



ORACLE[®]

Oracle HTTP Server 11gR1



Agenda

- Overview of Oracle HTTP Server
 - Intro to OHS and differences with Apache
 - Support policy
- OHS Bundling and Installation
 - OHS Bundling
 - Install Concepts and Options
- Management of OHS
 - Management Tools for OHS
 - Common OHS management tasks
- Configuring OHS
 - OHS Configuration Files
 - Configuring Ports and Logs
- WebLogic Server Plug-In
 - Overview of mod_weblogic
 - Difference between mod_wl_ohs and mod_weblogic
 - Configuring mod_wl_ohs



Introduction to Oracle HTTP Server

- Oracle HTTP Server (OHS)
 - Is the web server component for FMW
 - Is based on Apache
 - Uses latest release available prior to ship
 - Apache v2.2 in OHS 11gR1
 - Serves static and dynamic content
 - Supports content generation in many languages, such as Java, C, C++, PHP, PERL, or PL/SQL
 - Is integrated with Oracle WebLogic Server
 - Using mod_wl_ohs
 - Can be a proxy server, both forward and reverse
 - Also integrated with Oracle Web Cache



Introduction to Oracle HTTP Server Cont.

- **SSL Support**
 - Comes with Oracle's implementation of mod_ssl (mod_ossll)
 - Based on Oracle's standard NZ library
- **Authentication/Authorization Support**
 - Supports standard authentication as well as single sign-on
 - Using the Oracle Single Sign-On infrastructure
 - Comes pre-bundled with mod_osso
- **Supports IPv6**
 - Internet access can be IPv6 or IPv4
 - Can proxy IPv6 to IPv4



Oracle HTTP Server Modules

- Consist of two components to handle client requests:
 - HTTP listener
 - To handle incoming requests and route them to the appropriate processing utility
 - Modules (mods)
 - Implement and extend the basic functionality of the listener
 - Many of the standard Apache modules are included
 - Several FMW specific mods are also included
 - To support integration between OHS and other FMW components



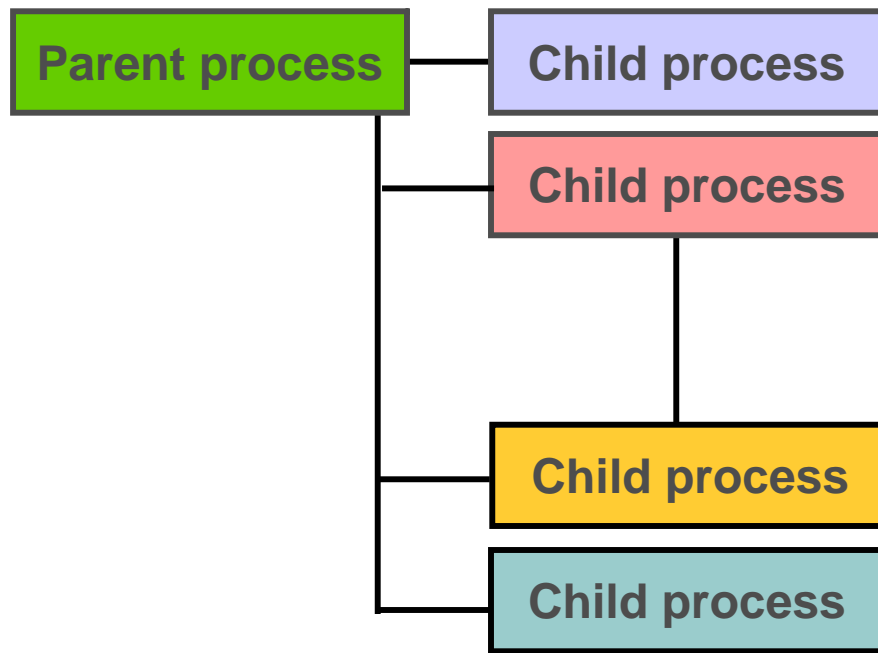
Oracle HTTP Server Modules Cont.

Base Apache and 3 rd Party Mods				Oracle-developed Mods
mod_access	mod_actions	mod_alias	mod_asis	mod_certheaders
mod_auth	mod_auth_anon	mod_auth_dbm	mod_autoindex	mod_dms
mod_cgi	mod_cern_meta	mod_define	mod_digest	mod_onsint
mod_dir	mod_env	mod_example	mod_expires	mod_oradav
mod_fastcgi	mod_headers	mod_imap	mod_include	mod_ossI
mod_info	mod_log_agent	mod_log_config	mod_log_referer	mod_osso
mod_mime	mod_mime_magic	mod_mime_static	mod_negotiation	mod_plsql
mod_perl	mod_proxy	mod_rewrite		mod_wl_ohs

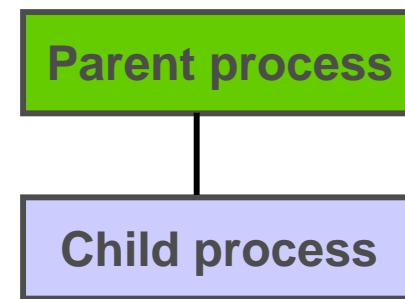


OHS Processing Model

- Different model for Unix and Windows platforms



Unix platform



Windows platform



Difference Between OHS and Apache

- Extends the standard Apache distribution
- Provides all base Apache functionality plus:
 - Better process management
 - Via integration with OPMN (Oracle Process Manager and Notification Server)
 - Provides death detection and auto-restart
 - Server and child processes treated as one unit
 - Auto termination of parent due to abnormal shutdown of children
 - Better performance analysis
 - Via integration with DMS (Dynamic Monitoring Service)
 - Provides in depth performance metrics



Difference Between OHS and Apache Cont.

