Oracle Discoverer 4i Plus Firewall and SSL Tips

An Oracle White Paper
February 2002

ORACLE

Oracle Discoverer 4i Plus
Firewall and SSL Tips

Oracle Discoverer 4i Plus
Firewall and SSL Tips

INTRODUCTION

This document is intended for a technical audience. It benefits those involved in deploying or troubleshooting an Oracle Discoverer 4i Plus deployment as part of Oracle9i Application Server version 1.0.2.2. This document should be read entirely, from beginning to end.

The same information is provided in the "Oracle9i Discoverer Plus and Viewer Configuration Guide". However, that information covers a wider range of configurations and may be more complicated than necessary unless read carefully. Before its inclusion in Oracle9iAS, the components of Discoverer 4i Plus could be distributed across multiple servers. When installed with Oracle9iAS, many of those configurations no longer apply and the documentation can be simplified and narrowed. This document supplements rather than replaces the Configuration Guide.

BACKGROUND

Pre-requisites to properly use the information in this document:

- Basic computer networking skills (understand protocols, routing, topology, etc.)
- Basic knowledge of network firewalls
- Basic knowledge of Secure Socket Layer (SSL) protocol

To learn more about these topics, readers may try these resources*:

Networking

"Absolute Beginner's Guide to Networking (3rd Edition)" by Joseph Habraken, Mark Gibbs, ISBN 0789725452

"TCP/IP Network Administration, 2nd Edition" by Craig Hunt, ISBN 1-56592-322-7

Firewalls

http://www.interhack.net/pubs/fwfaq/

SSL

http://developer.netscape.com/tech/security/ssl/howitworks.html

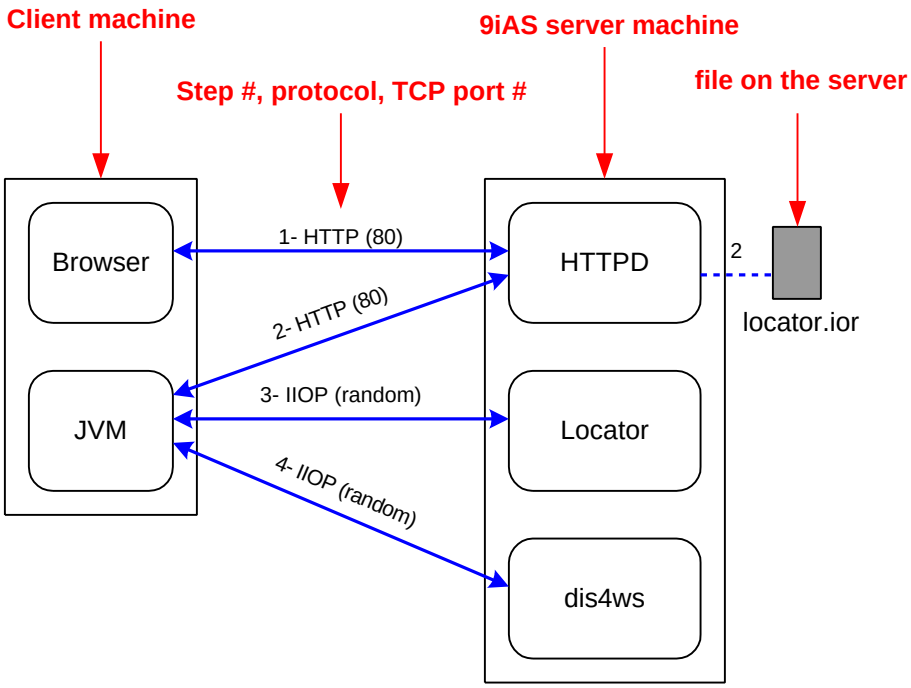
*This list is not an endorsement of these resources or a guarantee of their validity.

SCENARIOS

In all scenarios, use of “Discoverer” or “Plus” refer to Oracle Discoverer 4i Plus as shipped with Oracle9iAS 1.0.2.2.

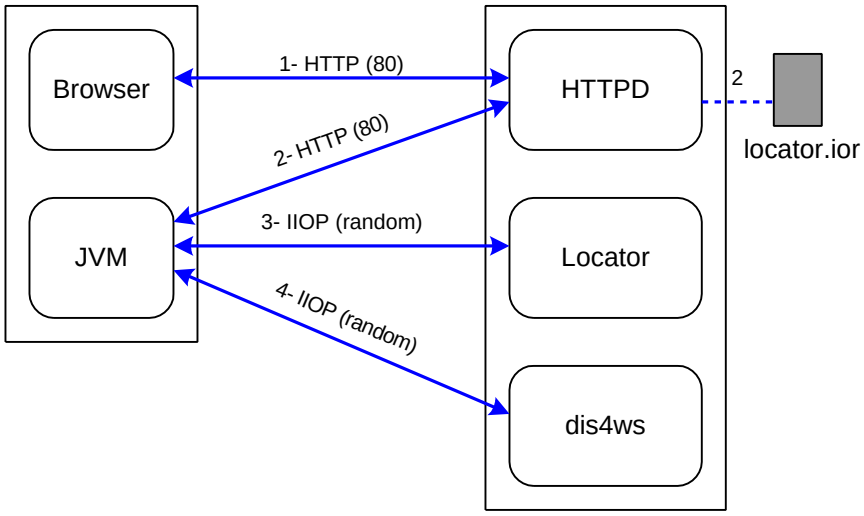
The diagram below explains the diagramming conventions used in all sections. It is largely self-explanatory, except for the following details:

