

# Oracle9i Application Server: Options for Running Active Server Pages

*An Oracle White Paper  
July 2001*

# Oracle9i Application Server: Options for Running Active Server Pages

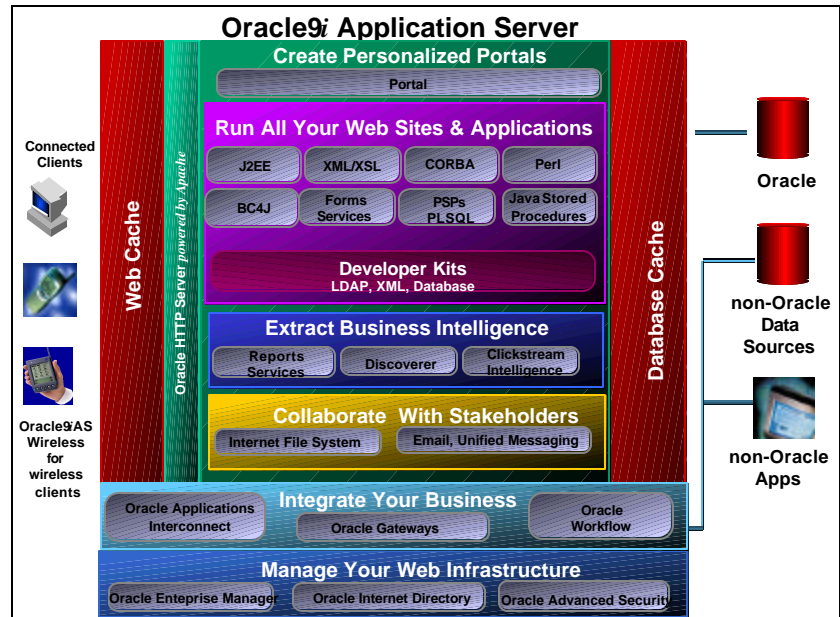
PROBLEM SUMMARY .....	3
INTRODUCTION.....	3
Other Technologies .....	4
OHS Extensibility with Modules.....	4
THE SOLUTION OPTIONS.....	5
Option #1: Expose both IIS and Oracle HTTP Server to Browsers ..	5
Option #2: Expose only IIS to the end user Browsers .....	6
Option #3: Expose Only Oracle 9iAS to Browsers .....	7
Note that the above RewriteRule works adequately in simple situations but has not been tested in all scenarios. It is meant to serve as an example. For production deployment you will need to ensure the regular expression meets your needs or modify it appropriately...	8
Option #4: Not having IIS in the picture at all .....	8
Option #5: Not exposing either IIS or 9iAS - Use Other Proxy Servers .....	8
SUMMARY .....	8

## PROBLEM SUMMARY

Active Server Pages has been a popular way of building dynamic web applications on Microsoft technology stack. This paper describes how to preserve existing investment in that technology, while moving forward to the new J2EE technology implemented by Oracle 9iAS.

## INTRODUCTION

The picture in Figure 1 provides an overview of Oracle9iAS. We will pin point the



**Figure 1: Oracle 9iAS Components**

components of interest to this paper, and will not cover all the components of Oracle 9iAS.

1. Oracle HTTP Server (OHS) contains a web server (powered by Apache).
2. mod\_proxy is a plug-in contained in OHS that enables OHS to act as a caching reverse proxy server.
3. Oracle Proxy Plugin (not available now; part of 9iAS v2.0 plan) contained in OHS can be installed on Microsoft IIS or Netscape's iPlanet Server, and it makes these two web servers behave (in a limited fashion) as a reverse proxy server to OHS.
4. mod\_rewrite is a plug-in contained in OHS that can rewrite (modify / change) the URLs. Regular expression based rules for URL manipulation can be defined.

5. OC4J (Oracle Containers For Java) implements the J2EE specification. The servlets, JSP etc. all run within this component. It also contains an HTTP listener, which is leveraged in some of the deployment choices in this paper.

### Other Technologies

Active Server Pages - Similar to JSP (Java Server Pages). These pages enable dynamic web applications by adding scripting (via VBScript) to an HTML page.

Microsoft's IIS - Internet Information Server - provides the container to run Active Server Pages.

Microsoft VBScript - The scripting language for Active Server Pages. Very similar to Visual Basic with some limitations.

