MacNotes

ServiceProvider Notes

Successful C++ SP Builds

- OS X 10.3 using gcc3
- OS X 10.4 using gcc3
- OS X 10.4 using gcc4
- NOTE: Upgrade type installations of OS X 10.4 will run into strange compilation errors: `/usr/bin/libtool: for architecture: cputype (16777234) cpusubtype (0) file: -lSystem is not an object file (not allowed in a library)`. This may be remedied by installing a fresh version of XCode Tools 2.1.

Build Environment Setup

- Set `DYLD_LIBRARY_PATH` to the eventual lib path into which you'll be installing the packages you're building (e.g. `/opt/shibboleth-sp/lib`)
- Set `MACOSX_DEPLOYMENT_TARGET` to your OS X version (e.g. 10.3 or 10.4)
- Set `XERCESCROOT` to the root of the unpacked Xerces-C tarball

`configure` Commands

- OpenSSL: (not needed, just use what comes with OS X)
- libcurl: (not needed on 10.4, or use `./configure --disable-static --without-ca-bundle --enable-thread --prefix=/opt/shibboleth-sp`)
- log4cpp: `./configure --disable-static --disable-doxygen --prefix=/opt/shibboleth-sp`
    - For log4cpp 0.3.5rc3 and most likely others, the file `../include/log4cpp/Manipulator.hh` contains an extra semi-colon (`;`) on line 29 that needs to be removed before building log4cpp.
- Xerces-C: `./runConfigure -p macosx -c gcc -x g++ -r pthread -n native -b 32 -P /opt/shibboleth-sp`
- XML-Security-C: `./configure --without-xalan --prefix=/opt/shibboleth-sp`
- OpenSAML: `./configure --with-curl=/opt/shibboleth-sp --with-log4cpp=/opt/shibboleth-sp --prefix=/opt/shibboleth-sp -C`
- Shibboleth: `./configure --with-saml=/opt/shibboleth-sp --with-log4cpp=/opt/shibboleth-sp --enable-apache-13 --with-apxs=/usr/sbin/apxs --prefix=/opt/shibboleth-sp -C`

Build Notes

- On all versions of OS X, the ONC RPC library used by Shibboleth for www/shibd connectivity is not thread-safe for clients or servers. Since shibd is multi-threaded, the Shibboleth-supplied version of `liboncrpc.dylib` must be used instead of the Apple version. This is automatically handled by the `configure` script, but does result in various warnings about duplicate symbols while compiling and linking. This is normal and does not appear to prevent the proper symbols from being found at run time.

- It's possible to build some of the libraries in either flat- or two-namespace mode. This means that all exported symbols can be global (normal on Unix) or scoped to the shared library (normal on Windows). Either approach appears to work, but scoping symbols is usually preferable, and can be enabled by setting the MACOSX_DEPLOYMENT_TARGET environment variable to 10.4 (or whatever the OS X version is). Anything value newer than 10.1 will default to two-namespace mode during the build.

- For Xerces to fetch remote URLs (something the `siterefresh` tool currently depends on, you must enable the "native" Xerces NetAccessor on this platform. This is done with the option `-n native` when running `runConfigure`

Runtime Notes

- Set `DYLD_LIBRARY_PATH` to the lib path containing the dependencies and Shibboleth libraries (e.g. `/opt/shibboleth-sp/lib`). This is required both in the script that starts the `shibd` daemon and when starting Apache (so you might put it in `/usr/sbin/apachectl`).

- On at least 10.4, it appears to be necessary to "preload" some symbols from some of the libraries in order for C++ RTTI to work when Apache loads mod_shib. This may be an artifact of changes in gcc4. To do this, you must set DYLD_INSERT_LIBRARIES in the Apache processes to `/opt/shibboleth-sp/lib/libshib-target.dylib:/opt/shibboleth-sp/lib/libshib.dylib` (substituting the actual path of these libraries, if not in the location shown). The simplest thing to do is place a command to set and export this variable in `/usr/sbin/apachectl` such as:

```
export DYLD_INSERT_LIBRARIES=/opt/shibboleth-sp/lib/libshib-target.dylib:/opt/shibboleth-sp/lib/libshib.dylib
```