Oracle9*i*AS Web Cache

Release Notes

Release 2 (9.0.2)

April 2002

Part No. A97301-01

This document summarizes the differences between Oracle9*i*AS Web Cache and its documented functionality.

> **See Also:** *Oracle9i Application Server Release Notes*

1 General Issues and Workarounds

This section describes general issues and their workarounds for Oracle9*i*AS Web Cache. It contains the following sections:

- Section 1.1, "Default Buffer Sizes"
- Section 1.2, "Insufficient Input Checking in Oracle9iAS Web Cache Manager"
- Section 1.3, "Oracle Application Server Limitations"
- Section 1.4, "TCP/IP Tuning"

1.1 Default Buffer Sizes

Oracle9*i*AS Web Cache uses 2 KB for the access log buffer size and the following for cached documents:

- 4 KB for the HTTP headers
- 3 KB for a single HTTP header field
- 32 KB for the HTTP response body

  If the HTTP response body is less than 4 KB, then Oracle9*i*AS Web Cache uses a 4 KB buffer size.

These default sizes should be sufficient for most deployments. If you need to change any of these default sizes, contact Oracle Support Services or consultants.

1.2 Insufficient Input Checking in Oracle9*i*AS Web Cache Manager

Oracle9*i*AS Web Cache Manager, the graphical user interface tool for configuring Oracle9*i*AS Web Cache, does not enforce the same level of consistency checking upon receiving configuration input that Oracle9*i*AS Web Cache does upon starting up. Therefore, there may be instances where configuration changes are accepted by the Oracle9*i*AS Web Cache Manager, but Oracle9*i*AS Web Cache does not start up with the resulting configuration.

This is especially a problem when the `admin` server process is shut down (with the `webcachectl stop` or `webcachectl stopadm` command) after applying bad configuration changes. In that case, the `admin` server process will not be able to start up, and the Oracle9*i*AS Web Cache Manager will become inaccessible. To work around this problem, run the `webcachectl reset` command to restore to the previous version of the configuration. `webcachectl` is located in `$ORACLE_HOME/bin` on UNIX and *`ORACLE_HOME`*`\bin` on Windows.

1.3 Oracle Application Server Limitations

> **Note:** This limitation applies to Oracle Application Server using the Oracle Web Listener only. This does not apply to other listeners used with Oracle Application Server. It also does not apply to Oracle9*i* Application Server.

Oracle Application Server, when used specifically with the Oracle Web Listener, strictly enforces virtual-host checking using the `Host` request header, that is sent by almost all browsers. The `Host` header contains the string "*`hostname:port`*" where *`hostname`* and *`port`* are as entered in the location bar of the browser, even if the *`hostname`* in the location bar is an IP address. If the Oracle Web Listener receives a `Host` header that does not match an entry in the `Host Name` and `Port` columns of the `Network` section of the configuration (corresponding to the `[Multiport]` section in the `sv*.cfg` configuration file for the listener), it returns an HTTP error code 400 indicating that the request did not specify a valid virtual host.

1.3.1 Deployments

This limitation applies to the following deployments:

1. In Oracle Application Server 4.*x*, this is strictly enforced for HTTP/1.1 requests only. For HTTP/1.0 requests, only the host name has to match an entry in the `Network` section. The port number in the `Host` header does not matter.
2. Oracle Web Listener, as shipped with Oracle Web Application Server 3.0 enforces this strictly for all HTTP requests, that is HTTP/1.0 and HTTP/1.1.

1.3.2 Entries for Network

To make an Oracle Web Listener recognize and accept a `Host` header, a corresponding entry must be added to the `Network` section for that listener. When you add an entry with host name h1 and port `p1`, `h1` is only used to match incoming `Host` headers and does not otherwise effect the operation of the listener. h1 does not need to be a DNS host name of the computer. However, `p1` is used as a port that the Oracle Web Listener tries to listen on. Hence there should be no other process on the computer listening on port `p1`.

1.3.3 Oracle9*i*AS Web Cache Behavior

Oracle9*i*AS Web Cache does not change the Host header that it receives as part of a request when relaying that request to the application Web server.

If you set up Oracle9*i*AS Web Cache to listen on port 1100 on a computer with DNS host name `m1` and the Application Web Server is on computer `m2` on port 80, and you use a browser to access `http://m1:1100/`, the `Host` header received by Oracle9*i*AS Web Cache is "`Host: m1:1100`". This is exactly the `Host` header that will be received by the application Web server.

1.3.4 What Restrictions Does This Imply?

If you are using Oracle Application Server with the Oracle Web Listener as your application Web server, this means that the `Host` header sent to Oracle9*i*AS Web Cache must be recognized by the Oracle Web Listener, that is, there must be a corresponding entry in the `Network` section.

If you are using Oracle9*i*AS Web Cache's host name and port directly in your browser, and if Oracle9*i*AS Web Cache and the Oracle Web Listener are on the same computer, the Oracle Web Listener will need to accept Oracle9*i*AS Web Cache's host name and port number in the `Host` header, and for that to occur Oracle9*i*AS Web Cache's port number needs to be in the `Network` section of the Listener's configuration. This would mean that both Oracle9*i*AS Web Cache and the Oracle Web Listener will try to listen on that port, which is not possible. See Section 1.3.2, "Entries for Network".

When you are using your browser to connect directly to Oracle9*i*AS Web Cache, ensure that Oracle9*i*AS Web Cache and the Oracle Web Listener are not on the same computer.

To deploy Oracle9*i*AS Web Cache and the Oracle Web Listener on the same computer, there has to be a port-translating switch between the browser and Oracle9*i*AS Web Cache that translates the port number to which the browser connects to Oracle9*i*AS Web Cache's listening port.

