Upgrading Avalon 5.1.6 to Avalon 6.0

For upgrades from Avalon 5 to 6.1, perform the actions in Upgrading Avalon 6.0 to Avalon 6.1 after completing the migration detailed on this page.

See What's New in Avalon 6 for more details.

Prepare R6 for migration

The option to upgrade in place is not available to go from Release 5 of Avalon to Release 6. Instead, follow these steps.

1. Back up your Avalon 5.0 installation
2. Stand up an R6 instance following the Manual Installation Instructions and run it concurrently with an existing R5 instance.
3. Make R5 fedora accessible to the server running the R6 instance. (See the fedora migration step below.)
4. Use the same authentication scheme for R6 as the one used for R5. Otherwise there could be issues with users/roles.
5. Ensure the R6 instance is empty. Run wipeout rake task to clean out R6 database, solr, and fedora.

```
# ensure that Fedora, Solr, and the DB are clean
# it will ask you to verify the targets are correct
bundle exec rake avalon:wipeout
```

Migrate Config Files

- config/authentication.yml – has a new structure with a different section for each rails environment. Edit your authentication particulars to incorporate this new structure. See example: config/authentication.yml.example.
- config/controlled_vocabulary.yml – should be copied from R5 to R6 to support data migrations
- config/initializers/permalink.rb – if in use, it will likely need to be adjusted to refer to objects in the Fedora 4 way (id, instead of pid)

Dump R5 database

Data from your R5 database will be loaded directly into your R6 database then migrated for R6 in a later step. Role_map data will be needed in R6 for the Fedora migrate step, so it needs to be imported first.

```
# This example is for mysql. Adjust as necessary for your R5 database.
mysqldump -h <hostname> -u <username> -p <databasename> --no-create-info --complete-insert --tables annotations
api_tokens bookmarks courses identities ingest_batches playlist_items playlists role_maps  users > /tmp/avalon.
r5.dump.sql
```

Load to R6 database

SSH into your R6 box and load the R5 database dump. *Remember to make sure your R6 is clean by doing the wipeout command.*

```
# this example assumes R6 database is mysql. Adjust as necessary for your R6 database.
bundle exec rails dbconsole
mysql> source /tmp/avalon.r5.dump.sql
```

If you are a developer wanting to test your migration on sqlite locally...

> Sqllite3 doesn't like mysql dump files so a transformation script needs to be run on it. Download https://raw.githubusercontent.com/dumblob/mysql2sqlite/master/mysql2sqlite, make it executable, and run it on the dump file.

```
chmod +x mysql2sqlite
./mysql2sqlite avalon.r5.dump.sql > avalon.r5.dump.sqlite

#then on r6
bundle exec rails dbconsole
sqlite> .read /tmp/avalon.r5.dump.sqlite
```

".read" returns "memory", which is good.

Migrate Fedora Objects

If migrating from fedora3 on a different host, SSH tunnel or open up your fedora3 server to your new Avalon app.

Choose an available local port to use as a mirror of your fedora3.host:8983 import source # In this example 9999 is used. It must match the port used in fedora3.yml in the next step

```
ssh -L 9999:localhost:8983 user@fedora3.host
```

keep the ssh mirror running in another tab as you do the next steps. Or run it in the background

```
ssh -fN -L 9999:localhost:8983 user@fedora3.host
```

Setup config/fedora3.yml in your Avalon 6 app

The fedora3.yml should work without change, but if you do have trouble, here's an example fedora3.yml for the avalon demo server.

```
development:
  user: fedoraAdmin
  password: <password>
  url: http://localhost:9999/fedora
test:
  user: fedoraAdmin
  password: <password>
  url: http://localhost:9999/fedora
production:
  user: fedoraAdmin
  password: <password>
  url: http://localhost:9999/fedora
```

Configure IDs (optional)

Avalon 6 has adopted Noids for object ids instead of the default Fedora 4 UUIDs. All objects migrated from Avalon 5.x to Avalon 6 will be assigned a new id. The old Avalon 5.x Fedora 3 PID will be preserved using the **prov:wasDerivedFrom** predicate. Avalon uses active_fedora-noid's default template, but this can be overwritten in **config/initializers/active_fedora-noid.rb** using the gem's instructions.

Run the migration!

The repository migration tool has many options but it can be run with default options as shown.

```
bundle exec rake avalon:migrate:repo
```

You can watch it run on your target avalon by going to **<your avalon url>/admin/migration_report .** Go here to find more information on how to read the report and command output. After this command is finished, look through the report and see which items have failed and troubleshoot using the errors listed.

Migrate Database Tables

```
# Run script to map fedora 3 pids in database to newly minted fedora 4 noids
bundle exec rake avalon:migrate:db
```

Delete Failed Bookmarks

```
#This will delete any bookmarks that point to failed items
bundle exec rake avalon:migrate:bookmark_cleanup
```