

# AtoM's Command Line Tasks

```
> an introduction
```

[illegible]

# Documentation

For a detailed outline of how to use the tools, and the options that each task has available.

[Home](#)[Downloads](#)[Documentation](#)[Community](#)[News](#)[Wiki](#)[Demo](#)

Version 2.3.1 is our latest release.

## Command line tools

There are a number of command line tools that can help you troubleshoot various AtoM problems.

See below for Common AtoM database queries.

### See also

- Load digital objects via the command line
- CSV import

## Find out what version of AtoM you're running

You can always find out what version of AtoM you have installed via the [user interface](#) by navigating to **Admin > Settings > Global** - the application version is the first thing listed on the Settings page. See: [Application version](#) for more information.

However, if you'd like to check the application version from the command-line, you can run the following command from AtoM's root folder:

```
php symfony tools:get-version
```

## Add a SuperUser (Admin) account

You can create a new **administrator** account from the command-line using the following command:

```
php symfony tools:add-superuser --email="your@email@example.com" --password="mySuperPAssw0rd" --username
```

The username should not have any spaces in it.

If you'd like to quickly create a user for demonstration, diagnostic purposes, you can use the following command:

```
php symfony tools:add-superuser -demo
```

This will create a superuser with username "demo", email "demo@example.com @", and password "demo".

## Promote a user to an administrator

If you'd like to promote a user account in AtoM to have full **administrator** access privileges via the command-line, you can use the following task. You will need to know the user name of the account.

```
php symfony tools:promote-user-to-admin <username>
```

If the user doesn't exist you'll see the following error:

```
Unknown user.
```

If the user is already an administrator:

```
The given user is already an administrator.
```

## Version 2.3.1

[\[Table of contents\]](#)

### CONTENTS

Command line tools

- Find out what version of AtoM you're running
- Add a SuperUser (Admin) account
- Promote a user to an administrator
- Change a password
- Regenerating derivatives
- Re-indexing PDF text
- Rebuild the nested set
- Generate slugs
  - Notes on slugs in AtoM
- Taxonomy normalization
  - Notes on use
- Update the publication status of a description
  - Notes on use
- Delete a description
- Delete all draft descriptions
- Remove HTML content from archival description fields
- Purging all data
- Backing up the database
  - Using the import:bulk command
- Bulk import of XML files
  - Using the export:bulk command
- Bulk export of XML files
  - Using the export:bulk command
  - Exporting EAC-CPF XML for authority records
- Generate an XML sitemap for search engine optimization

Common AtoM database queries

- Update all draft archival descriptions to published
- Update all draft archival descriptions from a particular repository to published
- Truncate slugs to maximum character length
- Delete slugs from AtoM

### SEARCH

Open the [general index](#) or type your search in the search box.

### AVAILABLE DOCUMENTS

- Version 2.0
- Version 2.1
- Version 2.2
- Version 2.3.1
- Version 2.4

<https://www.accesstomemory.org/docs/latest/admin-manual/maintenance/cli-tools/>



# Documentation

Our Troubleshooting page is the first place to check if you encounter an issue!

@tom

Home Downloads Documentation Community News Wiki Demo

Version 2.5 is our latest release.

## Troubleshooting

If you're running into issues with your AtoM installation, there are several maintenance actions and command-line tasks you might want to try using to see if they resolve the issue, prior to seeking external support. This page will outline some of the most common issues, and how to resolve them.

### Tip

If you are new to working with the Unix command-line, you might find the following slide deck useful to get started:

- <https://www.slideshare.net/accesstomemory/commandline-101>

### Jump to:

- Figure out the nature of the problem
- Error logs and Debug mode
- Restarting services
- Monitoring active processes and checking execution limits
- Running AtoM's most common maintenance tasks
- Dealing with data corruption
- Getting support
- Troubleshooting FAQ

## Figure out the nature of the problem

The first thing you should do is try to isolate exactly what is causing the issue. Specifically:

- What are the exact steps required to reproduce this issue?
- What is the error encountered?
- What is the expected outcome?

Being able to articulate these clearly will often help you figure out what solutions you should try first - and if you can't resolve the issue, you'll need this information to seek further support.

