



Metadata Extraction Tool

Installation Guide

Version: 3.5.

Metadata Extraction Tool

Installation Guide

Version: 3.5.

Table of Contents

✓	What is the Metadata Extraction Tool	3
✓	Source Directory Structure	4
✓	Binary Directory Structure	4
✓	Installing from Source	5
✓	Installing from Binary	5
✓	Configuring your Environment	5
✓	Running the Tool	6
✓	Troubleshooting.....	7

✓ ***What is the Metadata Extraction Tool***

The metadata extraction tool is a tool built by Sytec Resources for the *National Library of New Zealand Te Puna Mātauranga o Aotearoa (National Library)* to process digital master files and extract metadata about those files. Metadata is descriptive information about an object – not the object itself. In this case metadata information about a Image would be things like size, colours, resolution, etc... There are two output formats that an output metadata file can take

1. Native form. An XML (Extensible Mark-up Language) file based on a DTD (Document Type Definition) that reflects all information available from the digital master.
2. National Library Preservation Metadata Data Dictionary – nlnz-presmet.xsd. This is the primary format.

For more information about any of these file formats see the Solution Architecture or Software Architecture documents for this extraction tool. Note: The Proof of Concept output types of demta.dtd and pmeta.dtd have been deprecated; they are not supported in the production tool.

✓ **Source Directory Structure**

The source directory structure is as follows:

Directory	Description
BASE	Contains the build.xml and license files. Is the destination for the ZIP files for the distributables.
BASE/dist	The build directory for assembling the distributables. This directory will be deleted and recreated when running the build script.
BASE/docs	Contains the main documentation for the application.
BASE/docs/apidocs	Contains the javadocs for the application.
BASE/legal	Contains the license and notice files for all the libraries distributed with the Metadata Metadata Extraction Tool.
BASE/lib	Contains the libraries that the Metadata Metadata Extraction Tool is dependent on.
BASE/src	Root directory for all source elements.
BASE/src/java	Root directory for the Java sourcecode.
BASE/src/help	Root directory for the online HTML help guide.
BASE/src/images	Root directory for the images used in the GUI.
BASE/src/java	Root directory for the Java sourcecode.
BASE/src/scripts	Root directory for the batch files and shell scripts used to run the application.
BASE/src/xml	Root directory for the XML configuration files, DTDs and XSLT files.
BASE/target	The destination directory for the <i>javac</i> ANT task. This directory will be deleted and recreated each time the ANT script is run.

✓ **Binary Directory Structure**

The binary distributable directories are described below:

Directory	Description
BASE	Holds the configuration file, Metadata Extraction Tool license file and batch/shell scripts.
BASE/adapters	Holds the full set of adapters.
BASE/help_files	Contains the online help files.
BASE/installed-adapters	Contains the JAR files for all of the installed adapters.
BASE/legal	Contains the license and notice files for all the libraries distributed with the Metadata Extractor.
BASE/lib	Contains the libraries that the Metadata Extractor is dependent on.
BASE/ xml	Root directory for the XML configuration files, DTDs and XSLT files.

✓ *Installing from Source*

The Metadata Extraction Tool is built from source using ANT. ANT can be downloaded from <http://ant.apache.org/>. The build file has been tested against version 1.6.1.

With ANT in the classpath, change into the root directory of the Metadata Extraction Tool and run *ant*. The default target will clean the directories, compile the code, and produce the binary and source distributables.

To regenerate the JavaDocs, run *ant javadoc*.

When ANT has finished, a binary distributable will be found in

BASE\metadata-bin-2-0.zip

Once a binary distributable version is built, you can install from binary as described below.

✓ *Installing from Binary*

Unzip the ZIP file to a desired location. It is strongly recommended to choose a directory name that does not contain spaces.

✓ *Configuring your Environment*

Configuration of the tool is automatic assuming the following:

- For Windows, Java is in the path.
- For Linux, the JAVA_HOME environment variable is set.

If these are true, or if you are unsure, just run the *metadata.bat* or *metadata.sh* scripts.

The Windows scripts assume that Java is in your path and can be found without specifying its exact location. If the *metadata.bat* or *extract.bat* scripts fail to run, you may need to edit them and provide an explicit path for your Java installation.

