Item Level Versioning

1 What is Item Level Versioning?
2 Important warnings
3 Disabling Item Level Versioning
4 Initial Requirements
5 User Interface
    5.1 General behaviour: Linear Versioning
    5.2 Creating a new version of an item
    5.3 View the history and older versions of an item
6 Architecture
    6.1 Versioning model
7 Configuration
    7.1 Versioning Service Override
    7.2 Identifier Service Override
    7.3 Version History Visibility
        7.3.1 Hide Editor/Submitter details in version table
    7.4 Allowing submitters to version their items
8 Identified Challenges & Known Issues
    8.1 Conceptual compatibility with Embargo
    8.2 Conceptual compatibility with Item Level Statistics

What is Item Level Versioning?

Versioning is a new functionality to build the history of an item. Users will have the opportunity to create new version of an existing item any time the will make a change.

Supported in 7.1 or above

Item level versioning was not fully supported in DSpace 7.0 (you were only able to view existing versions). It was restored in DSpace 7.1. See DSpace Release 7.0 Status

Important warnings

Item Level Versioning on Entities configuration

Configurable Entities are supported in Item Level Versioning support starting from version 7.3. More details about the configuration specific to Configurable Entities can be found on that page.

AIP Backup & Restore functionality only works with the Latest Version of Items

If you are using the AIP Backup and Restore functionality to backup / restore / migrate DSpace Content, you must be aware that the "Item Level Versioning" feature is **not yet compatible** with AIP Backup & Restore. **Using them together may result in accidental data loss.** Currently the AIPs that DSpace generates only store the *latest version* of an Item. Therefore, past versions of Items will always be lost when you perform a restore / replace using AIP tools. See https://github.com/DSpace/DSpace/issues/4751.

DSpace 6+ changed the way Handles are created for versioned Items

Starting with 6.0, the way DSpace crates Handles for versioned Items was changed. If you want to keep the old behavior of DSpace 4 and 5 you have to enable the `VersionedHandleIdentifierProviderWithCanonicalHandles` in the XML configuration files `[dspace]/config/spring/api/identifier-service.xml`. See IdentifierServiceOverride below for details and the comments in the configuration file.

Disabling Item Level Versioning

By default, Item Level Versioning is enabled in DSpace 7. You may choose to disable it by updating this configuration in your local.cfg:

```
versioning.enabled = false
```

Additionally, you will need to make the following changes to disable all versioning-related features:

- Switch to using the basic "HandleIdentifierProvider" in your `[dspace]/config/spring/api/identifier-services.xml` . Make sure to comment out the "VersionedHandleIdentifierProvider" and replace it with this:

```
<!-- This HandleIdentiferProvider should be used when versioning is disabled -->
<bean id="org.dspace.identifier.HandleIdentifierProvider" class="org.dspace.identifier.
HandleIdentifierProvider" scope="singleton">
    <property name="configurationService" ref="org.dspace.services.ConfigurationService"/>
</bean>
```

- Remove the "versioning" consumer from the list of default Event Consumers in either your dspace.cfg or local.cfg. Look for this configuration:

```
# Remove the "versioning" entry in this list
# (NOTE: Your list of consumers may be different based on the features you've enabled)
#event.dispatcher.default.consumers = versioning, discovery, eperson

# For example:
event.dispatcher.default.consumers = discovery, eperson
```

Once these changes are made, you will need to restart your servlet container (e.g. Tomcat) for the new settings to take effect.

Initial Requirements

The Item Level Versioning implementation builds on following requirements identified by the stakeholders who supported this contribution: Initial Requirements Analysis

