Axis c++ windows installation guide

Description

This document describes the installation process of Axis C++ on windows OS , using Apache web server, to the point where you have a running Calculator sample (provided with Axis) web service.

What does Axis do?

Web service is a way to expose services to developers over the web. For example, a Calculator object reside on a web server might expose an Add service , so that a client software can call this function and get a result over the web.

The protocol used for this communication is SOAP, which is a XML based protocol.

What Axis do is parse this SOAP message, deserialize and invoke the C++ class which perform the actual calculation, and than serialize the result back to SOAP protocol and return it to the callee.

Installation

The installation can be divided to 2 parts:

1. Installing Axis C++ server.
2. Creating the Web service.

1. Installing Axis C++ server

How does it work?

Axis is implemented as a module (mod_axis.dll for apache 1.3 or mod_axis2.dll for apache 2.0) which is loaded by the web server. A module is a piece of software which declares its interest in the messages sent to the web server.

In order to be loaded as a module, this dll should be placed in the server's *modules* folder, and be registered in the server's httpd.conf file.
The module itself requires several dlls to properly work. Some of them come with Axis and some with the apache server. In order for them to be found, we will add the path to their containing folders to the *Path* environment variable.
Axis use a wsdd file (server.wsdd) to list the services exposed to web clients and to map these services to the actual C++ classes, like the Calculator web service (Calculator.dll).
In addition, Axis use a configuration file (axiscpp.conf) which contains paths to the

above mentioned dlls, the wsdd file and path to a log file.

Step by step installation

Before we start

The following instructions are based on the Axis C++ Installation Guide and the Axis C++ Windows Installation Guide.

Although this document contains all the required information, you might want to print these guides as a reference.
Dependency walker

> The best tool for the installation job is the Dependency Walker (will be referred as Depends.exe in this document). Please get it.

In case you're not familiar with Depends.exe – this tool loads dlls and report any problem discovered in the loading process. All you have to do is run Depends.exe → File → open → and browse to the tested dll.

> **Important Note** : loading a dll using Depends.exe, which results with only warnings is considered a successful load. A very common warning is "**Warning: At least one module has an unresolved import due to a missing export function in a delay-load dependent module.".** You can safely ignore it, as long as it's not an error.

Environment variables

> When a dll is loaded and require other dlls, the system looks for them in the paths appearing in the *Path* environment variable. You can edit it using the following instructions (for setting the *Path* variable) :

In Windows NT/2000/XP, follow the following steps:
1. Open Control Panel
2. Click the System icon.
3. Go to the Advanced pane.
4. Click the Environment Variables button.
5. in the "System variables" look for Path, and click it.
6. click on the Edit button
7. add a semicolon (;) and the the desired path

Important Note : Usually editing the environment variables takes affect only after you log out or restart the computer.

Each step will contain a *Success Test* section which describes the test you can perform in order to verify the success of this step. Please perform these tests as trying to solve problems later in the process is a real pain.

Installing the apache server

Axis can work with a *simple axis server* instead of a web server. This document does not reffer to this option.

You can download apache server from here.
Before the installation stop your IIS server (if you have one).

Install apache web server and locate the installation folder(For the default installation (apache 1.3.X) the path is "C:\Program Files\Apache Group\Apache" and for apache 2.X the path is "C:\Program Files\Apache Group\Apache2"). Lets say this folder is [Apache_Folder].

This Folder cantains a *lib* folder, which contains dlls needed by the Axis module (libhttpd.dll and libapr.dll). Please add the path to this lib folder to the Path environment variable, so that they can be found.

Success test : type http://localhost/ in your browser. You should get apach test page stating something like "If you can see this, it means that the installation of the Apache web server software on this system was successful". This statement might be in the language of your locale.

Installing the Xerces XML parser

Axis C++ needs an XML parser to parse SOAP messages and WSDD files . It has a parser abstraction layer that helps users to select/switch between parsers. However only one parser library can be used at a time (this parser library - AxisXMLParser.dll, will be later configured in the Axiscpp.conf file, in the XMLParser key). This parser library uses the Xerces XML parser.(xerces-c_2_2_0.dll)

