<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<!-- This file is an ANT build script. ANT is a Java based build tool. -->

<!-- It is available from http://ant.apache.org -->

<project name="log4j" default="usage" basedir="." >

<!-- The build.properties file defines the path to local jar files -->

<property file="build.properties"/>

<!-- if a Maven build has been attempted,

the Maven repository can provide all the dependencies. -->

<property name="m2\_repo" location="${user.home}/.m2/repository"/>

<property name="javamail.jar" location="${m2\_repo}/javax/mail/mail/1.4.3/mail-1.4.3.jar"/>

<property name="geronimo-jms.jar"

value="${m2\_repo}/org/apache/geronimo/specs/geronimo-jms\_1.1\_spec/1.0/geronimo-jms\_1.1\_spec-1.0.jar"/>

<available property="jms.jar" value="${geronimo-jms.jar}" file="${geronimo-jms.jar}"/>

<property name="jms.jar" location="${m2\_repo}/javax/jms/jms/1.1/jms-1.1.jar"/>

<property name="jmx.jar" location="${m2\_repo}/com/sun/jmx/jmxri/1.2.1/jmxri-1.2.1.jar"/>

<!-- Read the system environment variables and stores them in properties, -->

<!-- prefixed with "env". -->

<property environment="env"/>

<property name="version" value="1.2.17"/>

<!-- The base directory relative to which most targets are built -->

<property name="base" value="."/>

<property name="deprecation" value="on"/>

<!-- The directory where java source files are stored. -->

<property name="java.source.dir" value="src/main/java/"/>

<!-- The directory where resource files are stored. -->

<property name="resource.source.dir" value="src/main/resources/"/>

<!-- The directory where javadoc files are stored. -->

<property name="javadoc.source.dir" value="src/main/javadoc/"/>

<!-- The directory where the package-list file is found, ./ or -->

<!-- build/ -->

<property name="packaging.dir" value="build"/>

<!-- distribution directory -->

<property name="dist.dir" value="dist"/>

<!-- Destination for compiled files -->

<property name="javac.dest" value="${dist.dir}/classes"/>

<!-- Destination for generated jar files -->

<property name="jar.dest" value="${dist.dir}/lib"/>

<!-- The jar file that the jar task will generate -->

<property name="jar.filename" value="log4j-${version}.jar"/>

<property name="sources-jar.filename" value="log4j-${version}-sources.jar"/>

<!-- Destination for documentation files -->

<property name="docs.dest" value="./docs"/>

<!-- Source directory for xml docs -->

<property name="xdocs.src" value="./src/xdocs"/>

<!-- Destination for javadoc generated files -->

<property name="javadoc.dest" value="docs/api"/>

<!-- Icons source directory. -->

<property name="icons.source" value="icons"/>

<!-- The stem where most log4j source code is located. -->

<property name="stem" value="org/apache/log4j"/>

<!-- Some targets needs a more precise stem. -->

<property name="BSTEM" value="${java.source.dir}/${stem}"/>

<!-- Directory where release images go. -->

<property name="dist.images" value="${dist.dir}/images"/>

<!-- Directory for temporary files. -->

<property name="dist.tmp" value="${dist.dir}/tmp"/>

<property name="javac.source" value="1.4"/>

<property name="javac.target" value="1.4"/>

<!-- destination for generated documentation on Apache web host -->

<property name="apache.javadoc\_dest" value="/www/logging.apache.org/log4j/docs"/>

<!-- Apache web host -->

<property name="apache.host" value="people.apache.org"/>

<!-- destination for Maven generated documentation -->

<property name="svnrepo.url" value="https://svn.apache.org/repos/asf"/>

<property name="svnsite.url" value="${svnrepo.url}/logging/site/trunk/docs/log4j/1.2"/>