1. What should be *Versionable*
    1. Versioning happens at the level of an Individual Item
    2. Versioning should preserve the current state of *metadata*, *bitstreams* and *resource policies* attached to the item.
2. Access, Search and Discovery
    1. Only the most recent version of an item is available via the search interface
    2. Previous versions of Items should continue to be visible, citable and accessible
    3. The Bitstreams for previous versions are retained. If something was once retrievable, it should always be retrievable.
3. Identifiers
    1. Each version of an Item is represented by a separate "*versioned*" identifier
    2. A base "*versionhistory*" Identifier points to the most recent version of the Item.
    3. A revision identifier also exists that is unique to the specific version.
    4. When a new version of an Item is deposited, a new revision identifier will be created.
4. Presentation
    1. On the item page, there is a link to view previous/subsequent versions.
    2. By examining the metadata or identifiers, it is possible to determine whether an item is the most recent version, the original version, or an intermediate version.
5. Access Control and Rights
    1. Certain roles should be able to generate a new version of the item via submission.
    2. To submitters, collection manager, administrators will be given to option to create new version of an item.
    3. Rights to access a specific Item should transmute as well to previous versions
    4. Rights to access a specific Bitstream should also transmute to previous versions.
6. Data Integrity
    1. The relationships between versions should not be brittle and breakable by manipulating Item metadata records.
    2. The relationships between versions should be preserved and predictable in various Metadata Exports (OAI, Packagers, ItemExport)
    3. The relationships between versions should be maintained in SWORD and AIP packaging and be maintained in updates and restorations.

User Interface

General behaviour: Linear Versioning

From the user interface, DSpace offers **linear** versioning. As opposed to hierarchical versioning, linear version has following properties:

- A new version can be created started from any available version but will be always be put at the end of the version history (it will be the latest)
- Only one in-progress version can exist at any time. When new version has been created and still needs to pass certain steps of the workflow, it is temporarily impossible to create another new version until the workflow steps are finished and the new version has replaced the previous one.

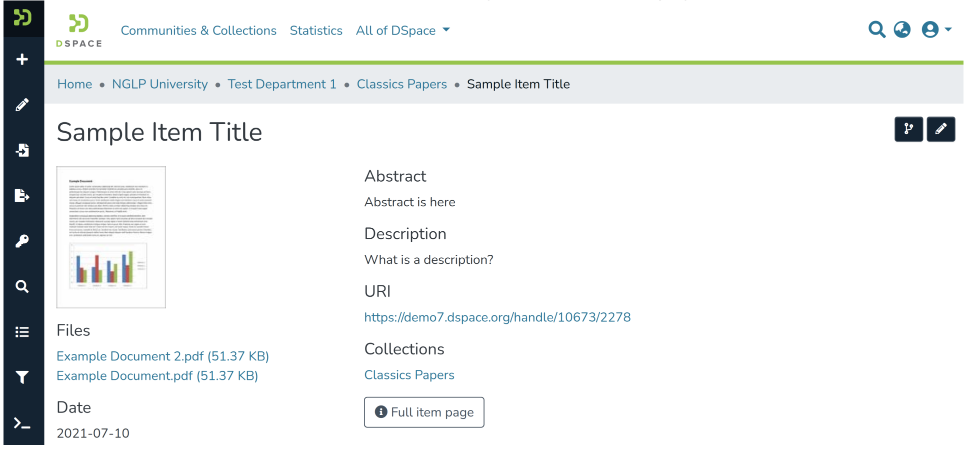
Creating a new version of an item

Administrators, collection/community administrators and eventually the original submitter can create new versions of an item from the Item View page. By default the original submitter is not allowed to create new version but the permission can be granted with the following property

modules/versioning.cfg

```
versioning.submitterCanCreateNewVersion=false
```

1. Click "Create a new version" from the buttons on the right side of the item page.



Sample Item Title

Files

Example Document 2.pdf (51.37 KB)
Example Document.pdf (51.37 KB)

Date

2021-07-10

Abstract

Abstract is here

Description

What is a description?