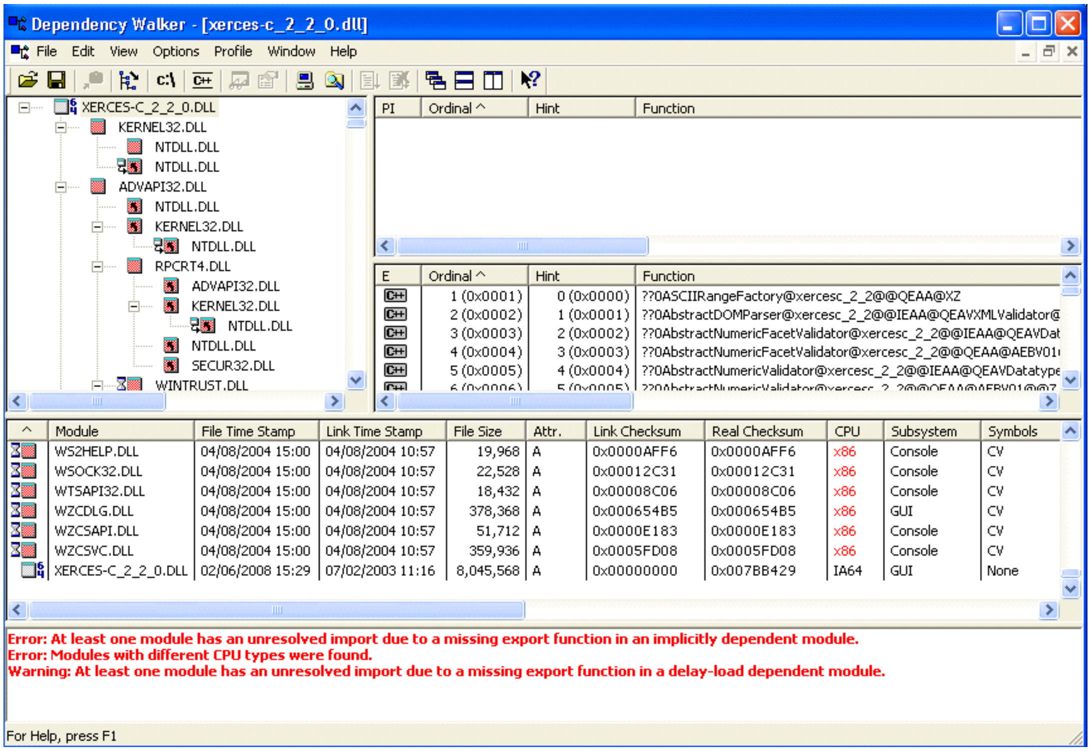
Were do I get it?

Please note that the link appear in the installation guides (http://xml.apache.org/xerces-c/download.cgi) brings you to a page where all versions of the dll are not c_2_**2**_0 but c_2_**8**_0. So instead using this, just google for "xerces-c_2_2_0.dll download", and hopefully you'll get the right one.

Place the dll in a temporary folder you choose. we will refer to this folder as [Xerces_Folder].

Success test : use Depends.exe to load the dll. In case it loads without errors your'e good to go.

Troubleshooting: One potential problem can be a mismatch between the used xerces-c_2_2_0.dll and your system:


Dependency Walker - [xerces-c_2_2_0.dll]
File Edit View Options Profile Window Help

```
XERCES-C_2_2_0.DLL
    KERNEL32.DLL
        NTDLL.DLL
        NTDLL.DLL
    ADVAPI32.DLL
        NTDLL.DLL
        KERNEL32.DLL
            NTDLL.DLL
        RPCRT4.DLL
            ADVAPI32.DLL
            KERNEL32.DLL
                NTDLL.DLL
            NTDLL.DLL
            SECUR32.DLL
        WINTRUST.DLL
```

PI	Ordinal ^	Hint	Function

E	Ordinal ^	Hint	Function
C++	1 (0x0001)	0 (0x0000)	??0ASCIIRangeFactory@xercesc_2_2@@QEAA@XZ
C++	2 (0x0002)	1 (0x0001)	??0AbstractDOMParser@xercesc_2_2@@IEAA@QEAVXMLValidator@
C++	3 (0x0003)	2 (0x0002)	??0AbstractNumericFacetValidator@xercesc_2_2@@IEAA@QEAVDat
C++	4 (0x0004)	3 (0x0003)	??0AbstractNumericFacetValidator@xercesc_2_2@@QEAA@AEBV01
C++	5 (0x0005)	4 (0x0004)	??0AbstractNumericValidator@xercesc_2_2@@IEAA@QEAVDatatype

