

Upgrading Avalon 6.0 to Avalon 6.1



For upgrades from Avalon 5 to 6.1, perform the actions in this document after completing the migration steps detailed in [Upgrading Avalon 5.1.6 to Avalon 6.0](#).

See [What's New in Avalon 6.1](#) for more details.

Upgrade Steps

For manual install:

Download and deploy new code from [avalon repo](#)

```
git checkout master
git pull
```

Update gems and DB

```
bundle install
RAILS_ENV=production bundle exec rake db:migrate
```

Restart passenger (if using)

```
passenger-config restart-app
```

Restart Resque (background jobs processor)

```
pkill -f resque
RAILS_ENV=production BACKGROUND=yes bundle exec rake resque:scheduler
RAILS_ENV=production BACKGROUND=yes QUEUE=* bundle exec rake resque:work
```

For Docker install:

For users who installed Avalon using our [docker-compose script](#)

Fetch new script

```
git checkout master
git pull
```

Download new images and update the containers

```
docker-compose pull
docker-compose up
```

Upgrade to using Database-backed Noid ID Minter

Avalon 6.1 switched to using a database-backed noid id minter by default since it is more robust and recommended by the `active_fedora-noid` gem.

Set up the DB for minter state

```
bundle exec rake db:migrate
```

Backup your minter state file (default location is `/tmp/minter-state`). Skip if you do not have one.

```
cp /tmp/minter-state /tmp/minter-state.bak
bundle exec rake active_fedora:noid:migrate:yaml_to_marshall
bundle exec rake active_fedora:noid:file_to_database
# Remove the minter state file
rm /tmp/minter-state
```

If the following command runs without any output then it was successful

```
bundle exec rails r "MediaObject.new.assign_id"
```

Post-Upgrade steps

```
# Add a Fedora4 record identifier to existing MODS record
bundle exec rake avalon:migrate:record_identifier
```