



Inicio

Q Buscar en RODERIC

  
 Búsqueda avanzada

Acerca de RODERIC

- [Qué es RODERIC](#)
- [Políticas RODERIC](#)
- [Derechos de autor](#)
- [Memorias](#)
- [Projecte Cèsar](#)

Àrea de Usuaris

- [Depositar documentos](#)
- [Formulario de registro](#)
- [Perfiles RODERIC](#)

Recursos

- [Para bibliotecarios](#)
- [Para investigadores](#)



Cumplimos 10 años!!

2018-09-30

Cumplimos una década. El 30 de Septiembre de 2008 el Consejo de Gobierno de la Universitat firmaba el acuerdo por el que se sumaba a la Declaración de Berlín sobre Acceso Abierto y en consecuencia creaba el repositorio institucional RODERIC.

RODERIC ha crecido hasta posicionarse entre los diez repositorios con más contenidos entre las universidades españolas, según datos de REBIUN. Si nos fijamos exclusivamente en contenidos de investigación, que constituyen el objeto de ser de un repositorio institucional, RODERIC se sitúa en cuarto lugar, siendo el único de los cuatro que ofrece contenidos exclusivamente a texto completo y 100% de acceso abierto.

# RODERIC. Dspace at University of Valencia

Technical aspects

# Table of contents

---

- System and application configuration
  - System specifications
  - Main Dspace configuration
  - Database Configuration
  - File storage
  - Logging
  - Indexing configuration (SolR)
- Repository configuration
  - Communities, subcommunities, collections and items
  - Visibility and embargo
  - Submission process
  - Web interface
  - Controlled vocabularies
  - Localization
  - Other issues: Java performance, development, updates, integration...

# Main Dspace Configuration

---

- Dspace is composed by a set of web applications in a Tomcat servlet container, a core application based in Java, and a database in a RDMS (Postgresql or Oracle). It is open source and coded in Java.
- You need tools like Ant and Maven if you want to change some functionality, but you just need to know XML, CSS, and understand some config files for configuration and customization tasks.
- Main configuration is located in [dspace]/config/dspace.cfg
- Module configuration is located in [dspace]/config/modules/, for example LDAP authentication, SOLR Indexes, etc
- We are currently using 4.1 version in production. The last version of Dspace is 6.3. (<https://duraspace.org>)

# System specifications

---

- Our system currently has 50.000 items (and growing up), so it should be considered as a Mid-Range Dspace Production System.
- According to Dspace Recommendations, the required system should have those features:
  - 4 GB of RAM (2 for Tomcat and 2 for Database), plus the Memory required by the Operating System, so the minimum amount of memory you need is 8GB.
  - 200 GB of Storage Space. But it depends on the nature of your documents. We have 250GB in our Dspace System and 2 TB in an external systems than contains images of old books pages, linked from DSpace. You can add Network Attached Storage or connect to an Storage Area Network depending on your needs.

# Database Configuration

---

- You have to set the name of DBMS, JDBC connection URL, Driver and user/password
- `db.name = postgres`
- `db.url = jdbc:postgresql://localhost:5432/dspace`
- `db.driver = org.postgresql.Driver`
- `db.username = username`
- `db.password = password`

# Dspace File Storage

---

- DSpace storage is escalable, just adding new volumes to the assets store parameter in dspace.cfg config you can use a new space to storage new objects. You have to configure the asset store for incoming objects.
- `assetstore.dir = [dspace]/assetstore`
- `assetstore.dir.1 = /second/assetstore`
- `assetstore.dir.2 = /third/assetstore`
- `assetstore.incoming= 2`

# Logging configuration

---

- Logging in Dspace is based in log4j logging system, commonly used in Java applications.

- Log4j configuration is located in

`log.init.config = [dspace]/config/log4j.properties`

- Log location:

`log.dir = [dspace]/log`

# Indexing Configuration (SOLR)

---

- Indexing is maintained by Apache SOLR, that have three cores:
  - Search. Indexing for search and browse operations. Including Discovery indexes.  
<http://roderic.uv.es/handle/10550/53>
  - Statistics. Mantains usage statistics:  
<http://roderic.uv.es/handle/10550/64451/statistics>
  - Oai: serves records using OAI-PMH Protocol.  
<http://roderic.uv.es/oai/request?verb=Identify>

Solr configuration is held in [dspace]/solr/ and [dspace]/config/modules/solr-statistics.cfg



# Communities, subcommunities, collections, items

---

- DSpace is organized in a hierarchical way: At the top levels we have communities, that may have inside subcommunities or not, and below we have collections.
- Items can only be submitted to collections, the last level in the hierarchy.
- An item can be in a collection and mapped to another collection.

