# Apache Axis2 Installation Guide



This document provides information on Axis2 distribution packages, system prerequisites and setting up environment variables and tools followed by detailed instructions on installation methods.

*Send your feedback to: java-dev@axis.apache.org* mailing list. (Subscription details are available on Axis2 site.) Kindly prefix every email subject with [Axis2].

## Contents

## Axis2 Distributions

Axis2 is distributed in several convenient distribution packages and can be installed either as a standalone server or as part of a J2EE compliant servlet container. Axis2 is distributed under the Apache License, version 2.0. This Installation Guide will mainly focus on running Apache Axis2 using the Standard Binary Distribution.

Download distribution packages of the Apache Axis2 1.7.9 version (latest).

Download distribution packages of all versions of Apache Axis2.

The distribution packages provided are as follows:

### 1. Standard Binary Distribution

This is the complete version of Axis2 and includes samples and convenient scripts as well.

Download Standard Binary Distribution

### 2. WAR (Web Archive) Distribution

This is the Web application of Axis2, which can be deployed in most of the servlet containers.

Download WAR (Web Archive) Distribution

### 3. Documents Distribution

This contains all the documentation in one package. The package includes the xdocs and the Java API docs of this project.

Download Documents Distribution

### 4. Source Distribution

This contains the sources of Axis2 standard distribution, and is mainly for the benefit of advanced users. One can generate a binary distribution using the source by typing $mvn -Drelease install. You need to set up the Axis2 environment before running this command. Step by step details on how to create the binary distribution is available in the Advanced section.

Download Source Distribution

## System Requirements

| | |
|---|---|
| Java Development Kit (JDK) | 1.5 or later (For instructions on setting up the JDK in different operating systems, visit http://java.sun.com) |
| Disk | Approximately 11 MB separately for standard binary distribution |
| Operating system | Tested on Windows XP, Linux, Mac OS X, Fedora core, Ubuntu, Gentoo |
| **Build Tool**-Apache Ant<br>To run samples and to build WAR files from Axis2 binary distribution. | Version 1.6.5 or higher (download). |
| **Build Tool**- Apache Maven 2.x<br>Required *only* for building Axis2 from Source Distribution | 2.0.7 or higher in Maven 2.x series (download). **Please download Maven 2.x version. Axis2 does not support Maven 1.x anymore.** |

Make sure that the above prerequisites are available for the Axis2 installation.

## Installing Axis2 as a Standalone Server using the Standard Binary Distribution

This section provides you with the following information

1. Install Axis2 as a standalone server using the Standard Binary Distribution
2. Start up the Axis2 standalone server
3. Building the axis2.war file (using the Standard Binary Distribution) which is required to

run Axis2 as part of a J2EE compliant servlet container
4. Running Axis2 convenient scripts

## 1. Download and Install the Apache Axis2 Binary Distribution

Download and install a Java Development Kit (JDK) release (version 1.5 or later). Install the JDK according to the instructions included with the release. Set an environment variable JAVA_HOME to the pathname of the directory into which you installed the JDK release.

Download and unpack the Axis2 Standard Binary Distribution into a convenient location so that the distribution resides in its own directory. Set an environment variable AXIS2_HOME to the pathname of the extracted directory of Axis2 (Eg: /opt/axis2-1.7.9). Linux users can alternatively run the setenv.sh file available in the AXIS2_HOME/bin directory to set the AXIS2_HOME environment variable to the Axis2 classpath.

## 2. Starting up Axis2 Standalone Server

The standalone Axis2 server can be started by executing the following commands:
%AXIS2_HOME%\bin\axis2server.bat (Windows)
$AXIS2_HOME/bin/axis2server.sh (Unix)

After startup, the default web services included with Axis2 will be available by visiting http://localhost:8080/axis2/services/

## 3. Building the Axis2 Web Application (axis2.war) Using Standard Binary Distribution

Download and install Apache Ant (version 1.6.5 or later). Install Apache Ant according to the instructions included with the Ant release.

Locate the Ant build file (build.xml) inside the webapp directory, which resides in your Axis2 home directory (i.e:- $AXIS_HOME/webapp)". Run the Ant build by executing "ant create.war" inside the AXIS2_HOME/webapps folder. You can find the generated axis2.war inside the AXIS2_HOME/dist directory. All the services and modules that are present in the AXIS2_HOME/repository will be packed into the created axis2.war together with the Axis2 configuration found at AXIS2_HOME/conf/axis2.xml.

