ServletExec™ 6.0 Installation Guide

for Microsoft Internet Information Server
SunONE Web Server
Sun Java System Web Server
and Apache HTTP Server

NEW ATLANTA COMMUNICATIONS, LLC

ServletExec™ 6.0 Installation Guide

November 20, 2007
Version 6.0

100 Prospect Place • Alpharetta, Georgia 30005-5445
Phone 678.256.3011 • Fax 678.256.3012
http://www.newatlanta.com

This product includes software developed by the Apache Software Foundation (http://www.apache.org).

New Atlanta Communications, LLC makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, New Atlanta Communications, LLC reserves the right to revise this document and to make changes from time to time in its content without being obligated to notify any person of such revisions or changes.

The Software described in this document is furnished under a Software License Agreement ("SLA"). The Software may be used or copied only in accordance with the terms of the SLA. It is against the law to copy the Software on tape, disk, or any other medium for any purpose other than that described in the SLA.

Contents

1

1. Introduction

New Atlanta ServletExec 6.0 is a high-performance, reliable, and cost-effective web application server that implements the Java™ Servlet API 2.5, JavaServer Pages™ (JSP) 2.1, and JSP Standard Tag Library (JSTL) 1.2 standards defined by Sun Microsystems, Inc. as component technologies of the Java 2 Platform, Enterprise Edition (J2EE™). Additional information about J2EE, servlets, and JSP technologies can be found on Sun's web site:

http://java.sun.com/j2ee/

ServletExec enables you to deploy servlets and JSPs on Microsoft® Internet Information Server (IIS), SunONE Web Server, Sun Java System Web Server (SJSWS), and Apache HTTP Server in a standard, robust, high-performance environment.

1.1 ServletExec Products

ServletExec supports two basic configurations (in-process and out-of-process) with two products. The key characteristics of each configuration are listed in Table 1. ServletExec Configurations.

- **ServletExec 6.0 ISAPI**, an in-process web server plugin that adds high-performance servlet and JSP support to Microsoft IIS on Windows® 2008/Vista/2003/XP/2000.

- **ServletExec 6.0 Application Server [AS]**, an out-of-process web application server that includes web server adapters for deployment with Microsoft IIS, Apache SunONE, and SJSWS. It also comes with a built-in, java-based web server which can be used (for development) in lieu of running behind a commercial grade web server. SE AS is available for Windows 2008/Vista/2003/XP/2000 and various UNIX platforms, including SPARC Solaris, Linux, HP-UX, and AIX.

We recommend using SE AS unless you're using Microsoft IIS **and** have specific reasons to use SE ISAPI.

ServletExec ISAPI	ServletExec AS
• Runs within the web server process (loaded as a Windows DLL)	• Runs as a standalone process, separate from the web server
• Best performance	• Greatest architectural flexibility
• Easiest to install and configure	• Enhanced safety (process isolation)
• Single Java VM per web server	• Multiple Java VMs per web server
• ServletExec starts/stops automatically when the web server starts/stops	• ServletExec starts/stops independently of the web server

Table 1. ServletExec Configurations

There are separate ServletExec installers based on your web server, operating system, and choice of ServletExec Configuration. Make sure you've downloaded the correct installer, and then refer to the appropriate chapter of this manual for detailed installation instructions.

1.2 Technical Support

If you're having difficulty installing, configuring, or using ServletExec, be sure to utilize the ServletExec Technical Support webpage:

`http://www.newatlanta.com/support/servletexec/index.jsp`

There you will find a rich set of free support resources such as FAQs, Documentation, and a Product Mailing List (with online searchable archive). In addition there are paid support options that are described there.

1.3 64-bit (x64) Support

ServletExec 6.0 supports 64-bit web servers as follows:

- ServletExec ISAPI with IIS on 64-bit Editions of Windows® (AMD64/Intel64 CPU Architecture only).
- ServletExec AS with IIS on 64-bit Editions of Windows® (AMD64/Intel64 CPU Architecture only).
- ServletExec AS with Apache on 64-bit Unix/Linux. CPU Architecture should not matter since the native adapter used by SE AS on Unix/Linux gets built/compiled right on the target machine.

References to "x64" in this document mean the AMD64/Intel64 CPU Architecture (**not** the Intel Itanium (IA64) Architecture).

2

2. ServletExec/ISAPI: Microsoft IIS Plugin (Windows)

ServletExec/ISAPI is a plugin servlet engine for Microsoft Internet Information Server (IIS) on Microsoft Windows® 2008/Vista/2003/XP/2000. ServletExec/ISAPI enables you to deploy Java Servlets and JavaServer Pages (JSP) on Microsoft IIS in a standard, robust, high-performance environment.

The ServletExec/ISAPI installer is named `ServletExec_ISAPI_60.exe`. If you have not already done so, you can download this installer from:

http://www.newatlanta.com/c/products/servletexec/download/home

This chapter contains important information that will enable you to verify your installation of ServletExec/ISAPI. It will also be useful if you ever need to uninstall ServletExec/ISAPI and ensure you've completely removed all installed components.

2.1 Upgrading from a Previous Version

Upgrading from a previous version always involves uninstalling the older version and then installing the newer one. There are no updaters that will update your SE installation "in place". The only question is whether or not you can reuse the old SE configuration files in the newer version of SE. ServletExec 6.0 can use the configuration files from all previous versions of SE except versions 3.0C, 3.0E, and 2.2 (or earlier). If you have any of those older versions installed, you will not be able to reuse the configuration files at all. You'll need to use the SE Admin UI on the new installation to manually reenter your configuration data.

To upgrade to a new version of ServletExec/ISAPI and maintain your old configuration settings

1. Stop ServletExec ISAPI.
   - For IIS 7.0 on Windows 2008/Vista: To completely stop SE ISAPI, you can either: (a) stop the World Wide Web Publishing Service using the **Services/Administrative Tools** control panel or, (b) execute the `STOP_IIS_BAT` command.

- For IIS 7.0/6.0/5.1/5.0 on Windows 2008/Vista/2003/XP/2000: To completely stop SE, you can either: (a) stop the IIS Admin Service using the **Services/Administrative Tools** control panel or, (b) execute the `STOP_IIS_BAT` command.

2. Close all open applications, including DBMON.
3. Make backup copies of the `ServletExec Data`, `Servlets`, `classes` (if you've added an additional one), and `webapps` (versions 3.1 and above only) subdirectories of the `ServletExec ISAPI` directory. The location of this directory varies depending upon your version of ServletExec.
   - For ServletExec/ISAPI 3.0 and later: The default location for the `ServletExec ISAPI` directory is within the `C:\Program Files\New Atlanta` directory.
   - For ServletExec/ISAPI 2.2 and earlier: The default location for the `ServletExec ISAPI` directory is within the `C:\InetPub` directory.
4. Uninstall the old version of ServletExec/ISAPI. Click the **Add/Remove** or **Change/Remove** control panel (depending on your version of Windows) and follow the instructions on the **Install/Uninstall** tab. Or double-click the installer file directly and follow the instructions to uninstall.
5. Run the ServletExec/ISAPI 6.0 installer, installing into the same location as the previous installation.

   Important

   For a successful upgrade, you must install everything into the same directory as the previous version.

6. Copy the `ServletExec Data`, `Servlets`, `classes` (if you've added an additional one), and `webapps` (versions 3.1 and above only) subdirectories you made backup copies of in Step 2, to the new `ServletExec ISAPI` directory created by the installer in Step 5.
7. Start IIS.

After restarting your web server, ServletExec/ISAPI 6.0 will run using your old configuration settings.

Tip

If you have any problems, you can restore the subdirectories from the backups you made in Step 3.

2.2 System Requirements

2.2.1 Operating System and Web Server

- ServletExec/ISAPI 6.0 is not supported on Windows 95, Windows 98, Windows ME, or Windows NT.
- ServletExec/ISAPI 6.0 supports the following operating systems and web servers:

OS	Web Server
Windows 2008/Vista (both x86 & x64 editions)	Internet Information Server (IIS) 7.0
Windows 2003 R2 (both x86 & x64 editions)	Internet Information Server (IIS) 6.0
Windows 2003 (both x86 & x64 editions)	Internet Information Server (IIS) 6.0
Windows XP Professional	Internet Information Server (IIS) 5.1
Windows 2000	Internet Information Server (IIS) 5.0

Table 2. Supported Operating Systems and Web Servers

Important

For IIS 7 (on Win 2008/Vista) you must have the “ISAPI Extensions” and “ISAPI Filters” Features installed. Otherwise IIS 7 will never load the installed ServletExec DLL and servlet requests will result in 404 responses. This is true for both SE ISAPI and SE AS.

Those 2 Windows Features may be enabled as follows:

`Control Panel – Programs – Programs & Features – Turn Windows features on or off`

From there navigate as follows:

+ Internet Information Services

  + World Wide Web Services

    + Application Development Features

      # ISAPI Extensions

      # ISAPI Filters

Check the checkboxes to enable ISAPI Extensions and ISAPI Filters. You will then at least need to restart the IIS Admin Service, and possible even reboot Windows.

It is not necessary to install the Metabase Compatibility Feature.

2.2.2 JDK or JRE

ServletExec/ISAPI 6.0 is not supported on JDK or JRE 1.4.x or earlier.

To install ServletExec/ISAPI 6.0, you must first install one of these Java Development Kit (JDK™) or Java™ Runtime Environment (JRE) choices:

- JDK/JRE 1.5.x, or 1.6.x for Windows from Sun Microsystems

Important

The JRE does not contain a Java compiler, which is required by ServletExec to support JavaServer Pages (JSP). Be sure to install the JDK, and not the JRE, if you plan to use JSP.

You can download the Sun JDKs for Windows from:

`http://java.sun.com/j2se/`

Note

ServletExec/ISAPI 6.0 (when run on a 64-bit Edition of Windows) requires that the JDK/JRE also be a 64-bit version.

2.2.3 JDK/JRE Installation

To install ServletExec/ISAPI, you must have first installed a compatible version of the JDK or JRE as explained in Section 2.2.2.

Important

The JDK or JRE must be installed on a local drive and not on a mapped network drive. If the JDK/JRE has been installed on a mapped network drive, ServletExec/ISAPI will not be able to load and initialize the Java VM.

2.2.3.1 Multiple JDK/JRE Versions Installed

You may have multiple versions of the JDK/JRE installed on your system. ServletExec/ISAPI (including the installer) uses registry entries to determine which installed JDK/JRE to use. It will look for an installed JDK first and if it doesn't find one it will look for a JRE. Here's the complete algorithm:

1. Look for a Sun JDK using the following registry key:
   `HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Development Kit`
   - If Sun JDK found, go to Step 2. Otherwise, look for a Sun JRE using the following key:

     `HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment`
   - If Sun JRE found, go to Step 2.
   - If no Sun JDK/JRE found using above registry entries, abort ServletExec installer.
2. Read the `CurrentVersion` variable from the key found in Step 1. Currently, the only valid values for this variable start with "1.5" or "1.6" (including, for example "`1.5.12`").
3. Append the value of the `CurrentVersion` variable from Step 2 to the key from Step 1 to create a new key. For example:

   `HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Development Kit\1.5`

   or…

   `HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment\1.5`

4. Read the value of the `JavaHome` variable for the key from Step 3 to find the location of the JDK or JRE.

If you launch the `DBMON` debugging tool and then restart your web server, ServletExec/ISAPI displays the Java VM settings in the `DBMON` console window during initialization. You can examine the classpath displayed by ServletExec/ISAPI in the `DBMON` console window to see which version of the JDK is being used.

See Section 12.4 of the ServletExec 6.0 User Guide for a discussion of `DBMON`.

2.3 Uninstalling Other Servlet Engines

Very Important

You **must** uninstall any other servlet engines previously installed for use with Microsoft IIS before installing and using ServletExec/ISAPI. In particular, the Filter DLL registry entries or ISAPI Filters metabase entries associated with other servlet engines must be removed

See Section 2.5.4 for more information.

2.4 Running the ServletExec/ISAPI Installer

Double-click the ServletExec/ISAPI installer icon to launch Setup. Please review the ServletExec License Agreement carefully and check the READ ME displayed by the setup program for any last minute notices before continuing.

2.5 What Was Installed and/or Modified?

When you installed ServletExec/ISAPI, three changes were made to your system:

- The `Servlet Exec ISAPI` directory was created.
- The `ServletExec_ISAPI.dll` file was installed.
- Registry and/or metabase entries for ServletExec/ISAPI were created or modified.

The following sections describe each of these changes.

2.5.1 The ServletExec ISAPI Directory

The `ServletExec ISAPI` directory was created in the location you selected during the installation process. The default location suggested by the installer is:

```
C:\Program Files\New Atlanta\ServletExec ISAPI
```

There are no restrictions on the location of the `ServletExec ISAPI` directory.

Important

Do not move the `ServletExec ISAPI` directory after installation. There is a registry entry that allows ServletExec/ISAPI to find this directory. If you move this directory, ServletExec/ISAPI will not be able to find its configuration files.

See Section 2.5.4 for more information.

The `ServletExec ISAPI` directory contains the following subdirectories:

`classes`

The classes subdirectory is automatically added to the ServletExec VM classpath. Place class files that will be shared by multiple web applications in this directory.

See the READ ME file within the classes subdirectory for more information.

`Documentation`

This subdirectory contains the ServletExec User Guide, and the Servlet 2.5 and JSP 2.1 API documentation from Sun Microsystems.

`Examples`

This subdirectory contains examples of a web application and a JSP page with an embedded bean.

See the appropriate chapters of the ServletExec User Guide for instructions on running these examples.

`lib`

The `lib` subdirectory contains the Java archive (.jar) files required by ServletExec/ISAPI.

Important

Do not modify the contents of the `lib` directory!

`Servlet Logs`

This subdirectory contains the `Servlet.log` files that capture the output from the servlet `log()` method. In some cases, this directory isn’t created until the first ServletExec initialization.

`ServletExec Data`

This subdirectory contains the ServletExec/ISAPI configuration files. These files are generally modified using the ServletExec Admin UI, but can also be edited manually.

See the ServletExec User Guide for more information on the Admin UI.

`Servlets`

This subdirectory contains the class files for legacy servlets hosted by ServletExec/ISAPI. These servlets run under the context of the web application named “default-app”.

See the READ ME in the Servlets directory and the ServletExec User Guide for more information.

`webapps`

This directory contains a subdirectory for each configured SE virtual server [VS]. A web application can be automatically deployed to an SE VS by placing it in the subdirectory for that virtual server as a WAR file or open directory structure.

See the READ ME in the webapps directory and the ServletExec User Guide for more information.

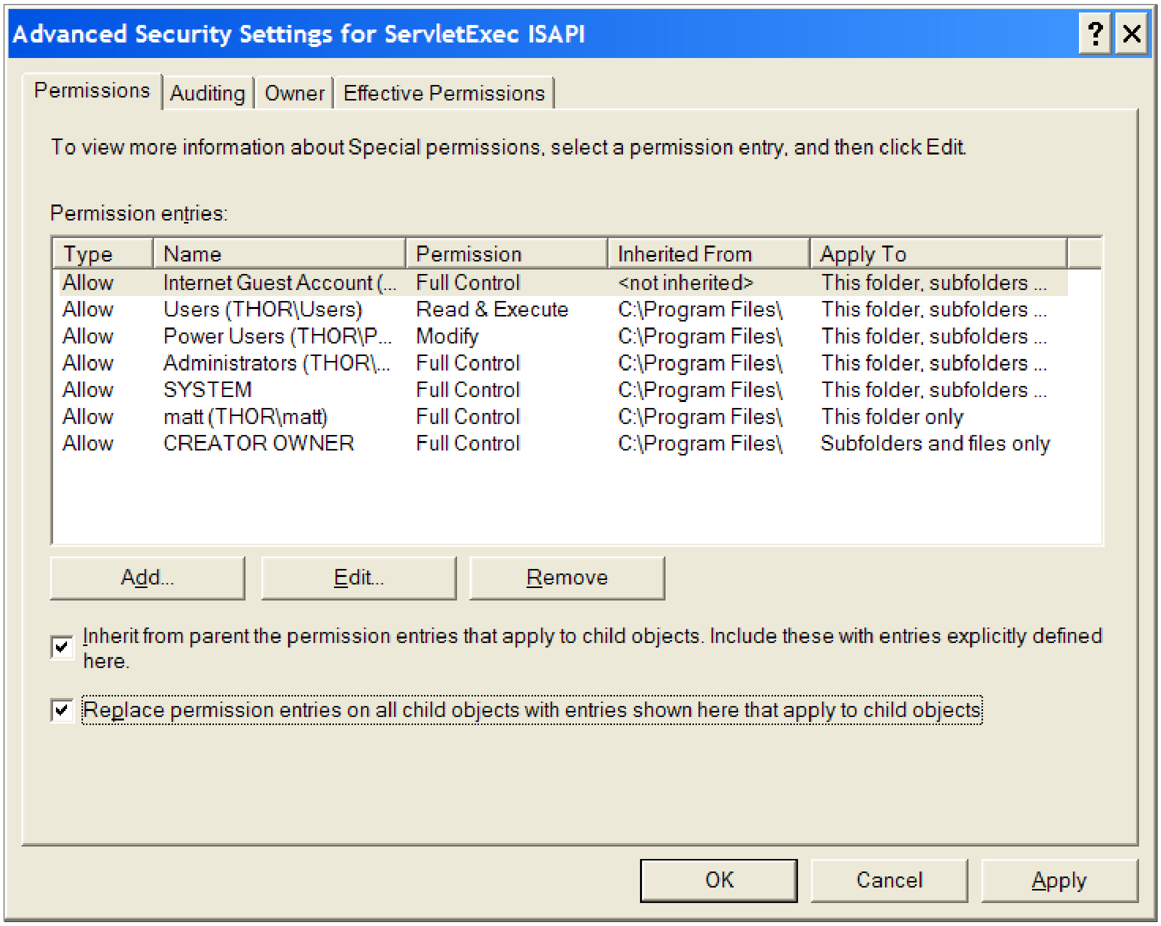
2.5.2 NTFS Permissions

Important

If you’re using the NT File System (NTFS), permissions for the `ServletExec ISAPI` directory and its subdirectories **must** be set so that ServletExec/ISAPI has read and write access to these directories. Because ServletExec/ISAPI runs as part of the IIS process, it will run as different users at different times.

The following Groups should be granted Full Control to the `ServletExec ISAPI` directory and its subdirectories: `SYSTEM` and `Authenticated Users`.

The following User should be granted Full Control to the `ServletExec ISAPI` directory and its subdirectories: `IUSR_`*`<server-name>`* (the user created by IIS for processing requests for anonymous users).



To view more information about Special permissions, select a permission entry, and then click Edit.

Permission entries:

Type	Name	Permission	Inherited From	Apply To
Allow	Internet Guest Account (...	Full Control	<not inherited>	This folder, subfolders ...
Allow	Users (THOR\Users)	Read & Execute	C:\Program Files\	This folder, subfolders ...
Allow	Power Users (THOR\P...	Modify	C:\Program Files\	This folder, subfolders ...
Allow	Administrators (THOR\...	Full Control	C:\Program Files\	This folder, subfolders ...
Allow	SYSTEM	Full Control	C:\Program Files\	This folder, subfolders ...
Allow	matt (THOR\matt)	Full Control	C:\Program Files\	This folder only
Allow	CREATOR OWNER	Full Control	C:\Program Files\	Subfolders and files only

☑ Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here.

☑ Replace permission entries on all child objects with entries shown here that apply to child objects

Figure 1. ISAPI NTFS Permission Settings (Windows XP)

With IIS 6.0, the Application Pool Identity used for the ServletExec worker process should also be granted Full Control to the ServletExec ISAPI directory and its subdirectories.