It's also useful to try to rule out any factors that are unrelated to the issue. For example: can you reproduce the error using a different browser? What about a different computer? If your AtoM site is publicly accessible, does it behave differently if accessed from home as opposed to work (i.e. might firewalls, VPNs, or other restrictions be affecting your access)? Is the issue particular to a specific user or group, or does it affect all users? If it's an error related to a task supported by the job scheduler, is it just this one task that isn't working, or are all jobs failing? And so forth.

With a detailed step-by-step guide to reproducing the issue, we can now better diagnose the problem and consider what we can try to resolve it.

## Error logs and Debug mode

It can be useful to see if there is further information available on the nature of the error in your webserver logs, particularly if you have encountered a 500 internal server error:



Oops! An Error Occurred

Sorry, something went wrong.

The server returned a 500 Internal Server Error.

Try again a little later or ask in the [discussion group](#).

## Version 2.5

[\[Table of contents\]](#)

### CONTENTS

#### Troubleshooting

- Figure out the nature of the problem
- Error logs and Debug mode
- Restarting services
  - Restarting PHP-FPM
  - Restarting Nginx
  - Restarting memcached
  - Restarting the job scheduler
- Monitoring active processes and checking execution limits
  - Monitoring active processes with httpd
  - Adjusting PHP execution limits
- Running AtoM's most common maintenance tasks
  - Get information about your search index
  - Populating the search index
  - Clearing the application cache
  - Rebuilding the nested set
  - Generating slugs
- Dealing with data corruption
  - Data corruption in information objects
  - Data corruption in other entities
- Getting support
- Troubleshooting FAQ
  - Why do I get a blank white screen?
  - Why do I get a 500 (Internal Server) error?
  - Why do I get a "Max execution time" exceeded error?
  - Why do I get a "Allowed memory size" exhausted error?
  - Why do I get a "Too many connections" error?
  - Why do I get a "MySQL has gone away" error?
  - Why do I get a "Parent Resource id: 'XXXXX' does not exist" error?
  - Why do I get warnings when populating the search index?
  - Why do I get a SearchPhaseExecutionException when trying to search?
  - Why do I get a 504 Connection timed out error?
  - Why am I seeing strange behavior in the AtoM treeview?
  - Why can't I upload (large) digital objects?
  - Why can't I log into AtoM?
  - What should I do if I get an error that isn't described here?