Read Installing Axis2 in a Servlet Container to find out how to deploy the Axis2 Web application in a servlet container.

## 4. Getting Familiar with the Convenient Axis2 Scripts

It is advised to add the AXIS2_HOME/bin to the PATH, so that you'll be able to run the following scripts from anywhere.

| Script Name | Description |
|---|---|

| | |
|---|---|
| axis2.{bat\|sh} | You can use this script to run web service clients written using Axis2. This script calls the "java" command after adding the classpath for Axis2 dependent libraries (*.jar files present in your AXIS2_HOME/lib), setting the Axis2 repository location (AXIS2_HOME/repository) and setting the Axis2 configuration file location(AXIS2_HOME/conf/axis2.xml) for you. With this you can be relieved from setting all the above Axis2 specific parameters.<br>*Usage : axis2.{sh.bat} [-options] class [args...]* |
| axis2server.{sh\|bat} | This script will start a standalone Axis2 server using the AXIS2_HOME/repository as the Axis2 repository and the AXIS2_HOME/conf/axis2.xml as the Axis2 configuration file. This will start all the transport listeners listed in the AXIS2_HOME/conf/axis2.xml.<br>For example, if you want to deploy a service using a standalone Axis2 server,then copy your service archive to the AXIS2_HOME/repository/services directory. Next, go to the "Transport Ins" section of the AXIS2_HOME/conf/axis2.xml and configure the transport receivers (simpleHttpServer in port 8080 is listed by default). Then invoke this script.<br><br>The server can be started in debug mode by adding the `-xdebug` option to the command line. A remote debugger can then be attached by connecting to port 8000. |
| wsdl2java.{bat\|sh} | This script generates Java code according to a given WSDL file to handle Web service invocations (client-side stubs). This script also has the ability to generate web service skeletons according to the given WSDL.<br>*Usage: wsdl2java.{sh\|bat} [OPTION]... -uri <Location of WSDL>*<br><br>e.g., wsdl2java.sh -uri ../wsdl/Axis2Sample.wsdl<br><br>A more detailed reference about this script can be found here |
| java2wsdl.{bat\|sh} | This script generates the appropriate WSDL file for a given Java class.<br>*Usage: Java2WSDL.{sh\|bat} [OPTION]... -cn <fully qualified class name>*<br><br>e.g., Java2WSDL.sh -cn ../samples/test/searchTool.Search<br><br>A more detailed reference about this script can be found here |

