

Component Registry test plan

This is a test plan for the Component Registry ([alpha](#), [beta](#), [production](#)). Thank you for testing the application!

Please make a **copy of this document**, and add your findings inline using [‘suggesting’ mode](#) or add [comments](#) (or edit normally in a distinct colour if you prefer). You don’t have to do anything for steps that succeed. When you are done, send a (link to) your copy of this document to testing@clarin.eu.

- Name of tester: **[please fill in your full name]**
- E-mail address: **[please fill in your e-mail address]**
- Application version: **[please fill in, e.g. 2.2.0-beta1]**
- Operating system used while testing: **[please fill in, e.g. Firefox 50.0]**
- Browser used while testing: **[please fill in, e.g. Firefox 49]**
- Date finished testing: **[please fill in the date]** (YYYY-MM-DD)

General remarks

Please write down any general issues you encountered, remarks or suggestions below:

[...]

The test plan begins on the next page.

Existing functionality

Please note: Before testing, make sure to ask the application developers or test manager to add you as a tester to two teams in the Component Registry. You will need them later on. To do this, **log into the Component Registry once** first, then **send** your username as shown in the application to the application developer or test manager.

Browsing unauthenticated

1. **Open the application** in a browser (write down what browser + version, operation system + version)
2. Scroll through the list of public Profiles in the Component Browser. Click a number of Profiles and **look at their contents** in the 'view' tab. Linked child Components should be collapsed initially at all levels, but **should all be expandable**. Also verify that matching content is shown in the 'xml' tab.
3. Do the same for public Components.
4. Find a Component or Profile with one or more comments (you can sort the 'comments' column to make this easier). You should be able to **read** all the comments in the 'comments' tab below the table **but not submit** any yourself. You **should not be allowed to delete** any existing comments.
5. Use the text box in the top right corner of the Component Browser to limit the items shown to those matching some text (e.g. 'cmdi' or 'corpus') in their title, description, group name or creator name and see if the remaining items **actually match the filter**. Do this for both Components and Profiles. The **filter text should stay** when switching type.
6. Try entering a filter text that leads to no results. An option to clear the filter should appear in the table.
7. See if you can find a way to alter (edit, copy, delete, change status etc.) a Component or Profile. This should **not** be possible.
8. Find and click the help icon. Does it bring you to a suitable web page?
9. **Download** the XML and XSD for a Profile (CMDI 1.1 and CMDI 1.2 versions).
10. **Download** the XML for a Component (CMDI 1.1 and CMDI 1.2 versions).

Authenticated browsing

1. **Log in** (using your home institution's IdP or your CLARIN website account). You should automatically **return to the Component Browser** after logging in.
2. Your username should **show up** in the place of the login link.
3. There should be a 'settings' option next to it that takes you to a setting page (in a new tab or window). **Click** it and **enter** your full name in the 'User display name' field.
4. The 'New' button should be **enabled**
5. The 'Edit as new', 'Delete' and 'Status' buttons should be **enabled if an item is selected**
6. Verify that you **cannot actually delete** any item (that you did not recently publish yourself) or change its status (e.g. from *production* to *deprecated*).

Creating and modifying Components and Profiles

1. Make sure you are logged in and switched to the private workspace. **Press create new**. This **should** take you to a **new page** to create a new Profile or Component (depending on the space selected in the Component Browser)
2. **Press 'cancel'** at the top. You **should be back** to the Component Registry.
3. **Click 'Create new'** again. **Enter** some (arbitrary) **values** in the fields that are shown until you get to the a priori domain selection option. Also **set a concept link**, you can use 'test' as a search term.
4. **Click the "+Element" link** near the bottom of the page to add an Element.
 - a. Fill in all the text fields that are empty. First, leave the field 'Name' empty and move on to another field;
 - b. **Verify** that an error message appears next to the field indicating that the field cannot be empty.
 - c. Also **verify** that you can select a concept link for the Element.
 - d. **Set** the value of display priority to 1.
5. For the Min and Max Occurrence fields **check** whether the drop down allows you to choose among different numbers (0-10 for min, 1-10 and Unbounded for max, but 'min' **never more** than 'max' and 'max' **never less** than 'min')
6. **Check the checkbox 'multilingual'** below the Element description fields. The Max occurrence field **should** be greyed out and set to Unbounded, making it impossible for you to browse in the respective scroll down menu.
7. The type field **should** be set to 'string' by default. After pressing the edit button a dialogue **should** pop up, containing three different options for setting the type (type, controlled vocabulary, pattern).
8. In the 'Type' tab, use the drop down menu to select a type.
 - a. Try to **set it to a different option** than string. Press 'Use' next to the selection option.
 - b. The displayed type in the type field **should** change accordingly.
 - c. Press the edit button again, select 'String' and **press 'Cancel'**. The action **should be interrupted** immediately without changing the value in the type field.
9. Go again to the type dialogue and click on the 'Controlled vocabulary' tab.
 - a. The selected vocabulary type **should be** 'Open' by default. **Change** it to 'Closed'; a table should appear.
 - b. **Add** two items and insert some arbitrary text in the 'value' and 'description' columns.
 - c. **Add** a concept link to one of the items by clicking the 'Add concept link' button, searching for a concept link and **selecting** one.
 - d. Finally, **click on 'use controlled vocabulary'**. The result of this action **should** be that the type dialogue closes and a drop down menu is shown for the edited type field representing the items in the vocabulary.
10. Go again to the type dialogue and go to the 'Pattern' tab this time.
 - a. There you can **enter** a valid regular expression e.g. 'hdl' into the field.