- Solid blue lines show insecure network connections
- Solid green lines show secure network connections using SSL
- Dashed blue lines connect files to the components that use them
- Dashed blue line step #s correspond to the first step that uses that file



Lastly, what is the Gatekeeper? It does not appear in the diagram above, but it does in almost all the other diagrams in this document. The Gatekeeper is a daemon that runs on the Oracle9iAS machine. Discoverer installs it but by default it is not started with the other Discoverer services (OAD, OSAgent, Locator). The Gatekeeper performs a variety of functions and may be thought of as a proxy server running on the Oracle9iAS machine.

Basics of Starting Discoverer



The Discoverer 4i Plus client is a Java applet. The diagram above shows how Plus is launched and subsequently communicates with the Oracle9iAS middle-tier.

1. The browser uses an HTTP connection to retrieve the HTML "start" page for Plus from the HTTP server. The instructions to download the Plus Java applet are embedded in that HTML page.
2. The browser's JVM (JInitiator or the MS JVM) uses HTTP to retrieve the applet files from the HTTP server. Once the applet files are downloaded, the applet is launched. The applet immediately retrieves the locator.ior file from the HTTP server.
3. The applet reads the information in the locator.ior file (IP address and TCP port) and uses IIOP to connect to the Locator. The Locator tells the applet the location of a Discoverer Session (a dis4ws process on the middle-tier).
4. From this point forward, the Plus applet communicates directly with the same dis4ws process on the middle-tier using IIOP. Plus prompts the user for login information.

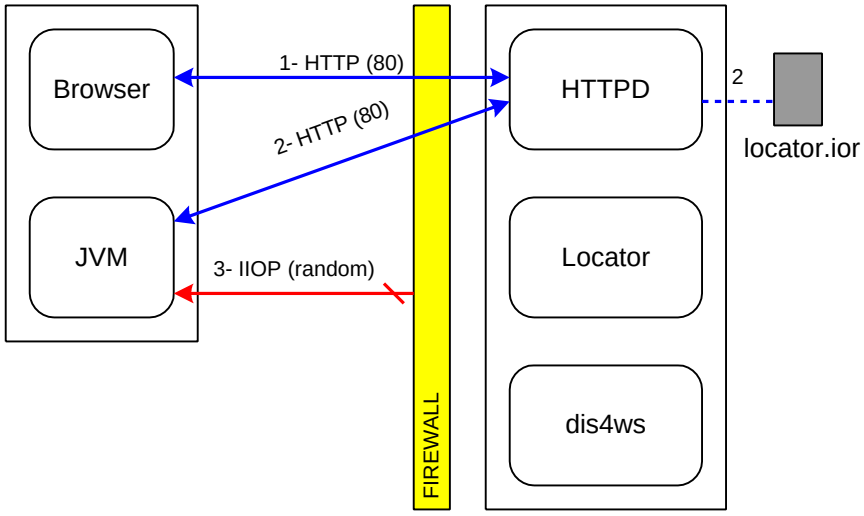
It is crucial to understand that Plus does ***not*** use the browser for any data communication. The Plus applet runs inside a JVM (this may be Oracle JInitiator or Microsoft's JVM for IE) and the only interaction with the browser is the display of the HTML page that launches the Plus applet.

Also note that IIOP does not restrict use to a specific TCP port for connections. This is indicated by the "(random)" TCP port value in the diagram.

Firewalls

Most firewalls do not explicitly support IIOP. They may allow it to go through if the firewall administrator configures it to allow all TCP/IP traffic to the Oracle9iAS machine, but few firewalls recognize IIOP specifically.