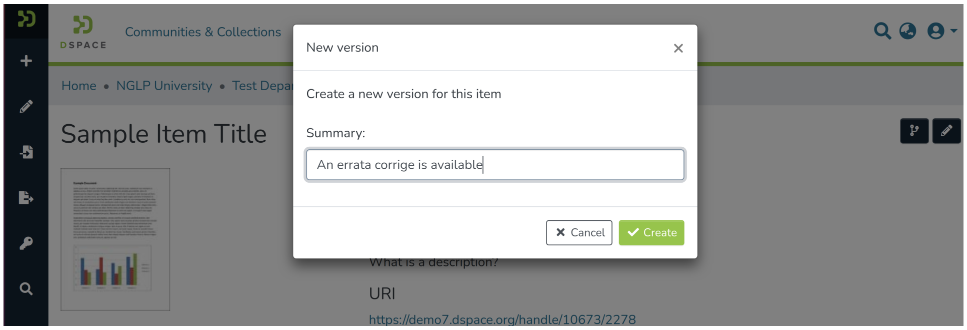
URI

https://demo7.dspace.org/handle/10673/2278

Collections

Classics Papers

2. Provide the reason for creating a new version that will later on be stored and displayed in the version summary.

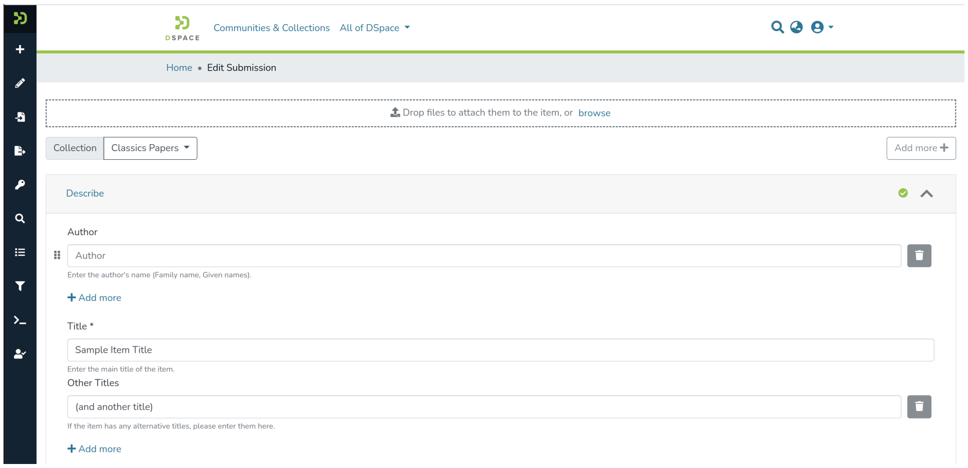


New version

Create a new version for this item

Summary: An errata corrige is available

3. Your new version is now creates as a new Item in your Workspace. It requires you to go through the submission and workflow steps like you would do for a normal, new submission to the collection. The rationale behind this is that if you are adding new files or metadata, you will also need to accept the license for them. In addition to this, the versioning functionality does not bypass any quality control embedded in the workflow steps.


DSPACE
Communities & Collections
All of DSpace
Home
Edit Submission
Drop files to attach them to the item, or browse
Add more
Describe

Collection: Classics Papers

Author: Author

Enter the author's name (Family name, Given names).

Title *: Sample Item Title

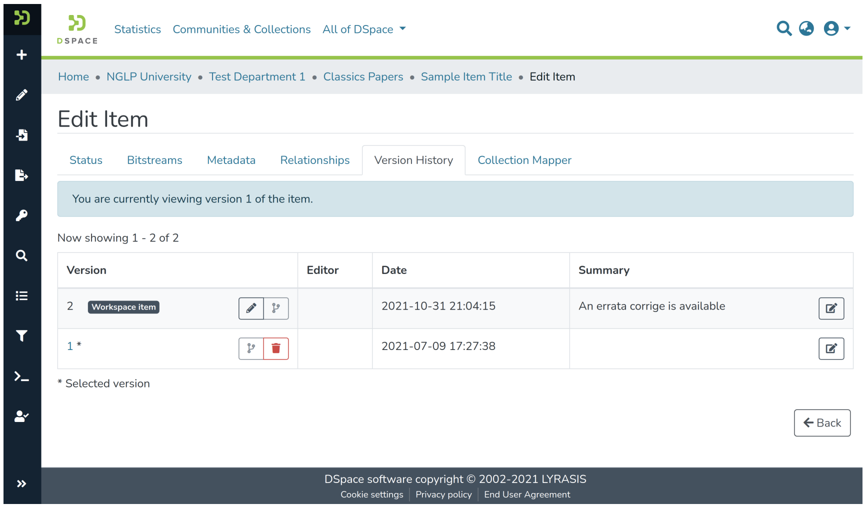
Enter the main title of the item.