For example, assume that Oracle9*i*AS Web Cache is listening on port 7777 on computer `m1.aaa.com`, and Oracle Web Listener is the application Web server listening on port 80 on the same computer `m1.aaa.com`. In this example, the user enters `http://www.aaa.com/` in the browser. The browser will attempt to connect to port 80 on `www.aaa.com`. `www.aaa.com` should be resolved through DNS to the switch, which should redirect requests for `www.aaa.com` on port 80 to computer m1 on port 1100. Note that the Host header will be: "`Host: www.aaa.com:80`". Oracle9*i*AS Web Cache will forward requests as needed to computer m1 port 80, that is, the Oracle Web Listener. For the Oracle Web Listener to accept this `Host` header, you will need to have added an entry to `Network` containing host name `www.aaa.com` and port 80. See Section 1.3.2, "Entries for Network" for how this can done on computer `m1`.

1.4 TCP/IP Tuning

If you want Oracle9*i*AS Web Cache to handle a large number of concurrent HTTP requests, you may need to tune TCP/IP settings for your operating system, such as the maximum TCP/IP connection queue length.

> **See Also:** Operating-system-specific documentation. For example, `http://www.rvs.uni-hannover.de/people/voeckler/tune/EN/tune.html` describes how to tune Solaris 2.*x* TCP/IP parameters.

In particular, if you run stress tests against Oracle9*i*AS Web Cache and continuously open more TCP/IP connections from one client computer to Oracle9*i*AS Web Cache, you may experience periodic oscillation of throughput. This is usually caused by TCP/IP connection `TIME_WAIT` in your operating system. In real world deployments, this is not an issue since it is unlikely that a single client will generate a huge number of connections.

In case of denial of service attack, availability problems usually arise in the network layer in your operating system. For example, if one client generates large number of connections, TCP/IP connection problems generally arise in your operating system.

The following examples demonstrate utilities that set some of the TCP/IP parameters for Solaris, HP-UX, AIX, Linux, and Compaq Tru64.

Example Utilities Script for Solaris 2.x

```
#!/usr/bin/bash -x
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q 10240
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q0 10240
/usr/sbin/ndd -set /dev/tcp tcp_xmit_hiwat 32768
/usr/sbin/ndd -set /dev/tcp tcp_recv_hiwat 32768
/usr/sbin/ndd -set /dev/tcp tcp_time_wait_interval 1000
```

Example Utilities Script for HP-UX

```
#!/usr/bin/ksh
/usr/bin/ndd -set /dev/tcp tcp_conn_req_max_q 10240
/usr/bin/ndd -set /dev/tcp tcp_time_wait_interval 1000
```

Example Utilities for AIX

Use the `no` utility to set the `tcp` tunable values on AIX.

For example, to see current values, enter:

```
prompt> no -a
```

To increase the size of send and receive buffers, enter:

```
prompt> no -o tcp_sendspace=65536
prompt> no -o tcp_recvspace=65536
```

Example Utilities for Linux

On Linux, the tunable TCP/IP parameters can be set through the `/proc` file system.

The `/proc/sys/net/ipv4` directory contains the files which can be edited to change the TCP/IP default values.

For example, enter the following commands:

```
prompt> echo 900 > /proc/sys/net/ipv4/tcp_keepalive_time
prompt> echo 10 > /proc/sys/net/ipv4/tcp_fin_timeout
```

Also, you can set the size of TCP/IP sender and receiver windows using the following command, where *w_value* and *r_value* are the new sizes of the windows:

```
prompt> echo w_value /proc/sys/net/core/wmem_max
prompt> echo r_value /proc/sys/net/core/rmem_max
```

Example Utilities for Compaq Tru64

Use the `sysconfig` utility to set the TCP/IP tunable values for Tru64.

For example, to set the value of `tcbhashsize` to 16384, enter:

```
prompt> sysconfig -r inet tcbhashsize=16384
```

To enable keepalive, enter:

```
prompt> sysconfig -r inet tcp_keepalive_default=1
```

To set the maximum number of pending TCP/IP connections, enter:

```
prompt> sysconfig -r socket somaxconn=65535
```

2 Configuration Issues and Workarounds

This section describes configuration issues and their workarounds for Oracle9*i*AS Web Cache. It contains the following sections:

- Section 2.1, "Oracle9iAS Web Cache Configuration Basics"
- Section 2.2, "Configuration File Upgrades"
- Section 2.3, "Oracle9iAS Web Cache Default Ports"
- Section 2.4, "Mismatched Oracle Home Definitions Causes Web Cache to Fail to Start"
- Section 2.5, "Hierarchical Caching"
- Section 2.6, "Invalidation Time-outs"
- Section 2.7, "Directives to Oracle HTTP Server"
- Section 2.8, "ESI Support"
- Section 2.9, "+wcdebug Limitations"
- Section 2.10, "Load Balancing Changes"
- Section 2.11, "Session Binding"
- Section 2.12, "Concurrent Users on Windows NT Workstation"
- Section 2.13, "Browser Limitations"
- Section 2.14, "Cache Cluster Event Logs"
- Section 2.15, "Peer-to Peer-Logging"

2.1 Oracle9*i*AS Web Cache Configuration Basics

To start initial Oracle9*i*AS Web Cache configuration:

1. If not currently logged on to the Oracle9*i*AS Web Cache computer, log in with the user ID of the user that performed the installation.
2. Start Oracle9*i*AS Web Cache.

   From the command line, enter `webcachectl start`.
3. Point your browser to the URL `http://`*`web_cache_hostname`*`:4000/`
4. When prompted for the administrator user ID and password, enter administrator for the user name and the appropriate password. The first time you log in, the password is `administrator`.