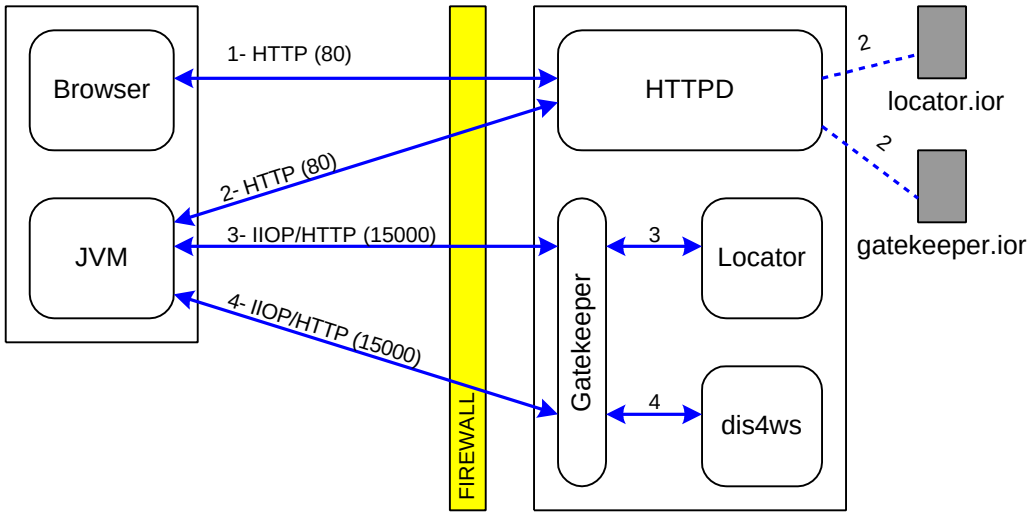
Discoverer Problems Due to Firewalls



The diagram above shows a firewall placed between the client machine and the Oracle9iAS server. Like most firewalls, this one allows HTTP traffic to port 80 of the Oracle9iAS machine and possibly a few other specific port/protocol combinations- all others are blocked. The client's browser can retrieve the HTML start page for Plus, and the JVM can download the applet files and the locator.ior file. However, the Plus applet will fail when it tries to connect to the Locator using IIOP. At this point the user sees a message "Wide Area Network Connection Failed". (Note- this error message is not specific to firewall configurations. It also appears if the Discoverer components on the middle-tier are down.)

Again, this could work if the firewall administrator allowed unrestricted traffic to and from the Oracle9iAS machine. However, that is highly unlikely as it largely defeats the purpose of having the firewall.

Using Gatekeeper to Go Through a Firewall



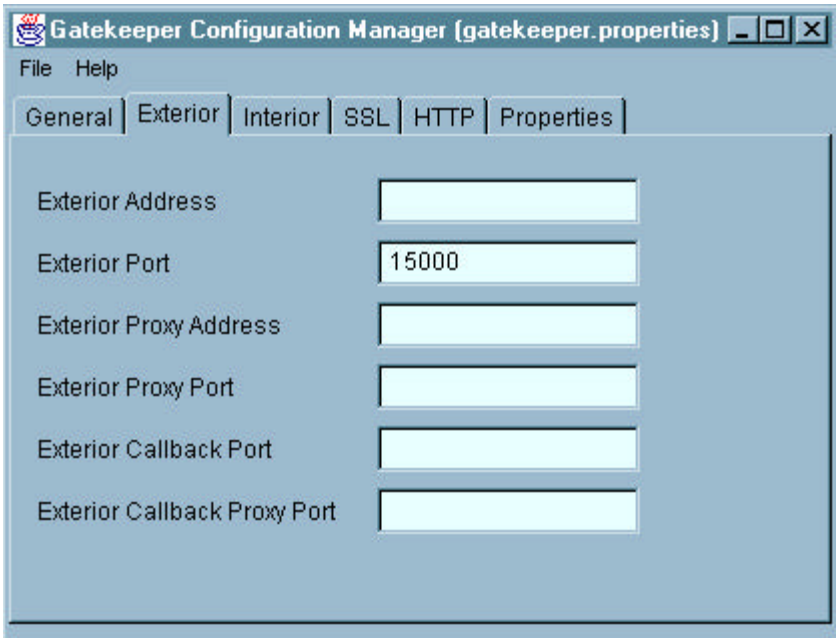
A firewall administrator *is* likely to open the firewall for traffic to a specific port on the Oracle9iAS machine. As stated above, the problem is that IIOP does not restrict the TCP port used for connections. The Gatekeeper resolves this problem:

1. The browser uses an HTTP connection to retrieve the HTML "start" page for Plus from the HTTP server. The instructions to download the Plus Java applet are embedded in that HTML page.
2. The browser's JVM (JInitiator or the MS JVM) uses HTTP to retrieve the applet files from the HTTP server. Once the applet files are downloaded, the applet is launched. The applet immediately retrieves the **gatekeeper.ior** and locator.ior files from the HTTP server.
3. The applet reads the information from the gatekeeper.ior and locator.ior files (IP address and TCP port) and uses either IIOP or HTTP* to connect through the Gatekeeper to the Locator.
   a. The gatekeeper.ior file specifies the IP address and TCP port the Gatekeeper "listens" to. The default port number is 15000 (as shown above).
   b. The applet uses the locator.ior information just as before, except it knows it must use the Gatekeeper as a proxy for all communications.
   c. Through the Gatekeeper, the Locator receives the applet's request for a Discoverer Session and replies to the Gatekeeper. In turn, the Gatekeeper sends the reply back to the applet.
4. From this point forward, the Plus applet communicates through the Gatekeeper with the same dis4ws process on the middle-tier using IIOP or HTTP*. Plus prompts the user for login information.

*The Gatekeeper may be configured to simply proxy IIOP traffic or to proxy IIOP tunneled through HTTP. See the last section of this document for an explanation of how this works and when to use it. Regardless of tunneling, the Gatekeeper listens to the port specified in the gatekeeper.ior file.

Configuring the Middle-Tier

Read the notes in the final section on using the gkconfig tool. Using the gkconfig tool, set the port number as shown below:



On the Exterior tab, specify the port number in the Exterior Port field. The Gatekeeper will listen for both IIOP and HTTP traffic on that port.

