Oracle® Application Server

Best Practices Guide

10*g* (10.1.4.0.1)

B31762-02

December 2006

ORACLE®

Oracle Application Server Best Practices Guide, 10*g* (10.1.4.0.1)

B31762-02

Primary Authors: William Bathurst, Darren Calman, Fermin Castro, Michael Mesaros, Olaf Stullich, Frank Villavicencio, Mark Wilcox

Contents

3 Oracle Internet Directory

4 Security

Preface

The *Oracle Application Server Best Practices Guide* provides a collection of common practices regarding usage, deployment, or development of Oracle Application Server 10*g* (10.1.4.0.1).

Audience

This guide is intended for anyone deploying Oracle Application Server 10*g* (10.1.4.0.1).

Documentation Accessibility

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

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the Oracle Application Server Documentation Library.

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.

1

Introduction to Best Practices for the Oracle Identity and Access Management Suite

This book provides a collection of best practices for Oracle Application Server 10*g* (10.1.4.0.1), focusing on the Oracle Identity and Access Management Suite. These best practices cover configuration, deployment, and development. A best practice provides a recommendation or a common practice on how to perform certain tasks and actions. This recommendation may involve a combination of tools and manual processes to achieve a desired result. This collection covers the following technology areas:

- Oracle Access Manager
- Oracle Internet Directory
- Security
- Oracle Virtual Directory
- High availability

1.1 Key Best Practices for Oracle Identity and Access Management Suite Deployments

By following a few key best practices, you can significantly increase the success of any given deployment. These best practices are:

- Think strategically, act tactically: Any infrastructure project is long-term by definition. It is important to plan and design the solution from a strategic perspective, defining a long-term roadmap that is not necessarily tied to or limited by tactical deadlines. The plan should be broken into measurable and attainable phases that each contribute to the strategic goal, while showing return on investment (ROI) with every milestone.
- Seek the advice of the experts: This is particularly important during the initial stages of the deployment (planning, requirements gathering and design). There is tremendous value in engaging experts with years of industry experience who have completed projects in various industry verticals. The time and cost involved in engaging these experts will be quickly realized in mitigating risks, adopting industry best practices, and defining a validated strategy that will achieve success. Industry expertise is typically twofold:
  - Industry expertise: Knowledge of established policies, procedures, and standards; familiarity with compliance with governmental regulations and guidelines; information security best practices; security management and operations; technical infrastructure.

  - Technical expertise: Best use of the technology in addressing business requirements; technical infrastructure; Oracle technology and application security; identity management; access management, Web services security, information assurance, and privacy management.

- Invest in knowledge: Prior to the start of the design phase, ensure that your staff has the right knowledge of the technology and products involved. Starting the process of the knowledge transfer and familiarity with the environment at the beginning of the project will be more effective than starting this process later on. Oracle strongly recommends that you train your staff in the specific products involved in the deployment, and strive to capture and transfer knowledge throughout the deployment, so you are capable of independently supporting, maintaining and evolving the infrastructure. This investment is key to a successful deployment.

2

Oracle Access Manager

This chapter describes best practices for Oracle Access Manager. It contains the following topics:

2.1 General Best Practices

This section describes general best practices for Oracle Access Manager. It includes the following topics:

2.1.1 Deploy Oracle Access Manager in Multiple Environments to Minimize Service Disruptions

Typically, an Oracle Access Manager deployment includes at least four distinct environments:

- Development: This is the environment where you perform initial development and configuration, including applying new patches or versions of the software
- Integration: This is usually a controlled environment where application integration is tested and verified.
- QA (or Pre-Production): This environment should closely resemble production in performance and in failover and redundancy characteristics. You should use this environment to test behaviors that you expect in production or to reproduce production issues when troubleshooting.
- Production

  Keep the environments distinct to minimize service disruptions in production during rollouts and upgrades. It is a best practice to keep all of the environments at exactly the same patch level of the Oracle Access Manager software to avoid discrepancies and minimize the risk of finding difference in behavior across environments.

Implementation Details

See Also: Chapter 1, "Capacity Planning," in the *Oracle Access Manager Deployment Guide*

2.1.2 Deploy Oracle Access Manager Access and Identity Servers on Dedicated Hardware to Improve Reliability

Oracle recommends running Oracle Access Manager Access and Identity Servers on independent, dedicated hardware. For example, a typical Oracle Access Manager deployment for up to 15,000 users includes four servers, two running Identity Servers and two running Access Servers. Deploy IP switching technology in front of each pair of Web servers to provide load balancing and failover.

Implementation Details

See Also: Chapter 1, "Capacity Planning," in the *Oracle Access Manager Deployment Guide*

2.1.3 Store Configuration and Policy Data in a Separate Directory to Provide Greater Deployment and Upgrade Flexibility

If feasible, store the Oracle Access Manager configuration and policy data in a separate directory from the user and group data. This will allow for greater flexibility during upgrade and minimize the impact on the user and group directory. The user and group directory is typically a shared, enterprise directory, so separating the data this will be particularly beneficial when workflow-driven processes generate significant load on the directory. Configuring separate logical directory instances will also ensure that each directory can be tuned and managed independently, improving overall performance.

If directories cannot be separated due to hardware or topology related issues, at minimum you should create a dedicated suffix to hold the Oracle Access Manager configuration and policy data.

Implementation Details

See Also: Chapter 2, "Performance Tuning," in the *Oracle Access Manager Deployment Guide*

2.1.4 Point Directly to a Domain Controller to Avoid Potential Data Inconsistency Problems

When deploying in a Microsoft Active Directory environment, always point directly to a domain controller to avoid potential data inconsistency problems

When using Microsoft Active Directory as the user and group directory, ensure that you point directly to the domain controller and not to the DNS alias. This will avoid problems that are caused by transient inconsistencies. For example, this practice avoids the possibility of dynamic DNS or round robin aliasing diverting connections to servers that are slow, remote, or contain out-of-date data. To implement high availability in an Active Directory environment, configure each of the domain controllers as a directory connection, and then tune the performance for reads and writes from Oracle Access Manager.

This recommendation also applies to Active Directory forests that contain multiple sub-domains. For this you should create separate directory profiles for each sub-domain in addition to the root domain, with each sub-domain pointing at the appropriate domain controller server or servers.

Implementation Details

See Also: Appendix A, "Deploying with Active Directory," in the *Oracle Access Manager Identity and Common Administration Guide*

2.1.5 Use LDAP Over SSL Rather than ADSI When Connecting to Microsoft Active Directory

When deploying Oracle Access Manager on Windows against Microsoft Active Directory, it is important to consider whether using LDAP over SSL is more appropriate than ADSI, as LDAP over SSL typically performs and scales better than ADSI, particularly in environments with high transaction volume. Also, using LDAP over SSL allows Oracle Access Manager (Access Server, Identity Servers, and Policy Manager) to rely on specified Active Directory instances. In contrast, when using ADSI, the specific Active Directory instance that Oracle Access Manager connects to is determined on-the-fly. This instance select may be undesirable if the overall response and performance across all available Active Directory domain controllers varies significantly.

Implementation Details

See Also: Appendix A, "Deploying with Active Directory," in the *Oracle Access Manager Identity and Common Administration Guide*

2.1.6 When Deploying on Top of Microsoft Active Directory, Fine Tune the Appropriate Active Directory Configuration Parameters to Optimize Performance

When deploying in a Microsoft Active Directory environment, it is important that the configuration parameters in Active Directory be set appropriately, so that Oracle Access Manager's overall performance is optimized. Table 2–1 lists the most relevant Active Directory configuration parameters.

Table 2–1 Active Directory Configuration Parameters

Active Directory Configuration Parameter	Description	Default Value	Impact for Oracle Access Manager
`MaxActiveQueries`	Specify the maximum number of concurrent LDAP search operations that are permitted to run at the same time on a domain controller. When this limit is reached, the LDAP server returns a "busy" error. **Note:** This control has an incorrect interaction with the `MaxPoolThreads` value. `MaxPoolThreads` is a per-processor control, while `MaxActiveQueries` defines an absolute number. Starting with Windows Server 2003, `MaxActiveQueries` is no longer enforced. Additionally, `MaxActiveQueries` does not appear in the Windows Server 2003 version of `Ntdsutil.exe`.	20	This value should be greater than the total number of service threads in Oracle Access Manager, for all service threads to be able to perform search operations at the same time. The total number of service threads in Oracle Access Manager is the summation of number of threads in Identity and Access servers and the number of threads in the Web server hosting the Policy Manager.
`MaxConnections`	Specify the maximum number of simultaneous LDAP connections that a domain controller will accept. If a connection comes in after the domain controller reaches this limit, the domain controller drops another connection.	5000	This value should be greater than or equal to the number of connections that Oracle Access Manager establishes with any Active Directory domain controller.
`MaxConnIdleTime`	Specify the maximum time, in seconds, that the client can be idle before the LDAP server closes the connection. If a connection is idle for more than this time, the LDAP server returns an LDAP disconnect notification.	900 seconds	If the maximum session time is set in Oracle Access Manager, then this value should not slightly higher than it.

Table 2–1 (Cont.) Active Directory Configuration Parameters

Active Directory Configuration Parameter	Description	Default Value	Impact for Oracle Access Manager
`MaxPageSize`	Specify the value for controlling the maximum number of objects that are returned in a single search result, independent of how large each returned object is. To perform a search where the result might exceed this number of objects, the client must specify the paged search control. This is to group the returned results in groups that are no larger than the `MaxPageSize` value. To summarize, `MaxPageSize` controls the number of objects that are returned in a single search result.	1,000	This value should be greater than the number of entries returned in any search request made by any Oracle Access Manager component. Oracle Access Manager component performs search operations for the following two cases: ■ When a user requests search on other users or groups and other entities ■ Oracle Access Manager component internally performs searches on configuration data while processing requests If blank, the search is allowed (in general, any search that results in all user entries in the system), then this value should be greater than the number of users in the system. If in general this kind of user searches are restricted or no one does these kinds of requests, then this value should be greater than the highest number of nodes under the following two nodes in Oracle Access Manager's configuration data: ■ `obapp=PSC,o=Oblix,<config_base>` ■ `obcontainerId=workflowInstances,o=Oblix,,<config_base>`
`MaxPoolThreads`	The maximum number of threads-per-processor that a domain controller dedicates to listening for network input or output. This value also determines the maximum number of threads per-processor that can work on LDAP requests at the same time.	4 threads per-processor	If the Identity or Access Server run a single processor server, then this value should be greater than the number of connections established by Oracle Access Manager. This way, the domain controller can perform all operations in parallel.