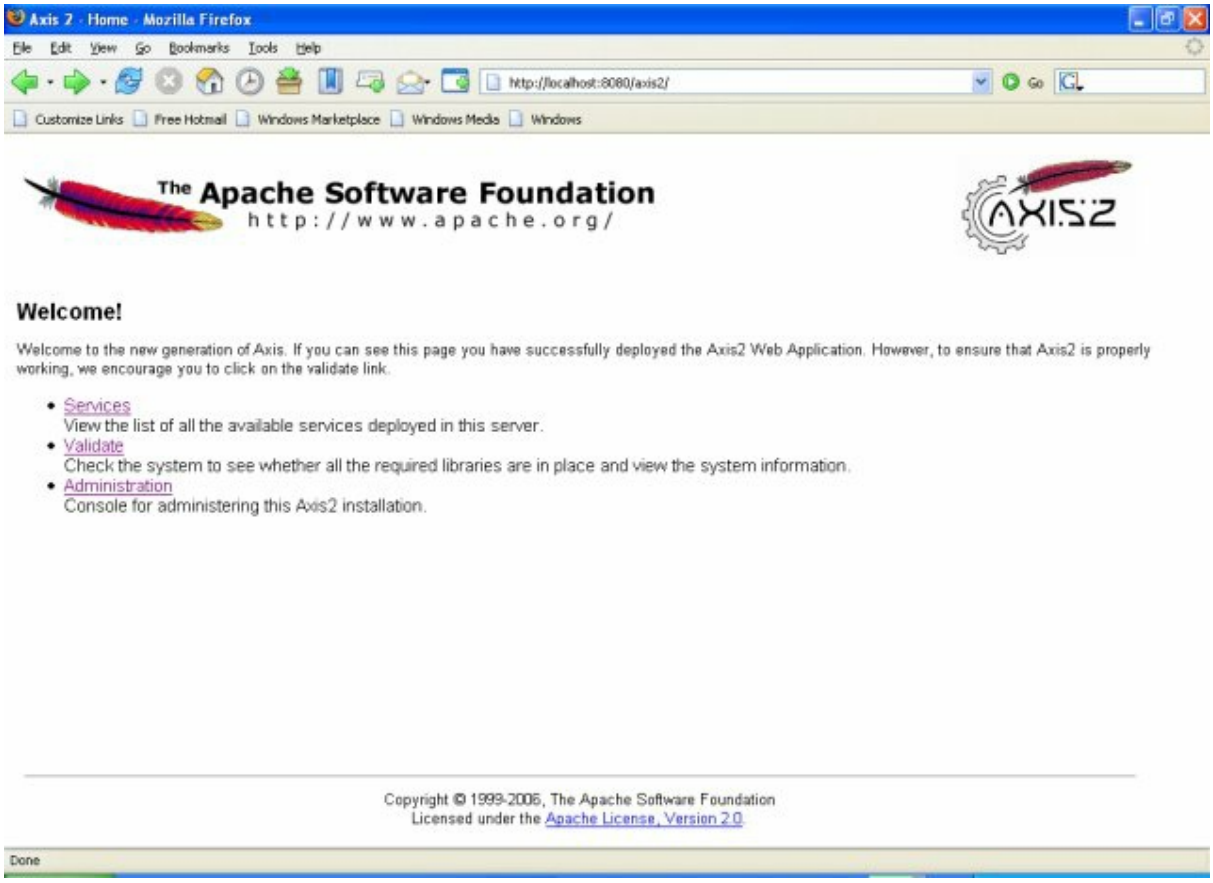
## Installing Axis2 in a Servlet Container

Whichever the distribution, installing Axis2 in a J2EE compliant servlet container is as follows:

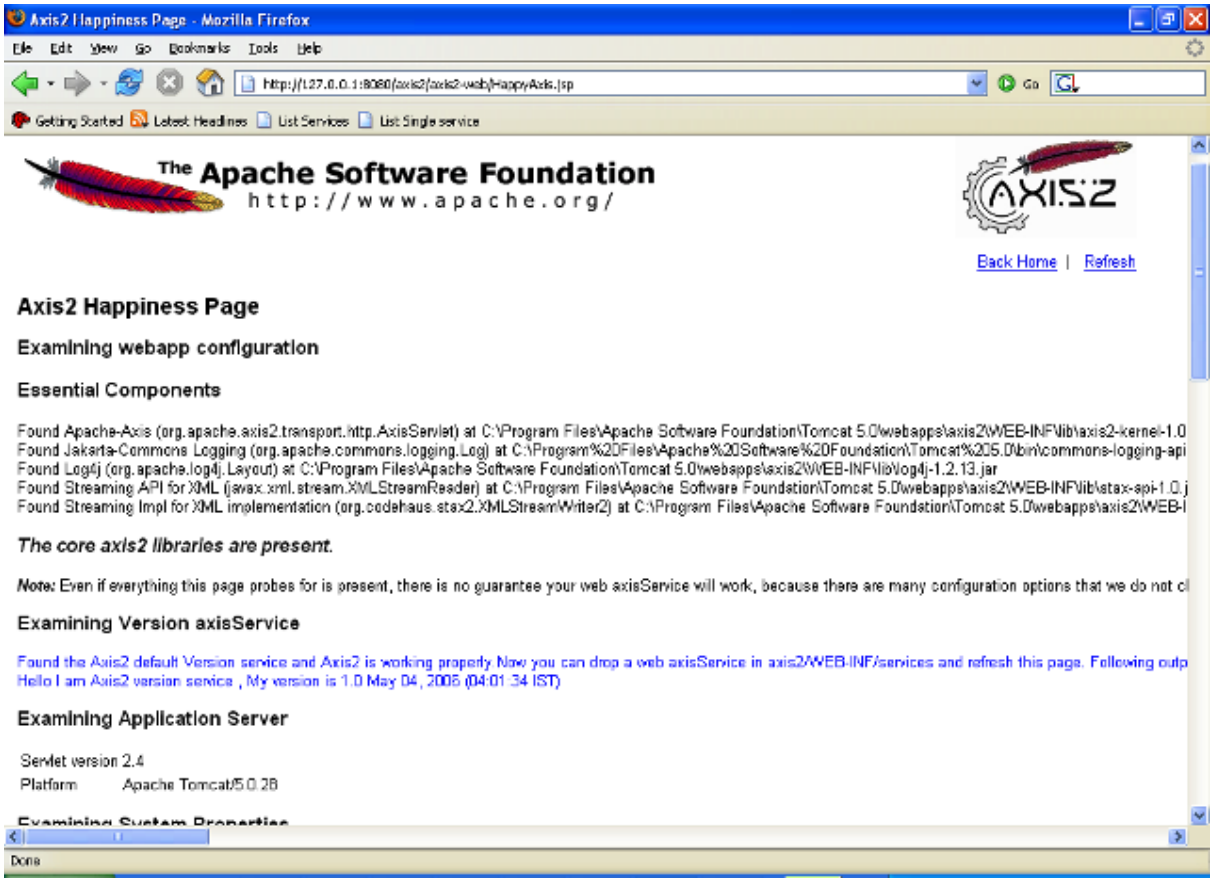
1. Build the Axis2 WAR file using the Axis2 Standard Binary Distribution. (Alternatively you

