

Concept Mapping - Fedora 3 to 4

This is a draft document. The contents are subject to change until this warning is removed.

Upgrading/migrating from Fedora 3 to 4? Wondering what has changed and what has stayed the same? Look no further - this page will outline the main differences and map important concepts between Fedora 3 and 4.

New Concepts

Fedora 4 implements the [W3C Linked Data Platform](#) (LDP) recommendation. This standard defines how applications can find resources and follow links, publish new resources, edit and delete existing ones using the HTTP protocol. LDP specifies a new set of concepts for Fedora 4 that can be approximately mapped to Fedora 3 concepts.

Fedora 3	Fedora 4	Comments
Object	Container	A container is roughly equivalent to a Fedora 3 object - a conceptual representation of a thing that can contain files or other containers.
Datastream	Non-RDF Source	A Non-RDF Source (or binary) is simply a bitstream (e.g. JPG, PDF, MP3, etc.).
	Resource	Both Containers and Non-RDF Sources are types of resources with dereferenceable URIs .

Differences

Object Model

Fedora 3 objects are FOXML (Fedora Object eXtensible Markup Language) documents. Fedora 4 resources are not composed of XML; instead, they are stored in [ModeShape](#) as nodes with RDF properties. Fedora 4 resources cannot contain inline XML, so any inline XML contained in Fedora 3 objects will need to be converted to separate XML documents or RDF properties.

Repository Structure

There is no inherent tree structure in Fedora 3; objects are all at the top level of the repository. By contrast, Fedora 4 resources are laid out in a tree structure descending from a root resource. You can define your own tree structure or allow the system to automatically generate this hierarchy for you. Note that this is separate from any semantic relationships you may define between resources using RDF triples.

A balanced tree, where no resource has more than ~3000 child resources, offers the best performance. There is an issue in the [ModeShape issue tracker](#) to address the performance impact associated with too many child resources of a given resource. If this issue is important to you, [please add your vote](#).

File System

In Fedora 3, there is an objects directory and a datastreams directory where content is stored in a [PairTree](#). Storage is much more configurable in Fedora 4 - by default, containers are stored in a database (e.g. LevelDB) and binaries are stored in a [PairTree](#) on the file system. You may alter the configuration to store containers and binaries differently, though this may have an effect (positive or negative) on performance.

Identifiers

Fedora 3 objects have Persistent Identifiers (PIDs) that provide a means of uniquely identifying each object within the repository. Since Fedora 4 is an LDP server, the repository resource identifiers are fundamentally their URIs. Fedora 3 PIDs (or any other identifiers) may still be stored as properties on Fedora 4 resources.