How to collect data about CRIS entities

DSpace-CRIS has three ways to collect data related to CRIS entities:

1. Manual input the information via the administrative UI (good for small organisation with limited data), see Entities management
2. Importing / updating data from external systems that act as master for specific entities (HR system for most of the researcher profile information, grants database, etc.).

   Right now this can be done using xls file as described here. It is envisioned, as part of the work forward to DSpace(-CRIS) 7 to add support to READ and WRITE CRIS objects via REST webservices
3. Creating the entities as part of a DSpace item submission.

Here we will describe in more details the 3rd way.

A DSpace consumer has been configured in the dspace.cfg to monitor for new archived items

```
# crisorcid consumer
event.consumer.crisorcid.class = org.dspace.app.cris.integration.authority.CrisConsumer
event.consumer.crisorcid.filters = Item+INSTALL|Modify|Modify_Metadata
```

Please note that as the consumer was initially developed to deal with the import of bibliographic information of the authors from ORCID it was named crisorcid but it now manage all kind of object (rp, orgunit, project, 2nd level entities su as journal, etc.) and metadata

when an item is archived the consumer looks to the item metadata that are using an authority implementing the *org.dspace.app.cris.integration.CRISAuthority* interface and so potentially linked to CRIS objects. For each metadata values that doesn't have an authority key or that have a temporary authority (one that starts with "will be generated::") the consumer create a new CRIS object of the proper type (rp, orgunit, ...) setting the name of the created instance to the text value of the original metadata. The type of the new CRIS object to create is provided by the Authority implementation associated with the specific metadata, so for instance

```
choices.plugin.dc.contributor.author = RPAuthority
...
choices.plugin.dc.relation = ProjectAuthority
...
choices.plugin.dc.relation.dept = OUAuthority
```

the above configuration means that a new value for the metadata dc.crontributor.author creates a Researcher Profile, a new value for the dc.relation a Project and a new value for dc.relation.dept an OrgUnit. When the authority is org.dspace.app.cris.integration.DOAuthority or a subclass

```
#####  Authority Control Settings  #####
plugin.named.org.dspace.content.authority.ChoiceAuthority = \
        ...
 org.dspace.app.cris.integration.DOAuthority = JOURNALAuthority,\
...
choices.plugin.dc.relation.ispartof = JOURNALAuthority
...
cris.DOAuthority.dc.relation.ispartof.new-instances = journals
```

the exact type of 2nd level object to create is specified by the property cris.DOAuthority.<metadata>.new-instances. The above example means that new value not present in the authority will lead to the creation of ResearchObject (2nd level entity) of type: journals

The system provides an extension point to enrich the created object using additional logic, this extension point is also referred as "ImportFiller framework". There are two main interfaces that allow you to extends

- org.dspace.app.cris.integration.authority.ImportAuthorityFiller - implements this interface if you need to add metadata / property to the created CRIS object
- org.dspace.app.cris.integration.authority.TargetMetricFillerPlugin - implements this interface if you need to add metrics to the created CRIS object

The logic is injected in the system using spring, the configuration service bean with id org.dspace.app.cris.integration.authority.AuthoritiesFillerConfig is retrieved. By default it is defined in the **[dspace-installDir]/config/spring/cris/cris-plugin.xml** file as follow

```
<util:constant id="CrisConsumer-SOURCE_INTERNAL"
                static-field="org.dspace.app.cris.integration.authority.CrisConsumer.SOURCE_INTERNAL" />

        <bean id="org.dspace.app.cris.integration.authority.AuthoritiesFillerConfig" class="org.dspace.app.cris.
integration.authority.AuthoritiesFillerConfig">
                <property name="fillers">
                        <map>
                                <entry key="orcid">
                                        <bean class="org.dspace.app.cris.integration.authority.
ORCIDImportFiller" parent="fullitemMetadataConfiguration" />
                                </entry>
                                <entry key-ref="CrisConsumer-SOURCE_INTERNAL">
                                        <bean class="org.dspace.app.cris.integration.authority.
ItemMetadataImportFiller" parent="fullitemMetadataConfiguration" />
                                </entry>
                        </map>
                </property>
        </bean>
```

The filler map refers the implementation to use depending on the value of the authority key. If the metadata that have originated the new CRIS object doesn't have an authority key at all the special SOURCE_INTERNAL value is assumed so the org.dspace.app.cris.integration.authority. ItemMetadataImportFiller implementation is used. Instead, if the authority key start with "will be generated::**orcid**::" the org.dspace.app.cris.integration. authority.ORCIDImportFiller implementation is used.

