

Doc Library

Main Topic

Administrator/Demo
23/02/2006 05:02

Subject: Configuring IIS as a Front End server to Domino
Category: Walkthroughs

Originator
Administrator/Demo

Reviewers

Review Options

Type of review: One reviewer at a time
Time Limit No time limit for each review
Options:
Notify originator after: final reviewer

Configuring IIS

First up, you need to configure IIS to use the plugin (Combined from the Domino Admin help and a technote)

1. Create the following directory structure on the IIS machine (you can use any drive)
 - C:\WebSphere\AppServer\bin
 - C:\WebSphere\AppServer\config
 - C:\WebSphere\AppServer\etc
 - C:\WebSphere\AppServer\logs
2. Copy the following files from the Domino server to the IIS server:

Copy data\domino\plugins\plugin-cfg.xml to c:\Websphere\Appserver\config

Copy data\domino\plugins\<WAS version>\w32\iisWASPlugin_http.dll to c:\websphere\AppServer\bin

3. Start IIS and create a new virtual directory for the web site instance that you intent to work with Websphere Application Server. These instructions assume that you are using the Default Web Site
4. Click Programs > Administrative Tools > Internet Information Services (IIS) Manager on a Windows Server 2003 Standard Edition system, for example.
5. Expand the tree on the left until you see Default Web Site. Right-click Default Web Site > New > Virtual Directory to create the directory with a default installation.
6. Type sePlugins in the Alias field in the Virtual Directory Alias panel of the Virtual Directory Creation Wizard, then click Next.
7. Browse to the location of the plugins directory under the Domino Web server, or the WebSphere\bin folder you created.
8. Select the appropriate permission check boxes in the Virtual Directory Access Permissions panel of the wizard. Select the Read check box and the Execute (such as ISAPI applications or CGI) check box, for example.
9. Click Next to add the sePlugins virtual directory to your default Web site.
10. Click Finish when the success message displays.
11. Create a text file named plugin-cfg.loc in the same directory as the plugins. The first line of the plugin-cfg.loc file identifies the location of the plugin-cfg.xml file. for example:
C:\WebSphere\config\plugin-cfg.xml.
Note that the last line of the .loc file should be a blank line (carriage return).
12. Right-click the Web Sites folder in the left pane navigation tree of the IIS Manager panel, and click Properties.
13. Add the Internet Services Application Programming Interface (ISAPI) filter into the IIS configuration.
14. In the Web Site Properties panel, perform the following steps:
15. Click the ISAPI Filters tab.
16. Click Add to open the Add/Edit Filter Properties dialog window.
17. Type iisWASPlugin in the Filter name field.
18. Click Browse and point at the iisWASPlugin_http.dll in the plugins directory for the value of the Executable field.

19. Click OK to close the Add/Edit Filter Properties dialog window.
20. Click OK to close the Web Site Properties window.
21. Configure the Web server to run WebSphere Application Server extensions:
22. Expand the left pane navigation tree until you see the Web Service Extensions folder in the IIS Manager panel.
23. Click Web Service Extensions to display information about what Web service extensions are allowed.
24. Click All Unknown ISAPI Extensions on the right side of the panel.
25. Click Allow in the middle pane. The status field for the All Unknown ISAPI Extensions changes to Allowed.
26. NOTE: If you are concerned with allowing all unknown ISAPI extensions because you have other ISAPI extensions on this server, follow the steps in Technote 1168244 (or see below) to add a new Web service extension for Domino.

Authorising the ISAPI extension

To add the ISAPI extension, so it can be specifically allowed, do the following:

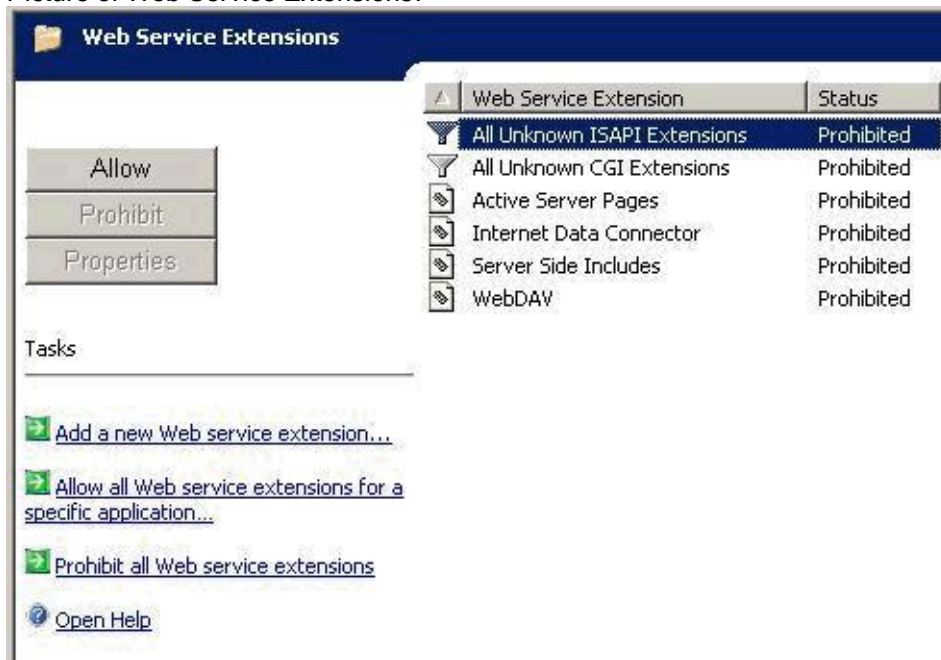
For this configuration to work correctly with Microsoft IIS 6.0, you must perform two additional setup steps.

First, you need to set the IIS server to run in Isolation mode. To do so, go to the properties of the "Web Sites" folder -> Service, and check "Run WWW service in IIS 5.0 isolation mode."

Second, add a new Web service extension to process the NSF file request correctly. To do so, use the following steps:

1. On the Microsoft Windows 2003 server, go to the IIS Manager, and click Add a new Web Service Extension.

Picture of Web Service Extensions:



2. In the New Web Service Extension dialog box, enter "Domino" as the name for the new Web Service Extension.
3. For Required files, add the required iisWASPlugin_http.dll file, and click to set the extension status to Allowed.

Picture of New Web Service Extension dialog:



4. Restart the IIS service.

You should now be able to request NSF files from the Microsoft IIS server.

Configuring the XML file

Now, you need to configure the xml configuration file (From Domino Admin help)

1. Open plugin-cfg.xml in Notepad.
2. Modify the <Transport> element to target the appropriate Domino server. To do this, change the Hostname and Port parameters to the proper values required for the plug-in to reach your back-end server's HTTP task. For example:

```

<!-- Server groups provide a mechanism of grouping servers together. -->
  <ServerGroup Name="default_group">
    <Server Name="default_server">
      <!-- The transport defines the hostname and port value that the web server
      plug-in will use to communicate with the application server. -->
      <Transport Hostname="mydomino.server.com" Port="81" Protocol="http"/>
    </Server>
  </ServerGroup>

```

3. Add these directives to the top of the <UriGroup> section. These directives specify common URL patterns needed for accessing Domino Web applications.

```

  <UriGroup Name="default_host_URIs">
    <Uri Name="/*.nsf"/>
      <Uri Name="/icons/*"/>
      <Uri Name="/domjava/*"/>
      <Uri Name="/execcgi/*"/>
  </UriGroup>

```

```
<Uri Name="/cgi-bin/*"/>
<Uri Name="/servlet/*"/>
<Uri Name="/download/*"/>
<Uri Name="/mail/*" />
```

Note: there are some other things you may want to do!

Certainly, I want URL's with .NSF in the middle of them to be served by domino, so change the line `<Uri Name = "/* .nsf"/>` to : `<Uri Name = "/* .nsf/*"/>`

Next up, for DWA and java applets to load, you need to enable the HTML direcectory too. Be careful with this one though, as you may well have a HTML directory in IIS too. Add the line: `<Uri Name = "/html/*"/>`

Also, for DWA/Sametime integration to work, you need to add this line:

```
<Uri Name="/sametime/stlinks/*" />
```

For Quickplace to work, add the following lines:

```
<Uri Name="/quickplace/*" />
<Uri Name="/qphtml/*" />
```

Configuring Domino

Add the following line the Domino servers' notes.ini

```
HTTPEnableConnectorHeaders=1
```

You should also change the following two settings in the servers' server document. Both are under "Internet Protocols\Domino Web engine"

Generating references to the server \ Does this server use IIS: Change to Yes

Generating references to the server \ Protocol, Port and hostname: as appropriate

Apparently (according to some Domino 5 release notes), these settings ensure that Domino appends ".nsf" to autogenerated URLs, that otherwise wouldn't have them (mainly DBID's). So it should be safe to turn this on, even if the server isn't using IIS. The hostname/port etc should be fairly obvious.

