IIS

Half of Shibboleth runs within the web server. For IIS, this half is implemented in an IIS "native module" packaged in a file named **iis7_shib.dll**. Because IIS provides very minimal support for the configuration of extensions on its own, all of the runtime configuration is handled by the standard SP configuration file (*shibboleth2.xml*) with the exception of the basic installation of the module into IIS.

New Version in V3 of the SP

As of V3, a new IIS plugin is available which provides a richer and more secure integration with IIS version 7 and later. New Installations should use this version unless they are constrained to use IIS version 6 or earlier.

The old ISAPI filter/extension DLL (**isapi_shib.dll**) is still shipped and existing installations will be updated to the latest version of the older DLL.

The configuration for the new module is backward-compatible with the old extension (although it uses Server variables rather than HTTP headers for AttributeAccess by default).

The new module takes full advantage of the breadth of the IIS7 APIs. Two notable advantages are:

- By default, it passes values to application using Server Variables rather than HTTP Headers.
- It can be easily configured to support native Roles-based Authorization where the roles are derived directly from attribute passed to the SP. An example use of roles based authorization is URL Authorizaion.

Additionally the new plugin allows form-preservation across a SSO login by setting the `postData` attribute in the `<Sessions>` element.

Upgrading an Existing Installation

Details on upgrading an installation of the old IIS plugin to the new module are given here.

For new installations, refer to the rest of this page.

Installing the Module

If the Installer detects IIS7 or later, then the **iis7_shib.dll** module is used. This automatically configures the IIS7 module, adding support for both 32-bit and 64-bit app pools.

No need for 32-bit shibd on 64-bit OS

The 32-bit web server modules can function with a 64-bit shibd service, so there is no need to install a special version of shibd to handle 32-bit cases as with older SP versions.

Typically, environment variables are used to set the appropriate path information to enable the library to locate the configuration file and initialize itself when IIS or its child processes are started (though this is all handled by the installer when required).

Startup Failures

If you experience startup problems, you should do the following:

- Verify the configuration is generally valid by running **%SHIBSP_PREFIX%/sbin/shibd.exe -check** from the command line.
- Make sure the system path contains the location of the SP's library DLLs (and make sure you reboot after installation before assuming that's happened).
- Assuming that reports no serious errors, verify that all of the machine accounts used by IIS have read permission to the SP installation tree.

Configuring

IIS has a design difference that separates it from Apache: it doesn't support true virtual hosting because it provides no mechanism to securely establish the "canonical" properties of a web site such as its virtual hostname or port. Instead it divides the web server into "site instances" that can have properties like names and ports attached to them (with no reliable/secure way for applications to obtain them or to override the physical settings with logical ones).

The SP's internals don't understand the concept of a "site", so to correct for this, an IIS-specific piece of XML configuration must be included within the `<IISAPI>` element that performs a mapping between a site instance number/ID and the associated "canonical" virtual host information. Note that the hostname can be inferred by the plugin from the client request, but it is usual not to, and can create security vulnerabilities.

It is critical when performing initial setup, and when adding new Shibboleth-enabled web sites to an IIS server, to create those mappings. Failure to do so will result in the system ignoring requests to unmapped sites, or handling requests improperly. Note that this is also a feature: any site instances you provide no mapping for will be ignored by the software.

Any time you manipulate the `<ISAPI>` configuration section, you'll need to restart IIS completely.

Once the necessary site instance mappings are created, the rest of the per-request configuration is handled exclusively by the `<RequestMapper>` compo nent, which essentially takes the place of what would be done in Apache with its command format.

Displaying Server Variables

The following file, saved with extension `.aspx` will display specific information an application may be able to see. As far as anybody has been able to determine, the usual mechanisms to loop over and dump all available variables or headers don't ever include the ones set by the SP module, but accessing them directly does work. The names below are just examples.

```
<% @ Page Language="C#" %>
<%
Response.Write("<h3>Server Variables</h3>");
Response.Write("eduPersonPrincipalName = " + Request["eduPersonPrincipalName"] + "<br>");
Response.Write("eduPersonScopedAffiliation = " + Request["eduPersonScopedAffiliation"] + "<br>");
Response.Write("displayName = " + Request["displayName"] + "<br>");
%>
```


If you are using HTTP headers (which is **not recommended)** be aware that allowing end users to see the HTTP headers en masse will compromise the header spoofing checks the SP implements because it exposes the random value used to detect manipulation (it's like a secret password).

Authorization

The IIS plugin has limited support for Roles Based Authorization. This is performed by adding `<Roles>` elements to the `<ISAPI>` element. However it is currently more usual to either perform the authorization within your application, or rely on the `<RequestMapper>` and the XML-based Access Control plugin (or an alternative plugin to the SP).

Content Settings

The SP supports an extensible set of content settings, properties that control how it interacts with requests and enforces various requirements. On IIS, these settings can be controlled only by attaching properties using the `<RequestMapper>` mechanism in the SP configuration.

For more information about using the `<RequestMapper>` feature, refer to the How To topic.