> **See Also:** *Oracle9iAS Web Cache Administration and Deployment Guide* (available at `http://otn.oracle.com/products/ias/web_cache/`) for complete configuration coverage

Oracle9*i*AS Web Cache uses two configuration files: `webcache.xml` and `internal.xml`. The Oracle9*i*AS Web Cache Manager writes its configuration information to the `webcache.xml` file. Oracle9*i*AS Web Cache uses `internal.xml` file. These files are located in the `$ORACLE_HOME/webcache` directory on UNIX and *`ORACLE_HOME`*`\webcache` directory on Windows. Do not edit these configuration files manually, except in the cases described in these Release Notes, or when directed to do so by Oracle Support Services. Improper editing of these configuration files may cause problems in Oracle9*i*AS Web Cache.

2.2 Configuration File Upgrades

In past releases, the following attributes required manual modification of the `internal.xml` file:

- `KEEPALIVE_TIMEOUT` attribute specifies the time, in seconds, for Oracle9*i*AS Web Cache to keep a connection open to the browser after it has returned a response.
- `OSRECV_TIMEOUT` attribute specifies the time, in seconds, for the origin server to generate a response to Oracle9*i*AS Web Cache.

In this release of Oracle9*i*AS Web Cache, these attributes have been merged into the Oracle9*i*AS Web Cache Manager (`webcache.xml`). During migration, these modifications are not preserved. If you modified these attributes, you have two choices. You can:

- Use the Network Timeouts page (**Cache-Specific Configuration** > **Network Timeouts**) of Oracle9*i*AS Web Cache Manager to reconfigure
- Preserve any change that you made to the old version of `internal.xml`, and copy the changes to the appropriate place in `webcache.xml`.

If you are upgrading an existing Oracle9*i*AS Web Cache installation, the passwords for administration and invalidation are reset. The user name and password for administration is `administrator`/`administrator` and the user name and password for invalidation is `invalidator`/`invalidator`. You can change both passwords in the Security page (**General Configuration** > **Security**) of Oracle9*i*AS Web Cache Manager.

2.3 Oracle9*i*AS Web Cache Default Ports

By default, Oracle9*i*AS Web Cache is configured to use the following default TCP ports:

- Listening HTTP Requests: 7777
- Listening HTTPS Requests: 4443
- Administration HTTP Requests: 4000
- Invalidation HTTP Requests: 4001
- Statistics HTTP Requests: 4002

If these ports are in use, the installation procedure attempts to assign other port numbers from a range of possible port numbers.

See Also: *Oracle9i Application Server Installation Guide*

The default configuration does not enable HTTPS for the operations (administration, invalidation, or statistics monitoring) requests. Instead, these ports are configured for HTTP basic authentication. The passwords for the `administrator` user and the `invalidator` user can be decoded when they are sniffed out of the HTTP traffic. To avoid breach of security information for unprotected and insecure networks, modify the protocol to HTTPS in the Operations page (**Cache-Specific Configuration** > **Operations Ports**) to ensure that the passwords for these requests are secure.

The Oracle HTTP Server is configured to use the following default ports:

- HTTP Requests: 7778
- HTTPS Requests: 4444

At the end of installation, Oracle9*i*AS Web Cache will attempt to start. Depending on your environment (port conflicts, file/directory permissions, and so on), the `admin` server process and `cache` server process may fail to start.

If the `admin` server could not start because the default administration port is already in use by other running applications, you need to change the administration port in the `webcache.xml` file.

To change the administration port:

1. Locate the following line in the `webcache.xml` file:

   ```
   <LISTEN IPADDR="ANY" PORT="4000" PORTTYPE="ADMINISTRATION"/>
   ```

2. Change 4000 to an unused port, such as 3999.
3. Execute the `webcachectl start` command to start the `admin` server and `cache` server processes.

If the `admin` server process starts, but the `cache` server process does not start, point your browser `http://`*`web_cache_hostname`*`:`*`administration_port`*`/` to reconfigure the `cache` server process, and then start the `cache` server process from the Operations page (**Administration** > **Operations**).

If any of the port settings conflict with existing settings in the installed computer, reconfigure through `http://`*`web_cache_hostname`*`:`*`administration_port`*`/`. For example, if Oracle9*i*AS Web Cache is installed on a computer named `server1`, then you can reconfigure through `http://server1:4000/`.

After reconfiguring the administration port, you must restart the `admin` server process. After reconfiguring any other ports, you must restart the `admin` server process.

To restart the `admin` server process, execute either the `webcachectl restart` command or the `webcachectl restartadm` command.

2.4 Mismatched Oracle Home Definitions Causes Web Cache to Fail to Start

If the definition of Oracle home in the `webcache.xml` configuration file is different than the definition of Oracle home in your environment, Oracle9*i*AS Web Cache may fail to start.

For example on UNIX, if $ORACLE_HOME was defined as /home/oracle_home_ias during the installation, that definition is written to the ORACLEHOME attribute in the webcache.xml file. Then, if your environment defines $ORACLE_HOME as /private/oracle_home_ias and you invoke the webcachectl executable to start Oracle9*i*AS Web Cache, Oracle9*i*AS Web Cache will fail to start.

In this case, you may see a message similar to the following:

```
Error:  No matching CACHE element found in webcache.xml for current host
name (webcache-host) and ORACLE_HOME (/private/oracle_home_ias). Admin
Server failed to start.
```

You can remedy this situation by either redefining $ORACLE_HOME in your environment or editing the webcache.xml file so that the definitions are identical. (In the webcache.xml file, you modify the ORACLEHOME attribute of the CACHE NAME element. In a cluster environment, there is more than one CACHE NAME element, one for each cluster member. Be sure to modify the correct element.)

[Reference Bug: 2286732]

2.5 Hierarchical Caching

A cache hierarchy whereby one Oracle9*i*AS Web Cache server acts as an origin server to another Oracle9*i*AS Web Cache server is not enabled by default.

