

# State and University Library of Denmark - Fedora Enhanced Content Models

## Fedora Enhanced Content Models

**Warning** This page is no longer updated. The project is moved to <http://ecm.sourceforge.net>

Fedora is at the core of the new DOMS (Digital Object Management System) being developed at the State and University Library. For this system, we needed more powerful content models. More specifically, we needed content models that describe the xml schema for datastreams, cardinality restrictions on relations and allowed types for the target of relations. In addition, we needed to retain compability with the original fedora system.

We did this by enhancing the DS-COMPOSITE-MODEL datastream in content models, and adding the ONTOLOGY datastream to content models. The ONTOLOGY datastream specifies restrictions on relations. DS-COMPOSITE-MODEL specifies a location for an xml schema for the described datastream.

To make sure the repository stayed consistent, we build a webservice to validate objects. Given a pid, this webservice will look to the object's content models, and validate the object against the definitions there.

All we needed now was a way to make sure that the validator was run after all changes, and only changes that did not produce errors were allowed. This was complicated by the fact that "a change" to an object might not be the same as "one call to API-M", but rather a series of calls. Fedora already has a STATE property for each object, restricted to the values "Active", "Inactive" and "Deleted". There seem to be no predefined meaning attached to this state, so we added our own.

We used the decorator system for the Management module to add the following meaning. If an object is in the "Active" state, the only allowed change is changing state to "Inactive". All other changing API-M methods throws exceptions. So, you cannot change an "Active" object, without making it "Inactive" first.

Then, in "Inactive" all changes are allowed, but when you try to change the state back to "Active", the validator webservice is invoked. If the object passes validation, the state is changed, otherwise the operation throws an exception with an error report message.

So, the meaning attached to the state is now:

- Active: The object is published and available. The object has been validated against the content models that describes it.
- Inactive: The object is not publicly available. It is not guaranteed to be valid in regards to the content models that describes it.
- Deleted: The object is deleted, and should not be available to anyone. It is in no way guaranteed to be valid. It is still in the repository, and special administration tools should be able to resurrect it.

For further description and code look to the children of this page

## Presentations

I gave a presentation at the Developers Happiness Days (dev8d) in February 2009. The slides can be found [here](#)

I gave a presentation at the "2nd European Workshop on the Use of Digital Object Repository Systems in Digital Libraries (DORS DL2)", the slides of which can be found [here](#). This version is now somewhat outdated.