

Custom presentation & dynamic components

- Custom JSPs
- CRIS Components
 - Relationship definition
 - Relation Preference Management
 - Relation Clustering
- CRIS Processors
- Customize the Home page
- Defining the site sections (Explore)
- Defining a list layout

Custom JSPs

To customize the presentation of the data of a CRIS entity other than rearranging the order of the data inside a box and deciding positioning, see “[Group Metadata, aka boxes](#)”, the system allow you to define your custom JSPs used instead of the out-of-box layout.

You can use custom JSPs for example to add javascripts both in the visualization mode than in edit mode, show content retrieved from external systems, add static or dynamic content implementing the components or processors extension points (see below).

In this case, even if the configuration is edited via UI, the layout does not change, because it is defined by the JSP fragment.

CRIS Components

The system allows extension through the use of component that can extends the presentation, tools and functionalities available for any entities.

The components are registered in the `[installDir]/config/spring/cris/cris-components.xml` spring file. For any CRIS entity type has been defined a service where that manage the components, these services need to be named `rpComponentsService`, `pjComponentsService`, `ouComponentsService` and `doComponentsService`. The first two letters of the service name refer to the short name of the related CRIS Entity: ResearcherPage (rp), Project (pj), OrgUnit (ou), Dynamic Object (do).

The snippet below show the default configuration for the Researcher Page components Service

```
<!-- Components -->
<bean id="rpComponentsService" class="org.dspace.app.cris.integration.CrisComponentsService">
  <property name="components">
    <map>
      <entry key="dspaceitems" value-ref="publicationlist" />
      <entry key="projects" value-ref="projectlist" />
    </map>
  </property>
</bean>
```

For every components, a box with the same name and a custom JSP file must be defined. Placing such box in one or more Tab will enable the component execution and visualization.

Components are evaluated (executed) only when a tab including its associated box is requested. If the components is not associated to any box or the box is not included in a visualized tab the code used by the component is not executed and the additional information retrieved by the components are not available. If you need to extend the visualization of a CRIS entity and you need some extra code to be always executed you can use the processors (see below)

Out-of-box the system include configurable components to explore inverse relationship between a CRIS object and DSpace items and a CRIS object and other CRIS objects. This approach other than be time saving make also the data model more consistent avoiding the creation of wrong unilateral relationship. For example: if a researcher belongs to a department it has to be set up a pointer field via UI (see “Field Definition”). However, the OrgUnit entity not need a pointer to track the relationships between the department and all its researchers. The component executes the appropriate SOLR query to retrieve the referencing objects. The snippet below show the configuration of the publicationList component for the ResearcherPage

```

<!-- Publication component for RP -->
<bean id="publicationlist"
  class="org.dspace.app.webui.cris.components.ItemsConfigurerComponent">
  <property name="relationConfiguration" ref="relationRPVSIItemConfiguration" />
  <property name="commonFilter">
    <util:constant static-field="org.dspace.app.webui.cris.util.RelationPreferenceUtil.HIDDEN_FILTER"/>
  </property>
  <property name="target" value="org.dspace.app.cris.model.ResearcherPage" />
  <property name="types">
    <map>
      <entry key="selected" value-ref="selectedComponent" />
      <entry key="journal" value-ref="journalComponent" />
      <entry key="conference" value-ref="conferenceComponent" />
      <entry key="book" value-ref="bookComponent" />
      <entry key="patent" value-ref="patentComponent" />
      <entry key="other" value-ref="otherComponent" />
    </map>
  </property>
</bean>

```

org.dspace.app.webui.cris.components.ItemsConfigurerComponent is the implementation of the abstract class *org.dspace.app.webui.cris.components.ASolrConfigurerComponent* *<T extends DSpaceObject, IBC extends ICrisBeanComponent>* able to manage relationship between CRIS objects and DSpace Items. The following class provide similar functionalities for the other kinds of relationship

- *org.dspace.app.webui.cris.components.CRISDOConfigurerComponent* able to explore relationship with Dynamic Object
- *org.dspace.app.webui.cris.components.CRISOUConfigurerComponent* able to explore relationship with OrgUnit
- *org.dspace.app.webui.cris.components.CRISPJConfigurerComponent* able to explore relationship with Project
- *org.dspace.app.webui.cris.components.CRISRPCConfigurerComponent* able to explore relationship with ResearcherPage

