



Configuring ArcIMS and ServletExec for use in a Corporate Firewall Environment

Introduction

What is a firewall and why might I want to use one?

A firewall is a program or hardware device that filters the traffic (packets) moving between the Internet and your web server or application server. It can be configured to implement security rules and filters that allow or deny certain network traffic moving in either direction. The filtering rules are normally based on network addresses and ports (*i.e. http=port 80, https=port 443, telnet=port 23, SMTP=port 25, RMI Registry=port 1099, ...*).

A firewall's primary purpose is to prevent outside agents or "hackers" from gaining access to sensitive data. A firewall can also be used to monitor and control sensitive files that may not be allowed to leave a company over the network. Access to data is controlled by defined policies that force all requests for sensitive data through a set of packet filtering rules.

The ArcIMS Application Server uses numerous ports for communication with its various clients. For example, the ArcIMS Spatial Server, Monitor, and Tasker all communicate with the ArcIMS Application Server on the Registry Port (5353). The ArcIMS Servlet Connector, which runs inside a servlet engine such as ServletExec, communicates with the ArcIMS Application Server on the Client Port (5300). For security reasons it may be desirable to protect access to the ArcIMS Application Server by placing a firewall in front of it.

The servlet container (a.k.a. servlet engine, or web container) is usually installed on the same machine as the web server. However, if the ArcIMS Application and Spatial Servers were behind a firewall on a separate machine from the web server machine, then having the servlet engine installed on the same machine as the web server forces the use of sluggish or possibly prohibited techniques.

For example:

- You could map a drive through the firewall, in order to allow the Spatial Server to write the map image files to a directory that's accessible to the web server.
- Or you could use a reverse proxy server, but some organizations may have policies that prohibit use of a production web server on the internal (ArcIMS) machine.

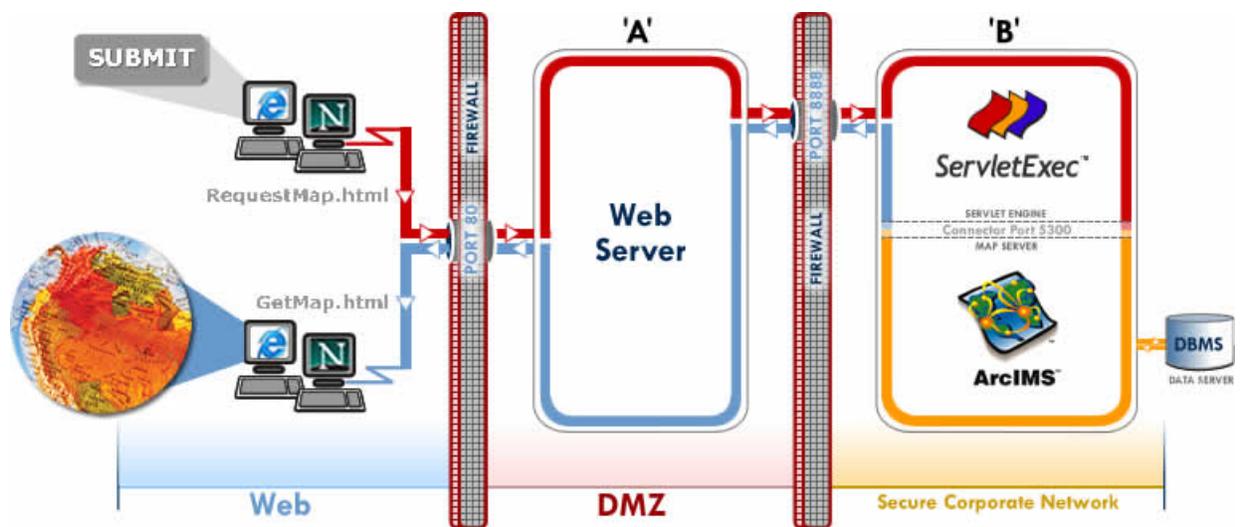
A more elegant, secure and efficient configuration would be to install the servlet engine on a machine separate from the web server. One such servlet engine that supports this configuration is ServletExec AS (not ServletExec ISAPI or NSAPI).

Installing ServletExec with ArcIMS in a Firewall Configuration

ServletExec AS is a Java Servlet and JavaServer Pages (JSP) engine that runs as a standalone process, separate from the web server process. ServletExec can be installed on the same machine as the web server software, or on a separate machine (i.e. web server software running on machine 'A', while ServletExec AS Engine runs on machine 'B').

Note: In both instances, ServletExec AS still has a small native component called an adapter or redirector which “hooks-into” the web server.

In this configuration, machine 'B' could even be the machine where ArcIMS is installed. Optionally, there could even be a firewall between machine 'A' and machine 'B'.



The diagram above and text below outline the components required on each server.

