Oracle Identity Analytics
System Administrator's Guide

11*g* Release 1

ORACLE®

110504@25097

Contents

Preface

About This Guide

This guide describes how to configure and administer Oracle® Identity Analytics 11gR1 software at a systems level.

Who Should Read This Guide

The *Oracle Identity Analytics 11gR1 System Administrator's Guide* is written for system administrators, deployment engineers, and service providers who are responsible for administering the Oracle Identity Analytics 11gR1 software at a systems level.

- Deployment engineers who are responsible for integrating Oracle Identity Analytics 11gR1 with other IT systems should see the *Oracle Identity Analytics 11gR1 System Integrator's Guide*.

- Compliance officers and IT specialists who need to configure and maintain role management and compliance functionality should see the *Oracle Identity Analytics 11gR1 Business Administrator's Guide*.

- Business managers and other users in a supervisory role who need information about how to use the Oracle Identity Analytics 11gR1 software to grant employees and partners access to applications, check for access violations, and so on should see the *Oracle Identity Analytics 11gR1 User's Guide*.

1

Understanding and Configuring the System Log

The System Log

The Oracle® Identity Analytics 11gR1 System Log (`rbacx.log`) captures information that is useful for both troubleshooting and general monitoring purposes. It particular, the System Log captures information about exceptions that arise while running the application.

The system log is located in the `$RBACX_HOME/logs` folder.

Note — The `$RBACX_HOME` environment variable denotes the path to the directory in which the Oracle Identity Analytics 11gR1 software is installed.

Note – Oracle Identity Analytics writes messages to two other logs: the Audit Event Log and the Import-Export Log.

- The Audit Event Log records actions that affect users, such as Add, Modify, and Delete user actions, user password updates, and login and logout actions.
- The Import-Export Log records details about import jobs, including user imports, account imports, and glossary imports, and export jobs.

These logs are documented in the *Audit Event Log and Import-Export Log* chapter in the *Oracle Identity Analytics 11gR1 Business Administrator's Guide*.

Configuring the System Log

This document explains how to configure Oracle Identity Analytics logging.

Oracle Identity Analytics uses a Java® logging framework called log4j. Oracle Identity Analytics log files are listed in the `log4j.properties` file, which is located in the `$RBACX_HOME/WEB-INF` folder. The `log4j.properties` file is used to configure different logging levels within Oracle

Identity Analytics and also to enable and disable logging as needed. Each line in the `log4j.properties` file corresponds to a component in Oracle Identity Analytics, and each component can output messages to a log.

Logging Levels

The following table defines the four levels of logging that can be set within log4j for Oracle Identity Analytics components.

Level	Description
FATAL	This level will record events only if a severe error condition occurs. Use this setting if the minimum amount of logging is desired.
ERROR	This level will record events if there is an ERROR condition.
WARN	This is the default level. It records events if potentially harmful situations occur.
DEBUG	Use this setting if the maximum amount of logging is desired.

Understanding How the `log4j.properties` File Is Organized

The following table describes the more than two dozen sections that make up the Oracle Identity Analytics `log4j.properties` file. Set properties in a given section to control logging for individual components. Sections are listed in the table in the same order that they appear in the default Oracle Identity Analytics `log4j.properties` file. For reference, see the "Understanding How the `log4j.properties` File Is Organized" on page 8 file at the end of this chapter.

Section	Description
Console Appender	These properties control how Java console logging events are formatted. **Note** - These properties apply to the logging *format* only.
File Appender	These properties configure the `rbacx.log` file.
Tomcat logging	Configures Tomcat-related logging if Oracle Identity Analytics is deployed to a Tomcat application server. Logs are generated and stored in the `$RBACX_HOME/tomcat55/logs/` folder in a file named `rbacx.log`.
VAAU commons logging	Configures message logging having to do with component interactions within Oracle Identity Analytics.

Section	Description
Oracle Identity Analytics Core logging	Configures the logging of core events inside Oracle Identity Analytics. **Note - These properties should be changed when troubleshooting in Oracle Identity Analytics.**
Oracle Identity Analytics Security logging	Configures the logging of events related to security in Oracle Identity Analytics. This setting includes user creation and login/logout events in Oracle Identity Analytics.
Oracle Identity Analytics Scheduling logging	Configures scheduler component logging in Oracle Identity Analytics. Events recorded by the Quartz Job Scheduler component can be configured using this property.
Oracle Identity Analytics ETL	Configures the logging of events output by the ETL (Extract, Transform, and Load) process inside Oracle Identity Analytics.
Oracle Identity Analytics IAM logging	Configures the logging of events based on activity between Oracle Identity Analytics and an IAM (Identity Access Management) server. This includes any file-based imports that occur inside Oracle Identity Analytics.
Oracle Identity Analytics Reporting logging	Configures the logging of events related to the running of reports inside Oracle Identity Analytics.
Oracle Identity Analytics Audit logging	Configures the logging of auditing events inside Oracle Identity Analytics, such as login/logout events and changes made inside Oracle Identity Analytics.
Oracle Identity Analytics IDC logging	Configures the logging of events related to the Oracle Identity Analytics Identity Certification component.
System	Configures the logging of System component events in Oracle Identity Analytics.
Sandbox	Configures the logging of identity audit (IDA) inside Oracle Identity Analytics.
Workflow	Configures the logging of Oracle Identity Analytics workflow events.
SqlMap logging configuration	Configures the logging of events having to do with communications between the database and Oracle Identity Analytics.
Spring Framework	Configures the logging of events that have to do with the underlying Spring Framework. Oracle Identity Analytics is built using the Spring Framework. **Note - These properties should be changed when troubleshooting in Oracle Identity Analytics.**
JIAM log	Configures the logging of events based on activity between a JIAM connection, such as CA-Admin, and Oracle Identity Analytics.
Quartz scheduler	Configures the logging of events that have to do with the Quartz Job Scheduler that is used in Oracle Identity Analytics. Also see the "Oracle Identity Analytics Scheduling logging" property, above.
DWR	Configures the logging of events related to the web client (user interface) JSP® pages.