^	Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symbols
	WS2HELP.DLL	04/08/2004 15:00	04/08/2004 10:57	19,968	A	0x0000AFF6	0x0000AFF6	x86	Console	CV
	WSOCK32.DLL	04/08/2004 15:00	04/08/2004 10:57	22,528	A	0x00012C31	0x00012C31	x86	Console	CV
	WTSAPI32.DLL	04/08/2004 15:00	04/08/2004 10:57	18,432	A	0x00008C06	0x00008C06	x86	Console	CV
	WZCDLG.DLL	04/08/2004 15:00	04/08/2004 10:57	378,368	A	0x000654B5	0x000654B5	x86	GUI	CV
	WZCSAPI.DLL	04/08/2004 15:00	04/08/2004 10:57	51,712	A	0x0000E183	0x0000E183	x86	Console	CV
	WZCSVC.DLL	04/08/2004 15:00	04/08/2004 10:57	359,936	A	0x0005FD08	0x0005FD08	x86	Console	CV
	XERCES-C_2_2_0.DLL	02/06/2008 15:29	07/02/2003 11:16	8,045,568	A	0x00000000	0x007BB429	IA64	GUI	None

```
Error: At least one module has an unresolved import due to a missing export function in an implicitly dependent module.
Error: Modules with different CPU types were found.
Warning: At least one module has an unresolved import due to a missing export function in a delay-load dependent module.
```

As you can see, the used xerces is intended for IA64, while the system is x86.

In case you're not sure what the right dll is, try the following:

32 or 64 bit ? - check your version of windows in the system properties and if it doesn't say windows xp 64 bit edition then you are running 32 bit

X86 or x64 CPU type ? – this link might help you.

Getting and installing the Axis C++ server

1. You start with downloading the *Axis C++ binary distribution* : You start with the link to the Axis C++ download page .

Now click “Mirror sites” and choose the suggested site (the top most link). This should lead you to the *Apache Software Foundation Distribution Directory*.
Now go to *ws* directory → axis-c direcory → and download axis-c-win32-current-bin.zip

2. Unzip the binary distribution to a folder of your choice. Say this folder is Axis_Extract.

3. Inside the extracted folder Axis_Extract there is a sub folder called deploy. Copy the whole deploy folder to [Apache_Folder].

4. Rename copied "deploy" folder to "Axis". Lets say this folder is [Axis_Folder].

5. Set environment variable AXISCPP_DEPLOY to point to above folder.

6. Add [Axis_Folder]/lib to the PATH environment variable.(this is required since the axis module requires all the dlls reside in this folder).

7. [Axis_Module] is either mod_axis.dll for apache 1.3.X or mod_axis2.dll for apache 2.X. Copy [Axis_Module] from Axis_Extract/bin folder to [Apache_Folder]/modules

9. Copy axis_server.dll,HTTPTransport.dll and HTTPChannel.dll from Axis_Extract/bin folder to [Axis_Folder]/lib

10.Rename AxisXMLParser_Xerces.dll (which is located at Axis_Extract/bin) to AxisXMLParser.dll and copy it to [Axis_Folder]/lib

11. Copy xerces-c_2_2_0.dll from [Xerces_Folder]/bin/ to [Axis_Folder]/lib/

12. log off (no need to restart yet), so that the changes made to the path variable will take effect.

Success test : use Depends.exe to load the following dlls, in this order:

in [Axis_Folder]/lib/ folder:
xerces-c_2_2_0.dll, AxisXMLParser.dll, HTTPTransport.dll, HTTPChannel.dll, axis_server.dll
Now use Depends.exe to load [Apache_Folder]/modules/ [Axis_Module] (for apache 2.x it's mod_axis2.dll)
If all dll's loads well, you're good to go.

Loading the axis module into the server and exposing a web service

1. server.wsdd is located at [Apache_Folder]\Axis\conf\ for Apache 1.3 and Apache 2. (There will be three wsdd files in the [Apache_Folder]\Axis\conf\ rename the relevant wsdd file of your platform to server.wsdd).

In order to use the Calculator sample, you can delete all the content, and use this:

```
<?xml version="1.0" encoding="UTF-8"?>

<deployment xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:CPP="http://xml.apache.org/axis/wsdd/providers/CPP">
        <service name="Calculator" provider="CPP:RPC" description="Axis C++ web service">
                <parameter name="className" value="C:\Program Files\Apache
Group\Apache2\Axis\webservices\Calculator.dll"/>
                <parameter name="allowedMethods" value="add addRequest sub subRequest mul
mulRequest div divRequest "/>
        </service>
</deployment>
```

2. Open the [Axis_Folder]/axiscpp.conf_win for editing. The contents of the file will look as follows.

```
LogPath:XXXX
WSDDFilePath:YYYY
XMLParser:ZZZZ
Transport_http:AAAA
```

```
 Channel_HTTP:BBBB
```

You should assign the correct paths. i.e.

```
LogPath:C:\Program Files\Apache Group\Apache2\Axis\log\Axislog.txt
WSDDFilePath:C:\Program Files\Apache Group\Apache2\Axis\conf\server.wsdd
XMLParser:C:\Program Files\Apache Group\Apache2\Axis\lib\AxisXMLParser.dll
Transport_http:C:\Program Files\Apache Group\Apache2\Axis\lib\HTTPTransport.dll
Channel_HTTP:C:\Program Files\Apache Group\Apache2\Axis\lib\HTTPChannel.dll
```

Note : Make absolutely sure your paths are correct. Copy them and use the file explorer to make sure they are correct. It might save you a lot of pain.