The following attributes define the behavior of such classes

- **relationConfiguration** defines a relation between two Entity Class (CRIS Objects or DSpace Item), see the dedicated paragraph for further details
- **commonFilter** defines an additional SOLR filter query to apply to the relationship
- **target** must match the class canonical name of the target Entity in the relationship, i.e. if the relationship is Item – ResearcherPage (author) the target class is the ResearcherPage
- **types** is a map that allow clustering of the relationship under user-defined category (facet query), see the dedicated paragraph for further details

Relationship definition

The relationships are defined in the `[installDir]/config/spring/cris/cris-relationpreference.xml`

A relationConfiguration is shown in the snippet below

```

<bean id="relationRPVSIItemConfiguration" class="org.dspace.app.cris.configuration.RelationConfiguration">
  <property name="relationName" value="crisrp.publications" />
  <property name="relationClass" value="org.dspace.content.Item" />
  <property name="query">
    <value>author_authority:{0}</value>
  </property>
</bean>

```

It is defined by

- **relationName**, an unique name that characterize the relationship
- **relationClass**, the class canonical name of the Entity Class that holds the main direction of the relationship
- **query**, the solr query that can be used to retrieve the objects linked to the “current” object. {0} will be replaced with the authority key of the current object

Relation Preference Management

The relationConfiguration are used other than by the components by the RelationPreferenceService that can be enabled over one or more relationship so to allow the user to hide some relationship or make a selected list. The RelationPreferenceService is configured through the definition of a spring bean for the class *org.dspace.app.cris.configuration.RelationPreferenceServiceConfiguration* the default configuration is shown below

```

<bean class="org.dspace.app.cris.configuration.RelationPreferenceServiceConfiguration">
  <property name="list">
    <list>
      <bean class="org.dspace.app.cris.configuration.RelationPreferenceConfiguration">
        <property name="relationConfiguration" ref="relationRPVSIItemConfiguration"/>
        <property name="selectActionAuthLevel">
          <util:constant static-field="org.dspace.app.cris.configuration.RelationPreferenceConfiguration.ENABLED" />
        </property>
        <property name="hideActionAuthLevel">
          <util:constant static-field="org.dspace.app.cris.configuration.RelationPreferenceConfiguration.ENABLED" />
        </property>
        <property name="unlinkActionAuthLevel">
          <util:constant static-field="org.dspace.app.cris.configuration.RelationPreferenceConfiguration.
ONLY_SYSTEM_ADMIN" />
        </property>
        <property name="extraAction">
          <bean class="org.dspace.app.cris.configuration.ItemExtraAction" />
        </property>
        <property name="columnsVisualizationConfiguration">
          <list>
            <bean class="org.dspace.app.cris.configuration.ColumnVisualizationConfiguration">
              <property name="name" value="title" />
              <property name="metadata">
                <list>
                  <value>dc.title</value>
                </list>
              </property>
            </bean>
            <bean class="org.dspace.app.cris.configuration.ColumnVisualizationConfiguration">
              ...
            </bean>
          </list>
        </property>
      </bean>
    </list>
  </property>
</bean>

```

The RelationPreferenceServiceConfiguration keeps a list of RelationPreferenceConfiguration that defines for a specific relationConfiguration which actions (select, hide, unhide, unlink) are available and to whom (researcher and/or system administrators). Finally, the columnsVisualizationConfiguration allows to define which columns show in the relation management functionality (see) and the extraAction allows to inject the business logic needed to manage the preference changes (from a standard relationship to a selected one, from an hide relationship to a standard one and so on).

Relation Clustering

The configuration of the clustering categories are done using spring beans. The snippet below show the default configuration for a "journal article" category applied to the publication list

```

<bean id="journalComponent" class="org.dspace.app.webui.cris.components.BeanFacetComponent">
  <property name="componentIdentifier" value="journal" />
  <property name="facetQuery">
    <value>dc.type:Article -relationpreferences.crisrp.publications.selected:{1}</value>
  </property>
  <property name="order"><value>asc</value></property>
  <property name="rpp"><value>20</value></property>
  <property name="sortBy"><value>1</value></property>
</bean>

```

The facetQuery property is evaluated replacing {0} with the authority key of current object and {1} with the uuid.