<available property="svn-available" file="target/site-deploy/.svn"/>

<!-- Construct compile classpath -->

<path id="compile.classpath">

<pathelement location="${build.home}/classes"/>

<pathelement location="${javamail.jar}"/>

<pathelement location="${activation.jar}"/>

<pathelement location="${jaxp.jaxp.jar}"/>

<pathelement location="${jms.jar}"/>

<pathelement location="${jmx.jar}"/>

<pathelement location="${jndi.jar}"/>

<pathelement location="${javaee-api.jar}"/>

</path>

<!-- ================================================================= -->

<!-- Default target -->

<!-- ================================================================= -->

<target name="usage">

<echo>

These are the targets supported by this ANT build scpript:

build - compile all project files, if a certain library is missing,

then the compilation of its dependents are skipped.

javadoc - build project javadoc files

jar - build log4j-core and log4j jar files

dist - will create a complete distribution in dist/

Setting the env variable NO\_JAVADOC will build the distribution

without running the javadoc target.

release - will create a complete distribution in dist/

using stricter settings for public distribution.

</echo>

</target>

<target name="jaxpCheck">

<available classname="javax.xml.parsers.DocumentBuilderFactory"

property="jaxp-present"/>

</target>

<target name="jaxp" depends="jaxpCheck" if="jaxp-present">

<echo message="JAXP present."/>

</target>

<target name="jmxCheck">

<condition property="jmx-present" value="true">

<and>

<available classname="javax.management.MBeanInfo"

classpath="${jmx.jar}"/>

<available classname="com.sun.jdmk.comm.HtmlAdaptorServer">

<classpath>

<pathelement location="${jmx.jar}"/>

<pathelement location="${javaee-api.jar}"/>

</classpath>

</available>

</and>

</condition>

</target>

<target name="jmx" depends="jmxCheck" if="jmx-present">

<echo message="JMX is present."/>

</target>

<target name="jmsCheck">

<available classname="javax.jms.Message" property="jms-present">

<classpath>

<pathelement location="${jms.jar}"/>

<pathelement location="${javaee-api.jar}"/>

</classpath>

</available>

</target>

<target name="jms" depends="jmsCheck" if="jms-present">

<echo message="JMS is present."/>

</target>

<target name="jndiCheck">

<available classname="javax.naming.Context" property="jndi-present"/>

</target>

<target name="jndi" depends="jndiCheck" if="jndi-present">

<echo message="JNDI is present."/>

</target>

<target name="javamailCheck">

<available classname="javax.mail.MessageAware" property="javamail-present">

<classpath>

<pathelement location="${javamail.jar}"/>

<pathelement location="${javaee-api.jar}"/>

</classpath>

</available>

</target>

<target name="javamail" depends="javamailCheck" if="javamail-present">

<echo message="JAVAMAIL is present."/>

</target>

<!-- ================================================================= -->

<!-- Initialize variables -->

<!-- NOTE: all directories are relative to jakarta-log4j/ -->

<!-- ================================================================= -->

<target name="init">

<tstamp />

<property name="javac.includeAntRuntime" value="true"/>

<property name="javac.includeJavaRuntime" value="false"/>

<property name="javac.fork" value="false"/>

</target>

<target name="build" depends="init, build.core, build.examples, build.xml,

build.javamail, build.jms, build.jmx"/>

<target name="build.core" depends="init">

<mkdir dir="${javac.dest}" />

<javac srcdir="${java.source.dir}"

destdir="${javac.dest}"

includes="${stem}/\*\*/\*.java,

${stem}/xml/XMLLayout.java"

excludes="misc/\*, \*\*/UnitTest\*.java,

\*\*/StressCategory.java,

\*\*/doc-files/\*,

${stem}/xml/\*\*,

${stem}/test/serialization/\*\*,

${stem}/net/SMTPAppender.java,

${stem}/net/JMS\*.java,

${stem}/jmx/\*.java,

${stem}/or/jms/\*.java"

target="${javac.target}"

source="${javac.source}"

includeAntRuntime="${javac.includeAntRuntime}"

includeJavaRuntime="${javac.includeJavaRuntime}"

fork="${javac.fork}"

deprecation="${deprecation}"

debug="on">

<classpath refid="compile.classpath"/>

</javac>

<copy todir="${javac.dest}">

<fileset dir="${resource.source.dir}" includes="${stem}/lf5/\*\*/\*.properties"/>