To configure this advanced feature:

1. Use a text editor to open the internal.xml file.
2. Locate the <INVALIDATION/> element and add attribute ENABLEOUTBOUNDICC="YES" as follows:

   ```
   <INVALIDATION ENABLEOUTBOUNDICC="YES" />
   ```

3. Perform the procedure that applies to your configuration:
   - Distributed Deployments with Local and Remote Caches
   - Deployments with Subscriber and Provider Caches

2.5.1 Distributed Deployments with Local and Remote Caches

In a distributed deployment with a local cache and remote caches, the local cache stores content from application Web server, and the remote caches store content from the local cache. In other words, the local cache acts as an origin server to the remote caches.

To configure a distributed deployment with local and remote caches, perform the tasks in "Setting Up Oracle9*i*AS Web Cache" of Chapter 6, "Initial Setup and Configuration," in the *Oracle9iAS Web Cache Administration and Deployment Guide* for each cache. When performing the tasks, take special care to perform the following:

1. Configure the correct origin server:
   - For the local cache, configure the local origin server in the Application Web Servers page or Proxy Servers page (**General Configuration** > **Application Web Servers** or **Proxy Servers**).
   - For the remote caches, configure the local cache as the origin server in the Application Web Servers page.
2. Create the same site definition for both the local and remote caches in the Site Definitions page (**General Configuration** > **Sites**).
3. For both local and remote caches, map the site definition to the origin server (configured in Step 1) in the Site to Server Mapping page (**General Configuration** > **Site to Server Mapping**)
   - For the local cache, map the site to the application Web server or proxy server.
   - For the remote cache, map the site to the local cache.

When content from the local cache becomes invalid, an invalidation message is sent to its cache. In addition, the local cache propagates the invalidation message to the remote caches.

Note: In order for automatic propagation of invalidation messages to work, Oracle9*i*AS Web Cache passes the encoded `invalidator` password in the page request between the remote and local cache. This HTTP traffic is susceptible to network sniffing. If the network is unprotected and insecure, configure HTTPS ports as follows:

1. In the Listening Ports page (**Cache-Specific Configuration** > **Listening Ports**) of the local cache, disable the default HTTP port. An HTTPS port is already configured by default.
2. In the Operations page (**Cache-Specific Configuration** > **Operations Ports**) of the remote cache, disable the default HTTP port and configure an HTTPS port in its place.

Table 1 shows the example settings for local cache `webcache1-host` and remote cache `webcache2-host`.

Table 1 Settings for webcache1 and webcache2

Setting Location in Oracle9*i*AS Web Cache Manager	Local Cache webcache1-host	Remote Cache webcache2-host
Listening Port (**Cache-Specific Configuration** > **Listening Ports**)	`7777`	`7777`
Application Web Server (**General Configuration** > **Application Web Servers** or **Proxy Servers**)	`server-host` Port Number: `7777`	`webcache1-host` Port Number: `7777`
Site Definition (**General Configuration** > **Sites**)	Host Name: `www.company.com` Port Number: `80`	Host Name: `www.company.com` Port Number: `80`
Site-to-Server Mapping (**General Configuration** > **Site to Server Mapping**)	Site: `www.company.com:80` Origin Sever: `server-host:7777`	Site: `www.company.com:80` Origin Sever: `webcache1-host:7777`

2.5.2 Deployments with Subscriber and Provider Caches

In a deployment with subscriber and provider caches:

- The subscriber cache stores content from the origin servers for a local site and contacts provider caches for Edge Side Includes (ESI) assembly of content from other sites.
- The provider caches store content from the ESI provider sites.

For these deployments, the provider caches act as origin servers to the subscriber cache.

To configure an ESI deployment with subscriber and provider caches, perform the tasks in "Setting Up Oracle9*i*AS Web Cache" of Chapter 6, "Initial Setup and Configuration," in the *Oracle9iAS Web Cache Administration and Deployment Guide* for each cache. When performing the tasks, take special care to perform the following:

1. Configure the correct origin server:
   - For each provider cache, configure the origin servers of the ESI provider site in the Application Web Servers page or Proxy Servers page (**General Configuration** > **Application Web Servers** or **Proxy Servers**)
   - For the subscriber cache, configure the origin servers of the local site and the provider caches in the Application Web Servers page.
2. Create site definitions:
   - For each provider cache, create a site definition for the ESI provider site in the Site Definitions page (**General Configuration** > **Sites**).
   - For the subscriber cache, create site definitions for the local site and each ESI provider site in the Site Definitions page.

   Note: It may not be possible to specify a site definition for all external ESI provider sites. If an ESI request is made to a provider that does not match any application Web server mapping, then Oracle9*i*AS Web Cache uses Domain Name System (DNS) to resolve the site name. Note that this will not work if there is a firewall between the cache and the ESI provider. In that case, you must provide a proxy server mapping that directs the request to the appropriate proxy.

3. For both subscriber and provider caches, map the site definition to the origin server (configured in Step 1) in the Site to Server Mapping page (**General Configuration** > **Site to Server Mapping**)
   - For the provider cache, map the site definition to the origin server of the ESI provider site.
   - For the subscriber cache, map the local site definition to the origin server for that site, and map each ESI provider site definition to its respective provider cache

When content from the provider cache becomes invalid, an invalidation message is sent to its cache. In addition, the provider cache propagates the invalidation message to the subscriber cache.