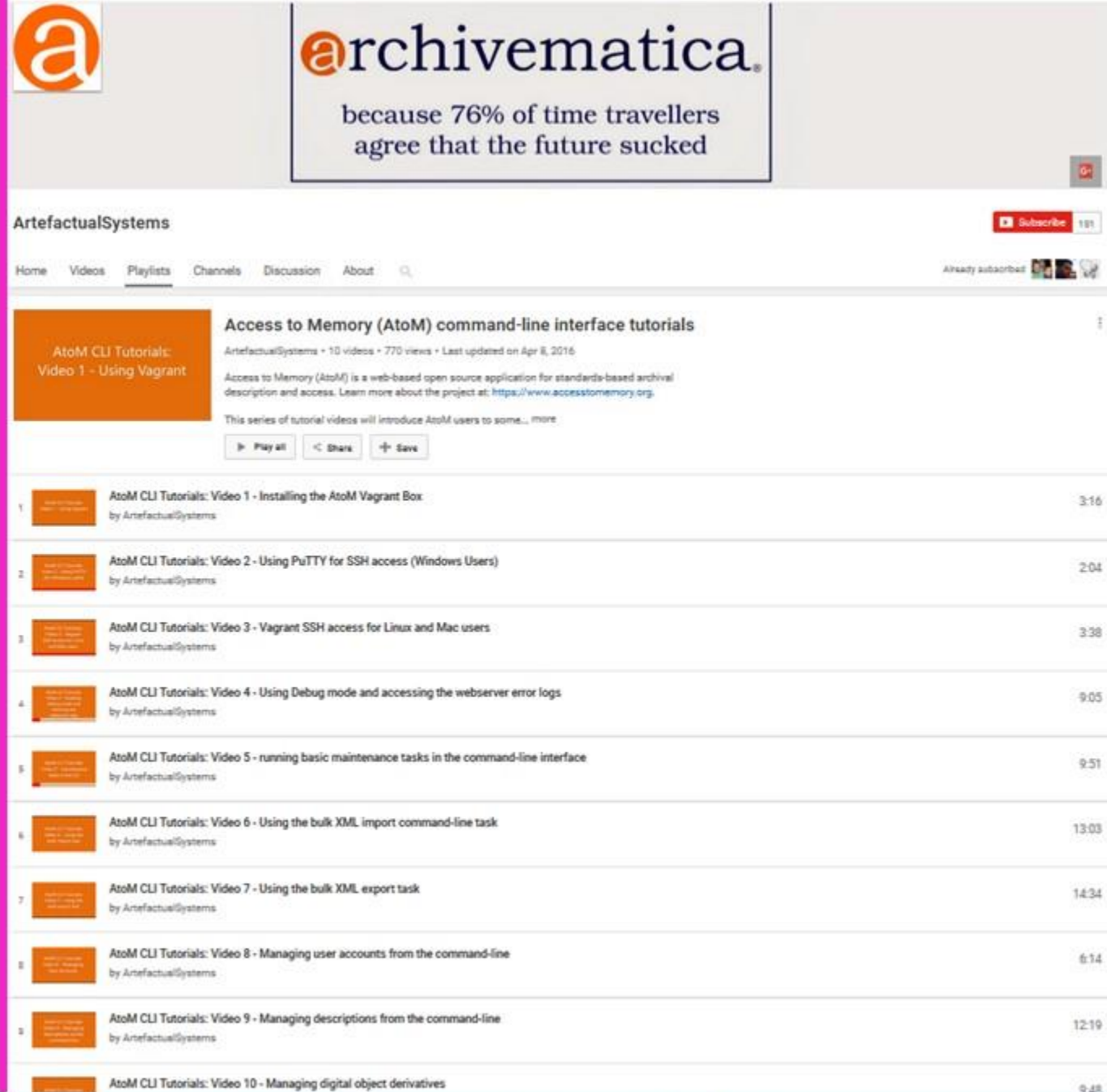
### SEARCH

<https://www.accesstomemory.org/docs/latest/admin-manual/maintenance/troubleshooting>

# Video Tutorials

For a guided demo of the tools in action, with some options shown. 10 videos, starting with configuring your local Vagrant test instance (so you can try them locally on a small data set first).

**NOTE:** commands for restarting services have changed in later versions of Ubuntu! Be sure to consult the documentation for the correct commands – these videos have not been updated accordingly



The screenshot shows the YouTube channel page for ArtefactualSystems. At the top, there is a banner with the 'a' logo and the text 'archivematica. because 76% of time travellers agree that the future sucked'. Below the banner, the channel name 'ArtefactualSystems' is displayed, along with a 'Subscribe' button and a subscriber count of 191. The navigation menu includes 'Home', 'Videos', 'Playlists', 'Channels', 'Discussion', and 'About'. The 'Playlists' tab is selected, showing a playlist titled 'Access to Memory (AtoM) command-line interface tutorials'. The playlist description states: 'ArtefactualSystems • 10 videos • 770 views • Last updated on Apr 8, 2016'. It also mentions that 'Access to Memory (AtoM) is a web-based open source application for standards-based archival description and access' and provides a link to the project website: <https://www.accessmemory.org>. The playlist contains 10 videos, each with a thumbnail and a title. The videos are: 1. AtoM CLI Tutorials: Video 1 - Installing the AtoM Vagrant Box (3:16), 2. AtoM CLI Tutorials: Video 2 - Using PuTTY for SSH access (Windows Users) (2:04), 3. AtoM CLI Tutorials: Video 3 - Vagrant SSH access for Linux and Mac users (3:38), 4. AtoM CLI Tutorials: Video 4 - Using Debug mode and accessing the webserver error logs (9:05), 5. AtoM CLI Tutorials: Video 5 - running basic maintenance tasks in the command-line interface (9:51), 6. AtoM CLI Tutorials: Video 6 - Using the bulk XML import command-line task (13:03), 7. AtoM CLI Tutorials: Video 7 - Using the bulk XML export task (14:34), 8. AtoM CLI Tutorials: Video 8 - Managing user accounts from the command-line (6:14), 9. AtoM CLI Tutorials: Video 9 - Managing descriptions from the command-line (12:19), and 10. AtoM CLI Tutorials: Video 10 - Managing digital object derivatives (9:48).

