

# The Fedora 4 object model

(Cribbed heavily from the JCR Repository Model documentation)

This document describes the objects and structures that compose a Fedora repository.

A Fedora 4 repository consists of a directed acyclic graph of resources where edges represent a parent-child relation.

For those without a computer science background, in repository terms:

- A **resource** is an object or datastream.
- An **edge** is a relationship (in this case, a parent-child relationship) that links two **resources**. Edges are represented as RDF properties in a Fedora object.
- A **graph** is a set of **resources** connected by **edges**. The entire set of objects and their relationships to each other form a graph.
- A **directed graph** is a graph whose **edges** have direction associated with them: that is, the relations that link the resources move in a given direction from one resource to the next. An example of an edge with a direction is *isChildOf*. The relationship expresses a direction from the child to the parent. The relationship *isParentOf* can also link the same two resources, but in the opposite direction.
- A **directed acyclic graph** is a graph in which edges never form loops (i.e., cycles) of resources. That is, if you start at resource A, then follow a set of identical relations from that resource to other resources in a chain, you will never end up back at resource A.
- A hierarchical tree, such as the one modeled in the Fedora repository by the relationships provided by the JCR underpinnings of Fedora, is a classic example of a **directed acyclic graph of resources**. In this sense, no Fedora object can ever be its own grandpa. Of course, the additional relationships that you, the user, distribute amongst your objects are entirely up to you. You can make them as simple or complex as you like.

The repository contains at least one item, the *root resource*. The root resource is the only item without a parent; all other items have at least one parent.

Resources can have zero or more child resources. Resources can have zero or more properties. Properties can hold one or more values.

The location of an item in the graph can be described by the path from the root resource to that item. The path consists of the name of each interceding resource in order from root to target item, much like a file system path.

The name of an item consists of the namespace prefix and a local name component. The namespace prefix must be registered before use.

The local name of a resource is limited to the range of valid XML characters (<http://www.w3.org/TR/xml/#NT-Char>)

#x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]

except

' | ':' | '[' | ']' | '\*'

In addition to a path, every resource also has an identifier. The identifier is independent of the path and provides an identity to the resource that is stable across moves within the repository.

Some resources may possess a "binary" payload with a length and digest. These resources are normally leaves in the tree (meaning they have no child resources of their own). They are the best match in Fedora 4 to the Fedora 3 concept of "datastream".

The binary content is de-duplicated across the repository. If you store the same binary stream in five binary resources, it will only be persisted once, unless you expressly configure otherwise.

Properties may be single- or multi-valued, and are typed. Some properties are reserved for internal-use under mutation only (e.g. those in the fedora-internal namespace), but can otherwise be used freely by implementors. By default, properties are multivalued string types.

Fedora internal properties exist in the following namespace:

<http://fedora.info/definitions/v4/repository>