Do ***not*** specify any of the other fields. It doesn't hurt to put the Oracle9iAS machine's IP address in the Exterior Address field, but it will be used by default anyway. Specifying the Exterior Proxy Address or Port ***will*** cause problems. Those fields are for situations where there is a proxy server or NAT device between the Gatekeeper and the client. Those configurations are not typical and are not covered in this document.

Lastly, follow the Configuration Guide instructions for starting the Gatekeeper (it differs between Windows and UNIX).

Configuring the Client

By default, the Plus applet tries a sequence of connection attempts using different options (in order):

1. Direct IIOP connection (looks for locator.ior)
2. IIOP proxying connection (looks for gatekeeper.ior)
3. HTTP tunneling connection (looks for gatekeeper.ior)

It may take several seconds for each attempt to time-out and fail. If all attempts fail, an error message is displayed: “Local Area Network connection failed”…”Wide Area Network connection failed.”

If IIOP proxying or HTTP tunneling are used, skipping the direct IIOP connection attempt saves time. This is accomplished by adding the appropriate URL parameter before launching Plus.

- ORBAlwaysProxy to immediately try IIOP proxying
- ORBAlwaysTunnel to immediately try HTTP tunneling

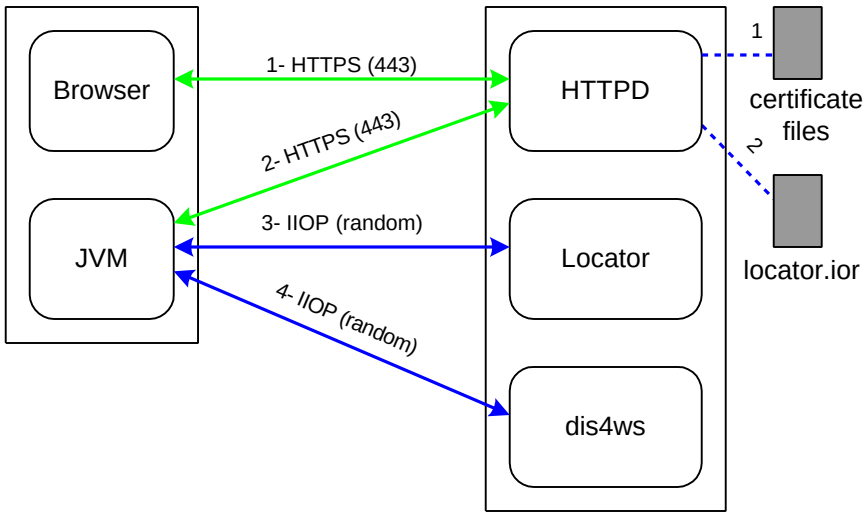
Do ***not*** use both in the same URL. For complete details and examples, see section 7.9.2 in the Configuration Guide.

SSL

SSL encrypts communications at the TCP socket level. Both HTTP and IIOP may be used over an SSL encrypted connection. Firewalls that “support” SSL cannot decipher the data in an SSL encrypted connection (that would defeat the purpose of encrypting it). Packet headers do indicate if the contents are SSL encrypted though, so a firewall can block non-SSL traffic on a given port.

For example, a firewall cannot differentiate between HTTPS (SSL encrypted HTTP) and IIOPS (SSL encrypted IIOP) on port 443. It can, however, block FTP traffic on port 443 while allowing SSL on port 443.

Most Common Mistake in SSL Setup



The most common mistake is to confuse enabling SSL on the HTTP server with enabling SSL with Discoverer Plus. They are very different activities and *almost* completely unrelated.

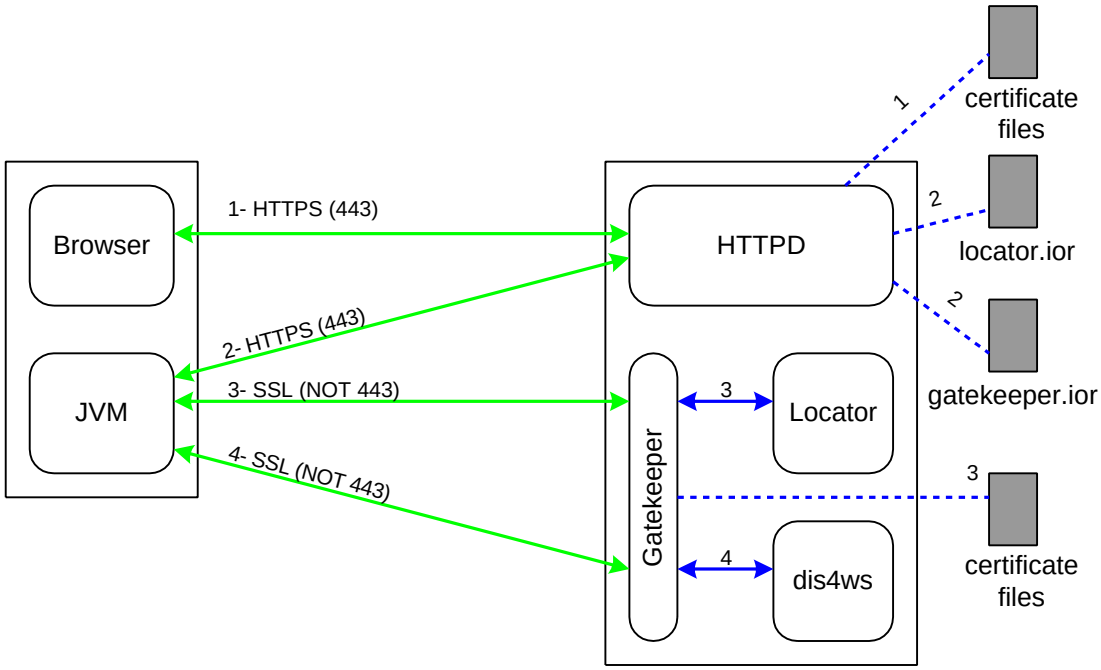
This common mistake is shown above:

1. The browser uses an HTTPS connection to retrieve the HTML "start" page for Plus from the HTTP server. SSL handshaking uses the certificate and key files on the HTTP server. The instructions to download the Plus Java applet are embedded in the HTML page.
2. The browser's JVM (JInitiator or the MS JVM) uses HTTPS to retrieve the applet files from the HTTP server*. Once the applet files are downloaded, the applet is launched. The applet immediately retrieves the locator.ior file from the HTTP server.
3. The applet reads the information in the locator.ior file (IP address and TCP port) and uses IIOP to connect to the Locator. The Locator tells the applet the location of a Discoverer Session (a dis4ws process on the middle-tier).
4. From this point forward, the Plus applet communicates directly with the same dis4ws process on the middle-tier using IIOP. Plus prompts the user for login information.

In steps 1 and 2, the communication is encrypted by SSL. In steps 3 and 4, the communication is ***not*** encrypted and uses IIOP. As stated at the beginning of this document, the applet does not use the browser to communicate with the Oracle9iAS middle-tier. However, because the HTML start page was retrieved over an SSL connection (the browser shows the lock icon), users mistakenly believe the Plus applet is also using SSL.

*Versions of Oracle JInitiator prior to 1.1.8.11 do not support HTTPS for loading Java class files. This may not be supported by all versions of the MS JVM for Internet Explorer either, but we do know it is supported by the latest version. This problem is detected by looking at the Java console as the applet is launched. URLs for class files are displayed in the console as they are downloaded. If the URLs start with "https" and the download fails, this is likely to be the cause.

Using Gatekeeper to Support SSL



The Locator and Discoverer Sessions objects (dis4ws processes) only use IIOP, they do not support SSL encryption. To secure communication between the Plus client and the Oracle9iAS middle-tier we once again turn to the Gatekeeper:

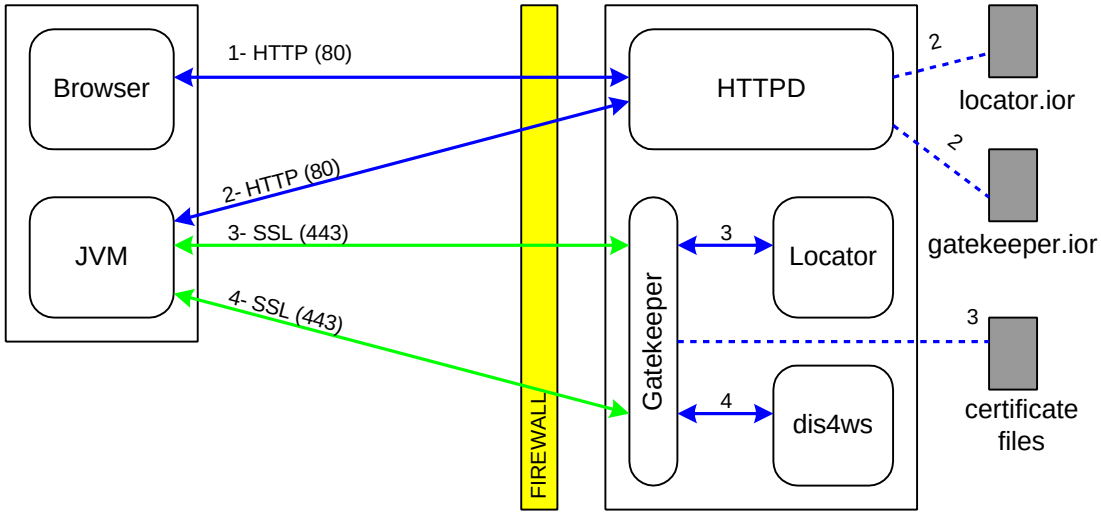
1. The browser uses an HTTPS connection to retrieve the HTML "start" page for Plus from the HTTP server. SSL handshaking between the browser and HTTP server uses the certificate and key files on the HTTP server. The instructions to download the Plus Java applet are embedded in the HTML page.
2. The browser's JVM (JInitiator or the MS JVM) uses HTTPS to retrieve the applet files from the HTTP server. Once the applet files are downloaded, the applet is launched. The applet immediately retrieves the gatekeeper.ior and locator.ior files from the HTTP server.
3. The applet reads the information from the gatekeeper.ior and locator.ior files (IP address and TCP port) and uses SSL to connect through the Gatekeeper to the Locator.
    a. The gatekeeper.ior file specifies the IP address and TCP port the Gatekeeper "listens" to. The default port number is 15000 (shown above as "not 443").
    b. The applet uses the locator.ior information just as before, except it knows it must use the Gatekeeper as a proxy for all communications. SSL handshaking between the applet and Gatekeeper uses the certificate and key files (not the same as the HTTP server's).
    c. Through the Gatekeeper, the Locator receives the applet's request for a Discoverer Session and replies to the Gatekeeper. In turn, the Gatekeeper sends the reply back to the applet.
4. From this point forward, the Plus applet uses SSL to communicate to the Gatekeeper and through to the same dis4ws process on the middle-tier*. Plus prompts the user for login information.