Table 2–1 (Cont.) Active Directory Configuration Parameters

Active Directory Configuration Parameter	Description	Default Value	Impact for Oracle Access Manager
`MaxQueryDuration`	The maximum time in seconds that a domain controller will spend on a single search. When this limit is reached, the domain controller returns a `timeLimitExceeded` error. Searches that require more time must specify the paged results control.	120 seconds	This value should be greater than the time limit value specified in database instance of the directory profile, which points to this domain controller. Otherwise, before the time limit specified in database instance itself is reached, Oracle Access Manager may get a `timeLimitExceeded` error from Active Directory.

Implementation Details

See Also: Appendix A, "Deploying with Active Directory," in the *Oracle Access Manager Identity and Common Administration Guide*

2.1.7 Size and Tune the Environment to Support Production Deployment

You should run a thorough, extended load test and benchmark analysis before deploying Oracle Access Manager into production. This will allow you to fine tune and predict the behavior of the overall system.

Based on the performance figures, you can tune performance, for example, by altering cache settings, timeout values, the number of directory connections, and increasing the number of threads on the Identity or Access Servers.

Implementation Details

See Also: Chapter 2, "Performance Tuning," in the *Oracle Access Manager Deployment Guide* for details on how to set these parameters based on performance results

2.1.8 Host Administration Interfaces on Dedicated Web Servers to Protect the Environment

It is a best practice to dedicate one or two internal-facing, highly-secured Web servers to host the administrative interfaces for Oracle Access Manager. The administrative interfaces consist of the Identity System Console and the Access System Console. This will help protect the authentication and authorization system in the event the application Web servers are compromised.

Implementation Details

See Also: Chapter 1, "Capacity Planning," in the *Oracle Access Manager Deployment Guide*

2.1.9 Use SSL Transport between Components to Secure the Environment

In any production environment, you should secure all transport between Oracle Access Manager components with SSL encryption transport, using either Simple or Certificate Mode. All intranet and extranet communications should be encrypted.

When enabling SSL with Simple or Certificate mode, ensure that the clocks of the servers hosting the WebGates, AccessGates, and WebPass as well as those hosting the Access or Identity Servers are within 1 minute of absolute time difference. This includes ensuring that daylight savings and time zones have no more than a 1-minute difference between the clocks in GMT. Oracle recommends that you use the Network Time Protocol (NTP) for ensuring server clocks are synchronized.

Implementation Details

See Also: Chapter 8, "Changing Transport Security Modes," in the *Oracle Access Manager Identity and Common Administration Guide*

2.1.10 Store Audit Trails in a Database to Maximize the Usability of Audit Data

When possible, use a database for recording and storing all Oracle Access Manager audit data. This protects the audit information and makes it easier to generate audit trails and reports.

Implementation Details

See Also: Chapter 10, "Logging," in the *Oracle Access Manager Identity and Common Administration Guide*

2.1.11 Take Steps to Simplify Management of Your Environment

You can take a number of steps to simplify management of your Oracle Access Manager environment:

- Standardize the layout of the file system in all your environments. For example, locate all Oracle Access Manager-specific files in the same directory path across all environments.
- When feasible, use the same versions of the Web server and directory server across all environments.
- Create a `/custom` directory and store all custom code in it. The `/custom` directory should not be in the installation directory of the Oracle Access Manager component. It is important to keep it separate because during re-installation or de-installation of Oracle Access Manager components, all subdirectories are deleted.
- Most importantly, document any changes or customizations to the environment.

Implementation Details

See Also: Chapter 1, "Upgrade Overview and Planning," in the *Oracle Access Manager Upgrade Guide*

2.2 Access System Best Practices

This section describes Access System best practices. It includes the following topics:

- Section 2.2.1, "Use IP Validation, HTTPS, and Secure Cookies to Mitigate The Risk of a Cookie Reply Attack"
- Section 2.2.2, "Avoid Using Nested Groups for Authorization to Improve Group Membership Performance"

2.2.1 Use IP Validation, HTTPS, and Secure Cookies to Mitigate The Risk of a Cookie Reply Attack

Oracle recommends that you always enable IP validation to mitigate the risk of a cookie reply attack. If exceptions are required, for example, when deploying using a reverse proxy topology, ensure that only allowed IP addresses are included in the exception list. Avoid ever turning off IP validation.

To avoid the risk of cookie reply attacks, you can also deploy content securely over HTTPS. This prevents unauthorized clients from eavesdropping on the `ObSSOCookie`. Also, specify the parameter `ssoCookie:secure` for the Challenge Parameter for a Form, Basic, or External challenge method to ensure that the `ObSSOCookie` set during authentication is sent only through SSL. This prevents the `ObSSOCookie` from being sent back to a non-secure Web server. This parameter setting requires configuring all protected Web servers for SSL. An SSL Web server will not perform single sign-on with a non-SSL Web server. A browser will not return a secure cookie obtained from an SSL Web server to a non-SSL Web server in the same domain.

Implementation Details

See Also: Chapter 5, "Configuring User Authentication," in the *Oracle Access Manager Access Administration Guide*

2.2.2 Avoid Using Nested Groups for Authorization to Improve Group Membership Performance

Avoid using nested groups for authorization to improve group membership performance

By default, Access Server checks for static, dynamic and nested group memberships to determine authorizations. Evaluation of nested group memberships is extremely expensive to evaluate. If you are not using nested groups, disabling the nested group membership check will improve performance. You can disable the nested group membership check by setting the `TurnOffNestedGroupEvaluation` parameter manually to `True` in the `globalparams.xml` file.

Implementation Details

See Also: Chapter 2, "Performance Tuning," in the *Oracle Access Manager Deployment Guide*

2.2.3 Configure Dynamic Groups Rather than Authorization Filters to Simplify Authorization Administration

Generally, you should use dynamic groups instead of authorization filters to specify authorization rules. Dynamic groups allow you to separate the management of the filter ("Who is a virtual member of the group?") from the management of the authorization rule. This enables you to delegate the management of the authorized role to a class of administrators that is separate from those who configure the access policies. Additionally, group management allows tracking of changes and approvals for changes, whereas Authorization rule filters do not.

Implementation Details

See Also: Chapter 5, "Configure User Authentication," in the *Oracle Access Manager Access Administration Guide*

2.2.4 Performance Considerations when Using ObMyGroups

There are several situations where the use of the special obmygroups action value proves to be a powerful approach for further integrating the Access System with applications to provide role-based information for the logged in user, in particular when the application needs to know which groups the particular user belongs to as this determined which menu items or functions would be presented.

Using `ObMyGroups` as an authentication or authorization action requires consideration of performance implications that resolving user group membership could have in the system for the LDAP directory and Access Server, and consequentially the Web server and application being accessed.

In general the user/group evaluation is an expensive operation from an Access Server perspective. Oracle strongly recommends that these always be configured with a qualifying LDAP filter. For example, enter `obmygroups:ldap:///o=company,c=us??sub?(group_type=role).`

Depending on the number of groups being searched, `ObMyGroups` processing may take a significant amount of time. It is best to specify obmygroups in an authentication rule rather than an authorization action and, if possible, have the action be a cookie so that the data is available to other applications under the same Web server without incurring an additional toll. And when `ObMyGroups` is used in an authorization rule, limit its use to as few resources (URLs) as practical.

Oracle strongly recommends that a thorough performance test, with the specific use cases relevant to `ObMyGroups` actions, be designed and executed prior to rollout. This allows you to benchmark, tune, and understand the actual performance and response times involved in your specific environment.

Implementation Details

See Also:

- Chapter 5, "Configure User Authentication," in the *Oracle Access Manager Access Administration Guide*
- Chapter 2, "Performance Tuning," in the *Oracle Access Manager Deployment Guide*

2.2.5 Consider Deploying WebGates On Reverse Proxies to Simplify Management

Consider deploying WebGates on reverse proxies to simplify management

There are a number of benefits to deploying WebGates on reverse proxies. These include:

- You can protect all Web content from a single logical component by directing all requests through the proxy. This is true even for platforms that are not supported by Oracle Access Manager. If you have different types of Web servers, for example, iPlanet, Apache, and so on, on different platforms, for example, MacOS, Solaris x86, mainframe and so on, all content on these servers can be protected. A reverse proxy can be a workaround for unsupported Web servers, eliminating the need to write custom AccessGates for unsupported Web servers or for platforms where there is no AccessGate support.
- You can install a WebGate on only the reverse proxy, rather than on every Web server. This creates a single management point. You can manage the security of all of the Web servers through the reverse proxy without establishing a footprint on the other Web servers.
- A reverse proxy provides architectural flexibility. Reverse proxies can enable you to expose the same application on the intranet and the extranet without requiring any changes to the application already deployed.

The main pitfall of using a proxy is the extra work involved in setup. If you deploy the WebGate on a Web server that resides behind a reverse proxy, the following are required:

- Ensure that any Web server that uses the reverse proxy for authentication only accept requests from the reverse proxies. You must configure the WebGate deployed on this Web server to not enforce IP validation on requests coming from the reverse proxy server or servers acting as its front end. You must configure the IP addresses of the reverse proxy server or servers in the IP Validation list for the WebGate. Oracle does not recommend turning IP validation off for the WebGate because it can expose a security risk.
- Update the virtual hosts that are configured in the Policy Manager so that the Access System intercepts requests that are sent to the reverse proxy.
- Prevent people from circumventing the proxy by entering URLs that point directly to the back-end system. You can add Access Control List (ACL) statements in the server to prevent users from bypassing the reverse proxy and directly accessing restricted content. Or, you can configure firewall filters.
- Since the proxy processes all user requests, you must deploy enough proxy servers to enable the system to handle the load.
- Redirect all existing URLs to the host name and port number of the reverse proxy server. This often requires configuring the reverse proxy to inspect content and rewrite URLs, for example, to prevent any absolute HTML links from resulting in a broken link. This is available in most reverse proxies, and it is functionality that is independent of the Access System. It is a best practice that you configure URL links exposed to the front-ended applications to contain only relative URLs (`../../sub-path/resource`) rather than absolute URLs (`http://hostname.domain:port/path/resource`) or pseudo-relative URLs (that is, `/path/resource`). Absolute URLs can break links on the end user's browser when deployed behind a reverse proxy.