The default configuration exclude the selected publications from their native category (article, book, etc.) so to have not overlapping categories but it can easily be changed modified the facet query definition removing the

```
-relationpreferences.crisrp.publications.selected:{1}
```

CRIS Processors

This feature has been introduced in DSpace-CRIS 5.5

A processor is a JAVA class that implements the interface *org.dspace.app.cris.util.ICrisHomeProcessor* <ACO extends *ACrisObject*> allowing the execution of extra code any times that a specific type of entity (Researcher Page, OrgUnits, etc.) is visualized. It has access to the request object so can provide additional information to a custom view to visualize more information maybe retrieved from external system or other dynamic behaviors.

It is the transposition to the CRIS entities of the DSpace Processor (Site, Item, Community, Collection in *org.dspace.plugin*).

Out of box it is provided the *org.dspace.app.webui.cris.metrics.MetricsCrisHomeProcessor* <ACO> that enable the retrieve and visualization of the metrics available for a specific object, it is defined as a spring bean in

[*dspace-installDir*]/*config/spring/cris/cris-processor.xml*

```
<bean class="org.dspace.app.webui.cris.metrics.MetricsCrisHomeProcessor"
      id="rpHomeProcessor">
  <property name="configurator" ref="org.dspace.app.webui.cris.metrics.
MetricsProcessorConfigurator"/>
  <property name="rankingLevels">
    <list>
      <value>1</value>
      <value>5</value>
      <value>10</value>
      <value>20</value>
      <value>50</value>
    </list>
  </property>
  <property name="metricTypes">
    <list>
      <value>view_aggregate</value>
      <value>download_aggregate</value>
      <value>scopus_aggregate</value>
      <value>scopus_count</value>
      <value>wos_aggregate</value>
      <value>wos_count</value>
      <value>pubmed_aggregate</value>
      <value>pubmed_count</value>
    </list>
  </property>
  <property name="clazz" value="org.dspace.app.cris.model.ResearcherPage" />
  <property name="searchService" ref="org.dspace.discovery.SearchService" />
</bean>
```

Customize the Home page

The home page includes a number of static text that can be edited using the Edit news functionality and dynamic content generated by SiteProcessor.

In addition to what is provided by a standard DSpace the following improvements and extra functionalities are available:

- It is possible to configure a footer news editable from the Edit news admin functionality that is shown in all the public pages
- A processor that build an infographics of the site contents aggregating the entities types under three icons
 - *org.dspace.app.webui.discovery.GlobalFacetProcessor*
- There are three additional process that allow to list in the home page the top items sorted respectively by number of citations, number of view, number of download
 - *org.dspace.app.webui.components.MostViewedMetricItemSite*
 - *org.dspace.app.webui.components.MostDownloadedMetricItemSite*
 - *org.dspace.app.webui.components.MostCitedItemSite*

These functionalities are all than the *MostCitedItemSite* plugin enabled by default in the DSpace-CRIS home.jsp and *dspace.cfg* configuration

```

plugin.sequence.org.dspace.plugin.SiteHomeProcessor = \
    org.dspace.app.webui.components.RecentSiteSubmissions,\
    org.dspace.app.webui.discovery.GlobalFacetProcessor,\
    org.dspace.app.webui.components.MostViewedMetricItemSite,\
    org.dspace.app.webui.components.MostDownloadedMetricItemSite