That's it!

Authenticating

Next you need to look at authentication. Domino doesn't support Digest authentication. In certain versions of the SUN JVM, there is also a JRE bug with NTLM Authentication (which also affects reverse proxys too). Although the site will authenticate - any java

applets (such as view outlines or toolbars) won't without an extra username/password box -

So, we'll start with Basic authentication and then try working up. You should be aware that basic authentication doesn't encrypt the username&password, so you should SSL enable the site too.

For basic authentication (and perhaps others), you will probably want to remove the need to enter domain\username in the user login form and just use Username / Password. To do this, you can set a default domain.

1. Right click on the web site
2. Select Directory Security
3. Select "Basic authentication - deselect any others (except anonymous, if you wish to allow this)
4. Enter the AD Domain name in the "Domain" dialog box
5. Users can now log in using just their AD username and password

Once you've got this working, try using NTLM authentication. The advantage of this is that your IE users won't need to enter any passwords at all - they will be logged in using their existing desktop login ID.

Enabling SSL

Next, you'll need to enable SSL on the IIS server. Here's the clever part, you can enable SSL on the IIS server and *not* on the domino server. IIS will act as a SSL accelerator, taking the load away from Domino (and the need to buy another certificate).

Setting up SSL on IIS is fairly simple, but if you want to setup a self certified certificate for testing you do need to install certificate services first. (This is from memory, so may not be exactly correct)

Go to Control Panel add/remove programs and select the windows components.

Add in certification services and let that install.

Open up the Certificate services administrator.

Right click on the server, click Actions\Create certificate (or similar)

Select an enterprise root certificate (top option)

Fill out the details, until you're finished.

Go back into IIS admin and right click the default site

Click on the Directory Security tab

Click on Certificates

Follow the wizard to select a certificate from your local Certificate services, and the self signed SSL certificate will install itself.

User accounts

Finally, you need to configure the user accounts, so that Domino recognises them. IIS passes back the username you have logged into IIS with, telling Domino that you have already been authenticated. So, you will need to add your AD account name to the "Username" field in Domino. For instance:

Warren.Elsmore

You shouldn't need to use the full AD name (DOMAIN\elsmorew), but in some circumstances you may do.

et voila! You should now be able to use a browser to connect to IIS over SSL. IIS will authenticate you against Active Directory, then pass you back on an unencrypted HTTP connection to Domino. Domino web access, authenticated via AD. Easy, eh?
<sarcasm mode off>

Update

With the 6.0 release of the plugins, you may get an error message on startup. If so, follow this abridged technote:

Problem - The following error was displayed on http_plugin.log. "ERROR: ws_common: getIISErrorLocation: Failed to get keysize of IIS Error Location, rc=2"

Cause - Internet Information Service (IIS) error location has not been configured correctly.

Solution - Setup of IIS Error Location is addressed in APAR PQ85348.

During IIS plug-in initialization, the plug-in queries the Windows® registry for IIS Error Location. If this key is not set, an ERROR message is logged in the http_plugin.log. The registry key is needed if you want the IIS plug-in to send designated HTTP error page back to client.

Important: Exercise caution when using the regedit tool. Using regedit incorrectly can cause serious problems that might require reinstalling the Windows operating system. IBM® cannot guarantee that problems resulting from the incorrect use of the Registry Editor can be resolved. Use regedit at your own risk.

Follow these steps to define the error page location in the registry:

1. Start regedit.exe, the registry editor.
2. Right-click the following key:
"HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere Application Server\version"
where version is your WebSphere® Application Server version (such as 5.0.0.0).
3. Follow the popup-menu: New > String value, and type IIS Error Location in the name

field, then press Enter.

4. Right-click the IIS Error Location, then select Modify. After filling in the directory in the value data part, click OK. The directory you entered must contain the error page files that you want the plug-in to send to client browsers. The error page file name must have following format:

xxx.htm, where xxx is the HTTP code.

Currently only 500, 503, and 413 HTTP error codes are used by the WebSphere Application Server plug-in.

Example, assuming above IIS Error Location has this value:

C:\WINNT\Help\iisHELP\Common, and plug-in gets the internal server error 500. The plug-in tries to open following file:

C:\WINNT\Help\iisHELP\Common\500.htm

If that directory and the error page file do exist, the plug-in returns HTTP error code 500 with the content of the above error page file to the client browser before it returns the error code to IIS.