can download the axis2.war file or you can build axis2.war using the Source Distribution.

2. Drop the WAR file in the webapps folder of the servlet container. Most servlet containers will automatically install the WAR file. (Some servlet containers may require a restart in order to capture the new web application. Refer to your servlet container documentation for more information.)
3. Once the WAR is successfully installed, test it by pointing the web browser to the **http://<host :port>/axis2.** It should produce the following page which is the **Axis2 Web Application Home Page**.




4. Use the link "Validate" to ensure that everything is running correctly. If the validation fails then the WAR has failed to install properly or some essential jars are missing. In such a situation, refer to the documentation of the particular servlet container to find the problem. The following page shows a successful validation. Note the statement that indicates the core Axis2 libraries are present.




**Note:** For any Application server specific installation information please refer to the Application Server Specific Configuration Guide.

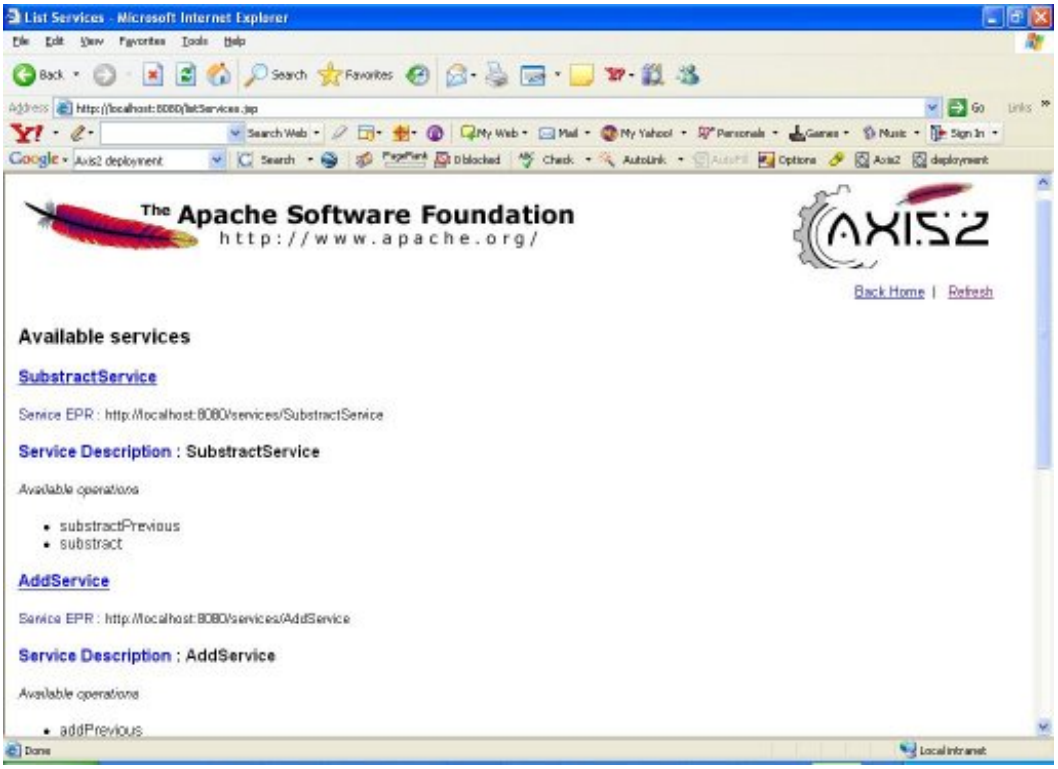