Microsoft COM - COM (Component Object Model) is a popular way to package framework / objects and reuse them across applications. In the context of our current problem, these are external to the active server pages, and are invoked by VB script within the ASP page.

### OHS Extensibility with Modules

This brief overview of module based architecture/behavior will be useful when we talk to the different plug-ins and how to leverage them for running active server pages with Oracle 9iAS.

The architecture of the Oracle HTTP Server is extremely modular. The core web server [for http protocol] is very small; all capabilities are implemented as modules that 'plug in,' and are invoked at the appropriate place during the request lifecycle.

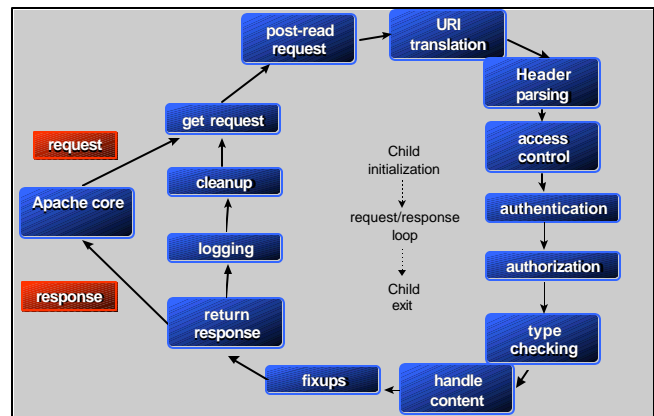


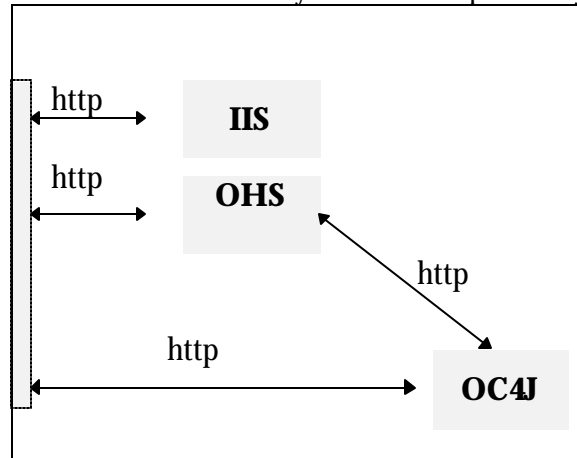
Figure 2: HTTP Request-Response Cycle

The lifecycle of an HTTP request is depicted in the adjacent figure. A web server process shepherds the request through this entire lifecycle. The 'plug-in', i.e. modules, register their API. This API is then either automatically invoked when the request reaches a certain stage in its lifecycle, or, can be configured to be invoked only in certain situations.

## THE SOLUTION OPTIONS

### Option #1: Expose both IIS and Oracle HTTP Server to Browsers

This scenario is to essentially have two independent systems - one powered by IIS,



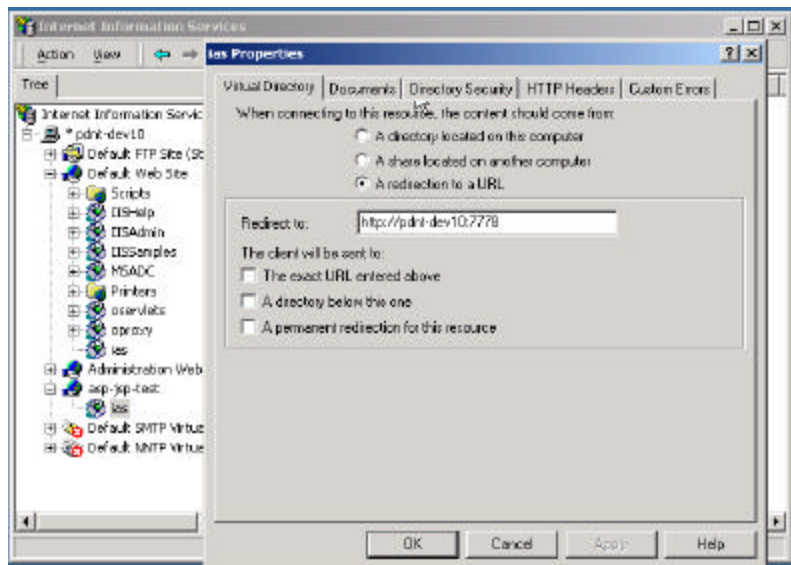
the other powered by Oracle 9iAS, with no integration between the two.

