Upgrading From 1.5 or 1.5.1 to 1.5.2

In the notes below `[dspace]` refers to the install directory for your existing DSpace installation, and *[dspace-source]* to the source directory for DSpace 1.5.1. Whenever you see these path references, be sure to replace them with the actual path names on your local system.

Upgrade Steps

The changes in DSpace 1.5.2 do not include any database schema upgrades, and the upgrade should be straightforward.

1. **Backup your DSpace** First and foremost, make a complete backup of your system, including:
   - A snapshot of the database
   - The asset store (*[dspace]/assetstore* by default)
   - Your configuration files and customizations to DSpace
   - Your statistics scripts (*[dspace]/bin/stat**) which contain customizable dates
2. **Download DSpace 1.5.2** Get the new DSpace 1.5.2 source code either as a download from DSpace.org or check it out directly from the SVN code repository. If you downloaded DSpace do not unpack it on top of your existing installation.
3. **Build DSpace** Run the following commands to compile DSpace.

   ```
   cd [dspace-source]/dspace/
     mvn package
   ```

   You will find the result in *[dspace-source]/dspace/target/dspace-1.5.2-build.dir/*; inside this directory is the compiled binary distribution of DSpace.
4. **Stop Tomcat** Take down your servlet container, for Tomcat use the *bin/shutdown.sh* script.
5. **Apply any customizations** If you have made any local customizations to your DSpace installation they will need to be migrated over to the new DSpace. Commonly these modifications are made to "JSP" pages located inside the *[dspace 1.4.2]/jsp/local* directory. These should be moved *[dspace-source]/dspace/modules/jspui/src/main/webapp/* in the new build structure. See Customizing the JSP Pages for more information.
6. **Update DSpace** Update the DSpace installed directory with new code and libraries. Inside the *[dspace-source]/dspace/target/dspace-1.5-build.dir/* directory run:

   ```
   cd [dspace-source]/dspace/target/dspace-1.5-build.dir/
   ant -Dconfig=[dspace]/config/dspace.cfg update
   ```

7. **Update configuration files** This ant target preserves existing files in *[dspace]/config _ and will copy any new configuration files in place. If an existing file prevents copying the new file in place, the new file will have the suffix _.new*, for example *[dspace]/local/dspace.cfg.new*. Note: there is also a configuration option *-Doverwrite=true* which will instead copy the conflicting target files to *.old suffixes and overwrite target file then with the new file (essentially the opposite) this is beneficial for developers and those who use the *[dspace-source]/dspace/config* to maintain their changes.

   ```
   cd [dspace-source]/dspace/target/dspace-1.5-build.dir/
   ant -Dconfig=[dspace]/config/dspace.cfg update_configs
   ```

   You must then verify that you've merged and differenced in the *[dspace]/config/*/*.new* files into your configuration. Some of the new parameters you should look out for in dspace.cfg include:
   - New option to restrict the expose of private items. The following needs to be added to dspace.cfg:

     ```
     #### Restricted item visibility settings ###
     # By default RSS feeds, OAI-PMH and subscription emails will include ALL items
     # regardless of permissions set on them.
     #
     # If you wish to only expose items through these channels where the ANONYMOUS
     # user is granted READ permission, then set the following options to false
     #harvest.includerestricted.rss = true
     #harvest.includerestricted.oai = true
     #harvest.includerestricted.subscription = true
     ```

   - Special groups for LDAP and password authentication.

```
##### Password users group #####

# If required, a group name can be given here, and all users who log in
# using the DSpace password system will automatically become members of
# this group. This is useful if you want a group made up of all internal
# authenticated users.
#password.login.specialgroup = group-name

##### LDAP users group #####

# If required, a group name can be given here, and all users who log in
# to LDAP will automatically become members of this group. This is useful
# if you want a group made up of all internal authenticated users.
#ldap.login.specialgroup = group-name
```

- new option for case insensitivity in browse tables.

```
# By default, the display of metadata in the browse indexes is case sensitive
# So, you will get separate entries for the terms
#
#   Olive oil
#   olive oil
#
# However, clicking through from either of these will result in the same set of items
# (ie. any item that contains either representation in the correct field).
#
# Uncommenting the option below will make the metadata items case-insensitive. This will
# result in a single entry in the example above. However the value displayed may be either 'Olive
oil'
# or 'olive oil' - depending on what representation was present in the first item indexed.
#
# If you care about the display of the metadata in the browse index - well, you'll have to go and
# fix the metadata in your items.
#
# webui.browse.metadata.case-insensitive = true
```

