

# Life and Documentation Cycle

IT components come and go. You plan, purchase and operate them and take them out of service eventually. Bingo: This article is about the lifecycle management. It plays an important role in the [IT documentation](#), since it can be used to track the state that a component has or should have. Furthermore, it should be possible to archive or delete the [IT documentation](#) itself.

## Contents

## Lifecycle of IT Components

The lifecycle of an IT component is documented in the CMDB status. The following CMD states are available with a fresh i-doit installation:

- planned
- ordered
- delivered
- assembled
- tested
- in operation (not editable)
- defect
- inoperative (not editable)
- under repair
- delivered from repair
- stored
- scrapped
- i-doit Status (not editable)
- Template (not editable)

The **CMDB status** can be displayed as a column in the object lists in order to see the target condition of a documented object as fast as possible.

The screenshot shows the i-doit CMDB interface. The main table displays a list of servers with columns for Title, Primary host address, Location path, Purpose, and CMDB status. The CMDB status column shows various states such as 'in operation', 'planned', and 'defect' with corresponding icons.

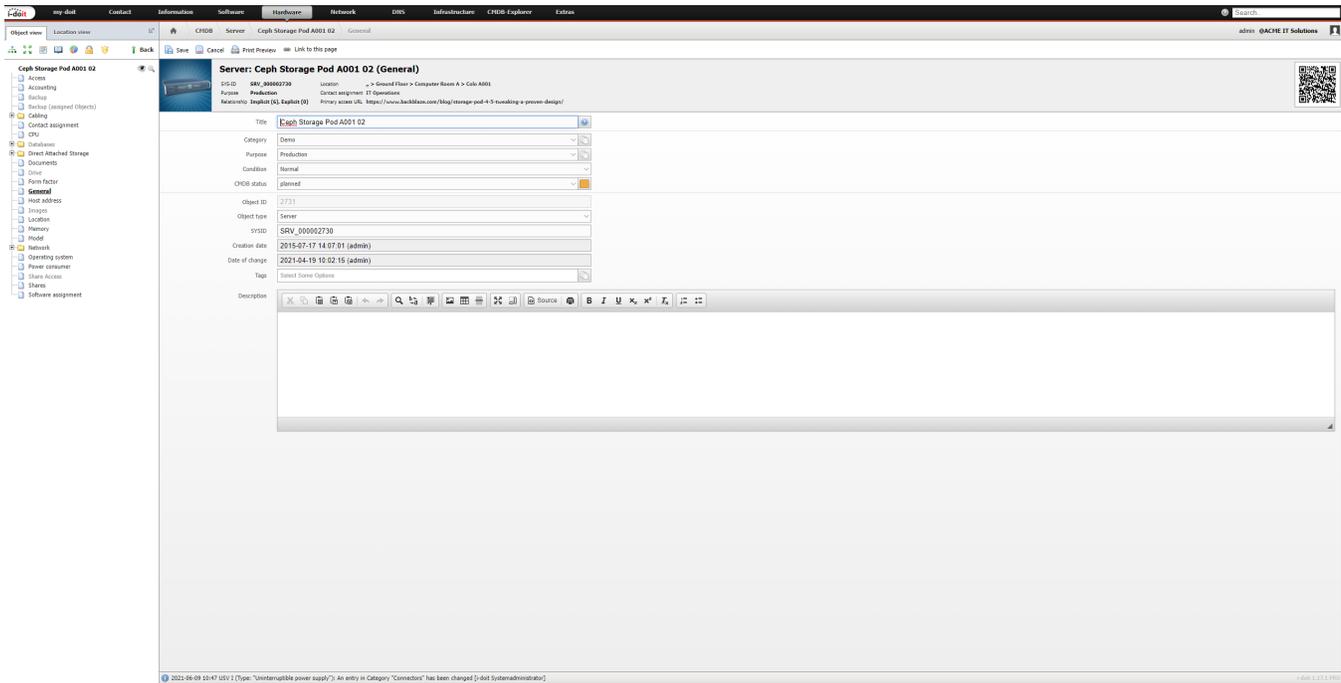
Title	Primary host address	Location path	Purpose	CMDB status
bank001	10.100.20.241	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
bank002	10.100.20.231	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
bank003	10.100.20.232	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
bank003	10.100.20.233	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
CapX Storage Pod A001 01	10.20.10.36	> ACME Data > Ground Fl > Computer > C06 A001	Production	planned
CapX Storage Pod A001 02	10.20.10.37	> ACME Data > Ground Fl > Computer > C06 A001	Production	planned
CapX Storage Pod A001 03	10.20.10.38	> ACME Data > Ground Fl > Computer > C06 A001	Production	planned
CapX Storage Pod A001 04	10.20.10.39	> ACME Data > Ground Fl > Computer > C06 A001	Production	planned
CapX Storage Pod A001 05	10.20.10.40	> ACME Data > Ground Fl > Computer > C06 A001	Production	planned
CapX Storage Pod A001 06	10.20.10.41	> ACME Data > Ground Fl > Computer > C06 A001	Production	planned
ib001	10.100.20.211	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
ib002	10.100.20.232	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
ib003	10.100.20.233	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
loadbalancer001	10.100.20.201	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
loadbalancer002	10.100.20.202	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
loadbalancer003	10.100.20.203	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web001	10.100.20.101	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web002	10.100.20.102	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web003	10.100.20.103	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web004	10.100.20.104	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web005	10.100.20.105	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web006	10.100.20.106	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web007	10.100.20.107	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web008	10.100.20.108	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web009	10.100.20.109	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation
web010	10.100.20.110	> ACME Data > Ground Fl > Computer > C06 A002	Production	in operation

