Fixity Checking

- Supported Digests
- Transmission Fixity
- Persistence Fixity
    - On-demand Digest Calculation
    - On-demand Fixity Check
- Default Digest Algorithm

Fedora supports two types of fixity operations: Transmission fixity and persistence fixity.

Note: the information in this document applies to both internal and external content binaries.

Supported Digests

For these operations, Fedora supports the following digest algorithms:

- SHA-1
- SHA-256
- SHA-512
- SHA-512/256
- MD5

Transmission Fixity

Transmission fixity provides a means to verify the integrity of the binary content when ingested, that is, when it is transmitted to the repository. When performing a PUT or a POST of binary content, you can supply one or more digests for the binary resource in a request header. If you provide a supported digest, Fedora will internally calculate the binary's checksum and compare it to the provided digest. If they match, the binary will be stored and an appropriate response (HTTP Code 201 usually) sent to the client. A mismatch will result in a HTTP 409 response.

Digests provided in this manner will be stored as metadata in Fedora, and can be found under the `<http://www.loc.gov/premis/rdf/v1#hasMessageDigest>` predicate when requesting the *fcr:metadata* resource for that binary.

Example: Creating a binary with a SHA1

```
# Create the binary with a SHA1
echo "test content" |
curl -XPUT http://localhost:8080/rest/binary_with_sha1 --data-binary @- -H "Digest:
sha=4fe2b8dd12cd9cd6a413ea960cd8c09c25f19527"

# HTTP/1.1 201 Created

# Retrieve metadata for the binary
curl -ufedoraAdmin:fedoraAdmin http://localhost:8080/rest/binary_with_sha1

# <http://localhost:8080/rest/binary_with_sha1>
#     ...
#      premis:hasMessageDigest <urn:sha1:4fe2b8dd12cd9cd6a413ea960cd8c09c25f19527> ;
```

With an invalid SHA1

```
# Create the binary with a SHA1
echo "test content" |
curl -XPUT http://localhost:8080/rest/binary_with_sha1 --data-binary @- -H "Digest:
sha=003d0450f6f7e6db635a04d23245b68e13365463" -v

# HTTP/1.1 409 Conflict
# Checksum mismatch, computed SHA digest 4fe2b8dd12cd9cd6a413ea960cd8c09c25f19527 did not match expected value
003d0450f6f7e6db635a04d23245b68e13365463
```

With multiple digests

```
echo "test content" |
curl -XPUT http://localhost:8080/rest/binary_with_sha1 --data-binary @- -H "Digest:
sha=003d0450f6f7e6db635a04d23245b68e13365463, md5=d6eb32081c822ed572b70567826d9d9d"
```

For more information about creating binaries, see See RESTful HTTP API - Containers, POST, Example 4

Persistence Fixity

Persistence fixity provides a means for you to compare an on-demand digest calculation to recorded digest values. There are two approaches to persistence fixity.

On-demand Digest Calculation

Users can perform a HEAD or GET request on a binary resource and include, as part of the request, a "Want-Digest" header, indicating that the server should recalculate the message digest per the provided algorithm and return the calculated digest as part of the response headers. In this way, the client can compare the calculated value with the stored value to determine the integrity of the on-disk stored content. A mismatch suggests some form of binary corruption and the repository manager should investigate and take appropriate actions to remediate.

See RESTful HTTP API - Fixity for usage.

Example: Requesting multiple digests

```
curl -I -H "Want-Digest: sha-256, sha-512/256" "http://localhost:8080/rest/binary_with_sha1"

# Digest: sha-512/256=00042c9c081986a401e25a1576b9aaa3c83d215a8efa180a73778203e3b4a678,sha-
256=a1fff0ffefb9eace7230c24e50731f0a91c62f9cefdfe77121c2f607125dffae
```

On-demand Fixity Check

An on-demand fixity check can be requested by making a GET request to the /fcr:fixity endpoint of a binary. This will recalculate the digests of each algorithm that is on record for the binary, and compare against those stored digests. The results of the check will be returned in an RDF serialization. See RESTful HTTP API - Fixity for usage.

Default Digest Algorithm

By default, Fedora will calculate and store a SHA-512 digest for all binaries if none were provided via "Digest" header. This digest will be available via the fcr:metadata endpoint for the binary, stored under a `<http://www.loc.gov/premis/rdf/v1#hasMessageDigest>` property.

The default digest algorithm used by Fedora may be set to either SHA-512 or SHA-256, via the `fcrepo.persistence.defaultDigestAlgorithm` option. This will affect the default digest for binaries, and will also determine the digest algorithm used for objects in OCFL.

In the example below, Fedora automatically adds a SHA-512:

```
# Create a binary without providing a digest
echo "test content" |
curl -XPUT http://localhost:8080/rest/binary_with_default --data-binary @-

# Retrieve the binary's metadata
curl http://localhost:8080/rest/binary_with_default/fcr:metadata

# <http://localhost:8080/rest/binary_with_default>
#         ...
#         premis:hasMessageDigest  <urn:sha-512:
b22137a0e8969282b85e3f9375448307d14c5aabf41be66c4f6a0323bd03a3935972021e4c34aa30914e37b03c22594fe180eea9790e9ff1
47016c9dfae39d5a> ;
```