## Uploading Services

The Axis2 Web application also provides an interface to upload services. Once a service archive file is created according to the service specification as described in the Advanced User's Guide, that .aar file can be uploaded using the upload page.

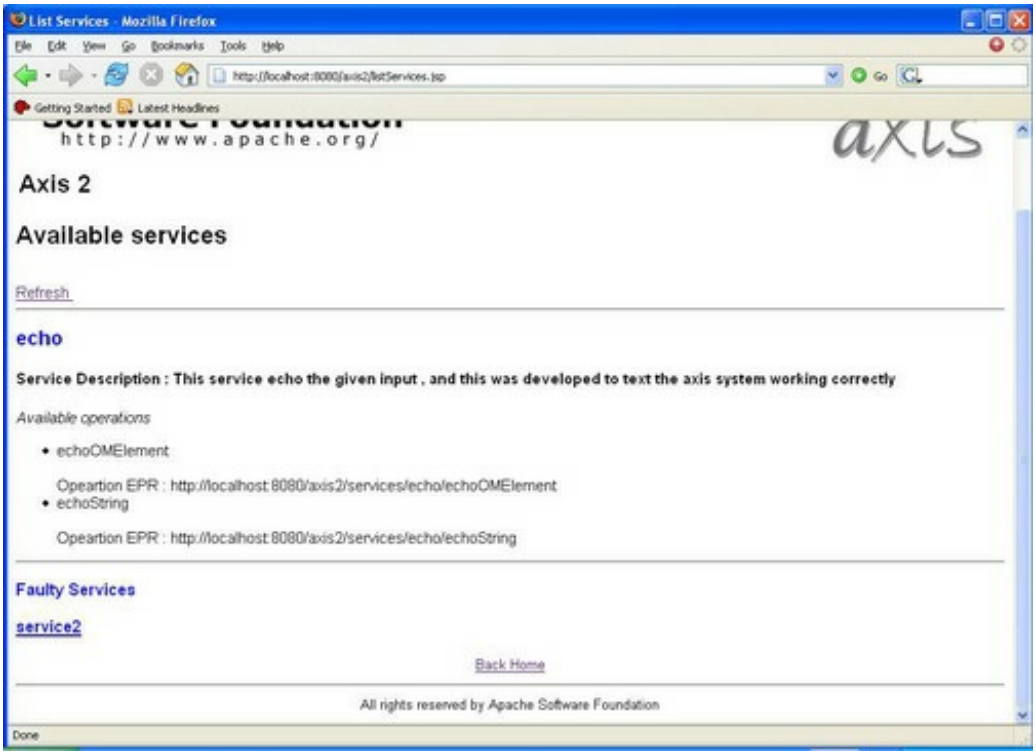

The uploaded .aar files will be stored in the default service directory. For Axis2, this will be the <webapps>/axis2/WEB-INF/services directory. Once a service is uploaded, it will be installed instantly.

Since Axis2 supports **hot deployment**, you can drop the service archive directly through the file system to the above mentioned services directory. It will also cause the service to be automatically installed without the container being restarted.

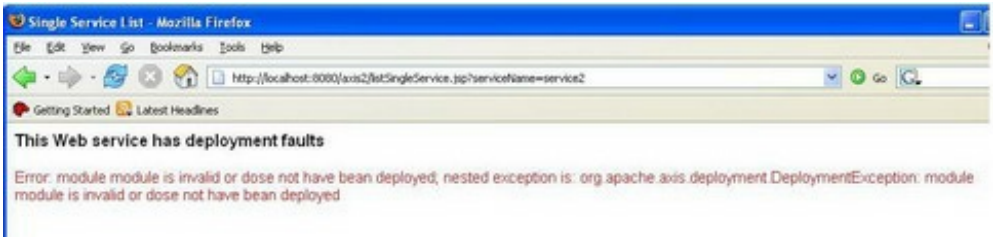
Use the 'Services' link on the Web Application home page to check the successful installation of a service. The services and the operations of successfully installed services will be displayed on the available services page.




