

# DSpace SIP Toolkit

This page describes the DSpace SIP toolkit, a class that makes it easier to write an application to produce valid Submission Information Packages (SIPs).

**UNSUPPORTED:** This is *unsupported* code. Use it at your own risk. It is simple enough that reading the code and examples should be enough to answer any questions you may have, but if not, there may not be any other help available. It is provided freely to the DSpace community but there are no support resources available.

## What it Does

In order to use the [package](#)-based ingester interfaces such as the LNI, you have to construct a *Submission Information Package* (SIP) out of your digital content and metadata. The default type of package, a DSpace SIP, is a Zip archive that includes:

1. a mandatory METS document which acts as the manifest
2. `_content files _` (the digital objects)
3. any metadata files referenced by the METS document

This toolkit helps an application construct a valid METS-based package and either write it to a file or transmit it on a stream. It takes care of most of the housekeeping and METS structure, which lets you produce a SIP-writing application much more quickly and easily.

You still have to write an application around it, but now you can produce a SIP by simply providing some descriptive metadata and content files. Almost all of the details of Zip and METS structure are hidden.

## Example Code

The best way to illustrate the toolkit is through an example. This code fragment constructs a SIP and writes it to a file:

```
import edu.mit.libraries.facade.app.DSpaceSIP;

// Create SIP; validate = false, ZIP compression = BEST_SPEED
DSpaceSIP sip = new DSpaceSIP(false, Deflater.BEST_SPEED);

// Optional: Set the METS OBJID
sip.setOBJID(myObjID);

// Optional: Set the METS creator
sip.addAgent("CREATOR", "ORGANIZATION", "MyUniversity Libraries");

// add content objects - last arg is "is Primary Bitstream"
sip.addBitstream(new File("thesis102.pdf"), "content/thesis.pdf", "ORIGINAL", true);
sip.addBitstream(new File("thesis102.doc"), "content/thesis.doc", "ORIGINAL", false);

// add the descriptive metadata as JDOM Element; the package also
// accepts a String of serialized XML or a file of any format.
Element modsElt = myMakeMetadata();
sip.addDescriptiveMD("MODS", modsElt);

// Write SIP to a file
File outfile = File.createTempFile(myObjID, ".zip", targetDir);
sip.write(outfile);
```

See the source code, which has Javadoc-style comments, for more complete documentation. You can also use `javadoc` to create web pages of API documentation.

# Downloading and Running DSpaceSIP

- Note this class and its dependencies have been defined in a modules in the DSpace SVN repository located here [[--Mark Diggory](#)] 16:40, 5 June 2009 (EDT)
- [DSpaceSIP.java](#)

To use this class in your application, you will also need the following Java libraries on your classpath:

1. The [Harvard METS Java toolkit](#), version 1.5
2. [JDOM 1.0](#) (simplified XML representation in Java)

NOTE: These libraries are also required by DSpace 1.4 and 1.5, so they should be available in the `lib` subdirectory of your DSpace runtime installation.

## Future Work

As mentioned, this is *completely unsupported* code. You are encouraged to use it, improve upon it, and use the Wiki to document and share your improvements.

If you find this documentation lacking, please add the missing answers to it.