### Planning

By representing the lifecycle it is possible to use *i-doit* for planning. Whether there is an upcoming big update or a relocation - the IT documentation is always present and offers valid statements.

## Set CMDB Status Per Object

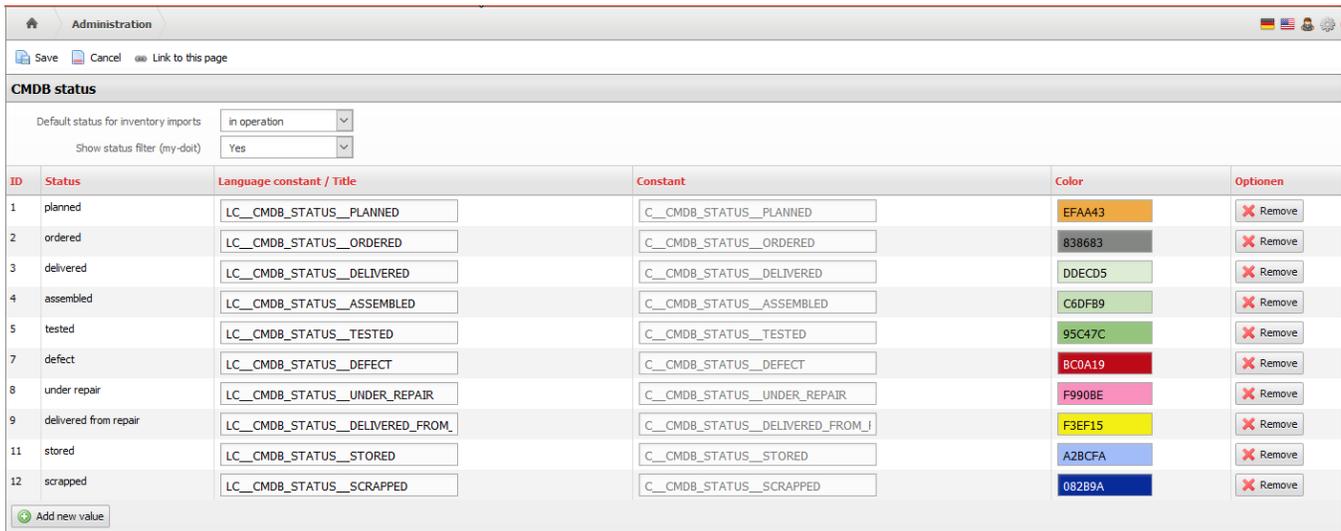
To represent the complete lifecycle of an **object**, the CMDB status **attribute** in the **General** category is used per **object**. If a new object is created, it receives **in operation** as its **CMDB status**, if not explicitly set to a different state.



## Manage CMDB Status

The adding, changing or deleting of a **CMDB status** is carried out via **Administration > CMDB settings > CMDB status**. For each **CMDB status** the following information is needed:

- Unique name/ [language constant \(for translation\)](#)
- Unique constant (helpful for using the [API](#), for example)
- Color (is used in object lists, the **General** category, in the **CMDB-Explorer** and many other cases)



## Lifecycle of the IT Documentation

Apart from the objects that are to be documented, the documentation itself can also be subject to a lifecycle. A documentation artifact can be archived once it is not needed anymore. An artifact can also be marked as deleted, so that a person who is responsible for the IT documentation can irrevocably remove this artifact via the **Purge** function.



### Deletion process

It may be worthwhile for bigger environments to establish the required processes for the archiving and deleting of documentation artifacts. At what point shall parts of the documentation be archived? Who may clean up the IT documentation? These kind of questions need to be addressed by the team. The permission system of i-doit provides the required settings to assign the rights to archive or to purge parts of the documentation to specific users.

## States

Almost all documentation artifacts (objects, category entries, values in [Dialog+-fields](#) etc.) receive a state:

- **Normal:** When working normally (create, edit), each artifact receives this state and can be used anywhere.
- **Archived:** The artifact is hidden in the [IT documentation](#). Further use, e.g. linking, is not possible anymore.
- **Deleted:** The artifact is supposed to be purged but still exists completely with all relations in the IT documentation. Apart from this, the state is similar to **Archived**.