<fileset dir="${resource.source.dir}" includes="${stem}/lf5/viewer/images/\*"/>

</copy>

</target>

<target name="build.examples" depends="build.core">

<mkdir dir="${javac.dest}" />

<javac srcdir="${basedir}"

destdir="${javac.dest}"

includes="examples/\*\*/\*.java"

excludes="misc/\*"

target="${javac.target}"

source="${javac.source}"

includeAntRuntime="${javac.includeAntRuntime}"

includeJavaRuntime="${javac.includeJavaRuntime}"

fork="${javac.fork}"

deprecation="${deprecation}"

debug="on">

<classpath>

<pathelement path="${classpath}"/>

<fileset file="${jaxp.jaxp.jar}"/>

</classpath>

</javac>

<rmic base="${javac.dest}" classname="examples.NumberCruncherServer"/>

<copy todir="${javac.dest}">

<fileset dir="." includes="examples/lf5/\*\*/\*.properties"/>

<fileset dir="." includes="examples/lf5/\*\*/\*.xml"/>

</copy>

</target>

<target name="build.xml" depends="init, jaxp" if="jaxp-present">

<javac srcdir="${java.source.dir}"

destdir="${javac.dest}"

includes="${stem}/xml/\*\*/\*.java"

excludes="${stem}/xml/examples/doc-files/\*\*.java,

${stem}/xml/Transform.java"

target="${javac.target}"

source="${javac.source}"

includeAntRuntime="${javac.includeAntRuntime}"

includeJavaRuntime="${javac.includeJavaRuntime}"

fork="${javac.fork}"

deprecation="${deprecation}"

classpath="${classpath}">

<classpath refid="compile.classpath"/>

</javac>

<copy file="${resource.source.dir}/${stem}/xml/log4j.dtd" tofile="${javac.dest}/${stem}/xml/log4j.dtd" />

</target>

<target name="build.javamail" depends="init, javamail"

if="javamail-present">

<javac srcdir="${java.source.dir}"

destdir="${javac.dest}"

includes="${stem}/net/SMTPAppender.java"

target="${javac.target}"

source="${javac.source}"

includeAntRuntime="${javac.includeAntRuntime}"

includeJavaRuntime="${javac.includeJavaRuntime}"

fork="${javac.fork}"

deprecation="${deprecation}">

<classpath refid="compile.classpath"/>

</javac>

</target>

<target name="build.jms" depends="init, jms, jndi" if="jms-present">

<javac srcdir="${java.source.dir}"

destdir="${javac.dest}"

includes="${stem}/net/JMS\*.java, ${stem}/or/jms/\*.java"

target="${javac.target}"

source="${javac.source}"

includeAntRuntime="${javac.includeAntRuntime}"

includeJavaRuntime="${javac.includeJavaRuntime}"

fork="${javac.fork}"

deprecation="${deprecation}">

<classpath refid="compile.classpath"/>

</javac>

</target>

<target name="build.jmx" depends="init, jmx, jndi" if="jmx-present">

<javac srcdir="${java.source.dir}"

destdir="${javac.dest}"

includes="${stem}/jmx/\*.java"

excludes="${stem}/jmx/T.java"

target="${javac.target}"

source="${javac.source}"

includeAntRuntime="${javac.includeAntRuntime}"

includeJavaRuntime="${javac.includeJavaRuntime}"

fork="${javac.fork}"

deprecation="${deprecation}">

<classpath refid="compile.classpath"/>

</javac>

</target>

<target name="build.nt" depends="log4j.jar" description="Build NTEventLogAppender.dll">

<mkdir dir="${dist.dir}/object"/>

<ant antfile="src/ntdll/build.xml">

<property name="target.dir" location="${dist.dir}/lib"/>

<property name="object.dir" location="${dist.dir}/object"/>

<property name="classes.dir" location="${javac.dest}" />

<property name="src.dir" location="src/ntdll" />

<property name="jni.include.dir" location="${java.home}/../include" />

</ant>

</target>

<!-- ================================================================= -->

<!-- Remove all generated (compiled) class files. -->

<!-- ================================================================= -->

<target name="clean" depends="init">

<delete dir="${dist.dir}/" />

</target>

<!-- ================================================================= -->

<!-- Runs checkstyle. Available from http://checkstyle.sf.net -->

<!-- ================================================================= -->

<target name="checkstyle" depends="init">

<taskdef resource="checkstyletask.properties"