Note: In order for automatic propagation of invalidation messages to work, Oracle9*i*AS Web Cache passes the encoded `invalidator` password in the page request between the subscriber and provider cache. This HTTP traffic is susceptible to network sniffing. If the network is unprotected and insecure, configure HTTPS ports as follows:

1. In the Listening Ports page (**Cache-Specific Configuration** > **Listening Ports**) of the provider cache, disable the default HTTP port. An HTTPS port is already configured by default.
2. In the Operations page (**Cache-Specific Configuration** > **Operations Ports**) of the subscriber cache, disable the default HTTP port and configure an HTTPS port in its place.

Table 2 shows the example settings for subscriber cache `webcache-host` and provider cache `webcache-providerhost`.

Table 2 Settings for webcache1 and webcache2

Setting Location in Oracle9*i*AS Web Cache Manager	Subscriber Cache webcache-host	Provider Cache webcache-providerhost
Listening Port (**Cache-Specific Configuration** > **Listening Ports**)	`7777`	`7777`

Application Web Server (**General Configuration** > **Application Web Servers** or **Proxy Servers**)	`server-host`
	`webcache-providerhost`
	Port Number: `7777`
		`webcache-providerhost`
		Port Number: `80`

Site Definition (**General Configuration** > **Sites**)	Host Name: `www.company.com` `www.providersite.com` Port Numbers: `80`	Host Name: `www.providersite.com` Port Number: `80`

Table 2 (Cont.) Settings for webcache1 and webcache2

Setting Location in Oracle9*i*AS Web Cache Manager	Subscriber Cache webcache-host	Provider Cache webcache-providerhost
Site-to-Server Mapping (**General Configuration** > **Site to Server Mapping**)	Site: `www.company.com:80` Origin Server: `server-host:7777` Site Name: `www.providersite.com:80` Origin Server: `webcache1-host:7777`	Site: `www.providersite.com:80` Origin Sever: `webcache1-host:7777`

2.6 Invalidation Time-outs

Invalidation has a default time-out of 300 seconds for the propagation of messages in the following deployments:

- Cache hierarchy whereby one Oracle9*i*AS Web Cache server acts as an origin server to another Oracle9*i*AS Web Cache server
- Cache cluster with multiple Oracle9*i*AS Web Cache servers

To change the default settings, modify the `webcache.xml` file:

1. Use a text editor to open the `webcache.xml` file.
2. Locate the `<CALYPSONETINFO...>` line:

   ```
   <CALYPSONETINFO INV_PEER_TIMEOUT="300"
                   INV_GLOBAL_TIMEOUT="300"/>
   ```

3. Change the value of `INV_PEER_TIMEOUT` to modify the time-out for invalidation messages sent in a cache hierarchy.
4. Change the value of `INV_PEER_TIMEOUT` to modify the time-out for invalidation messages sent in a cache cluster.

2.7 Directives to Oracle HTTP Server

At installation time, Oracle HTTP Server sets the `httpd.conf` file with the following directives that impact Oracle9*i*AS Web Cache:

- `Port=`*`web_cache_port`* specifies the Oracle9*i*AS Web Cache listening ports, enabling dynamically created URLs to be redirected to Oracle9*i*AS Web Cache
- `Listen=`*`Oracle_HTTP_Server_port`* specifies the HTTP and HTTPS ports obtained by Oracle HTTP Server
- `ServerName` specifies the host name of Oracle HTTP Server

- `UseCanonicalName On` instructs Oracle HTTP Server to use the host names and port values set in the `ServerName` and `Port` directives when redirecting a URL

For example:

```
##
## httpd.conf -- Apache HTTP server configuration file
##
...
Port 7777
Listen 7778
...
ServerName http_server.company.com
...
UseCanonicalName On
....
```

If you decide to disable Oracle9*i*AS Web Cache, then the Oracle HTTP Server administrator must modify the value of the `Port` directive to the same value set for the `Listen` directive.

For example:

```
##
## httpd.conf -- Apache HTTP server configuration file
##
...
Port 7778
Listen 7778
...
ServerName http_server.company.com
...
UseCanonicalName On
....
```

If Oracle9*i*AS Web Cache is deployed on a separate machine from the Oracle HTTP Server, then the Oracle HTTP Server administrator must modify the `ServerName` directive in `httpd.conf` for each site hosted by Oracle9*i*AS Web Cache. This will enable Oracle HTTP Server to redirect URLs to Oracle9*i*AS Web Cache. The following example shows `httpd.conf` modified to direct requests for `www.1st.company.com` and `www.2nd.company.com` to Oracle9*i*AS Web Cache, which listening on port 7777.

```
Port 7777
Listen 7778
...
ServerName  www.1st.company.com
ServerName  www.2nd.company.com
```

```
...
Usecanonicalname On
....
```

The `httpd.conf` file resides in `$ORACLE_HOME/Apache/Apache/conf/httpd.conf` on UNIX or *`ORACLE_HOME`*`\Apache\Apache\conf\httpd.conf` on Windows.

> **See Also:** *Oracle HTTP Server Administration Guide*

2.8 ESI Support

2.8.1 JESI Support

The Oracle proprietary language elements described in Appendix D, "Edge Sides Includes Languages," of the *Oracle9iAS Web Cache Administration and Deployment Guide* are not supported by the Edge Sides Includes for Java (JESI) in this release. Oracle Corporation plans to support these language elements in a future implementation.