Section	Description
ehcache	Configures the logging of events related to the cache on the Oracle Identity Analytics web client (user interface). (Ehcache is a Java cache library that is used in Oracle Identity Analytics.)
CloverETL	Configures the logging of events related to the CloverETL framework. The CloverETL framework is used in support of the Oracle Identity Analytics ETL (extract, transform, load) functionality.
C3pO	Configures the logging of events related to the c3p0 library. Oracle Identity Analytics uses the c3pO library to support common database pooling.
JasperReports	Configures the logging of events related to the JasperReports library. Oracle Identity Analytics uses the JasperReports library to create and output reports.

`log4j.properties` File

This section provides a sample `log4j.properties` file.

```
log4j.rootLogger=INFO, file
# Console Appender
log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.layout=org.apache.log4j.PatternLayout
log4j.appender.console.layout.ConversionPattern=%d{ABSOLUTE} %-5p [%c{1}] %m%n


# File Appender
log4j.appender.file=org.apache.log4j.DailyRollingFileAppender
log4j.appender.file.file=logs/rbacx.log
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %-5p [%c{1}] %m%n
log4j.appender.file.ImmediateFlush=true
log4j.appender.file.DatePattern='.'yyyy-MM-dd


# Tomcat logging
log4j.logger.org.apache.catalina=WARN

# DON'T EDIT FOLLOWING
log4j.logger.com.vaau.commons.springframework.context.ContextLifecycleListener=INFO

#VAAU commons logging
log4j.logger.com.vaau.commons=WARN

#RBACx Core logging
log4j.logger.com.vaau.rbacx= WARN
log4j.logger.com.vaau.rbacx.core= WARN
log4j.logger.com.vaau.rbacx.service= WARN
log4j.logger.com.vaau.rbacx.manager= WARN

# RBACx Security logging
log4j.logger.com.vaau.rbacx.security=WARN
```

```
#RBACx Scheduling logging
log4j.logger.com.vaau.rbacx.scheduling=DEBUG

# RBACx ETL
log4j.logger.com.vaau.rbacx.etl.manager=WARN

#RBACx IAM logging
log4j.logger.com.vaau.rbacx.iam= DEBUG

#RBACx Reporting logging
log4j.logger.com.vaau.rbacx.reporting=WARN

#RBACx Audit logging
log4j.logger.com.vaau.rbacx.audit=WARN

# RBACx IDC logging
log4j.logger.com.vaau.rbacx.idc=WARN

# SYSTEM
log4j.logger.com.vaau.rbacx.system=DEBUG

# Sandbox
log4j.logger.com.vaau.rbacx.sandbox.ida=WARN
log4j.logger.com.vaau.rbacx.sandbox.rme=WARN

# Workflow
log4j.logger.com.vaau.rbacx.workflow=WARN
log4j.logger.com.opensymphony.workflow.AbstractWorkflow=ERROR

# SqlMap logging configuration. Change WARN to DEBUG if want to see all sql statements
log4j.logger.com.ibatis=WARN
log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=WARN
log4j.logger.com.ibatis.common.jdbc.ScriptRunner=WARN
log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate=WARN
log4j.logger.org.springframework.jdbc.datasource.DataSourceTransactionManager=WARN
log4j.logger.java.sql.Connection=WARN
log4j.logger.java.sql.Statement=WARN
log4j.logger.java.sql.PreparedStatement=WARN

#Spring Framework
log4j.logger.org.springframework=WARN
log4j.logger.org.springframework.rules.values=WARN
log4j.logger.org.springframework.context.support=WARN
log4j.logger.org.springframework.transaction=WARN
log4j.logger.org.springframework.aop.interceptor=WARN
log4j.logger.org.springframework.security=WARN
log4j.logger.org.springframework.security.event.authentication.LoggerListener=FATAL

# For Trace Logging change them TRACE
log4j.logger.org.springframework.aop.interceptor.PerformanceMonitorInterceptor=WARN
log4j.logger.org.springframework.aop.interceptor.CustomizableTraceInterceptor=WARN

##JIAM log
log4j.category.com.ca=WARN
#log4j.category.com.ca.commons.jndi=DEBUG

#Quartz scheduler
log4j.logger.org.quartz=WARN
```

```
#DWR
log4j.logger.uk.ltd.getahead.dwr=FATAL
log4j.logger.org.directwebremoting=FATAL

#ehcache
log4j.logger.net.sf.ehcache=ERROR

#CloverETL
log4j.logger.org.jetel=ERROR

#C3p0
log4j.logger.com.mchange=ERROR

# JasperReports
log4j.logger.net.sf.jasperreports=ERROR
log4j.logger.com.vaau.rbacx.search=WARN
log4j.logger.com.vaau.commons.search=WARN
```

2

Using System Logs

This chapter provides guidance about messages that should be monitored in order to maintain system performance. Examples show the type of function, the message severity, the module name, and the log message.

Tomcat Logging

Tomcat logging captures messages related to the Tomcat application server.

All Tomcat-related errors are shown in the following manner in the log file. Monitor anything that contains this string: [org.springframework.web.context.ContextLoader].

15:11:56,500 ERROR [org.springframework*] **

15:11:56,500 FATAL [org.springframework*] **

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following sub-sections.

Context Initialization

The following error is a context initialization error that can occur when Oracle Identity Analytics starts. The error indicates that there is a context initialization failure, and the log message indicates which file caused the error.

In the following sample error message, the job.xml file under WEB-INF has caused the error.

```
Severity: ERROR

Module name: ContextLoader

Log message:

15:11:56,500 ERROR [org.springframework.web.context.ContextLoader] Context
initialization failed

org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'usersImportTrigger' defined in ServletContext resource [/WEB-INF/jobs.xml]:
Error setting property values; nested exception is
org.springframework.beans.PropertyAccessExceptionsException:
PropertyAccessExceptionsException (1 errors); nested propertyAccessExceptions are:
[org.springframework.beans.MethodInvocationException: Property 'cronExpression' threw
exception; nested exception is java.text.ParseException: '?' can only be specfied for
Day-of-Month or Day-of-Week.]

PropertyAccessExceptionsException (1 errors)
```

In the following example, the log message shows that the scheduling-context.xml file under
WEB-INF has caused the error. The log message also shows the line in the file that caused the
error.

```
Severity: ERROR

Module name: ContextLoader

Log message:

15:22:03,109 ERROR [org.springframework.web.context.ContextLoader]
Context initialization failed

org.springframework.beans.factory.BeanDefinitionStoreException: Line 137 in XML
document from ServletContext resource [/WEB-INF/scheduling-context.xml] is invalid;
nested exception is org.xml.sax.SAXParseException: The string "--" is not permitted
within comments.
```

VAAU Commons Logging

VAAU Commons logging captures messages having to do with component interactions within
Oracle Identity Analytics.