ArtefactualSystems

Home Videos Playlists Channels Discussion About

Subscribe 191

Already subscribed

**Access to Memory (AtoM) command-line interface tutorials**

ArtefactualSystems • 10 videos • 770 views • Last updated on Apr 8, 2016

Access to Memory (AtoM) is a web-based open source application for standards-based archival description and access. Learn more about the project at: <https://www.accessmemory.org>.

This series of tutorial videos will introduce AtoM users to some... more

Play all < Share + Save

- 1 AtoM CLI Tutorials: Video 1 - Installing the AtoM Vagrant Box by ArtefactualSystems 3:16
- 2 AtoM CLI Tutorials: Video 2 - Using PuTTY for SSH access (Windows Users) by ArtefactualSystems 2:04
- 3 AtoM CLI Tutorials: Video 3 - Vagrant SSH access for Linux and Mac users by ArtefactualSystems 3:38
- 4 AtoM CLI Tutorials: Video 4 - Using Debug mode and accessing the webserver error logs by ArtefactualSystems 9:05
- 5 AtoM CLI Tutorials: Video 5 - running basic maintenance tasks in the command-line interface by ArtefactualSystems 9:51
- 6 AtoM CLI Tutorials: Video 6 - Using the bulk XML import command-line task by ArtefactualSystems 13:03
- 7 AtoM CLI Tutorials: Video 7 - Using the bulk XML export task by ArtefactualSystems 14:34
- 8 AtoM CLI Tutorials: Video 8 - Managing user accounts from the command-line by ArtefactualSystems 6:14
- 9 AtoM CLI Tutorials: Video 9 - Managing descriptions from the command-line by ArtefactualSystems 12:19
- 10 AtoM CLI Tutorials: Video 10 - Managing digital object derivatives by ArtefactualSystems 9:48

<https://www.youtube.com/playlist?list=PLZiwlG5eSMeyeETe15EsEBSu5htPLK-wm>



# What do we use these CLI tasks for?

## Maintenance

- Installation and upgrades
- Repopulating the search index
- Regenerating derivatives

## Import and Export

- Bulk import of XML files
- Importing Authority record CSV files with Alias and Relationship CSVs
- Using options available in the CLI not in the user interface

## Troubleshooting

- Checking the error logs
- Restarting services
- Trying tasks that resolve common issues

## Doing things quicker

- Deleting large descriptions
- Purging all data
- Promoting a user account to Admin
- Checking your AtoM version

# Why should I learn this?

## Understanding how AtoM works... and how it breaks

- Some actions in the UI, if they fail, can be easily resolved with 1 or 2 tasks. Understanding the why and how will help you use AtoM better

## Better communication with IT

- Provide direct suggestions
- Understand what IT is doing
- Shared language

## Get better community support

- Provide error logs with Forum posts
- Let us know what you've tried
- Save back and forth, get better responses

## Learning the CLI is fun!

- Helps with other applications
- Become more confident on Linux O/S computers
- Pretend you are a hacker :P



# Tips for Getting Started

## AtoM CLI tasks are run from the root AtoM directory

This means the directory where you have AtoM installed in your current environment.

In production installations – if you have followed our [Recommended installation instructions](#), this is:

```
> /usr/share/nginx/atom
```

In the [Vagrant box](#), this is:

```
> /home/vagrant/atom
```

# CLI 101 Reminders

Autocomplete a directory/location/filename	> <code>TAB</code>
Show last command	> <code>UP ARROW</code>
Abort a process	> <code>CTRL+C</code>
Change to a directory	> <code>cd &lt;path/to/dir&gt;</code>
Go up one directory	> <code>cd ..</code>
Go to home directory	> <code>cd</code>
Clear current commands out of view	> <code>clear</code>

There are some great cheat sheets and tutorials out there – a simple internet search will uncover many options



# Tips for Getting Started

## CLI tasks often have help text built-in

AtoM command-line tasks generally have help text that will show the expected syntax, give a description, and display the available optional parameters that can be used

```
> php symfony help <command>
```

```
[~/atom] (qa/2.4.x)
vagrant$ php symfony help search:populate
Usage:
  symfony search:populate [--application[="..."]] [--env="..."] [--exclude-types[="..."]]
  [--show-types]

Options:
  --application      The application name (default: qubit)
  --env              The environment (default: cli)
  --exclude-types    Exclude document type(s) (command-separated) from indexing
  --show-types       Show available document type(s), that can be excluded, before indexing

Description:
  The search:populate task empties, populates, and optimizes the index
  in the current project. It may take quite a while to run.

  To exclude a document type, use the --exclude-types option. For example:

    php symfony search:populate --exclude-types="term,actor"

  To see a list of available document types that can be excluded use the --show-types option.
```

