The properties Module

Table of contents

1 Introduction

The `properties` module provides a service (InputModule) to configure publications using properties. The idea is to have a fallback of properties.

2 Sample

```
<properties>
  <property name="pubs.default.author" value="Miguel de Cervantes Saavedra"/>
</properties>
```

3 Order of precedence

> **Note:**
> The first values set get precedence, as in AntProperties

1. Publication (lazy loaded in loadPublicationPropertiesIfNotDone())
2. Lenya local
3. Modules (all modules, not only the ones referenced in the publication)
4. Lenya

4 Locations

1. {pubDir}/lenya.properties.xml
2. $LENYA_HOME/src/webapp/local.lenya.properties.xml
3. {module}/lenya.properties.xml
4. $LENYA_HOME/src/webapp/lenya.properties.xml

5 Usage

To activate your properties you need to save the above sample file in one of the above locations.

1. Publication - on a publication level you are able to implement all properties and override all possible default ones.
2. Lenya local - use this for global properties for all your lenya based pubs
3. Modules - modules can use properties to allow user to override module specific properties.
4. Lenya - global default properties

5.1 sitemap.xmap

Since the properties module is a cocoon input module, you can use it as any other input module.

```
...
<map:transform src="cocoon:/getDoctypeXSLT/{4}">
  <map:parameter name="author" value="{properties:pubs.default.author}"/>
  ...
</map:transform>
```

5.2 java

In your custom cocoon components use like follows.

```
InputModule inputModule = null;
ServiceSelector selector = null;
String propKey = "pubs.default.author";
  try {
     selector = (ServiceSelector) m_manager.lookup(InputModule.ROLE + "Selector");
     inputModule = (InputModule) selector.select("properties");
     String propValue = (String) inputModule.getAttribute(propKey, null,
                   objectModel)
...
```