*Even though the Plus applet and the Gatekeeper communicate with SSL, the Gatekeeper proxy communicates with the Locator and Discoverer Session using unencrypted IIOP. These are all on the same Oracle9iAS machine though so the security implications are minimal.

The diagram above shows SSL encryption used by both the HTTP server and the Gatekeeper. This configuration is not typical but it demonstrates some very important points:

- Regardless of protocol, the Gatekeeper and HTTP server cannot listen to the same IP address and TCP port. By default, the HTTP server will listen for SSL on port 443, so the diagram shows that Gatekeeper must listen on another unused port.
- Certificate and key files are used to establish an SSL connection. Each server process (whether HTTPD, Gatekeeper, etc.) must be configured for SSL separately.
- The same certificate and key files may be used by both the HTTP server and Gatekeeper, but only if the encryption and signing method used are supported by both. Section 6.8.4 of the Configuration Guide describes the requirements for Gatekeeper.

Using Gatekeeper to Support SSL Through a Firewall



The diagram above is almost identical to section 3.3.2. The differences are the inclusion of a firewall and configuring the HTTP server to listen to port 80 without SSL.

As discussed previously, most firewalls support SSL traffic for port 443. However, they may not allow SSL traffic on any other ports (this varies from vendor to vendor). Technically, SSL connections do not have to use port 443. Both the Oracle HTTP Server and Gatekeeper can be configured to use SSL on arbitrary port numbers. However, port 443 is the most likely to be supported by firewall vendors.

The diagram above reflects a situation where the firewall only allows HTTP traffic on port 80 and SSL traffic on port 443, and we only have one Oracle9iAS machine with one IP address*. Given these

constraints, it is recommended to enable SSL on port 443 for the Gatekeeper and run the HTTP server without SSL on port 80. That way the applet files are transmitted "in the clear", but the user's login information and query data are SSL encrypted.

*Other solutions that allow both the HTTP server and Gatekeeper to use SSL are possible:

- One is to have multiple IP addresses (i.e.- multiple NICs) on the Oracle9iAS machine, and configure the HTTP server and Gatekeeper to listen on separate IP addresses. This is not a supported solution. While possible (according to the VisiBroker documentation), many people have reported problems getting this to work reliably.
- The other is to install the Gatekeeper on a separate machine. VisiBroker Gatekeeper is only licensed for use with Discoverer, so an entire Oracle9iAS installation must be performed on the separate machine. Then the gatekeeper.ior file must be manually copied to the Oracle9iAS machine with the SSL enabled HTTP server. See section 7.10.3 of the Configuration Guide for more information.
- Finally, a more simplified way to accomplish the previous point is to purchase a license from Inprise to install just the Gatekeeper product (also explained in section 7.10.3 of the Configuration Guide).

Configuring the Middle-Tier

Follow section 6.8 of the Configuration Guide exactly. Some potential sources of confusion:

- Section 6.8.3 talks about deciding whether to run Gatekeeper on port 443 and on the same machine as the HTTP server. If this is not clear, see the bullet points in the previous section.
- Section 6.8.5 talks about installing the SSL certificates in Gatekeeper. The order matters and the root CA certificate should be added last.
- Section 6.8.5 specifies that the "Enable SSL on Exterior" box to be checked. This will enable SSL on the port specified as the "Exterior Port" on the "Exterior" tab.
- Do **NOT** check "Enable SSL on Interior". Discoverer does not support this feature.