VAAU Commons errors are shown as follows in the log file. For monitoring purposes, monitor
anything that contains [com.vaau.commons].

```
15:11:56,500 ERROR [com.vaau.commons*] **

15:11:56,500 FATAL [com.vaau.commons*] **
```

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following sub sections.

Context Initialization

The following example contains information about Oracle Identity Analytics. The log message shows the version of Oracle Identity Analytics that is running and the application status.

```
Severity: INFO

Module name: ContextLifecycleListener

Log message:

10:30:19,859 INFO [com.vaau.commons.springframework.context.ContextLifecycleListener]
Oracle Identity Analytics (build: 5.1.0.20080903_406_3061) Started
```

Oracle Identity Analytics Core Logging

Oracle Identity Analytics Core logging logs messages having to do with core Oracle Identity Analytics events.

This section provides sample Oracle Identity Analytics core logging messages. For monitoring purposes, monitor anything that contains [com.vaau.rbacx].

15:11:56,500 ERROR [com.vaau.rbacx *] **

15:11:56,500 FATAL [com.vaau.rbacx *] **

15:11:56,500 ERROR [com.vaau.rbacx.service *] **

15:11:56,500 FATAL [com.vaau.rbacx.service *] **

15:11:56,500 ERROR [com.vaau.rbacx.core *] **

15:11:56,500 FATAL [com.vaau.rbacx.core *] **

15:11:56,500 ERROR [com.vaau.rbacx.manager *] **

15:11:56,500 FATAL [com.vaau.rbacx.manager *] **

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following subsections.

Sequence Update

The following message shows Oracle Identity Analytics back-end activities. The log messages show that Oracle Identity Analytics is updating the sequence table in the database.

```
Severity: DEBUG

Module name: SequenceGeneratorServiceImpl

Log message:

13:22:35,203 DEBUG [com.vaau.rbacx.service.impl.SequenceGeneratorServiceImpl]
Getting MemorySequence for sequence name NamespaceKey

13:22:35,203 DEBUG [com.vaau.rbacx.service.impl.SequenceGeneratorServiceImpl]
Creating new MemorySequence for sequence name NamespaceKey

13:22:35,203 DEBUG [com.vaau.rbacx.dao.ibatis.SqlMapSequenceDao]
Getting next count for sequenceName=NamespaceKey, increment=10

13:22:35,218 DEBUG [com.vaau.rbacx.dao.ibatis.SqlMapSequenceDao]
Returning next count for sequenceName=NamespaceKey, count=1010

13:22:35,234 DEBUG [com.vaau.rbacx.service.impl.SequenceGeneratorServiceImpl]
Returning count for sequence name Name
```

Oracle Identity Analytics Security Logging

Oracle Identity Analytics Security logging logs events related to security, including user creation events, and login and logout events.

Oracle Identity Analytics security logging errors are logged as shown here. For monitoring purposes, monitor anything that contains [com.vaau.rbacx.security].

```
15:11:56,500 ERROR [com.vaau.rbacx.security *] **
```

```
15:11:56,500 FATAL [com.vaau.rbacx.security *] **
```

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the subsections below.

Login Error

The following message shows a Oracle Identity Analytics security warning. The warning indicates that the user's login or password is incorrect.

```
Severity: WARN

Module name: UserManagerImpl

Log message:

14:14:45,359 WARN [com.vaau.rbacx.security.manager.impl.UserManagerImpl]
RbacxUser with username: 'testuser' not found
```

User Creation

The following message shows that the user testuser has been created.

```
Severity: DEBUG

Module name: RbacxSecurityServiceImpl

Log message:

15:35:00,750 DEBUG
[com.vaau.rbacx.security.service.impl.RbacxSecurityServiceImpl]
adding user testuser

15:35:00,750 DEBUG
[com.vaau.rbacx.security.manager.impl.UserManagerImpl]
creating user: Last name: User; First name: Test; Email: testuser@oracle.com

15:35:00,765 DEBUG
[com.vaau.rbacx.security.manager.impl.UserManagerImpl]
setting credentials for testuser: 3dbb4a67672880904958500b68d4ab481116a1b9
```

User Deletion

The following user deletion message shows that the user testuser has been deleted.

```
Severity: DEBUG

Module name: RbacxSecurityServiceImpl

Log message:

15:34:23,359 DEBUG [com.vaau.rbacx.security.service.impl.RbacxSecurityServiceImpl]
deleting user testuser

15:34:23,375 DEBUG [com.vaau.rbacx.security.manager.impl.UserManagerImpl]
deleting user: Last name: user; First name: test; Email: testuser@oracle.com
```

Oracle Identity Analytics Scheduling Logging

Oracle Identity Analytics Scheduling logging logs messages related to the scheduler component.

This section shows example Oracle Identity Analytics scheduling error messages. For monitoring purposes, monitor anything that contains [`com.vaau.rbacx.scheduling`].

```
15:11:56,500 ERROR [com.vaau.rbacx.scheduling *] **
```

```
15:11:56,500 FATAL [com.vaau.rbacx.scheduling *] **
```

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following subsections.

Scheduled Certification Reminder

The following message shows the information for a scheduled execution. The message shows that a certification reminder has executed as scheduled.

```
Severity: INFO

Module name: VaauSchedulerEventListenerImpl

Warn message:

14:19:00,187 INFO
[com.vaau.rbacx.scheduling.impl.VaauSchedulerEventListenerImpl]
Job executed: Certification Reminder, SYSTEM
```

Scheduled Account Import Job

The following message shows the information for a scheduled execution. The message shows that an account import job has executed as scheduled.

```
Severity: DEBUG

Module name: IAMJob

Warn message:

14:21:00,062 DEBUG
[com.vaau.rbacx.scheduling.manager.providers.quartz.jobs.IAMJob]
Accounts Import job executed successfully = true
```

Oracle Identity Analytics Identity Access Management (IAM) Logging

Oracle Identity Analytics Identity Access Management (IAM) logging logs events based on activity between Oracle Identity Analytics and an Identity Access Management (IAM) server. This includes any file-based imports that occur inside Oracle Identity Analytics.