IIS and Oracle HTTP Server both serve HTTP requests by listening for browser requests at different ports.

Only one of these servers could be listening on the default port (80). Thus

users need to remember the port in addition to server to go to the (correct) default page for the different servers.

A workaround to having to remember the port number is to create a virtual directory in one of the servers - say IIS. All requests to this virtual directory can be configured to get redirected to go to the correct port (or same port on a different machine) on which the other server is running. The configuration screen for this setup for IIS is shown below.



A similar configuration for 9iAS is also possible. In httpd.conf include the following lines to redirect all requests to the /iis virtual directory to the correct port for the IIS machine.

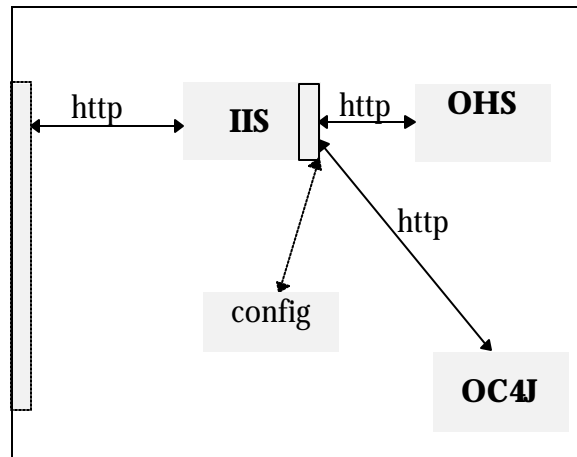
Redirect /iis	http://iis-machine:88/
---------------	------------------------

Most organizations standardize on a web server and ensure all security patches etc. for it are current. Thus, this option of exposing multiple web servers to the end-users, may not be suitable in many scenarios.

### Option #2: Expose only IIS to the end user Browsers

To overcome the shortcomings of the previous section, Oracle 9iAS (from v2.0 onwards) plans to include the Oracle 9iAS Proxy Plug-in. It, as the name implies, “plugs-in” into IIS and determines which requests need to be “proxied” to 9iAS and forwards them accordingly. The rest of the requests are served natively by IIS.

There are two relevant configuration files in this context - one that defines the host and port on which iAS is running (could be on the same machine), and another that defines which types of URL (ex. http://server/ias/\* or ones ending



in \*.jsp) need to be forwarded to which iAS instance.

Thus, depending on configuration, all .asp requests are now serviced by the local IIS, and all j2ee (or other Oracle 9iAS service requests) are proxied to the correct Oracle 9iAS installation. Except for a small performance hit for

the additional reverse proxy hop, the end-user has no indication about this proxying happening at the back-end. Sample configurations are shown below:

#### **#Define the servers**

```
oproxy.serverlist=ias1, ias2
#define where these servers list
oproxy.ias1.hostname=www1.us.oracle.com
oproxy.ias1.port=7777
...
```

#### **#Map all URLs** with /servlet/ to point to the right server

```
/servlet/*=ias1
...
```

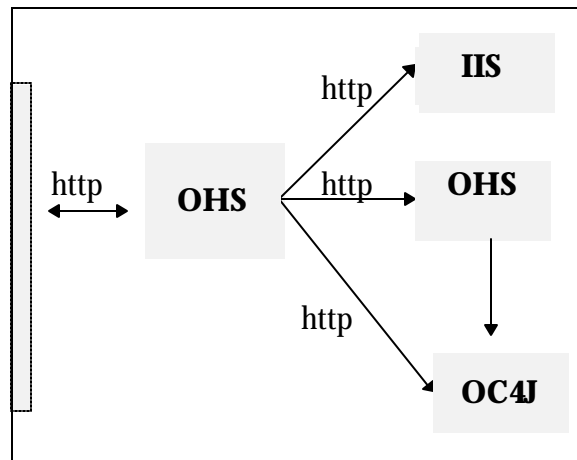
Thus the corporate requirement of standardizing on a single web server (in this case IIS) is also met, while all J2EE applications are serviced by an instance of

iAS. Another advantage is that the end-users do not have to remember different port numbers and hosts. Yet another benefit is that it allows heterogeneous hardware (IIS on NT, iAS on UNIX for example) to work together to service an IIS based site.