An example of running the help command with the search populate task: `php symfony help search:populate`

# Tips for Getting Started

There are 3 options you might see in some commands that you should **not** use

The `--application`, `--env`, and `--connection` options are part of the Symfony framework with which AtoM was built. AtoM relies on using their defaults to execute the command-line tasks properly. Ignore them!

```
[~/atom] (qa/2.4.x)
vagrant$ php symfony help digitalobject:regen-derivatives
Usage:
  symfony digitalobject:regen-derivatives [--application[="..."]] [--env="..."] [--connection="..."] [-l|--slug[="..."]] [-d|--type[="..."]] [-i|--index] [-f|--force] [-o|--only-externals] [-j|--json[="..."]] [--skip-to[="..."]]

Options:
  --application  The application name (default: qubit)
  --env          The environment (default: cli)
  --connection  The connection name (default: propel)
  --slug        (-l) Information object slug
  --type        (-d) Derivative type ("reference" or "thumbnail")
  --index       (-i) Update search index (defaults to false)
  --force       (-f) No confirmation message
  --only-externals (-o) Only external objects
  --json       (-j) Limit regenerating derivatives to IDs in a JSON file
  --skip-to     Skip regenerating derivatives until a certain filename is encountered
```

An example of running the help command with digital object derivative regeneration task:

```
php symfony help digitalobject:regen-derivatives
```

Note the 3 options outlined in pink, which you shouldn't invoke.



# Tips for Getting Started

## Commands are consistently structured

```
> php symfony type:task --options parameters
```

- All CLI tools added to AtoM will start with `php symfony`
- There are several recurring CLI task type namespaces reused
  - Examples include `tools:`, `digitalobject:`, `propel:`, `csv:`, etc.
- The task name tells AtoM what task you actually want to run
- Not all commands have options
- Some options require parameters, such as a filename and path, or other value
  - Example: `--slug="the-jane-doe-fonds"`
  - Example: `--alias-file="/path/to/my/alias-file.csv"`
- Not all commands have parameters
- Usually if there is a parameter at the end of a command, it is a path to a file or directory
  - Example: bulk EAD XML import command, with the index option used
  - `php symfony import:bulk --index /vagrant/my-ead-directory/`

# Repopulating the Search Index

```
> php symfony search:populate
```

## What

AtoM maintains an Elasticsearch search index to provide fast, full-text search results with faceting. It is this index that allows AtoM to find and display records in the user interface when browsing and searching.

Running this task will **delete the current index, then repopulate and optimize the index**. Depending on the number of records in your installation, this can take a while to run - for production sites, we recommend running this task after regular business hours.

## Why

Occasionally it is necessary to repopulate the Elasticsearch index from the primary database, especially after operations that affect many records, such as:

- Importing data (CSV or EAD import)
- Adding a new language to the Language menu
- Moving a large series to a new parent record
- Moving a Fonds or Collection to a different archival institution
- Doing any bulk search and replace operation
- Operations performed in the user interface that time out



# Repopulating the Search Index

## Using the `exclude-types` option

```
> php symfony search:populate --exclude-types="informationobject,term"
```

Keeps the part of the existing index for the listed entities, skipping them (useful if you only need to re-index one type as it is much quicker). In this example we are skipping archival descriptions and terms (e.g. subjects, places, etc).