- Better logging mechanism
 - Via integration with ODL (Oracle Diagnostic Logging)
 - Ability to trace log messages through all of FMW components
 - Enhanced rotation binary to support retention period
- Better management tools
 - GUI based management and monitoring tool (FMW Control)
 - CLI based provisioning and management tool (opmnctl)
- Enhanced security offering
 - Integration with Oracle SSO Server and Oracle Access Manager



OHS Support Policies

- Support for Oracle provided modules
 - All modules included in the Oracle distribution are supported by Oracle
 - Base Apache, 3rd party and Oracle-developed mods
- Support for non-Oracle provided modules
 - Customers can integrate non-Oracle provided mods with OHS
 - Just like it is done with generic Apache
 - No Oracle support for modules obtained from any other source
 - Including mods from the Apache Software Foundation
 - OHS with non-Oracle provided mods included will still be supported
 - Customers may be requested to reproduce the problem without the non-Oracle provided mod being included

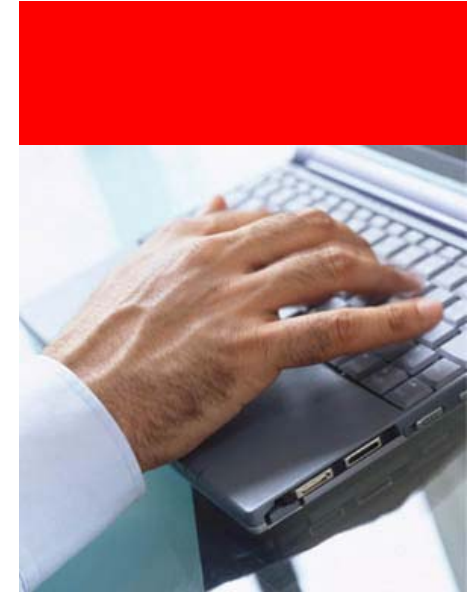


OHS Support Policies Cont.

- Bug fix policy for security vulnerabilities
 - Patch-sets and CPUs include latest Apache security fixes
 - Cannot apply generic Apache security patches to OHS
 - Oracle tests and appropriately modifies security patches before releasing to OHS users
 - Apache alerts may not be applicable
 - Because OHS might not have those components
 - For example, openSSL alerts
- Apache version upgrade policy
 - Customers cannot upgrade just Apache inside OHS
 - Newer version of Apache may be provided via patch-sets or future releases of OHS

Agenda

- Overview of Oracle HTTP Server
 - Intro to OHS and differences with Apache
 - Support policy
- OHS Bundling and Installation
 - OHS Bundling
 - Install Concepts and Options
- Management of OHS
 - Management Tools for OHS
 - Common OHS management tasks
- Configuring OHS
 - OHS Configuration Files
 - Configuring Ports and Logs
- WebLogic Server Plug-In
 - Overview of mod_weblogic
 - Difference between mod_wl_ohs and mod_weblogic
 - Configuring mod_wl_ohs





Oracle HTTP Server Bundling

- OHS 11gR1 available from two FMW packages
- Oracle Fusion Middleware Web Tier and Utilities DVD
 - Primary source to obtain OHS
 - Can be installed and configured independent to any component
 - For management via EM, choice of associating with any WLS domain
 - Comes with mod_plsql and mod_osso disabled (can be enabled)
- Oracle Portal, Forms, Reports and Discoverer DVD
 - Some of these components have dependency on OHS
 - Installed and configured automatically
 - For management via EM, auto associated with WLS domain configured for this package
 - Can not install just OHS but can create new OHS components
 - Comes with mod_plsql and mod_osso enabled and configured



Install Concepts and Options

- Separation of Config from Binaries
 - Unlike prior releases where binaries and config stayed together
- Two key terms:
 - Oracle Home (read only files)
 - Binaries and other files created at install time and never changes
 - Location represented by ORACLE_HOME env. variable
 - Oracle Instance (updateable files)
 - A collection of system components managed by OPMN
 - Contains configuration files, log files, static contents, temp files
 - Location represented by ORACLE_INSTANCE env. variable
- Can create multiple Oracle Instances per Oracle Home



Install Concepts and Options Cont.

- Installation and configuration can be done separately
 - Install Only (DVD/Disk1/runInstaller)
 - Installs the product binaries in an Oracle Home
 - All component binaries from the package are installed
 - Install and Configure (DVD/Disk1/runInstaller)
 - Installs the product binaries in an Oracle Home
 - All component binaries from the package are installed
 - Creates and configures an instance in an Oracle Instance
 - Choose what components get configured
 - Configure Only (\$OH/bin/config.sh)
 - Creates and configures an instance in an Oracle Instance
 - Choose what components get configured

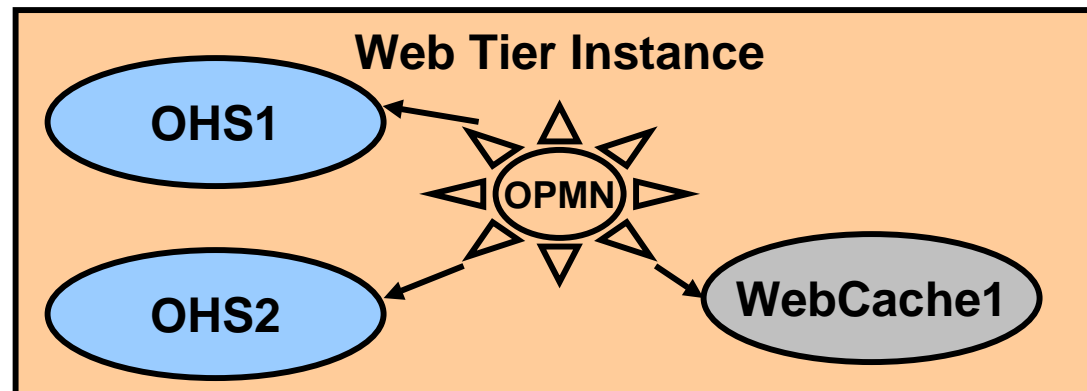


Install Concepts and Options Cont.