Other Titles: (and another title)

If the item has any alternative titles, please enter them here.

After the submission steps and the execution of subsequent workflow steps, the new version becomes available in the repository.

Versions can be also managed via the edit item page, in the dedicated versions tab



Edit Item

You are currently viewing version 1 of the item.

Now showing 1 - 2 of 2

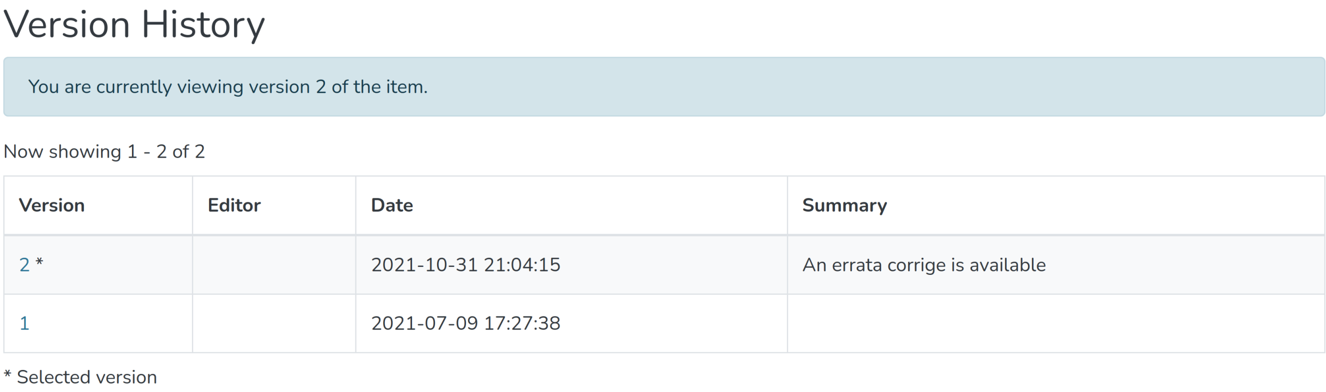
Version	Editor	Date	Summary
2 Workspace item		2021-10-31 21:04:15	An errata corrige is available
1 *		2021-07-09 17:27:38	

* Selected version

DSpace software copyright © 2002-2021 LYRASIS

View the history and older versions of an item

An overview of the version history, including links to older versions of an item, is available at the bottom of an Item View page. The repository administrator can decide whether the version history should be available to all users or restricted to administrators. By default, this information is available to all users. Information displayed includes the version number, Submitter/Editor name (only if enabled), date, and summary/description. As necessary, you may change the visibility of this table or the "Editor" column using the "Version History Visibility" configurations below.



Version History

You are currently viewing version 2 of the item.

