

# Versioning

## Overview

Versioning in Fedora implements the Memento protocol. Its formal name is HTTP Framework for Time-based Access to Resource States – Memento. This document focuses exclusively on how the Fedora software implements the Memento software and best practices when using Fedora Versioning.

It is worthwhile reading the [entire Memento specification](#) to become acquainted with the Memento terminology (such as Original Resource, a Memento resource, Timemap, Timegate) and various interaction patterns, but the most relevant sections of the Memento specification to Fedora are [section 4.1.1 and its relevant parent sections](#). Section 4.1.1 presents the HTTP/Memento interaction pattern implemented in Fedora. Also, familiarizing yourself with the [Versioning section \(Section 4\)](#) of the Fedora API Specification will be helpful to understanding concepts and terminology (LDPRv, LDPCv, LDPRm).

## Terminology

A mapping of terminology between the Memento definition of resources and those used in the Fedora API Specification is helpful:

### **Memento Original Resource = Fedora API LDPRv**

This is the resource of which versions can be made. Also called the versionable resource.

### **Memento Timegate = Fedora API LDPRv**

A Memento Timegate is a resource to which a request can be made for a relevant versioned resource (LDPRm, see below). The Memento specification allows the Original Resource and the Timegate to be the same resource. Fedora has taken advantage of this efficiency and uses the LDPRv as the Memento Timegate in addition to the LDPRv being the Memento Original Resource.

### **Memento “Memento” Resource = Fedora API LDPRm**

This is the versioned resource. It is usually a snapshot of a LDPRv (a.k.a Memento Original Resource).

### **Memento Timemap = Fedora API LDPCv**

This is a LDP container (hence the ‘C’) that aggregates versioned resources (LDPRm resources). A POST request to this resource will result in a new LDPRm (Memento Resource).

## RESTful API - Versioning

### RESTful HTTP API - Versioning

- [Working with a versionable resource \(a LDPRv\)](#)
  - [GET Retrieve the versioned resource from a specific point in time.](#)
- [Working with a versions container \(a LDPCv\)](#)
  - [GET Get a list of the available versions of an object](#)
  - [POST Create a new versioned resource \(a new LDPRm\)](#)
  - [DELETE Remove the versions container and all versioned resources](#)
- [Working with a versioned resource \(a LDPRm\)](#)
  - [GET Get a specific versioned resource](#)
  - [DELETE Remove a previous version of an object](#)

## Memento Datetime Negotiation Algorithm

The date algorithm Fedora uses to determine which LDPRm should be returned to a TimeGate request, as indicated by the value of the Accept-Datetime header, is as follows:

- Any Accept-Datetime value that is earlier than the first/oldest Memento will return a reference to the first/oldest Memento.
- Any Accept-Datetime value that is later than a specific Memento’s timestamp but earlier than another Memento’s timestamp will return the older Memento.
- An Accept-Datetime value that matches a stored Memento datetime value perfectly will return the Memento preserved at that timestamp.
- An Accept-Datetime value that is later than the last Memento will return a reference to the last/youngest Memento.

Fedora follows Memento Datetime negotiation pattern 1.1 as described in [section 4.1.1 of the Memento specification](#), in which the Original Resource Acts as its own TimeGate (URI-R=URI-G; 302-Style Negotiation; Distinct URI-M).

## Note bene

Fedora does not automatically initiate versioning for all resources. The client MUST initiate versioning by invoking a special Link header at the time a resource is created. See ?????

It is possible to convert a LDPR into a LDPRv after the LDPR exists, but this process effectively forces the client to recreate the existing resource as a new resource by performing a GET on the existing resource and then resubmitting the resource, via a PUT, with the proper Link header.

The Memento specification strongly suggests clients initiate negotiation with a HEAD request. This is largely because **all** interaction with Memento operates within HTTP Headers. They must be examined closely. Headers are the only way to learn that a resource is a LDPRv (a Memento Original Resource and a Memento Timegate) or a LDPRm (a Memento).

The subject of the resource returned from a GET response to a LDPRm (a Memento) will be **same** as the subject of the "parent" resource (the Memento Original Resource or LDPR). This is one of the reasons it is imperative to inspect the headers of these responses as it may only be possible to determine that the resource is a LDPRm from the response headers.

Fedora will emit messages after the creation of a LDPRv resource (Memento Original Resource), the creation of LDPRm resources (Mementos), and the deletion of LDPCv resources (Memento Timemaps) and LDPRm resources. These messages provide no hint that they pertain to versioning. If software acts on these messages, that software should perform a HEAD request on the resource and inspect the headers to determine if the resource is related to Fedora versioning, and if further action is warranted.

LDPRm resources should generally be created by issuing a simple POST request to the LDPCv (Memento Timemap) resource, but it is technically possible to issue a POST request with a message body, which will be stored as a LDPRm (Memento), and a timestamp, which will be used as the Memento Datetime. This functionality – providing the resource to be stored as a versioned resource - exists to address the need to be able to Import data to Fedora as a part of a migration. It is inadvisable to create LDPRm resources with POST and a message body without specific need to take advantage of this feature.