The Linux scripts require the JAVA_HOME variable to be set. If it is not set, you can add it into the *metadata.sh* and *extract.sh* scripts as follows, being sure to replace the path with the appropriate path for your Java installation.

```
JAVA_HOME=/usr/java/jdk1.5.0
export JAVA_HOME
```

Both versions attempt to guess the installation directory and will attempt to configure the initial *config.xml* file without manual intervention. If this fails, you will be asked to edit the scripts and set the METAHOME variable.

⇒ The version 3.5 of Metadata extractor has been compiled using Java 6. Hence if you are using the binary installation directly without building it from the source code, you may have to have Java 6 installed on your computer and your JAVA_HOME pointing to your jdk1.6 installed folder. However the code is not making use of any new features of Java 6. Hence if you intend to build from source code you may build it using the jdk1.4 version onwards.

✓ ***Running the Tool***

To run the tool, change into the BASE directory and run ***metadata.bat*** or ***metadata.sh*** to run the Metadata Extraction Tool.

If you wish to run the tool manually, or embed it in another application, you must set the classpath to contain:

1. All JARs in the BASE/lib directory.
2. The BASE directory itself – this is where the config.xml file is located.

Once the classpath is configured correctly, you can run the tool using:

\$JAVA_HOME/bin/java nz.govt.natlib.meta.ui.Main (Linux)

%JAVA_HOME%\bin\java nz.govt.natlib.meta.ui.Main (Windows)

✓ **Troubleshooting**

The following table lists a set of commonly encountered issues and the required resolution. The most common issues are around the directory locations specified in the configuration file. Following the instructions in the **Configuration** section above should avoid any of those issues.

If you get an error during harvesting, you will need to use the **Log Viewer** to get additional information about the error, or consult the Output.log file.

Symptom	Description	Solution
On startup, you see the message: The system cannot find the path specified. or -bash: java: command not found	The JAVA_HOME variable is not set correctly.	Set the JAVA_HOME variable as per the instructions in the Setting JAVA_HOME Environment Variable section of this document.
On startup, you see errors such as: LOG:1000, Adapter class nz.govt.natlib.adapter.bmp.BitmapAdapter not found java.lang.ClassNotFoundException: nz.govt.natlib.adapter.bmp.BitmapAdapter	The <i>jar-location</i> in the config.xml file is not set correctly.	Edit <i>config.xml</i> in the base directory and ensure that the <i>jar-location</i> URL attribute is pointing at a valid directory.
On startup, you see an error such as: java.io.FileNotFoundException: METADATA_BASE\logs\nlnz_Jan302007_171007.log	The log-dir element in <i>config.xml</i> points to a directory that does not exist.	Edit <i>config.xml</i> in the base directory and ensure that the <i>dir</i> attribute of the <i>log-dir</i> element is set to an existing directory. Note that there are two occurrences of the <i>log-dir</i> element in the default configuration. One at the top of the configuration

		file, and one in the <i>profile</i> section towards the bottom.
<p>During harvest, you see an error such as:</p> <pre>java.io.FileNotFoundException: METADATA_BASE\harvested\...\filename.xml at java.io.FileOutputStream.open(Native Method) at java.io.FileOutputStream.<init>(FileOutputStream.java:179) at java.io.FileOutputStream.<init>(FileOutputStream.java:131) at nz.govt.natlib.meta.harvester.SimpleObjectHarvester.startHarvestFile(SimpleObjectHarvester.java:89) at ...</pre>	<p>The harvest directory does not exist. If you have modified the default location, you can tell that this is the problem by either the fact that it is trying to find an XML file, or from the stack trace.</p>	<p>Edit the <i>config.xml</i> file in the base directory and check the <i>output-directory</i> element in the <i>configurations</i> section of the config file.</p> <p>Note that this property is part of each configuration.</p>
<p>You get the following error trying to harvest a file.</p> <pre>ERROR: 'C:\METADATA_BASE\xml\bmp_to_nlnz_presmet.xslt (The system cannot find the path specified)' FATAL ERROR: 'Could not compile stylesheet'</pre>	<p>The XSLT files cannot be found.</p>	<p>Check the <i>config.xml</i> file to ensure that the <i>xml-location</i> element is correctly set. By default, the XML/XSLT/DTD files exist under BASE/xml.</p>