Implementation Details

> **See Also:** Chapter 3, "Configuring WebGates and Access Servers," in the *Oracle Access Manager Access Administration Guide*

2.2.6 Design Document Protection Policies to Minimize WebGate Calls to the Access Server

For example, when specifying policies to protect all the documents on a Web server, there are two approaches that will work:

- Protecting all the documents from the root of the document tree, and specifically allowing access to specified documents
- Setting the `DenyOnNotProtected` flag, and specifically allowing access to specified documents

In general, the second approach will provide better performance. When protecting Web documents from the root, the WebGate will always need to contact the Access Server for each request to check if the user is authorized to access the resource. This will place additional load on the Access Server. When using the `DenyOnNotProtected` flag, the WebGate caches information from Access Server on whether a particular URL is protected by the Access System. As a result, it can simply deny access to subsequent requests for unprotected resources without contacting the Access Server thereby reducing server overhead.

Implementation Details

See Also: Chapter 3, "Configuring WebGates and Access Servers," in the *Oracle Access Manager Identity and Common Administration Guide*

2.2.7 Use Best Practices When Configuring Form-based Authentication to Avoid Login Errors

When implementing form-based authentication with Oracle Access Manager, develop code in such a way as to avoid login errors. This includes embedding code to validate input fields in the form to avoid posting the wrong credentials. For example, you can check that user name and password fields are not blank. In addition, use HTML code that prevents content caching, for example:

```
<meta http-equiv="pragma" content="no-cache">
```

Implementation Details

See Also: Appendix A, "Form-Based Authentication," in the *Oracle Access Manager Access Administration Guide*

2.2.8 Code API-Based Plug-ins to Avoid Access Server crashes

The following guidelines can reduce the risk of introducing instability to the Access Server:

- Since the Access Server is multithreaded, you should ensure that any API-based plug-ins that are deployed on the Access Server are thread-safe.
- Ensure that all authorization and authentication API plug-ins are persistent, and improve performance by implementing connection pooling and caching.
- Initialize all global and local variables used in the plug-in functions.
- Oracle recommends that all authorization and authentication plug-ins take input parameters from the Access Server in order to specify configuration information.

Implementation Details

See Also: Chapter 6, "Customizing Access Control with Plug-Ins," in the *Oracle Access Manager Customization Guide*

2.2.9 Use Best Practices to Secure Access Manager SDK (AccessGate) Clients

The configuration of Access API clients includes a secret password between the Access Server and the AccessGate configuration to prevent invalid clients from connecting to the Access Server. To protect these passwords, you should implement SSL to encrypt the communication between the Access Server and the Access API clients. In addition, treat the single sign-on token (typically the content of the `ObSSOCookie`) as a password, and do not provide it to external applications.

Implementation Details

See Also: Chapter 6, "Customizing Access Control with Plug-Ins" in the *Oracle Access Manager Customization Guide*

2.3 Identity System Best Practices

This section describes Identity System best practices. It includes the following topics:

- Section 2.3.1, "Avoid Searches to Improve Identity Administration Performance"
- Section 2.3.2, "Use the Manage Members Page of the Group Manager Application to Efficiently Manage Large Groups"
- Section 2.3.3, "Configure a Single Idle Timeout for the Entire Oracle Access Manager Deployment to Avoid Potential Discrepancies in User Behavior"
- Section 2.3.4, "Turn Off Tracking to Improve Workflow Performance"
- Section 2.3.5, "Periodically Clean Up Workflow Tickets to Improve Directory Performance"
- Section 2.3.6, "Build Event API Plug-Ins for Performance"
- Section 2.3.7, "Use PresentationXML to Customize the Look and Feel of Embeddable User Interface Elements"
- Section 2.3.8, "Use an XML/XSL Editor When Developing PresentationXML to Expedite Development and Test"
- Section 2.3.9, "Always Work from a Copy of The Default Style Sheet"
- Section 2.3.10, "Use Caution When Implementing Javascript Code in PresentationXML"

2.3.1 Avoid Searches to Improve Identity Administration Performance

When possible, keep search bases to a minimum. Instead, apply ACLs to the class attribute of the user object. This is known to improve performance. Also, avoid configuring search bases using substitution syntax, as this is known to adversely affect performance. Avoid search bases or ACLs that contain substring searches, for example "`...(...=*something*)...`". And avoid setting several search bases for each User, Group, or Org object class.

Implementation Details

See Also: Chapter 3, "Making Schema Data Available to the Identity System," in the *Oracle Access Manager Identity and Common Administration Guide*

2.3.2 Use the Manage Members Page of the Group Manager Application to Efficiently Manage Large Groups

For group management use the Manage Members page. This page optimizes the management of large groups (defined as static groups with 1000 or more members), as opposed to defining the member semantic attribute as part of the group profile page. This will significantly improve performance when managing large groups.

Implementation Details

See Also: Chapter 4, "Configuring User, Group, and Organization Manager," in the *Oracle Access Manager Identity and Common Administration Guide*

2.3.3 Configure a Single Idle Timeout for the Entire Oracle Access Manager Deployment to Avoid Potential Discrepancies in User Behavior

In general, Oracle Access Manager's timeout values should be configured to be the lowest of all application timeouts. Timeouts are enforced by WebGate or the Web server and not by application. The goal, therefore, is to avoid applications from timing out before Oracle Access Manager and thus having to potentially handle session issues within the applications.

In general, there are a number of considerations in selecting timeout values. For example, applications which are able to regenerate a session from an existing Oracle Access Manager session (or header variable) can timeout earlier than Oracle Access Manager. Oracle Access Manager's Identity System, in fact, is a good example of an application where having a shorter session timeouts than WebGates is recommended. In general, however, these timeout values should be close to each other.

One exception to this rule is for AccessGates, which are typically deployed downstream from a WebGate, for example supporting a BEA WebLogic implementation. In this case it is recommended that the AccessGate have a greater idle timeout than the WebGate to avoid the problem of a fresh browser session being rejected by the downstream AccessGate. For AccessGates, Oracle recommends configuring the idle and maximum timeouts to be the same.

Implementation Details

See Also: Chapter 3, "Configuring WebGates and Access Servers," in the *Oracle Access Manager Access Administration Guide*

2.3.4 Turn Off Tracking to Improve Workflow Performance

If workflows do not require any other input or approval steps, it is a good idea to enable the `WfInstanceNotRequired` parameter. This parameter prevents the generation and storage of workflow tracking data that can generate overhead for the directory server.

Implementation Details

To change this parameter:

1. Set the `WFInstanceNotRequired` flag to `true` in the following file:

   install-path`/identity/oblix/data/common/workflowdbparams.xml`

2. Restart the Identity Server.

> **See Also:** Chapter 2, "Performance Tuning," in the *Oracle Access Manager Deployment Guide*

2.3.5 Periodically Clean Up Workflow Tickets to Improve Directory Performance

Monitor the number of workflow tickets that are stored in the directory server, and periodically delete old tickets manually or using a script-based utility.

Implementation Details

> **See Also:** Chapter 2, "Performance Tuning," in the *Oracle Access Manager Deployment Guide*

2.3.6 Build Event API Plug-Ins for Performance

Avoid EXEC-type plug-ins, which spawn external processes. If you want an Identity Event API plug-in to trigger other events in the Identity Server using IdentityXML, ensure that the request goes to an Identity Server instance other than the one triggering the Event API plug-in. This practice balances the system load. Also, Event API plug-ins using C/C++ as shared objects (`.so` files) for performance and stability reasons.

The following are other best practices to apply in the development of Identity Event API plug-ins:

- All Identity Event API plug-ins must be thread-safe.
- Identity Event API plug-ins are not persistent, so you must initialize all global and local variables used in the plug-in functions.
- Use a single library for multiple Event API plug-in functions to minimize runtime image size.
- The plug-in should support using a configuration file to alter and adapt its operation in case of requirement changes.

Implementation Details

> **See Also:** Chapter 3, "Identity Event Plug-in API," in the *Oracle Access Manager Developer Guide*

2.3.7 Use PresentationXML to Customize the Look and Feel of Embeddable User Interface Elements

Oracle Access Manager Identity combines Extensible Style Language (XSL) stylesheets and Extensible Markup Language (XML) data to dynamically create almost all of the pages presented to its users. This capability, known as PresentationXML, provides developers with design flexibility and avoids the need for static HTML content.

PresentationXML is the recommended approach if your intent is to deal with front-end, user interface issues, for example, look and feel, layout of the tags,

enhancing the navigation, and so on. It is not recommended for back-end logic, for example, pre-filling values based on data on a database, computing values based on other input values, communicating with external systems, and so on.

Implementation Details

See Also: Chapter 2, "Designing the GUI with PresentationXML," in the *Oracle Access Manager Customization Guide*

2.3.8 Use an XML/XSL Editor When Developing PresentationXML to Expedite Development and Test

To expedite development and testing, use a powerful XML or XSL editor, for example, XMLSpy. These editors provide an integrated development environment (IDE) to simplify and speed up the process of XSL programming.

Implementation Details

See Also: Section "Setting Up Your Environment to Customize the Style Sheets" in Chapter 2, "Designing the GUI with PresentationXML," of the *Oracle Access Manager Customization Guide*

2.3.9 Always Work from a Copy of The Default Style Sheet

Before modifying or using a style sheet, create a new style based on the default (`style0`) style of Oracle Access Manager. Replicate all related graphics, stylesheets and Javascripts in the Identity Server and WebPass so that the default style remains unchanged.

Implementation Details

See Also: Chapter 2, "Designing the GUI with PresentationXML," in the *Oracle Access Manager Customization Guide*

2.3.10 Use Caution When Implementing Javascript Code in PresentationXML

When the need arises to insert JavaScript code to a front-end page through PresentationXML, the best practice is to encapsulate all of the JavaScript code into a file, then include the file in the XSL file. At deployment time, you must deploy this file on the appropriate WebPass installation directory.

When including JavaScript code in PresentationXML, do not modify the main `misc.js` file in the WebPass installation directory. This file is used for client-side processing and is common to all Oracle Access Manager components. Any modification can adversely affect all components.