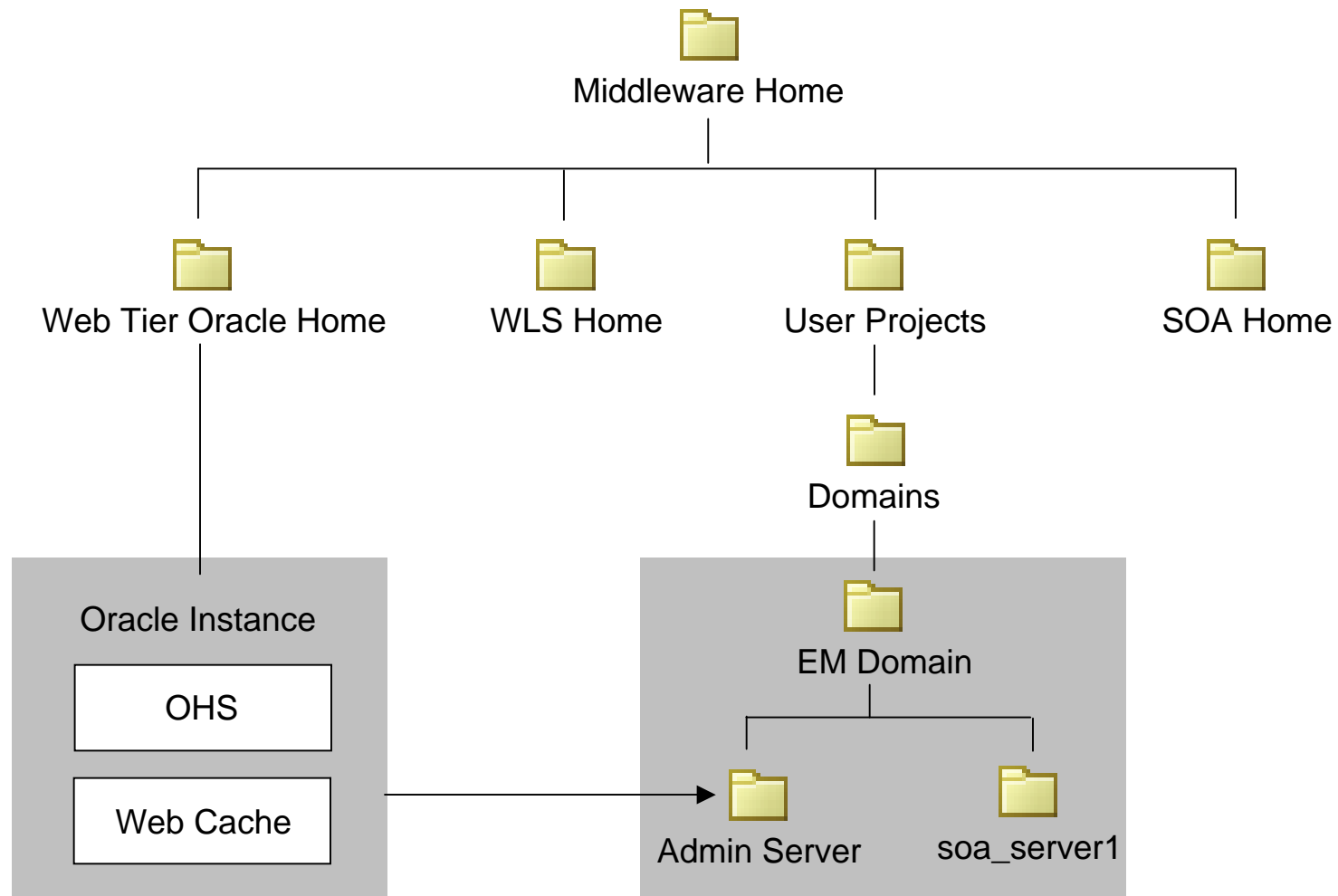
- Associating Oracle Instance with WLS domain
 - Two options to configure a Web Tier Oracle Instance
 - Associated with WLS domain
 - Allows system components to be managed via EM
 - A (JRF enabled) WLS domain needs to be pre-configured
 - Deploys system components' MBeans in Admin Server
 - Also loosely referred as “EM managed” mode
 - Not associated with WLS domain
 - Can not manage system components via EM
 - Components can still be managed via CLI (opmnctl)
 - Also loosely referred as “non-EM managed” mode

Instance of Web Tier and OHS

- A Web Tier Oracle Instance contains
 - One OPMN
 - 0 or more system component (such as OHS and Web Cache)
- OPMN manages all system components of an Oracle Instance



Middleware Home Directory Structure



* This is one of the many possible combinations for MW home directory structure



Oracle Home Directory Structure for OHS

Directory	Contents
ohs/bin	<ul style="list-style-type: none">• OHS binary files• For example, apachectl, rotatelog, httpd, apxs etc.
ohs/conf	<ul style="list-style-type: none">• OHS template configuration files• For example, httpd.conf, ssl.conf, mod_wl_ohs.conf etc.• Get provisioned to an Oracle instance at OHS creation• These files should only be edited by advanced users
ohs/htdocs	<ul style="list-style-type: none">• OHS template static files• For example, welcome pages, error docs etc.• Get provisioned to an Oracle instance at OHS creation• These files should only be edited by advanced users
ohs/modules	<ul style="list-style-type: none">• Modules .so files• For example, mod_wl_ohs.so, mod_osso.so etc.

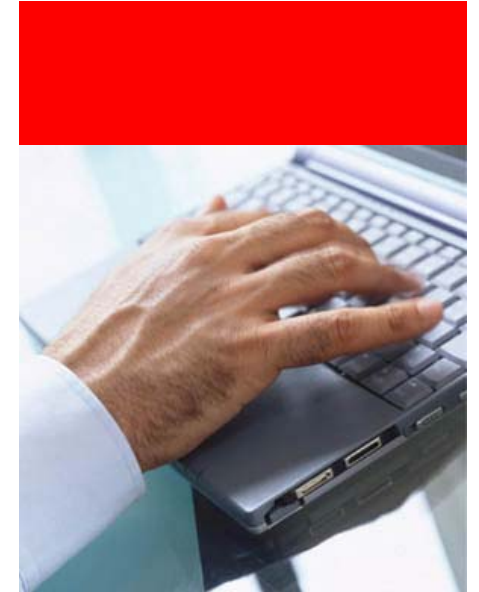


Oracle Instance Directory Structure for OHS

Directory	Contents
bin	<ul style="list-style-type: none">• opmnctl executable for the instance
config/OHS/<ohs_name>	<ul style="list-style-type: none">• OHS configuration files• For example, httpd.conf, mod_wl_ohs.conf, ssl.conf etc.
config/OHS/<ohs_name>/htdocs	<ul style="list-style-type: none">• Static content and CGI scripts• For example, welcome pages, error docs etc.
config/OHS/<ohs_name>/moduleconf	<ul style="list-style-type: none">• Configuration files that get automatically included• Don't put any files with a .conf extension here that shouldn't be included
diagnostics/logs/OHS/<ohs_name>	<ul style="list-style-type: none">• Log files• For example, error log, access log and console log

Agenda

- Overview of Oracle HTTP Server
 - Intro to OHS and difference with Apache
 - Support policy
- OHS Bundling and Installation
 - OHS Bundling
 - Install Concepts and Options
- Management of OHS
 - Management Tools for OHS
 - Common OHS management operations
- Configuring OHS
 - OHS Configuration Files
 - Configuring Ports and Logs
- WebLogic Server Plug-In
 - Overview of mod_weblogic
 - Difference between mod_wl_ohs and mod_weblogic
 - Configuring mod_wl_ohs