3. Add the following lines to the bottom of Apache configuration file httpd.conf located

in [Apache_Folder]/conf

```
LoadModule axis_module modules/[Axis_Module]

        <Location /axis>

        SetHandler axis
        </Location>
```

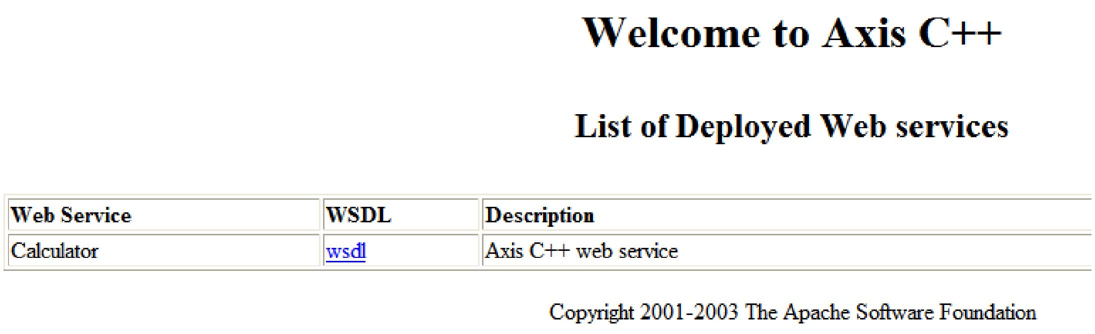
4. At this stage you should restart your computer (don't just log off).

Success test for loading Axis: Restart the apache server (you can do that from programs -> apache http server X.X -> control apache server).

If you have done correctly apache server will start without giving any error.

Success test for exposing the web service: browse to http://localhost/axis

In case all went well, you should get the following screen:



Welcome to Axis C++

List of Deployed Web services

Web Service	WSDL	Description
Calculator	wsdl	Axis C++ web service

Copyright 2001-2003 The Apache Software Foundation

Troubleshooting :

the two common problems are:

1. Error when restarting the apache server. When this occurs, you probably won't see any error in the apache log files, since it's not loaded. You should re-check that all dlls are well loaded using depends.exe, and all the paths are correct. If it still fails, try a restart.

2. Apache server loads successfully but when browsing to http://localhost/axis you

get the Axis C++ welcome page with the following line:

Deployment Descriptor Not Found

First, try the following suggestions (taken from here):

- The most frequent reason for this is that web server doesn't have write access to your log directory specified in the axiscpp.conf!!! Error message is misleading.
- Has not set the correct path to AXIS_HOME in the environment variable.For eg:-AXIS_HOME - C:\Program Files\Apache Group\Apache\Axis for Apache 1.3.x
- Has not placed Axiscpp.conf in the correct location.For eg: - Axis/axiscpp.conf
- Server.wsdd is not in the correct location.For eg:- [APACHE_HOME]\Axis\conf\server.wsdd

Or server.wsdd contains syntax errors.

In case none of the above helps, restart apache, and look at the log file placed under [apache install]/logs/error.log.

You might find the following error:

An error occurred while Axis was initializing: DLOPEN FAILED in loading parser library DLOPEN FAILED in loading parser library Failed to load parser within server engine:

Error Message='The specified module could not be found.'
Error Code='126'

Load lib error=''

This is caused since axis tries to load the AxisXMLParser.dll, which in turn, tries to load xerces-c_2_2_0.dll.

So first, use depends.exe to check if xerces-c_2_2_0.dll loads well.

After checking (using Depends.exe) that xerces-c_2_2_0.dll loads well (without errors. Warning is OK), check AxisXMLParser.dll and verify it loads as well.
Still doesn't work? Try these:

```
1. AXISCPP_DEPLOY environment variable is set
2. axiscpp.conf is in folder pointed by %AXISCPP_DEPLOY%
3. axiscpp.conf settings contain full paths
4. The folder containing Xerces 2.2.0 DLL is in your PATH variable
```

5. The folder containing Axis CPP DLLs are in your PATH variable

Check that the XMLParser path appears in axiscpp.config is correct (please, copy it, and run explorer with the copied path, you may be surprised).

