Backup and Restore

Overview

The Fedora 4 Backup capability allows a user, such as the repository manager, make a REST call to have the repository binaries and metadata exported to the local file system. Inversely, the Restore capability allows a user to make a REST call to have the repository restored from the contents of a previous Backup operation. In addition, with the default configuration, files are stored on disk named according to their SHA1 digest, so a filesystem backup approach can also be used.

Design Considerations
Historically, Fedora fulfilled its promise of durability by choosing transparent forms of persistence (e.g. human-readable XML) and using them in ways that systems outside the repository could readily penetrate if needed. Transparency in support of durability is as valid a principle as ever, but there is a weakness to it: transparent forms of persistence are not performant. What's more, many users didn't particularly care for that principle, but they were still stuck paying the performance costs associated with it. So in Fedora 4, we shifted responsibility for transparent persistence away from the core repository software. If you'd like to maintain some simple, human-readable form of your repository, that's fine, but you need to support that with an integration around the core. The form of persistence used by the core repository component itself is not meant to be manipulated directly by a human except in the most unusual situations, it's meant instead for use by the software to provide speedy service at the repository's API. You might compare this to the use of database software. You don't expect to directly manipulate database indexes, and if you are concerned for the durability of your data in the database, you take backups in a transparent format and use _those_ to ensure durability. An analogy: you may expect your bank to provide downloadable images for any checks you write, but you don't expect them to use those images to run their accounting software.

Usage

Backup

If a POST body specifying a writeable directory (local to Fedora 4 server) is not included in the request, the backup will be written to the system temp directory.

Perform a backup of a running Fedora 4 repository

Request

```
POST /rest/fcr:backup

> optional POST body
```

Response

On success

- HTTP/1.1 200 OK
- Path where the backup was written

Response body

- Absolute path of local backup directory

Restore

Note: Restoring a backup replaces the repository content with the contents of the backup, so any data in the repository will be lost.

Perform a restore of a running Fedora 4 repository

Request

```
POST /rest/fcr:restore

> with POST body
```

A POST body containing the full path to a previous backup.

Response

On success

- HTTP/1.1 204 No Content

Backup Format

Regardless of the repository configuration, the output of the backup process creates resources of the same format. Further details on backup contents and the underlying implementation can be found in ModeShape's documentation.

The backup directory will contain

- 'binaries' directory that contains the repository "content" binaries stored in a pair-tree like structure. The filename of the binary is the SHA-1 of the content with the extension '.bin'. The directory structure in which each binary is found is three levels deep based on the SHA-1.
    - For example, binary content in the repository with a SHA-1 of "5613537644c4d081c1dc3530fdb1a2fe843e570170d3d054", would look like

```
binaries
    44
        c4
            d0
                44c4d081c1dc3530fdb1a2fe843e570170d3d054.bin
```

- One or more "documents_00000n.bin.gz" files which contains a concatenated listing of the metadata for each of the repository objects in JSON format
    - For example

```
{ "metadata" :
  { "id" : "87a0a8c317f1e7/jcr:system/jcr:nodeTypes/nt:unstructured//undefined/1" ,
    "contentType" : "application/json" } ,
  "content" :
  { "key" : "87a0a8c317f1e7/jcr:system/jcr:nodeTypes/nt:unstructured//undefined/1" ,
    "parent" : "87a0a8c317f1e7/jcr:system/jcr:nodeTypes/nt:unstructured" ,
    "properties" :
    { "http://www.jcp.org/jcr/1.0" :
      { "primaryType" :
        { "$name" : "nt:propertyDefinition" } ,
        "onParentVersion" : "COPY" ,
        "multiple" : false ,
        "protected" : false ,
        "availableQueryOperators" :
          [ "jcr.operator.equal.to" ,
            "jcr.operator.greater.than" ,
            "jcr.operator.greater.than.or.equal.to" ,
            "jcr.operator.less.than" ,
            "jcr.operator.less.than.or.equal.to" ,
            "jcr.operator.like" ,
            "jcr.operator.not.equal.to" ] ,
        "requiredType" : "UNDEFINED" ,
        "mandatory" : false ,
        "autoCreated" : false }
    }
  }
}
```

Filesystem Backup

By default, files larger than 4KB are stored on disk named after their SHA1 digest, in the directory `fcrepo.binary.directory`. (4KB is the default, but can be changed by updating the `minimumBinarySizeInBytes` property in repository.json). That is, a file with the SHA1 "a1b2c369563c0465ab82cdb2789d45ce1c3585b1" would be stored on disk in `/path/to/fcrepo4-data/fcrepo.binary.directory/a1/b2/c3/a1b2c369563c0465ab82cdb2789d45ce1c3585b1`. So files in the repository can be backed up backing up the directory `fcrepo.binary.directory`.