<http://roderic.uv.es/handle/10550/44759>

- Every component of Dspace (item, collection or community) receive a unique and persistent identifier named handle, formed by a prefix (10550 at the UV) and a private identifier. You can get to any object by using the repository URL or the Handle URL:

<https://hdl.handle.net/10550/44759> [Handle.net](https://hdl.handle.net/) resolves the Unique URL to your repository URL. You have to register and pay to Handle.net (50\$ a year).

# Item visibility and embargo

---

- All our items are public, although the full text file, the bitstream, can be embargoed until the date the authors decides. The configuration is set using the following parameters :

`embargo.field.terms = dc.embargo.terms`

`embargo.field.lift = dc.embargo.liftdate`

`plugin.single.org.dspace.embargo.EmbargoSetter =  
org.dspace.embargo.DayTableEmbargoSetter`

`embargo.terms.days = 0 days:0, 1 month:30, 3 months:90, 6 months:180, 1 year:365`

- After configuring this, we have to set the embargo period in the submit form.

# Submission process

---

- Autosubmit: used by PhD authors. Their thesis are not in any other place before to be in our repository. We can set the forms in [dspace]/config/input-forms.xml
- Bulk import: used to ingest massively items that have been collected from other sources and have been transformed. We use Dspace command line utilities like [dspace]/bin/dspace import to deposit several items, each item has a folder with metadata in METS XML format and full text archive.
- SWORD Deposit. Used to integrate with other systems, like our Current Research Information System (CRIS). They send a zip package from there in a standard format and it is automatically ingested into Dspace. SWORD is a client-server protocol: we implement the Server and the CRIS the client. It uses https protocol.

# Web Interface (XMLUI)

---

XMLUI is a web application you can use as a web interface to your repository, other is JSPUI, but we don't use them nowadays. XMLUI es more powerful and flexible than JSPUI.

Dspace generates data in Dspace Intermediate Metadata (DIM) XML format. XSLT transforms them into HTML and CSS gives them the final look.

