

# LenyaProperties

## Introduction

Lenya and your publications can be configured with properties files. This can be useful for setting global, deployment and publication specific parameter values like database usernames and passwords, SMTP server hostnames and the like.

## Setting properties

You can set properties in two different ways:

- using Java system properties on the command line
- in properties files

### Properties files

Properties files can be found in the following places:

- Lenya properties at `$LENYA_HOME/lenya.properties.xml` and `$LENYA_HOME/local.lenya.properties.xml`
- Module properties at `$LENYA_HOME/lenya/modules/<modulename>/lenya.properties.xml`
- Publication properties at `$PUB_HOME/lenya.properties.xml`

### Order of precedence

Properties with the same name (including prefix) and located in different properties files can override each other. The order of precedence is (lower number wins):

1. Publication
2. System properties
3. Lenya local (`$LENYA_HOME/local.lenya.properties.xml`)
4. Module
5. Lenya

The publication properties files are aggregate-fallback enabled. This means that you can override properties of a template in a instance publication of that template.

Module properties are visible to all publications, regardless of the publication using a particular module or not.

System properties should probably override publication properties. At the moment they don't.

### Name prefixes

Property names should use name prefixes to avoid confusion and accidental overriding. The following name prefixes are suggested:

- Lenya properties: `lenya.<propertyname>`
- Module properties: `<modulename>.<propertyname>`
- Publication properties: `<pubname>.<propertyname>`
- System properties: `system.<propertyname>`

If you want to override a property, the fully qualified property name of the target property including prefix has to be used.

System properties are normally used to override existing Lenya or module properties. Publication properties cannot be overridden by system properties at the moment.

## Using properties

The properties are accessible by an input module. They can be used in sitemaps wherever the usage of an input module is appropriate. For example:

```
<map:pipeline type="...">
  <map:match pattern="...">
    <map:generate src="..." />
    <map:transform src="...">
      <map:parameter name="author" value="{properties:pubs.default.author}" />
    </map:transform>
  ...
</map:pipeline>
```

Properties can also be used in the Cocoon configuration for the `EmailNotifier` class. The Cocoon `.xconf` file would look like this:

```
<?xml version="1.0"?>
<xconf xpath="/cocoon" remove="/cocoon/component[@role = 'org.apache.lenya.notification.Notifier']">
  <component logger="lenya.notification"
    role="org.apache.lenya.notification.Notifier"
    class="org.apache.lenya.inbox.InboxNotifier"/>

  <!-- Use the following configuration to send e-mail notifications -->
  <component logger="lenya.notification"
    role="org.apache.lenya.notification.Notifier"
    class="org.apache.lenya.notification.EmailNotifier">
    <smtp host="{pubname.mail.smtp}" username="" password="" />
  </component>
</xconf>
```

The patch in bug 42989 would have to be applied for this feature to work, though. Other Cocoon components do not support this feature (yet).