, export logic and many more.
darchive	Stores temporary images for J2EE applications that need to be monitored as well as the results of analysis and modeling of the J2EE application. This directory is created once CAMM is up and running.

Table A–1 (Cont.) Directories in $CAMM_HOME Directory

Directory	Description
deploy	Contains agent libraries and configuration files, as well as *CAMM EJB* and *CAMM Admin Web* Application. These components are deployed on the remote host (Web or Application servers) using *deployer* utility found in bin directory of CAMM package.
j2sdk	Contains minimum Java SDK sufficient to run CAMM Server.
lib	Has all the libraries required for CAMM's proper functionality
log	Has all the diagnostics records of CAMM performance metrics collection activities. Also logs indicating successful deployment of CAMM Agents can be found here. This directory is created once CAMM is up and running.
mcconfig	Contains internal base instrumentation configuration. Do not modify these files.
tmp	Contains all the metadata definitions used for CAMM's needs.
Uninstall CAMM	Contains utility used to uninstall the CAMM.
userdata	Contains saved custom views per user.

A.1.2 Config Directory

Config directory has many files that potentially can be configured and make CAMM to run in a particular way. Any changes applied to files in this directory require restarting CAMM server.

Most of the files never get touched directly by user. The following are the main three files which can be configured manually to achieve desired effect:

File	Description
Acsera.properties	This file is the main CAMM configuration file customization of which helps to tune up CAMM.
configuration.xml	In this file you define location of Administration Server and credentials to access it. Usually you do not touch this file. The entire configuration is done through CAMM GUI.
dbconfig.xml	This file contains database configuration information for the CAMM metric repository.
export.xml	This file contains information that drives proper data export logic. It is used for manual and automatic export of performance metric and events data from the CAMM Data Repository.
UrlMap.properties	This file is used to map server addresses to load balancer addresses.

It is worth mentioning that Service Level Objective definitions and Actions associated with the SLOs are described in slo.xml and event.xml respectively. The content of these files is completely controlled by definitions applied from CAMM GUI (configuration tab).

A.1.3 BIN Directory

The /bin directory has all the executable files to start and stop CAMM, run deployer for Agent and CAMM EJB, run export utility.

There are two main reasons why one would need to customize the content of the files in this directory:

- To change the pointers to the location of Java Runtime Environment and CAMM installation directory (ACSERA_HOME)

- To configure CAMM Server JVM parameters, e.g. memory.

Only the files that are frequently customized will be described in this chapter.

All the files in this directory should have these lines, pointing to the CAMM Installation location:

For UNIX: `ACSERA_HOME=/home/CAMM; export ACSERA_HOME`

For Windows: `set ACSERA_HOME=C:\oracle\em10g\`

In addition, acseraenv.sh(.bat) should have a pointer to the JDK used by CAMM:

- For UNIX: `JAVA_HOME=/home/oracle/em10g/j2sdk;export JAVA_HOME`
- For Windows: `set JAVA_HOME=C:\oracle\em10g\j2sdk`

CAMM is a Java application and runs in its own JVM. Default size of JVM memory is 600 MB. Should you want to change this value you can do so by modifying -Xms and -Xmx parameters in acsera.sh (.bat) file.

Bear in mind that if you have installed CAMM as a Windows service and need to change the JVM Memory size, you need to change the size either by updating the Windows Registry or rerunning the createmanagerservice.bat utility with the new JVM parameters.

A.1.4 Deploy Directory

The /deploy directory contains CAMM Java Agent and OS Metric Agent distributables, including configuration files as well as corresponding libraries. These files are copied to the target systems hosting the Managed Servers when running the deployer utility. Rarely one needs to modify configuration files in this directory. Remember though if you modify the files they will be distributed to ALL targets within single server/cluster.

A.1.5 apache-tomcat-5.5.20 Directory

The /apache-tomcat-5.5.20 directory has the same structure and content as the standard Tomcat distributable. One interesting point about it is that CAMM GUI application acseraadimn.war is residing in `em10g/apache-tomcat-5.5.20/webapps` directory.

A.2 Acsera.properties File

The acsera.properties file contains global configuration parameters that define the operation of the CAMM Manager.

A.2.1 Log Files Management

This section of Acsera.properties file defines log rotation policies. Log.MaxFiles indicates max number of log files available at any given moment, whereas the Log.MaxFileSizeMB indicates maximum size of the log file.

Example A–1 Log Files Management Section

```
Log.CopyOut = false
Log.MaxFiles = 10
Log.MaxFileSizeMB = 30
Log.MergeLogs = true

Debug.CopyOut = false
Debug.LogLevel = all
Debug.MaxFiles = 10
Debug.MaxFileSizeMB = 30
```

Log files are stored in the log directory.

A.2.2 Security Features Configuration

This section enforces the password policies and the CAMM user session time out.

A.2.2.1 User Password Management

The following are the parameters to enforce password policies:

- length
- expiration
- complexity
- allowed special characters

Example A–2 User Password Management

```
ConfigurationManager.PasswordMinLength=0
ConfigurationManager.PasswordMaxLength=0
ConfigurationManager.PasswordExpDays=
ConfigurationManager.PasswordMinLetters=0
ConfigurationManager.PasswordMinDigits=0
ConfigurationManager.PasswordMinSecialChars=0
ConfigurationManager.PasswordSecialCharSet=!@#$%^&*()_+[]
```

The default is no length or complexity or expiration day limitation.