<https://roderic.uv.es:8443/metadata/handle/10550/67811/mets.xml>

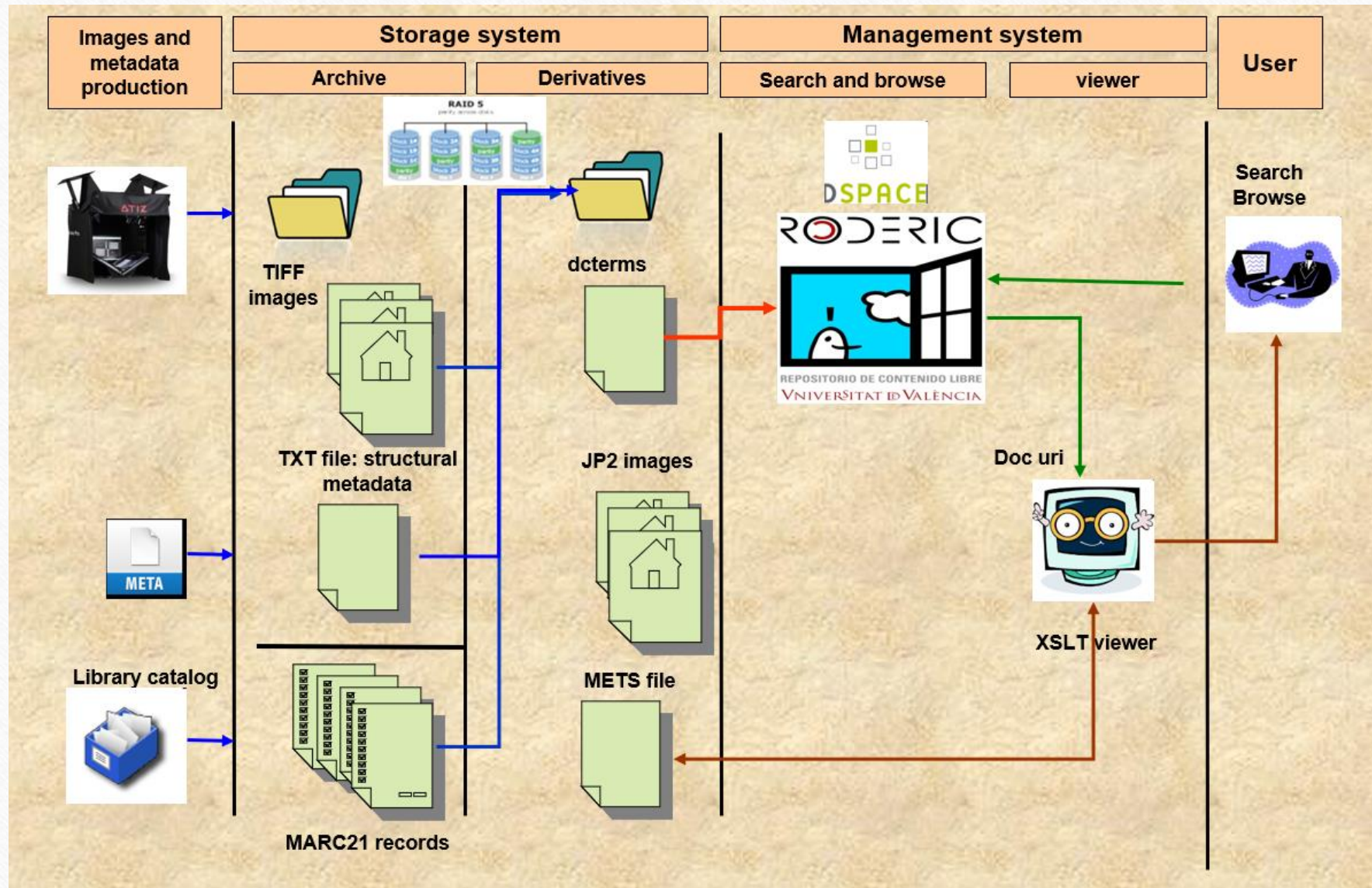
+ XSLT + CSS =

<https://roderic.uv.es:8443/handle/10550/67811>

Customization is based in “themes”, that are a set of XSLT files, images, CSS, etc.

There are several prebuild themes in the directory [dspace]/webapps/xmlui/themes/

You can select one and apply it to the full site, a collection, an item, etc. configuring it into [dspace]/config/xmlui.xconf at the <themes> section.



# Controlled vocabularies

---

- You can use controlled vocabularies as values of an input field in an input form. Previously you have to define that vocabulary in XML format at:

[dspace]/config/controlled-vocabularies

- There are several examples in that directory.
- If you want to use a vocabulary you have to include `<vocabulary>yourvocabulary</vocabulary>` into the `<field></field>` section of your input form file.

# Localization (i18n)

---

- DSpace is localizable.
- Translated texts must be defined in:  
`[dspace]/webapps/xmlui/i18n/messages_XX.xml`
- Where `XX` are the locale initials. Check if your language is included in DSpace distribution and, if not, you should create it.
- You can set locales in `dspace.cfg` changing `webui.supported.locales = a, b, c` and `default locale = a` directives, to set the supported locales (a,b,c) and the default locale 'a'. These files must exist before.

# Development: Install, update, Maven + ant

---

There is a source code directory where you can modify the Java code.

Then you should build it using Maven and finally deploy it using ant

We have a bash script that performs those tasks, basically:

```
cd [source_directory]
mvn -U clean package
cd [source_directory]/dspace/target/dspace-<version>-build/
ant update
/etc/init.d/tomcat restart
```

**NOTE:** every config change must be done into the source code before the build process. Otherwise all those changes will be lost in the live environment.



# Performance: Java Virtual Machine

---

There are hundreds of books about it. Basically you should set:

- `Xmx`: max size of the heap
- `-Xms`: initial size of the heap: it's a good idea to be the same as `Xmx`
- `--XX:MaxPermSize`: maximum memory size, usually half of the max heap.
- And the garbage collector policy.

You can set all of them adding

```
JAVA_OPTS="-Djava.awt.headless=true -Xmx2048m -Xms2048m -XX:MaxPermSize=1024m -XX:+UseParalellGC "
```

to the startup Tomcat script `/etc/init.d/tomcat<version>`

# Reindexing communities and collections

---

- In order to have our repository updated and optimized, we reindex our collections every night.
- Our repository content is huge, so totally deleting indexes and reindexing it everynight overlaps working hours. That's why we alternate different parts, for example one day research production, the next day old books, etc.
- We delete partially Solr indexes and then reindex them, not all of them the same day, ones in pairs days and the other ones in odd days.

[curl](#)

<http://roderic.uv.es/solr/search/update?stream.body=<update><delete><query>location.comm:commid</query></delete><commit/></update>>

`[dSPACE]/bin/dSPACE index-discovery`

# Integration

---

- DSpace supports Standard Dublin Core metadata schema and can interact with other systems working as a data provider or data consumer, using several protocols:
  - OAI-PMH: OAI is provided by the system as service (<http://roderic.uv.es/oai/request?verb=Identify>) and client (through the haversting function)
  - SWORD: Simple Web Service Offering Repository Diposit. Normally used as a server, DSpace receive content packages from other sources though SWORD Protocol.

Thank you for your attention

Roderic: [roderic@uv.es](mailto:roderic@uv.es)