For more details regarding File Permission issues with IIS, please see SE FAQ #60:
http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=60

2.5.3 ServletExec_ISAPI.dll and the Virtual Directory [VD] within IIS

The `ServletExec_ISAPI.dll` dynamic link library (DLL) must be installed in a physical directory that is pointed to by a Microsoft IIS virtual directory [VD] and that VD **must** have `Execute` permission set to “Scripts and Executables”.

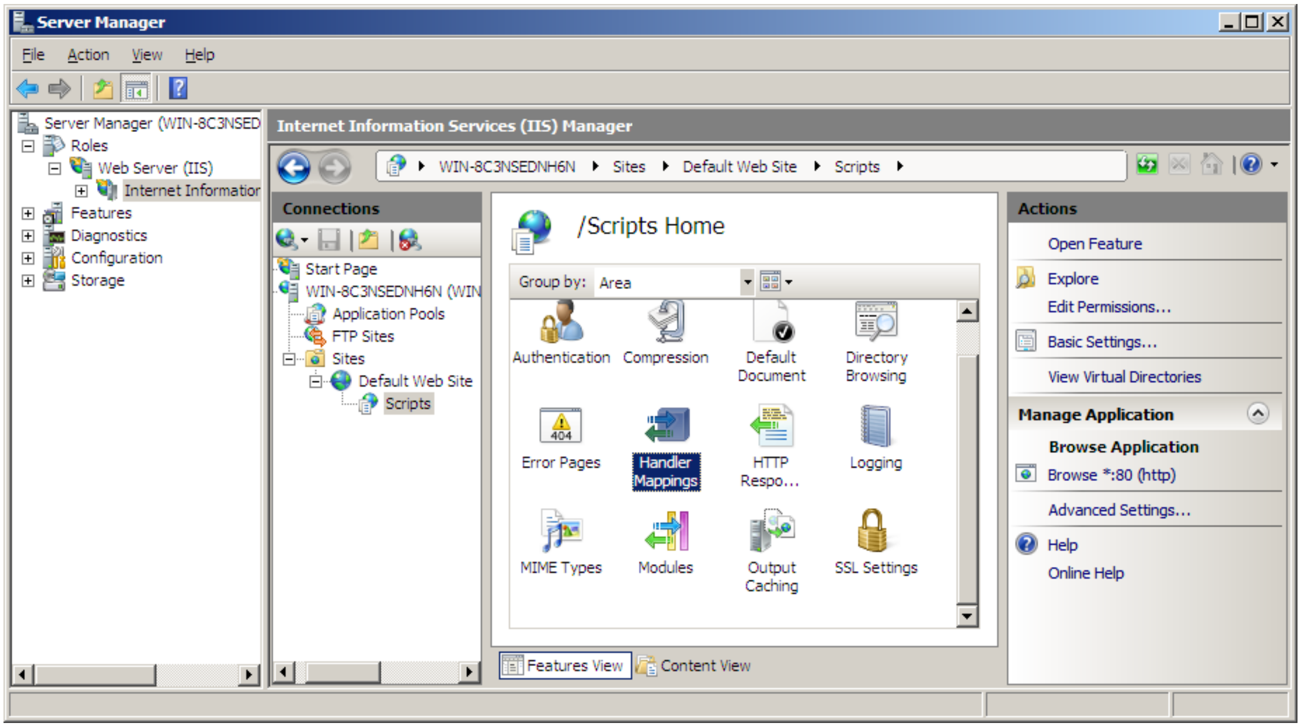
In a default IIS 5 installation, the `Scripts` VD is preconfigured and mapped to the `C:\InetPub\Scripts` physical directory. If the ServletExec/ISAPI installer found this default configuration, it automatically copied `ServletExec_ISAPI.dll` to the `C:\InetPub\Scripts` directory and did not prompt you for a location for this file.

If the ServletExec/ISAPI installer did NOT find a `Scripts` VD defined in the default website for IIS (IIS 6 & 7 do not have the `Scripts` VD preconfigured) then you will be prompted to select the physical directory where `ServletExec_ISAPI.dll` will reside.

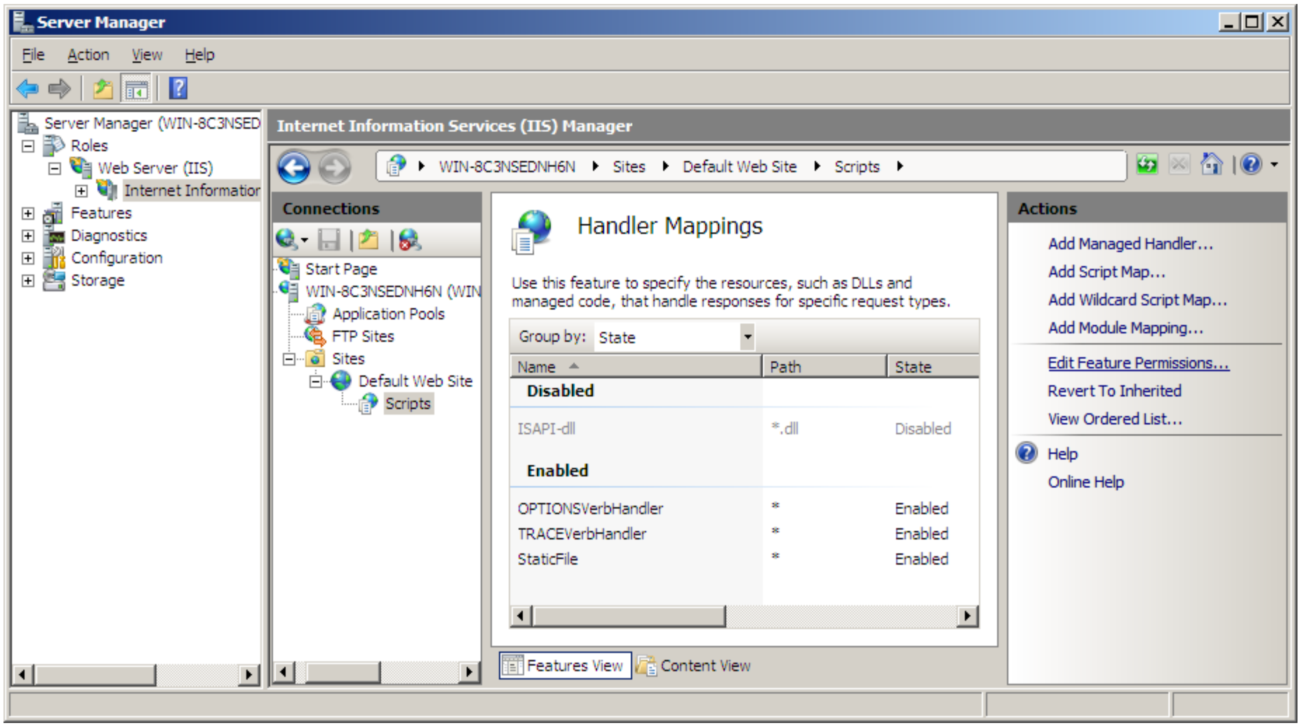
The SE ISAPI installer will then create a VD named `ServletExec` inside your IIS default website, which points to the physical folder you selected. In the case of IIS 6, a Web Service Extension will also be created in IIS which points to that DLL (with IIS 7 the equivalent to the Web Service Extensions are the “CGI & ISAPI Restrictions”). If you wish to use SE ISAPI (or AS) with an IIS website other than the *default* website, then please read SE FAQ #54:
http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=54

And if your version of IIS is IIS 7 (Win 2008/Vista) and you have manually created your own Virtual Directory [VD], its Execute setting is disabled by default. Here is how to enable it:

Click Start – right click on Computer and choose Manage to bring up the Server Manager Dialog. Expand “Roles”. Expand “Web Server (IIS)”. Click on Internet Information Services. Expand <Machine Name>. Expand “Sites”. Expand <your web site>. Click on your ServletExec VD that you created (with SE AS, the name of that VD must be “Scripts”). After clicking on that VD, double click its “Handler Mappings” icon. If you see “ISAPI-dll” listed as Disabled then click “Edit Feature Permissions…” and check the “Execute” checkbox. Then confirm that the state of “ISAPI-dll” is now “Enabled”. Here are 4 screenshots showing how to configure an IIS website to run SE AS when you have manually created the necessary VD.



Virtual Directory Screenshot #1



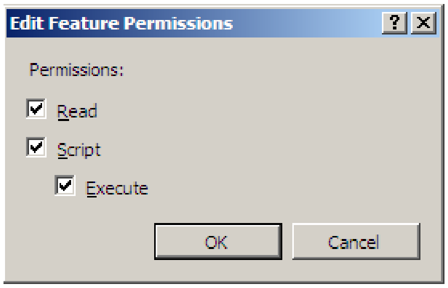
Handler Mappings

Use this feature to specify the resources, such as DLLs and managed code, that handle responses for specific request types.

Group by: State

Name	Path	State
Disabled		
ISAPI-dll	*.dll	Disabled
Enabled		
OPTIONSVerbHandler	*	Enabled
TRACEVerbHandler	*	Enabled
StaticFile	*	Enabled

Virtual Directory Screenshot #2



Edit Feature Permissions

Permissions:

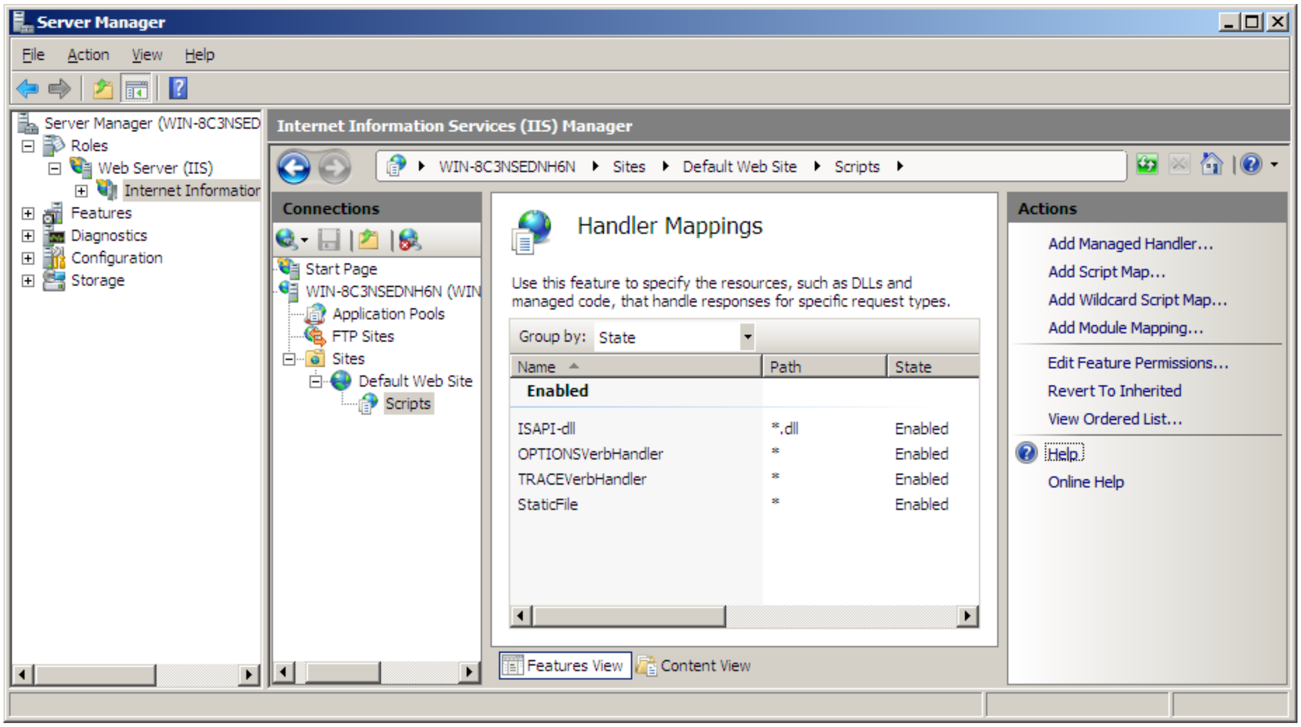
☑ Read

☑ Script

☑ Execute

OK Cancel

Virtual Directory Screenshot #3



Handler Mappings

Use this feature to specify the resources, such as DLLs and managed code, that handle responses for specific request types.

Group by: State

Name	Path	State
Enabled		
ISAPI-dll	*.dll	Enabled
OPTIONSVerbHandler	*	Enabled
TRACEVerbHandler	*	Enabled
StaticFile	*	Enabled

Virtual Directory Screenshot #4

To confirm your settings for IIS

1. In Internet Service Manager, open the Properties dialog for the virtual directory that maps to the physical directory in which `ServletExec_ISAPI.dll` resides. (With SE ISAPI, this is a VD named `ServletExec`; with SE AS the VD must be named `Scripts`).

2. Verify that the **Application Name** parameter under **Application Settings** is available and that execute permission is set to “Scripts and Executables”.

   Note: If using SE ISAPI with IIS 6 or IIS 7, and with more than 1 IIS Application Pool defined in IIS, we strongly recommend that you read this SE Interest List posting:

http://www.newatlanta.com/c/support/servletexec/self_help/archiveSearch/detail?messageId=221073

3. Click **Apply**, and click **OK**.

2.5.4 Registry & Metabase Entries

The ServletExec/ISAPI installer creates a new registry entry with the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\New Atlanta Communications\ServletExec ISAPI
```

This key contains a single parameter named `Home`, that contains the path to the `ServletExec ISAPI` directory.

Note

If you move the `ServletExec ISAPI` directory after installation, you must modify this key to contain the new path.

2.5.4.1 Filter DLLs Registry Entry

Some servlet engines do not use the metabase ISAPI Filter entry for Microsoft IIS (as described in the following section), but instead continue to use the old `Filter DLLs` registry entry. This parameter has the following key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
```

Very Important

You **must** remove entries for other servlet engines previously installed for Microsoft IIS from the `Filter DLLs` parameter. The uninstallers for most servlet engines do not automatically remove this entry.

2.5.4.2 Metabase ISAPI Filter Entry

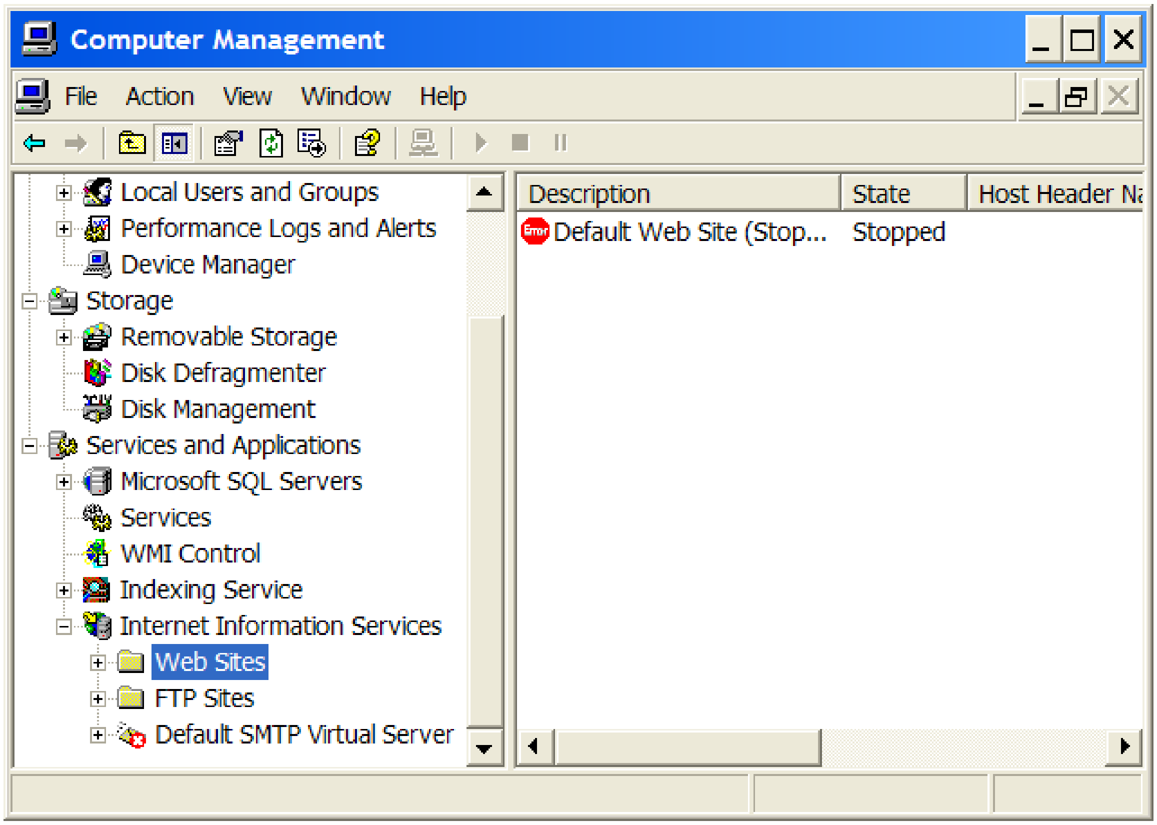
The ServletExec/ISAPI installer automatically modifies the IIS 7.0/6.0/5.1/5.0 (Windows 2008/Vista/2003/XP/2000) metabase to add the ISAPI Filter entry.

Very Important

You **must** remove entries for other servlet engines you may have previously installed. The uninstallers for most other servlet engines do not automatically remove the metabase ISAPI Filter entry.