The following information gives general directions for installing ServletExec AS on a machine separate from the web server machine. For complete details on these steps see the online installation guide at <http://www.newatlanta.com/partners/esri/installation.jsp>, as well as the ServletExec Installation Guide included on the ArcIMS CD. The following directions are for configuring on Windows OS, but the configuration steps for Unix/Linux are similar.

ArcIMS/ServletExec Server Machine (machine 'B')

1. Install a web server (IIS, Apache, or iPlanet) and a JDK or JRE*. The web server on this machine is required only temporarily, to enable proper configuration of ServletExec AS. After successful configuration, the web server can be disabled or removed.

Use a web browser to test the temporary web server to ensure that it functions properly, serving a simple static html file. Stop the temporary web server.



*The Java Runtime Environment (JRE) does not contain a Java compiler, which is required by ServletExec to support JavaServer Pages (JSP); be sure to install a Java Development Kit (JDK), and not the JRE, if you plan to use JSPs.

2. Install ServletExec AS against the temporary web server instance by running the ServletExec AS installer on machine 'B', and choosing: "*Install a ServletExec AS instance*". This choice will install a web server adapter also.

Restart the temporary web server. Use a web browser to test the temporary web server, ensuring that it can use ServletExec AS to invoke a simple servlet. Try:

<http://localhost/servlet/TestServlet>

<http://localhost/servletexec/admin> (SE 5.x or higher)

<http://localhost/servlet/admin> (SE 4.x or older)

3. Install ArcIMS 4.0.1 or higher

The ArcIMS post-installer will (among other things):

- a. Setup the ServletExec AS instance so that it has 2 webapps deployed:
/esriadmin
/metadataexplorer
 - b. Create 5 virtual directories in the temporary web server:
Website
Output
Manager
esriadmin - pointing to the web application root
metadataexplorer - pointing to the web application root
 - c. Install the ArcIMS servlet connector into ServletExec AS.
4. Use a web browser on machine 'B' to run the ArcIMS diagnostic tool to ensure that the ArcIMS Servlet Connector is correctly setup to communicate with the ArcIMS Mapping Server. This can be done by opening a browser and entering the following URL:

http://<webserver>/servlet/com.esri.esrimap.Esrimap?Cmd=ConnectorPing

You should see a simple response stating "*Test successful*".

5. Stop the ServletExec AS java instance and open the StartServletExec batch/script file in a plain text editor. This file is located in ...*<ServletExec install directory>* *se-<instancename>* \. At the end of the long startup string, add the IP address of machine 'A' with the "*-allow*" parameter.

For example, if the IP address of machine 'A' is "*62.8.8.5*" then:

... *-port 8888 -allow 62.8.8.5*

Alternatively, use an IP address mask, such as "*62.8.8.**". Then save the change and close the file. Start the ServletExec AS java instance again. (This tells the ServletExec AS java instance to accept communication from any ServletExec AS native adapter running on 62.8.8.5).

Web Server Machine (machine 'A')

1. Install the web server software (IIS, Apache, or iPlanet)
2. Install the ServletExec AS native web server adapter (a.k.a. "redirector") into the web server software.
 - a. If using ServletExec 5.x or higher:

Run the ServletExec AS installer on machine 'A' and choose "*Install or Update a web server adapter*". When prompted, enter the IP address of machine 'B', and the port number it will use to communicate with the ServletExec AS java instance there (port 8888 by default). The ServletExec AS installer will update the web server's configuration files for you automatically.

- b. If using ServletExec 4.x,

Install a JRE or JDK.

Run the ServletExec AS installer on machine 'A' and once again choose: "*Install a ServletExec AS instance*". (Do not choose "*Install a web server adapter*").

When prompted to give your ServletExec AS java instance a name, you may want to consider naming it something like "*unused*", since it will not be used at all. When prompted if you want to run the SE AS Java Instance as a Windows Service, you may want to choose "*no*", since you may not have a need to run an instance from machine 'A' at all. (This is a temporary instance that ensures that the installer will run correctly).

By default, the installer will setup the adapter to communicate with the temporary instance at 127.0.0.1:8888. After the installer has completed, you'll need to edit the `servletoexec.properties` file in `... \inetpub \Scripts` so that the value of the '*instances*' property reflects the *ip:port* of machine 'B'.

ArcIMS Image Output / Deploying Web applications

The ArcIMS Server will generate/output image files representing requested mapping data. These files will be generated on machine 'B' where ArcIMS resides, and will be placed on machine 'B's local hard-drive.

The files should be placed into the document root of a Java Servlet Web Application named "output". **Note:** this is not, and should not be inside the document root of any web server.

If you haven't studied the example web application "*exampleWebApp*" that comes with ServletExec, please do so now, in addition to reading the chapter about *Web Applications* in the *ServletExec User Guide*.



