

DSpace Item State Definitions

Workspace item

An item that is under submission and active edit by an authorized user. The workspace item is visible only to the submitter and the system administrators. (Currently there is no simple way to find/browse such items other than with the direct item ID or to use the supervisor functionality). Using the [supervisor functionality](#), a system admin can allow other authorized user to see/edit the item in the workspace state.

Expected use cases:

- Self deposit
- Collaboration over an in-progress submission for a small group of researchers. (This use case is implemented only with major limitations, using the supervision feature – concurrency, lack of delegation: supervision must be defined by the system administrators, etc.)

Workflow Item

An item that is under review for quality control and policy compliance. The workflow item is visible to the original submitter (currently only basic metadata are visible out-of-box in the mydspace summary list), users assigned to the specific workflow step where the item resides, and system administrators. (Currently there is no simple way to find/browse such items other than with the direct item ID or to use the abort workflow functionality).

Expected use cases:

- Quality control
- Improvements to the bibliographic record (metadata available in workflow can be different than those asked of the submitter)
- Check of policy / copyright

Withdrawn item

It is the removal of an Item from the archive. However, a withdrawn item is still available to Administrative users (and may optionally be restored to the archive at a later date). A withdrawn item disappears from DSpace (except from Administrative screens) and the item appears to be deleted.

Expected use cases:

- Staging area for item to be removed when copyright issues arise with publisher. If the copyright issue is confirmed, the item will be permanently deleted or kept in the withdrawn state for future reference.
- Logical deletion delegated to community/collection admin, where permanent deletion is reserved to system administrators
- Logical deletion, where permanent deletion is not an option for an organization
- Removal of an old version of an item, forcing redirect to a new up-to-date version of the item (this use case is not currently implemented out-of-box in DSpace)

By design, withdrawing an item is reversible. As an administrator, you can reverse the withdrawing of an item, through the action "reinstate". As a mechanism to support this, a resource policy state "WITHDRAWN_READ" was introduced.

When an item is withdrawn, all READ policies associated with the item and its underlying bundles and bitstreams, are changed into WITHDRAWN_READ policies. This achieves 2 things:

1. The READ policy information in itself is still preserved, and can get switched back to normal READ policies if the item gets reinstated.
2. As long as the item is withdrawn, those WITHDRAWN_READ policies should not give any users or groups read rights.

WITHDRAWN_READ was introduced in [DS-3097](#) after it was observed that even though an item was withdrawn, the related bitstreams were still accessible.

Non-Discoverable item (also known as Private item)

A non-discoverable (or "private") item is one that is simply **hidden** from all search/browse/OAI results, and is therefore only accessible via direct link (or bookmark). By default, all Items are *discoverable*, meaning they will appear in search/browse/OAI results.

It's important to clarify that non-discoverable items *may or may not be access restricted*. It is possible for an Item to be anonymously visible, but non-discoverable, so that you can only access the item if you are given a link to it.

This state should only refer to the discoverable nature of the item. A non-discoverable (or "private") item will not be included in any system that aims to help users to find items. So it will not appear in:

- Browse
- Recent submission
- Search result
- OAI-PMH (at least for the ListRecords and ListIdentifiers verb; though the OAI-PMH specification is not clear about inconsistent implementation of the ListRecords and GetRecord verb)
- REST list and search methods

It should be accessible under the actual Authorization Policies of DSpace using direct URL or query method such as:

- Splash page access (i.e. `/handle/<xxxxx>/<yyyyy>`)

- OAI-PMH GetRecord verb
- REST direct access /rest/item/<item-id> or equivalent

Expected use cases:

- Provide a light rights awareness feature where discovery is not enabled for search and/or browse
- Hide "special items" such as repository presentations, guides or support materials
- Hide an old version of an Item in cases where real versioning is not appropriate or liked
- Hide specific types of item such as "Item used to record Journal record: Journal Title, ISSN, Publisher etc." used as authority file for metadata (dc.relation.ispartof) of "normal item"

Archived/Published item

An item that is in a stable state, available in the repository under the defined Authorization Policies. Changes to these items are possible only for a restricted group of users (administrators) and should produce versioning according to the Institution's policy.

Embargoed Item

Are a special case of Archived/Published Item. The item has some time based access policy attached to it and/or the underlying bitstreams. Specifically, read permission for someone (EPerson Group) starting from a defined date. Typically embargo is applied to the bitstreams so that "fulltext" has initially very limited access (normally administrators or other "repository staff" groups) and only after a defined date will the fulltext become visible to all users (Anonymous group). This scenario is used to implement typical "embargo requirements" from publishers -- see [Delayed Open Access](#).

If the metadata of the item should be visible only to a specific group of users, it is possible to define an embargo policy also for the ITEM itself. A READ policy for a specific group will mean that only the users in that group will be able to access the item splash page. Note that the DSpace REST API & UI is fully rights aware (see [Discovery](#) documentation for more information, especially the section on "Access Rights Awareness"), meaning that an embargoed item is hidden automatically until the embargo expires.