Implementation Details

See Also: Chapter 2, "Designing the GUI with PresentationXML," in the *Oracle Access Manager Customization Guide*

3

Oracle Internet Directory

This chapter describes best practices for Oracle Internet Directory. It contains the following topics:

3.1 Use bulkload Utility to Bootstrap System

The bulkload utility checks standard LDIF formatted files for schema violations and duplicates, and generates SQL*Loader intermediate files for fast loading into the database tables underlying Oracle Internet Directory. Use the bulkload utility whenever there is an initial bootstrap required. For example, when setting up synchronization with Microsoft Active Directory or other LDAP directory servers.

Oracle recommends passing the LDIF file output from third-party LDAP directories into bulkload `check=true` mode, which will alert you to any problems with your existing LDAP schema.

Most third-party LDAP directories (including Oracle Internet Directory) support output to LDIF without any operational attributes (which typically cannot be loaded into another vendor's directory). If you are loading data into Oracle Internet Directory from another directory, which does not support this, you will have to manually remove any operational attributes prior to sending the LDIF file to bulkload `generate=yes` mode.

If your input LDIF file is from another Oracle Internet Directory instance, then you must use the `restore=yes` option to `bulkload.sh` to preserve these operational attributes as is during the bulkload.

Implementation Details

See Also: Chapter 4, "Oracle Internet Directory Data Management Tools," in the *Oracle Identity Management User Reference*

3.2 Replicate to Provide High Availability

Oracle Internet Directory supports both multimaster and fan-out styles of directory replication.

For high availability, consider placing an Oracle Internet Directory multimaster replication group behind a network load balancer to provide a single IP address to your LDAP client applications. If a replicated node becomes unavailable, you can configure the load balancer to re-route requests automatically to an available server.

Additionally, each Oracle Internet Directory node can run on Oracle Real Application Cluster, further improving availability through increased database uptime and data availability.

Implementation Details

See Also:

- Chapter 3, "Oracle Internet Directory Management Planning," in the *Oracle Identity Management Infrastructure Administrator's Guide*
- Part III, "Oracle Internet Directory in High Availability Topologies," in the *Oracle Application Server High Availability Guide*

3.3 Use TLS/SSL Binding to Secure Traffic

TLS/SSL is considered the Internet standard protocol for highly secure transportation of data. In addition to the strong PKI authentication using digital certificates, TLS/SSL provides multiple data integrity and data encryption layers to protect your communication channels. SSL provides multiple cipher suites with varieties of encryption algorithms for many security levels.

Oracle Internet Directory supports three TLS/SSL authentication modes:

1. Confidentiality mode

   In this mode, SSL cipher suites use the Diffie-Hellman algorithm to generate a session key for client or server at run time. The session key will be used to encrypt the communication channel. No server or user SSL wallet is necessary. In this mode, the channel will be encrypted using a Diffie-Hellman key.

   Note: Diffie-Hellman algorithm is used to generate an asymmetric key pair for key exchange only not for a symmetric key for encryption key.

2. Server Authentication only mode

   This mode essentially uses certificates for authentication. The client needs to verify the server certificate. This mode is most commonly used in the Internet

environment since any client that needs to communicate with aa SSL server does not require a certificate. A client can use their user and password identification to authenticate itself to the server. The username and password are protected by SSL encryption when being transferred on the wire.

3. Server and Client Authentication mode (Mutual authentication)

   In this mode, both client and server use X.509 v3 certificates to authenticate each other. First, the client authenticates the server by validating its certificate. In return, the server also requires the client to send its certificate to prove its authenticity.

In addition to choosing an authentication mode, you should choose appropriate security algorithms.

Implementation Details

See Also: Chapter 17, "Secure Sockets Layer (SSL) and the Directory," in the *Oracle Internet Directory Administrator's Guide*

3.4 Use Backup and Restore Utilities to Secure Data

Depending on your Oracle Application Server enterprise topology, you may want to consider backing up Oracle Internet Directory as part of backing up your entire application server environment.

Implementation Details

See Also:

- Chapter 15, "Backup and Restoration of a Directory," in the *Oracle Application Server Administrator's Guide* before deciding on an overall backup and recovery strategy for all of your Oracle Identity Management Infrastructure component
- Section IV, "Backup and Recovery," in the *Oracle Application Server Administrator's Guide* for general application server backup and recovery strategies

3.5 Monitor and Audit Oracle Internet Directory to Improve Availability

You can monitor and audit Oracle Internet Directory in one of four ways:

- The Oracle Enterprise Manager LDAP page provides a simple way to monitor the LDAP service and determine if it is up and running under its associated load for a standalone Oracle Internet Directory
- If more detailed information about the underlying DB and detailed statistics are required as well as to monitor other Identity Management components the usage of the Identity Management Grid Control is recommended
- You can also check the log files of various LDAP processes to ensure there are no errors showing up.
- LDAP audit log service provides more granular information such as security violation information or sensitive events. You can further customize the audit log to specific directory operations and events. Keep in mind that a large amount of data will be generated

Oracle recommends that you perform, at the very least, a weekly review of the audit and error logs. System administrators can do a more regular review with Enterprise Manager or Identity Management Grid Control to provide better availability. Monitoring bind and compare operations can be done by following the approach mentioned in Chapter 14, "Logging, Auditing, and Monitoring the Directory," in the *Oracle Internet Directory Administrator's Guide*.

Implementation Details

See Also:

- Chapter 5, "Identity Management Grid Control Plug-in," in the *Oracle Identity Management Infrastructure Administrator's Guide*
- Chapter 14, "Logging, Auditing, and Monitoring the Directory," in the *Oracle Internet Directory Administrator's Guide*

3.6 Use OPMN to Manage Oracle Internet Directory Processes

In Oracle Application Server, you no longer need to run `oidmon` and `oidctl` to start and stop Oracle Internet Directory processes. OPMN stores the proper sequences and controls these services.

Implementation Details

See Also: Chapter 6, "Process Control of Oracle Internet Directory Components," in the *Oracle Internet Directory Administrator's Guide*

3.7 Assign Oracle Internet Directory Privileges to Limit Access

While it is possible to install Oracle Application Server as an Oracle Internet Directory super user, Oracle recommends that this not be done, as it imparts more privileges than required.

To install Oracle Application Server, a user needs to be a member and owner of the Oracle Application Server administrator's group.

When installing Oracle Application Server, the directory administrator should add the installation user as a member and owner of the administrator's group. The administrator should then remove the member as the owner once the installation has completed.

3.8 Change Access Control Policies to Control User Administration

Oracle Internet Directory administrators should change the default access control policies to better control user administration as required.

Oracle Internet Directory administrators should adjust the default access control and password policies using Oracle Directory Manager, in accordance with specific administrative policies for directory access and passwords. This adjustment includes both value and state parameters.

Implementation Details

See Also: Chapter 18, "Directory Access Control," and Chapter 19, "Password Policies in Oracle Internet Directory," in the *Oracle Internet Directory Administrator's Guide*

3.9 Use User Attributes and Password Hints to Make Resetting Credentials Easier

Users that forget their OracleAS Single Sign-On passwords can reset them on their own by using the Oracle Internet Directory Self-Service Console. You must authenticate yourself in one of the following ways:

- If, while previously changing their password, a user specified a password hint question, then the Confirm Additional Personal Information window will prompt the user for the correct answer to the reminder question when attempting a password reset.
- Users who have not previously set a password hint will be presented with the Confirm Additional Personal Information window. This window prompts the user for other personal data, as configured by your administrator.

Implementation Details

> **See Also:** Chapter 4, "Managing Your Profile with the Oracle Internet Directory Self-Service Console," in the *Oracle Identity Management Guide to Delegated Administration*

3.10 Incorporate Group Assignment During User Creation to Avoid Multiple Steps

Rather than creating users and assigning them to groups as separate steps, consider incorporating the group assignment step during user creation.

Implementation Details

To do this:

1. Log in to the Oracle Internet Directory Self-Service Console as a Oracle Delegated Administration Services privileged user (`orcladmin` or designate).
2. Select the **Configuration** tab.
3. Select **User Entry** > **Add Role**.
4. Search for and select any commonly-subscribed group entries.

Whenever you or any other Oracle Delegated Administration Services privileged user performs a Create User sequence, the list of specified groups will appear in the next-to-last step, in a section called **Roles Assignment**. Simply click whichever checkboxes are relevant to the newly-created user, and that user will automatically be made a member of all the groups you specify.

> **See Also:** Chapter 5, "Managing Users and Groups with the Oracle Internet Directory Self-Service Console," in the *Oracle Identity Management Guide to Delegated Administration*

3.11 Configure Active Directory Synchronization to Enable Windows Native Authentication

Prior to configuring Windows Native Authentication, be sure to first configure the Active Directory Connector and bootstrap the appropriate `cn=users` and `cn=groups` containers within your desired Oracle Identity Management Realm. Do not configure

the External Authentication Plug-in for Active Directory if your goal is to enable Windows Native Authentication

See Also:

- Chapter 19, "Integrating with Microsoft Active Directory," in the *Oracle Identity Management Integration Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*

3.12 Oracle Directory Integration Platform Best Practices

This section describes Oracle Directory Integration Platform best practices. It includes the following topics:

- Section 3.12.1, "Use Identity Management Realms to Build Connectivity Between Oracle Internet Directory and Third-Party Directories"
- Section 3.12.2, "Configure Synchronization Service to Enable Users to Interact with Deployed Applications"
- Section 3.12.3, "Synchronize Oracle Human Resources and Oracle Internet Directory to Provide Access to OracleAS Single Sign-On and Oracle Delegated Administration Services"

3.12.1 Use Identity Management Realms to Build Connectivity Between Oracle Internet Directory and Third-Party Directories

Use Oracle Directory Integration Platform to build connectivity between Oracle Internet Directory and third-party directories. This feature provides seamless integration with other Oracle products. It enables the Oracle products to work in the presence of third-party directories in the enterprise and also provides sharing with the same identities in other directories.

You can join or unify the different identities for the same enterprise user from multiple LDAP directories into a single global identity in Oracle Internet Directory using Oracle Directory Integration Platform. Oracle Directory Integration Platform facilitates a true single sign-on environment in an enterprise using Oracle Internet Directory and Oracle Application Server Single Sign-On.