## Types

```
> php symfony search:populate --show-types
```

Will display the list of available entity types that can be skipped. You can then either cancel the command to re-enter it with the `--exclude-types` option, or continue to run the task and index all types

- `accession`
- `actor` (i.e. authority record)
- `aip` (indexed during a DIP upload from Archivematica)
- `function`
- `informationobject` (i.e. archival description)
- `repository` (i.e. archival institution)
- `term` (such as subjects and places, etc)

# Check the status of the Search Index

```
> php symfony search:status
```

## What

Provides useful information about AtoM's ES index, including:

- Search host
- Port
- Index name
- Document index status for all primary entity types in AtoM (including: Accession, Actor, AIP, Function, Information object, Repository, and Term)

## Why

Useful for troubleshooting! See what is and isn't indexed

```
vagrant$ php symfony search:status
Elasticsearch server information:
- Host: 127.0.0.1
- port: 9200
- Index name: atom

Document indexing status:
- Accession: 4/4
- Actor: 184/184
- Aip: 0/0
- Function: 2/2
- Information object: 464/464
- Repository: 7/7
- Term: 3154/3154
[~/atom] (qa/2.5.x)
```



# Clearing the application cache

```
> php symfony cc
```

## What

The Symfony 1.x framework that AtoM is built upon includes the ability to cache HTML content, for better responsiveness to HTTP requests.

"One of the ways to speed up an application is to store chunks of generated HTML code, or even full pages, for future requests. This technique is known as caching, and it can be managed on the server side and on the client side.

...The principle of HTML caching is simple: Part or all of the HTML code that is sent to a user upon a request can be reused for a similar request. This HTML code is stored in a special place (the cache/ folder in symfony), where the front controller will look for it before executing an action. If a cached version is found, it is sent without executing the action, thus greatly speeding up the process. If no cached version is found, the action is executed, and its result (the view) is stored in the cache folder for future requests." (from the [Symfony docs](#))

## Why

Occasionally, when errors occur or changes have been made, we need to flush the cached HTML so we are being served updated content, and not an outdated version of a web page. Running the **cache:clear** command will empty out the existing application cache so it will become repopulated overtime with updated versions of the web pages in AtoM that are served to you.

# Clearing the application cache

```
> php symfony cc
```

## Important: there are other caches to clear!

**PHP-FPM** (a PHP extension that AtoM uses) can also cache some content - if you are clearing the application cache, you should also consider restarting PHP-FPM (see below, in the "Other maintenance tasks" slides).

**Memcached** is also an external cache engine that can be used with AtoM - you should restart it as well. (see below)

Finally, don't forget that **your web browser** has its own cache - in some cases, if you are not seeing changes take affect, you might want to try clearing your web browser cache. Note that, because AtoM is session based, clearing the web browser cache might log you out of the application. Be sure you have any work you are doing in the user interface saved before doing so.



# Rebuilding the nested set

```
> php symfony propel:build-nested-set
```

## What

AtoM generally uses a relational database to store its data (generally, MySQL). However, relational databases, which are comprised of flat tables, are not particularly suited to handling hierarchical data.

As developer [Mike Hillyer](#) notes, "Hierarchical data has a parent-child relationship that is not naturally represented in a relational database table." One method of addressing this is to employ a "**Nested set model**" ([Wikipedia](#)).

AtoM makes use of a nested set to manage hierarchical relationships, such as between parent and child [terms](#) and [descriptions](#).

## Why

Sometimes, during operations that involve updates to large hierarchies, the nested set can become corrupted - especially if the server times out during an operation that reaches the execution limit settings. This task will rebuild all nested sets in AtoM, which often resolves issues with things like the treeview showing duplicate records, etc.

# Generating slugs

```
> php symfony propel:generate-slugs
```

## What

A slug is a word or sequence of words which make up the last part of a URL in AtoM. It is the part of the URL that uniquely identifies the resource and often is indicative of the name or title of the page (e.g.: in [www.yourwebpage.com/about](http://www.yourwebpage.com/about), the slug is [about](#)). The slug is meant to provide a unique, human-readable, permanent link to a resource.

Slugs in AtoM are sanitized so they do not contain any special characters or spaces. Some slugs are arbitrary (like for user account pages), while many others are based on the record's name or title (like archival descriptions, authority records, etc). When a slug is already in use (for example, 2 records named "Correspondence"), AtoM will automatically append a number to the end of the second slug to make it unique (e.g. [correspondence-2](#))

## Why

In some cases, AtoM may time out in the middle of an operation that involves the creation of new records. In such cases, it is possible that AtoM has died after creating an information object, but before having a chance to create a slug for the record. This can cause unexpected errors in the application - most notably, 500 errors when trying to access the records missing slugs through the application interface. This task will generate new slugs for any that are missing them in the database.



