NativeSPWindowsInstall

Windows Installation

The native Shibboleth SP is available for Windows with modules for all the supported web servers. There is an installer available for supported Windows versions:

- Windows 2008 Server
- Windows 2012 Server
- Windows 2016 Server

The following older versions are no longer supported by Microsoft and therefore are not officially supported by the project, but may still be usable. Be aware that they are no longer tested routinely and future Shibboleth versions may stop working on them at any time (and in fact it's all but certain to happen on the next upgrade due to Microsoft compiler changes):

- Windows XP SP2 and Later
- Windows 2003 Server SP1 and Later

Earlier Windows and Service Pack versions are **not** usable as of V2.4. Since earlier versions of the SP are no longer safe to use due to security issues, we do not recommend using them to obtain compatibility with older Windows versions.

> ⚠ The installer does not work fully in conjunction with the IIS "Shared Configuration" option available on newer releases of IIS. Disable it prior to installation. After re-enabling it, you will likely have to manually apply some of the configuration changes; refer to the appropriate topic below.

As of V2.5, the Windows installation has been revamped significantly to support enhanced operation and ease of patching and upgrades:

- You may now install the SP to any location you wish, rather than being forced to use **C:\opt\shibboleth-sp** (though this is still recommended for consistency).
- Most shared components are now installed to **Program Files\Shibboleth** and **ProgramData\Shibboleth**. Most files deployers are familiar with remain in the same relative locations, so these changes should be mostly invisible to you.
- The 64-bit install now contains both 32-bit and 64-bit binaries and you can switch architectures after installation (but not run both at the same time). When using a 64-bit OS, you **MUST** use the 64-bit installer now; the 32-bit installer will refuse to run to ensure this.

The Windows installer now actually contains a fourth version field that indicates the patch level within a particular SP release. Initially 0, it will be incremented if patches to software included with but not part of the SP need to be updated (e.g., OpenSSL). Subsequent patch level installers will upgrade older versions.

Note that as with older releases, if you want the installer to attempt IIS configuration for you when using IIS 7.x, the IIS 6 management compatibility feature must be installed.

Install for IIS 5
Install for IIS 6
Install for IIS 7 (this also applies to IIS 8 for now)
Install for Apache

Upgrades From 2.x Releases

It's also possible (though not simple) to build the SP and its dependencies from source using Visual Studio, but this should only be attempted by experienced Windows developers.

Build from Source

Shibboleth Service

Once installation is complete, you'll need to run the Shibboleth daemon, `shibd`, at all times. `shibd` is a console application that is usually installed as a Windows service.

- To run the process in console mode for testing or to diagnose major problems, supply a `-console` parameter when running it.
- If `shibd` won't start, use the `-check` option from the command line to echo most logging information to the console for debugging.

Other parameters can be used to install (or remove) `shibd` from the service database and subsequent control is generally via the Service Control Manager applet.

> ✓ **Monitoring the Service**
>
> Newer versions of Windows support automatic restart of failed services. We suggest using this feature to restart shibd when it fails. Although stability is good, maximum reliability will be achieved by monitoring the process.

Failure to Install

The most common reason for the installation failing is that the Shibboleth service (above) does not start correctly. In order to debug (versions 2.5.1 onwards) this you can instruct the installer to **not** try to start the service by specifying that the ALWAYS_START_SERVICE property contain the value FALSE. Do this from the command line:

Installing with no service start

```
c:\> msiexec /i Installer.msi ALWAYS_START_SERVICE=FALSE
```

You can then use the `-check` option described above to debug why the service will not start. Once this is completed you can start the service manually.

Starting the Shibboleth Service from the command line

```
c:\> sc start shibd_default
```

Initial Testing

You can test to ensure that the SP is running properly and the surrounding environment is correct by accessing https://localhost/Shibboleth.sso/Status **from the actual web server machine**. You **MUST** use "localhost" as the hostname or it WILL NOT WORK by default. If this test is successful, then the software is ready for further configuration.

> You can also access the Status handler from other clients or using a non-localhost name, but **only** if you change the `acl` parameter in the configuration to permit your client address or remove it entirely to open up access to anybody. The ACL is present by default because the Status handler can return some arguably sensitive information about your configuration.

Now you can progress to the Getting Started material, or if you're in the very early stages of evaluation, try a more controlled scenario by using the TestShib IdP. (Note that before using the TestShib IdP, you'll need to complete the first step from Getting Started, setting the entityID attribute in the ApplicationDefaults element of shibboleth2.xml.)

Once you've actually configured the SP with its own settings and metadata from at least one IdP, in order to check that the SP is "working":

1. Protect a directory by requiring a Shibboleth session. Usually, this is already done by default for the location "/secure".
2. Next, you typically place a script inside the protected directory that dumps the web server environment. With PHP for example you could in the easiest case just place a script there with the following:

   ```
   <?php print_r($_SERVER) ?>
   ```

   A more advanced version of such a script can be found here.
3. Make sure that the Shibboleth-supplied variables are present. If there is a non-empty variable called **Shib-Application-ID**, you successfully authenticated and have a valid session. However, you also should check if there are other non-empty Shibboleth variables defined in the **attribute-map.xml** file. If there are no variables like **mail** or **givenName** or **surname**, the IdP either releaseed no attributes, or the attribute request failed (the latter usually only applies when using an older IdP). In this case, have a look at the `shibd.log` file.