Implementation Details

See Also:

- Chapter 3, "Identity Management Infrastructure Deployment Planning," in the *Oracle Identity Management Infrastructure Administrator's Guide*
- Chapter 17, "Third-Party Directory Integration Concepts and Considerations," in the *Oracle Identity Management Integration Guide*

3.12.2 Configure Synchronization Service to Enable Users to Interact with Deployed Applications

When configuring Oracle Directory Integration Platform, specify only the containers and attributes, which are required in the connected directory or in Oracle Internet Directory. You can use LDAP filters as part of mapping configuration profiles to screen out unwanted attribute data and keep synchronization simple.

Set each connector and its associated mapping configuration file to an appropriate scheduling interval. No connector needs to fire at the same time or at the same interval as any another, as they are completely independent of one another.

When synchronizing external users and groups into Oracle Internet Directory for use with Oracle Application Server, be sure to establish connectors to the appropriate Identity Management Realm `cn=users` and `cn=groups` container. Oracle Directory Integration Platform will then provision all inbound user entries with the Oracle-specific attributes needed to enable users to interact with their deployed Oracle applications.

Implementation Details

See Also: Chapter 17, "Third-Party Directory Integration Concepts and Considerations," in the *Oracle Identity Management Integration Guide*

3.12.3 Synchronize Oracle Human Resources and Oracle Internet Directory to Provide Access to OracleAS Single Sign-On and Oracle Delegated Administration Services

If you use Oracle Human Resources as the source of truth for employee data in your enterprise, then you must synchronize between it and Oracle Internet Directory. Since the `Last Successful Execution Time` connector profile attribute is used to fetch the desired changes from connected directories at a given time, set it initially to some date in the past. Then enable the profile. Note this technique will potentially cause all entries in the connected directory to be synchronized all at once into Oracle Internet Directory.

It is a good idea to synchronize user data from connected directories to the public `cn=users` container within an Oracle Internet Directory Identity Management realm. This way, all users are immediately accessible to OracleAS Single Sign-On and Oracle Delegated Administration Services, such as the Self-Service Console.

Implementation Details

See Also:

- Chapter 10, "Synchronization with Oracle Human Resources," in the *Oracle Identity Management Integration Guide*
- Section 9.1, "bulkload," in the *Oracle Internet Directory Administrator's Guide*

4

Security

This chapter describes security and management best practices for Oracle Application Server. It includes the following topics:

4.1 General Best Practices

This section describes general best practices for security and management. It includes the following topics:

4.1.1 HTTPS Best Practices

The following are recommended for using HTTPS with Oracle Application Server:

- **Configure Oracle Application Server to fail attempts that use weak encryption.** You can configure Oracle Application Server to use only specific encryption ciphers for HTTPS connections. Connections from all old Web browsers that have not upgraded the client-side Secure Sockets Layer (SSL) library to 128-bit can be rejected. This functionality is especially useful for banks and other financial institutions because it provides server-side control of the encryption strength for each connection.
- **Use HTTPS to HTTP appliances for accelerating HTTP over SSL.** Huge performance overhead of HTTPS forces a trade-off in some situations. Use of HTTPS to HTTP appliances can change throughput from 20 to 30 transactions for each second on a 500 MHz Unix to 6000 transactions for each second for a relatively low cost, making this trade-off decision easier. This solution is better than math and crypto cards, which can be added to UNIX, Windows, and Linux computers.
- **Ensure that sequential HTTPS transfers are requested through the same Web server.** Expect 40/50 milliseconds CPU time for initiating SSL sessions on a 500 MHz computer. Most of this CPU time is spent in the key exchange logic, where the bulk encryption key is exchanged. Caching the bulk encryption key will significantly reduce CPU overhead on subsequent access, then the access is routed to the same Web server.
- Keep secure pages and pages not requiring security on separate servers. While it may be easier to place all pages for an application on one HTTPS server, the resulting performance cost is very high. Reserve your HTTPS server for pages that require SSL. Put pages that do not require SSL on an HTTP server.

  If secure pages are composed of many GIF, JPEG, or other files that would be displayed on the same screen, it is probably not worth the effort to segregate secure from non-secure static content. The SSL key exchange (a major consumer of CPU cycles) is likely to be called exactly once in any case, and the overhead of bulk encryption is not that high

4.1.2 Assign Lowest-Level Privileges Adequate for the Task to Contain Security Leaks

When assigning privileges to modules, use the lowest levels adequate to perform the modules functions. This assignment is essentially fault containment, that is, if security is compromised, it is contained within a small area of the network and cannot invade the entire intranet.

4.1.3 Cookie Security Best Practices

Use the following as guidelines for cookies:

- **Make sure that cookies have proper expiration dates.** Permanent cookies should have relatively short expiration dates of about three months or less. This configuration will avoid cluttering client Web browsers, which may cause errors if the Web browser cannot transmit all the valid cookies. Set non-permanent cookies to expire when the relevant application exits.
- **Make sure that information in cookies contains Method Authentication.** Use authentication to ensure that cookie data has not been changed since the application set the data. This authentication helps ensure that the cookie cannot be

modified and deceive the application. Also, this helps prevent application failures if the cookie is inadvertently corrupted.

- **Make sure that the size and varieties of cookies are kept low.** There is a finite number and aggregate size of cookies that Web browsers support. If this is exceeded, then the Web browsers will not send all the relevant cookies leading to application failures. Also, very large cookies can result in performance degradation.
- **Carefully use cookie domain name facilities.** Use of cookie domains should ensure that the domain is the smallest possible. Making the domain `oracle.com`, for instance, would mean that any host in `oracle.com` would get the cookie. With hundreds of applications on different parts of `oracle.com`, a domain of `oracle.com` for each of them results in attempts to send hundreds of cookies for each HTTP input operation.

4.1.4 Systems Setup Best Practices

Use the following as guidelines for system setup:

- **Apply all relevant security patches.** Check MetaLink (`http://metalink.oracle.com`) and OTN (`http://www.oracle.com/technology/index.html`) for current security alerts. Many of these patches address publicly announced security issues.
- When deploying software, change all default passwords and close accounts used for samples and examples.
- **Remove unused services from all hosts.** Examples of unused services are FTP, SNMP, NFS, BOOTP, and NEWS. HTTP or WebDAV may be good alternatives.
- **Limit the number of people with root and administrative privileges.**
- **In UNIX, disable the "r" commands if you do not need them.** For example, `rhost` and `rcp`.

4.1.5 Certificates Use Best Practices

Use the following guidelines when using certificates:

- **Ensure that certificate organization unit plus issuer fields uniquely identify the organization across the Internet.** One way to accomplish this would be to include the Dun and Bradstreet or IRS identification as identification for the issuer and the organizational unit within the certificate.
- **Ensure that certificate issuer plus distinguished name uniquely identify the user.** If the combination of issuer and distinguished name is used as identification, there is no duplication risk.
- **Include expiring certificates in tests of applications using certificates.** Expiration is an important consideration for a number of reasons. Unlike most username/password-based systems, certificates expire automatically. With longer duration certificates, fewer re-issues are required, but revocation lists become larger.

  In systems where certificates replace traditional usernames/passwords, expiring certificate situations may result in unexpected bugs. Careful consideration of the effects of expiration is required and new policies will have to be developed because most application and infrastructure developers have not worked in systems where authorization might change during transactions.

- **Use certificate re-issues to update certificate information.** Because certificates expire, infrastructure for updating expired certificates will be required. Take advantage of the re-issue to update organizational unit or other fields. In cases of mergers, acquisitions, or status changes of individual certificate holders, consider re-issuing even when the certificate has not yet expired. But pay attention to key management. If the certificate for a particular person is updated before it expires, for example, put the old certificate on the revocation list.
- **Audit certificate revocations.** Revocation audit trails can help you reconstruct the past when necessary. An important example is replay of a transaction to ensure the same results on the replay as during the original processing. If the certificate of a transaction participant was revoked between the original and the replay, failures may occur. These errors may not have occurred when the original transaction was processed. For these cases, view the audit trail to simulated authentication at the time when the transaction was initially processed.

4.1.6 Review Code and Content Against Already Known Attacks to Minimize the Attack Recurrence

It is quite common for viruses or known attacks to resurface in slightly altered shape or form. Thus, just because a threat has been apparently eliminated does not mean it will not resurface. Use the following as guidelines to minimize the recurrence of the threat:

- **Ensure that programs are reviewed against double encoding attacks.** There area many cases where special characters, such as <, >, | are encoded to prevent cross-site scripting attacks or for other reasons. For example, `<` might be substituted for >. In a double encoding, the attacker might encode the `&` so that later decoding might involve the inadvertent processing of a >, <, or | character as part of a script. Prevention of this attack, unfortunately, can only be provided by careful program review. You can use some utilities to filter escape characters that might result in double-encoding problems in later processing.
- **Ensure that programs are reviewed against buffer overflow for received data.**
- **Ensure that programs are reviewed against cross-site scripting attacks.** This attack typically tricks HTML and XML processing through input from Web browsers (or processes that act like Web browsers) to invoke scripting engines inappropriately. However, it is not limited to the Web technologies, and you should evaluate all code for this.

4.1.7 Firewall Best Practices

The following are some common recommended practices pertaining to firewalls; while not unique to Oracle Application Server, these are important to overall Oracle Application Server security:

- Place servers providing Internet services behind an exterior firewall of the stateful inspection type. Stateful inspection means that the firewall keeps track of various sessions by protocol and ensures that illegal protocol transitions are disallowed through the firewall. This configuration blocks the types of intrusion that exploit illegal protocol transitions.
- Set exterior firewall rules to allow Internet-initiated traffic only through specific IP and PORT addresses where SMTP, POP3, IMAP, or HTTP services are running. Some protocols, such as IIOP, leave ports open without receiving processes. Port and IP combinations that are not assigned to running programs should not be permitted.

- Set interior firewall rules to allow messages through to the intranet only if they originate from servers residing on the perimeter network. All incoming messages must first be processed in the perimeter network.
- Send outgoing messages through proxies on the perimeter network.
- Do not store the information of record on bastion hosts. Bastion hosts are fortified servers on the perimeter network. Segment information and processing such that the bastion hosts provide initial protocol server processing and generally do not contain information of a sensitive nature. The database of record and all sensitive processing should reside on the intranet.
- Disallow all traffic types unless specifically allowed. allow only the traffic required by Oracle Application Server for better security. For example, HTTP, AJP, OCI, LDAP.