# Regenerating all slugs

```
> php symfony propel:generate-slugs --delete
```

Note that by default, existing slugs will not be **replaced**. If you want to generate new slugs for existing objects, you will need to first delete the existing slugs from the database.

This can be useful for records in which a random slug has been automatically assigned, because the default user data used to generate the slug has not been provided (for example, creating an authority record with no authorized form of name entered).

Using the **--delete** option will first delete all exist slugs before regenerating them all.

## Generating slugs works for the following entities:

- information objects (e.g. archival descriptions)
- actors (e.g. authority records)
- terms
- physical objects (e.g. storage locations, etc)
- events (e.g. creation events, etc - usually the relationship between actors and information objects)
- accessions

If an error has left other areas in AtoM without slugs (for example, a user account's view page, a donor record, etc), **this task will not resolve the issue** - you will likely have to manually insert a slug into the database for that entity.



# Regenerating derivatives

```
> php symfony digitalobject:regen-derivatives
```

## What

When you upload digital objects to AtoM (such as images, audio or video files, PDFs, etc), AtoM will automatically generate 2 derivative copies for use in the application: a **reference display copy** (used on the view page of a description with a digital object attached), and a **thumbnail** (for use in search and browse results, the digital object carousel, etc). The original is called the **master digital object**.

When you link to an external digital object on the web instead, AtoM will generate the local derivatives, but instead of storing the master locally, will simply point to the web resource.

This command-line task will purge all existing derivatives in AtoM and regenerate new ones from the master digital objects. It can take a while to run - for production sites, we recommend running it after business hours.

## Why

Sometimes the derivatives will fail to load (especially during an upload or import that times out), or you have manually changed some and want to reset them all. If you are upgrading from ICA-AtoM, the derivatives are set to be a different default size, so after an upgrade, derivatives may appear blurry or pixelated. Alternatively, if you have changed the Digital object derivatives settings, you might want to regenerate your derivatives so that the new setting is used for multi-page content such as PDF derivatives.



# Regenerating derivatives

```
> php symfony help digitalobject:regen-derivatives
```

Only regen derivatives for one description

```
> --slug="<slug>"
```

Repopulate the search index as the task runs

```
> --index or -i
```

Force - no yes/no confirmation message first

```
> --force or -f
```

Only regenerate externally linked objects

```
> --only-externals
```

Skip ahead and start at a certain filename


```
> --skip-to="<file>"
```

**Example** - force start, index as you proceed, but skip ahead and start at the file silly-rabbit.jpg:

```
> php symfony digitalobject:regen-derivatives -f -i --skip-to="silly-rabbit.jpg"
```

# Check your AtoM version

```
> php symfony tools:get-version
```



```
vagrant$ php symfony tools:get-version  
2.4.0 v150  
[~/atom] (qa/2.4.x)  
vagrant$
```

## What

This simple command will output the version of AtoM you are using (e.g. 2.1; 2.3.1, etc) as well as the current database version (generally a 3 digit number - for example release 2.3 was at database version 138).

Every time new development is added to AtoM that requires changes to the database, we include a database schema migration to the next version.

If your database version is incorrect, **back up your data**, and then you can try running the **sql-upgrade** task to migrate to the latest database schema:

```
> php symfony tools:upgrade-sql
```

## Why

Sometimes when pulling in new changes to your AtoM instance (for example, updating your Vagrant box), the database may need to be migrated to the proper version - this doesn't happen automatically(!), and is part of the upgrade instructions for tarball installations (from our [Downloads](#) page) as well. Knowing the version can be useful information to share when seeking support - if it isn't correct, you can run the **sql-upgrade** task to migrate to the latest database schema. Warning: make sure you back up your data first!!!



# Create an Admin account

```
> php symfony tools:add-superuser <username>
```

## What

This command will create a new superuser (i.e. an Administrator account) in AtoM.

The command requires a minimum parameter of the new username. AtoM will then prompt you for an email and a new password.

However, you can also pass the email and the password in a single command as options, as demonstrated below:

## Why

If you forget your password and username, this can be an easy way to get back into the application - you can then find your old account and either delete it, or update the password then delete this one.