Management Tools for OHS

- Two primary management tools
 - Fusion Middleware Control (EM), a GUI tool
 - Main tool for process management, configuration and monitoring of OHS
 - OHS manageable when instance is registered with WLS domain
 - Communicates with OHS via Proxy MBean port
 - opmnctl, a command-line tool
 - Provides process management and provisioning operations
 - Located at two places
 - \$ORACLE_HOME/opmn/bin and \$ORACLE_INSTANCE/bin
 - Recommended to be used from the \$OI/bin directory
 - Same ORACLE_INSTANCE that OHS is running in

OHS in Fusion Middleware Control

The screenshot displays the Oracle Enterprise Manager 11g Fusion Middleware Control interface. The left-hand navigation pane shows a tree view of the 'Farm_webtier_domain' structure, with 'ohs1' and 'webcache1' highlighted by a red circle. The main content area is divided into several sections:

- Deployments:** Shows a green 3D gauge at 100% and a status indicator 'Up (5)'. Below it is a table with columns 'Name', 'Status', and 'Target'.

Name	Status	Target
Application Deployments		
Internal Applications		
- Fusion Middleware:** Shows a green 3D gauge at 100% and a status indicator 'Up (3)'. Below it is a table with columns 'Name', 'Status', 'Host', and 'CPU Usage (%)'.

Name	Status	Host	CPU Usage (%)
WebLogic Domain			
webtier_domain			
AdminServer	Up	stadk58.us.oracle.com	0.19
Web Tier			
ohs1	Up	stadk58.us.oracle.com	0.00
webcache1	Up	stadk58.us.oracle.com	0.00
- Farm Resource Center:** Contains links for 'Before You Begin' and 'Typical Administration Tasks'.

OHS in Fusion Middleware Control

ohs1 Logged in as **weblogic** | Host stadk58.us.oracle.com

Oracle HTTP Server Page Refreshed May 12, 2009 11:13:52 AM PDT

Response and Load

10:05:18 AM 11:24:18 AM

- Request Processing Time (milli seconds)
- Request Throughput (requests per second)

CPU and Memory Usage

10:05:18 AM 11:24:18 AM

- CPU Usage (%)
- Memory Usage (MB)

Virtual Hosts

Name	Request Throughput	Response Size (KB)	Por
stadk58.us.oracle.com:4443	0.00	0.00	444
stadk58.us.oracle.com:9999	0.00	0.00	999

Module Request Statistics

Name	Throughput	Processed	Pro
mod_log_config.c	0.00	0	
mod_authz_host.c	0.00	0	
mod_actions.c	0.00	0	
mod_authn_dbm.c	0.00	0	
mod_status.c	0.00	0	
mod_dir.c	0.00	0	
mod_expires.c	0.00	0	
mod_auth_basic.c	0.00	0	
mod_mime.c	0.00	0	
mod_speling.c	0.00	0	

Resource Center

Before You Begin

- Introduction to Oracle HTTP Server
- Getting Started with Managing Oracle HTTP Server with Oracle Enterprise Manager
- Typical Administration Tasks

Starting and Stopping Oracle HTTP Server

- OHS processing lifecycle is managed by OPMN
- Use FMW Control or opmnctl to start and stop OHS
- To start Oracle Instance (OPMN), run:

```
$ opmnctl start
```

- To start Oracle Instance and all components, run:

```
$ opmnctl startall
```

- To check status using opmnctl, run:

```
$ opmnctl status
```

```
Processes in Instance: instance1
```

ias-component	process-type	pid	status
ohs1	OHS	22246	Alive



Starting and Stopping Oracle HTTP Server Cont.

- Starting OHS using opmnctl:
 - To start all OHS components in an Oracle Instance
`$ opmnctl startproc process-type=OHS`
 - To start a specific OHS component in an Oracle Instance
`$ opmnctl startproc ias-component=ohs1`
- Stopping OHS using opmnctl:
 - To stop all OHS components in an Oracle Instance
`$ opmnctl stopproc process-type=OHS`
 - To stop a specific OHS component in an Oracle Instance
`$ opmnctl stopproc ias-component=ohs1`
- To restart OHS using opmnctl
`$ opmnctl restartproc ias-component=ohs1`



Verifying Ports Used by OHS

- OHS comes configured with three listen ports:
 - A non-SSL (HTTP) port
 - Defaults to 7777, configured in httpd.conf
 - If 7777 is already in use, the next available port from the range 7777-7877 is assigned
 - A SSL (HTTPS) port
 - Defaults to 4443, configured in ssl.conf
 - If 4443 is already in use, the next available port from the range 4443-4543 is assigned
 - An additional SSL port
 - Called Proxy MBean or Admin port
 - Used internally by OHS to communicate with FMW Control
 - Defaults to 9999, configured in admin.conf
- Specify ports of your choice during installation or when creating a new OHS component

Verifying Ports Used by OHS Cont.

- Verify looking at Port Usage or Port Config Page in FMW Control

ORACLE Enterprise Manager 11g Fusion Middleware Control Setup Help Log Out

Farm | Topology

ojs1 ohs1
Oracle HTTP Server

Logged in as **weblogic** | Host stadk58.us.oracle.com
Page Refreshed May 12, 2009 11:27:22 AM PDT

Port Usage

Port in Use	IP Address	Component	Protocol
9999	ALL	ohs1	https
4443	ALL	ohs1	https
7782	ALL	ohs1	http

ORACLE Enterprise Manager 11g Fusion Middleware Control Setup Help Log Out

Farm | Topology

ojs1 ohs1
Oracle HTTP Server

Logged in as **weblogic** | Host stadk58.us.oracle.com
Page Refreshed May 12, 2009 11:28:59 AM PDT

Ports Configuration

All ports configured for this system component are shown. Some ports configured of type Admin cannot be edited or deleted.

View Create... Edit... Delete...

Port	Port Type	Host Name	IP Address	Protocol
7782	Listen	stadk58.us.oracle.com	ANY	HTTP
4443	Listen	stadk58.us.oracle.com	ANY	HTTPS
9999	Admin	stadk58.us.oracle.com	ANY	HTTPS

Verifying Ports Used by OHS Cont.