classpath="${checkstyle.jar}"/>

<!-- by default checkstyle supports the Sun coding standard. -->

<checkstyle lcurlyMethod="nlow"

lcurlyOther="nlow"

lcurlyType="nlow"

maxMethodLen="500"

maxConstructorLen="500">

<fileset dir="src/java/org/apache/log4j/chainsaw" includes="\*\*/\*.java"/>

</checkstyle>

</target>

<!-- ================================================================= -->

<!-- Runs Chainsaw -->

<!-- ================================================================= -->

<target name="chainsaw" depends="build">

<!-- Need to fork to avoid problems -->

<java classname="org.apache.log4j.chainsaw.Main" fork="yes"

classpath="${javac.dest};${ant.home}/lib/crimson.jar">

</java>

</target>

<!-- ================================================================= -->

<!-- Remove the temporary manifest file, actual work is done in the -->

<!-- dependencies. -->

<!-- ================================================================= -->

<target name="jar" depends="log4j.jar, log4j-sources.jar">

</target>

<!-- ================================================================= -->

<!-- Create log4j.jar, excluding tests and other odds and ends. -->

<!-- ================================================================= -->

<target name="log4j.jar" depends="build">

<mkdir dir="${jar.dest}"/>

<delete>

<fileset dir="${jar.dest}">

<include name="${jar.filename}"/>

</fileset>

</delete>

<jar jarfile="${jar.dest}/${jar.filename}" basedir="${javac.dest}"

includes="${stem}/\*.class,

${stem}/xml/log4j.dtd,

${stem}/config/\*.class,

${stem}/helpers/\*.class,

${stem}/spi/\*.class,

${stem}/net/\*.class,

${stem}/jdbc/\*.class,

${stem}/varia/\*.class,

${stem}/chainsaw/\*.class,

${stem}/lf5/\*\*/\*.class,

${stem}/lf5/\*\*/\*.properties,

${stem}/lf5/\*\*/\*.gif,

${stem}/nt/\*.class,

${stem}/xml/\*.class,

${stem}/jmx/\*.class,

${stem}/or/\*.class,

${stem}/or/sax/\*.class,

${stem}/or/jms/\*.class,

${stem}/config/\*.class"

excludes="\*\*/UnitTest\*\*">

<manifest>

<section name="org/apache/log4j">

<attribute name="Implementation-Title" value="log4j"/>

<attribute name="Implementation-Version" value="${version}"/>

<attribute name="Implementation-Vendor" value='"Apache Software Foundation"'/>

</section>

</manifest>

<metainf dir="." includes="NOTICE,LICENSE"/>

</jar>

</target>

<!-- ================================================================= -->

<!-- Create log4j-sources.jar -->

<!-- ================================================================= -->

<target name="log4j-sources.jar">

<mkdir dir="${jar.dest}"/>

<delete>

<fileset dir="${jar.dest}" includes="${sources-jar.filename}"/>

</delete>

<jar jarfile="${jar.dest}/${sources-jar.filename}" basedir="${java.source.dir}"

includes="\*\*/\*.java">

<manifest>

<section name="org/apache/log4j">

<attribute name="Implementation-Title" value="log4j"/>

<attribute name="Implementation-Version" value="${version}"/>

<attribute name="Implementation-Vendor" value='"Apache Software Foundation"'/>

</section>

</manifest>

<metainf dir="." includes="NOTICE,LICENSE"/>

</jar>

</target>

<!-- ================================================================= -->

<!-- This target builds the javadoc files. -->

<!-- ================================================================= -->

<target name="javadoc" depends="init" unless="env.NO\_JAVADOC">

<mkdir dir="${javadoc.dest}" />

<javadoc sourcepath="${java.source.dir}"

destdir="${javadoc.dest}"