This section shows example Oracle Identity Analytics IAM (Identity Access Management) errors. For monitoring purposes, monitor anything that contains [com.vaau.rbacx.iam].

15:11:56,500 ERROR [com.vaau.rbacx.iam *] **

15:11:56,500 FATAL [com.vaau.rbacx.iam *] **

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following subsections.

User Import

The following message shows an error from a user file import. The error indicates that there is a data type violation. The first column in the schema file is specified by statusKey, and in the user file, it has Active corresponding to the statusKey. The statusKey, however, is defined to be either 1 as active, or 2 as inactive. The correct data type corresponding to the statusKey should be either 1 or 2.

```
Severity: ERROR

Module name: UserFileReader

Log message:

11:29:00,593 ERROR [UserFileReader] PropertyAccessExceptionsException (1 errors)

Schema file:

statusKey,manager,primaryEmail,firstName,middleName,username

User file imported with error:

Active,testmanager,testuser@email.com,Test,User,testuser

Correct user file:

1,testmanager,testuser@email.com,Test,User,testuser
```

Account Import

The following message shows an account file import error. The error indicates that the account file does not match the schema file. The schema file shows that one role will be imported to the account. However, the account file has two roles. Instead of importing two roles as two attributes, import the two roles into the role attribute as a single attribute.

```
Severity: ERROR

Module name: CSVAccountFileReader

Log message:

11:53:02,625 ERROR [CSVAccountFileReader] BAD RECORD FORMAT:
File: UNX_01_accounts, line no. 1, doesn't match schema,
found [testuser,JOB_1,JOB_2,UNX]

Schema file:

name<CorrelationKey>,role,endpoint

Account file imported with error:

testuser,JOB_1,JOB_2,UNX

Correct user file:

testuser,"JOB_1,JOB_2",UNX
```

Unknown User

In this example of an account file import error, the error indicates that the user of the account file does not match any users in the database. In this case, configure the system to either drop the account or to correlate to a default global user.

```
Severity: ERROR

Module name: CSVAccountFileReader

Log message:

12:12:01,015 ERROR
[CSVAccountFileReader] CORRELATION ERROR: Unknown global user 'test-user' for
account 'test-user'

Schema file:

name<CorrelationKey>,role,endpoint

Account file imported with error:

test-user, "JOB_1,JOB_2",UNX
```

```
Correct user file:

testuser,"JOB_1,JOB_2",UNX
```

Oracle Identity Analytics Reports Logging

Oracle Identity Analytics Reports logging logs events related to the running of reports inside Oracle Identity Analytics.

All Oracle Identity Analytics reporting-related errors will be shown in the following manner. For monitoring purposes, monitor anything that contains [com.vaau.rbacx.reporting].

```
15:11:56,500 ERROR [com.vaau.rbacx.reporting *] **
```

```
15:11:56,500 FATAL [com.vaau.rbacx.reporting *] **
```

Note —

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following subsections.

Generating Reports

The following example shows the information that is logged when reports are generated. The log messages show the type of report that is generated and the location of the file.

```
Severity: DEBUG

Module name: JasperReportsUtils

Log message:

10:33:09,390 DEBUG
[com.vaau.rbacx.reporting.renderer.jasperreports.JasperReportsUtils] --->
checking if 'C:\Vaau\rbacx-4.0\reports\BusinessUnitUsers.jrxml' has
any sub-reports that need compiling

10:33:09,406 DEBUG
[com.vaau.rbacx.reporting.renderer.jasperreports.JasperReportsUtils] --->
C:\Vaau\rbacx-4.0\reports\BusinessUnitUsers.jrxml has following
sub-reports --> []

10:33:09,406 DEBUG
[com.vaau.rbacx.reporting.renderer.jasperreports.JasperReportsUtils] --->
following sub-reports require compilation []
```

```
10:33:09,406 DEBUG
[com.vaau.rbacx.reporting.renderer.jasperreports.JasperReportsUtils]
checking if report file --> 'C:\Vaau\rbacx-4.0\reports\BusinessUnitUsers.jrxml'
requires compilation

10:33:09,406 DEBUG
[com.vaau.rbacx.reporting.renderer.jasperreports.JasperReportsUtils] -->
compiling reports []
```

Oracle Identity Analytics Audit Logging

Oracle Identity Analytics Audit logging logs auditing events inside Oracle Identity Analytics, such as login and logout events and changes made inside Oracle Identity Analytics.

The following example messages show audit-related errors. For monitoring purposes, monitor anything that contains [com.vaau.rbacx.rbacxaudit].

```
15:11:56,500 ERROR [com.vaau.rbacx.rbacxaudit *] **
```

```
15:11:56,500 FATAL [com.vaau.rbacx.rbacxaudit *] **
```

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following subsections.

Audit Logging

These example messages show the information that is logged when users log in to or log out of Oracle Identity Analytics. The log also captures changes made in Oracle Identity Analytics.

```
Severity: DEBUG
```

```
Module name: RbacxAuditUtil
```

```
Log message:
```

```
10:33:09,390 DEBUG [com.vaau.rbacx.rbacxaudit.util.RbacxAuditUtil] ---> User
"testuser" logged in
```

Oracle Identity Analytics Identity Certification (IDC) Logging

Oracle Identity Analytics Identity Certification (IDC) logging logs events related to the Oracle Identity Analytics Identity Certification component.

Identity certification (IDC) errors are shown as follows in the log file. For monitoring purposes, monitor anything that contains [com.vaau.rbacx.idc].

```
15:11:56,500 ERROR [com.vaau.rbacx.idc *] **
```

```
15:11:56,500 FATAL [com.vaau.rbacx.idc *] **
```

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the sections below.