The cycle provides that every documentation artifact receives the **Normal** status. Later on, the states will be **Archived** and then **Deleted**. A restoration to the respective previous state is possible at any time.

in addition to these three states there are also special cases concerning objects:

- **Unfinished:** This state is assigned to a new object that has been created but not saved. This happens, for example, when a new object is created but the **save** button is not pressed. These objects can only be found via a report and can then be used. You should remove the unfinished objects on a regular basis via **Administration Systemtools Cache / Database Remove unfinished objects**. As an alternative, the unfinished objects can be removed automatically. More details can be found further below.
- **Template:** An object can be used as a [template](#) for other objects.
- **Change template:** An object can be used as a change template for [mass changing](#).

If you want to delete a documentation artifact irrevocably, mark it as **Deleted** first and then use the **Purge** function. This, however, is not a state because all data (including the previous state and any logbook entries) will be deleted during the **Purge** process. This means that the object and its existence can't be traced anymore; thus you should use the function with particular caution.

## Archive Objects, Mark them as Deleted or Purge them

The state of an object is visible in the **General** category. If you wish to archive an object, mark it as deleted or purge it, you can do so via the [object lists](#). For this purpose, the checkboxes of the corresponding objects are marked and one of the buttons **Archive**, **Delete** or **Purge** needs to be pressed.

It is only possible to change to the next possible state. If an object is in the **Normal** state, it will only be possible to change it to the **Archived** state. You can only delete an object by using the filtered list of archived objects in the right-hand corner. Using **Recycle** changes the object to its previous state.

ID	Objectlink	Use primary hostaddress	Hostname	Manufacturer	Model	Form factor	Location path	CPU	Title	Share name	Category	CMDB status
1459	Fileserver01		fileserver01.i-doit.int	IBM	x3550 M2	19"	> Düsseldorf > Zentrale > Keller: Server.. > Rack 01	4@ 2.53 GHz	Xeon E5540	Daten		in operation
3341	Testserver	-		Hewlett Packard		19"	> Düsseldorf > Zentrale > Raum 00.02 > Testrack	-			Other	in operation
2406	Webserver bei 2 & 2	-		Not specified			-	-				in operation
281	ZN-Server01	192.168.0.40	zn-srv01.i-doit.int	IBM	x3550 M2	19"	> Düsseldorf > Zentrale > Keller: Server.. > Rack 01	4@ 2.53 GHz	Xeon E5540			in operation
287	ZN-Server02	192.168.0.37	zn-srv02.i-doit.int	IBM	x3550 M2	19"	> Düsseldorf > Zentrale > Keller: Server.. > Rack 01	4@ 2.53 GHz	Xeon E5540			in operation
293	ZN-Server03	192.168.0.38	zn-srv03.i-doit.int	Hewlett Packard		19"	> Düsseldorf > Zentrale > Keller: Server.. > Rack 01	4@ 2.53 GHz	Xeon E5540			in operation

There is no further query when purging an object except if relations to other objects exist.

## Archive Category Entries, Mark them as Deleted or Purge them

A similar functionality as with the objects exists for some [list categories](#) ("Multi-value"). With this function you can archive category entries, mark them as deleted or purge them.

## Simplified Deleting (Quickpurge)

In order to purge a documentation artifact it first has to be archived and marked as deleted. To shorten this cycle, it is possible to activate the **Quickpurge** button. This is done via **Administration System settings System parameters Activate Quickpurge button**. By this means you can purge an object or category entry regardless of its state.

## Listing of all Objects that are Archived or Marked as Deleted

In order to receive a list with all objects which are archived or marked as deleted, you can use a report which can be configured via the Query Builder.

## Remove Objects or Category Entries that are Unfinished/ Archived/ Marked as Deleted Collectively (Purge)

In almost all cases, unfinished objects are unwanted since they are not visible and thus can not be edited. However, also archived objects or objects marked as deleted are often undesired. The same applies to category entries. Therefore it is a good idea to purge these unwanted documentation artifacts. To this end you can use two methods: manual or automatic deletion.

### Manual Deletion

You can delete the artifacts via the web GUI. The respective function is located at **Administration Systemtools Cache / Database Objects** or **Categories**. After using one of the provided buttons, a notification showing the amount of objects/ category entries which you are about to delete, will be displayed. After the deleting process it also shows statistics of the amount of objects/ category entries that have been deleted.

### Automated Deletion

The [controller](#) also offers a possibility to delete undesired objects irrevocably. The corresponding handler is called `cleanup_objects`. The `-t` parameter determines the group of objects that are to be deleted based on their states:

- `-t 1`: delete unfinished objects
- `-t 3`: delete archived objects
- `-t 4`: delete objects that are marked as deleted

Example for the deletion of unfinished objects:

```
sudo -u www-data console.php system-objectcleanup -u admin -p admin -i 1 --objectStatus 3
```

The automation consists of executing the controller on a regular basis via cron job. An example with other important cron jobs can be found in [the article regarding the controller](#).

## Change of States in the Logbook

Changes of states are recorded completely in the logbook. Only when an object/ category entry is purged, the associated logbook entries will be permanently deleted.