```

The org.dspace.app.webui.discovery.GlobalFacetProcessor is configured as a spring bean in **[dspace-installDir/config/spring/cris/cris-layout-processor.xml]**

```

<bean class="org.dspace.app.webui.discovery.GlobalFacetProcessorConfigurator" name="org.dspace.app.webui.
discovery.GlobalFacetProcessorConfigurator">
    <property name="groups">
        <map>
            <entry key="group-left">
                <list>
                    <bean class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                        <property name="name" value="researcherprofiles"
/>
                    </bean>
                </list>
            </entry>
            <entry key="group-center">
                <list>
                    <bean class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                        <property name="name" value="orgunits"
/>
                    </bean>
                    <bean class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                        <property name="name" value="fundings"
/>
                    </bean>
                    <bean class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                        <property name="name" value="journal"
/>
                    </bean>
                    <bean class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                        <property name="name" value="events"
/>
                    </bean>
                </list>
            </entry>
            <entry key="group-right">
                <list>
                    <bean
                        class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                            <property name="name" value="publications" />
                            <!-- <property name="secondLevelFacet"> <list>
<value>itemtype</value>
                                </list> </property> -->
                        </bean>
                    <bean
                        class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                            <property name="name" value="conferencematerials" />
                        </bean>
                    <bean
                        class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
                            <property name="name" value="datasets" />
                        </bean>
                    <bean
                        class="org.dspace.app.webui.discovery.

```

```

GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
    <property name="name" value="patents" />
    </bean>
    <bean
        class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
    <property name="name" value="theses" />
    </bean>
    <bean
        class="org.dspace.app.webui.discovery.
GlobalFacetProcessorConfigurator.InnerGlobalFacetProcessorConfigurator">
    <property name="name" value="others" />
    </bean>
    </list>
    </entry>
    </map>
    </property>
</bean>

```

To enable the MostCitedItemSite it is necessary to add the processor to the previous configuration snippet and configure the retrieval of the citation counts from Pubmed, Scopus or Web of Science. See Integration with external databases

These processors are indeed based on the use of spring beans with the following ID viewedTopItemManager, downloadedTopItemManager and citedTopItemManager. So it is possible further customize the behavior editing the spring definition that can be found in **[dspace-installDir]/config/spring/cris/cris-processor.xml**

```

<bean class="org.dspace.app.webui.components.TopItemManager"
    name="citedTopItemManager">
    <property name="sortCriteria" value="scopus" />
    <property name="searchService" ref="org.dspace.discovery.SearchService" />
    <property name="maxResults" value="5" />
</bean>

```

For example changing the sortCriteria from scopus to pubmed the component will list the first five publications (maxResults) for number of citation in Pubmed (assuming that pubmed is the name assigned to the pubmed citation count metric).

Defining the site sections (Explore)

This [feature](#) has been introduced in DSpace-CRIS 5.5

It is possible to highlight specific area of the database defining an entry page for each section. The sections are automatically defined as the entity types defined by the Global Search configuration but not all the entity types need to be included in the top navigation bar.

The section / entity type included in the navigation bar are defined by the following property in the **[dspace-installDir]/config/modules/cris.cfg**

```

navbar.cris-entities = publications,researcherprofiles,orgunits,fundings

```

For each section it is possible to define which browses proposes and which dynamic components include. The system looks up for a Singleton Service id org.dspace.app.webui.cris.components.ExploreMapProcessors by default it is defined in **[dspace-installDir]/config/spring/cris-processor.xml** indeed each piece of information to include in the section page is linked to an ExploreProcessor

```

<bean class="org.dspace.app.webui.cris.components.ExploreMapProcessors" id="org.dspace.app.webui.cris.
components.ExploreMapProcessors">
    <property name="processorsMap">
        <map>
            <entry key="publications">
                <list>
                    <bean class="org.dspace.app.webui.cris.components.
BrowseProcessor">
                        ...
                    </bean>
                    ...
                </list>
            </entry>
            <entry key="researcherprofiles">
                ...
            </entry>
            ...
        </map>
    </property>
</bean>

```

each section is an entry of in the above map containing a list of implementation of the *org.dspace.app.webui.cris.components.ExploreProcessor* interface.

By default two implementations of such interface are available

- *org.dspace.app.webui.cris.components.BrowseProcessor* - it allows to set the list of browse to include in the section
- *org.dspace.app.webui.cris.components.TopObjectsProcessor* - it allows to configure a list a top objects customizing the sorting criteria (most recent, most viewed, most downloaded, etc.)

