SWORDv2 Server

SWORD (Simple Web-service Offering Repository Deposit) is a protocol that allows the remote deposit of items into repositories. DSpace implements the SWORD protocol via the 'sword' web application. The specification and further information can be found at http://swordapp.org/.

SWORD is based on the Atom Publish Protocol and allows service documents to be requested which describe the structure of the repository, and packages to be deposited.

Enabling SWORD v2 Server

To enable DSpace's SWORD v2 server, just make sure the `[dspace]/webapps/swordv2/` web application is available from your Servlet Container (usually Tomcat).

Configuring SWORD v2 Server

Configuration File:	`[dspace]/config/modules/swordv2-server.cfg`
Property:	`url`
Example Value:	`url = http://www.myu.ac.uk /swordv2`
Informational Note:	The base url of the SWORD 2.0 system. This defaults to `${dspace.baseUrl}/swordv2` (where `dspace.baseUrl` is defined in your `dspace.cfg` file).
Property:	`collection.url`
Example Value:	`collection.url = http://www. myu.ac.uk/swordv2/collection`
Informational Note:	The base URL of the SWORD collection. This is the URL from which DSpace will construct the deposit location URLs for collections. This defaults to `${dspace.baseUrl}/swordv2/collection` (where `dspace.baseUrl` is defined in your `dspace.cfg` file).
Property:	`servicedocument.url`
Example Value:	`servicedocument.url = http://www.myu.ac.uk/swordv2 /servicedocument`
Informational Note:	The service document URL of the SWORD collection. The base URL of the SWORD service document. This is the URL from which DSpace will construct the service document location urls for the site, and for individual collections. This defaults to `${dspace.baseUrl}/swordv2 /servicedocument` (where `dspace.baseUrl` is defined in your `dspace.cfg` file).
Property:	`accept-packaging.collection`

| Example Value: | ```
accept-packaging.collection.
METSDSpaceSIP = http://purl.
org/net/sword/package
/METSDSpaceSIP
accept-packaging.collection.
SimpleZip = http://purl.org
/net/sword/package/SimpleZip
accept-packaging.collection.
Binary = http://purl.org/net
/sword/package/Binary
``` |
|---|---|
| Informational Note: | The accept packaging properties, along with their associated quality values where appropriate. |
| **Property:** | `accept-packaging.item` |
| Example Value: | ```
accept-packaging.item.
METSDSpaceSIP = http://purl.
org/net/sword/package
/METSDSpaceSIP
accept-packaging.item.
SimpleZip = http://purl.org
/net/sword/package/SimpleZip
accept-packaging.item.Binary
= http://purl.org/net/sword
/package/Binary
``` |
| Informational Note: | The accept packaging properties for items.   It is possible to configure this for specific collections by adding the handle of the collection to the setting, for example `accept-packaging.collection.[handle].` `METSDSpaceSIP =` http://purl.org/net/sword-types /METSDSpaceSIP |
| **Property:** | `accepts` |
| Example Value: | ```
accepts = application/zip,
image/jpeg
``` |
| Informational Note: | A comma-separated list of MIME types that SWORD will accept. |
| **Property:** | `expose-communities` |
| Example Value: | ```
expose-communities = false
``` |
| Informational Note: | Whether or not the server should expose a list of all the communities to a service document request.   As deposits can only be made into a collection, it is recommended to leave this set to false. |
| **Property:** | `max-upload-size` |

Example Value:	max-upload-size = 0
Informational Note:	The maximum upload size of a package through the SWORD interface (measured in bytes). This will be the combined size of all the files, metadata, and manifest file in a package - this is different to the maximum size of a single bitstream. If this is set to 0, no maximum file size will be enforced.
Property:	keep-original-package
Example Value:	keep-original-package = true
Informational Note:	Should DSpace store a copy of the orignal SWORD deposit package? This will cause the deposit process to be slightly slower and for more disk to be used, however original files will be preserved. It is recommended to leave this option enabled.
Property:	bundle-name
Example Value:	bundle-name = SWORD
Informational Note:	The bundle name that SWORD should store incoming packages within if keep-original-package is set to true.
Property:	keep-package-on-fail
Example Value:	keep-package-on-fail = false
Informational Note:	In the event of package ingest failure, provide an option to store the package on the file system. The default is false. The location can be set using the failed-package-dir setting.
Property:	failed-package-dir
Example Value:	failed-package-dir = /dspace /upload
Informational Note:	If keep-package-on-fail is set to true, this is the location where the package would be stored.
Property:	on-behalf-of.enable
Example Value:	on-behalf-of.enable = true
Informational Note:	Should DSpace accept mediated deposits? See the SWORD specification for a detailed explanation of deposit On-Behalf-Of another user.
Property:	generator.url

Example Value:	generator.url = http://www. dspace.org/ns/sword/2.0/
Informational Note:	The URL which identifies DSpace as the software that is providing the SWORD interface.
Property:	generator.version
Example Value:	generator.version = 2.0
Informational Note:	The version of the SWORD interface.
Property:	auth-type
Example Value:	auth-type = Basic
Informational Note:	Which form of authentication to use. Normally this is set to Basic in order to use HTTP Basic.
Property:	upload.tempdir
Example Value:	upload.tempd = /dspace/upload
Informational Note:	The location where uploaded files and packages are stored while being processed.
Property:	updated.field
Example Value:	updated.field = dc.date. updated
Informational Note:	The metadata field in which to store the updated date for items deposited via SWORD.
Property:	slug.field
Example Value:	slug.field = dc.identifier. slug
Informational Note:	The metadata field in which to store the value of the slug header if it is supplied.
Property:	author.field
Example Value:	author.field = dc.contributor. author
Informational Note:	The metadata field in which to store the value of the atom entry author if it supplied.

