

Installing Ant

Getting Ant

Binary Edition

The latest stable version of Ant is available from the Ant web page <http://ant.apache.org/>. If you like living on the edge, you can download the latest version from <http://cvs.apache.org/builds/ant/nightly/>.

Source Edition

If you prefer the source edition, you can download the source for the latest Ant release from <http://ant.apache.org/srcdownload.cgi>. Again, if you prefer the edge, you can access the code as it is being developed via CVS. The Jakarta website has details on [accessing CVS](#). Please checkout the ant module. See the section [Building Ant](#) on how to build Ant from the source code. You can also access the [Ant CVS repository](#) on-line.

System Requirements

Ant has been used successfully on many platforms, including Linux, commercial flavours of Unix such as Solaris and HP-UX, Windows 9x and NT, OS/2 Warp, Novell Netware 6 and MacOS X.

To build and use Ant, you must have a JAXP-compliant XML parser installed and available on your classpath.

The binary distribution of Ant includes the latest version of the [Apache Xerces2](#) XML parser. Please see <http://java.sun.com/xml/> for more information about JAXP. If you wish to use a different JAXP-compliant parser, you should remove `xercesImpl.jar` and `xml-apis.jar` from Ant's `lib` directory. You can then either put the jars from your preferred parser into Ant's `lib` directory or put the jars on the system classpath.

For the current version of Ant, you will also need a JDK installed on your system, version 1.2 or later.

Note: The Microsoft JVM/JDK is not supported.

Note #2: If a JDK is not present, only the JRE runtime, then many tasks will not work.

Installing Ant

The binary distribution of Ant consists of the following directory layout:

```
ant
+--- bin // contains launcher scripts
|
+--- lib // contains Ant jars plus necessary dependencies
|
+--- docs // contains documentation
|     +--- ant2 // a brief description of ant2 requirements
|     |
```

```
|      +--- images  // various logos for html documentation
|      |
|      +--- manual  // Ant documentation (a must read ;-)
+--- etc // contains xsl goodies to:
      // - create an enhanced report from xml output of various tasks.
      // - migrate your build files and get rid of 'deprecated' warning
      // - ... and more ;-)
```

Only the `bin` and `lib` directories are required to run Ant. To install Ant, choose a directory and copy the distribution file there. This directory will be known as `ANT_HOME`.

Windows 95, Windows 98 & Windows ME Note:

*On these systems, the script used to launch Ant will have problems if `ANT_HOME` is a long filename (i.e. a filename which is not of the format known as "8.3"). This is due to limitations in the OS's handling of the "FOR" batch-file statement. It is recommended, therefore, that Ant be installed in a **short**, 8.3 path, such as `C:\Ant`.*

On these systems you will also need to configure more environment space to cater for the environment variables used in the Ant launch script. To do this, you will need to add or update the following line in the `config.sys` file

```
shell=c:\command.com c:\ /p /e:32768
```

Setup

Before you can run ant there is some additional set up you will need to do:

- Add the `bin` directory to your path.
- Set the `ANT_HOME` environment variable to the directory where you installed Ant. On some operating systems the ant wrapper scripts can guess `ANT_HOME` (Unix dialects and Windows NT/2000) - but it is better to not rely on this behavior.
- Optionally, set the `JAVA_HOME` environment variable (see the [Advanced](#) section below). This should be set to the directory where your JDK is installed.

Note: Do not install Ant's `ant.jar` file into the `lib/ext` directory of the JDK/JRE. Ant is an application, whilst the extension directory is intended for JDK extensions. In particular there are security restrictions on the classes which may be loaded by an extension.

Optional Tasks

Ant supports a number of optional tasks. An optional task is a task which typically requires an external library to function. The optional tasks are packaged together with the core Ant tasks.

The external libraries required by each of the optional tasks is detailed in the [Library Dependencies](#) section. These external libraries may either be placed in Ant's `lib` directory, where they will be picked up automatically, or made available on the system `CLASSPATH` environment variable.