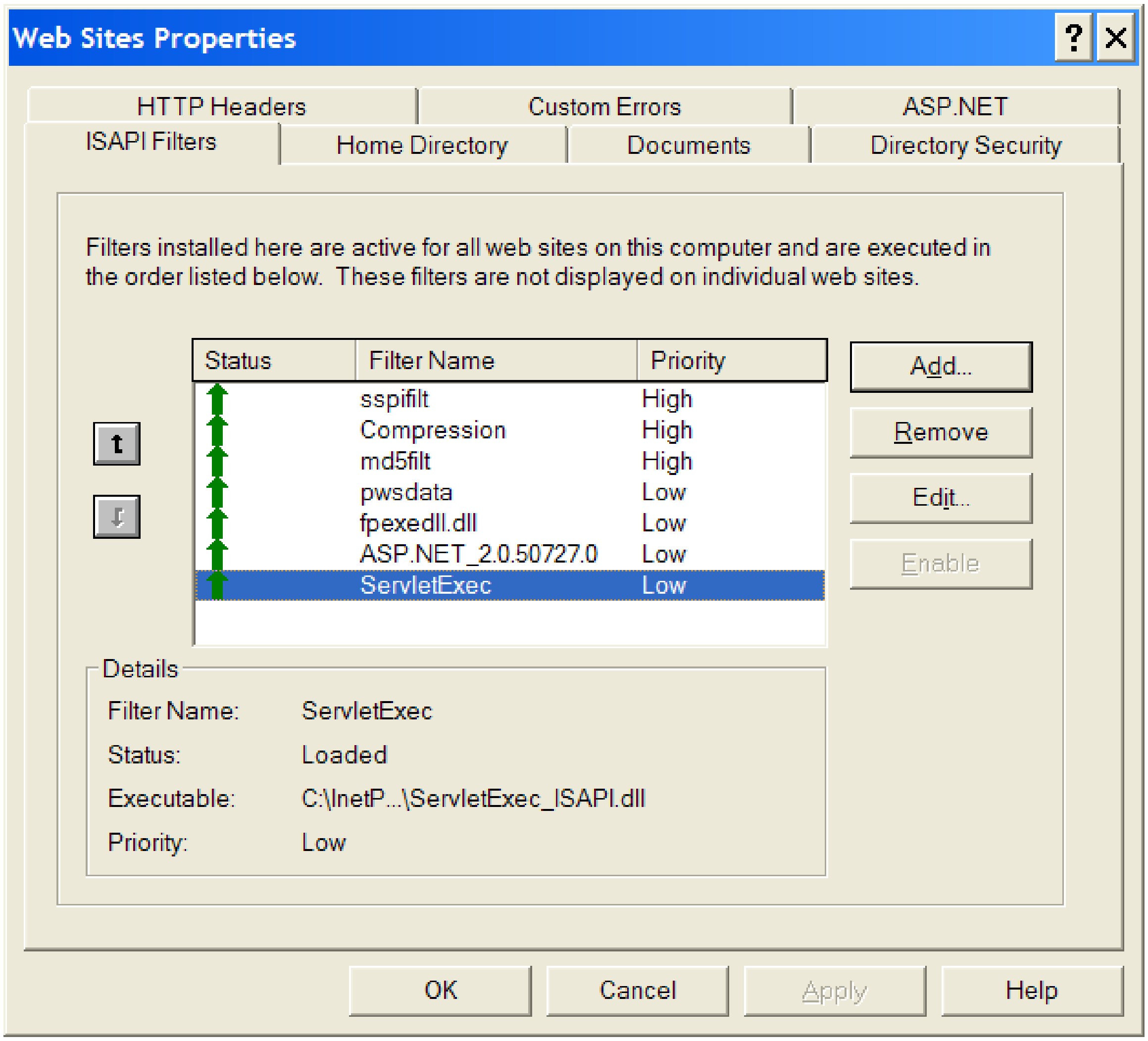
To examine or manually modify the metabase ISAPI Filter under IIS 5.1/5.0

1. Right-click on **My Computer** and choose **Manage**.
2. Expand **Services & Applications** (lower left).
3. Expand **Internet Information Services**.
4. Right-click on **Websites** and choose **Properties** (*see* Figure 2).
5. Click the **ISAPI Filters** tab (*see* Figure 3).



Description	State	Host Header Na
Default Web Site (Stop...	Stopped	

Figure 2. “Web Sites” Properties (IIS 5.1)



Filters installed here are active for all web sites on this computer and are executed in the order listed below. These filters are not displayed on individual web sites.

Status	Filter Name	Priority
↑	sspifilt	High
↑	Compression	High
↑	md5filt	High
↑	pwsdata	Low
↑	fpexedll.dll	Low
↑	ASP.NET_2.0.50727.0	Low
↑	ServletExec	Low

Details

Filter Name: ServletExec

Status: Loaded

Executable: C:\InetP...\ServletExec_ISAPI.dll

Priority: Low

Figure 3. ISAPI Filters (IIS 5.1)

6. Click any previously installed servlet engines, and then click **Remove**.
7. Click **ServletExec,** and then click **Edit…** to examine or modify it (if you wish).
8. If you do not see **ServletExec**, and you are using IIS 5.1/5.0 (Windows XP/2000) then click **Add** and follow the instructions to add the ServletExec/ISAPI entry, making sure to point to `ServletExec_ISAPI.dll`.

Note that with IIS 6 you should **not** modify the location of the SE DLL here since IIS 6 is also “pointing” to that same DLL via its Web Service Extensions and the Web Service Extensions are not modifiable. Both the Web Service Extensions and the ISAPI Filters must be pointing to the same exact copy of the DLL.

2.6 Basic Authentication Security with IIS

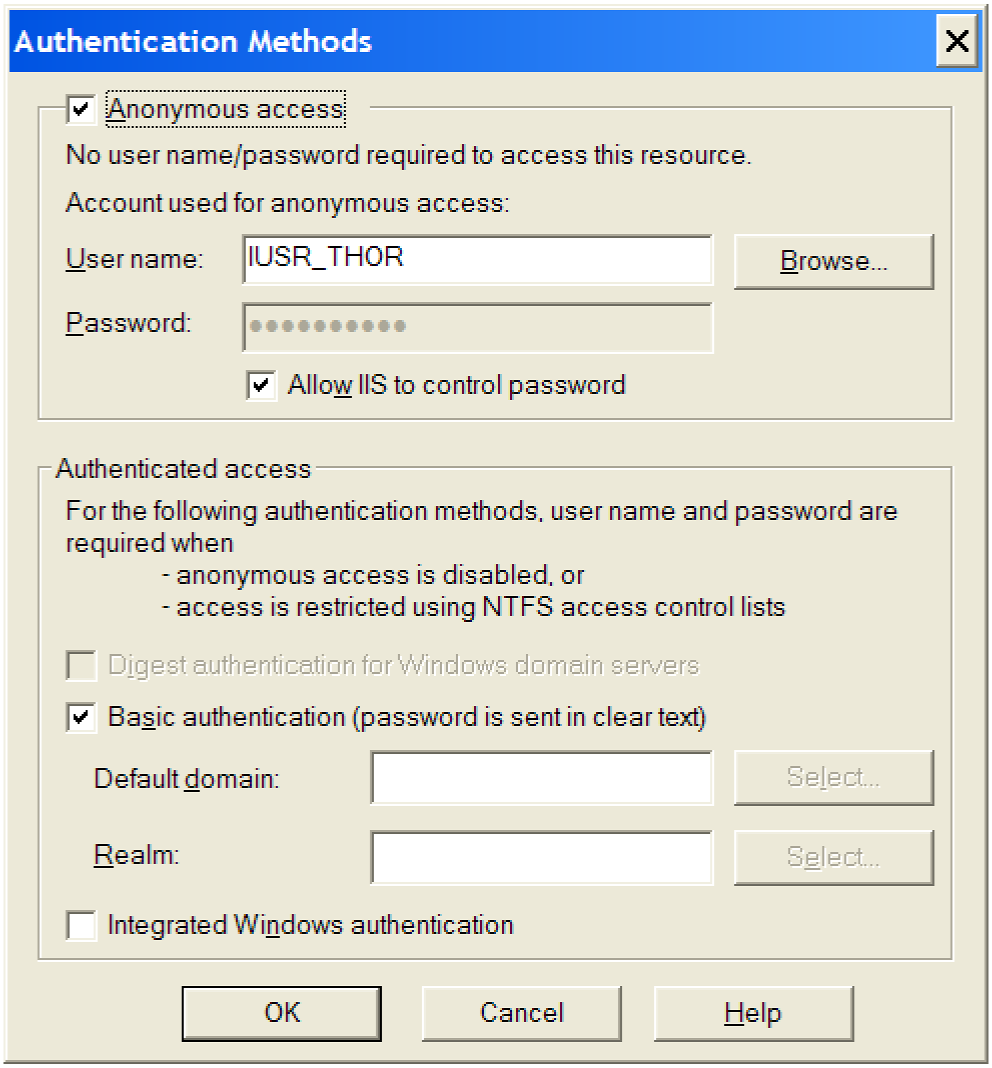
If you deploy any web applications that utilize Basic Authentication then the username and password used to access such protected resources must match those of an existing Windows user as defined in the Windows User Manager.

2.6.1 Enabling Basic Authentication in IIS

Basic Authentication is disabled by default in IIS, so you must enable it in this case. Please read SE FAQ #65:
http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=65
for helpful details, and realize that this can be enabled at various “levels” within IIS.

To enable Basic Authentication for a specific IIS website

1. Right-click on **My Computer** and choose **Manage**.
2. Expand **Services & Applications** (lower left).
3. Expand **Internet Information Services**.
4. Expand **Websites**
5. Right-click the website you wish to modify, and then click **Properties**.
6. On the **Directory Security** tab and under **Anonymous Access and Authentication Control**, click **Edit…**. You should then see a dialog similar to Figure 4.



☑ Anonymous access

No user name/password required to access this resource.

Account used for anonymous access:

User name: IUSR_THOR

Password: •••••••••••

☑ Allow IIS to control password

Authenticated access

For the following authentication methods, user name and password are required when
- anonymous access is disabled, or
- access is restricted using NTFS access control lists

☐ Digest authentication for Windows domain servers

☑ Basic authentication (password is sent in clear text)

Default domain:

Realm:

☐ Integrated Windows authentication

Figure 4. Authentication Methods (IIS 5.1 on Windows XP)

7. Check to make sure **Basic Authentication** is selected. By default, IIS authenticates users only in the local domain. If you need to authenticate users in a network domain, beside **Default domain for basic Authentication:**, click **Edit** and specify the network domain.
8. Make sure that **Integrated Windows authentication** is <u>not</u> selected. You will not be able to access any resources protected by Basic Authentication if this option is selected.
9. Click **OK** to close each dialog.

2.7 User Accounts for Microsoft IIS

Because ServletExec/ISAPI runs as part of the Microsoft IIS process, your servlets will run under different user accounts at different times. This will primarily affect their ability to read and write to the file system because access to the NT File System (NTFS) is based on the user account of the process.

Rules Governing IIS User Accounts

1. During normal request processing in your servlet's `service()`, `doGet()`, or `doPost()` method, your servlet will be running under the user account of the authenticated user, if the user had to enter a username and password to access your servlet. Otherwise, your servlet will be running under the `IUSR_`*`<server-name>`* account (the account under which IIS runs anonymous user requests).
2. If your servlet is configured to be loaded by ServletExec/ISAPI during initialization, your init() method will be executed under the `SYSTEM` account for IIS 5.1 and earlier and under the Application Pool Identity used for the ServletExec worker process for IIS 6.0. Otherwise, if your servlet is loaded when it receives its first request, the rules for item #1, above, apply
3. Normally, your servlet's destroy() method will run under the SYSTEM account for IIS 5.1 and earlier, and under the Application Pool Identity used for the ServletExec worker process for IIS 6.0 and newer when it is invoked due to a shutdown of ServletExec/ISAPI. However, if your servlet is reloaded due to a class file modification, the rules for item #1, above, apply

Make sure the appropriate users are given NTFS read and/or write permission for any directories for which your servlet needs to read and/or write files.

2.8 Reinitializing ServletExec/ISAPI

To reinitialize ServletExec/ISAPI (which must be done after modifying the ServletExec/ISAPI classpath, for example), you must completely stop and restart IIS.

To reinitialize IIS 7.0/6.0/5.1/5.0 on Windows 2008/Vista/2003/XP/2000

There are three ways to do this in Windows 2008/Vista/2003/XP/2000.

Using the Internet Services Manager administrative tool…

- Just click **Restart IIS**.

Using the Services control panel…

1. In Control Panel, click **Services**.
2. Click **IIS Admin Service**, and then click **Restart**.

Using the command line…

- Execute the batch file `stop_iis.bat` (found in the `ServletExec ISAPI` directory).

2.9 Uninstalling ServletExec/ISAPI

To completely uninstall ServletExec/ISAPI

1. Stop **IIS** as described in Section 2.8.

2. Close any open DBMON windows.
3. Open **Control Panel** and click **Add/Remove Programs**.
4. In the list, click **ServletExec 6.0 ISAPI**, and then click **Add/Remove** or **Change/Remove**, depending on your operating system.
5. Sometimes the ServletExec/ISAPI uninstaller may not be able to remove all of the installed files. Check to see if the uninstaller deleted the `ServletExec ISAPI` directory, and if not, delete it.

3

3. ServletExec/AS: Application Server (Windows)

The ServletExec Application Server (ServletExec/AS) is a standalone web application server that hosts Java Servlets and JavaServer Pages (JSP). ServletExec/AS can be used by itself using its built-in web server or in conjunction with any of these commercial-grade web servers: Microsoft IIS, SunONE Web Server, Sun Java System Web Server [SJSWS], or Apache HTTP Server. Native Web server adapters for these web servers are included in the ServletExec/AS installer.

The ServletExec/AS for Windows installer is named `ServletExec_AS_60.exe`. If you haven't already done so, you can download the installer file from:

http://www.newatlanta.com/c/products/servletexec/download/home

This chapter contains important information that will allow you to verify your installation of ServletExec/AS for Windows. It will also be useful should you decide to uninstall ServletExec/AS and need to make sure you've completely removed all installed components.

3.1 Upgrading from a Previous Version

Upgrading from a previous version always involves uninstalling the older version and then installing the newer one. There are no updaters that will update your SE installation "in place". The only question is whether or not you can reuse the old SE configuration files in the newer version of SE. ServletExec 6.0 can use the configuration files from all previous versions of SE except versions 3.0C, 3.0E, and 2.2 (or earlier). If you have any of those older versions installed, you will not be able to reuse the configuration files at all. You'll need to use the SE Admin UI on the new installation to manually reenter your configuration data.

To upgrade to ServletExec/AS 6.0 from 3.0 or 3.1 and maintain your old configuration settings

1. Stop your web server (*for IIS 7.0 stop the WWW Publishing Service, for IIS 6.0 or older stop the IIS Admin Service*). If you've installed ServletExec/AS adapters for multiple web servers, stop each one that is running.

2. Make backup copies of the `ServletExec Data`, `Servlets`, `classes` (if you've placed any of your own there), and `webapps` (versions 3.1 and above only) subdirectories for all ServletExec/AS instances.
   - For ServletExec/AS 3.0 and higher, the default locations are within:

     `C:\Program Files\New Atlanta\ServletExec AS\se-<`*`instance-name`*`>`
3. Close all open applications, making sure to include DBMON.
4. Uninstall each of the old versions of ServletExec/AS. In Control Panel, click **Add/Remove Programs** and follow the instructions on the **Install/Uninstall** tab, choosing To uninstall ServletExec/AS or a web server adapter when prompted.
5. Run the ServletExec/AS 6.0 installer. You will need to re-run the installer for each ServletExec/AS instance.

   Note

   The default installation directory for versions 2.2 and earlier is different.

   Important

   For a successful upgrade, you must install everything into the same directory as the previous version.
6. Copy the `ServletExec Data`, `Servlets`, `classes` (if you've added an additional one), and `webapps` (versions 3.1 and above only) subdirectories to the new `ServletExec AS` directory created by the installer in Step 5. In ServletExec/AS 6.0, the default location of the `ServletExec AS` directory is:

   `C:\Program Files\New Atlanta\ServletExec AS`
7. Start your web server.

After restarting your web server, the new version of ServletExec/AS will run using your old configuration settings.

Tip

If you have any problems, you can restore the subdirectories from the backups you made in Step 3.

3.2 System Requirements

3.2.1 Operating System and Web Server

- ServletExec/AS 6.0 is not supported on Windows 95, Windows 98 or Windows ME.
- ServletExec/AS 6.0 supports the following operating systems and web servers through the use of web adapters:

OS	Web Server
Windows 2008/Vista (both x86 & x64)	▪ Microsoft IIS 7.0 ▪ Apache HTTP Server 1.3.9 - 1.3.x, 2.0.43 - 2.0.x, and 2.2.4 – 2.2.x ▪ Sun Java System Web Server [SJSWS 7.0u1] ▪ SunONE Web Server 6.1
Windows 2003 (both x86 & x64)	▪ Microsoft IIS 6.0 ▪ Apache HTTP Server 1.3.9 - 1.3.x, 2.0.43 - 2.0.x, and 2.2.4 – 2.2.x ▪ Sun Java System Web Server [SJSWS 7.0u1] ▪ SunONE Web Server 6.1
Windows XP Professional	▪ Microsoft IIS 5.1 ▪ Apache HTTP Server 1.3.9 - 1.3.x, 2.0.43 - 2.0.x, and 2.2.4 – 2.2.x ▪ Sun Java System Web Server [SJSWS 7.0u1] ▪ SunONE Web Server 6.1
Windows 2000	▪ Microsoft IIS 5.0 ▪ Apache HTTP Server 1.3.9 - 1.3.x, 2.0.43 - 2.0.x, and 2.2.4 – 2.2.x ▪ Sun Java System Web Server [SJSWS 7.0u1] ▪ SunONE Web Server 6.1

Table 3. AS-Windows Supported Operating Systems and Web Servers

Important

For IIS 7 (on Win 2008/Vista) you must have the “ISAPI Extensions” and “ISAPI Filters” Features installed. Otherwise IIS 7 will never load the installed ServletExec DLL and servlet requests will result in 404 responses. This is true for both SE ISAPI and SE AS.

Those 2 Windows Features may be enabled as follows:

```
Control Panel – Programs – Programs & Features – Turn Windows features on or off
```

From there navigate as follows:

+ Internet Information Services

+ World Wide Web Services

+ Application Development Features

# ISAPI Extensions

# ISAPI Filters

Check the checkboxes to enable ISAPI Extensions and ISAPI Filters. You will then at least need to restart the IIS Admin Service, and possible even reboot Windows.

It is not necessary to install the Metabase Compatibility Feature.

3.2.2 JDK or JRE

ServletExec/AS 6.0 is not supported on JDK or JRE 1.4.x or earlier.

To install ServletExec/AS 6.0, you must first install one of these Java Development Kit (JDK™) or the Java™ Runtime Environment (JRE) choices:

- JDK/JRE 1.5.x, or 1.6.x for Windows from Sun Microsystems

  Important

  The JRE does not contain a Java compiler, which is required by ServletExec to support JavaServer Pages (JSP). Be sure to install the JDK, and not the JRE, if you plan to use JSP.

You can download the Sun JDKs for Windows from:

`http://java.sun.com/j2se/`

3.2.3 JDK/JRE Installation

To install ServletExec/AS, you must have first installed a compatible version of the JDK or JRE as explained in Section 3.2.2.

Important

The JDK or JRE must be installed on a local drive and not on a mapped network drive. If the JDK/JRE has been installed on a mapped network drive, ServletExec/AS will not be able to load and initialize the Java VM.

3.2.3.1 Multiple JDK/JRE Versions Installed

You may have multiple versions of the JDK/JRE installed on your system. The ServletExec/AS installer uses registry entries to determine which installed JDK/JRE to use. It will look for an installed JDK first and if it doesn't find one, it will look for a JRE. It will look for JDK/JREs from vendors in the following order: Sun Microsystems, and then IBM. Study the complete algorithm below, to gain a better understanding.

How ServletExec selects an installed JDK/JRE

1. Look for a Sun JDK using the following registry key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Development Kit
   ```

   - If Sun JDK found, go to Step 2. Otherwise, look for a Sun JRE using the following key:

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment
     ```

   - If Sun JRE found, go to Step 2. Otherwise, look for an IBM JDK or JRE using the following keys .

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\IBM\Java Development Kit
     ```

     or…

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\Java Runtime Environment
```

- If no Sun or IBM JDK/JRE found using the above registry keys, then abort ServletExec installer.

2. Read the `CurrentVersion` variable from the key found in Step 1. Currently, the only valid values for this variable start with "1.3" or "1.4" (including, for example "`1.4.2`").
3. Append the value of the `CurrentVersion` variable from Step 2 to the key from Step 1 to create a new key. For example:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Development Kit\1.4
   ```

   or…

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment\1.4
   ```

4. Read the value of the `JavaHome` variable for the key from Step 3 to find the location of the JDK or JRE.

3.3 Uninstalling Other Servlet Engines

Very Important

You **must** uninstall any other servlet engines previously installed before installing and using ServletExec/AS. In particular, modifications to the obj.conf, magnus.conf and httpd.conf configuration files for other servlet engines must be removed.

- For Microsoft IIS, the ISAPI Filters metabase entries associated with other servlet engines must be removed.
  See Section 3.5.3.2 for more information on ISAPI filters metabase entries.
- For SunONE Web Server & SJSWS, make sure the internal servlet and JSP support is disabled.
  See Section 3.5.4.2 for information on disabling servlet engines.
- For Apache HTTP Server, any directives associated with other servlet engines in `httpd.conf` must be removed.
  See Section 3.5.5.2 for more information on `httpd.conf` entries.