It's also quicker than using the user interface to create a new account :)

```
> php symfony tools:add-superuser --email="email@example.com" --password="MYSUPERPASSWORD" <username>
```

# Other maintenance tasks

## Restarting services

- Restarting PHP-FPM
- Restarting your webserver
- Restarting memcached
- Restarting the job scheduler

The following commands will help restart some of the dependencies that AtoM uses. These tasks are **not** managed via Symfony or AtoM - instead, they depend on your particular installation environment. The following instructions assume you have followed our [recommended installation instructions](#) - if you have made changes, some of the commands may be different for you!

Restarting services can be a useful first step in trying to resolve issues - if the service that AtoM uses is in a bad state as a result of an error that has occurred, then often restarting it can return it to a condition known to be working.



# Other maintenance tasks

## Restarting PHP-FPM

On Ubuntu 16.04 (with PHP 7.0)

```
> sudo systemctl restart php7.0-fpm
```

On Ubuntu 18.04 (with PHP 7.2)

```
> sudo systemctl restart php7.2-fpm
```

PHP-FPM is a PHP extension that allows for better interaction with an application's web server, via the use and configuration of FastCGI pools in PHP.

Learn more: <https://php-fpm.org/about/>

FastCGI is a binary protocol for interfacing interactive programs with a web server. FastCGI is a variation on the earlier Common Gateway Interface (CGI); FastCGI's main aim is to reduce the overhead associated with interfacing the web server and CGI programs, allowing a server to handle more web page requests at once.

From Wikipedia: <https://en.wikipedia.org/wiki/FastCGI>

# Other maintenance tasks

## Restarting memcached

On Ubuntu 14.04 (with PHP 5.x)

```
> sudo service memcached restart
```

On Ubuntu 16.04 / 18.04

```
> sudo systemctl restart memcached
```

Memcached is an open-source, general purpose distributed memory object caching system. It is often used to speed up dynamic database-driven websites by caching data and objects in memory to reduce the number of times an external data source (such as a database) must be read.

Read more: <https://www.memcached.org/>



# Other maintenance tasks

## Restarting Nginx

On Ubuntu 14.04 (with PHP 5.x)

```
> sudo service nginx restart
```

On Ubuntu 16.04 / 18.04

```
> sudo systemctl restart nginx
```

**Nginx** (pronounced "engine x") is an open source HTTP web server (among other things). It manages the interaction between your web browser and AtoM. Your browser requests a page in AtoM, and this is sent to Nginx, which communicates with AtoM to retrieve and serve the page, or else return an error message.

Learn more:

- <http://nginx.org/en/>
- <https://www.nginx.com/resources/glossary/nginx/>

# Other maintenance tasks

## Restarting the job scheduler

On Ubuntu 14.04 (with PHP 5.x)

```
> sudo restart atom-worker
```

On Ubuntu 16.04 / 18.04

```
> sudo systemctl restart atom-worker
```

AtoM relies a **job scheduler** in order to execute certain long-running tasks asynchronously in the background (instead of synchronously via your web browser, making you wait until the task is done and the page loaded before continuing), to guarantee that web requests are handled promptly and work loads can be distributed across multiple machines.

We use **Gearman** as our job scheduler in AtoM. "Gearman provides a generic application framework to farm out work to other machines or processes that are better suited to do the work. It allows you to do work in parallel, to load balance processing, and to call functions between languages."

Read more: <http://gearman.org/>



# Other maintenance tasks

## Checking the Nginx error logs

```
> sudo tail -f /var/log/nginx/error.log
```

Useful when you encounter 500 errors,  
to get more information



Oops! An Error Occurred

Sorry, something went wrong.

The server returned a **500 Internal Server Error**.

Try again a little later or ask in the [discussion group](#).

[Back to previous page](#).

If you have encountered an error in AtoM and none of the previous maintenance tasks have helped to resolve the issue, it can be useful to get more specific information from the Nginx error logs.

If you run into a 500 error and want to seek advice and basic support from the [AtoM User Forum](#), check the error logs first, and include any relevant error message with your post - it will help other users better understand the issue and be able to offer suggestions on how best to resolve the problem.

# There's plenty more!

## Explore the AtoM documentation and other resources

- AtoM Documentation: <https://www.accesstomemory.org/docs/latest/>
- AtoM User forum: <https://groups.google.com/forum/#!forum/ica-atom-users>
- AtoM wiki: <https://wiki.accesstomemory.org/>

**Questions?**

info@artefactual.com