4.1.8 Leverage Declarative Security

Oracle HTTP Server has several features that provide security to an application without requiring the application to be modified. Leverage or evaluate these features before programming similar functionality as those features into the application. Specifically:

- Authentication: Oracle HTTP Server can authenticate users and pass the authenticated user-id to an application in a standard manner. It also supports single sign-on, thus reusing existing login mechanisms.
- Authorization: Oracle HTTP Server has directives that can allow access to your application only if the end user is authenticated and authorized. Again, no code change is required.
- Encryption: Oracle HTTP Server can provide transparent SSL communication to end customers without any code change on the application.

Leverage these three features before designing any application-specific security mechanisms.

4.1.9 Use Switched Connections in DMZ

Oracle recommends that all DMZ attached devices be connected by switched, not bussed connections. Furthermore, devices such as the Cisco 11000 series devices, which can provide IP, port, and protocol rules between each pair of connected devices are preferred.

4.1.10 Place Application Server in the DMZ to Prevent Security Issues

Application servers should exist in the DMZ. In this architecture, OracleAS Web Cache only forwards requests to computers containing Web servers. Web servers only forward requests to application servers or through PL/SQL to database servers. The application servers only forward inward requests to the database or, perhaps, special message processing processors in the intranet. This configuration provides excellent fault containment because a compromised Web server must somehow compromise an application server before the database can be attacked.

4.1.11 Use Secure Sockets Layer Encryption to Secure LDAP and HTTP Traffic

You can use Secure Sockets Layer (SSL) encryption to secure both LDAP and HTTP traffic that passes between the various components of the Oracle Application Server. To ensure that all LDAP queries being sent to Oracle Internet Directory are

SSL-encrypted, you need to configure your Oracle Internet Directory instance to run with a configuration set that supports only SSL-encrypted LDAP connections. The default mode installed with Oracle Application Server allows a given Oracle Internet Directory instance to be configured to listen on both SSL and non-SSL ports.

SSL encryption is unrelated to the installation or use of HTTPS, which allows users to access Oracle Application Server components over HTTP while using SSL to encrypt Web client packets.

> **See Also:** *Oracle Internet Directory Administrator's Guide* for more details on configuring Oracle Internet Directory instances with SSL

4.1.12 Tune the SSLSessionCacheTimeout Directive to Meet Your Application Needs

The Apache server in Oracle Application Server caches a client SSL session information by default. With session caching, only the first connection to the server incurs high latency.

In a simple test to connect and disconnect to an SSL-enabled server, the elapsed time for five connections was approximately 11.4 seconds without SSL session caching as opposed to approximately 1.9 seconds when session caching was enabled.

The default `SSLSessionCacheTimeout` is 300 seconds. Note that the duration of a SSL session is unrelated to the use of HTTP persistent connections. You can change the `SSLSessionCacheTimeout` directive in `httpd.conf` file to meet your application needs.

4.1.13 Plan Out The Final Topology Before Installing Oracle Application Server Security Components

Consult the *Oracle Application Server Enterprise Deployment Guide* and the *Oracle Identity Management Infrastructure Administrator's Guide* when planning out the final target topology. Identify the steps in installing and configuring the various Oracle Application Server components consistent with the options of the Oracle Universal Installer, rather than approaching the desired topology on an adhoc basis.

4.2 Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider Best Practices

Oracle Application Server provides an implementation of OracleAS JAAS Provider for J2EE applications that is fully integrated with J2EE declarative security. This implementation allows J2EE applications to take advantage of the JAAS constructs, such as principal-based security and pluggable login modules. Optionally, the OracleAS JAAS Provider implementation allows J2EE applications running on OC4J to leverage the central security services of Oracle Identity Management.

4.3 J2EE Security Best Practices

This section describes Oracle Containers for J2EE (OC4J) security best practices. It includes the following topics:

4.3.1 Avoid Writing Custom User Managers and Instead Use Included APIs to Focus Time on Business Logic

The OC4J container continues to provide several methods and levels of extending security providers. You can extend the `UserManager` class to build a custom user manager that enables you to leverage the functionality provided by the OracleAS JAAS Provider. Both OracleAS Single Sign-On and Oracle Internet Directory provide APIs to integrate with external authentication servers and directories respectively, thus allowing developers more time to focus on actual business logic instead of infrastructure code.

4.3.2 Use the Authentication Mechanism with the JAAS Provider to Leverage Benefits

OC4J allows different authentication options for J2EE applications. Oracle recommends leveraging the OracleAS Single Sign-On server whenever possible for the following reasons:

- It is the default mechanism for most Oracle Application Server components such as OracleAS Portal, OracleAS Forms Services, OracleAS Reports Services, and OracleAS Wireless.
- It is easy to setup in a declarative fashion and does not require any custom programming.
- It provides a seamless way for PKI integration.

For environments where OracleAS Single Sign-On is not available, and custom authentication is required, one should use JAAS-compliant `LoginModules` to extend OC4J authentication. When using `LoginModules`, it is important to only use application relevant principals (roles) associated with the authenticated subject to preserve least privilege.

4.3.3 Use Fine-Grained Access Control

Unlike the coarse-grained J2EE authorization model as it exists today, the OracleAS JAAS Provider integrated with OC4J allows any protected resource to be modeled using Java permissions. The Java permission model (and associated `Permission` class) is extensible and allows a flexible way to define fine-grained access control.

For example, you can write a servlet with `Subject.doAs` or `Subject.doPrivileged` to control code that executes sensitive operations.

4.3.4 Use Oracle Internet Directory as the Central Repository to Provide LDAP Standard Features

Although the OracleAS JAAS Provider supports a flat-file XML-based repository useful for development and testing environments, configure it to use Oracle Internet Directory for production environments. Oracle Internet Directory provides LDAP standard features for modeling administrative metadata and is built on the Oracle

database platform inheriting all of the database properties of scalability, reliability, manageability, and performance. To optimize performance, adjust the caching configurations appropriate for your environment.

4.3.5 Develop Appropriate Logout Functionality to Prevent Users from Closing the Web Browsers

Simple J2EE applications using HTTP Basic authentication do not support the concept of logout, relying instead on the user to close the Web browser. When using other forms of authentication, including OracleAS Single Sign-On, it is important to plan out various logout and timeout flows. OC4J has an adjustable HTTP session inactivity parameter that is set to 20 minutes by default. If J2EE applications are leveraging OracleAS Single Sign-On and want to support full logout functionality, write them with the appropriate logout dynamic directives.

> **See Also:** *Oracle Application Server Single Sign-On Administrator's Guide*.

4.3.6 Secure the OC4J Environment

This section provides best practices for securing the OC4J environment. It contains the these topics:

- Section 4.3.6.1, "Restrict Access to the OC4J Standalone Administration Application Server Control"
- Section 4.3.6.2, "Remove Unneeded Features"
- Section 4.3.6.3, "Disable Debug Mode"
- Section 4.3.6.4, "Disable Default Invoker"
- Section 4.3.6.5, "Disable Directory Browsing"
- Section 4.3.6.6, "Disable File Aliases"
- Section 4.3.6.7, "Change Your Account Passwords"
- Section 4.3.6.8, "Use Password Indirection"
- Section 4.3.6.9, "Restrict Access to Network Service Ports"

4.3.6.1 Restrict Access to the OC4J Standalone Administration Application Server Control

The OC4J administration application is installed by default in every OC4J instance. Access to this application should be restricted to the local machine, that is, connections coming from 127.0.0.1 only, or to a set of trusted hosts. This is achieved by using OC4J's built in IP access controls.

Implementation Details

1. Open the `orion-web.xml` file located in the `application-deployments` directory, usually located at:

   ```
   $ORACLE_
   HOME/j2ee/home/application-deployments/ascontrol/ascontrol/orion-web.xml
   ```

2. Add the following element to `orion-web-app`:

   ```
   <access-mask default="deny">
        <ip-access ip="127.0.0.1" mode="allow"/>
   ```

```
</access-mask>
```

4.3.6.2 Remove Unneeded Features

OC4J's default configuration provides many features that enable ease of use and development. Many features are enabled by default, and the product is configured to provide informative error messages. This is not really a problem when OC4J is used in a development environment. However, this configuration may not meet your needs in a production environment.

4.3.6.3 Disable Debug Mode

Disabling debug mode will stop OC4J from echoing stack traces back to end user through HTTP. However, stack traces will still be available in log files to assist with debugging of applications.

Implementation Details

You can turn debug mode off by setting Java system properties when running OC4J, or in `opmn.xml` with:

```
java -DDebug=false -jar oc4j.jar
```

4.3.6.4 Disable Default Invoker

The Default Invoker is a legacy convention that allows a given servlet to be invoked directly by class name, rather than by a specific mapping in a configuration file. For example, assume there is a servlet named `servlet com.foo.bar.FooBarServlet`. This servlet is mapped to `/FooBar` by a configuration file and there is access control on it. Only authenticated users are allowed to invoke it because it performs some sensitive function. If OC4J has the Default Invoker functionality enabled, then this restriction can be bypassed as an attacker can invoke the servlet directly, for example, by requesting `/servlet/com.foo.bar.FooBarServlet`. As a result, you must remove this functionality.

Implementation Details

To implement this best practice:

- Edit the `<orion-web-app>` element in the `global-web-application.xml` or `orion-web.xml` file with the following attribute:

  ```
  servlet-webdir-""
  ```

- Add `-Dhttp.webdir.enable=false` as a startup parameter in the `opmn.xml` file.

4.3.6.5 Disable Directory Browsing

Disable directory browsing to avoid exposing how an application works to malicious users.

Implementation Details

Ensure the `<orion-web-app>` element has the following attribute in the `orion-web.xml` file:

```
directory-browsing="deny"
```

This setting is the default setting.

4.3.6.6 Disable File Aliases

Certain attacks against J2EE containers and Web servers rely on tricking the container into returning an unprocessed resource. Returning the source code to a `.jsp` rather than compiling and executing it server side is one example. To prevent such attacks, disable file aliases.