- Verify the allocated ports from ports.prop file in \$ORACLE_INSTANCE/config/OPMN/opmn directory

```
/ohs1/ListenPort=7782
```

```
/ohs1/SSLPort=4443
```

```
/ohs1/ProxyPort=9999
```

- For a running OHS, using opmnctl status cmd with -l:

```
$ opmnctl status -l
```

```
Processes in Instance: instance1
```

```
-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
ias-component          | process-type      | pid | status |
uid | memused | uptime | ports
-----+-----+-----+-----+-----+
ohs1                   | OHS               | 22246 | Alive  | 21039
2698 | 51300 | 165:08:22 | https:9999,https:4443,http:7782
```



Provisioning Using opmnctl

- Use opmnctl to provision Oracle Instances and system components

- Create a new Oracle Instance

```
$ opmnctl createinstance -oracleInstance /oracle/inst1  
-adminHost myahostname -adminPort 7001 -  
adminRegistration OFF
```

- Register and unregister an Oracle instance with a WebLogic domain

```
$ opmnctl registerinstance -adminHost myhostname  
-adminPort 7001
```

- Create and delete an OHS component

```
$ opmnctl createcomponent -componentType OHS -  
componentName ohs2
```

```
$ opmnctl deletecomponent -componentName ohs2
```

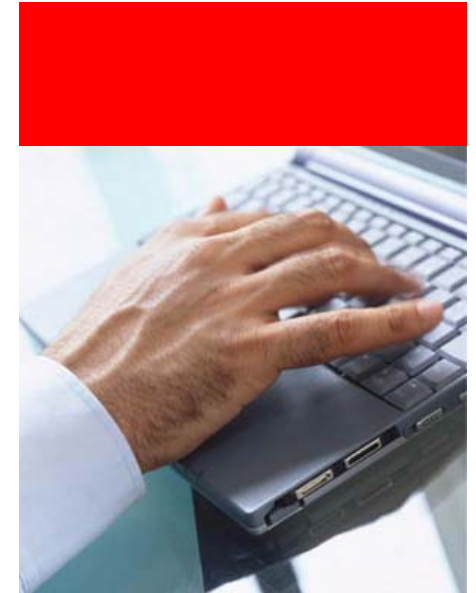


Provisioning Using opmnctl Cont.

- Ports are automatically assigned at create time
 - Define your own ports using additional parameters:
 - listenPort: HTTP listening port
 - sslPort: HTTPS (SSL) listening port
 - proxyPort: Proxy MBean (admin) port
 - Update an Oracle instance or component registration with the WebLogic domain
 - Needed when OHS's Admin/Proxy MBean port is changed
- ```
$ opmnctl updatecomponentregistration -
componentName ohs1 -proxyPort 8989
```

# Program Agenda Example

- Overview of Oracle HTTP Server
  - Intro to OHS and differences with Apache
  - Support policy
- OHS Bundling and Installation
  - OHS Bundling
  - Install Concepts and Options
- Management of OHS
  - Management Tools for OHS
  - Common OHS management tasks
- **Configuring OHS**
  - **OHS Configuration Files**
  - **Configuring Ports and Logs**
- WebLogic Server Plug-In
  - Overview of mod\_weblogic
  - Difference between mod\_wl\_ohs and mod\_weblogic
  - Configuring mod\_wl\_ohs





## OHS Configuration Files

- OHS configuration files are contained in the config directory of an Oracle Instance
  - \$OI/config/OHS/<ohs\_name>
- Main OHS config file is httpd.conf that includes:
  - Directives to configure server and modules
  - Reference to other configuration files
    - Such as ssl.conf, mod\_wl\_ohs.conf, admin.conf etc.
- Three ways to make OHS configuration changes:
  - Config pages in FMW Control, primary and recommended
  - Direct file editing, when you know what you are doing
    - Editing file directly from file system using text editor
    - Using Advanced Configuration page in FMW Control
  - Using WLST and OHS config MBeans, good for scripting

# Using FMW Control to Configure OHS

The screenshot displays the Oracle HTTP Server (OHS) FMW Control interface. The top bar shows the user is logged in as 'weblogic' on host 'stadk58.us.oracle.com'. The interface is divided into several sections:

- Navigation Menu:** A sidebar on the left contains a tree view with categories: Home, Monitoring, Control, Logs, Port Usage, Administration, Security, and General Information. The 'Administration' category is expanded, showing sub-items: Virtual Hosts, Performance Directives, Log Configuration, Server Configuration, MIME Configuration, Ports Configuration, mod\_perl Configuration, mod\_wl\_ohs Configuration, and Advanced Configuration.
- Monitoring Graphs:** Two line graphs are visible. The first graph shows CPU and Memory usage over time (11:52 to 11:56). The second graph, titled 'CPU and Memory Usage', shows a similar trend from 11:44 AM to 11:56 on May 12, 2009. A search box below the graphs shows the time '10:05:18 AM'.
- Virtual Hosts Table:** A table listing virtual hosts with columns for Name, Size (KB), and Port. Two entries are visible: 'stadk58.us.oracle.com:443' and 'stadk58.us.oracle.com:9999'.
- Module Request Statistics Table:** A table showing the performance of various modules. The columns are Name, Throughput, Processed, and Pro (likely Processed/Total). All modules listed have a throughput of 0.00 and 0 processed requests.

| Name             | Throughput | Processed | Pro |
|------------------|------------|-----------|-----|
| mod_log_config.c | 0.00       | 0         |     |
| mod_authz_host.c | 0.00       | 0         |     |
| mod_actions.c    | 0.00       | 0         |     |
| mod_authn_dbm.c  | 0.00       | 0         |     |
| mod_status.c     | 0.00       | 0         |     |
| mod_dir.c        | 0.00       | 0         |     |
| mod_expires.c    | 0.00       | 0         |     |
| mod_auth_basic.c | 0.00       | 0         |     |
| mod_mime.c       | 0.00       | 0         |     |
| mod_speling.c    | 0.00       | 0         |     |



## Configuring Listen Address

- To create a new listening address
  - Use Port Configuration page in FMW Control, or
  - Manually add `Listen` directive in a configuration file
- A port can listen on a specific IP address, hostname or any available network interfaces, e.g.
  - `Listen 140.87.8.163:8787`
  - `Listen stadk58.us.oracle.com:8787`
  - `Listen 8787`
- SSL for a port is configured via Virtual Host
- Changing admin/proxy MBean port
  - Should be done manually, can not edit using FMW Control
  - Must run `opmnctl updatecomponentregistration` command to have port updated in WLS domain