The following configuration snippet define six browses to include in that specific order in a section

```

<bean class="org.dspace.app.webui.cris.components.BrowseProcessor">
    <property name="browseNames">
        <list>
            <value>itemdept</value>
            <value>author</value>
            <value>title</value>
            <value>type</value>
            <value>dateissued</value>
            <value>subject</value>
        </list>
    </property>
</bean>

```

here are two examples of configuration for the *TopObjectsProcessor*

```

<bean class="org.dspace.app.webui.cris.components.TopObjectsProcessor">
    <property name="viewConfiguration" ref="dspaceitemsLayoutView" />
    <property name="type" value="recentsubmission" />
    <property name="sortField" value="dateaccessioned_dt" />
    <property name="order" value="desc" />
    <property name="num" value="4" />
</bean>
<bean class="org.dspace.app.webui.cris.components.TopObjectsProcessor">
    <property name="viewConfiguration" ref="dspaceitemsLayoutView" />
    <property name="type" value="view" />
    <property name="sortField" value="crismetrics_view" />
    <property name="order" value="desc" />
    <property name="extraInfo">
        <value>crismetrics_view</value>
    </property>
    <property name="num" value="4" />
</bean>

```

The indexes proposed for the advanced search are instead derived by the discovery configuration associated with the entity type alias (publications, researcherprofiles, etc.)

Defining a list layout

The Global search and the previous defined processor (TopObjects, etc.) used in the section explore and in the home page refer to a "viewConfiguration" that is usually defined in the *config/spring/api/global-list-layout.xml* file

The file contains one or more bean definitions for the class *org.dspace.discovery.configuration.DiscoveryViewConfiguration*

```
<bean id="dspaceitemsLayoutView"
      class="org.dspace.discovery.configuration.DiscoveryViewConfiguration">
  <property name="metadataHeadingFields">
    <list>
      <bean
        class="org.dspace.discovery.configuration.
DiscoveryViewFieldConfiguration">
        <property name="mandatory" value="true" />
        <property name="preHtml" value="<b>" />
        <property name="postHtml" value="</b>" />
        <property name="field" value="dc.title" />
        <property name="decorator" value="title" />
      </bean>
      <bean
        class="org.dspace.discovery.configuration.
DiscoveryViewFieldConfiguration">
        <property name="mandatory" value="true" />
        <property name="preHtml" value="<span class='text-muted'>" />
        <property name="postHtml" value="</span><br/>" />
        <property name="field" value="dc.contributor.author" />
        <property name="decorator" value="itemcrisref" />
      </bean>
      <bean
        class="org.dspace.discovery.configuration.
DiscoveryViewFieldConfiguration">
        <property name="mandatory" value="true" />
        <property name="postHtml" value="<br/>" />
        <property name="field" value="dc.identifier.citation" />
      </bean>
      <bean
        class="org.dspace.discovery.configuration.
DiscoveryViewFieldConfiguration">
        <property name="mandatory" value="true" />
        <property name="preHtml" value="<small>" />
        <property name="postHtml" value="</small>" />
        <property name="field" value="dc.type" />
      </bean>
    </list>
  </property>
  <property name="metadataDescriptionFields">
    <list>
      <bean
        class="org.dspace.discovery.configuration.
DiscoveryViewFieldConfiguration">
        <property name="field" value="dc.subject" />
        <property name="postHtml" value="<br />" />
      </bean>
      <bean
        class="org.dspace.discovery.configuration.
DiscoveryViewFieldConfiguration">
        <property name="field" value="dc.description.abstract" />
        <property name="postHtml" value="<br />" />
      </bean>
    </list>
  </property>
</bean>
```

The bean has two properties

- *metadataHeadingFields*: defines which metadata are included as heading of each result
- *metadataDescriptionFields*: defines which metadata are listed in a key - value layout under each result

both properties accept a list of *DiscoveryViewFieldConfiguration* beans with the following properties

- mandatory: default false, if true the information will be always displayed. Otherwise it is only displayed if the field match the user query, in such case a snippet of the content is presented
- preHtml and postHtml: any additional html code to include before and after the value. Please note that empty values will be always skipped
- field: the metadata to use
- decorator: an optional reference to a `org.dspace.app.webui.util.IDisplayMetadataValueStrategy` plugin defined in the `dspace.cfg` (the plugin name must be used, i.e. `itemcrisref`, `doi`, etc.)