Implementation Details

Define the following Java startup parameter:

```
-Dhhtp.file.allowAlias=false
```

4.3.6.7 Change Your Account Passwords

OC4J Standalone forces a password to be set at installation time for the OC4J administrator account, `oc4jadmin`. However, when you install other versions of OC4J, particularly those consumed by other parts of the Oracle product stack, well-known passwords are automatically set. Be sure to change all of your passwords to ones that cannot be easily guessed or broken.

By default, `system-jazn-data.xml` is used by OC4J. The `oc4jadmin` account is created in this file, but de-activated out of the box. You have to activate this account by providing a new password either during install time or during first time start-up of OC4J.

4.3.6.8 Use Password Indirection

Use password indirection to avoid the presence of clear text passwords in any of the OC4J XML configuration files. With password indirection the stored password is obfuscated, and can be retrieved when needed from the OC4J security sub-system. This mechanism is only available when the security provider of your application is set to be the XML file-based or Oracle Identity Management providers.

Implementation Details

Add `->UserName` in the XML configuration to represent the password.

4.3.6.9 Restrict Access to Network Service Ports

Remote Method Invocation (RMI) by default listens on port 23791. You can restrict it with the following methods:

- Add the `host` attribute to the `rmi-server` element in `rmi.xml`.

  ```
  <rmi-server port="23791" host="127.0.0.1">
  ```

- Use the access region set element

  ```
  <access-mask default="deny">
       <ip-access ip="127.0.0.1" mode="allow"/>
  </access-mask>
  ```

The Java Messaging Service (JMS) listens by default on port 9127. You can restrict access to specific hosts by adding the `host` attribute to the `jms-server` element.

```
<jms-server port="9127" host="127.0.0.1">
```

4.4 OracleAS Single Sign-On Best Practices

This section describes Oracle Application Server Single Sign-On (OracleAS Single Sign-On) best practices. It features the following topics:

4.4.1 Configure for High Availability to Prevent Inaccessible Applications

Single sign-on failure is catastrophic, since it means no single sign-on protected application can be accessed. Two recommendations for high availability of OracleAS Single Sign-On are:

- Carefully consider inclusion of any other types of processing on the single sign-on servers since this can make instability more likely.
- Consider deploying multiple single sign-on servers fronted by load balancing hardware to protect against failures in single sign-on listeners. In this case, the address of the load balancer is used as the single sign-on address and the single sign-on listener configuration information is replicated. Oracle also recommends that the database be a Oracle Real Application Cluster configured for additional improvements in availability.

See Also:

- *Oracle Application Server Single Sign-On Administrator's Guide*
- Whitepaper *Expose your Intranet Portal to the Outside World in a Secured Manner* available from the Oracle Technology Network at `http://www.oracle.com/technology/index.html` for configuring multiple single sign-on servers

4.4.2 Leverage OracleAS Single Sign-On to Optimize Administration and Customer Experience

Use OracleAS Single Sign-On as the primary point of security. This component provides benefits from an administrative point of view and is a major convenience to application customers. Also, OracleAS Single Sign-On is well integrated with the rest of Oracle Application Server Infrastructure and can, with Oracle Internet Directory and other means, be integrated with non-Oracle application and infrastructure. Also, as single sign-on becomes a single point for authentication, opportunities to attack the multiple authentication entities of sites today are reduced.

Using an OracleAS Single Sign-On single-authenticated user for all applications provides better control for more uniform authorization.

4.4.3 Use an Enterprise-Wide Directory to Eliminate User Data in Multiple Systems

In order to deploy an effective single sign-on solution, the user population must be centralized in a directory, preferably an LDAP-based directory, such as Oracle Internet Directory. Having users represented in multiple systems, such as in multiple Microsoft Windows domains, makes setting up the infrastructure for a common identity more difficult. In addition, clearly defining and automating the user provisioning process makes managing the single sign-on environment much easier.

4.4.4 Use OracleAS Single Sign-On to Validate User Credentials

OracleAS Single Sign-On provides the infrastructure to validate credentials and allows for various different authentication mechanisms, such as username, password, and X.509 certificates. Moreover, since these mechanisms can be shared across different applications and Web sites, end users do not have to create a new username, password for each different corporate application.

4.4.5 Always Use SSL with Oracle Application Server to Protect Applications

The OracleAS Single Sign-On server simplifies user interaction by providing a mechanism to have a single username and password, which can be used by multiple partner applications. With this ease of use, comes the caution that the single sign-on server should always be accessed in the correct fashion. A breach of the common password can now put all partner applications at risk. Therefore, always configure the single sign-on server to allow connections in SSL mode only. This configuration protects the end user's credentials going across the wire. Applications where security and data confidentiality are important should also be protected by SSL. From a performance perspective, use of SSL hardware accelerators is recommended.

4.4.6 Provide Username and Password Only on Login Screen to Prevent Users from Providing Credentials to Inappropriate Servers

The OracleAS Single Sign-On server provides a standard login screen. This login page is serviced from the single sign-on server, which typically is installed on a different computer from the one the end user is trying to access. Thus, it is critical that before the end user enters their login and password, that a valid single sign-on screen is observed. This screen prevents users from unknowingly providing their username or password to inappropriate servers.

4.4.7 Log Out to Prevent Active Cookies

Most users do not log out of Internet applications and this issue creates problems at two levels:

1. A security risk. Another person accessing the work station can now reuse the cookie. Also, since the session remains valid until it times out, a hacker from another computer has a longer time window to guess the session ID or cookie value.
2. The system resources on the server associated with the cookie are not released until the session is ended or invalidated.

For application developers and administrators, configure single sign-on session duration and inactivity timeouts appropriately. For example, configure one-hour inactivity timeouts for sensitive applications.

For external applications, OracleAS Single Sign-On cannot log out users. Therefore, closing all Web browser windows is important.

5

Oracle Virtual Directory

This chapter describes best practices for Oracle Virtual Directory. It includes the following topics:

5.1 Give Each Adapter Its Own Namespace to Simplify Configuration

With Oracle Virtual Directory the directory namespace is very flexible and can be completely virtualized. It is possible for multiple adapters to have the exact same base Relative Distinguished Name (RDN), such as `ou=employees,dc=mycompany,dc=com`. However, it is easier to configure Oracle Virtual Directory with less need for customization if you give each adapter its own namespace.

Implementation Details

To implement this best practice, you simply give each adapter a unique branch name.

To give each adapter a unique branch name:

1. Create a new adapter.
2. In the **Mapped Namespace** field, make sure the value is unique.

   Adapters can share the same base Distinguished Name (DN) but should have their own branch RDN.

> **See Also:** Chapter 2, "Planning the Virtual Directory," and Chapter 4, "Oracle Virtual Directory," in the *Oracle Virtual Directory Product Manual* available from the Oracle Technology Network at `http://www.oracle.com/pls/wocprod/docs/page/ocom/technology/products/id_mgmt/ovds/pdf/b28833.pdf`

5.2 Use Routing Priority to Control How Order Entries Are Returned for Better Performance

When you perform a search, multiple adapters are searched. You can control the search order in which adapters by prioritizing routing. This feature enables you to improve search performance by having Oracle Virtual Directory search the adapters that are faster first. It also enables control over which entries are the master entries when using the UniqueEntry plug-in.

Implementation Details

1. Go to the adapter's **Routing** tab.
2. Position the priority selector to its proper priority.

   The lower the number, the higher the priority.
3. Repeat for each adapter.

   If multiple adapters have the same priority, they will be searched in the order they were added to Oracle Virtual Directory.

> **See Also:** Chapter 2, "Planning the Virtual Directory," in the *Oracle Virtual Directory Product Manual* available from the Oracle Technology Network at
> http://www.oracle.com/pls/wocprod/docs/page/ocom/technology/products/id_mgmt/ovds/pdf/b28833.pdf

5.3 Use Attribute Flow to Improve Security, Performance and Flexibility

You can use an adapter's attribute flow to provide better control over which attributes can be retrieved or stored in a particular adapter. This can provide you with additional security by restricting control to attributes, even if someone has LDAP administrator privileges to the Oracle Virtual Directory server. It can also improve performance because if a search operation is trying to retrieve attributes that Oracle Virtual Directory knows cannot even be returned from an adapter, it will not waste time searching that adapter. Finally it gives you more flexibility, such as being able to do schema extensions at the Oracle Virtual Directory layer by leveraging a database instead of needing to extend your enterprise directory schema.

Implementation Details

1. Go to **Adapter Router** tab.
2. Go **Attribute Flow** section
3. Enter comma-delimited list of attributes in proper fields.

   If you list attributes in any of these fields, only those attributes will be allowed or restricted.

> **See Also:** Chapter 4, "Oracle Virtual Directory Manager" in the *Oracle Virtual Directory Product Manual* available from the Oracle Technology Network at
> http://www.oracle.com/pls/wocprod/docs/page/ocom/technology/products/id_mgmt/ovds/pdf/b28833.pdf

5.4 Use Mapping Scripts to Unify Schema

Oracle Virtual Directory can connect to heterogeneous types of LDAP directories and they can have different types of schema. In particular Microsoft Active Directory has its own proprietary user schema different than any other LDAP server. LDAP client applications will not function properly if the LDAP server comes back with entries of different types of schema. You can use mapping scripts, such as the provided Active `DirectorytoInterorg` mapping script, to make all directory servers appear to have the same schema to LDAP clients connecting to Oracle Virtual Directory.

Implementation Details

1. In Oracle Virtual Directory Manager, expand the **Engine** tree.
2. Right-click **Mapping**, and select **New** > **Mapping**.
3. Choose the proper mapping.
4. Click **Finish**.
5. Edit the mapping configuration properties.
6. Right-click the map file, and choose **Deploy to Server**.
7. Select the adapter you want to apply the mapping to and add the mapping.

> **See Also:** Chapter 8, "Mapping System" in the *Oracle Virtual Directory Product Manual* available from the Oracle Technology Network at `http://www.oracle.com/pls/wocprod/docs/page/ocom/technology/products/id_mgmt/ovds/pdf/b28833.pdf`

5.5 Add Microsoft Schema if Using ActiveX Data Objects to Query Oracle Virtual Directory

If you are using Microsoft .NET APIs—Visual Basic (VB) and Visual Basic Scripting (VBScript)—or ActiveX Data Objects (ADO) to query Oracle Virtual Directory, add the Microsoft schema to Oracle Virtual Directory for this to function properly. The schema is included in 10.1.4.