## Configuring OHS to Run on Privileged Port

- On Unix, ports below 1024 are considered privileged
  - Only processes with root privileges can listen on those ports
- By default, OHS runs as a non-root user
  - The user that installed the product
- To run on port <1024, enable OHS to run as root
  - Login as root user
  - Go to `ORACLE_HOME/ohs/bin` directory and run

```
$ chown root .apachectl
```

```
$ chmod 6750 .apachectl
```
  - Make sure `User` and `Group` directives (in `httpd.conf`) are set to the one that installed the product
  - Stop and start OHS (do not use `opmnctl restartproc` command)



## Configuring OHS Logs

- Types of OHS logs:
  - Console log (console~OHS~1.log)
    - Records stdout and stderr messages
    - Generated by OPMN
  - Error log (<ohs\_name>.log)
    - Records server problems
  - Access log (access\_log)
    - Records which applications are being accessed and by whom
  - PID file (httpd.pid)
    - Records the process ID of the parent httpd process
- Default location for all OHS log files:
  - \$OI/diagnostics/logs/OHS/<ohs\_name>



## OHS Log Rotation

- Log rotation is necessary to control the log files size
- Rotation via the use of piped logs
  - Instead of writing directly to file, log messages are written through a pipe to another process
- Achieved using `odl_rotatelog` program
  - Used by default for error and access logs
  - It is Oracle's version of `rotatelog` from Apache
  - Provides all `rotatelog` functionality plus log retention feature
- Supports two types of log rotation policies
  - Size-based
    - Rotate whenever configured size is reached
  - Time-based
    - Rotate whenever configured time is reached



## Configuring OHS Error Logs

- Two log formats
  - Legacy Apache message format, text
  - Oracle Diagnostic Logging (ODL), choice of text or XML
    - A common format for diagnostic and log messages for all components across FMW
    - Provides a mechanism (ECID) for correlating messages from various components
- ODL-Text is the default format for recording log messages
- Configure error log
  - Using Log Configuration page in FMW Control
  - Manually editing directives in httpd.conf file

# OHS Error Log Configuration in FMW Control

ohs1  
Oracle HTTP Server

Logged in as **weblogic** | Host stadk58.us.oracle.com  
Page Refreshed May 12, 2009 4:03:57 PM PDT

**Information**  
All changes made in this page require a server restart to take effect.

## Log Configuration

Apply Revert

### Error Log

Oracle HTTP Server records server error information in error logs. Specify the error log settings, including whether to generate log messages in Oracle Diagnostic Logging (ODL) text, ODL XML or Apache format. ODL is a standard format and mechanism for correlating the diagnostics messages from components across Oracle Fusion Middleware. To view log messages choose Logs from the Oracle HTTP Server menu.

#### General

File Format  ODL-Text  ODL-XML  APACHE

Log File/Directory

Level

#### Rotation Policy

No Rotation  Size Based  Time Based

\* Maximum Log File Size (MB)

Maximum Files To Retain

Start Time

Rotation Frequency

Retention Period



## OHS Error Log Directives

- OraLogMode
  - Specifies the error log format (Apache, ODL-Text, ODL-XML)
- ODL error log directives
  - Used only when OraLogMode is set to ODL-Text or ODL-XML
  - OraLogDir
    - Specifies the error log path
  - OraLogSeverity
    - Enables you to set error message severity
    - `OraLogSeverity <msg_type>[:msg_level]`
  - OraLogRotationParams
    - Enables you to choose the error log rotation policy
    - `OraLogRotationParams <type> <policy>`



## OHS Error Log Directives Cont.

- type: S (sized-based) or T (time-based)
- policy: <maxFileSize:allFilesSize> | <freq retentionTime startTime>
- Apache error log directives
  - Used only when OraLogMode is set to Apache
  - ErrorLog
    - Sets the path and name of the error log file
    - Rotation policy is specified within this directive
    - `ErrorLog "|odl_rotatelog <file> <rotation policy>"`
  - LogLevel
    - Sets the verbosity of the error log messages
    - `LogLevel <level>`
    - Default is "warn"



## Configuring OHS Access Logs

- Contains information about every HTTP txn handled by the server
- For access log, you configure
  - Log format
    - What information to include in the access log
  - Log file path/name
    - Path and name of the access log file
  - Log rotation policy
    - Same as for error logs – size or time based
- Can define multiple access logs
  - Each with its own log format and rotation policy
- Configure access log using
  - Log Configuration page in FMW Control
  - Manually editing directives in httpd.conf file



## Configuring OHS Access Logs Cont.

- Access log format
  - Can define multiple named access log formats
  - The default format is the Common Log Format (CLF):

```
%h %l %u %t \"%r\" %>s %b
```

### Where

%h: Remote host

%l: Remote log name, if supplied

%u: Remote user

%t: Time in common log format

%r: First line of request

%s: Status

%b: Bytes sent, excluding HTTP headers

# OHS Access Log Configuration in FMW Control

## Access Log

Access logs record all requests processed by the server. The logs contain basic information about every HTTP transaction handled by the server. If you want to create a new log format, update an existing one with a new value, or delete one, click **Manage Log Formats**.

| <a href="#">+ Create</a> <a href="#">✎ Edit...</a> <a href="#">✖ Remove</a>   <a href="#">Manage Log Formats</a> |            |
|------------------------------------------------------------------------------------------------------------------|------------|
| Log File Path                                                                                                    | Log Format |
| <code>\${ORACLE_INSTANCE}/diagnostics/logs/\${COMPONENT_TYPE}/\${COMPONENT_NAME}/access_log</code>               | common     |

### Edit Access Log

\* Log File Path:

Log Format:

**Rotation Policy**

No Rotation  Size Based  Time Based

\* Maximum Log File Size (MB):  Start Time:

Maximum Files To Retain:  Rotation Frequency:

Retention Period:

## Manage Custom Access Log Formats

Use the following section to create, edit, or remove log formats for access logs.

| <a href="#">+ Add Row</a> <a href="#">✖ Remove</a> |                                                                                                              |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Log Format Name                                    | Log Format Pattern                                                                                           |
| <input type="text" value="combined"/>              | <input %&gt;s="" %b="" %r\"="" \"%{referer}i\"="" \"%{user-agent}i\""="" type="text" value="%h %l %u %t \"/> |
| <input type="text" value="common"/>                | <input %&gt;s="" %b"="" %r\"="" type="text" value="%h %l %u %t \"/>                                          |
| <input type="text" value="referer"/>               | <input %u"="" -&gt;="" type="text" value='\"%{Referer}i'/>                                                   |
| <input type="text" value="agent"/>                 | <input type="text" value='\"%{User-agent}i"/'/>                                                              |