packagenames="org.apache.log4j,

org.apache.log4j.config,

org.apache.log4j.helpers,

org.apache.log4j.jmx,

org.apache.log4j.lf5,

org.apache.log4j.net,

org.apache.log4j.nt,

org.apache.log4j.or,

org.apache.log4j.jdbc,

org.apache.log4j.or.sax,

org.apache.log4j.or.jms,

org.apache.log4j.performance,

org.apache.log4j.spi,

org.apache.log4j.varia,

org.apache.log4j.chainsaw,

org.apache.log4j.xml,

org.apache.log4j.xml.examples"

version="true"

protected="true"

author="true"

use="true"

overview="${docs.dest}/overview.html"

doctitle="log4j version ${version}&lt;br&gt;API Specification"

windowtitle="Log4j Version ${version}"

header='&lt;b&gt;Log4j ${version}&lt;/b&gt;&lt;!-- Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the &apos;License&apos;); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an &apos;AS IS&apos; BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

--&gt;'

bottom="Copyright 2000-2007 Apache Software Foundation.">

<link href="http://java.sun.com/j2se/1.3/docs/api/"/>

<link href="http://java.sun.com/j2ee/sdk\_1.3/techdocs/api/"/>

<classpath refid="compile.classpath"/>

</javadoc>

<!-- Extra files referenced by JavaDocs -->

<copy todir="${javadoc.dest}">

<fileset dir="${javadoc.source.dir}"/>

</copy>

</target>

<!-- ================================================================= -->

<!-- Compare the current API with a previous release -->

<!-- ================================================================= -->

<target name="jdiff" description="Generate comparison to reference API">

<property name="reference-api.version" value="1.2.9"/>

<property name="reference-api.dir" location="${user.home}/logging-log4j-${reference-api.version}"/>

<property name="reference-api.source.dir" location="${user.home}/logging-log4j-${reference-api.version}/src/java"/>

<!-- Must be set to the root of where JDiff is installed. -->

<property name="JDIFF\_HOME" value="${user.home}/jdiff"/>

<!-- Just one simple way to tell Ant about the JDiff task -->

<taskdef name="jdiff" classname="jdiff.JDiffAntTask"

classpath="${JDIFF\_HOME}/lib/antjdiff.jar"/>

<jdiff verbose="on" destdir="build/jdiff">

<old name="Version ${reference-api.version}">

<dirset dir="${reference-api.source.dir}" includes="org/\*\*"/>

</old>

<new name="Version ${version}">

<dirset dir="${java.source.dir}" includes="org/\*\*" excludes="org/apache/log4j/test/\*\*"/>

</new>

</jdiff>

</target>

<!-- ================================================================= -->

<!-- Compare the current API with a previous release -->

<!-- ================================================================= -->

<target name="clirr" depends="jar" description="Check binary compatibility with arbitrary release">

<property name="clirr.version" value="0.6"/>

<property name="clirr-core.jar" value="${m2\_repo}/net/sf/clirr/clirr-core/${clirr.version}/clirr-core-${clirr.version}.jar"/>

<available property="clirr-core.jar-available" file="${clirr-core.jar}"/>

<fail unless="clirr-core.jar-available">clirr-core-${clirr.version}.jar not in maven repo. Run mvn clirr:check to download.</fail>

<property name="bcel.version" value="5.1"/>

<property name="bcel.jar" value="${m2\_repo}/bcel/bcel/${bcel.version}/bcel-${bcel.version}.jar"/>

<available property="bcel.jar-available" file="${bcel.jar}"/>

<fail unless="bcel.jar-available">bcel-${bcel.version}.jar not in maven repo. Run mvn clirr:check to download.</fail>

<property name="reference.version" value="1.2.14"/>

<property name="reference.jar" value="${m2\_repo}/log4j/log4j/${reference.version}/log4j-${reference.version}.jar"/>

<available property="reference.jar-available" file="${reference.jar}"/>

<fail unless="reference.jar-available">log4j-${reference.version}.jar not in maven repo. Run mvn clirr:check to download.</fail>