> **See Also:** *Oracle9iAS Containers for J2EE JSP Tag Libraries and Utilities Reference* for a description of the supported JESI tags

2.8.2 Protocol Support

Oracle9*i*AS Web Cache supports HTTP and HTTPS in the `src` attribute of the `<esi:include>` tag. Oracle9*i*AS Web Cache permits the template and fragments to use different protocols. Take note of the following:

- If the `src` attribute specifies a fragment's relative path, such as `src="/PersonalizedGreeting"`, then the template's protocol is used.
- If the protocol used in the `src` attribute does not match the protocol specified in the Site to Server Mapping page (**General Configuration** > **Site to Server Mapping**), then Oracle9*i*AS Web Cache uses the protocol configured for the origin server in the Site to Server Mapping page. Oracle9*i*AS Web Cache also reports the following warning message to the event log:

  ```
  Date Warning: ESI Include protocol does not match Origin Server
  protocol: Origin Server Protocol=protocol URL=URL
  ```

  For example, if the template page is configured with `<esi:include> src="https://www.company.com/gifs/frag1.gif"/>` and the Site to Server mapping specifies HTTP for the origin server, then `http://www.company.com/gifs/frag1.gif` is used and the following message appears in the event log:

```
11/Jan/2002:19:25:59 +0000 Warning: ESI Include protocol does not
match Origin Server protocol: Origin Server Protocol=http
URL=https://www.company.com/gifs/frag1.gif
```

2.8.3 Nesting with Inline Tags

When a non-fetchable `<esi:inline>` fragment is not found in the cache, Oracle9*i*AS Web Cache re-fetches the fragment's parent template. This behavior implies that the parent cannot be another non-fetchable `<esi:inline>` fragment. If the parent is an `<esi:inline>` non-fetchable fragment, then the response returned to the browser is undefined.

2.9 +wcdebug Limitations

Note the following limitations when appending the string `+wcdebug` to the URL of the document to see the diagnostic information string embedded in the response body:

- In a cache cluster, if a cache member receives a request with `+wcdebug` for a content owned by another cache member, the page with debug information may be stored in the cache as an on-demand cached page. During a subsequent request for the same page without `+wcdebug`, you will retrieve the on-demand cached page with the debug information.

  [Reference Bug: 2142446]

- Appending debug information to compressed page causes an error in the browser. If you access the page with `telnet`, you will see the debug information prepended to the compressed page.

  [Reference Bug: 2133691]

> **See Also:** Chapter 10, "Troubleshooting Oracle9*i*AS Web Cache Configuration" of the *Oracle9iAS Web Cache Administration and Deployment Guide* for further information about using `+wcdebug`

2.10 Load Balancing Changes

Most Web sites are served by multiple origin servers running on multiple computers that share the load of HTTP and HTTPS requests. All requests that Oracle9*i*AS Web Cache cannot serve are passed to the origin servers. Oracle9*i*AS Web Cache balances the load among origin servers by determining the percentage of the available capacity, the **weighted available capacity** of each origin server. Oracle9*i*AS Web Cache sends a request to the origin server with the highest weighted available capacity. The weighted available capacity is determined by the following formula:

```
(Capacity - Load) / Capacity
```

where:

- `Capacity` is the maximum number of concurrent connections that the origin server can accept
- `Load` is the number of connections currently in use

If the weighted available capacity is equal for multiple origin servers, then Oracle9*i*AS Web Cache sends requests to the origin servers in round-robin fashion. With round-robin, the first origin server in the list of configured servers receives the request, then the second origin server receives the second request.

If the weighted available capacity is not equal, Oracle9*i*AS Web Cache sends the request to the origin server with the highest weighted available capacity.

Consider the following cases:

- Assume Oracle9*i*AS Web Cache is balancing the load between two origin servers, `company1-host` and `company2-host`, with capacities of 50 each.

  The requests to company1-host and company2-host will be distributed between the two origin servers so that they maintain an equal load. The first request is sent to `company1-host`. The second request is sent to `company2-host` if `company1-host` is still processing the first request. The third and subsequent requests are sent to the origin server that has the highest weighted available capacity.

- Assume Oracle9*i*AS Web Cache is balancing the load between three origin servers, `server1-host`, `server2-host`, and `server3-host` with capacities of 150, 50, and 50, respectively.

  The first request is sent to `server1-host`.

  The second request is sent to `server2-host`, because `server1-host` now has a weighted available capacity of 99.3 percent and `server2-host` has a weighted available capacity of 100 percent.

  The third request is sent to `server3-host` because `server2-host` now has a weighted available capacity of 98 percent and `server3-host` has a weighted available capacity of 100 percent.

  The fourth request is sent to `server1-host` because `server2-host` and `server3-host` now have weighted available capacities of 98 percent.

The fifth request is sent to server1-host because its weighted available capacity is 98.6 percent, still greater than that of server2-host and server3-host.

To configure load balancing, set the capacity of each origin server.

2.11 Session Binding

If an origin server is busy, then Oracle9*i*AS Web Cache disables session binding to that origin server.

[Reference Bug: 2180999]

2.12 Concurrent Users on Windows NT Workstation

Oracle9*i*AS Web Cache permits only five concurrent users on Windows NT Workstation. If you require more than five concurrent users, then consider using a Windows NT Server, such as Windows NT or Windows 2000.

2.13 Browser Limitations

Table 3 describes browser limitations and their impact on Oracle9*i*AS Web Cache.

Table 3 Browser Issues

Problem	Description
Compressing JavaScript Files	**Problem Description:** Compressed JavaScript files cause some Netscape browsers to behave erratically and possibly fail. This issue only effects files that are referenced with the `src` attribute of the `script` tag; it does not include files that contain inline JavaScripts. **Known browsers effected:** Netscape 4.*x* **Example:** `<script language="JavaScript" src="copyright.js"></script>` If `copyright.js` is compressed for Netscape, then the browser may fail. **Workaround:** By default, compression is turned off for included JavaScript files. View the Cacheability Rules page to see this setting (**General Configuration > Cacheability Rules**).