3.4 Running the ServletExec/AS Installer

To run the ServletExec/AS Installer

1. Double-click the ServletExec/AS installer icon. After unpacking, the installer will display the ServletExec 6.0 Application Server Setup window.
2. Click **Next** to bring up the Setup Type window.

   After clicking **Next** on the welcome window, you will be prompted to choose a setup type, as shown in Figure 4.

   The first time you run the ServletExec/AS installer, you will be offered two options for setup type. For subsequent installations, you will have additional options.

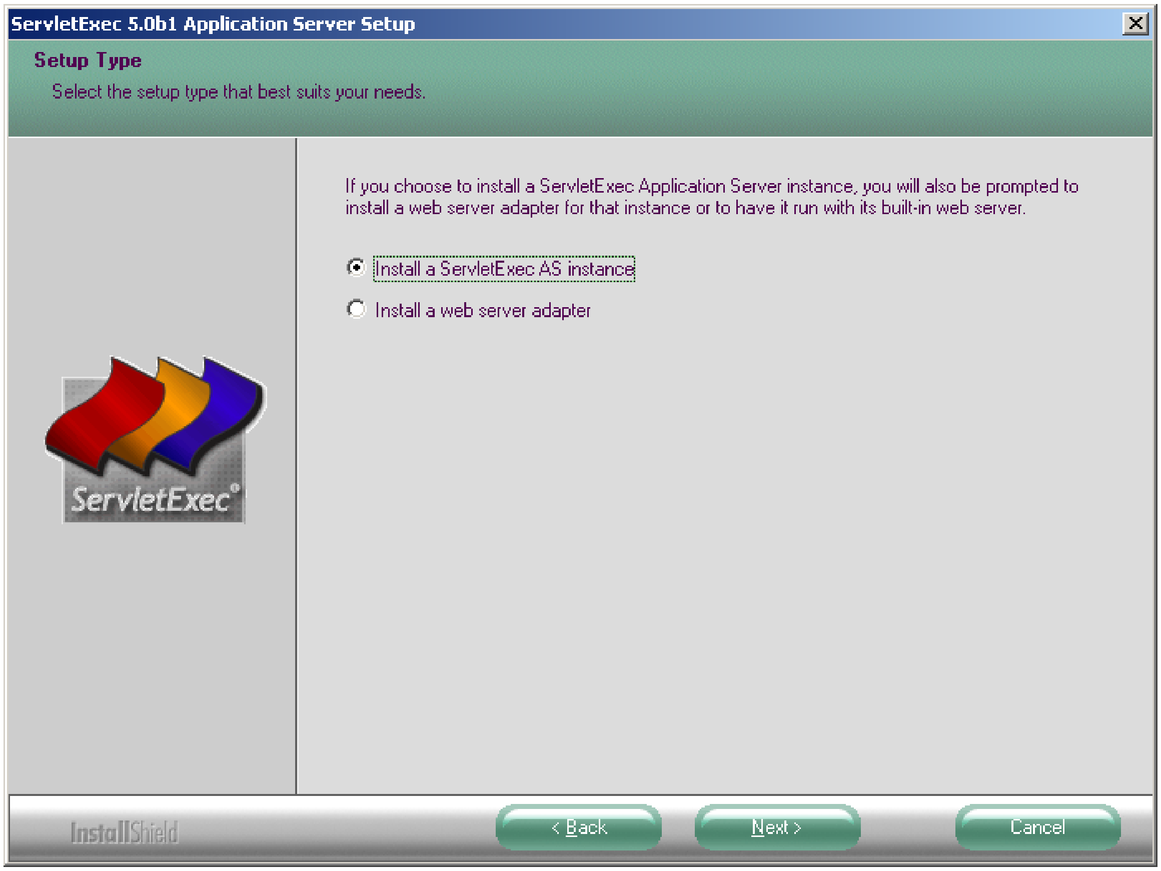
Typically, you will choose **Install a ServletExec/AS instance**, which will install a single ServletExec/AS instance and a single web server adapter.

Important

You must install at least one web server adapter whenever you install a ServletExec/AS instance.

The only time you would choose **Install a web server adapter** is after you have previously installed a ServletExec/AS instance on a different physical machine from the web server, and now wish to install a ServletExec/AS adapter to allow the web server to communicate with the ServletExec/AS instance running on the remote machine.

The remainder of this section applies only if you have selected **Install a ServletExec/AS instance**.



ServletExec 5.0b1 Application Server Setup

Setup Type

Select the setup type that best suits your needs.

If you choose to install a ServletExec Application Server instance, you will also be prompted to install a web server adapter for that instance or to have it run with its built-in web server.

● Install a ServletExec AS instance

○ Install a web server adapter

Figure 4. Setup Type

3. Click **Next**. Review the license agreement carefully and check the READ ME for any last minute notices before continuing.
4. After accepting the license agreement and reviewing the READ ME, the ServletExec/AS installer prompts you for the destination folder for installing the

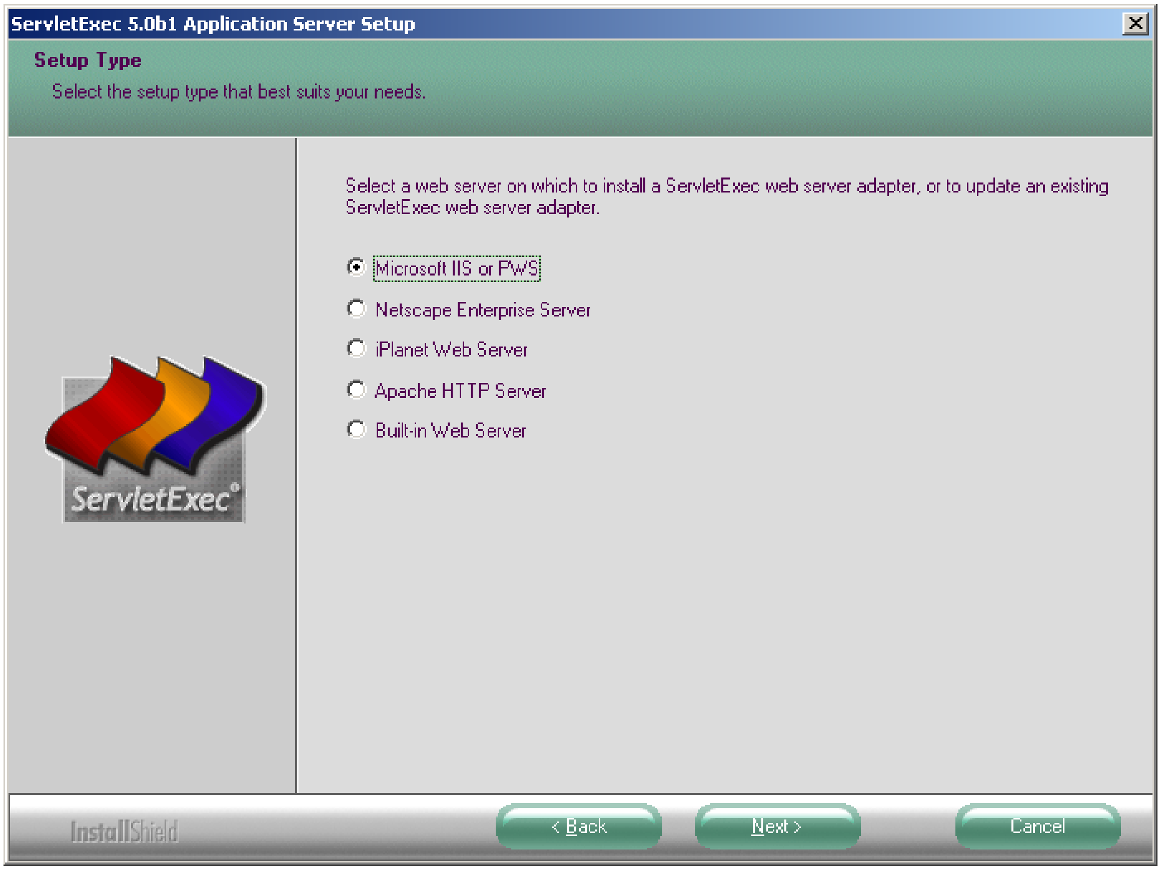
ServletExec/AS files. You may choose the default or click **Browse** to select a different folder.

Important

Do NOT move or rename the folder you choose after ServletExec/AS is installed.

5. Enter an **Instance Name**. After selecting a destination folder, the ServletExec/AS installer prompts you for the ServletExec/AS instance name. This name is used to uniquely identify the ServletExec/AS instance for administration, and is especially important if you install more than one instance on your server. By default, the installer suggests using the computer name as defined in Control Panel's **Network**, but you're free to use any unique name. Avoid using spaces or other special characters (">", "/", etc.) in the instance name.
6. Select a web server. After defining the ServletExec/AS instance name, the installer prompts you to select a web server on which to install a web server adapter for communicating with the ServletExec/AS instance that is about to be installed. The selection dialog is illustrated in Figure 5.

   If you select SunONE or SJSWS, the ServletExec/AS installer will then prompt you to select a server instance from a list of configured servers (both SunONE and SJSWS support multiple servers per physical machine).



ServletExec 5.0b1 Application Server Setup

Setup Type

Select the setup type that best suits your needs.

Select a web server on which to install a ServletExec web server adapter, or to update an existing ServletExec web server adapter.

● Microsoft IIS or PWS

○ Netscape Enterprise Server

○ iPlanet Web Server

○ Apache HTTP Server

○ Built-in Web Server

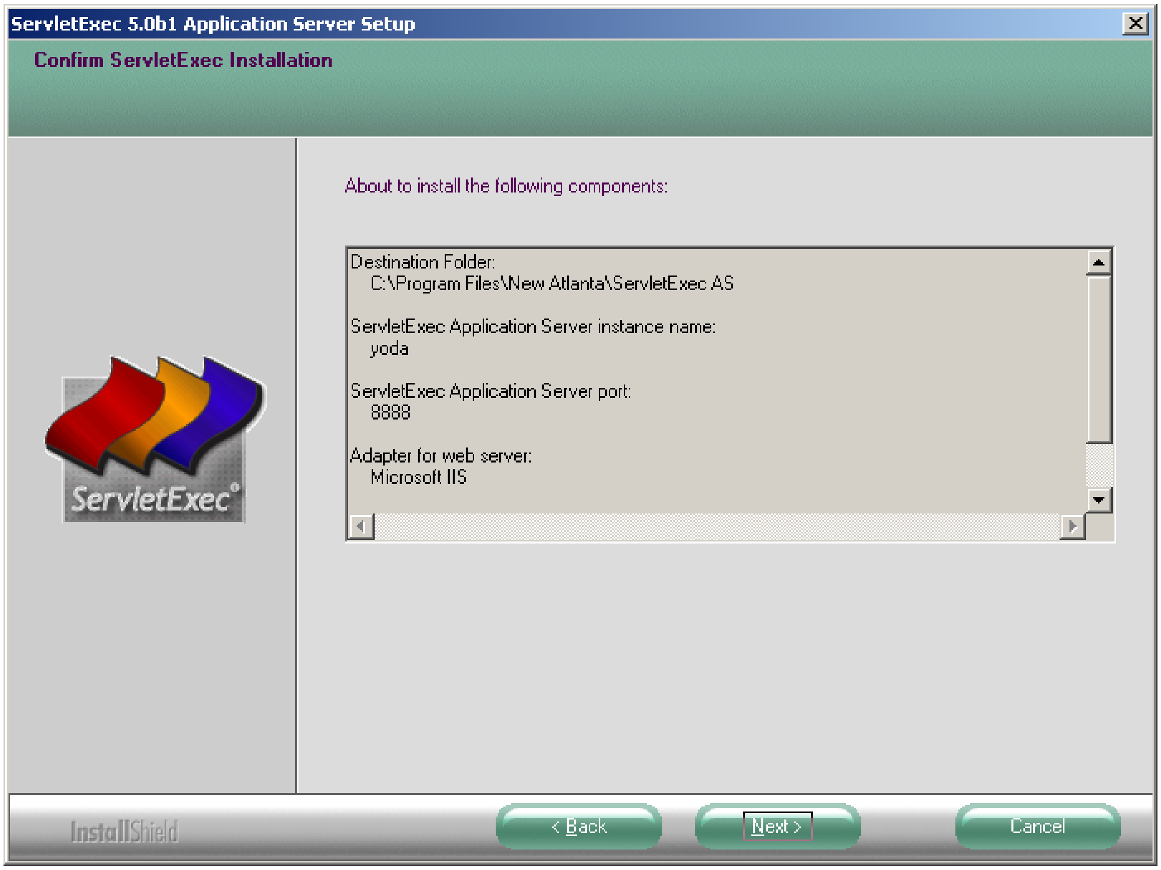
Figure 5. Web Server Adapter

7. Before making any change to your computer, the ServletExec/AS installer displays a confirmation dialog, as represented in Figure 6. Click **Back** to modify any of your selections, or **Cancel** to abort the installation. After clicking **Next**, the ServletExec/AS installer will begin copying files to your computer.

8. Install ServletExec as a Windows Service? The installer will prompt you to install the ServletExec/AS instance as a Windows service. Installing the instance as an Windows service has the following advantages:

   - You will be able to start and stop the ServletExec/AS instance from the **Services** control panel.
   - The ServletExec/AS instance will continue running if you log off Windows.
   - The ServletExec/AS instance can be configured using the **Services** control panel to automatically start when Windows restarts.

   If you choose not to install the ServletExec/AS instance as a Windows Service, you will need to manually start and stop the instance as described in Section 3.7, below.


ServletExec 5.0b1 Application Server Setup
Confirm ServletExec Installation
ServletExec®
InstallShield
< Back
Next >
Cancel

About to install the following components:

Destination Folder:
C:\Program Files\New Atlanta\ServletExec AS

ServletExec Application Server instance name:
yoda

ServletExec Application Server port:
8888

Adapter for web server:
Microsoft IIS

Figure 6. Confirm ServletExec Installation

9. Update the server configuration file. If you're installing a web server adapter for Apache HTTP Server, SunONE, or SJSWS, the ServletExec/AS installer will prompt you to update the server configuration file. The web server configuration file **must** be updated for the ServletExec/AS web server adapter to work properly, so it is recommended that you allow the installer to do so. If you choose not to allow the installer to update the web server configuration file you must do so manually as described in Section 3.5.

If you install a web server adapter, the ServletExec/AS installer will automatically create the webadapter.properties configuration file. After completing the ServletExec/AS installation, be sure to restart the web server on which you installed the ServletExec/AS web server adapter.

3.5 What Was Installed and/or Modified?

When you installed ServletExec/AS, these changes were made to your system:

- The `"ServletExec AS"` directory was created
- ServletExec/AS registry entries were created or modified
- The web server configuration file was modified to support the ServletExec/AS web server adapter

The following sections describe each of these changes.

3.5.1 The ServletExec AS Directory

The `ServletExec AS` directory was created in the location you selected during the installation process. The default location suggested by the installer is:

```
C:\Program Files\New Atlanta\ServletExec AS
```

Important

Do not move the `ServletExec AS` directory after installation. If you move this directory, ServletExec will be unable to find its configuration files.

There are no restrictions on the location of the `ServletExec AS` directory.

The `"ServletExec AS"` directory contains the following subdirectories:

`bin`

The `bin` subdirectory contains the native code web server adapters used by ServletExec/AS. It also contains utility programs for installing and uninstalling a ServletExec/AS instance as a Windows service.

Important

Do not modify the contents of this directory!

`config`

The `config` directory contains the webadapter.properties file.

`Documentation`

Contains the ServletExec User Guide, and the Servlet 2.5 and JSP 2.1 API documentation from Sun Microsystems.

`Examples`

This subdirectory contains examples of a web application and a JSP page with an embedded bean.

See the appropriate chapters of the ServletExec User Guide for instructions on running these examples.

`ExampleServlets`

The `ExampleServlets` subdirectory contains example servlets that are copied to the `Servlets` subdirectory for each ServletExec/AS instance as it is installed. The example servlets are here just for use by the installer.

`se-<instance-name>`

There is one `se-<instance-name>` subdirectory for each ServletExec/AS Instance [ASI] that is installed on the computer.

See further discussion of this directory directly below.

`installerFiles`

The `installerFiles` subdirectory contains files used by the ServletExec/AS setup program for installing multiple ServletExec/AS instances. This directory and its contents are only used by the installer.

`lib`

The `lib` subdirectory contains the Java archive (.jar) files required by ServletExec/AS.

Important

Do not modify the contents of this directory!

The `se-<instance-name>` subdirectories of the `ServletExec AS` directory each contain the following subdirectories.

`classes`

The classes subdirectory is automatically added to the ServletExec VM classpath. Place class files that will be shared by multiple web applications in this directory.

See the READ ME file within the classes subdirectory for more information.

`ServletLogs`

This subdirectory contains the `Servlet.log` files that capture the output from the servlet `log()` method. In some cases, this directory isn't created until the first ServletExec initialization.

`ServletExecData`

Contains the ServletExec/AS configuration files. These files are generally modified using the ServletExec

Admin UI, but can also be edited manually.

`Servlets`

This subdirectory contains the class files for legacy servlets hosted by ServletExec/ISAPI. These servlets run under the context of the default web application.

See the READ ME in the Servlets directory and the ServletExec User Guide for more information.

`webapps`

This directory contains a subdirectory for each configured virtual server. A web application can be automatically deployed to a virtual server by placing it in the subdirectory for that virtual server as a WAR file or open directory structure.

See the READ ME in the webapps directory and the ServletExec User Guide for more information.