## OHS Access Log Directives

- LogFormat

- Specifies a named access log format
- By default, OHS comes with four formats defined:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b
 \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

```
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent
```

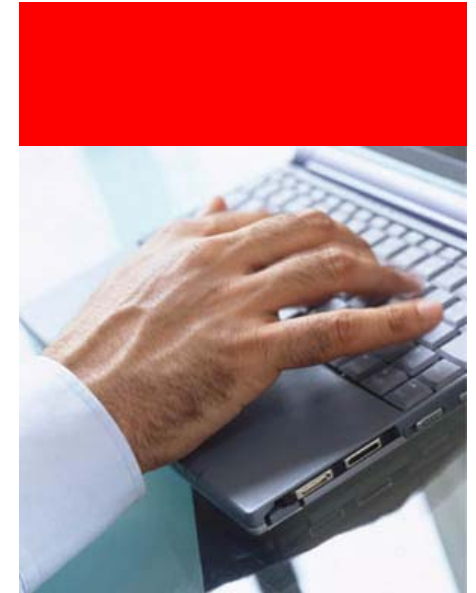
- CustomLog

- Sets the path and name of access log file and the format to use

```
CustomLog "|odl_rotatelogs <file> <rotation
 policy>" <format>
```

# Program Agenda Example

- Overview of Oracle HTTP Server
  - Intro to OHS and differences with Apache
  - Support policy
- OHS Bundling and Installation
  - OHS Bundling
  - Install Concepts and Options
- Management of OHS
  - Management Tools for OHS
  - Common OHS management tasks
- Configuring OHS
  - OHS Configuration Files
  - Configuring Ports and Logs
- **WebLogic Server Plug-In**
  - **Overview of mod\_weblogic**
  - **Difference between mod\_wl\_ohs and mod\_weblogic**
  - **Configuring mod\_wl\_ohs**





## Overview of WLS Plug-in for Apache (mod\_weblogic)

- Allows requests to be proxied from Apache HTTP server to WebLogic Server
  - Proxies requests based on user defined configuration
  - Can proxy based on
    - The URL of the request (or a portion of the URL)
    - The MIME type of the requested file
  - Uses basic round robin method to load balance requests among available WebLogic Servers
- Enhances Apache by allowing WLS to handle requests that require WLS's dynamic functionality
- Intended for use where Apache serves static pages, and another part is delegated to WLS
- Operates as a module within Apache HTTP server



## Clustering Failover with Plug-in

- The plug-in ensures stickiness
  - It decipheres the session cookie to find WLS server the session was originally established
  - Then routes the request to that WLS server
  - Cookie contains primary and secondary server info (sent by WLS)
- Failover with single server
  - Server defined with WebLogicHost and WebLogicPort parameters
  - If first connection attempt fails
    - Continues trying to connect for max of either two or ConnectTimeoutSecs/ConnectRetrySecs times
      - Sleeps for ConnectRetrySecs between each attempt
    - If unsuccessful, returns HTTP 503 error



## Clustering Failover with Plug-in Cont.

- Failover with multiple servers
  - Servers defined with WebLogicCluster parameter
  - If connection attempt to primary server fails
    - Connects to secondary server (obtained from cookie)
  - If connection attempt to secondary server fails
    - Using round robin obtains next server from
      - Dynamic list when DynamicServerList is set to ON
      - Otherwise from WebLogicCluster parameter
  - If connection attempt to next server fails
    - Continues trying to connect for max of either two or ConnectTimeoutSecs/ConnectRetrySecs times
      - Sleeps for ConnectRetrySecs between each attempt
    - If still unsuccessful, returns HTTP 503 error



## Overview of WLS Plug-in for OHS (mod\_wl\_ohs)

- Provides the same functionality as mod\_weblogic except for following differences:
  - Uses Oracle's security layer (NZ) to provide SSL support
    - Mod\_weblogic uses Certicom API
  - Supports two-way SSL between OHS and WLS
    - Mod\_weblogic supports one-way SSL
  - Supports IPv6 for communication with WLS
- Installed with OHS but not configured by default
- Supported on all platforms where FMW is supported
  - Refer [FMW Supported System Configurations](#) for details
  - Mod\_weblogic carries its own platform certification

# Configuring mod\_wl\_ohs in FMW Control

**Information**  
All changes made in this page require a server restart to take effect.

**mod\_wl\_ohs Configuration** Apply Revert

**General**

The mod\_wl\_ohs module allows requests to be proxied from an Oracle HTTP Server to Oracle WebLogic Server. Specify parameters common for all URLs directed to the mod\_wl\_ohs handler, and use the MatchExpression and Location sections to specify any overrides.

WebLogic Cluster

WebLogic Host

WebLogic Port

Dynamic Server List ON

Error Page

Debug

Log File

WebLogic Temp Directory

Exclude Path or Mime Type

**Match Expressions**

+ Add Row ✕ Remove

| Expression   |
|--------------|
| <br><br><br> |

**TIP** For example MatchExpression can have the following value:  
\*.jsp WebLogicHost=myHost|WebLogicPort=7001|Debug=ON|PathPrepend=/test2

**Locations**

+ Add Row ✕ Remove

| Location              | WebLogic Cluster | WebLogic Host | WebLogic Port | Path Trim |
|-----------------------|------------------|---------------|---------------|-----------|
| No Locations Defined. |                  |               |               |           |



## Configuring mod\_wl\_ohs

- Module comes as a shared object (.so)
  - Module located under \$OH/ohs/modules directory
  - Enabled by default to load when OHS starts
- Configuration defined in mod\_wl\_ohs.conf
  - Located under \$OI/config/OHS/<ohs\_name> directory
  - Included in httpd.conf via `Include` directive
  - When using SSL between OHS and WLS
    - Mod\_weblogic requires config not being defined in separate file and included via `Include` directive
    - No such restriction with mod\_wl\_ohs



## Configuring mod\_wl\_ohs Cont.

- The main sections of mod\_wl\_ohs.conf are:
  - LoadModule
    - Loads the weblogic\_module when OHS starts
  - IfModule
    - Specify the host and port of the WLS server or cluster
    - Specify MatchExpression for proxying by MIME type
  - Location
    - Specify application context root for proxying by path
    - If both MIME type and proxying by path are configured, proxying by path takes precedence
- For config page in FMW Control to work properly
  - Commented directives in IfModule block are needed
  - All directives should be defined within IfModule block