Table 3 (Cont.) Browser Issues

Problem	Description

Compressing Documents with `Content-Disposition` Response-Header Fields	**Problem Description:** Documents with `Content-Disposition` response-header fields show incorrect file names when they are compressed.
	Known browsers effected: Internet Explorer 5.0, 5.5, and 6.0
	Example: Response headers for URL `/reportgen` include to following:
	`Content-Type: application/excel`
	`Content-Disposition: attachment; filename="file.csv"`
	When the document is not compressed, a **Save As** dialog appears with `file.csv` as the default filename. However, if it is compressed, `reportgen` appears as the default name. Without the correct extension, the file will not open correctly on Windows operating systems.
	Workaround: Even if compression is selected, Oracle9*i*AS Web Cache does not compress documents containing a `Content-Disposition` response-header field.

Decompressing Documents with `Content-Disposition` Response-Header Fields	**Description:** Documents with `Content-Disposition` response-header fields are not decompressed when you choose **File** > **Save As** from Netscape browsers. **Known browsers effected:** Netscape 4.*x* **Workaround:** Even if compression is selected, Oracle9*i*AS Web Cache does not compress documents containing a `Content-Disposition` response-header field.
Compressing Style Sheets	**Description:** Compressed style sheets can cause background attributes, such as background images, to not appear in the output. **Known browsers effected:** Internet Explorer 6.0 **Workaround:** Disable compression for style sheets in the Cacheability Rules page (**General Configuration > Cacheability Rules**).

2.14 Cache Cluster Event Logs

Table 4 describes event logs related to cache clusters. This information supplements information in Appendix E, "Event Log Messages," of the *Oracle9iAS Web Cache Administration and Deployment Guide*.

Table 4 Cluster Event Logs

Message	Description
Information Events	
`A` *`number`* `node cluster successfully initialized`	Oracle9*i*AS Web Cache started the cache cluster with the specified number of cache cluster members.
Warning Events	
`Add Cluster member#` *`cluster_ID`*	Oracle9*i*AS Web Cache added the cache cluster member to the cache cluster.
`Cluster Member sent invalid configuration error`	A bad configuration response header was returned from a request. However, this request was not a Oracle9*i*AS Web Cache cluster peer request.
`Cluster member#` *`cluster_ID`* `already marked alive`	An attempt to mark the cache cluster member as active and running has been made. However, a previous concurrent request has already noted the member as being active and running.
`Cluster member#` *`cluster_ID`* `already marked dead`	An attempt to mark the cache cluster member as being inactive or down has been made. However, a previous concurrent request has already noted the member as being inactive or down.
`Cluster Member` *`cluster_ID`* `down. Start pinging.`	Oracle9*i*AS Web Cache detects a cache cluster member is down, and it starts to poll the member by sending requests a configured URL.
`Cluster Member` *`cluster_ID`* `up. Stop ping.`	Oracle9*i*AS Web Cache received a success response from the downed cache cluster member. Oracle9*i*AS Web Cache considers the cache cluster member to up again. It recalculates the relative capacity of the cache cluster members, and it reassigns ownership of cache content.
`Cluster Member` *`cluster_ID`* `-` *`cluster_name`* `configuration is invalid.`	The cache cluster member sent a message to Oracle9*i*AS Web Cache that indicates it has a configuration file different than the local configuration file for the cache cluster. A cluster runtime operates with the same `webcache.xml` configuration file for all members. To resolve this error, compare the two configuration files, and then propagate the proper version to all cluster members.

Table 4 (Cont.) Cluster Event Logs

Message	Description
`Cluster Member` *`cluster_ID`* `-` *`cluster_name`* `configuration is valid.`	The cache cluster member was previously excluded from the cluster due to a configuration file that was different than the local view. The configuration files between these two members are now consistent.
`Cluster Member` *`cluster_ID`* `-` *`cluster_name`* `sent invalid configuration error.`	The cache cluster member sent a response to Oracle9*i*AS Web Cache that indicates it has a configuration file different than the local configuration file for the cache cluster. A cluster runtime operates with the same `webcache.xml` configuration file for all members. To resolve this error, compare the two configuration files, and then propagate the proper version to all cluster members.
`Remove Cluster member#` *`cluster_ID`*	Oracle9*i*AS Web Cache removed the cache cluster member from the cache cluster.
`Unknown Cluster Member sent invalid configuration error.`	The cache cluster member sent a response to Oracle9*i*AS Web Cache that indicates it has a configuration file different than the local configuration file for the cache cluster. Additionally, this peer member cache is not specified in the local `webcache.xml` configuration file. A cluster runtime operates with the same `webcache.xml` configuration file for all members. To resolve this error, compare the two configuration files, and then propagate the proper version to all cluster members.
Error Events	
`Cache memory allocation for the cluster configuration block`	Oracle9*i*AS Web Cache ran out of memory on the system.
`Cache memory allocation for the cluster member table`	Oracle9*i*AS Web Cache ran out of memory on the system.
`Cluster member count` *`count`* `is larger than maximum allowed 99`	The number of cache cluster members specified in the cache cluster configuration is larger than the maximum limit of 99.

2.15 Peer-to Peer-Logging

By default, peer requests between two members of a cache cluster are not logged in the access log. Only client requests to the cluster are logged. Peer request logging can be enabled for individual cache cluster members by adding the ACCESSLOGIGNOREPEERREQUEST attribute to the MISCELLANEOUS element in the internal.xml configuration file.

The valid values for this attribute are:

- YES
- NO

The default value is YES.

The following example shows the MISCELLANEOUS element with peer-to-peer logging disabled:

```
<MISCELLANEOUS
    ERRORLOGFILE="my_oracle_home/webcache/logs/event_log"
    ACCESSLOGIGNOREPEERREQUEST="NO"/>
```

3 Security Issues and Workarounds

This section describes security limitations for Oracle9*i*AS Web Cache. It contains the following sections:

- Section 3.1, "HTTP Authentication"
- Section 3.2, "Single Sign-On"
- Section 3.3, "Set-User ID Permission for webcachectl"
- Section 3.4, "User Certificates"

3.1 HTTP Authentication

Oracle9*i*AS Web Cache does not cache pages that support basic HTTP authentication. These pages result in cache misses.