If the service has deployment time errors it will list those services as faulty services. If you click on the link, you will see the deployment fault error messages.




Deployment time error message




Axis2 Administration is all about configuring Axis2 at the run time and the configuration will be transient. More descriptions are available in the Axis2 Web Administration Guide

# Advanced

# Axis2 Source Distribution

By using the Source Distribution, both binary files (which can be downloaded as the Standard Binary Distribution) and the axis2.war file (which can be downloaded as the WAR distribution) can be built using Maven commands.
Required jar files do not come with the distribution and they will also have to be built by running the maven command. Before we go any further, it is necessary to install Maven2 and set up its environment, as explained below.

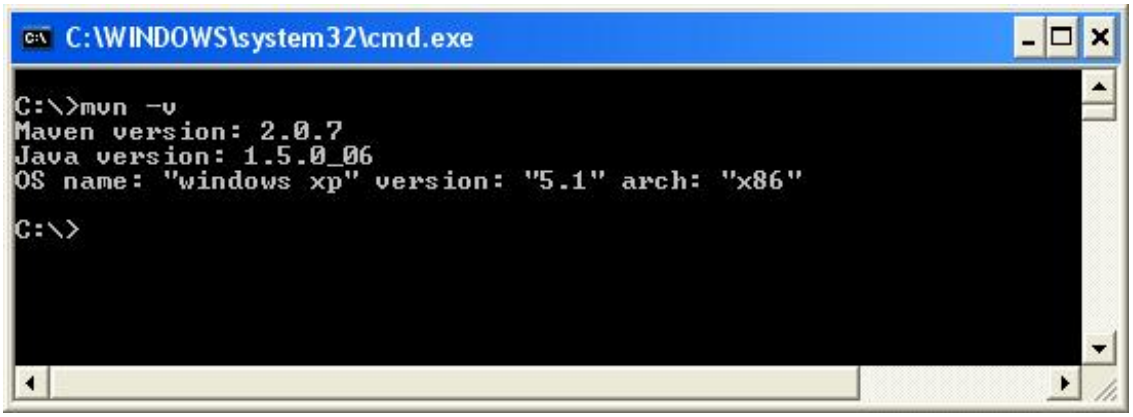
## Setting Up the Environment and Tools

### Maven

The Axis2 build is based on Maven2 . Hence the only prerequisite to build Axis2 from the source distribution is to have Maven installed. Extensive instruction guides are available at the Maven site. This guide however contains the easiest path for quick environment setting. Advanced users who wish to know more about Maven can visit this site.

MS Windows

1. Download and run the Windows installer package for Maven.
2. Set the 'Environment Variables' ( create system variable MAVEN_HOME and edit path. eg: "C:\Program Files\Apache Software Foundation\maven-2.0.7"; path %MAVEN_HOME%\bin)
3. Make sure that the system variable JAVA_HOME is set to the location of your JDK, eg. C:\Program Files\Java\jdk1.5.0_11
4. Run mvn -v or mvn -version to verify that it is correctly installed.




Unix based OS (Linux etc)

The tar ball or the zip archive is the best option. Once the archive is downloaded expand it to a directory of choice and set the environment variable MAVEN_HOME and add MAVEN_HOME/bin to the path as well. More instructions for installing Maven in Unix based operating systems.

Once Maven is properly installed, you can start building Axis2.

Maven commands that are frequently used in Axis2 are listed on the FAQs page.

## Building Binaries and the WAR File Using the Source Distribution

The Source Distribution is available as a zipped archive. All the necessary build scripts are included with the source distribution. Once the source archive is expanded into a directory of choice, moving to the particular directory and running `mvn install` command will build the Axis2 jar file.

Once the command completes, the binaries (jar files in this case) can be found at a newly created "target" directory.

**Note: For the first Maven build (if the maven repository is not built first) it will take a while since the required jars need to be downloaded. However, this is a once only process and will not affect any successive builds.**

The default maven build will generate the war under modules/webapp/target directory