<taskdef resource="clirrtask.properties"

classpath="${clirr-core.jar};${bcel.jar}"/>

<delete dir="target/clirr"/>

<mkdir dir="target/clirr/orig"/>

<mkdir dir="target/clirr/new"/>

<copy todir="target/clirr/orig">

<fileset file="${jms.jar}"/>

<fileset file="${jmx.jar}"/>

<fileset file="${reference.jar}"/>

</copy>

<copy todir="target/clirr/new">

<fileset file="${jms.jar}"/>

<fileset file="${jmx.jar}"/>

<fileset dir="${jar.dest}" includes="${jar.filename}"/>

</copy>

<clirr failOnBinError="no" failOnSrcError="no">

<origfiles dir="target/clirr/orig" includes="\*.jar"/>

<newfiles dir="target/clirr/new" includes="\*.jar"/>

<formatter type="xml" outfile="target/clirr/compatibility.xml"/>

</clirr>

</target>

<!-- ============================================== -->

<!-- Build the site files using Maven -->

<!-- ============================================== -->

<target name="site">

<exec executable="mvn">

<arg value="site"/>

</exec>

</target>

<!-- ================================================================= -->

<!-- Build a complete distribution. Results go to ${dist.images} -->

<!-- ================================================================= -->

<target name="dist" depends="init, clean, javadoc, jar, site, build.nt">

<delete verbose="true">

<fileset dir=".">

<patternset>

<include name="\*\*/\*.bak"/>

<include name="${dist.dir}/velocity.log"/>

<include name="${BSTEM}/\*\*/temp\*"/>

<include name="${BSTEM}/performance/test"/>

<include name="${BSTEM}/test/current.\*"/>

<include name="${BSTEM}/test/current.\*"/>

<include name="${BSTEM}/examples/test"/>

<include name="${BSTEM}/test/logging.\*"/>

<include name="${BSTEM}/test/log4j.properties"/>

<include name="{$BSTEM}/test/socket.lcf"/>

<include name="${BSTEM}/test/file"/>

<include name="${BSTEM}/test/output.\*"/>

<include name="${BSTEM}/net/test/loop.log"/>

<include name="${BSTEM}/net/test/loop.log.1"/>

<include name="${jar.dest}/manifest.mf"/>

<include name="${javac.dest}/\*.class"/>

</patternset>

</fileset>

</delete>

<mkdir dir="${dist.images}" />

<mkdir dir="${dist.tmp}/apache-log4j-${version}" />

<copy todir="${dist.tmp}/apache-log4j-${version}">

<fileset dir="${base}"

includes="src/main/java/\*\*,

docs/\*\*,

examples/\*\*,

build/\*,

build.xml,

build.properties.sample,

manifest.mf,

INSTALL,

LICENSE,

NOTICE,

${dist.dir}/lib/log4j-${version}.jar,

${dist.dir}/lib/NTEventLogAppender.dll,

${dist.dir}/classes/\*\*,

contribs/\*\*"

excludes="make/make.loc,

\*\*/\*.bak, \*\*/goEnv.bat,

\*\*/Makefile, \*\*/goEnv.bat,

docs/pub-support/\*,

${dist.dir}/classes/org/\*\*,

src/main/java/org/apache/log4j/test/\*\*/\*,

\*\*/.#\*,

\*\*/\*.o, \*\*/\*.res, \*\*/\*.h, \*\*/EventLogCategories.rc"/>

</copy>

<fixcrlf srcdir="${dist.tmp}/apache-log4j-${version}"

includes="build.sh" cr="remove"/>

<fixcrlf srcdir="${dist.tmp}/apache-log4j-${version}"

includes="build.bat" cr="add"/>

<chmod dir="${dist.tmp}/apache-log4j-${version}"

includes="build.sh" perm="+x"/>

<tar tarfile="${dist.images}/apache-log4j-${version}.tar.gz"

basedir="${dist.tmp}"

includes="apache-log4j-${version}/\*\*"

compression="gzip" />

<zip zipfile="${dist.images}/apache-log4j-${version}.zip"

basedir="${dist.tmp}"

includes="apache-log4j-${version}/\*\*" />

<delete dir="${dist.tmp}" />

</target>

<!-- ================================================================= -->

<!-- Build a complete distribution for release. -->

<!-- ================================================================= -->

<target name="release">

<property name="javac.includeAntRuntime" value="false"/>

<property name="javac.includeJavaRuntime" value="false"/>

<property name="javac.fork" value="true"/>

<property name="build.compiler" value="classic"/>

<antcall target="clean-site"/>

<antcall target="dist"/>

<!-- check that jar file is not missing

classes due to missing dependencies -->

<available property="jaxp-found"