3.2 Single Sign-On

Oracle9*i*AS Web Cache does not cache login requests or authenticated pages that use mod_sso static directives. To ensure that responses for those pages using dynamic directives with mod_sso are not cached, add the Surrogate-Control: no-store response-header field to identify the page as non-cacheable.

When configured with HTTPS listening ports, Oracle9*i*AS Web Cache is unable to forward browser certificates to origin servers. If browsers are using certificate-based single sign-on authentication, do not use Oracle9*i*AS Web Cache. This restriction will be lifted in a future release of Oracle9*i*AS Web Cache.

3.3 Set-User ID Permission for webcachectl

If the `root.sh` script was run, then the `webcachectl` executable has a set-user ID of `root`. As a result, the user that installs Oracle9*i*AS Web Cache can gain root privileges. To restrict root privileges, remove set-user ID `root` from the `webcachectl` executable. Note that set-user ID `root` is required in the following cases:

- Privileged port numbers less than 1024 are being used for Oracle9*i*AS Web Cache listening ports
- There are more than 1,024 file descriptors being used for connections to Oracle9*i*AS Web Cache.
- The current `webcachectl` user does not match the configured user in the Process Identity page (**Cache-Specific Configuration** > **Process Identity**) of Oracle9*i*AS Web Cache Manager

Oracle Corporation plans to resolve this issue in the next release.

[Reference Bug: 2302587]

3.4 User Certificates

If a wallet contains a user certificate as a trustpoint for an origin server, then a core dump will occur when the user connects to the origin server. Oracle Corporation recommends not adding user certificates to trustpoints in the Oracle wallet but instead install the certificate authority (CA) signers' certificate as a trustpoint.

Oracle Corporation plans to resolve this issue in the next release.

[Reference Bugs: 2295542 and 2295884]

4 Documentation Errata

This section describes known errors and omission in the documentation for Oracle9*i*AS Web Cache. It contains the following sections:

- Section 4.1, "Process Identity Page"
- Section 4.2, "cs(request_header) and sc(response_header) Access Log Fields"
- Section 4.3, "Additional Error Message"

4.1 Process Identity Page

The documentation for the Process Identity page (**Cache-Specific Configuration** > **Process Identity**) of Oracle9*i*AS Web Cache Manager describes how to change the user ID and group ID of the Oracle9*i*AS Web Cache executables. In addition to using the Process Identity page, you must also manually change the ownership of the following files and directories to the new user ID and group ID with the `chown` command:

- `$ORACLE_HOME/webcache`
- `$ORACLE_HOME/webcache/internal.xml`
- `$ORACLE_HOME/webcache/internal_admin.xml`
- `$ORACLE_HOME/webcache/webcache.xml`
- `$ORACLE_HOME/webcache/logs/event_log`
- `$ORACLE_HOME/webcache/logs/access_log`

4.2 cs(*request_header*) and sc(*response_header*) Access Log Fields

The documentation for the `cs(`*`request_header`*`)` and `sc(`*`response_header`*`)` fields for access log does not provide sufficient examples of the request and response headers that can be used for these user-specified fields.

Table 5 lists examples of HTTP/1.1 headers that can be used for the `cs(`*`request_header`*`)` and `sc(`*`response_header`*`)` fields. This table lists only some of the possible headers. It is not an exhaustive list.

Table 5 Examples of HTTP/1.1 Header Fields

cs(*request_header*) Field	sc(*response_header*) Field
`Accept`	`Server`
`Authorization`	`Cache-Control`

Table 5 (Cont.) Examples of HTTP/1.1 Header Fields

cs(*request_header*) Field	sc(*response_header*) Field
Connection	Content-Encoding
Host	Content-Language
Referer	Content-Length
User-Agent	Content-Type
Cache-Control	Date
Content-Encoding	ETag
Content-Language	Expires
Content-Length	Last-Modified
Content-Type	Pragma
If-None-Match	Transfer-Encoding
If-Modified-Since	Via
Last-Modified	
Pragma	
Range	
TE	
Via	
Date	

Table 6 lists examples of cookie-related headers that can be used for the `cs(`*`request_header`*`)` and `sc(`*`response_header`*`)` fields.

Table 6 Supported Cookie-Related Header Fields

cs(*request_header*) Field	sc(*response_header*) Field
Cookie	Set-Cookie

Table 7 lists examples of Oracle9*i*AS Web Cache headers that can be used for the `cs(`*`request_header`*`)` and `sc(`*`response_header`*`)` fields.

Table 7 Supported Oracle9iAS Web Cache Header Fields

cs(*request_header*) Field	sc(*response_header*) Field
Surrogate-Capability	Surrogate-Control

See Also: Chapter 8, "Administering Oracle9*i*AS Web Cache," in the *Oracle9iAS Web Cache Administration and Deployment Guide* for information about user-specified access log fields

4.3 Additional Error Message

The following event log error message has been added to Oracle9*i*AS Web Cache:

```
Process out of memory on malloc/realloc request. Exiting process.
```

This error indicates that insufficient virtual memory remains in the `cache` server process to satisfy the request. The error can be caused by either of the following situations:

- The request is for an inordinate amount of memory, thus causing the system limit for virtual memory to be exceeded.
- The cache's memory is extremely full and this request puts the `cache` server process over the system limit.

When this error occurs, the `cache` server process is stopped. If auto-restart is enabled, then the `auto-restart` process automatically restarts the `cache` server process. If auto-restart is not enabled, then restart the `cache` server process from the Operations page (**Administration** > **Operations**).