The entry key is "generated" by the authority implementation, for example the org.dspace.app.cris.integration.ORCIDAuthority return for each name retrieved from the ORCID registry the authority key "will be generated::**orcid**::<ORCiD ID>" in this way the "Filler" is able to use the additional information (it was retrieved from ORCID, he have a specific ORCID ID) to enrich the CRIS object

The org.dspace.app.cris.integration.authority.ORCIDImportFiller implementation uses the ORCiD ID to retrieve from ORCID the bibliographic information about the author and, using the reverse mapping implicitely defined by the orcid preference properties (see ORCID Integration) populate the new Researcher Page. In additional, as the ORCIDImportFiller is an extension of the ItemMEtadataImportFiller it is also able to fill additional information of the new ResearcherPage with information provided in other metadata of the orginal item.

The org.dspace.app.cris.integration.authority.ItemMetadataImportFiller is defined as spring bean

```
<bean class="org.dspace.app.cris.integration.authority.ItemMetadataImportFiller" id="
fullitemMetadataConfiguration" abstract="true">
        <property name="applicationService" ref="applicationService" />
        <property name="metricsPersistenceService" ref="org.dspace.app.cris.metrics.common.services.
MetricsPersistenceService" />
        <property name="allowsUpdateByDefault" value="true" />
        <property name="configurations">
                <bean class="java.util.HashMap" />
        </property>
</bean>
```

the default configuration is "empty" as the configurations property holds a clean map, it is here that you need to configure the system depending on your needs and the information that you are able to collect during the item submission.

A complete configuration looks like

```
<property name="configurations">
                        <map>
                                <entry key="dc.contributor.author">
                                        <ref local="researcherPageCommonFillerMetadata" />
                                </entry>
                                <entry key="dc.source.title">
                                        <ref local="journalCommonFillerMetadata" />
                                </entry>...<property>
```

where each map value is something like

```
        <bean
                class="org.dspace.app.cris.integration.authority.ItemMetadataImportFillerConfiguration"
                id="researcherPageCommonFillerMetadata">
                <property name="mapping">
                        <map>
                                <entry key="custom.contributor.affiliation">
                                        <bean
                                                class="org.dspace.app.cris.integration.authority.
ItemMetadataImportFillerConfiguration.MappingDetails">
                                                <property name="shortName" value="principalaffiliation" />
                                                <property name="useAll" value="false" />
                                                <property name="visibility" value="1" />
                                        </bean>
                                </entry>
                                <entry key="custom.contributorid.scopusid">
                                        <bean
                                                class="org.dspace.app.cris.integration.authority.
ItemMetadataImportFillerConfiguration.MappingDetails">
                                                <property name="shortName" value="scopusid" />
                                                <property name="useAll" value="false" />
                                                <property name="visibility" value="1" />
                                        </bean>
                                </entry>
                                </map>
                </property>
        </bean>
```

the above example means that when a new researcher profile is created due to a metadata not linked to an existent rp

- the *principalaffiliation* property is populated with the content of the metadata *custom.contributor.affiliation*
- the *scopusid* property is populated with the content of the metadata *custom.contributorid.scopusid*

It is important to note that

- it is possible to set the visibility of the created property to private (0) or public (1)
- it is possible to manage repeatable metadata such as authors using the useAll flag. False mean that the filler expects to have exactly the same number of primary metadata (*dc.contributor.author* in our example) than additional metadata (*custom.contributor.affiliation*) and use the additional metadata to fill the researcher profile created by the primary metadata with the same place. The first custom.contributor.affiliation is added to the first contributor.author, etc.

  The submission understand such configuration and react adding more box to input the additional metadata any time that a new value for the primary metadata is added. Currently the user experience is not optimal as the primary value and the additional metadata are NOT placed aside and there are no extra validation that assure a proper use of the UI (the user could input three author names but only two affiliation, etc.).
- The filler works on all the metadata so if the custom.contributor.affiliation is controlled by an OUAuthority and the value is not selected from the existing OrgUnits list a new OU object can be created and filled with this infrastructure.

The automatic creation of entities as part of a DSpace item submission is managed by the configuration at cris.cfg, both at a global and at single metadata level. Global configuration is configured through import.submission.enabled.entity key. Specific metadata fields configuration is done with import.submission.enabled.entity.<schema>.<element>.<qualifier> key. For instance:

```
import.submission.enabled.entity = true
import.submission.enabled.entity.dc.contributor.author = false
```

will mean that dspace-cris will create new cris objects as usual (i.e. journals, projects, etc.), with the exception of new authors at dc.contributor.author metadata field.