3.5.2 Registry Entries

The ServletExec/AS registry entries are stored under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\New Atlanta Communications\ServletExec AS
```

Important

These registry entries are modified by the ServletExec/AS installer and uninstaller, and should not be edited manually.

3.5.3 Microsoft IIS

If you installed a web server adapter for Microsoft IIS, the installer modified your system as described in the following sections.

3.5.3.1 ServletExec_Adapter.dll and the Virtual Directory [VD] within IIS

The `ServletExec_Adapter.dll` dynamic link library (DLL) was installed in the `C:\InetPub\Scripts` directory, which is mapped to the `Scripts` virtual directory [VD] in the default IIS installation; or if the `Scripts` VD was not defined then `ServletExec_Adapter.dll` was copied to the directory you selected during the installation process, and a VD named Scripts (which points there) was created by the installer. The directory in which `ServletExec_Adapter.dll` resides **must** be mapped to a Microsoft IIS VD and that VD **must** have `execute` permission enabled. VD’s for Microsoft IIS are configured using the Microsoft Management Console (Internet Service Manager).

If you are using IIS 6 or IIS 7 you will find that your IIS does not have a VD named Scripts by default. The ServletExec AS installer will create that VD for you in the IIS website named “Default Website”. If you ever have need to create (or re-create) that VD manually with IIS 6 or IIS 7 then you will need to read section 2.5.3 of this document and see the screenshots given in that section as they pertain to both SE ISAPI and SE AS.

3.5.3.2 Metabase ISAPI Filter Entry

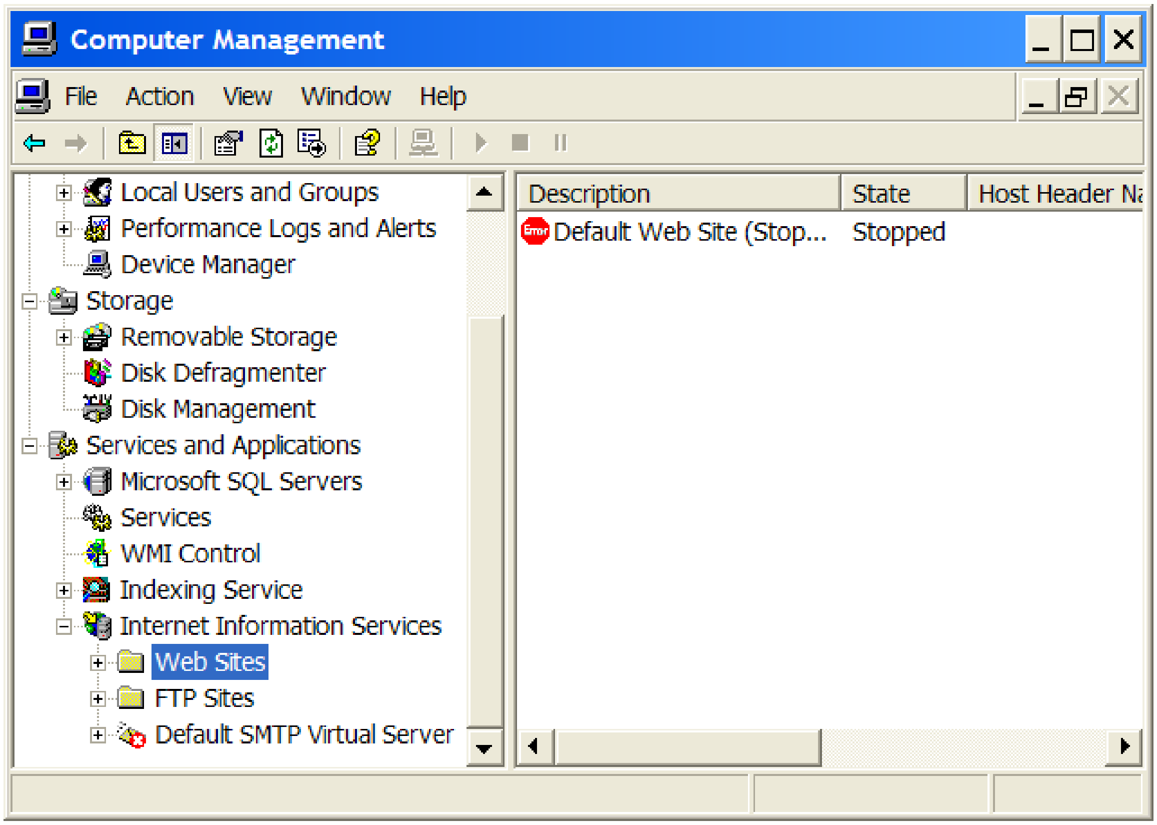
The ServletExec/AS installer automatically modifies the IIS 7.0/6.0/5.1/5.0 (Windows 2008/Vista/2003/XP/2000) metabase to add the ISAPI Filter entry.

Very Important

You **must** remove entries for other servlet engines you may have previously installed. The uninstallers for most servlet engines do not automatically remove the metabase ISAPI Filter entry.

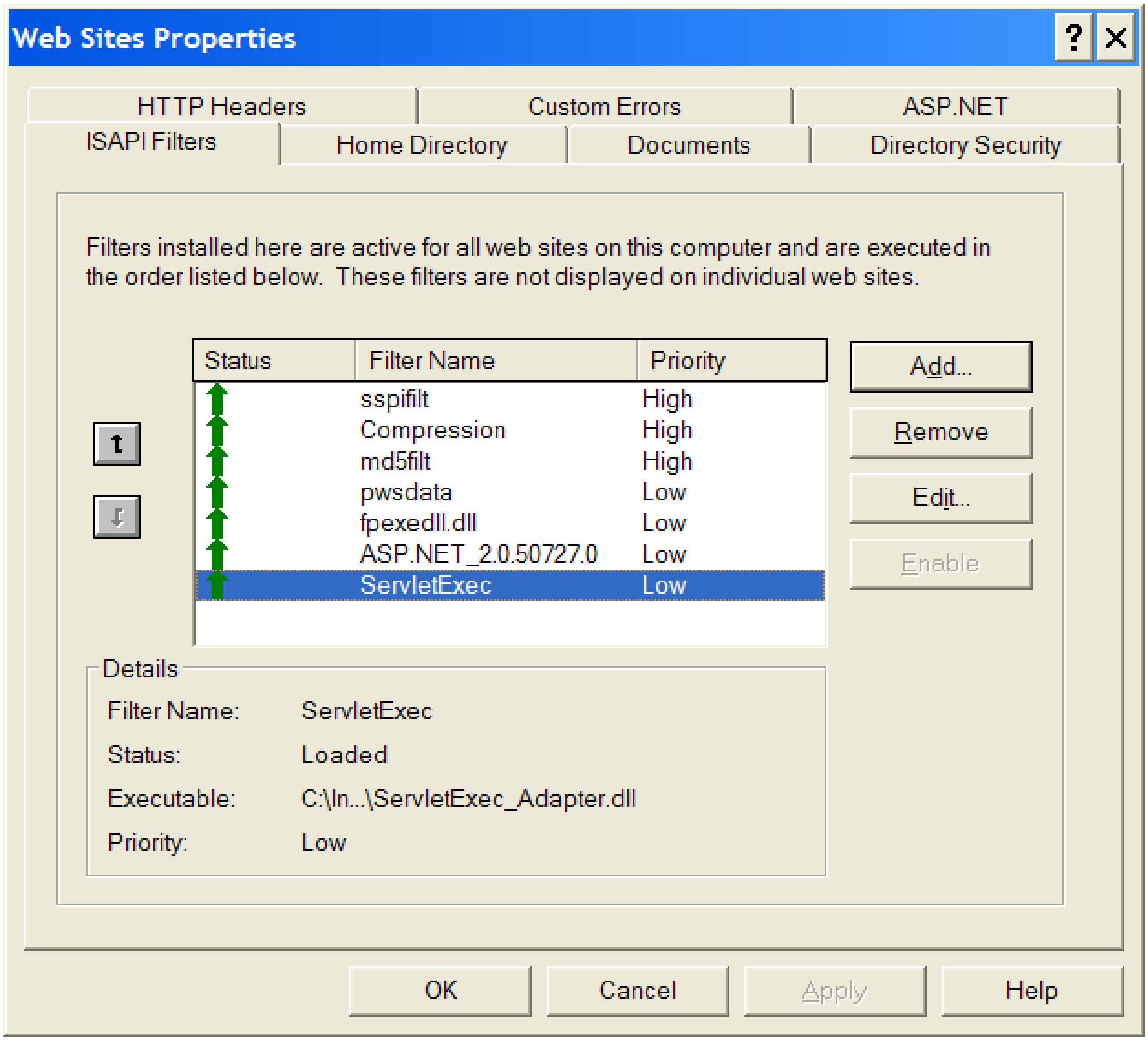
To examine or manually modify the metabase ISAPI Filter under IIS 5.1/5.0

1. Right-click on **My Computer** and choose **Manage**.
2. Expand **Services & Applications** (lower left).
3. Expand **Internet Information Services**.
4. Right-click on **Websites** and choose **Properties** (*see* Figure 14).
5. Click the **ISAPI Filters** tab (*see* Figure 15).



Description	State	Host Header Na
Default Web Site (Stop...	Stopped	

Figure 14. "Web Sites" Properties (IIS 5.1)



Filters installed here are active for all web sites on this computer and are executed in the order listed below. These filters are not displayed on individual web sites.

Status	Filter Name	Priority
⬆	sspifilt	High
⬆	Compression	High
⬆	md5filt	High
⬆	pwsdata	Low
⬆	fpexedll.dll	Low
⬆	ASP.NET_2.0.50727.0	Low
⬆	ServletExec	Low

Details

Filter Name: ServletExec

Status: Loaded

Executable: C:\In...\ServletExec_Adapter.dll

Priority: Low

Figure 15. "Web Sites" Properties (IIS 5.1)

6. Click any previously installed servlet engines, and then click **Remove**.
7. Click **ServletExec,** and then click **Edit…** to examine or modify it (if you wish).
8. If you do not see **ServletExec**, and you are using IIS 5.1/5.0 (Windows XP/2000) then click **Add** and follow the instructions to add the ServletExec/AS entry, making sure to point to `ServletExec_Adapter.dll`.

Note that with IIS 6 you should **not** modify the location of the SE DLL here since IIS 6 is also "pointing" to that same DLL via its Web Service Extensions and the Web Service Extensions are not modifiable. Both the Web Service Extensions and the ISAPI Filters must be pointing to the same exact copy of the DLL.

3.5.3.3 The `webadapter.properties` Configuration File

The `webadapter.properties` file was created in the config subdirectory of your ServletExec AS 6.0 installation. This file contains the configuration properties used by the ServletExec adapter. These properties are used by the adapter for routing HTTP requests from the web server to a ServletExec Application Server (AS) instance.

Important

The IIS adapter (ServletExec_Adapter.dll) will first look in the directory where it resides for the webadapter.properties file. If the file is not found, then it will look in the config directory of your ServletExec installation. You'll only want to do this if you need a separate copy of the webadapter.properties file for different IIS websites.

With Apache, SunONE, & SJSWS Web Servers you can specify the location of the webadapter.properties file if needed.

You can find more information about the webadapter.properties configuration file in section 3.6 of this document.

3.5.4 SunONE / SJSWS

Before installing ServletExec behind a SunONE/SJSWS instance, you must first disable the built-in Servlet/JSP Engine that is bundled with it. You must do this for each SunONE/SJSWS instance with which you intend to use ServletExec.

You would use the browser-based SunONE/SJSWS Administration tool to disable Sun's built-in Servlet/JSP Engine. For SunONE this is a fairly straightforward thing to do. Just be certain to **Apply** the configuration changes so that your server configuration files (obj.conf and magnus.conf) will be rewritten with the built-in engine directives removed.

For SJSWS you will also need to apply/save the changes, however since SJSWS provides clustering support, there are some additional complexities regarding obj.conf of which you need to be aware. Section 3.5.4.1 attempts to describe further.

3.5.4.1 Using ServletExec/AS with SJSWS in a Clustered Environment

Here are some tips/steps for disabling Sun's built-in Servlet/JSP Engine with SJSWS as well as some clustering caveats to know about:

- To disable the built-in Servlet Engine so that SE can be installed, access the SJSWS Admin UI and go to the "Common Tasks" page. Select the desired configuration from the drop-down list. Then click on "Edit Java Settings". Uncheck the "Enable Java" checkbox and then click the "Save" button on the right. Then look above that "Save" button and you'll now see a link labeled "Deployment Pending". If you click that link you'll get a new browser window that lets you deploy/propagate those changes to all instances running in all nodes.

  This will cause the SJSWS instances (i.e. configurations) that are defined on the effected nodes to no longer use obj.conf as their configuration file but instead use a newly created/updated file whose name is:

  <configuration-name>-obj.conf

  So a configuration whose name is "machineA-01" would then use a file named machineA-01-obj.conf rather than obj.conf. This means that ServletExec Directives would need to be added to machineA-01-obj.conf (not obj.conf).

The SE installer will take care of detecting this condition and will add the directives to the correct .conf file.

The original obj.conf will still contain directives for the SJSWS's built-in Servlet/JSP engine. Directives such as:

```
NameTrans fn="ntrans-j2ee" name="j2ee"
PathCheck fn="find-index-j2ee"
ObjectType fn="type-j2ee"
Error fn="error-j2ee"
```

while the new file will not contain these directives.

- The SJSWS Admin UI provides a page that lists the Configurations. If the state of a given Configuration is "Deployment Pending" then it is possible to "deploy the current saved configuration to all nodes".

  Be careful if/when you do this as it's possible to overwrite ServletExec directives for some or all of your SJSWS instances, thus forcing you to re-enter them.

- For the remainder of this document, we will refer to the config file as merely obj.conf

3.5.4.2 Server Configuration Files

To install a ServletExec/AS web server adapter for SunONE/SJSWS, modifications must be made to the web server's obj.conf and magnus.conf files. This section describes the modifications to these files made by the ServletExec/AS installer.

Important

If you chose not to allow the installer to make these modifications, you must make them manually.

Several lines must be added to these configuration files for each server for which ServletExec/AS is installed (the location of these lines within these files is very important):

1. Add the following lines to the beginning of `magnus.conf` before the other `Init` directives:

```
Init    fn=”load-modules”
        shlib="<path>/bin/nsapi/ServletExec_Adapter.dll"
        funcs="ServletExecInit, ServletExecFilter, ServletExecService"
Init    fn=”ServletExecInit”  configFile="<path>/config
        /webadapter.properties"
```

where `<path>` is the full path to the ServletExec/AS installation directory; by default, this is `C:\Program Files\New Atlanta\ServletExec AS`.

Note:

The first `Init` directive will normally appear on a single line within the `magnus.conf` file. It’s shown as spanning 2 lines here for formatting reasons. It may span 2 lines within `magnus.conf`, in which case the second line must begin with a tab or space character). In general, the format of the second `Init` directive is:

```
Init fn=ServletExecInit configFile="<path>/config
    /webadapter.properties""
```

where the value of configFile is the location of the webadapter.properties configuration file. `<path>` contains the location of your ServletExec/AS installation, and the file is put in the config directory by default. You can modify this attribute to point to any location you desire.

Note

See Section 3.6 of the ServletExec Installation guide for more information on the webadapter.properties.

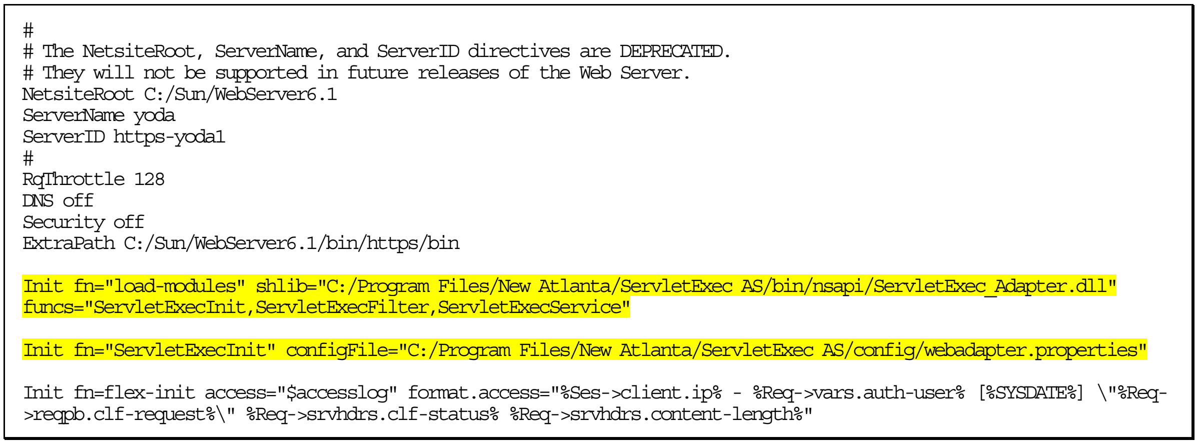
2. Lines similar to the following must be added to `obj.conf` within the `<Object name=default>` directives:

```
    NameTrans fn="ServletExecFilter"
    Service method=(GET|HEAD|POST) type=magnus-internal/nac
    fn="ServletExecService"
```

Figures 16 & 17 show complete magnus.conf and obj.conf files for the SunONE Web Server 6.1 with the ServletExec/AS directives highlighted.

Important

You must stop and restart SunONE/SJSWS web server after modifying the configuration files.



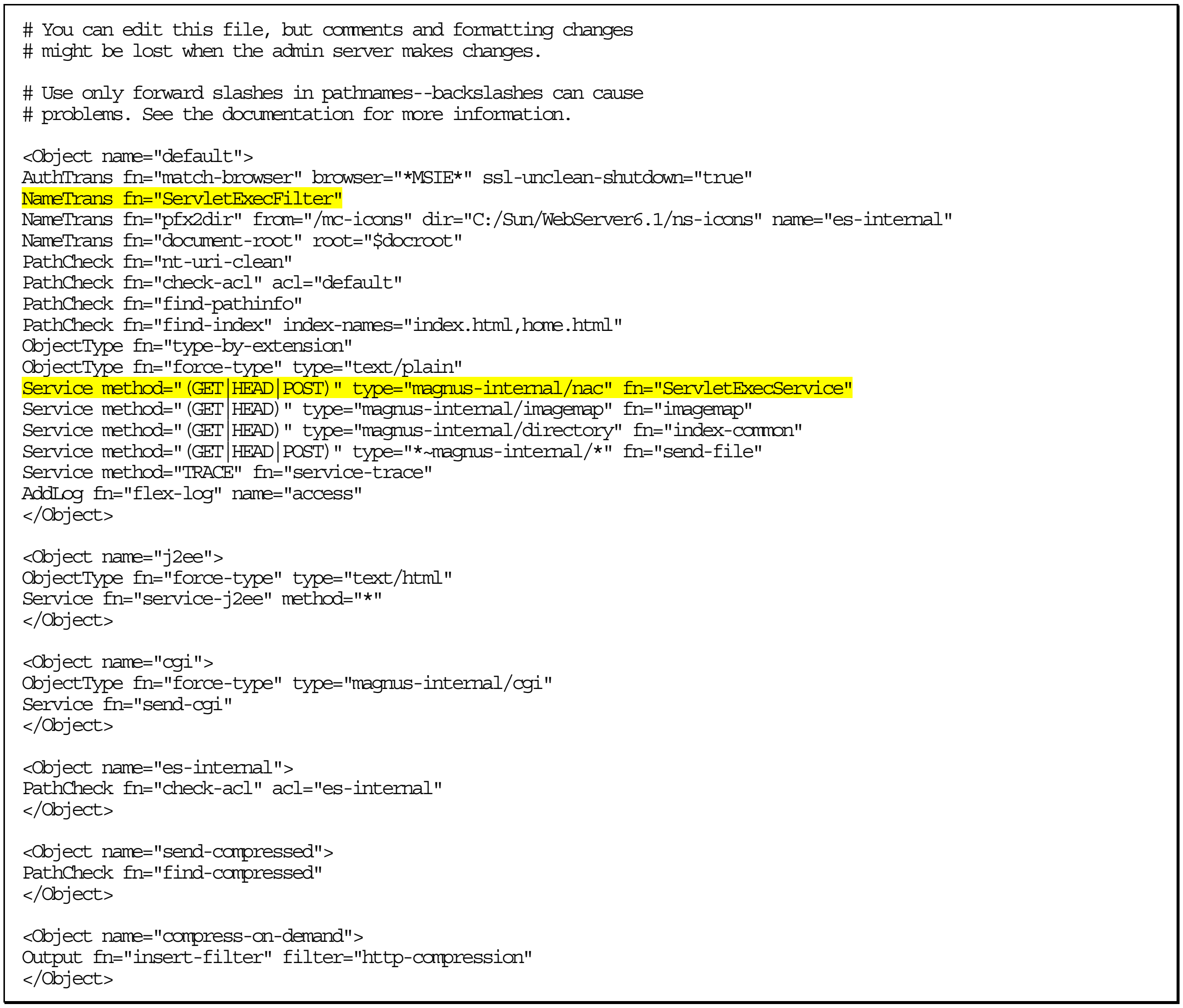
```
#
# The NetsiteRoot, ServerName, and ServerID directives are DEPRECATED.
# They will not be supported in future releases of the Web Server.
NetsiteRoot C:/Sun/WebServer6.1
ServerName yoda
ServerID https-yoda1
#
RqThrottle 128
DNS off
Security off
ExtraPath C:/Sun/WebServer6.1/bin/https/bin

Init fn="load-modules" shlib="C:/Program Files/New Atlanta/ServletExec AS/bin/nsapi/ServletExec_Adapter.dll"
funcs="ServletExecInit,ServletExecFilter,ServletExecService"

Init fn="ServletExecInit" configFile="C:/Program Files/New Atlanta/ServletExec AS/config/webadapter.properties"

Init fn=flex-init access="$accesslog" format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] \"%Req-
>reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

Figure 16. `magnus.conf` **Configuration File for SunONE Web Server 6.1**



```
# You can edit this file, but comments and formatting changes
# might be lost when the admin server makes changes.

# Use only forward slashes in pathnames--backslashes can cause
# problems. See the documentation for more information.

<Object name="default">
AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
NameTrans fn="ServletExecFilter"
NameTrans fn="pfx2dir" from="/mc-icons" dir="C:/Sun/WebServer6.1/ns-icons" name="es-internal"
NameTrans fn="document-root" root="$docroot"
PathCheck fn="nt-uri-clean"
PathCheck fn="check-acl" acl="default"
PathCheck fn="find-pathinfo"
PathCheck fn="find-index" index-names="index.html,home.html"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"
Service method="(GET|HEAD|POST)" type="magnus-internal/nac" fn="ServletExecService"
Service method="(GET|HEAD)" type="magnus-internal/imagemap" fn="imagemap"
Service method="(GET|HEAD)" type="magnus-internal/directory" fn="index-common"
Service method="(GET|HEAD|POST)" type="*~magnus-internal/*" fn="send-file"
Service method="TRACE" fn="service-trace"
AddLog fn="flex-log" name="access"
</Object>

<Object name="j2ee">
ObjectType fn="force-type" type="text/html"
Service fn="service-j2ee" method="*"
</Object>

<Object name="cgi">
ObjectType fn="force-type" type="magnus-internal/cgi"
Service fn="send-cgi"
</Object>

<Object name="es-internal">
PathCheck fn="check-acl" acl="es-internal"
</Object>

<Object name="send-compressed">
PathCheck fn="find-compressed"
</Object>

<Object name="compress-on-demand">
Output fn="insert-filter" filter="http-compression"
</Object>
```

Figure 17. `obj.conf` Configuration File for SunONE Web Server 6.1

3.5.5 Apache HTTP Server

If you installed a web server adapter for Apache HTTP Server, the installer modified your system as described in the following sections.