Certification Reminder

This message shows the number of certification reminders that have fired.

```
Severity: DEBUG

Module name: ReminderManagerImpl

Log message:

12:43:00,171 DEBUG [ReminderManagerImpl] No. of first certification
reminders = 19

12:43:00,171 DEBUG [ReminderManagerImpl] No. of second certification
reminders = 0

12:43:00,171 DEBUG [ReminderManagerImpl] No. of third certification
reminders = 0

12:43:00,171 DEBUG [ReminderManagerImpl] No. of fourth certification
reminders = 0

12:43:00,171 DEBUG [ReminderManagerImpl] No. of fifth certification
reminders = 0

12:43:00,171 DEBUG [ReminderManagerImpl] [finished] firing certification
reminders, took 31ms
```

Certification Creation, Example 1

The following message shows certification information. The messages show that the users and the accounts involved in the certification are not being created because the users or the accounts are not active.

```
Severity: DEBUG

Module name: RbacxIDCServiceImpl

Log message:

16:20:59,375 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
creating full certification 'T2'

16:20:59,390 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
loading global users from business units --> [3066]

16:20:59,453 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
loaded 2 GlobalUsers, took 63ms

16:20:59,453 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
creating IDC Users...

16:20:59,500 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
loading Accounts, Roles and Policies for 2 users

16:20:59,515 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
loaded Accounts, Roles and Policies for 2 users, took 15ms

16:20:59,515 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
--> user: ZRC0217

16:20:59,546 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
----> user 'ZRC0217' has no accounts to certify

16:20:59,546 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
---> user Craig, Ryan has not accounts, and will not inlcluded

16:20:59,546 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
--> user: ZTJ0071

16:20:59,546 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
----> user 'ZTJ0071' has no accounts to certify

16:20:59,546 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
---> user Jorgensen, Thomas has not accounts, and will not inlcluded

16:20:59,546 DEBUG [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
created 0 IDC Users , took 93ms
```

Certification Creation, Example 2

In this example, the message shows that the business unit involved in the certification is not being created because either the users or the accounts are not active.

```
Severity: ERROR

Module name: RbacxIDCServiceImpl

Log message:

16:40:01,203 ERROR [com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl]
ERROR: unable to create certification: T3_Aaron Hackett - ZAH0140

com.vaau.rbacx.idc.IDCInvalidArgumentException: BusinessUnit 'Aaron Hackett -
ZAH0140' has no acitve users to certify

at com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl.
createCertification(RbacxIDCServiceImpl.java:511)

at com.vaau.rbacx.idc.service.impl.RbacxIDCServiceImpl.
createCertifications(RbacxIDCServiceImpl.java:256)

at com.vaau.rbacx.scheduling.executor.certification.
CertificationJobExecutor.execute(CertificationJobExecutor.java:22)

at com.vaau.rbacx.scheduling.manager.providers.quartz.
jobs.AbstractQuartzJob.execute(AbstractQuartzJob.java:58)

at org.quartz.core.JobRunShell.run(JobRunShell.java:191)

at org.quartz.simpl.SimpleThreadPool$WorkerThread.
run(SimpleThreadPool.java:516)
```

JIAM Logging

JIAM logging logs events based on activity between a JIAM (Java Identity Access Management) connection, such as CA-Admin, and Oracle Identity Analytics.

JIAM errors are shown as follows in the log file. For monitoring purposes, monitor anything that contains [com.vaau.rbacx.idc].

15:11:56,500 ERROR [com.vaau.rbacx.idc *] **

15:11:56,500 FATAL [com.vaau.rbacx.idc *] **

Note -

* Refers to the specific module.

** Refers to the actual error message.

Examples are shown in the following subsections.

IDM Connection

The following message shows an error establishing a connection to an IDM environment. The log messages show that there is a failure establishing the connection to CA Etrust IAM Solution.

Severity: DEBUG

Module name: operations

Log message:

```
16:27:43,765 DEBUG [com.ca.commons.jndi.beans.operations] Search[0] Failed:
javax.naming.CommunicationException: localhost:20390 [Root exception is
java.net.ConnectException: Connection refused: connect]
```

```
16:27:43,796 ERROR [com.vaau.rbacx.iam.ca.ETrustIAMSolution] Error getting
ETrust Iam Server
```

```
com.ca.iam.model.IAMCommunicationException: javax.naming.CommunicationException:
localhost:20390 [Root exception is java.net.ConnectException:
Connection refused: connect]
```

```
at com.vaau.rbacx.iam.ca.ETrustIAMSolution$ETrustIamServerPoolFactory.
makeObject(ETrustIAMSolution.java:1725)
```

3

Oracle Identity Analytics Troubleshooting

To Verify That Oracle Identity Analytics Started Properly

To verify that Oracle Identity Analytics started properly, do one of the following:

- Check the log file.

  A successful launch of Oracle Identity Analytics is recorded in the application log file, which is named rbacx.log. The entry listed in the log file includes the Oracle Identity Analytics version number and a time stamp that records when the application started.

  An example log entry is listed here:

  ```
  "17:27:11,218 INFO [ContextLifecycleListener] Oracle Identity Analytics
  (build: 4.1.0.20080903_406_3061) Started"
  ```

- Open the application in a browser.

  You can verify that Oracle Identity Analytics started correctly by typing http://localhost:9080/rbacx in a browser on the local application server machine.

  The localhost portion of the URL assumes that you are checking the application from the local application server (that is, you are not checking from a remote machine). The URL also assumes that Oracle Identity Analytics is deployed on TCP port 9080.

  If you are opening the URL from a remote machine, format the URL as follows: http://*application server hostname*:*port*/rbacx.

To Start / Stop Oracle Identity Analytics

Start and stop Oracle Identity Analytics using the Application Control Panel of your application server.

For example, if you are using the Glassfish® application server, use the Administrative Console to start and stop Oracle Identity Analytics.

Troubleshooting Common Errors

This section describes solutions to errors that are commonly encountered when working with Oracle Identity Analytics.

JDBC Connection Error

This error occurs when Oracle Identity Analytics is unable to connect to the database. The error is logged in the rbacx.log file.

```
Failed to obtain DB connection from data source
'springNonTxDataSource.QuartzScheduler': java.sql.SQLException:
Connections could not be acquired from the underlying database!
[ See nested exception: java.sql.SQLException: Connections
could not be acquired from the underlying database!  ]
```

To Resolve the Error

- Check the jdbc.properties configuration file in the $RBACX_HOME/conf folder.
- Check the conf-context.xml file in the /WEB-INF/ folder.
- Ensure that JDBC drivers corresponding to the database type are present in /WEB-INF/lib.
- Verify that database server connectivity can be established from the application server.

