FAQs and How-Tos

Platform Description

The Fedora 4 platform is a ground-up reimagining of the Fedora repository architecture. We've built atop mature products in the content repository space to allow us to rapidly iterate to build a robust, scalable, and durable system.

Fedora 4 introduces a number of new features and stuff:

- a different kind of object model that allows (and, even encourages) hierarchy
- "native" RDF expressions of object properties
- a RESTful RESTful HTTP API that is consistent and follow-your-nose
- integrated support for low-level fixity checks
- asynchronous, event-driven points of extensibility

We expose our underpinning technologies (at the Java API level, at least) for developers, so it is also helpful (and sometimes even necessary) to be familiar with the features and functions those technologies offer:

Modeshape

Modeshape is an implementation of the JSR-283 Java API for content repositories ("JCR"). Fedora 4 wraps the Modeshape Java API with our own REST and Java APIs.

> *ModeShape is a distributed, hierarchical, transactional, and consistent data store with support for events, versioning, references, and flexible schemas. It is very fast, highly available, extremely scalable, and it is 100% open source and written in Java.*
>
> *ModeShape is perfect for data that is organized in a tree-like hierarchical structure where related data is stored close together, where navigation to related content is just as common and important as fast key-based lookups or queries. The hierarchical organization is similar to a file system, making ModeShape a natural for storing files annotated with metadata. ModeShape can even automatically extract the structured information within the files so that clients can navigate or use typed queries to find files satisfying complex, structurally-oriented criteria. ModeShape is an excellent store for data with a complex schema, since the schema can vary over the database and evolve over time. ModeShape is the perfect distributed data store for all kinds of applications, including repositories, content management systems, historical data services, provisioning and governance systems, and metadata management systems.*

Modeshape stores object metadata in an Infinispan cache. Binary content MAY be stored in Infinispan, or in an alternative BinaryStore. Binary values are de-duplicated based on the SHA-1 hash of their content at the BinaryStore layer (meaning if you add 2 datastreams with identical content, it'll only store that 1 time in the storage system).

Modeshape provides additional points of extensibility (e.g. Sequencers), and support for widely implemented APIs like CMIS, WebDAV, etc.

Modeshape also provides a "federation" feature, where:

- Clients (eg: such as Fedora 4) can access internal data (owned by ModeShape) and external data (owned by an external system) in exactly the same way, using the JCR API. ModeShape might *cache* this external data (for performance reasons), but it would never *store* any of this external data.

Fedora uses this feature to provide "instant ingest", where you can stage content on a filesystem, initiate an ingest into Fedora, and while that process occurs, Fedora can still serve up the content directly from the filesystem.

See also:

ModeShape Notes

Infinispan

Infinispan is the storage subsystem used by Modeshape for storing object structure, and (optionally) binary content. It supports cluster-based scale out and high availability, data persistence into a variety of CacheStore architectures (filesystem, JDBC database, Amazon S3), and distributed execution (including but not limited to e.g. Map/Reduce).

Fedora 4 ships with a handful of example Infinispan configurations to get up and running quickly.

file	no-frills, FileCacheStore backed

clustered	a trivial, cluster-ready example; replicates metadata, distributes 2 copies of content
leveldb-default	a leveldb backed metadata store (that's really fast)
leveldb	a leveldb backed metadata store, with separate caches for resource, properties, and binaries
ram	an in-RAM-only cache store for testing

See also:

Clustering modes

Cache Loaders and Stores

Eviction

Introduction

First Steps

The Fedora object model

RESTful HTTP API

How-Tos

- Setup Camel Message Integrations
- Setup OAI-PMH Provider
- How to inspect LevelDB
- How to Run Fedora4 in AWS
- How to Get Involved
- How to Create a new JAX-RS resource
- How to setup a Fedora cluster
- Islandora - Test Migration
- Custom Fedora 3.x - Test Migrations
- Islandora - Install Migration Tools
- Export and Review Fedora 3.x Content
- Custom Fedora 3.x - Update Client Applications
- Custom Fedora 3.x - Install Migration Tool
- How to Configure Servlet Container Authentication
- Basic Understanding of OCFL