Configuring the Client

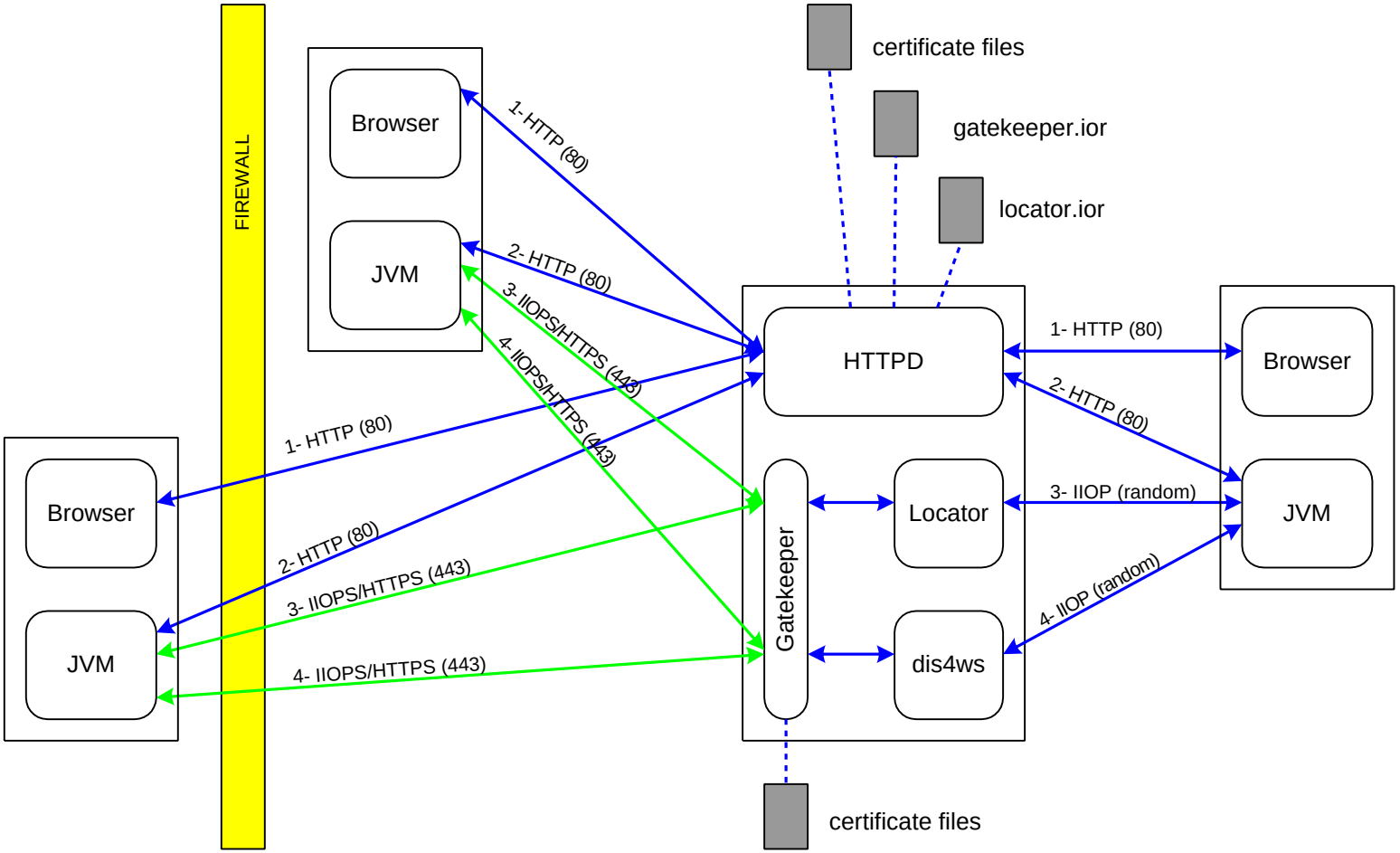
Section 6.8.6 and 6.8.7 of the Configuration Guide provide instructions for installing required SSL support files **on each client machine**. These files **must** be installed or else Plus will not support SSL communication to the Oracle9iAS middle-tier.

Sections 6.8.8 and 6.8.9 describe the required URL parameter ORBEnableSSL and valid combinations with the other URL parameters. Basically, users outside the firewall should use the ORBEnableSSL=yes parameter and users inside the firewall should use both ORBEnableSSL=yes and ORBAlwaysProxy=yes parameters (to enable SSL). Do not use ORBEnableSSL with ORBAlwaysTunnel in the same URL.

Common problems and mistakes:

- At the end of sections 6.8.6 and 6.8.7 of the Configuration Guide, URLs are provided so clients can download the required SSL support files. **By default, the Oracle9iAS HTTP server will incorrectly transfer those files to the clients.** This occurs because the HTTP server does not associate ".jar", ".dll", or ".so" files with a binary MIME type. See the last section of this document for details on how to correct this.
- Do NOT use "https://" in the URL for Discoverer 4i Plus unless the HTTP server is running with SSL enabled.
- Even with everything configured correctly, users do not see the lock icon in the browser so they assume Plus is not using SSL. To verify SSL is enabled and working in Plus, view the Java console or use a packet sniffer. The Java console (the easier of the two) will show a message indicating that SSL is enabled shortly after the applet is launched.
- The certificate chain was installed in the wrong order in Gatekeeper. This is demonstrated by "Invalid Cert Chain" messages in the Java console shortly after the Plus applet launches. Check the order of certificates and try again. If the order is correct and the error message persists, see the next bullet point.
- Gatekeeper does not recognize the certificate format. See the last section of this document for details and a solution.

All Client Configurations with a Single Server



ADDITIONAL NOTES

HTTP Tunneling

Because of the widespread use of the World Wide Web, the HTTP protocol is almost always permitted through firewalls. HTTP tunneling works by the sender "wrapping" another protocol in HTTP and the recipient "unwrapping" the original protocol.