3.5.5.1 ApacheModuleServletExec.dll

The ServletExec/AS web server adapter for Apache HTTP Server is implemented by the `ApacheModuleServletExec.dll` library, which is copied by the ServletExec/AS installer to the `modules` subdirectory of the Apache installation directory.

3.5.5.2 Server Configuration File (httpd.conf)

To install a ServletExec/AS web server adapter for Apache, modifications must be made to the web server’s `httpd.conf` file. This section describes the modifications to `httpd.conf` made by the ServletExec/AS installer.

Important

If you chose not to allow the installer to make these modifications, you must make them manually.

The following directive was added to `httpd.conf` with the other `LoadModule` directives (near the top of the file):

```
LoadModule servletexec_module modules/ApacheModuleServletExec.dll
```

If the `httpd.conf` file contains a `ClearModuleList` directive followed by `AddModule` directives, then the following directive was added for ServletExec/AS:

```
AddModule mod_servletexec.c
```

These directives cause Apache to load the ServletExec/AS web server adapter.

The following line was added to the end of the `httpd.conf` file:

```
ServletExecAdapterConfigFile "C:/Program Files/New
Atlanta/ServletExec AS/config/webadapter.properties"
```

This directive defines the location of the webadapter.properties file used by ServletExec/AS.

If you are using Apache 2.2.x you will likely find that you receive 403 - Access Forbidden responses from Apache when you request any Servlets or JSPs (the SE Admin UI for example). This is due to how Apache 2.2.x is configured out-of-the-box and is easy to fix. Just modify your httpd.conf file to either remove or comment-out the “Deny from all” condition in the default <Directory> tag. Here is an example:

```
<Directory />
Options FollowSymLinks
AllowOverride None
Order deny,allow
# Deny from all
Satisfy all
</Directory>
```

Then save the change, restart Apache, and re-request the resource.

If you not comfortable “opening up” access to your Apache 2.2.x in this manner then you may leave the `Deny from all` as it is, and instead add entries such as this to the end of your httpd.conf file:

# for JSPs that execute in the context of the webapp whose context path is /

```
<Files “*.jsp”>
 Order allow, deny
 Allow from all
</Files>
```

# for servlets that execute in the context of the web-app whose context path is /

```
<Location “/servlet”>
 Order allow, deny
 Allow from all
</ Location>
```

# for the web-app whose context path is `/servletexec` (the SE Admin UI)

```
<Location "/servletexec">
 Order allow, deny
 Allow from all
</ Location>
```

# for a web-app whose context path is `/spaceShuttle`

```
<Location "/spaceShuttle">
 Order allow, deny
 Allow from all
</ Location>
```

NOTE: If you choose this technique, you'll need to add a Location directive for each web-app context that you deploy in ServletExec.

Important

The Apache 2.2.x web server is preconfigured to deny servlet & JSP requests. Configure your Apache 2.2.x as described above to work around that.

See Section 3.6 of the ServletExec Installation guide for more information on the webadpater.properties.

3.6 Configuring webadapter.properties

All ServletExec/AS native adapters can be configured using the webadapter.properties file. The file is always put in the config directory of your ServletExec installation by default, but it can be put in alternate locations. See the information in this document regarding your brand of web server for specifics on specifying an alternate location.

The webadapter.properties file is used by the ServletExec/AS native adapter, no matter which brand of web server is used. The top of the webadapter.properties file contains extensive comments about the various properties that can be set in that file. This section will describe some of the properties that can be controlled from this file. Property names are always set up in the form of:

servletexec.<*instance name*>.<*property name*>

The <*instance name*> portion of the property name is used to "group" sets of properties together. Each "grouping" of properties applies to 1 single SE AS Instance.

Properties in this file are used to modify connection pooling between the web server adapter and the ServletExec AS Instance(s), find ServletExec instances that are set up to handle the incoming requests, and forward the request to instances set up to handle specific hostnames and/or specific web applications that are deployed.

3.6.1.1 Connection Pooling Between the Web Server Adapter and ServletExec

The installer automatically enables connection pooling between the web server adapter and ServletExec, using default settings. The purpose of this connection pooling is to improve performance through better resource management.

You can change the default settings by modifying the configuration file as explained below. The two parameters that provide the connection pooling are: `pool max idle` (sets maximum number of idle connections that can be in the pool… default setting is 10) and `pool increment`. (sets the number of idle connections to be added to the pool when the pool is empty and another request is received… default setting is 5). Once the `pool max idle` limit is reached, freed connections will be disconnected.

A good guideline for changing these parameters from their default settings would be to set the `pool max idle` to your average number of connections, and to set the `pool increment` to the average spike number of connections.

To modify Connection Pooling for ServletExec/AS

- Add the following to the end of the `webadapter.properties` file on two separate lines:

```
servletexec.<instance-name>.pool-increment=<value>
servletexec.<instance-name>.pool-max-idle=<value>
```

3.6.1.2 Hosts

The *hosts* property is used to find all ServletExec AS instances that are configured to process requests for the designated host. The host name is taken from the HOST header field of the HTTP request. The *hosts* property should contain a comma separated list of IP addresses and/or hosts names. The special value `all` can be used to indicate that all hosts should be processed by the specified instance name.

To update the hosts property in `webadapter.properties`

- Use the following form:

  servletexec.<instance-name>.hosts=www.newatlanta.com

3.6.1.1 ServletExec Instances

The *instances* property contains the IP address and port number of 1 (not multiple) ServletExec AS instance. This is used by the adapter for forwarding HTTP requests to that instance. Each instance must contain a unique IP address and port number combination. The default value is 127.0.0.1:8888.

To update the instances property in `webadapter.properties`

- Use the following form:

  servletexec.<instance-name>.instances=127.0.0.1:8888

3.6.1.2 Global Properties

In addition to the per-instance properties there are also a few global properties.

- servletexec.aliasCheckInterval

  The default value is `10` (measured in seconds).
  Setting it to `-1` turns it off, so that there is no request-time polling.
  An example of using this property in webadapter.properties is:

  ```
  servletexec.aliasCheckInterval=20
  ```

  This global property governs how often the adapter will ping/query the SE AS instances (ASIs) that are defined in that file, to ask them if there are any changes to the aliases (webapp context paths, uri mappings for security, filters, servlets, etc...). The adapter stores a local copy of such mappings in what's referred to as an *Alias Cache* (1 alias cache per ASI) and periodically attempts to update that cache (at request time) if warranted.
  If an ASI is not running and the adapter tries to ping/query it, there may be a timeout delay before the socket knows that there is nothing at the
  other end to respond to the ping/query. This is usually 1 second, but may differ depending on OS, and other factors. If there are several ASI's that are not running or are otherwise unreachable then the delay is increased proportionally (1 second delay for each ASI that's down). This situation can be avoided by enabling the multiInstanceMode property (see next bullet).
  For additional details about the aliasCheckInterval property, please see:
  http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=259

- servletexec.multiInstanceMode

  The default value is `disabled`.
  An example of using this property in webadapter.properties is:

  ```
  servletexec.multiInstanceMode=enabled
  ```

  Enabling multi-instance mode will cause the adapter to:
  - only ping an unresponsive ASI every aliasCheckInterval seconds (rather than continually trying to ping it on every request in an attempt to update its outdated/expired alias cache).
  - clear the in-memory alias cache for any ASI that becomes unresponsive.
    This provides better support for using SE in a failover/clustered environment. It allows the request to "filter down" to the other ASI's or the webserver itself rather than giving **Failed To Connect to ServletExec** messages over and over. Just be aware that you may get 404 Not Found responses in this case.

  To confirm that your adapter is running in multi-instance mode, look for the presence of the following line in your adapter's startup messages:
  This adapter will run with multi-instance optimizations.

3.7 ServletExec/AS Operation

A ServletExec/AS instance [ASI] is a standalone Java application that communicates with the web server (through the ServletExec/AS web server adapter) using network sockets.

If you installed an ASI as a Windows service, you can start and stop it using the **Services** control panel. The Windows **Services** mechanism simply calls either the `StartServletExec.bat` file or the `StopServletExec.bat` file when starting or stopping an ASI. Those batch files are located in the `se-<`*`instance-name`*`>` subdirectories of the `ServletExec AS` installation directory. Each ASI has its own copy of those 2 batch files. Do not move or delete these batch files.

To manually install an ASI as a Windows Service (if you did not let the installer do so automatically), look in the `bin` subdirectory of the `ServletExec AS` installation directory and read `NTService READ ME.txt.`

3.7.1 Starting ServletExec/AS

A ServletExec/AS instance may be started as a Windows Service, or by manually invoking the `StartServletExec.bat` file. There are 2 ways to manually invoke that batch file: (1) double-click it, or (2) open a DOS prompt, cd to folder that contains the file, and then use the keyboard to type out `StartServletExec.bat`. This 2nd way is very useful for troubleshooting since if the ASI is having problems starting up, you'll likely see the errors there on the DOS prompt.

3.7.1.1 StartServletExec.bat

If you examine this batch file in a plain text editor you can see that it uses standard *java options*, and also several required and optional *program arguments*. You may edit `StartServletExec.bat` as needed. The next section features a complete list of ServletExec/AS command line arguments.

3.7.1.2 How StartServletExec.bat uses the java Command

The information in this section can help you edit the `StartServletExec.bat` file to modify the startup parameters for a ServletExec/AS instance (if needed).

The general form is:

```
java <java options> ServletExec <ServletExec/AS program arguments>
```

Important

The `java` options must include the `–classpath` option specifying the path to the `servlet-api.jar, jsp-api.jar, ServletExec60.jar, ServletExecAdmin.jar, el-api.jar, jasper-el.jar, jstl.jar, appserv-jstl.jar, activation.jar, mail.jar`

archives.

Start ServletExec/AS arguments

If an option or argument value contains spaces then the entire value must be placed inside double quotes, as in the following example:

```
-name "Local Instance"
```

See the following table for a complete list of available arguments.

Argument	Explanation
`-help`	Displays all available arguments and explanations.
`-port <port number>`	**Optional**. The default value is 8888. This argument specifies the TCP/IP port on which the ServletExec/AS instance communicates with the web server adapter(s). If you modify this value, you must also modify the web server configuration file(s) as described above, and you must also modify the –port value used in the StopServletExec.bat file.
`-backlog <length>`	**Optional**. The default value is 50. This argument specifies the size of the ServletExec/AS incoming request queue. Requests received when the queue is full are rejected. Increasing the size of the request queue may prevent requests from being rejected under heavy loads, but may increase per-request response times.
`-name` *`<instance-name>`*	**Optional**. This is used to specify the name of the ServletExec/AS instance and is used only for display in the Admin UI to identify the instance.
`-home <path to the ServletExec/AS instance home directory>`	**Optional**. The default is the directory from which the `java` command is executed. This specifies the directory in which the ServletExec/AS instance will look for the `ServletExecData`, `ServletLogs`, and `Servlets` subdirectories.
`-log <path to the ServletExec/AS instance log directory>`	**Optional**. This can be used to specify an alternate directory for writing the ServletExec/AS log files. By default, log files are written to the `ServletLogs` subdirectory of the ServletExec/AS home directory as specified by the `–home` argument.
`-root <path to the web server's document root directory>`	**Required**. Specifies the **case-sensitive** path to the web server's document root directory. If using SE virtual servers [VS], the doc root of each SE VS must be configured using a separate `–root`

Argument	Explanation

	argument using the following format:
	`-root <virtual server>=<folder>`
	for example:
	`-root www.abc.com=C:\Apache\htdocs\abcdocs`

`-root <path to the virtual server's document root directory>`	*See previous explanation.*

`-addl <url=directory>`	**Optional**. This can be used to specify additional **case-sensitive** document directories (other than the root) and is specified in two forms:
	`-addl <url>=<physical directory>`
	`-addl <virtual server/url>=<physical directory>`
	For example:
	`-addl /test=C:\Test`
	`-addl www.abc.com/support=C:\Support`

`-addl <virtual server/url=directory>`	*See previous explanation.*
`-mimetypes <path to the web server's mime.types file>`	**Optional**. This specifies the path to the web server's `mime.types` file. If this argument isn't provided, the `ServletContext.getMimeType()` method always returns `null`. (For Microsoft IIS, which doesn't have a `mime.types` file, the `ServletContext.getMimeType()` method always returns `null`).

Argument	Explanation
`-allow <ip1,ip2,…,ipn>`	**Optional**. This specifies the IP address(es) of the web server(s) that are allowed to communicate with the ServletExec/AS instance. Either IPv4 or IPv6 Addresses may be used. IPv4 addresses must be of the form x.x.x.x (where x is in the range 0-255) IPv6 addresses must be of the form [y:y:y:y:y:y:y:y] (where y is a 4-digit hexadecimal number). IP addresses can include the "*" character to indicate a subrange (for example: `168.121.97.*`, [*:*:*:*:*:*:*:*]). If the `–allow` argument is not specified, the ServletExec/AS instance only accepts requests from web servers running on the local machine.

Table 4. Start ServletExec/AS Command Line Arguments

Note: The `–root` and `–addl` arguments are used by the webapp named "default-app" only, and have no effect on other deployed webapps. For more details please see SE FAQ #278: http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=278

3.7.1.3 Closing the DOS Window

When starting a ServletExec/AS instance manually using the `java` command (either using the `StartServletExec.bat` file or from the command line), the DOS window must remain open while the ServletExec/AS instance is running. To be able to close the DOS window, use the `javaw` command instead of `java`. After the ServletExec/AS instance has started, you can close the DOS window and the instance will continue running.

3.7.2 Stopping ServletExec/AS

You must stop a ServletExec/AS instance using the `StopServletExec.bat` file. If a ServletExec/AS instance is not shut down properly, servlet `destroy()` methods are not invoked, sessions are not saved, Web Application Lifecycle Events are not generated, and buffered log messages are lost.

3.7.2.1 StopServletExec.bat

`StopServletExec.bat` invokes the `StopServletExec` Java program to stop the ServletExec/AS instance. That program opens a TCP port and sends a message to the instance, telling it to shutdown.

3.7.2.2 How StopServletExec.bat uses the java Command

The information in this section can help you edit the `StopServletExec.bat` file to modify the shutdown parameters for a ServletExec/AS instance (if needed).

The general form is:

```
java <java options> StopServletExec <program arguments>
```

The only edits you'd likely ever need to make to this file is to change the value of the –`port` argument to match the –`port` value specified in `StartServletExec.bat`, or to change the version of java being used.

Stop ServletExec/AS arguments

If an option/argument value contains spaces then the entire value must be placed inside double quotes, as in the following example:

```
-name "Local Instance"
```

See the following table for a complete list of available arguments.

Argument	Explanation
`-help`	Displays all available arguments and explanations.
`-port`	Specifies the port number used by the ServletExec/AS instance

Table 5. Stop ServletExec/AS Command Line Arguments

3.7.3 Remote Operation

By default, a ServletExec/AS web server adapter is installed on the same machine as the ServletExec/AS instance, and the ServletExec/AS instance (by default) only accepts requests from web server adapters running on the local machine. It is possible to install and configure web server adapters to run on a different machine (a "remote" machine) than the ServletExec/AS instance.

To run a web server adapter on a different machine

1. Edit the `StartServletExec.bat` file for the ServletExec/AS instance to add the following argument at the end of the `java` command line:
   ```
   -allow <ip address>
   ```
   where `<ip address>` is the IP address of the remote machine on which the web server adapter will be installed

   See table 4 in section 3.7.1.2 for a complete discussion of the –allow argument.

2. Run the ServletExec/AS installer on the remote machine on which the web server adapter is to be installed. Choose **Install a web server adapter** when prompted.

   See Section 3.4 for more information.

3. Enter the **name**, **IP address**, and **port number** of the ServletExec/AS instance when prompted.

Note

You can install multiple web server adapters on multiple machines or on a single machine to forward requests to a single (or multiple) ServletExec/AS instance(s).

3.7.4 Multiple ServletExec/AS Instances (Multiple Java VMs)

It’s possible to install and run multiple ServletExec/AS instances on a single server machine (or on unique machines). Each ServletExec/AS instance will run independently of the others, each within its own Java Virtual Machine (VM).

To install multiple ServletExec/AS instances

Run the ServletExec/AS installer again to install another instance. Remember to give each instance a unique name and port number during the installation process.

To configure a web server adapter to forward HTTP requests to more than one instance

It’s possible (but not required) to configure a single web server adapter to forward HTTP requests to more than one ServletExec/AS instance. The adapter will use both the hostname of the request and the context path of the request to decide which (if any) SE AS instance should be handed the request. The hostname portion of the request is matched against the hosts defined in webadapter.properties. The context path portion of the request is matched against the context paths of all webapps deployed on all AS instances whose hostname matched the request.

One option to configure this is to use unique webapp context paths (unique across all SE instances). Another option is to modify the webadapter.properties file to define a unique set of hostnames or IPs [via the .hosts value] for each instance. Yet another option is to use some combination of those 2 options. For details please read
SE FAQ #259:
http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=259

3.8 Uninstalling ServletExec/AS

To uninstall ServletExec/AS or a web server adapter

1. Open Control Panel and click Add/Remove Programs.
2. In the list, click **ServletExec 6.0 AS**, and then click **Add/Remove** or **Change/Remove**, depending on your operating system. This launches the ServletExec installer/uninstaller.
3. Click the appropriate option from the following choices: uninstall a ServletExec/AS instance, uninstall a web server adapter, or completely uninstall ServletExec/AS.
4. If you are uninstalling a web server adapter (or completely uninstalling ServletExec/AS), you must manually edit the web server configuration file(s) (`obj.conf & magnus.conf` for SunONE/SJSWS, or `httpd.conf` for Apache) to remove the ServletExec/AS directives

   If you are completely removing ServletExec/AS from your system, delete the `ServletExec AS` directory.