Windows and OS/2

Assume Ant is installed in `c:\ant\`. The following sets up the environment:

```
set ANT_HOME=c:\ant
```

```
set JAVA_HOME=c:\jdk1.2.2
set PATH=%PATH%;%ANT_HOME%\bin
```

Unix (bash)

Assume Ant is installed in `/usr/local/ant`. The following sets up the environment:

```
export ANT_HOME=/usr/local/ant
export JAVA_HOME=/usr/local/jdk-1.2.2
export PATH=${PATH}:${ANT_HOME}/bin
```

Unix (csh)

```
setenv ANT_HOME /usr/local/ant
setenv JAVA_HOME /usr/local/jdk-1.2.2
set path=( $path $ANT_HOME/bin )
```

Advanced

There are lots of variants that can be used to run Ant. What you need is at least the following:

- The classpath for Ant must contain `ant.jar` and any jars/classes needed for your chosen JAXP-compliant XML parser.
- When you need JDK functionality (such as for the [javac](#) task or the [rmic](#) task), then for JDK 1.1, the `classes.zip` file of the JDK must be added to the classpath; for JDK 1.2 or JDK 1.3, `tools.jar` must be added. The scripts supplied with Ant, in the `bin` directory, will add the required JDK classes automatically, if the `JAVA_HOME` environment variable is set.
- When you are executing platform-specific applications, such as the [exec](#) task or the [cvs](#) task, the property `ant.home` must be set to the directory containing where you installed Ant. Again this is set by the Ant scripts to the value of the `ANT_HOME` environment variable.

The supplied ant shell scripts all support an `ANT_OPTS` environment variable which can be used to supply extra options to ant. Some of the scripts also read in an extra script stored in the users home directory, which can be used to set such options. Look at the source for your platform's invocation script for details.

Building Ant

To build Ant from source, you can either install the Ant source distribution or checkout the ant module from CVS.

Once you have installed the source, change into the installation directory.

Set the `JAVA_HOME` environment variable to the directory where the JDK is installed. See [Installing Ant](#) for examples on how to do this for your operating system.

Note: The bootstrap process of Ant requires a greedy compiler like Sun's `javac` or `jikes`. It does not work with `gcj` or `kjc`.

Make sure you have downloaded any auxiliary jars required to build tasks you are interested in. These should either be available on the `CLASSPATH` or added to the `lib` directory. See [Library Dependencies](#) for a list of jar requirements for various features. Note that this will make the auxiliary jars available for the building of Ant only. For running Ant you will still need to make the jars available as described under [Installing Ant](#).

You are now ready to build Ant:

```
build -Ddist.dir=<directory_to_contain_Ant_distribution> dist (Windows)
```

```
build.sh -Ddist.dir=<directory_to_contain_Ant_distribution> dist (Unix)
```

This will create a binary distribution of Ant in the directory you specified.

The above action does the following:

- If necessary it will bootstrap the Ant code. Bootstrapping involves the manual compilation of enough Ant code to be able to run Ant. The bootstrapped Ant is used for the remainder of the build steps.
- Invokes the bootstrapped Ant with the parameters passed to the build script. In this case, these parameters define an Ant property value and specify the "dist" target in Ant's own `build.xml` file.

On most occasions you will not need to explicitly bootstrap Ant since the build scripts do that for you. If however, the build file you are using makes use of features not yet compiled into the bootstrapped Ant, you will need to manually bootstrap. Run `bootstrap.bat` (Windows) or `bootstrap.sh` (UNIX) to build a new bootstrap version of Ant.

If you wish to install the build into the current `ANT_HOME` directory, you can use:

```
build install (Windows)
```

```
build.sh install (Unix)
```

You can avoid the lengthy Javadoc step, if desired, with:

```
build install-lite (Windows)
```

```
build.sh install-lite (Unix)
```

This will only install the `bin` and `lib` directories.

Both the `install` and `install-lite` targets will overwrite the current Ant version in `ANT_HOME`.

Library Dependencies

The following libraries are needed in your `CLASSPATH` or in the install directory's `lib` directory if you are using the indicated feature. Note that only one of the `regexp` libraries is needed for use with the mappers (and Java 1.4 and higher includes a `regexp` implementation which Ant will find automatically). You will also need to install the Ant optional jar containing the task definitions to make these tasks available. Please refer to the [Installing Ant / Optional Tasks](#) section above.