The SSL requests terminate at the first web server - in this case IIS. All future communication is non-SSL. The authentication and single sign on solution in this scenario is complex and will be discussed in a future version of this paper. Currently, Oracle Proxy Plug-In is not in production and some of these details have not been finalized.

### Option #3: Expose Only Oracle 9iAS to Browsers

This option is the “inside-out” version of the previous option - It makes Oracle HTTP Server as the front-end web server that receives all the requests and IIS as the back-end server servicing .asp requests.



As stated earlier OHS includes two powerful modules:

- `mod_rewrite` which changes the URL to be served
- `mod_proxy`, which proxies the request to another server.

In addition `mod_rewrite` provides several options on how the rewritten URL should be serviced (including servicing via `mod_proxy` with a [P] - Proxy Throughput flag.).

In this option the end user's request is first serviced by 9iAS Oracle HTTP Server (OHS). If the URL meets certain complex criteria (ex. ends in \*.asp or has a certain string within it), it is rewritten to be proxied to a different server. For simpler test criteria `mod_proxy` is adequate for reverse proxying requests (via the `ProxyPass` and `ProxyPassReverse` directives). Sample configurations:

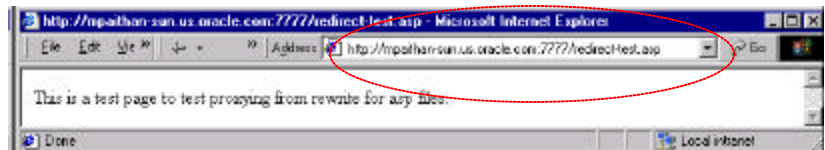
#### #mod\_proxy based configuration

ProxyRequests On

ProxyPass /iis/ http://iis-machine/

ProxyPassReverse /iis/ http://iis-machine

`mod_rewrite` based configuration produces similar results and a screen shot of browser showing the 9iAS URL with contents served from IIS is shown below.



**#mod\_rewrite based configuration**

RewriteEngine on

RewriteLog /tmp/rewrite.log

RewriteLogLevel 3

RewriteRule ^/([^\.])(.asp)(.\*)\$ http://iis-machine/\$1.asp\$3 [P]

Note that the above RewriteRule works adequately in simple situations but has not been tested in all scenarios. It is meant to serve as an example. For production deployment you will need to ensure the regular expression meets your needs or modify it appropriately.

**Option #4: Not having IIS in the picture at all**

There are two scenarios that can enable this:

- Rewriting the application to be pure J2EE so that OC4J can host it - This is of course a longer term option.
- Leverage Oracle Migration Technology. The details can be found at <http://technet.oracle.com/tech/migration/asp/>. This technology helps end customers migrate their Microsoft based applications to J2EE.

**Option #5: Not exposing either IIS or 9iAS - Use Other Proxy Servers**

Another approach, if none of the earlier options are viable in your environment is to use a standalone proxy server from other companies (or an open source one). Most of these products enable partitioning the URL and allowing different name spaces to be serviced by different web servers. Thus, .asp and J2EE applications can appear to co-exist on the same host, even though they may be serviced from different machines internally.

This is not recommended because it adds yet another server in the critical path of request-response.

**SUMMARY**

This paper provides several options on how current Microsoft ASP technology customers can adopt Oracle 9iAS and move towards J2EE compliant applications.

The options include using IIS as the only front-end listener (with Oracle Proxy Plugin installed), or, having both IIS and Oracle HTTP Server listen to browser requests (different ports, virtual directory setup), or, having Oracle HTTP Server as the only front-end listener (with mod\_proxy or mod\_rewrite or the Oracle Migration Toolkit).





Oracle9i Application Server - Options for Running Active Server Pages

July 2001

Author: Mukul Paithane

Contributing Authors:

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

[www.oracle.com](http://www.oracle.com)

Oracle Corporation provides the software  
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various  
product and service names referenced herein may be trademarks  
of Oracle Corporation. All other product and service names  
mentioned may be trademarks of their respective owners.

Copyright © 2000 Oracle Corporation

All rights reserved.