- New usage event handler for collecting statistics:

```
### Usage event settings ###
# The usage event handler to call.  The default is the "passive" handler, which ignores events.
# plugin.single.org.dspace.app.statistics.AbstractUsageEvent = \
#   org.dspace.app.statistics.PassiveUsageEvent
```

- The location where sitemaps are stored is now configurable.

```
#### Sitemap settings #####
# the directory where the generated sitemaps are stored
sitemap.dir = ${dspace.dir}/sitemaps
```

- MARC 21 ordering should now be used as default. Unless you have it set already, or you have it set to a different value, the following should be set:

```
plugin.named.org.dspace.sort.OrderFormatDelegate = org.dspace.sort.OrderFormatTitleMarc21=title
```

- Hierarchical LDAP support.

```
##### Hierarchical LDAP Settings #####
# If your users are spread out across a hierarchical tree on your
# LDAP server, you will need to use the following stackable authentication
# class:
#  plugin.sequence.org.dspace.authenticate.AuthenticationMethod = \
#        org.dspace.authenticate.LDAPHierarchicalAuthentication
#
# You can optionally specify the search scope. If anonymous access is not
# enabled on your LDAP server, you will need to specify the full DN and
# password of a user that is allowed to bind in order to search for the
# users.

# This is the search scope value for the LDAP search during
# autoregistering. This will depend on your LDAP server setup.
# This value must be one of the following integers corresponding
# to the following values:
# object scope : 0
# one level scope : 1
# subtree scope : 2
#ldap.search_scope = 2

# The full DN and password of a user allowed to connect to the LDAP server
# and search for the DN of the user trying to log in.  If these are not specified,
# the initial bind will be performed anonymously.
#ldap.search.user = cn=admin,ou=people,o=myu.edu
#ldap.search.password = password

# If your LDAP server does not hold an email address for a user, you can use
# the following field to specify your email domain. This value is appended
# to the netid in order to make an email address. E.g. a netid of 'user' and
# ldap.netid_email_domain as '@example.com' would set the email of the user
# to be 'user@example.com
#ldap.netid_email_domain = @example.com
```

- Shibboleth authentication support.

```
#### Shibboleth Authentication Configuration Settings ####
# Check https://mams.melcoe.mq.edu.au/zope/mams/pubs/Installation/dspace15/view
# for installation detail.
#
#       org.dspace.authenticate.ShibAuthentication
#
# DSpace requires email as user's credential. There are 2 ways of providing
# email to DSpace:
# 1) by explicitly specifying to the user which attribute (header)
#    carries the email address.
# 2) by turning on the user-email-using-tomcat=true which means
#    the software will try to acquire the user's email from Tomcat
# The first option takes PRECEDENCE when specified. Both options can
# be enabled to allow fallback.

# this option below specifies that the email comes from the mentioned header.
# The value is CASE-Sensitive.
authentication.shib.email-header = MAIL

# optional. Specify the header that carries user's first name
# this is going to be used for creation of new-user
authentication.shib.firstname-header = SHIB-EP-GIVENNAME

# optional. Specify the header that carries user's last name
# this is used for creation of new user
authentication.shib.lastname-header = SHIB-EP-SURNAME

# this option below forces the software to acquire the email from Tomcat.
authentication.shib.email-use-tomcat-remote-user = true

# should we allow new users to be registered automatically
# if the IdP provides sufficient info (and user not exists in DSpace)
authentication.shib.autoregister = true

# this header here specifies which attribute that is responsible
# for providing user's roles to DSpace. When not specified, it is
# defaulted to 'Shib-EP-UnscopedAffiliation'. The value is specified
# in AAP.xml (Shib 1.3.x) or attribute-filter.xml (Shib 2.x).
# The value is CASE-Sensitive. The values provided in this
# header are separated by semi-colon or comma.
# authentication.shib.role-header = Shib-EP-UnscopedAffiliation

# when user is fully authN on IdP but would not like to release
# his/her roles to DSpace (for privacy reason?), what should be
# the default roles be given to such users?
# The values are separated by semi-colon or comma
# authentication.shib.default-roles = Staff, Walk-ins

# The following mappings specify role mapping between IdP and Dspace.
# the left side of the entry is IdP's role (prefixed with
# "authentication.shib.role.") which will be mapped to
# the right entry from DSpace. DSpace's group as indicated on the
# right entry has to EXIST in DSpace, otherwise user will be identified
# as 'anonymous'. Multiple values on the right entry should be separated
# by comma. The values are CASE-Sensitive. Heuristic one-to-one mapping
# will be done when the IdP groups entry are not listed below (i.e.
# if "X" group in IdP is not specified here, then it will be mapped
# to "X" group in DSpace if it exists, otherwise it will be mapped
# to simply 'anonymous')
#
# Given sufficient demand, future release could support regex for the mapping
# special characters need to be escaped by \
authentication.shib.role.Senior\ Researcher = Researcher, Staff
authentication.shib.role.Librarian = Administrator
```