   See Section 3.5 for more information

4

4. ServletExec/AS: Application Server (UNIX/Linux)

The ServletExec Application Server (ServletExec/AS) is a standalone web application server that hosts Java Servlets and JavaServer Pages (JSP). ServletExec/AS can be used by itself using its built-in web server or in conjunction with Apache HTTP Server, SunONE Web Server, or Sun Java System Web Server [SJSWS]. Web server adapters for these web servers are included in the ServletExec/AS installer.

The ServletExec/AS for UNIX® installer is named `ServletExec_AS_60.sh`. If you haven’t already done so, you can download the installer from:

http://www.newatlanta.com/c/products/servletexec/download/home

The installer is a binary file (not an ASCII file), so be certain it is treated as such by your browser or any other file transfer software you may use. Otherwise it will become corrupted.

This chapter contains important information that will allow you to verify your installation of ServletExec/AS for UNIX. It will also be useful should you decide to uninstall ServletExec/AS and need to make sure you’ve completely removed all installed components.

4.1 Upgrading From a Previous Version

Upgrading from a previous version always involves uninstalling the older version and then installing the newer one. There are no updaters that will update your SE installation “in place”. The only question is whether or not you can reuse the old SE configuration files in the newer version of SE. ServletExec 6.0 can use the configuration files from all previous versions of SE except versions 3.0C, 3.0E, and 2.2 (or earlier). If you have any of those older versions installed, you will not be able to reuse the configuration files at all. You’ll need to use the SE Admin UI on the new installation to manually reenter your configuration data.

To upgrade to a new version of ServletExec/AS and maintain your old configuration settings

1. Stop your web server. If you've installed ServletExec/AS adapters for multiple web servers, stop them all.
2. Make backup copies of the `ServletExecData`, `Servlets`, `classes` (if you've added an additional one), and `webapps` (versions 3.1 and above only) directories for all ServletExec/AS instances.

   - For ServletExec/AS 3.0 and later, the default locations are within:

     `/usr/local/NewAtlanta/ServletExecAS/se-<`*`instance-name`*`>`

3. Close all open applications.
4. Uninstall the old version of ServletExec/AS by running the installer for that version. Choose **remove ServletExec/AS completely** when prompted by the installer.
5. Run the ServletExec/AS 6.0 installer. You will need to re-run the installer for each ServletExec/AS instance, installing into the same location as the previous installation..

   Important

   For a successful upgrade, you must install everything into the same directory as the previous version.

6. Copy the `ServletExecData`, `Servlets`, `classes` (if you've added an additional one), and `webapps` (versions 3.1 and above only) directories to the new `ServletExecAS` directory created by the installer in Step 5. In ServletExec/AS 6.0, the default location of the `ServletExecAS` directory is:

   `/usr/local/NewAtlanta/ServletExecAS`

7. Restart your web server.

After restarting your web server, the new version of ServletExec/AS will run using your old configuration settings.

Tip

If you have any problems, you can restore the subdirectories from the backups you made in Step 2.

4.2 System Requirements

4.2.1 Operating System and Web Server

ServletExec/AS 6.0 for UNIX supports the following operating systems and web servers. Again, it supports these web servers using native web adapters which communicate with SE AS Instance(s) via TCP sockets.

OS	Web Server
SPARC Solaris 2.6, 7, 8, 9, & 10	▪ Apache HTTP Server 1.3.4 – 1.3.x 2.0.43 – 2.0.x 2.2.0 – 2.2.x ▪ SunONE 6.1 ▪ Sun Java System Web Server [SJSWS 7.0u1]
HP-UX 11.0, 11i v1.0 (11.11), and 11v2 (11.23)	▪ Apache HTTP Server 1.3.4 – 1.3.x 2.0.43 – 2.0.x 2.2.0 – 2.2.x ▪ SunONE 6.1 ▪ Sun Java System Web Server [SJSWS 7.0u1]
AIX 5.1, 5.2, and 5.3	▪ Apache HTTP Server 1.3.4 – 1.3.x 2.0.43 – 2.0.x 2.2.0 – 2.2.x ▪ SunONE 6.1
Red Hat Linux 7.x, 8.x and 9.x Red Hat Enterprise Linux 3, 4, & 5	▪ Apache HTTP Server 1.3.4 – 1.3.x 2.0.43 – 2.0.x 2.2.0 – 2.2.x ▪ SunONE 6.1 ▪ Sun Java System Web Server [SJSWS 7.0u1]
SuSE 8.2 & 9.1	▪ Apache HTTP Server 1.3.4 – 1.3.x 2.0.43 – 2.0.x 2.2.0 – 2.2.x

Table 6. AS-UNIX Supported Operating Systems and Web Servers

Notes

- In addition to the tested and officially supported UNIX operating system versions, ServletExec/AS should work on any UNIX variant when used with Apache 1.3.4 or higher. This is accomplished by providing the ServletExec web server adapters for the Apache HTTP Server in source-code form and utilizing the DSO capability of Apache.

Because there are a number of potential manufacturer variants and/or release differences in UNIX systems that may cause the installation to fail, ServletExec is not fully supported in these environments. Correcting these types of problems come with issues that are both technical and/or business-related and therefore are not officially supported. If you do have any problems, first contact ServletExec Tech Support at `support@newatlanta.com`. Be sure to include the installation log `/tmp/SEinstall.log` with your email so that the problems may be accurately analyzed.

- In addition to a JDK 1.5 or 1.6 -compliant VM, these installations require an ANSI C compiler (GNU version 2.8.1 or higher preferred. Vendor-specific C compilers are not supported) and a system-specific dynamic shared object loader.

See Section 4.4 for more information on the installation log.

4.2.2 JDK or JRE

ServletExec/AS 6.0 is not supported on JDK or JRE 1.4.x or earlier.

To install ServletExec/AS 6.0, you must first install a JDK 1.5.x, or 1.6.x compliant VM. Visit your operating system vendor’s web site for information about downloading and installing a Java VM.

4.2.3 Web server adapters

ServletExec/AS includes web server adapters for the following web servers on UNIX:

- Apache HTTP Server 2.2.0 – 2.2.x
- Apache HTTP Server 2.0.43 – 2.0.x
- Apache HTTP Server 1.3.4 – 1.3.x
- Sun Java System Web Server [SJSWS 7.0u1]
- SunONE Web Server 6.1

4.2.3.1 Apache HTTP Server

Building the ServletExec/AS web server adapter for Apache requires use of the Apache **apxs** utility, which in turn requires that you have **perl**, a **C** compiler, and a dynamic shared linker installed prior to running the installer.

Apache HTTP Server must be built with Dynamic Shared Object (DSO) support enabled before installing a ServletExec/AS web server adapter.

Important

If you have previously built Apache without DSO enabled, you'll also need to rebuild Apache's **apxs** utility. Perform the following steps to rebuild Apache and **apxs** with DSO enabled:

```
# cd <apache source directory>
# rm src/support/apxs
# ./configure --prefix=/usr/local/apache --enable-module=so
# make
# make install
```

4.3 Uninstalling Other Servlet Engines

Very Important

You **must** uninstall any other servlet engines previously installed before installing and using ServletExec/AS. In particular, modifications to the `obj.conf, magnus.conf,` and `httpd.conf` configuration files for other servlet engines must be removed.

- For SunONE Web Server and SJSWS, make sure the internal servlet and JSP support is disabled.
  See Section 3.5.4 for information on disabling Sun's servlet engine.
- For Apache HTTP Server, any directives associated with other servlet engines in `httpd.conf` must be removed.
  See Section 3.5.5.2 for more information on `httpd.conf` entries.

4.4 Running the ServletExec/AS Installer

The file `ServletExec_AS_60.sh` is a non-graphical installation shell script. It is a binary file and must be treated as such. For more details about that please see SE FAQ #336: http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=336

To use the ServletExec/AS Installer

1. Give the file `execute` permission.

   ```
   # chmod +x ServletExec_AS_60.sh
   ```

2. From a shell command line, run it as follows:

   ```
   # ./ServletExec_AS_60.sh
   ```

3. Review the ServletExec License Agreement carefully and check the README displayed by the installer for any last minute notices before continuing.

4. Continue following the prompts and instructions until you see confirmation of a successful installation.

You will be able to see a record of the completed installation steps and errors in the log file `/tmp/SEinstall.log`. Should you have any problems with your installation, please include this log file in a message to the ServletExec Technical Support at support@newatlanta.com.

Note

The log file is overwritten each time the installer runs.

4.5 What Was Installed and/or Modified?

When you installed ServletExec/AS, these changes were made to your system:

- The `ServletExecAS` directory was created.
- The web server configuration file(s) were updated to support the ServletExec/AS web server adapter.

The following sections describe each of these changes.

4.5.1 The ServletExecAS Directory

The `ServletExecAS` directory was created within the `/usr/local/NewAtlanta` directory, or the directory you specified during the installation.

Important

Do not move the `ServletExecAS` directory after installation. If you move this directory, ServletExec/AS will be unable to find its configuration files.

There are no restrictions on the location of the `ServletExecAS` directory.

The `ServletExecAS` directory contains the following subdirectories:

`bin`

The `bin` subdirectory contains the native code web server adapters used by ServletExec/AS.

Important

Do not modify the contents of this directory!

`config`

The `config` directory contains the webadapter.properties file.

`Documentation`

This subdirectory contains the ServletExec User Guide, and the Servlet 2.5 and JSP 2.1 API documentation from Sun Microsystems.

`Examples`

This subdirectory contains examples of a web application and a JSP page with an embedded bean.

See the appropriate chapters of the ServletExec User Guide for instructions on running these examples.

`se-<instance-name>`

There is one `se-<instance-name>` subdirectory for each ServletExec/AS installed on the computer.

See further discussion of this directory directly below.

`installerFiles`

The `installerFiles` subdirectory contains files used by the ServletExec/AS setup program for installing multiple ServletExec/AS instances. This directory and its contents are only used by the installer.

`lib`

The `lib` subdirectory contains the Java archive (.jar) files required by ServletExec/AS.

Important

Do not modify the contents of this directory!

The `se-<instance-name>` subdirectories of the `ServletExecAS` directory each contain the following subdirectories.

`classes`

The classes subdirectory is automatically added to the ServletExec VM classpath. Place class files that will be shared by multiple web applications in this directory.

See the READ ME file within the classes subdirectory for more information.

`ServletLogs`

This subdirectory contains the `Servlet.log` files that capture the output from the servlet `log()` method. In some cases, this directory doesn't get created until the first ServletExec initialization.

`ServletExecData`

This subdirectory contains the ServletExec/AS configuration files. These files are generally modified using the ServletExec Admin UI, but can also be edited manually.

`Servlets`

This subdirectory contains the class files for legacy servlets hosted by ServletExec/AS. These servlets run under the context of the default web application.

See the READ ME in the Servlets directory and the ServletExec User Guide for more information.

`webapps`

This directory contains a subdirectory for each configured virtual server. A web application can be automatically deployed to a virtual server by placing it in the subdirectory for that virtual server as a WAR file or open directory structure.

See the READ ME in the webapps directory and the ServletExec User Guide for more information.

4.5.2 SunONE / SJSWS

Before installing ServletExec behind a SunONE/SJSWS instance, you must first disable the built-in Servlet/JSP Engine that is bundled with it. You must do this for each SunONE/SJSWS instance with which you intend to use ServletExec.

You would use the browser-based SunONE/SJSWS Administration tool to disable Sun's built-in Servlet/JSP Engine. For SunONE this is a fairly straightforward thing to do. Just be certain to **Apply** the configuration changes so that your server configuration files (obj.conf and magnus.conf) will be rewritten with the built-in engine directives removed.

For SJSWS you will also need to apply/save the changes, however since SJSWS provides clustering support, there are some additional complexities regarding obj.conf of which you need to be aware. Section 4.5.2.1 attempts to describe further.

4.5.2.1 Using ServletExec/AS with SJSWS in a Clustered Environment

Here are some tips/steps for disabling Sun's built-in Servlet/JSP Engine with SJSWS as well as some clustering caveats to know about:

- To disable the built-in Servlet Engine so that SE can be installed, access the SJSWS Admin UI and go to the "Common Tasks" page. Select the desired configuration from the drop-down list. Then click on "Edit Java Settings". Uncheck the "Enable Java" checkbox and then click the "Save" button on the right. Then look above that "Save" button and you'll now see a link labeled "Deployment Pending". If you click that link you'll get a new browser window that lets you deploy/propagate those changes to all instances running in all nodes.

  This will cause the SJSWS instances (i.e. configurations) that are defined

  on the effected nodes to no longer use obj.conf as their configuration file but instead use a newly created/updated file whose name is

  <configuration-name>-obj.conf

  So a configuration whose name is "machineA-01" would then use a file named machineA-01-obj.conf rather than obj.conf. This means that ServletExec Directives would need to be added to machineA-01-obj.conf (not obj.conf).

  The SE installer will take care of detecting this condition and will add the directives to the correct .conf file.

  The original obj.conf will still contain directives for the SJSWS's built-in Servlet/JSP engine. Directives such as:

  ```
  NameTrans fn="ntrans-j2ee" name="j2ee"
  PathCheck fn="find-index-j2ee"
  ObjectType fn="type-j2ee"
  Error fn="error-j2ee"
  ```

  While the new file will not contain those directives.

- The SJSWS Admin UI provides a page that lists the Configurations. If the state of a given Configuration is "Deployment Pending" then it is possible to "deploy the current saved configuration to all nodes".

  Be careful if/when you do this as it's possible to overwrite ServletExec directives for some or all of your SJSWS instances, thus forcing you to re-enter them.

- For the remainder of this document, we will refer to the config file as merely obj.conf

4.5.2.2 Server Configuration Files

To install a ServletExec/AS web server adapter for SunONE/SJSWS, modifications must be made to the web server's obj.conf and magnus.conf files. This section describes the modifications to these files made by the ServletExec/AS installer.

Important

If you chose not to allow the installer to make these modifications, you must make them manually.

Several lines must be added to these configuration files for each server for which ServletExec/AS is installed (the location of these lines within these files is very important):

5. Add the following lines to the beginning of `magnus.conf` before the other `Init` directives:

```
Init   fn=”load-modules”
       shlib="<path>/bin/nsapi/ServletExec_Adapter.so"
       funcs="ServletExecInit, ServletExecFilter, ServletExecService"
Init   fn=”ServletExecInit”  configFile="<path>/config
       /webadapter.properties"
```

   where `<path>` is the full path to the ServletExec/AS installation directory; by default, this is `C:\Program Files\New Atlanta\ServletExec AS`.

   Note:

   The first `Init` directive will normally appear on a single line within the `magnus.conf` file. It’s shown as spanning 2 lines here for formatting reasons. It may span 2 lines within `magnus.conf`, in which case the second line must begin with a tab or space character). In general, the format of the second `Init` directive is:

```
Init fn=ServletExecInit configFile=”<path>/config
    /webadapter.properties"”
```

   where the value of configFile is the location of the webadapter.properties configuration file. *<path>* contains the location of your ServletExec/AS installation, and the file is put in the config directory by default. You can modify this attribute to point to any location you desire.

   Note

   See Section 4.6 of this guide for more information on the webadapter.properties.

6. Lines similar to the following must be added to `obj.conf` within the `<Object name=default>` directives:

```
NameTrans fn="ServletExecFilter”
Service method=(GET|HEAD|POST) type=magnus-internal/nac
fn="ServletExecService"
```

Important

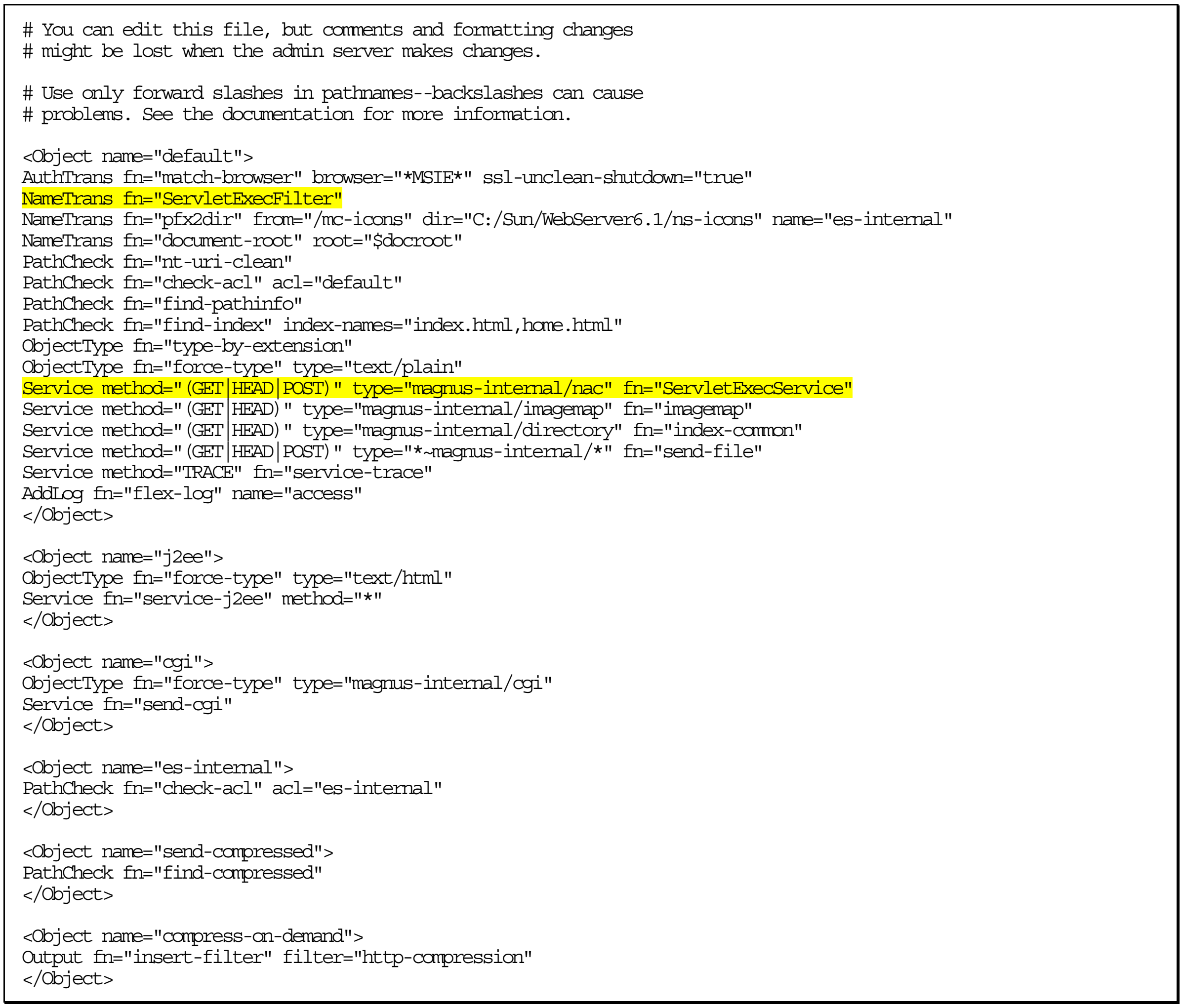
You must stop and restart SunONE/SJSWS web server after modifying the configuration files.