Error Loading Workflow

The following error occurs when the workflows.xml file is not configured properly in the /WEB-INF/classes folder.

```
Error loading workflow Role Membership Workflow
com.opensymphony.workflow.FactoryException:
Error in workflow descriptor: [file:/]
<WORKFLOWS_FILE_PATH>role-user-membership-workflow.xml:
root cause: <$RBACX_HOME>\conf\workflows\role-user-membership-workflow.xml
(The device is not ready)
```

To Resolve the Error

Verify that the $RBACX_HOME variable in workflows.xml in /WEB-INF/classes is set correctly.

Error Rendering Report

The following error is generated when Oracle Identity Analytics reports cannot be rendered by the system:

```
20:44:43,498 ERROR [JasperPrintRenderer] Error rendering report:
java.io.FileNotFoundException:<$FILE_PATH>\<$FILE_NAME>.jasper
```

To Resolve the Error

- Validate the file path listed in reporting-context.xml.

- Verify that the report being rendered (<$FILE_NAME>.jasper) exists in the reports folder.

Oracle Identity Analytics Configuration Error

Any inaccuracies in the Oracle Identity Analytics configuration would generate errors and cause Oracle Identity Analytics not to launch. conf-context.xml and reporting-context.xml are two common files where configuration errors can cause failure. Some of the common errors are listed here:

```
java.io.FileNotFoundException: C:\Vaau\rbacx-4.1\conf\jdbc.properties
(The system cannot find the path specified)
java.io.FileNotFoundException: C:\Vaau\rbacx-4.1\conf\mail.properties
(The system cannot find the path specified)
java.io.FileNotFoundException: C:\Vaau\rbacx-4.1\conf\ldap.properties
(The system cannot find the path specified)
java.io.FileNotFoundException: C:\Vaau\rbacx-4.1\conf\iam.properties
(The system cannot find the path specified)
```

To Resolve the Error

Verify that the $RBACX_HOME path outlined in conf-context.xml and reporting-context.xml is accurate.

Java Heap Out of Memory Error

The java.lang.OutOfMemoryError exception in the log is caused by Java heap fragmentation. This fragmentation occurs when no contiguous chunk of free Java heap space is available from which to allocate Java objects.

To Resolve the Error

Increasing the size of the JVM® memory pool and clearing out the Java cache solves the exception. The recommended setting for min. / max. value is 512 MB / 1024 MB, respectively.

4

Tuning GlassFish To Run Oracle Identity Analytics

This chapter describes how to configure the GlassFish® application server to best run Oracle Identity Analytics.

Overview

GlassFish is the leading open-source and open-community application server. You can significantly improve the performance of Oracle Identity Analytics running on GlassFish by adjusting application server parameters and operating system configuration. The GlassFish Admin Console has numerous features that simplify and automate performance-related administrative tasks. Follow the recommendations in this chapter to configure and set up the application server.

Understanding Operation Requirements

Before deploying the application and tuning the application server, it is important to understand and clearly define the operational environment and performance goals. An optimal configuration for a production environment might not be optimal for a development environment.

Application Architecture

The J2EE® application model is very flexible, allowing the application architect to split application logic functionality into many tiers. The presentation layer is typically implemented using servlets and JavaServer Pages® (JSP®) technology, and executes in the Web container. Based on the application architecture, the server administrator can employ the application server in multiple tiers, or simply host both the presentation and business logic on a single tier.

Hardware Resource

The type and quantity of hardware resources available greatly influence performance tuning and site planning. The Oracle Identity Analytics solution provides excellent vertical scalability. It can scale to efficiently utilize multiple high-performance CPUs, using just one application server process. Such servers must also contain large amounts of memory, disk space, and network capacity to cope with increased load.

The Oracle Identity Analytics application can also be deployed on large "farms" of relatively modest hardware units. Using one or more external load balancers can efficiently spread user access across all the application server instances. A horizontal scaling approach can improve availability and lower hardware costs.

Process Overview

Tuning GlassFish to run Oracle Identity Analytics is a three-step process:

1. Tune the application server
2. Tune the Java runtime settings
3. Tune the operating system and platform

Each step is discussed separately in the following sections.

Tuning the Application Server

This section describes how to tune the GlassFish application server for optimum performance.

Deployment Settings

Deployment settings can have a significant impact on performance. The recommendations that follow ensure optimum performance when deploying Oracle Identity Analytics on the application server.

Disable "Auto-Deployment"

Enabling auto-deployment can adversely affect deployment because the Reload Poll Interval setting can impact performance significantly. For a production system, disable auto-deploy to optimize performance.

Disable auto-deployment through the Admin Console by choosing Stand-Alone Instances > Server (Admin Server) on the Advanced/Applications Configuration tab.

Disable "Dynamic Application Reloading"

When dynamic reloading is enabled the application server periodically checks for changes in deployed applications and automatically reloads them if changes are found. Dynamic reloading is intended for development environments and is incompatible with session persistence. To improve performance, disable dynamic class reloading.

Disable dynamic class reloading for an application that is already deployed by opening the Admin Console and choosing Stand-Alone Instances > Server (Admin Server) on the Advanced/Applications Configuration tab.

Logger Settings

The application server writes messages and exception stack trace output to the log file in the logs directory of the instance, located here: *appserver-rootdomain-name* /logs. A high volume of log activity can impact server performance, particularly during benchmarking situations.

General Settings

In general, writing to the system log slows down performance slightly, as does increased disk access. Increased disk access results when log levels are increased and file rotation limits or time limits are decreased.

Also, make sure that any custom log handler does not log to a slow device like a network file system since this can adversely affect performance.

Log Level Settings

Set the log level for the application server and its subsystems by opening the Admin Console, selecting the Logger Settings page, and clicking the Log Levels tab. The page enables you to specify the default log level for the server (labeled Root), the default log level for `javax.enterprise.system` subsystems (labeled Server) such as the EJB® container, MDB container, Web container, Classloader, JNDI naming system, and Security, and for each individual subsystem.

Log levels vary from FINEST, which provides maximum log information, through SEVERE, which logs only events that interfere with normal program execution. The default log level is INFO. The individual subsystem log level overrides the Server setting, which in turn overrides the Root setting.

For example, the MDB container can produce log messages at a different level than the application server default. To get more debug messages, set the log level to FINE, FINER, or FINEST. For best performance under normal conditions, set the log level to WARNING. Under benchmarking conditions, it is often appropriate to set the log level to SEVERE.