Now showing 1 - 2 of 2

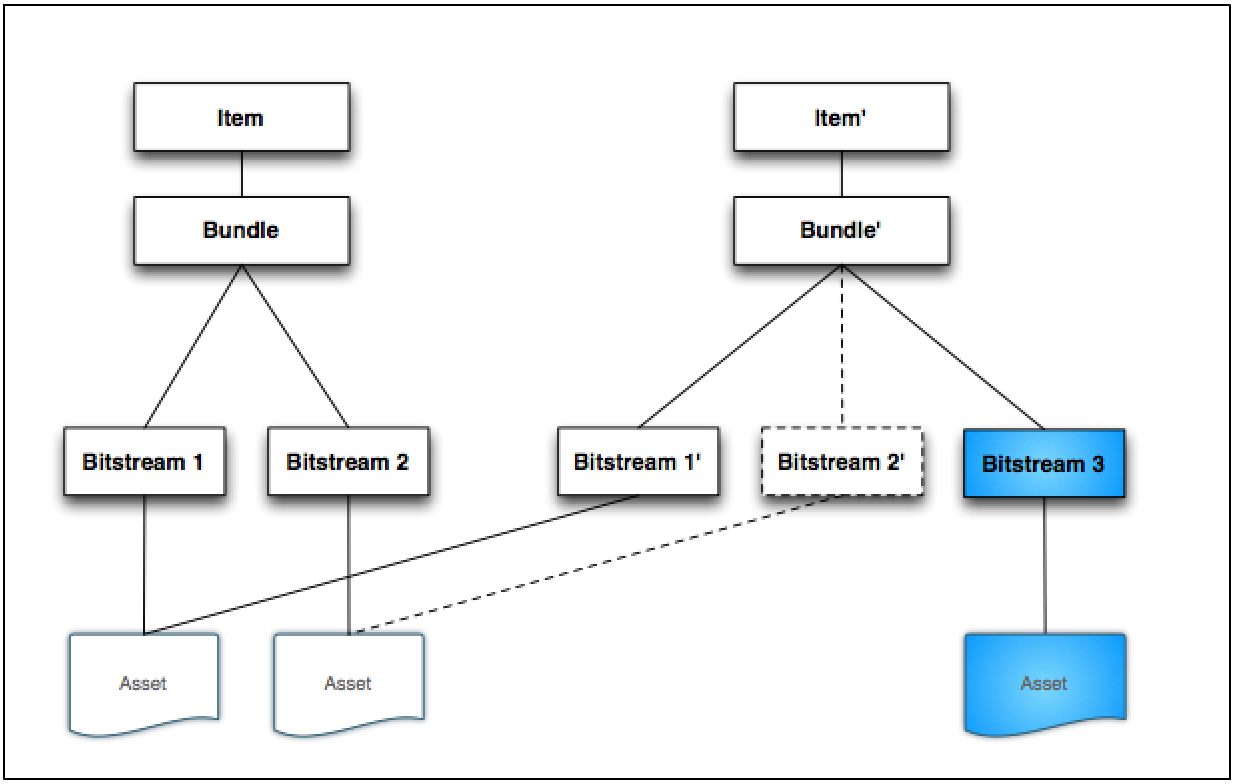
Version	Editor	Date	Summary
2 *		2021-10-31 21:04:15	An errata corrige is available
1		2021-07-09 17:27:38	

* Selected version

Architecture

Versioning model

For every new Version a separate DSpace Item will be created that replicates the metadata, bundle and bitstream records. The bitstream records will point to the same file on the disk.



The *Cleanup* method has been modified to retain the file if another Bitstream record point to it (the dotted lines in the diagram represent a bitstream deleted in the new version), in other words the file will be deleted only if the Bitstream record processed is the only one to point to the file *(count(INTERNAL_ID)=1).*

Configuration

Versioning Service Override

You can override the default behaviour of the Versioning Service using following XML configuration file, deployed under your dspace installation directory:

[dspace_installation_dir]/config/spring/api/versioning-service.xml

In this file, you can specify which metadata fields are automatically "reset" (i.e. cleared out) during the creation of a new item version. By default, all metadata values (and bitstreams) are copied over to the newly created version, with the exception of **dc.date.accessioned** and **dc.description.provenance.** You may specify additional metadata fields to reset by adding them to the "ignoredMetadataFields" property in the "versioning-service.xml" file:

```
<!-- Default Item Versioning Provider, defines behavior for replicating
     Item, Metadata, Budles and Bitstreams. Autowired at this time. -->
<bean class="org.dspace.versioning.DefaultItemVersionProvider">
    <property name="ignoredMetadataFields">
         <set>
            <value>dc.date.accessioned</value>
            <value>dc.description.provenance</value>
         </set>
    </property>
</bean>
```

Identifier Service Override

Persistent Identifiers are used to address Items within DSpace. The handle system is deeply integrated within DSpace, but since version 4 DSpace can also mint DOIs. DSpace 4 and 5 supported one type of versioned handle: The initial version of an Item got a handle, e.g. 10673/100. This handle was called the canonical one. When a newer version was created, the canonical handle was moved to identify the newest version. The previously newest version got a new handle build out of the canonical handle extended by a dot and the version number. In the image below you see a version history of an item using this handle strategy.

The canonical handle will always point to the newest version of an Item. This makes sense if you hide the version history. Normal users won't be able to find older versions and will always see just the newest one. Please keep in mind, that older versions can be accessed by "guessing" the versioned Handle if you do not remove the read policies manually. The downside of this identifier strategy is that there is no permanent handle to cite the currently newest version, as it will get a new Handle when a newer version is created.