- Password length control equal to 0 means no restriction. Any other number, we will check for min and/or max length.
- Password expiration - if any number set then the password will expire after the specified days.
- Password complexity - forces to check letter, digits, and special characters. Default is no complexity check. If you set any number on any of these properties, CAMM checks complexity. For example, PasswordMinLetters=2 means you must have at least 2 letters in your password.

For special characters checking, there is a list of default special characters. This is configurable.

A.2.2.2 CAMM User Session Time Out

CAMM User Session can be set to time out after a specified time.

Example A–3 CAMM User Session Time Out

```
# ConfigurationManager.SessionTimeout's unit is minutes.
ConfigurationManager.SessionTimeout=10
```

The default is 10 minutes, which means that after 10 minutes of no activity on the applet the applet will close automatically and redirect the user back to the login screen.

A.2.3 Multi-Domain Monitoring Configuration

One can limit number of domains to be monitored by setting resource limit parameter: ConfigurationManager.ResourceLimit=2

Example A–4 Multi-Domain Monitoring Configuration

```
ConfigurationManager.ResourceLimit=2
```

The default is 2, this means that by default CAMM is set to monitor no more than two Application Server domains.

A.2.4 CAMM RMI Port Assignment

CAMM uses RMI ports for communication with the agents and collects incoming performance metrics from a particular RMI port. By default, the RMI port is set on the same machine that hosts CAMM. RMI.Registry.Host needs to be un-commented and have a value other than localhost if the host is multi-homed (such as, many network interfaces or has any ipv6 addresses) and you need to make sure that CAMM listens to the incoming traffic on the particular interface.

You may need to change RMI.Registry.Port value in case the default 51099 port number has been allocated to an other application. Also if CAMM is running in multi-instance mode, the port number will be different from instance to instance.

Example A–5 CAMM RMI Port Assignment

```
#RMI.Registry.Host = localhost
RMI.Registry.Port = 51099
RMI.Registry.UseExternal = false  [NK- what exactly this means?]
```

A.2.5 CAMM Aggregation and Data Life Time Configuration

CAMM has sophisticated multi-tiered logic for aggregation (or compression) of performance data. This helps to optimize performance of interaction with the internal data repository both when querying data for presentation or inserting new performance metrics.

Users who want to store longer term data should look for this section in Acsera.properties:

```
#########################
# Production setting
# NOTE: use Model.GlobalSamplingRateSecs to configure Metric.Grain.0
#########################
Metric.Grain.0 0s
Metric.TableInterval.0 = 4h
Metric.DataLife.0 = 8d

Metric.Grain.1 = 3m
Metric.TableInterval.1 =1d
Metric.DataLife.1 = 8d

#Metric.Grain.2 = 30m
#Metric.TableInterval.2 = 7d
#Metric.DataLife.2 = 420d
```

and uncomment the last 3 lines for the Metric.*.2 properties

A.2.6 Aggregating Incoming Metrics On the Fly

CAMM by default aggregates data coming from multiple cluster members by application thus minimizing rate of insertion in to the data repository. This greatly improves performance of CAMM in heavily loaded environment.

As a side effect of this approach though, the user is unable to see metrics from instrumentation (processes and portals) on per server level. If you need to enable this then set the JavaMIP.AggregateInserts to *false*.

A.2.7 Configuring List of Applications To Be Monitored or Excluded From Monitoring

To avoid overhead of unnecessary monitoring of certain applications, you can explicitly state which applications to monitor, or which applications to exclude from monitoring.

Users should append the name of their application to the property ComponentProvider.Application.Exclude.

Example A–6 Specifying Which Applications to Monitor

```
# Control which applications to analyze
#
ComponentProvider.Application.Include=
ComponentProvider.Application.Exclude=WLI System EJBs,WLI-AI
Design-time,B2BDefaultWebAppApplication,WLI
Worklist,JWSQueueTransport,Deployer,BEA_WLS_DBMS_ADK,Acsera
```

A.2.8 Firewall Mitigation (for Internal RMI Ports)

If there is a firewall between the CAMM Manager and the monitored application servers, ports need to be opened between them. In addition to the application server's JMX access ports, the following two properties in Acsera.properties indicate the ports used specifically by CAMM:

- RMI.Registry.Port (51099 by default)
- RMI.JavaProvider.ServerPort (55003 by default)

A.2.9 SLO Dampening

There are times when you deliberately want to cut down on the number of repeated notifications should SLO violation persist for a given period of time. To suppress notifications of the same violation in a short period of time, CAMM provides the SLO Dampening feature. Once enabled, should a SLO violation occur and be repeated several times in a short period, CAMM will not fire the SLO violation notification for the time period defined in SLO.RearmDelay. To disable this feature, set the value of this parameter to 0.

SLO.SuppressDelayedAsserts indicates that if the violation still persists upon time period expiry CAMM, should fire the SLO notification. By default it is *false*, for example, fire the notification.

Example A–7 SLO Dampening

```
# The following property is specified in units of
# minutes (m), hours (h) or days (d)
SLO.RearmDelay = 15m
SLO.SuppressDelayedAsserts = false
```

A.3 UrlMap.properties

The UrlMap.properties file should be created in the CAMM Manager's config directory and used to provide address mappings between load balancers and application servers. The format of this file is:

```
# Format:
#    $app_server_ip = $load_balancer_id
# E.g:
#    http\://localhost\:7001 = http\://localhost\:7005
#
# Note: ":" character need to be escaped with "\"
#
http\://192.168.128.53\:7002 = http\://192.168.3.187\:80
http\://192.168.128.53\:7003 = http\://192.168.3.187\:80
http\://192.168.128.54\:7005 = http\://192.168.128.54\:7011
http\://192.168.128.54\:7006 = http\://192.168.128.54\:7011
```