It is also best to set all Oracle Identity Analytics logging levels in `log4j.properties` to WARN.

Web Container Settings

Set Web container properties in the Admin Console by choosing Configurations > *Config-name* > Web Container.

Session Properties: Session Timeout

Session timeout determines how long the application server maintains a session if a user does not explicitly invalidate the session. The default value is 30 minutes. Setting a very large value for session timeout can degrade performance by causing the server to maintain too many sessions in the session store. However, setting a very small value can cause the server to reclaim sessions too soon.

Disable Dynamic JSP Reloading

On a production system, improve Web container performance by disabling dynamic JSP reloading. To do so, edit the default-web.xml file in the config directory for each instance.

Change the servlet definition for a JSP file to look like this:

```
<servlet>
<servlet-name>jsp</servlet-name>
<servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
<init-param>
<param-name>development</param-name>
<param-value>false</param-value>
</init-param>
<init-param>
<param-name>xpoweredBy</param-name>
<param-value>true</param-value>
</init-param>
<init-param>
<param-name>genStrAsCharArray</param-name>
<param-value>true</param-value>
</init-param>
 <load-on-startup>3</load-on-startup>
</servlet>
```

Thread Pool Settings

You can both monitor and tune thread pool settings through the Admin Console. To configure monitoring with the Admin Console, choose Configurations > *Config-name* > Monitoring. To view monitoring information with the Admin Console, choose Stand-Alone Instances > *Instance-name* (Monitor).

Tuning Thread Pool Settings (UNIX and Linux Only)

Configure thread pool settings through the Admin Console by choosing Configurations > *Config-name* > Thread Pools.

Because UNIX and Linux threads are always operating system (OS)-scheduled, as opposed to user-scheduled, UNIX and Linux users do not need to use native thread pools. Therefore, this option is not offered in a UNIX or Linux user interface. It is possible, however, to edit the OS-scheduled thread pools and add new thread pools, if needed, using the Admin Console.

Tuning the Java Runtime Environment

This section discusses tuning the Java Runtime Environment for optimum performance.

Java Virtual Memory Settings

Version 5.0 of Java 2 Platform, Standard Edition (J2SE® platform) provides two implementations of the Java HotSpot® virtual machine:

- The client VM is tuned for reducing startup time and memory footprint. Invoke it by using the `-client` JVM® command-line option.

- The server VM is designed for maximum program execution speed. Invoke it by using the `-server` JVM command-line option.

By default, the application server uses the Virtual Machine for the Java Platform (JVM machine) setting appropriate to the purpose:

- The Developer Profile, targeted at application developers, uses the `-client` JVM flag to optimize startup performance and conserve memory resources.

- The Enterprise Profile, targeted at production deployments, uses the default JVM startup mode. By default, the application server uses the Java HotSpot client virtual machine. However, if an application server VM is needed, it can be specified by creating a `<jvm-option>` named `-server`.

You can override the default by changing the JVM settings in the Admin Console under Configurations > *Config-name* > JVM Settings (JVM Options).

Managing Memory and Garbage Collections

The efficiency of any application depends on how well memory and garbage collection are managed. The following sections provide information on optimizing memory and allocation functions:

- "Tuning the Garbage Collector" on page 36
- "Tracing Garbage Collection" on page 36
- "Other Garbage Collection Settings" on page 37
- "Tuning the Java Heap Size" on page 37

Tuning the Garbage Collector

Garbage collection (GC) reclaims the heap space previously allocated to objects no longer needed. The process of locating and removing the dead objects can stall any application and consume as much as 25 percent throughput.

Almost all Java runtime environments come with a generational object memory system and sophisticated GC algorithms. A generational memory system divides the heap into a few carefully sized partitions called generations. The efficiency of a generational memory system is based on the observation that most of the objects are short-lived. As these objects accumulate, a low memory condition occurs forcing GC to take place.

The heap space is divided into the old generation and the new generation. The new generation includes the new object space (eden) and two survivor spaces. The JVM allocates new objects in the eden space, and moves longer-lived objects from the new generation to the old generation.

The young generation uses a fast copying garbage collector that employs two semi-spaces (survivor spaces) in the eden, copying surviving objects from one survivor space to the second. Objects that survive multiple young space collections are tenured, meaning they are copied to the tenured generation. The tenured generation is larger and fills up less quickly. So, it is garbage collected less frequently, and each collection takes longer than a young-space-only collection. Collecting the tenured space is also referred to as doing a full generation collection.

The frequent young space collections are quick (a few milliseconds), while the full generation collection takes longer (tens of milliseconds to a few seconds, depending upon the heap size).

Other GC algorithms, such as the Concurrent Mark Sweep (CMS) algorithm, are incremental. They divide the full GC into several incremental pieces. This provides a high probability of small pauses. This process comes with an overhead and is not required for enterprise web applications.

Tracing Garbage Collection

The two primary measures of garbage collection performance are throughput and pauses. Throughput is the percentage of the total time spent on other activities apart from GC. Pauses are times when an application appears unresponsive due to GC.

Two other considerations are footprint and promptness. Footprint is the working size of the JVM process, measured in pages and cache lines. Promptness is the time between when an object becomes dead and when the memory becomes available. This is an important consideration for distributed systems.

A particular generation size makes a trade-off between these four metrics. For example, a large young generation likely maximizes throughput, but at the cost of footprint and promptness. Conversely, using a small young generation and incremental GC will minimize pauses, and thus increase promptness, but decrease throughput.

JVM diagnostic output will display information on pauses due to GC. If you start the application server in verbose mode using the command asadmin start-domain -verbose domain, then the command-line argument -verbose:gc prints information for every collection.

Here is an example of output of the information generated with this JVM flag:

```
[GC 50650K->21808K(76868K), 0.0478645 secs]
[GC 51197K->22305K(76868K), 0.0478645 secs]
[GC 52293K->23867K(76868K), 0.0478645 secs]
[Full GC 52970K->1690K(76868K), 0.54789968 secs]
```

On each line, the first number is the combined size of live objects before GC, the second number is the size of live objects after GC, and the number in parentheses is the total available space, which is the total heap size minus one of the survivor spaces. The final figure is the amount of time that the GC took. This example shows three minor collections and one major (full) collection. In the first GC, 50650 KB of objects existed before collection and 21808 KB of objects existed after collection. This means that 28842 KB of objects were dead and collected. The total heap size is 76868 KB. The collection process required 0.0478645 seconds.