With DSpace 6 versioned DOIs (using DataCite as DOI registration agency) were added and the default versioned Handle strategy was changed. Starting with DSpace 6 the `VersionedHandleIdentifierProvider` creates a handle for the first version of an item. Every newer version gets the same handle extended by a dot and the version number. To stay by the example from above, the first version of an Item gets the Handle 10673/100, the second version 10673/100.2, the third version 10673.3 and so on. This strategy has the downside that there is no handle pointing always to the newest version. But each version gets an identifier that can be use to cite exactly this version. If page numbers changes in newer editions the old citations stay valid.

In DSpace 4 and 5 only the strategy using canonical handles (one handle that always points to the newest version) were implemented. In DSpace 6 the strategy of creating a new handle for each version was implemented. With DSpace 6 this new strategy become the default. The strategy using canonical handle still exists in DSpace but you have to enable the `VersionedHandleIdentifierWithCanonicalHandles` in the file `[dspace]/config/spring/api/identifier-service.xml`. With DSpace 6 versioned DOIs were introduced using the strategy that every new version gets a new DOI (extended by a dot and the version numbers for versions >= 2). To use versioned Handle you have to enable DOIs, you have to use DataCite as registration agency and you have to enable the `VersionedDOIIdentifierProvider` in the named configuration file.

You can configure which persistent identifiers should be used by editing following XML configuration file, deployed under your dspace installation directory:

[dspace_installation_dir]/config/spring/api/identifier-service.xml

No changes to this file are required to enable Versioning. This file is currently only relevant if you want to keep the identifier strategy from DSpace 4 and 5 or if you want to enable DOIs or even versioned DOIs.

Version History Visibility

By default, **all** users will be able to see the version history. To ensure that only administrators can see the Version History, enable `versioning.item.history.view.admin` in the `[dspace]/config/modules/versioning.cfg` OR in your `local.cfg` file.

```
# Setting this to "true" will hide the entire "Version History" table from
# all users *except* Administrators
versioning.item.history.view.admin=false
```

Hide Editor/Submitter details in version table

In either the `[dspace]/config/modules/versioning.cfg` configuration file or your `local.cfg`, you can customize the configuration option `versioning.item.history.include.submitter`. By default this is set to false, which means that information about the submitter is only viewable by administrators. If you want to expose the submitters information to everyone (which be useful if all submitters uses generic institutional email addresses, but may conflict with local privacy laws if personal details are included) you can set this configuration property to true.

```
# This property controls whether the name of the submitter/editor
# of a version should be included in the version history of an item.
# Set to "true" to show the submitter to all users who have access to the table.
# Set to "false" to hide the submitter information from everyone except Administrators
versioning.item.history.include.submitter=false
```

Allowing submitters to version their items

With DSpace 6.0 it became possible to allow submitters to create new versions of their own items. The new versions are going through the normal workflow process if the collection is configured this way. To allow submitters to create new versions of Item they originally submitted, you have to change the configuration property `versioning.submitterCanCreateNewVersion` and set it to `true`. It is part of the configuration file `[dspace]/config/modules/versioning.cfg` but can be overridden in your `local.cfg` too.

Identified Challenges & Known Issues

Item Level Versioning has a substantial conceptual impact on many DSpace features. Therefore it has been accepted into DSpace as an optional feature. Following challenges have been identified in the current implementation. As an early adopter of the Item Level Versioning feature, your input is greatly appreciated on any of these.

Conceptual compatibility with Embargo

Lifting an embargo currently does not interact with Item Level Versioning. Additional implementation would be required to ensure that lifting an embargo actually creates a new version of the item.

Conceptual compatibility with Item Level Statistics

Both on the level of pageviews and downloads, different versions of an item are treated as different items. As a result, an end user will have the impression that the stats are being "reset" once a new version is installed, because the previous downloads and pageviews are allocated to the previous version.

One possible solution would be to present an end user with aggregated statistics across all viewers, and give administrators the possibility to view statistics per version.