Implementation Details

1. In Oracle Virtual Directory Manager, go to the **Engine** > **Server** > **Settings** tab.
2. In the **Schema** > **Files** field, replace the existing contents with `conf/schema.ms.xml,conf/schema.user.xml`.

6

Oracle Application Server High Availability Solutions

This chapter describes best practices for various highly available configurations and features for OracleAS High Availability Solutions. It contains the following topics:

6.1 Oracle Application Server Cluster (Identity Management)

Use a consistent and standardized configuration for the instances that participate in an OracleAS Cluster (Identity Management). From the OracleAS Single Sign-On perspective, all the instances in the cluster are seen as a single entity that provide a single service. To avoid errors in updates to the configuration, follow these tips:

- Use the same Oracle home path for the Identity Management instances in your OracleAS Cluster (Identity Management).
- Use the same instance name for the Identity Management instances. The host name prefix the the installer adds ensures that the application server instance is unique.
- While Oracle allows the HTTP listen server to listen on different ports in each Oracle Application Server Single Sign-On/Oracle Delegated Administration Services host, Oracle recommends to use the same ports.
- It is better to have same ports used for all other components and services, including Application Server Control. Use `staticports.ini` for all installs to achieve this.
- The cluster name you specify in the OracleAS Cluster configuration page in the during OracleAS Cluster (Identity Management) installation is case sensitive. During installation, it is a good practice to always use all upper case or lower case to avoid configuration errors.

See Also:

- Chapter 3, "Active-Active Topologies," in the *Oracle Application Server High Availability Guide*
- Chapter 9, "Installing in High Availability Environments: OracleAS Cluster (Identity Management)," in the *Oracle Application Server Installation Guide*

6.2 Load Balancers

This section covers the following topics:

6.2.1 Use Fault-Tolerant Hardware Load Balancers to Avoid Single Points of Failure

The load balancer is the entry point to many different services. Independent of providing redundancy for these services, the load balancer itself must be highly available. Use load balancers that can be configured for automatic failover in case of load-balancer failures.

6.2.2 Use Monitoring of Services to Automatically Disable Traffic to Unavailable Nodes

It is common in most load balancer products not to detect the failure of a service through one of its monitors. For these products, connections will remain idle and requests will hang.

Implementation Details

Make sure that the load balancer has been configured to monitor the services and fails over all the traffic to other nodes immediately in case of a failure.

6.2.3 Configure All Idle Time Timeouts to Maximize Time for Unused or Idle Service

Set the idle timeouts to the maximum time you expect a service to be unused or idle. Otherwise, the load balancer will cut the connections even when they appear as available to the invocation clients

See Also: your load balancer product's documentation

6.3 Oracle Application Server Cold Failover Clusters

This section covers the following topics:

- Section 6.3.4, "Standardize Port Allocation and Pre-Allocate Ports to the OracleAS Cold Failover Cluster Instances to Avoid Failures"

6.3.1 Use Shared Oracle Home Installs for OracleAS Cold Failover Cluster (Middle-Tier) to Simplify Administration

Installing OracleAS Cold Failover Cluster in a multiple Oracle home configuration will require every administration change to be applied multiple times, including the deployment of J2EE applications. Oracle recommends using a shared drive for all install types, including the ones where non-shared is possible.

6.3.2 Use Oracle Universal Installer Commands to Attach OracleAS Cold Failover Cluster Oracle Home with the oraInventory

An OracleAS Cold Failover Cluster installation updates the `oraInventory` directory in a local file system unless the installer is specifically pointed to an `oraInventory` directory in a shared location. If you install additional software from the node in the hardware cluster that was not used for the OracleAS Cold Failover Cluster installation, the cold failover cluster installation will not be detected. Use Oracle Universal Installer to attach your OracleAS Cold Failover Cluster Oracle home with the `oraInventory` directory on the non-install nodes.

Implementation Details

In order to associate and attach the Oracle home to the `oraInventory` directory, use the following command:

```
./runInstaller -silent -attachHome -invPtrLoc <oraInst.loc location> ORACLE_
HOME="ORACLE_HOME_LOCATION" ORACLE_HOME_NAME="ORACLE_HOME_NAME" CLUSTER_NODES="{}"
LOCAL_NODE="node_name"
```

> **See Also:** Chapter 8, "Installing in High Availability Environments: OracleAS Cold Failover Cluster," in the *Oracle Application Server Installation Guide*

6.3.3 Use Disk Redundancy for OracleAS Cold Failover Cluster to Avoid Oracle Home Failures

Resize the disk to hold all application deployments, maximum JMS messages persisted, and hold OracleAS Portal and Oracle Identity Management data when applicable. It is critical to use some kind of disk redundancy to secure all the binaries, data and metadata used by an OracleAS Cold Failover Cluster. If you are using Automatic Storage Management (ASM) and co-existing with other databases that use ASM, upgrade to clustered ASM for the entire cold failover cluster

6.3.4 Standardize Port Allocation and Pre-Allocate Ports to the OracleAS Cold Failover Cluster Instances to Avoid Failures

If the ports are not available when the active instances fails over to the passive node, Oracle Application Server will not be able to start. Keep record of the ports used by your active-passive installations and use `staticports.ini` to install any other Oracle Application Server instances so that those ports remain free.

If you plan to implement the passive node in a machine that is already running other applications, then try to configure the ports for the OracleAS Cold Failover Cluster so that they do not conflict with existing components.

When you perform any new installation or application deployments in the passive node, ensure that the new components do not have port conflict with the passive instance. Often the ports allocated to passive components will appear as free to the operating system.

> **See Also:** Chapter 8, "Installing in High Availability Environments: OracleAS Cold Failover Cluster," in the *Oracle Application Server Installation Guide*

6.4 Oracle Application Server Guard

This section contains these topics:

6.4.1 Clean Up Invalid Records to Avoid Instantiation and Synchronization Errors

Remove all invalid Oracle software (invalid Oracle Databases or Oracle Application Server installation) from the `oraInventory` directory. You can accomplish this task by using the Oracle Universal Installer. For example, an instance is installed and later removed manually by deleting an Oracle home. Manually remove this instance's information from the `Inventory.xml` file; otherwise, OracleAS Guardmay fail to perform the instantiation and synchronization operations.

6.4.2 Use the Same Ports for OracleAS Guard to Avoid Manual Configuration Steps in Synchronization Operations

Try to use the same OracleAS Guard ports in the primary and standby sites. Otherwise, you will have to manually edit the `dsa.conf` file to enable the communication between production and standby site peers.

6.4.3 Use Different Labels and Colors in OracleAS Guard Shells to Avoid Errors

Due to the host name symmetry requirements for OracleAS Guard, shells in both primary and standby environments may appear under the same host name and prompt. It is a very common mistake to perform operations in a standby shell that were intended to be preferred in the production site, causing errors and inconsistent configuration changes. Use a fixed and standard way to label and position command shells to make sure that operations are issued in the correct site.

6.4.4 Enable High-Logging Level to Troubleshoot OracleAS Guard Operations

You can trace OracleAS Guard operations to a great detail by using the `set trace on all` command. By default, the trace level is set to `off`. Use this type of logging when trying to solve issues with OracleAS Guard.

Implementation Details

From the OracleAS Guard prompt, enter:

```
set trace on | off <traceflags>
```

> **See Also:** Chapter 12, "OracleAS Guard asgctl Command-line Reference," in the *Oracle Application Server High Availability Guide*

6.5 Backup and Recovery

This section contains these topics:

- Section 6.5.1, "Whenever an Operation is Exposed through Application Server Control, Use It as the Standard Way to Perform Backup and Recovery to Avoid the Common Errors and Typos in Command-Line Operations"
- Section 6.5.2, "Use Instance-Level Backup to Guarantee Consistency"
- Section 6.5.3, "Perform an Image Backup to Recover from Loss of Host Scenario"
- Section 6.5.4, "Use Incremental Backups to Save Time and Disk Space"

6.5.1 Whenever an Operation is Exposed through Application Server Control, Use It as the Standard Way to Perform Backup and Recovery to Avoid the Common Errors and Typos in Command-Line Operations

Application Server Control provides an easy and convenient way to configure Oracle Application Server Backup and Recovery Tool and perform many backup and recovery operations. You can view the OracleAS Backup and Recovery Tool operational log files through Application Server Control. This interface is less error prone than the command line.

Implementation Details

> **See Also:** Section 19.2.4, "Performing an Instance Backup of Oracle Application Server Using Application Server Control Console," in the *Oracle Application Server Administrator's Guide*

6.5.2 Use Instance-Level Backup to Guarantee Consistency

The instance-level backup includes the configuration and repository backup. The repository can be database-based or file-based. This functionality offers consistency between configuration and repository backups. You should try to use this option as much as possible as opposed to backing up configuration and repository separately.

Implementation Details

> **See Also:** Chapter 19, "Backup Strategy and Procedures," in the *Oracle Application Server Administrator's Guide* for information about the available instance-level backup options

6.5.3 Perform an Image Backup to Recover from Loss of Host Scenario

The OracleAS Backup and Recovery Tool offers recovery from loss of host scenarios. After doing an install, make sure to take an image backup that includes the Oracle home, `OraInventory` directory, Registry entries, instance backup, and so on. You can use this information to recover from a loss of host.

Implementation Details

In order to create an image backup, use the following commands

On UNIX:

```
bkp_restore.sh -m node_backup -o image_backup -P <archive path>
```

On Windows:

```
bkp_restore.bat -m node_backup -o image_backup -P <archive path>
```

> **See Also:** Section 18.6, "OracleAS Backup and Recovery Tool Usage Summary," and Chapter 19, "Backup Strategy and Procedures," in the *Oracle Application Server Administrator's Guide*

6.5.4 Use Incremental Backups to Save Time and Disk Space

The OracleAS Backup and Recovery Tool offers incremental backup both for configuration files and database, by which only the modified data gets backed up. You should choose this option if you do not want to backup redundant data every time and you like the backups to be small in size.

> **See Also:** Section 18.6, "OracleAS Backup and Recovery Tool Usage Summary," and Chapter 19, "Backup Strategy and Procedures," in the *Oracle Application Server Administrator's Guide*

Index

N

O

P

R

S

T

V

W

X