- DOI and handle identifiers can now be rendered in the JSPUI.

```
# When using "resolver" in webui.itemdisplay to render identifiers as resolvable
# links, the base URL is taken from <code>webui.resolver.<n>.baseurl</code>
# where <code>webui.resolver.<n>.baseurl</code> matches the urn specified in the metadata value.
# The value is appended to the "baseurl" as is, so the baseurl need to end with slash almost in
any case.
# If no urn is specified in the value it will be displayed as simple text.
#
#webui.resolver.1.urn = doi
#webui.resolver.1.baseurl = http://dx.doi.org/
#webui.resolver.2.urn = hdl
#webui.resolver.2.baseurl = http://hdl.handle.net/
#
# For the doi and hdl urn defaults values are provided, respectively http://dx.doi.org and
# http://hdl.handle.net are used.
#
# If a metadata value with style: "doi", "handle" or "resolver" matches a URL
# already, it is simply rendered as a link with no other manipulation.
```

In configuration sections such as webui.itemdisplay.default, values can be changed from (e.g.) metadata.dc.identifier.doi to metadata.doi.dc.identifier.doi

- The whole of the SWORD configuration has changed. The SWORD section must be removed and replaced with

```
#---------------------------------------------------------------#
#-------------SWORD SPECIFIC CONFIGURATIONS-------------------#
#---------------------------------------------------------------#
# These configs are only used by the SWORD interface            #
#---------------------------------------------------------------#

# tell the SWORD METS implementation which package ingester to use
# to install deposited content.  This should refer to one of the
# classes configured for:
#
# plugin.named.org.dspace.content.packager.PackageIngester
#
# The value of sword.mets-ingester.package-ingester tells the
# system which named plugin for this interface should be used
# to ingest SWORD METS packages
#
# The default is METS
#
# sword.mets-ingester.package-ingester = METS

# Define the metadata type EPDCX (EPrints DC XML)
# to be handled by the SWORD crosswalk configuration
#
mets.submission.crosswalk.EPDCX = SWORD

# define the stylesheet which will be used by the self-named
# XSLTIngestionCrosswalk class when asked to load the SWORD
# configuration (as specified above).  This will use the
# specified stylesheet to crosswalk the incoming SWAP metadata
# to the DIM format for ingestion
#
crosswalk.submission.SWORD.stylesheet = crosswalks/sword-swap-ingest.xsl

# The base URL of the SWORD deposit.  This is the URL from
# which DSpace will construct the deposit location urls for
# collections.
#
# The default is {dspace.url}/sword/deposit
#
# In the event that you are not deploying DSpace as the ROOT
# application in the servlet container, this will generate
# incorrect URLs, and you should override the functionality
# by specifying in full as below:
#
# sword.deposit.url = http://www.myu.ac.uk/sword/deposit
```

```
# 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
#
# The default is {dspace.url}/sword/servicedocument
#
# In the event that you are not deploying DSpace as the ROOT
# application in the servlet container, this will generate
# incorrect URLs, and you should override the functionality
# by specifying in full as below:
#
# sword.servicedocument.url = http://www.myu.ac.uk/sword/servicedocument

# The base URL of the SWORD media links.  This is the URL
# which DSpace will use to construct the media link urls
# for items which are deposited via sword
#
# The default is {dspace.url}/sword/media-link
#
# In the event that you are not deploying DSpace as the ROOT
# application in the servlet container, this will generate
# incorrect URLs, and you should override the functionality
# by specifying in full as below:
#
# sword.media-link.url = http://www.myu.ac.uk/sword/media-link

# The URL which identifies the sword software which provides
# the sword interface.  This is the URL which DSpace will use
# to fill out the atom:generator element of its atom documents.
#
# The default is:
#
# http://www.dspace.org/ns/sword/1.3.1
#
# If you have modified your sword software, you should change
# this URI to identify your own version.  If you are using the
# standard dspace-sword module you will not, in general, need
# to change this setting
#
# sword.generator.url = http://www.dspace.org/ns/sword/1.3.1

# The metadata field in which to store the updated date for
# items deposited via SWORD.
#
sword.updated.field = dc.date.updated