Other useful monitoring options include:

- -XX:+PrintGCDetails for more detailed logging information
- -Xloggc:file to save the information to a log file

Other Garbage Collection Settings

For applications that do not dynamically generate and load classes, the size of the permanent generation affects GC performance. For applications that dynamically generate and load classes (for example, JSP applications), the size of the permanent generation does affect GC performance because filling the permanent generation can trigger a full GC. Tune the maximum permanent generation with the -XX:MaxPermSize option.

Tuning the Java Heap Size

This section discusses topics related to tuning the Java heap size for performance:

- "Guidelines for Java Heap Sizing" on page 37
- "Heap Tuning Parameters" on page 38

Guidelines for Java Heap Sizing

Maximum heap size depends on the maximum address space per process.

Maximum heap space is always smaller than the maximum address space per process because the process also needs space for stack, libraries, and so on. To determine the maximum heap space that can be allocated, use a profiling tool to examine the way memory is used. Gauge the maximum stack space that the process uses and the amount of memory taken up by libraries

and other memory structures. The difference between the maximum address space and the total of those values is the amount of memory that can be allocated to the heap.

You can improve performance by increasing the heap size or by using a different garbage collector. In general, for long-running server applications, use the J2SE throughput collector on machines with multiple processors (`-XX:+AggressiveHeap`) and as large a heap space as you can fit in the free memory of your machine.

The recommended Java heap size for the Oracle Identity Analytics application is discussed in the following section.

Heap Tuning Parameters

You can control the heap size with the following JVM parameters:

- `-Xms value`
- `-Xmx value`
- `-XX:MinHeapFreeRatio=minimum`
- `-XX:MaxHeapFreeRatio=maximum`
- `-XX:NewRatio=ratio`
- `-XX:NewSize=size`
- `-XX:MaxNewSize=size`
- `-XX:+AggressiveHeap`

The `-Xms` and `-Xmx` parameters define the minimum and maximum heap sizes, respectively. Since garbage collection occurs when the generations fill up, throughput is inversely proportional to the amount of memory available. By default, the JVM grows or shrinks the heap at each garbage collection to try to keep the proportion of free space to the live objects at each collection within a specific range. This range is set as a percentage by the parameters `-XX:MinHeapFreeRatio=minimum` and `-XX:MaxHeapFreeRatio=maximum`, and the total size bounded by `-Xms` and `-Xmx`.

Set the values of `-Xms` and `-Xmx` equal to each other for a fixed heap size. When the heap grows or shrinks, the JVM must recalculate the old and new generation sizes to maintain a predefined `NewRatio`.

The `NewSize` and `MaxNewSize` parameters control the new generation's minimum and maximum sizes. Regulate the new generation size by setting these parameters to the same value. The bigger the younger generation, the less often minor collections occur. The size of the young generation relative to the old generation is controlled by `NewRatio`. For example, setting `-XX:NewRatio=3` means that the ratio between the old and young generation is 1:3, and the combined size of eden and the survivor spaces will be fourth of the heap.

By default, the application server is invoked with the Java HotSpot server virtual machine. The default `NewRatio` for the Server VM is 2: the old generation occupies 2/3 of the heap while the new generation occupies 1/3. The larger new generation can accommodate many more short-lived objects, decreasing the need for slow major collections. The old generation is still sufficiently large enough to hold many long-lived objects.

To size the Java heap:

- Decide the total amount of memory you can afford for the JVM. Accordingly, graph your own performance metric against young generation sizes to find the best setting.

- Make plenty of memory available to the young generation. The default is calculated from `NewRatio` and the `-Xmx` setting.

- Larger eden or younger generation spaces increase the spacing between full garbage collections. But young space collections could take a proportionally longer time. In general, keep the eden size between one fourth and one third the maximum heap size. The old generation must be larger than the new generation.

This is an example heap configuration used by Oracle Identity Analytics on GlassFish (RedHat Linux 64-bit) for a large role consolidation and entitlements discovery application: `-Xms2048m -Xmx2048m`

Tuning the Operating System

This section discusses tuning the operating system (OS) for optimum performance.

Server Scaling

This section provides recommendations for optimal performance scaling for the following application server subsystems:

- Processors
- Memory
- Disk Space
- Networking

Processors

The application server automatically takes advantage of multiple CPUs. In general, the effectiveness of multiple CPUs varies with the operating system and the workload, but more processors will generally improve dynamic content performance.

Static content involves mostly input/output (I/O) rather than CPU activity. If the application server is tuned properly, increasing primary memory will increase its content caching and thus increase the relative amount of time it spends in I/O versus CPU activity. Studies have shown that doubling the number of CPUs increases servlet performance by 50 to 80 percent.

Memory

For optimal Oracle Identity Analytics performance, have at least 4 Gbytes of system memory. GlassFish specific memory requirements can be found in the "Hardware and Software Requirements" section in the GlassFish release notes.

Disk Space

It is best to have enough disk space for the OS, document tree, and log files. In most cases, have 20 Gbytes of free disk space for optimal operation of Oracle Identity Analytics, with the database and application running on separate machines.

Networking

To determine the bandwidth the application needs, determine the following values:

- The number of peak concurrent users (Npeak) that the application server needs to handle.
- The average request size on your site, r. The average request can include multiple documents. When in doubt, use the home page and all its associated files and graphics.
- Decide how long, t, the average user will be willing to wait for a document at peak utilization.

    The formula for the bandwidth required is Npeakr / t

For example, to support a peak of 50 users with an average document size of 24 Kbytes, and transferring each document in an average of five seconds, requires 240 Kbytes (1920 Kbit/s). So the site needs two T1 lines (each 1544 Kbit/s). This bandwidth also allows some overhead for growth.

The application server's network interface card must support more than the WAN to which it is connected. For example, if you have up to three T1 lines, you can get by with a 10BaseT interface. Up to a T3 line (45 Mbit/s), you can use 100BaseT. But if you have more than 50 Mbit/s of WAN bandwidth, consider configuring multiple 100BaseT interfaces, or look at Gigabit Ethernet technology.