Still not working ? Restart your computer.

2. Creating the web service

Using WSDL2WS

See the instructions provided in the Axis C++ Windows User Guide under “How to use the WSDL2WS tool on the command line” section.

Creating the Calculator.dll

In the previous section we started with a WSDL file, and the outcome is a bunch of files that will be used in order to create the Calculator.dll, which will later be used by Axis.

Calculator.dll is the implementation of the web service – the component which exposes methods as a web service, and returns results.

You can use one of the following documents in order to create the calculator.dll:

1. End-to-End Calculator Sample Tutorial – the relevant section is "*Building the server side code*". According to this document, you can create the Calculator.dll using a single command line call. In case it works for you, great. I had problems with calling the cl.exe, so I did it using the following way:
2. Axis C++ Windows User Guide - the relevant section is "*Creating the web service* ". According to this document, you can create the Calculator.dll using VC (I used Microsoft visual studio 6.0). In the following section I copied the instructions from the document, and added my remarks (in red) for this to actually work:

(It starts from #3, since 1-2 refer to the WSDL2WS outcomes)

3) Create a VC workspace.(name it CalculatorWS)

4) Create a 'Win32 Static Library' project in this workspace.(name it MyCalculator)

5) From the generated 'ServerOut'folder (the wsdl2ws output folder), add the following files to this project.
Calculator.cpp Calculator.h (or calculator.hpp)
AxisServiceException.cpp AxisServiceException.hpp

6) Set the include path to the include directory of the binary distribution (These include files are in Axis_Extract/include/).

7) Fill the empty methods of the generated skeletons.

8) Generate the lib (eg: MyCalculator.lib)

9) Now create a 'Win32 Dynamic-Link Library' project.(name it Calculator)

10) From the generated 'ServerOut'folder,add the following files to this project.
CalculatorService.cpp, CalculatorWrapper.cpp and CalculatorWrapper.h AxisServiceException.hpp Calculator.hpp

11) Set the include path to the include directory of the binary distribution.

12) Add the above created lib (MyCalculator.lib) as the input library of this project. In addition add `<inst_dir>/lib/axis/axis_server.lib` as another input library of this project

13) Build and create the DLL. (Calculator.dll)

In case it compiles well, you are good to go with this dll.

If all is well, you should get a similar page:

Now it's time to call the deployed web services.

Now you'll use the Axis C++ Windows User Guide to build the web service:
Using the wsdl2ws tool

Issues encountered when trying to call the web service:
Library loading failed**Library loading failed – Library loading failed – you get a**

Library loading failed – Library loading failedLibrary loading failed – you get a AxisEngineException:Library loading failed Library loading failedLibrary loading failed – you get a AxisEngineException:Library loading failed exception.Library loading failed

Possible resolution:

AxisEngineException means that this is an Axis exception (as oppose to apache exception), so the axis log file might come in handy. The log file path appears in axiscpp.conf -> LogPath (e.g C:\Program Files\Apache Group\Apache2\Axis\log\Axislog.txt).

Look for the "Library loading failed" inside the log file. You might find something like these: (I omitted the timestamp)

```
WSDDDeployment < getLibName @005C5208,@1 "Calculator"
[andlerLoader > loadLib 04112100
[002111040000000000000000002111104FFFFFFD0505C0000000000000000000000000
0] <.!...........!..ıP\............>
[callLoadLib [04211104] <.!..>
- < callLoadLib [00000000] <....>
- > getPlatformErrorMessage 126- < getPlatformErrorMessage 005C6320
[20635C00003A080020635C00FFFFFF88FFFFFFF7FFFFFF8004FFFFFFCB6C7F00FF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE8FFFFFFF5FFFFFF8004] < c\..:.. c\.ˆ
ק€..ֽl......ץט€.>
 AxisEngineException > AxisEngineException @0480F1A4,------------>
- I 34, "Failed to load handler within server engine:
Error Message='The specified module could not be found.Error Code='126'
Load lib error="
AxisEngineException > getMessageForExceptionCode @0480F1A4,34
AxisEngineException < getMessageForExceptionCode
@0480F1A4,"AxisEngineException:Library loading failed"
```

As you can see, the library that failed to load is the calculator. Why?
First, try to load it using Depends.exe, and solve any issues.
Still doesn't work?
Make sure you used the /MD flag when building the application or DLL used by axis.
In case it's still doesn't work, than it's probably a problem related to the used dll. Go back to the "Creating the Calculator.dll" section to get a working dll.