Access the browser-based ServletExec Administrative User Interface from machine 'A'
(ServletExec 5.x: *http://<machineA>/servletexec/admin*)
(ServletExec 4.x: *http://<machineA>/servlet/admin*)
to create a brand new web application to hold the ArcIMS output.
You can run the browser from either machine.

If running ServletExec 5.x AS (or higher):

1. Application Name can be anything, but it may be most descriptive to just call it "output".
2. URL Context Path should be */output/*
3. Location can be anywhere on machine 'B', some examples are:
C:\ArcIMS\Output, or wherever the ArcIMS output directory is located.
4. Consider setting File Caching to *disabled*, since most ArcIMS output files will be retrieved only once.
5. Click "Submit" to create the web application. If a message indicating that necessary resources (WEB-INF folder, etc.) are not present appears, click the "Create" button to create those resources. A message indicating the web application has been successfully created/deployed should appear.

If running ServletExec 4.x AS (or earlier):

1. Follow the preceding 5 steps, to create the "output" web app on machine 'B'.
2. Configure the ServletExec native web server adapter to the */output* context path. The easiest way to do this is to rerun the ServletExec AS installer on machine 'A' again, only this time choose the 2nd option labeled: "Install or Update a web server adapter" (Don't confuse this choice with "Install a web server adapter"). When prompted, enter */output* (no quotes, only 1 slash). Then restart the web server on machine 'A'.

The ServletExec native web server adapter will now know that when it receives a request beginning with */output* that it should route the request over to the ServletExec AS java instance on machine 'B' so that the requested resource can be found and served.

Note: This manual step must be performed every time you add a new web app, or change the context path of an existing web app. Also note that there is no need to create a virtual directory for *output* on any web server.

Notes regarding ArcIMS web site files

1. ArcIMS web sites should have no references to machine 'B'. They should point to machine 'A' instead, since that is the location of the production web server. For example, with HTML Viewers you'll need to make the following adjustments: In *ArcIMSparam.js*, modify the *imsURL* and *imsOVURL* to point to machine 'A'.

For example: If the web server on machine 'A' is "*webserver.mydomain.com*" you'd need:

<http://webserver.mydomain.com/servlet/com.esri.esrimap.Esrimap?ServiceName=mymapservice>

2. You can choose to place your static web site files/Viewers in the traditional location (i.e. the document root of the web server on machine 'A'). Or, you can choose to place them elsewhere on the hard drive of machine 'A' (not in the document root), and create a virtual directory in the web server pointing to this other location. In both instances, requests for static web site files (*.js*, *.css*, *.html*, *.htm*, *etc...*) will be served by the web server on machine 'A' without any communication to machine 'B' (i.e. no communication to ServletExec). This gives the best performance for the serving of such static content, since there is no communication overhead between machine 'A' and machine 'B'.

A third deployment option may provide certain benefits including slightly easier administration and better portability (within an already complex setup). This deployment option places your ArcIMS websites into one or more Java Servlet web applications placed on machine 'B'. ServletExec includes an example web application "*exampleWebApp*", which is described within the ServletExec User Guide. The standard framework for a web-application is defined in the Java Servlet Specification. Deploying your web content from within the context of a web application offers many benefits such as portability, dynamic class reloading, event listeners, request filters, Access Control mechanisms (see [SE FAQ #243](#) for more information), and standardized deployments. Earlier in these instructions, we explained how to deploy the ArcIMS "*output*" web application.

To put your ArcIMS web sites into a Java Servlet web application, follow the same instructions used to create a new web app named "website" or some other appropriate name. Choose a URI context path and then configure the ServletExec native adapter (ServletExec 4.x only) to handle the new context path. You should not create virtual directories in your web server for these web sites. Either deploy as a web application, or deploy using virtual directories. Do not do both for the same ArcIMS web site.

More details about deploying ArcIMS websites can be found at:

http://caromap.esri.com/Website/BB/ScriptsTips/ServletEngineSeparateFromWebServer.htm#_Creating_ArcIMS_Websites



Notes regarding Firewall Connections

1. If the connection between the web server machine (machine 'A') and the ArcIMS/ServletExec server machine (machine 'B') goes through a firewall, then ServletExec's connector port (port 8888 by default) must be opened for two-way traffic, even though the web server will always initiate the connection
2. If using other connectors (ActiveX, Java or ColdFusion), the ArcIMS Connector Port (default 5300) must be opened through the firewall for two-way traffic that is initiated on the web server machine.
3. Timeout settings on connections through the firewall may disrupt connections to ServletExec. If not possible to disable the timeout, you may need to write a script that requests a map often enough to keep the port open.

For more information on using ServletExec with ArcIMS
<http://www.newatlanta.com/arcims>

Comments or Suggestions?
Please send them to: esri-support@newatlanta.com