# The metadata field in which to store the value of the slug
# header if it is supplied
#
sword.slug.field = dc.identifier.slug

# the accept packaging properties, along with their associated
# quality values where appropriate.
#
# Global settings; these will be used on all DSpace collections
#
sword.accept-packaging.METSDSpaceSIP.identifier =
http://purl.org/net/sword-types/METSDSpaceSIP
sword.accept-packaging.METSDSpaceSIP.q = 1.0

# Collection Specific settings: these will be used on the collections
# with the given handles
#
# sword.accept-packaging.[handle].METSDSpaceSIP.identifier =
http://purl.org/net/sword-types/METSDSpaceSIP
# sword.accept-packaging.[handle].METSDSpaceSIP.q = 1.0

# Should the server offer up items in collections as sword deposit
# targets.  This will be effected by placing a URI in the collection
# description which will list all the allowed items for the depositing
# user in that collection on request
```

```
#
# NOTE: this will require an implementation of deposit onto items, which
# will not be forthcoming for a short while
#
sword.expose-items = false

# Should the server offer as the default the list of all Communities
# to a Service Document request.  If false, the server will offer
# the list of all collections, which is the default and recommended
# behavior at this stage.
#
# NOTE: a service document for Communities will not offer any viable
# deposit targets, and the client will need to request the list of
# Collections in the target before deposit can continue
#
sword.expose-communities = false

# The maximum upload size of a package through the sword interface,
# in bytes
#
# This will be the combined size of all the files, the metadata and
# any manifest data.  It is NOT the same as the maximum size set
# for an individual file upload through the user interface.  If not
# set, or set to 0, the sword service will default to no limit.
#
sword.max-upload-size = 0

# Should DSpace store a copy of the original sword deposit package?
#
# NOTE: this will cause the deposit process to run slightly slower,
# and will accelerate the rate at which the repository consumes disk
# space.  BUT, it will also mean that the deposited packages are
# recoverable in their original form.  It is strongly recommended,
# therefore, to leave this option turned on
#
# When set to "true", this requires that the configuration option
# "upload.temp.dir" above is set to a valid location
#
sword.keep-original-package = true

# The bundle name that SWORD should store incoming packages under if
# sword.keep-original-package is set to true.  The default is "SWORD"
# if not value is set
#
# sword.bundle.name = SWORD

# Should the server identify the sword version in deposit response?
#
# It is recommended to leave this enabled.
#
sword.identify-version = true

# Should we support mediated deposit via sword?  Enabled, this will
# allow users to deposit content packages on behalf of other users.
#
# See the SWORD specification for a detailed explanation of deposit
# On-Behalf-Of another user
#
sword.on-behalf-of.enable = true

# Configure the plugins to process incoming packages.  The form of this
# configuration is as per the Plugin Manager's Named Plugin documentation:
#
# plugin.named.[interface] = [implementation] = [package format identifier] \
#
# Package ingesters should implement the SWORDIngester interface, and
# will be loaded when a package of the format specified above in:
#
# sword.accept-packaging.[package format].identifier = [package format identifier]
#
# is received.
```

```
#
# In the event that this is a simple file deposit, with no package
# format, then the class named by "SimpleFileIngester" will be loaded
# and executed where appropriate.  This case will only occur when a single
# file is being deposited into an existing DSpace Item
#
plugin.named.org.dspace.sword.SWORDIngester = \
  org.dspace.sword.SWORDMETSIngester =
http://purl.org/net/sword-types/METSDSpaceSIP \
  org.dspace.sword.SimpleFileIngester = SimpleFileIngester
```

8. **Restart Tomcat** Restart your servlet container, for Tomcat use the *bin/startup.sh* script.