Jar Name	Needed For	Available At
An XSL transformer like Xalan or XSL:P	style task	<p>If you use JDK 1.4, an XSL transformer is already included, so you need not do anything special.</p> <ul style="list-style-type: none"> • XALAN :

		http://xml.apache.org/xalan-j/index.html <ul style="list-style-type: none"> • XSL:P : used to live at http://www.clc-marketing.com/xslp/, but the link doesn't work any longer and we are not aware of a replacement site.
jakarta-regexp-1.3.jar	regexp type with mappers	http://jakarta.apache.org/regexp/
jakarta-oro-2.0.7.jar	regexp type with mappers and the performe tasks To use the FTP task, you need jakarta-oro 2.0.1 or later, and commons-net	http://jakarta.apache.org/oro/
junit.jar	junit tasks	http://www.junit.org/
xalan.jar	junitreport task	http://xml.apache.org/xalan-j/
stylebook.jar	stylebook task	CVS repository of http://xml.apache.org/
testlet.jar	deprecated test task	Build from the gzip compress tar archive in http://avalon.apache.org/historiccv/testlet/
antlr.jar	antlr task	http://www.antlr.org/
bsf.jar	script task Note: Ant 1.6 and later require Apache BSF, not the IBM version. I.e. you need BSF 2.3.0-rc1 or later.	http://jakarta.apache.org/bsf/
Groovy jars	Groovy with script and scriptdef tasks You need to get the groovy jar and two asm jars from a groovy installation. The jars are groovy-[version].jar, asm-[vesion].jar and asm-util-[version].jar. As of groovy version 1.0-beta-7, the jars are groovy-1.0-beta-7.jar, asm-1.4.3.jar and asm-util-1.4.3.jar.	http://groovy.codehaus.org/ The asm jars are also available from the creators of asm - http://asm.objectweb.org/
netrexx.jar	netrexx task, Rexx with the script task	http://www2.hursley.ibm.com/netrexx/
js.jar	Javascript with script task If you use Apache BSF 2.3.0-rc1, you must use rhino 1.5R3 - later versions of BSF work with 1.5R4 as well.	http://www.mozilla.org/rhino/
jython.jar	Python with script task Warning : jython.jar also contains classes from jakarta-oro. Remove these classes if you are also using jakarta-oro.	http://jython.sourceforge.net/
jpython.jar	Python with script task deprecated, jython is the preferred engine	http://www.jpython.org/
jacl.jar and tcljava.jar	TCL with script task	http://www.scriptics.com/software/java/
BeanShell JAR(s)	BeanShell with script task. Note: Ant 1.6 and later require BeanShell version 1.3	http://www.beanshell.org/

	or later	
jruby.jar	Ruby with script task	http://jruby.sourceforge.net/
judo.jar	Judoscript with script task	http://www.judoscript.com/index.html
commons-logging.jar	CommonsLoggingListener	http://jakarta.apache.org/commons/logging/index.html
log4j.jar	Log4jListener	http://jakarta.apache.org/log4j/docs/index.html
commons-net.jar	ftp, rexec and telnet tasks jakarta-oro 2.0.1 or later is required in any case together with commons-net. For all users, a minimum version of commons-net of 1.2.2 is recommended. Earlier versions did not support autodetection of system type or had significant bugs.	http://jakarta.apache.org/commons/net/index.html
bcel.jar	classfileset data type, JavaClassHelper used by the ClassConstants filter reader and optionally used by ejbjar for dependency determination	http://jakarta.apache.org/bcel/
mail.jar	Mail task with Mime encoding, and the MimeMail task	http://java.sun.com/products/javamail/
jsse.jar	Support for SMTP over TLS/SSL in the Mail task Already included in jdk 1.4	http://java.sun.com/products/jsse/
activation.jar	Mail task with Mime encoding, and the MimeMail task	http://java.sun.com/products/javabeans/glasgow/jaf.html
jdepend.jar	jdepend task	http://www.clarkware.com/software/JDepend.html
resolver.jar 1.1beta or later	xmlcatalog datatype <i>only if support for external catalog files is desired</i>	http://xml.apache.org/commons/ .
jsch.jar	sshexec and scp tasks	http://www.jcraft.com/jsch/index.html
JAI - Java Advanced Imaging	image task	http://java.sun.com/products/java-media/jai/
IContract	icontract task Warning : the icontract jar file contains also antlr classes. To make the antlr task work properly, remove antlr/ANTLRGrammarParseBehavior.class from the icontract jar file installed under \$ANT_HOME/lib.	http://www.reliable-systems.com/tools/