An example may help:

> Suppose you must send a package from one office in your company to another, and the offices are halfway around the globe from each other. You put your package into an "interoffice" envelope and address it with an interoffice code. Your office mail clerk picks-up your package, notes that it needs to go halfway around the globe, and decides to ship it via a commercial shipping company (UPS, FedEx, DHL, etc.). The mail clerk cannot simply hand the interoffice envelope to the commercial shipper- he must put the interoffice envelope inside another shipping envelope with the address of the other office's mailroom. The mail clerk in the other office receives this package, opens it, recognizes that it is an interoffice envelope and delivers it to the interoffice address.

In the analogous example, interoffice shipping and commercial shipping are two different protocols. The mail clerk "wrapped" the interoffice package inside the commercial shipping package, and another clerk "unwrapped" the original interoffice package and forwarded it along.

That is exactly what HTTP tunneling does for IIOP. Firewalls may not allow IIOP packets through, but they will allow HTTP packets. However, this requires the client and the Gatekeeper to constant wrap/unwrap packets. There is a performance penalty, though it is not significant for most users.

For many customers, HTTP tunneling or SSL are the ***only*** solutions and IIOP proxying will not work. If there were multiple firewalls between the client machine and the Oracle9iAS machine, each firewall would have to allow IIOP traffic through a specific port. This may occur when an organization has multiple internal firewalls, in which case the customer may be able to open that port for IIOP on all firewalls. However, it usually occurs when the client is behind one company's firewall while the Oracle9iAS server is behind another company's firewall (B2B). In that case, one company does not control another company's firewall, but both allow HTTP or SSL traffic so HTTP tunneling or SSL must be used.

Using GKCONFIG

It is very important to pay attention to how gkconfig is launched. This affects the location of the gatekeeper.properties and gatekeeper.ior files.

To summarize the most important points of the Configuration Guide (section 7.10.2):

Discoverer uses the Gatekeeper installed in

```
<ORACLE_806_HOME>/vbroker/bin/gatekeeper
```

The gkconfig tool should be invoked from

```
<iSUITES_HOME>/Apache/Apache/htdocs/discwb4/applet
```

and ***not***

```
<ORACLE_806_HOME>/vbroker/bin/gatekeeper
```

If the previous point is followed, the default location of the **gatekeeper.properties**, **gatekeeper.ior**, and **gatekeeper.log** files will be

```
<iSUITES_HOME>/Apache/Apache/htdocs/discwb4/applet
```

If gkconfig is ***not*** run from the location specified above, the biggest problem is that the gatekeeper.properties files will not be in the correct location. This means that any configuration changes will not take effect even though the file was saved. If the documentation for starting Gatekeeper is followed, it expects the gatekeeper.properties file to be in a specific location. A default file is already provided, so the Gatekeeper will still start (making it easy to miss this problem), but users will have problems connecting with Plus.

Also, it is very easy to check for this problem:

After saving configuration changes in gkconfig, check the timestamp of the gatekeeper.properties file in

```
<iSUITES_HOME>/Apache/Apache/htdocs/discwb4/applet
```

After the gatekeeper starts, check the timestamp of the gatekeeper.ior file in

```
<iSUITES_HOME>/Apache/Apache/htdocs/discwb4/applet
```

If the files are missing or their timestamps are not correct, follow the documentation ***exactly*** for invoking gkconfig.

Setting MIME Types in HTTPD.CONF

As documented in the Configuration Guide, client machines must download a few files to support SSL with Discoverer 4i Plus. Unfortunately, the default configuration of the Oracle HTTP Server does not specify a binary MIME type for these files. Depending on the client browser*, without the correct MIME type settings, these files will be saved incorrectly on client machines. The file download will

complete successfully, but the file contents will be corrupt. Users do not see any indication of a problem with the files.

To correct this problem, simply add the relevant MIME types to the Oracle HTTP Server main configuration file:

On the HTTP server, edit the httpd.conf file.

Go to the section that begins with:

AddType allows you to tweak mime.types without actually editing it, or to
make certain files to be certain types.

Add the following lines:

```
AddType application/octet-stream .so
AddType application/octet-stream .dll
AddType application/octet-stream .jar
```

Save the file and restart the Oracle HTTP Server.

On the client machine, download the required files for SSL again.

*Some versions of MS Internet Explorer recognize the files as binary even without the MIME type information from the HTTP server. Netscape Navigator 4.x always encounters this problem.

Invalid or Incomplete Certificate Chain

When two machines attempt to establish an SSL-encrypted connection to each other, they go through a process known as “handshaking”. That term generically refers to the initial negotiation process for any protocol, but this section deals with a specific problem that may occur during SSL handshaking.

If SSL is enabled, the Gatekeeper verifies the validity of its certificate chain as soon as it launches. The Gatekeeper will not work correctly if the chain is incomplete, in the wrong order, or if the certificates are not in a valid format. See section 6.8.4 of the Configuration Guide for the supported formats.

Common problems and mistakes:

- A certificate chain supported by the Oracle HTTP Server does not necessarily work with Gatekeeper. The Oracle HTTP Server supports more formats than Gatekeeper.
- When using trial certificates, check the expiration dates. For example, trial certificates generated for a project last year have probably expired.

Oracle Discoverer 4i Plus Firewall and SSL Tips
February 2002
Author: Christopher H. Barron
Contributing Authors: Tejas Shah

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