# Configuring mod\_wl\_ohs Cont.

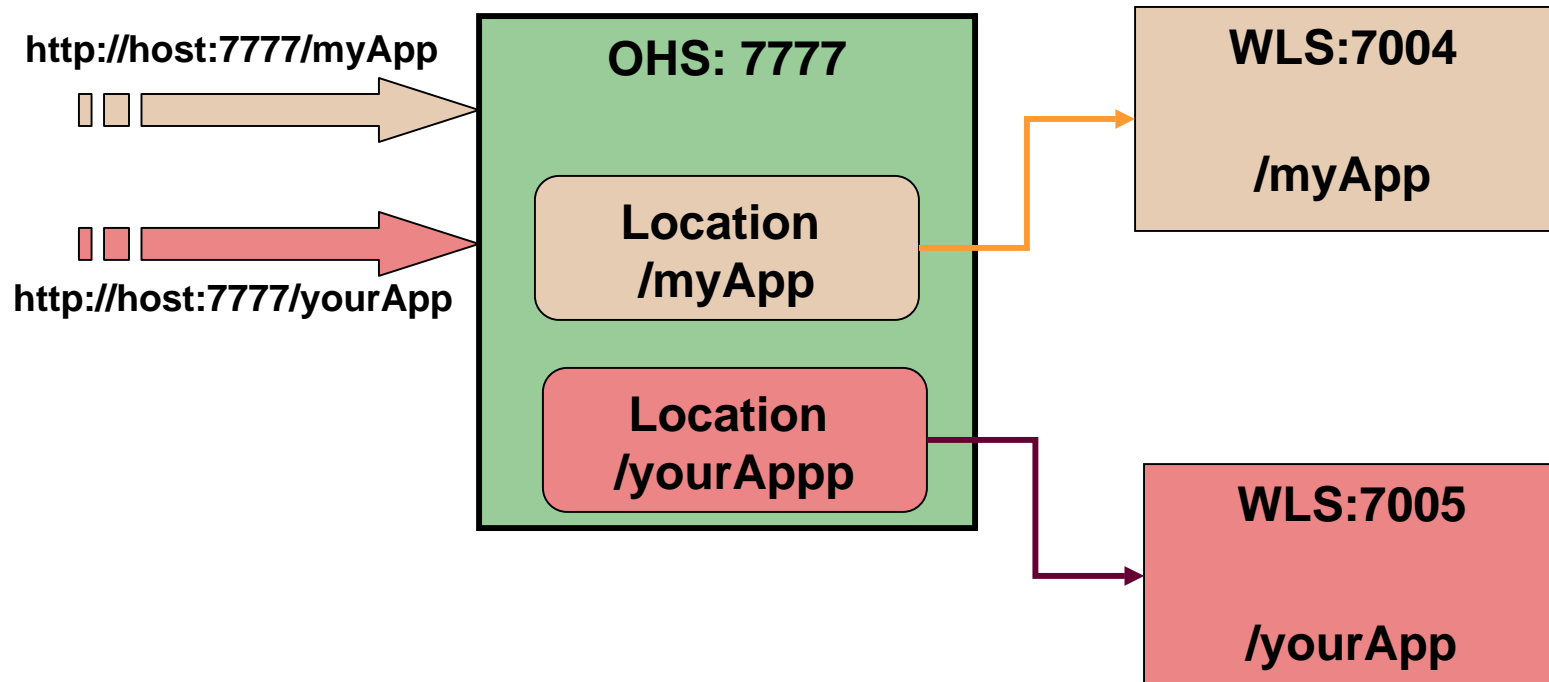
- Sample configuration file

```
LoadModule weblogic_module "${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"

This empty block is needed to save mod_wl related configuration from EM to this
file when changes are made at the Base Virtual Host Level
<IfModule mod_weblogic.c>
 # WebLogicHost <WEBLOGIC_HOST>
 # WebLogicPort <WEBLOGIC_PORT>
 # Debug ON
 # WLogFile /tmp/weblogic.log
 # MatchExpression *.jsp
 WebLogicHost stadk58.us.oracle.com
 WebLogicPort 7001
 WLogFile /tmp/mod_wl_ohs.log
 MatchExpression *.jsp
 <Location /em>
 SetHandler weblogic-handler
 WebLogicHost stadk58.us.oracle.com
 WebLogicPort 7001
 </Location>
</IfModule>
```

## Configuring mod\_wl\_ohs Cont.

- Can have one OHS proxying to multiple WLS that are serving different applications





## Configuring SSL with mod\_wl\_ohs

- Can configure SSL for communication between OHS and WLS
- To enable one-way SSL:
  - Generate a custom keystore for WLS containing a certificate
  - Import the certificate into OHS wallet as a trusted certificate
  - Add following new directive in mod\_wl\_ohs.conf under IfModule block:

```
WlSSLWallet
 "${ORACLE_INSTANCE}/config/OHS/ohs1/keystores/default"
```

Here ohs1 is OHS component name and default is OHS wallet name
  - Ensure that WLS is configured to use the custom keystore generated earlier
  - Restart WLS and OHS



## Configuring SSL with mod\_wl\_ohs Cont.

- To enable two-way SSL:
  - Perform one-way SSL configuration steps
  - Generate a new trust store, trust.jks, for WLS
    - Keystore created for one-way SSL could be used, but it is recommended to create a separate truststore
  - Export the user certificate from OHS wallet, and import it into the truststore created above
  - At Kystores tab in WLS Admin Console for the server set
    - The custom trust store with the trust.jks file location
    - The keystore type as JKS
    - The passphrase used to create the keystore
  - Under the SSL tab
    - Ensure trusted CA is set as from Custom Trust Keystore



## References & Contact

- Oracle HTTP Server Admin Guide
  - [http://download.oracle.com/docs/cd/E15523\\_01/web.11111/e10144/toc.htm](http://download.oracle.com/docs/cd/E15523_01/web.11111/e10144/toc.htm)
- Using Web Server Plug-ins with WLS Guide
  - [http://download.oracle.com/docs/cd/E15523\\_01/web.11111/e14395/toc.htm](http://download.oracle.com/docs/cd/E15523_01/web.11111/e14395/toc.htm)