classname="org.apache.log4j.xml.DOMConfigurator"

classpath="${jar.dest}/${jar.filename}"/>

<fail unless="jaxp-found" message="JAXP was not found."/>

<available property="jmx-found"

classname="org.apache.log4j.jmx.Agent"

classpath="${jar.dest}/${jar.filename}"/>

<fail unless="jmx-found" message="JMX was not found."/>

<available property="jms-found"

classname="org.apache.log4j.net.JMSAppender"

classpath="${jar.dest}/${jar.filename}"/>

<fail unless="jms-found" message="JMS or JNDI was not found."/>

<available property="javamail-found"

classname="org.apache.log4j.net.SMTPAppender"

classpath="${jar.dest}/${jar.filename}"/>

<fail unless="javamail-found" message="JAVAMAIL was not found."/>

<checksum forceOverwrite="yes" fileext=".md5">

<fileset dir="${dist.images}" includes="apache-log4j-${version}.tar.gz apache-log4j-${version}.zip"/>

</checksum>

<checksum forceOverwrite="yes" fileext=".sha1" algorithm="SHA">

<fileset dir="${dist.images}" includes="apache-log4j-${version}.tar.gz apache-log4j-${version}.zip"/>

</checksum>

<exec executable="gpg" dir="${dist.images}">

<arg value="--armor"/>

<arg value="--output"/>

<arg value="apache-log4j-${version}.tar.gz.asc"/>

<arg value="--detach-sig"/>

<arg value="apache-log4j-${version}.tar.gz"/>

</exec>

<exec executable="gpg" dir="${dist.images}">

<arg value="--armor"/>

<arg value="--output"/>

<arg value="apache-log4j-${version}.zip.asc"/>

<arg value="--detach-sig"/>

<arg value="apache-log4j-${version}.zip"/>

</exec>

</target>

<target name="fixcrlf" depends="init"

description="Fixes CR-LF and tabs in source files">

<fixcrlf srcDir="."

includes="\*\*/\*.java \*\*/build.xml"

tab="remove"

tablength="3"/>

</target>

<target name="checkout-site" unless="svn-available">

<delete dir="target/site-deploy"/>

<exec executable="svn">

<arg value="co"/>

<arg value="${svnsite.url}"/>

<arg value="target/site-deploy"/>

</exec>

</target>

<target name="update-site" if="svn-available">

<exec executable="svn" dir="target/site-deploy" failonerror="true">

<arg value="update"/>

</exec>

</target>

<target name="post-site" depends="checkout-site, update-site"/>

<target name="mime=html">

<exec executable="svn">

<arg value="propset"/>

<arg value="svn:mime-type"/>

<arg value="text/html"/>

<arg value="${src.html}"/>

</exec>

</target>

<target name="mime=css">

<exec executable="svn">

<arg value="propset"/>

<arg value="svn:mime-type"/>

<arg value="text/css"/>

<arg value="${src.css}"/>

</exec>

</target>

<target name="site-deploy">

<!-- Add any new files (and generate innocuous warnings for the existing content) -->

<delete file="target/site-deploy/svn-commit.tmp~"/>

<exec executable="bash" dir="target/site-deploy" failonerror="true">

<arg line='-c "svn add --force \*"'/>

</exec>

<taskdef name="foreach" classname="net.sf.antcontrib.logic.ForEach" />

<foreach target="mime=html" param="src.html">

<path>

<fileset dir="target/site-deploy" includes="\*\*/\*.html"/>

</path>

</foreach>

<foreach target="mime=css" param="src.css">

<path>

<fileset dir="target/site-deploy" includes="\*\*/\*.css"/>

</path>

</foreach>

<!-- requires that SVN\_EDITOR, VISUAL or EDITOR being set to edit commit description -->

<exec executable="svn" dir="target/site-deploy" failonerror="true">

<arg value="commit"/>

</exec>

</target>

</project>