Property:	title.field

Example Value:	dc.title

Informational Note:	The metadata field in which to store the value of the atom entry title if it supplied.

Property:	disseminate-packaging

Example Value:	disseminate-packaging. METSDSpaceSIP = http://purl. org/net/sword/package /METSDSpaceSIP disseminate-packaging. SimpleZip = http://purl.org /net/sword/package/SimpleZip

Informational Note:	Supported packaging formats for the dissemination of packages.

Property:	plugin.single.org.dspace.sword2.WorkflowManager

Example Value:	plugin.single.org.dspace. sword2.WorkflowManager = org. dspace.sword2. WorkflowManagerDefault

Informational Note:	Which workflow manager to use.

Property:	mets-ingester.package-ingester

Example Value:	mets-ingester.package- ingester = METS

Informational Note:	Which package ingester to use for METS packages.

Property:	restore-mode.enable

Example Value:	restore-mode.enable = false

Informational Note:	Should the SWORD server enable restore-mode when ingesting new packages. If this is enabled the item will be treated as a previously deleted item from the repository. If the item has previously been assigned a handle then that same handle will be restored to activity.

Property:	simpledc.*

| Example Value: | ```
simpledc.abstract = dc.
description.abstractsimpledc.
date = dc.datesimpledc.rights
= dc.rights
``` |
| Informational Note: | Configuration of metadata field mapping used by the SimpleDCEntryIngester. |
| **Property:** | `multipart.entry-first` |
| Example Value: | ```
multipart.entry-first = false
``` |
| Informational Note: | The order of precedence for importing multipart content. If this is set to `true` then metadata in the package will override metadata in the atom entry, otherwise the metadata in the atom entry will override that from the package. |
| **Property:** | `workflow.notify` |
| Example Value: | ```
workflow.notify = true
``` |
| Informational Note: | If the workflow gets started (the collection being deposited into has a workflow configured), should a notification get sent? |
| **Property:** | `versions.keep` |
| Example Value: | ```
versions.keep = true
``` |
| Informational Note: | When content is replaced, should the old version be kept? This creates a copy of the ORIGINAL bundle with the name V_YYYY-MM-DD.X where YYYY-MM-DD is the date the copy was created, and X is an integer from 0 upwards. |
| **Property:** | `state.*` |
| Example Value: | ```
state.workspace.uri =
http://localhost:8080/xmlui
/state/inprogress
state.workspace.description =
The item is in the user
workspace
state.workflow.uri =
http://localhost:8080/xmlui
/state/inreview
state.workflow.description =
The item is undergoing review
prior to acceptance in the
archive
``` |
| Informational Note: | Pairs of states (URI and description) than items can be in. Typical states are `workspace`, `workflow`, `archive`, and `withdrawn`. |

Other configuration options exist that define the mapping between mime types, ingesters, and disseminators. A typical configuration looks like this:

```
plugin.named.org.dspace.sword2.SwordContentIngester = \
  org.dspace.sword2.SimpleZipContentIngester = http://purl.org/net/sword
/package/SimpleZip, \
  org.dspace.sword2.SwordMETSIngester = http://purl.org/net/sword/package
/METSDSpaceSIP, \
  org.dspace.sword2.BinaryContentIngester = http://purl.org/net/sword
/package/Binary, \
  org.dspace.swordpackagers.SwordDocXIngester = application/vnd.
openxmlformats-officedocument.wordprocessingml.document, \
  org.dspace.swordpackagers.SwordXifIngester = image/jpeg

plugin.single.org.dspace.sword2.SwordEntryIngester = \
  org.dspace.sword2.SimpleDCEntryIngester

plugin.single.org.dspace.sword2.SwordEntryDisseminator = \
  org.dspace.sword2.SimpleDCEntryDisseminator

# note that we replace ";" with "_" as ";" is not permitted in the
PluginManager names
plugin.named.org.dspace.sword2.SwordContentDisseminator = \
  org.dspace.sword2.SimpleZipContentDisseminator = http://purl.org/net
/sword/package/SimpleZip, \
  org.dspace.sword2.FeedContentDisseminator = application/atom+xml, \
  org.dspace.sword2.FeedContentDisseminator = application
/atom+xml_type_feed

# note that we replace ";" with "_" as ";" is not permitted in the
PluginManager names
plugin.named.org.dspace.sword2.SwordStatementDisseminator = \
  org.dspace.sword2.AtomStatementDisseminator = atom, \
  org.dspace.sword2.OreStatementDisseminator = rdf, \
  org.dspace.sword2.AtomStatementDisseminator = application
/atom+xml_type_feed, \
  org.dspace.sword2.OreStatementDisseminator = application/rdf+xml
```