Figure 7 and Figure 8 show complete magnus.conf and obj.conf files for the SunONE Web Server 6.1 with the ServletExec/AS directives highlighted.

```
#
# The NetsiteRoot, ServerName, and ServerID directives are DEPRECATED.
# They will not be supported in future releases of the Web Server.
NetsiteRoot /export/home/sunone61
ServerName sunone61
ServerID https-sunone61
#
RqThrottle 128
DNS off
Security off
PidLog /export/home/sunone61/https-sunone61/logs/pid
User root
StackSize 131072
TempDir /tmp/https-SE-6.0rc4-NSAPI-86a2d0cf

Init fn="load-modules" shlib="/export/home/NewAtlanta/ServletExecAS/bin/ServletExec_Adapter.so"
funcs="ServletExecInit,ServletExecFilter,ServletExecService"
Init fn="ServletExecInit" configFile="/export/home/NewAtlanta/ServletExecAS/config/webadapter.properties"
Init fn=flex-init access="$accesslog" format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] \"%Req-
>reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

Figure 7. `magnus.conf` **Configuration file for SunONE Web Server 6.1**



```
# You can edit this file, but comments and formatting changes
# might be lost when the admin server makes changes.

# Use only forward slashes in pathnames--backslashes can cause
# problems. See the documentation for more information.

<Object name="default">
AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
NameTrans fn="ServletExecFilter"
NameTrans fn="pfx2dir" from="/mc-icons" dir="C:/Sun/WebServer6.1/ns-icons" name="es-internal"
NameTrans fn="document-root" root="$docroot"
PathCheck fn="nt-uri-clean"
PathCheck fn="check-acl" acl="default"
PathCheck fn="find-pathinfo"
PathCheck fn="find-index" index-names="index.html,home.html"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"
Service method="(GET|HEAD|POST)" type="magnus-internal/nac" fn="ServletExecService"
Service method="(GET|HEAD)" type="magnus-internal/imagemap" fn="imagemap"
Service method="(GET|HEAD)" type="magnus-internal/directory" fn="index-common"
Service method="(GET|HEAD|POST)" type="*~magnus-internal/*" fn="send-file"
Service method="TRACE" fn="service-trace"
AddLog fn="flex-log" name="access"
</Object>

<Object name="j2ee">
ObjectType fn="force-type" type="text/html"
Service fn="service-j2ee" method="*"
</Object>

<Object name="cgi">
ObjectType fn="force-type" type="magnus-internal/cgi"
Service fn="send-cgi"
</Object>

<Object name="es-internal">
PathCheck fn="check-acl" acl="es-internal"
</Object>

<Object name="send-compressed">
PathCheck fn="find-compressed"
</Object>

<Object name="compress-on-demand">
Output fn="insert-filter" filter="http-compression"
</Object>
```

Figure 8. `obj.conf` **Configuration file for SunONE Web Server 6.1**

4.5.3 Apache HTTP Server

If you installed a web server adapter for Apache, changes were made to your system as described in the following sections.

4.5.3.1 mod_servletexec.so

The `mod_servletexec.so` library implements the ServletExec/AS web server adapter for Apache HTTP Server. It does so after the ServletExec/AS installer copies it to the `libexec` subdirectory of the Apache installation directory.

4.5.3.2 Server Configuration File (httpd.conf)

To install a ServletExec/AS web server adapter for Apache, modifications must be made to the web server's `httpd.conf` file. This section describes the modifications to `httpd.conf` made by the ServletExec/AS installer.

Important

If you chose not to allow the installer to make these modifications, you must make them manually.

The following directive was added to `httpd.conf` with the other `LoadModule` directives (near the top of the file):

```
LoadModule servletexec_module modules/mod_servletexec.so
```

If the `httpd.conf` file contains a `ClearModuleList` directive followed by `AddModule` directives, then the following directive was added for ServletExec/AS:

```
AddModule mod_servletexec.c
```

These directives cause Apache to load the ServletExec/AS web server adapter.

The following line was added to the end of the `httpd.conf` file:

```
ServletExecAdapterConfigFile
"/usr/local/NewAtlanta/ServletExecAS/config/webadapter.properties”
```

This directive defines the location of the webadapter.properties file used by ServletExec/AS.

If you are using Apache 2.2.x you will likely find that you receive 403 - Access Forbidden responses from Apache when you request any Servlets or JSPs (the SE Admin UI for example). This is due to how Apache 2.2.x is configured out-of-the-box and is easy to fix. Just modify your httpd.conf file to either remove or comment-out the “Deny from all” condition in the default <Directory> tag. Here is an example:

```
<Directory />
Options FollowSymLinks
AllowOverride None
Order deny,allow
# Deny from all
Satisfy all
</Directory>
```

Then save the change, restart Apache, and re-request the resource.

If you not comfortable “opening up” access to your Apache 2.2.x in this manner then you may leave the `Deny from all` as it is, and instead add entries such as this to the end of your httpd.conf file:

# for JSPs that execute in the context of the webapp whose context path is /

```
<Files “*.jsp”>
 Order allow, deny
 Allow from all
</Files>
```

# for servlets that execute in the context of the web-app whose context path is `/`

```
<Location “/servlet”>
 Order allow, deny
 Allow from all
</ Location>
```

# for the web-app whose context path is `/servletexec` (the SE Admin UI)

```
<Location “/servletexec”>
 Order allow, deny
 Allow from all
</ Location>
```

# for a web-app whose context path is `/spaceShuttle`

```
<Location “/spaceShuttle”>
 Order allow, deny
 Allow from all
</ Location>
```

NOTE: If you choose this technique, you’ll need to add a Location directive for each web-app context that you deploy in ServletExec.

Important

The Apache 2.2.x web server is preconfigured to deny servlet & JSP requests. Configure your Apache as described above to work around that.

See Section 4.6 of this guide for more information on the webadapter.properties.

4.6 Configuring webadapter.properties

ServletExec/AS native adapters can be configured using the webadapter.properties file. The file is always put in the config directory of your ServletExec installation by default, but it can be put in alternate locations. See the information in this document regarding your brand of web server for specifics on specifying an alternate location.

The webadapter.properties file is used by the ServletExec/AS native adapter, no matter which brand of web server is used. The top of the webadapter.properties file contains extensive comments about the various properties that can be set in that file. This section will describe some of the properties that can be controlled from this file. Property names are always set up in the form of:

servletexec.<*instance name*>.<*property name*>

The <*instance name*> portion of the property name is used to “group” sets of properties together. Each “grouping” of properties applies to 1 single SE AS Instance.

Properties in this file are used to modify connection pooling between the web server adapter and the ServletExec AS Instance(s), find ServletExec instances that are set up to handle the incoming requests, and forward the request to instances set up to handle specific hostnames and/or specific web applications that are deployed.

4.6.1.1 Connection Pooling Between the Web Server Adapter and ServletExec

The installer automatically enables connection pooling between the web server adapter and ServletExec, using default settings. The purpose of this connection pooling is to improve performance through better resource management.

You can change the default settings by modifying the configuration file as explained below. The two parameters that provide the connection pooling are: `pool max idle` (sets maximum number of idle connections that can be in the pool… default setting is 10) and `pool increment`. (sets the number of idle connections to be added to the pool when the pool is empty and another request is received… default setting is 5). Once the `pool max idle` limit is reached, freed connections will be disconnected.

A good guideline for changing these parameters from their default settings would be to set the `pool max idle` to your average number of connections, and to set the `pool increment` to the average spike number of connections.

Note

This feature is not available for ServletExec/AS on Apache Http Server 1.3.x (UNIX). It is not needed due to the architecture of Apache 1.3.x on Unix

To modify Connection Pooling for ServletExec/AS

- Add the following to the end of the `webadapter.properties` file on two separate lines:

```
servletexec.<instance-name>.pool-increment=<value>
servletexec.<instance-name>.pool-max-idle=<value>
```

4.6.1.2 Hosts

The *hosts* property is used to find all ServletExec AS instances that are configured to process requests for the designated host. The host name is taken from the HOST header field of the HTTP request. The *hosts* property should contain a comma separated list of IP addresses and/or hosts names. The special value `all` can be used to indicate that all hosts should be processed by the specified instance name.

To update the hosts property in `webadapter.properties`

- Use the following form:

  servletexec.<instance-name>.hosts=www.newatlanta.com

4.6.1.3 ServletExec Instances

The *instances* property contains the IP address and port number of 1 (not multiple) ServletExec AS instance. This is used by the adapter for forwarding HTTP requests to that instance. Each instance must contain a unique IP address and port number combination. The default value is 127.0.0.1:8888.

To update the instances property in `webadapter.properties`

- Use the following form:
  servletexec.<instance-name>.instances=127.0.0.1:8888

4.6.1.4 Global Properties

In addition to the per-instance properties there are also a few global properties.

- servletexec.aliasCheckInterval

  The default value is `10` (measured in seconds).
  Setting it to `-1` turns it off, so that there is no request-time polling.
  An example of using this property in webadapter.properties is:
  `servletexec.aliasCheckInterval=20`

  This global property governs how often the adapter will ping/query the SE AS instances (ASIs) that are defined in that file, to ask them if there are any changes to the aliases (webapp context paths, uri mappings for security, filters, servlets, etc...). The adapter stores a local copy of such mappings in what's referred to as an *Alias Cache* (1 alias cache per ASI) and periodically attempts to update that cache (at request time) if warranted.
  If an ASI is not running and the adapter tries to ping/query it, there may be a timeout delay before the socket knows that there is nothing at the other end to respond to the ping/query. This is usually 1 second, but may differ depending on OS, and other factors. If there are several ASI's that are not running or are otherwise unreachable then the delay is increased proportionally (1 second delay for each ASI that's down). This situation can be avoided by enabling the multiInstanceMode property (see next bullet).
  For additional details about the aliasCheckInterval property, please see:
  http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=259

- servletexec.multiInstanceMode

  The default value is `disabled`.
  An example of using this property in webadapter.properties is:
  `servletexec.multiInstanceMode=enabled`

  Enabling multi-instance mode will cause the adapter to:
  - only ping an unresponsive ASI every aliasCheckInterval seconds (rather than continually trying to ping it on every request in an attempt to update its outdated/expired alias cache).
  - clear the in-memory alias cache for any ASI that becomes unresponsive. This provides better support for using SE in a failover/clustered environment. It

allows the request to “filter down” to the other ASI's or the webserver itself rather than giving **Failed To Connect to ServletExec** messages over and over. Just be aware that you may get 404 Not Found responses in this case.

To confirm that your adapter is running in multi-instance mode, look for the presence of the following line in your adapter's startup messages:
This adapter will run with multi-instance optimizations.

4.7 ServletExec/AS Operation

A ServletExec/AS instance [ASI] is a standalone Java application that communicates with the web server (through the ServletExec/AS web server adapter) using network sockets.

The `StartServletExec` and `StopServletExec` scripts are installed into the `se-<instance-name>` subdirectories of the `ServletExecAS` installation directory. Each ASI has its own copy of those 2 scripts.

4.7.1 Starting ServletExec/AS

A ServletExec/AS instance may be started by running the `StartServletExec` script as follows:

```
# ./StartServletExec &
```

For more details please see SE FAQ #115:

http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=115

4.7.1.1 StartServletExec

If you examine this script you can see that it uses standard *java options*, and also several required and optional *program arguments*. You may edit `StartServletExec` as needed. The next section features a complete list of ServletExec/AS command line arguments.

4.7.1.2 How StartServletExec uses the java Command

The information in this section can help you edit the `StartServletExec` script to modify the startup parameters for a ServletExec/AS instance (if needed).

The general form is:

```
java <java options> ServletExec <ServletExec/AS program arguments>
```

Important

The `java` options must include the `-classpath` option specifying the path to the `servlet-api.jar, jsp-api.jar, ServletExec60.jar, ServletExecAdmin.jar, el-api.jar, jasper-el.jar, jstl.jar, appserv-jstl.jar, activation.jar, mail.jar` archives.

Start ServletExec/AS arguments

If an option or argument value contains spaces then the entire value must be placed inside double quotes, as in the following example:

```
-name "Local Instance"
```

See the following table for a complete list of available arguments.

Argument	Explanation
`-help`	Displays all available arguments and explanations.
`-port <port number>`	**Optional**. The default value is 8888. This argument specifies the TCP/IP port on which the ServletExec/AS instance communicates with the web server adapter(s). If you modify this value, you must also modify the web server configuration file(s) as described above, and you must also modify the –port value used in the StopServletExec.bat file.
`-backlog <length>`	**Optional**. The default value is 50. This argument specifies the size of the ServletExec/AS incoming request queue. Requests received when the queue is full are rejected. Increasing the size of the request queue may prevent requests from being rejected under heavy loads, but may increase per-request response times.
`-name <`*`instance-name`*`>`	**Optional**. This is used to specify the name of the ServletExec/AS instance and is used only for display in the Admin UI to identify the instance.
`-home <path to the ServletExec/AS instance home directory>`	**Optional**. The default is the directory from which the `java` command is executed. This specifies the directory in which the ServletExec/AS instance will look for the `ServletExecData`, `ServletLogs`, and `Servlets` subdirectories.
`-log <path to the ServletExec/AS instance log directory>`	**Optional**. This can be used to specify an alternate directory for writing the ServletExec/AS log files. By default, log files are written to the `ServletLogs` subdirectory of the ServletExec/AS home directory as specified by the `–home` argument.
`-root <path to the web server's document root directory>`	**Required**. Specifies the **case-sensitive** path to the web server's document root

Argument	Explanation

	directory. If using SE virtual servers [VS], the doc root of each SE VS must be configured using a separate `–root` argument using the following format:
	`-root <virtual server>=<folder>`
	for example:
	`-root www.abc.com=C:\Apache\htdocs\abcdocs`

`-root <path to the virtual server’s document root directory>`	*See previous explanation.*

`-addl <url=directory>`	**Optional**. This can be used to specify additional **case-sensitive** document directories (other than the root) and is specified in two forms:
	`-addl <url>=<physical directory>`
	`-addl <virtual server/url>=<physical directory>`
	For example:
	`-addl /test=/usr/local/test`
	`-addl www.abc.com/support=/usr/support`

`-addl <virtual server/url=directory>`	*See previous explanation.*
`-mimetypes <path to the web server’s mime.types file>`	**Optional**. This specifies the path to the web server’s `mime.types` file. If this argument isn’t provided, the `ServletContext.getMimeType()` method always returns `null`.

Argument	Explanation
-allow <ip1,ip2,…,ipn>	**Optional**. This specifies the IP address(es) of the web server(s) that are allowed to communicate with the ServletExec/AS instance. Either IPv4 or IPv6 Addresses may be used. IPv4 addresses must be of the form x.x.x.x (where x is in the range 0-255) IPv6 addresses must be of the form [y:y:y:y:y:y:y:y] (where y is a 4-digit hexadecimal number). IP addresses can include the "*" character to indicate a subrange (for example: `168.121.97.*`, [*:*:*:*:*:*:*:*]). If the `–allow` argument is not specified, the ServletExec/AS instance only accepts requests from web servers running on the local machine.

Table 7. Start ServletExec/AS Command Line Arguments

Note: The `–root` and `–addl` arguments are used by the webapp named "default-app" only, and have no effect on other deployed webapps. For more details please see SE FAQ #278: http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=278

4.7.2 Stopping ServletExec/AS

You must stop a ServletExec/AS instance using the `StopServletExec` script. If a ServletExec/AS instance is not shut down properly, servlet `destroy()` methods are not invoked, sessions are not saved, Web Application Lifecycle Events are not generated, and buffered log messages are lost.

4.7.2.1 StopServletExec

`StopServletExec` invokes the `StopServletExec` Java program to stop the ServletExec/AS instance. That program opens a TCP port and sends a message to the instance, telling it to shutdown.

A ServletExec/AS instance may be stopped by running the `StopServletExec` script as follows:

```
# ./StopServletExec
```

4.7.2.2 How StopServletExec uses the java Command

The information in this section can help you edit the `StopServletExec` script to modify the shutdown parameters for a ServletExec/AS instance (if needed).

The general form is:

```
    java <java options> StopServletExec <program arguments>
```

Most likely, the only edits you'd ever need to make to this file is to change the value of the `–port` argument to match the `–port` value specified in `StartServletExec`, or to change the version of Java it uses.

Stop ServletExec/AS arguments

If an option/argument value contains spaces then the entire value must be placed inside double quotes, as in the following example:

```
-name "Local Instance"
```

See the following table for a complete list of available arguments.

Argument	Explanation
`-help`	Displays all available arguments and explanations.
`-port`	Specifies the port number used by the ServletExec/AS instance

Table 8. Stop ServletExec/AS Command Line Arguments

4.7.3 Remote Operation

By default, a ServletExec/AS web server adapter is installed on the same machine as the ServletExec/AS instance, and the ServletExec/AS instance (by default) only accepts requests from web server adapters running on the local machine. It is possible to install and configure web server adapters to run on a different machine (a "remote" machine) than the ServletExec/AS instance.

To run a web server adapter on a different machine

7. Edit the `StartServletExec` script for the ServletExec/AS instance to add the following argument at the end of the `java` command line:

    ```
    -allow <ip address>
    ```

    where `<ip address>` is the IP address of the remote machine on which the web server adapter will be installed

    See table 4 in section 4.7.1.2 for a complete discussion of the –allow argument.

8. Run the ServletExec/AS installer on the remote machine on which the web server adapter is to be installed. Choose **Install a web server adapter** when prompted.

    See Section 4.4 for more information.

9. Enter the **name**, **IP address**, and **port number** of the ServletExec/AS instance when prompted.

Note

You can install multiple web server adapters on multiple machines or on a single machine to forward requests to a single (or multiple) ServletExec/AS instance(s).

4.7.4 Multiple ServletExec/AS Instances (Multiple Java VMs)

It's possible to install and run multiple ServletExec/AS instances on a single server machine (or on unique machines). Each ServletExec/AS instance will run independently of the others, each within its own Java Virtual Machine (VM).

To install multiple ServletExec/AS instances

Run the ServletExec/AS installer again to install another instance. Remember to give each instance a unique name and port number during the installation process.

To configure a web server adapter to forward HTTP requests to more than one instance

It's possible (but not required) to configure a single web server adapter to forward HTTP requests to more than one ServletExec/AS instance. The adapter will use both the hostname of the request and the context path of the request to decide which (if any) SE AS instance should be handed the request. The hostname portion of the request is matched against the hosts defined in webadapter.properties. The context path portion of the request is matched against the context paths of all webapps deployed on all AS instances whose hostname matched the request.

One option to configure this is to use unique webapp context paths (unique across all SE instances). Another option is to modify the webadapter.properties file to define a unique set of hostnames or IPs [via the .hosts value] for each instance. Yet another option is to use some combination of those 2 options. For details please read
SE FAQ #259:

`http://www.newatlanta.com/c/support/servletexec/self_help/faq/detail?faqId=259`

4.8 Uninstalling ServletExec/AS

To uninstall ServletExec/AS or a web server adapter

1. Re-run the ServletExec/AS installation script.
2. Click the appropriate option from the following choices: uninstall a ServletExec/AS instance, uninstall a web server adapter, or completely uninstall ServletExec/AS.
3. If you are uninstalling a web server adapter or completely uninstalling ServletExec/AS, you must manually edit the web server configuration files (`obj.conf & magnus.conf` for SunONE/SJSWS, or `httpd.conf` for Apache) to remove any changes you may have made for ServletExec/AS.

See Section 4.5 for more information.