- b. **Press 'Use Pattern'**. Now the sequence typed in as pattern should be displayed in the type field.
11. **Click the "+Attribute" link** below the header fields add an Attribute. Repeat the steps above related to Element for the new Attribute except for display value and cardinality, which do not apply to Attributes.
12. **Click the "+Component" link** near the bottom of the editing interface to add a Component.
 - a. **Choose "add inline Component"** to add a Component within the Profile you are editing on the level of the Element(s) and Attribute(s) added before.
 - b. **Insert values** for name, concept link, documentation
 - c. **Check** if the min/max occurrence selectors function properly (same as for Elements).
13. **Click the "+Component" link** within the newly created Component and choose "Link existing Component" you will see a selection of various existing Components.
 - a. Pick an arbitrary Component and **add it** using the "+" button.
 - b. **Verify** that the Component can be **expanded** and its child Components, Elements and Attributes are visible (but **not editable**).
 - c. You **should** be able to change its min/max occurrence values while it is expanded.
14. **Click the "+Component" link** at the same level and choose "add inline Component" to create an inline child Component at the second level.
 - a. **Insert a value** for 'Name'
 - b. **Add a child Element** to it with a name and display priority of 1.
15. Use the up/down arrows on the right hand side of the Component boxes' header to **swap the order** of the two sibling Components.
 - a. **Verify** that the order has indeed changed.
 - b. **Add** another linked Component and **verify** that changing the order also works for three or more siblings.
16. Use the 'X' icon on the right hand side of the header of one of the linked Components to **remove** it. **Verify** that it does indeed get removed.
17. **Test the reordering and deletion** of Elements and Attributes in the same way.
18. Press cancel in the top of the editor. Press 'Yes' to **confirm** and you return to the Component Browser.
19. **Create** a new Profile containing at least three Elements, three Attributes, two inline Components and two linked Components.
 - a. **Make sure** that there are at least three hierarchical levels (e.g. an Element within an inline Component within an inline Component). Use any of the following value schemes at least once: 'type', 'pattern' and both open and closed 'controlled vocabularies'.
 - b. **Write down** the structure of the document or make a screenshot.
 - c. Press Save new, and **resolve** any validation issues that may occur.
 - d. You **should return** to your private workspace in the Component Browser with the newly created Profile selected.
 - e. **Verify** that its content in the 'view' tab **matches** the definition that you provided in the Component Editor.
20. Press the 'Edit' button to **edit** the Profile you just created.

- a. **Change** the names of one or more items (Components, Elements or Attributes) and **move** and **delete** some other items.
 - b. Write down your changes, press **Save** and verify that the changes have been applied once the Profile is shown in the Component Browser again.
21. Press the 'Edit' button to
- a. **edit** the Profile you just created, change its name (no further changes required) and **press** the Save new button.
 - b. **Verify** that an exact copy of the Profile has been saved into your private workspace with the new name and is shown correctly in the Component Browser.
22. In the Component Browser, **switch** to Components (while staying in the private workspace), and press the 'New' button to create a new Component. **Follow the same instructions** that you followed for the new Profile in the previous steps.

Private and team workspaces

1. In the private workspace you **should see** the Profiles and Components you have created.
 - a. The name of the creator **should be equal** to the name you have typed when setting the user display name.
2. Next to the 'New' and 'Edit' buttons you can use the dropdown menu 'Move to team' where it should be possible to **select one of your teams**.
 - a. If you select one of the teams, a warning **should** appear telling you that after moving the Profile to a team you cannot move it back anymore.
 - b. **Press 'No'**. The warning should close and you should see the normal workspace again.
 - c. Now **try the same operation again**, this time clicking 'yes' when the warning appears.
 - d. You **should** be redirected to the target team space you chose with the moved item selected.
3. **Select** the moved Profile in the team and check whether you can edit it when pressing edit at the top. You should be directed to the editor.
 - a. Make a few **small changes**, press Save, and **verify** that the changes have been applied properly while the Profile remains in the team space.
4. In the Component Browser with the same team selected, **choose another group** from the 'move to...' dropdown.
 - a. Again a warning should appear. Press 'yes'.
 - b. You **should** be redirected to the target team space you chose with the moved item selected.
5. **Select** a Component in your private workspace.
 - a. **Find** the Component's ID by going to the 'XML' tab. The id appears between the <ID>...</ID> tags. **Write it down**.
 - b. **Click** the 'Edit' button. The Component Editor should appear.
 - c. **Press** the the dropdown at the top: 'Publish' and select the 'Publish' option. A message should appear informing you that if your Profile is ready you can publish it. It also offers you the option to abort if you are not ready. **Press No'**.

- d. Repeat the operation, this time **clicking 'Yes'**.
- e. You **should** return to your private workspace. **Verify** that the Component is no longer present.
- f. Locate the published Component in the public workspace and confirm that it is present, and has the correct creator name, date and ID (see the 'XML' tab).

New features and changes (Component Registry 2.2.0)

Component lifecycle management:

1. Go to the public space in the Component Browser, **verify** that there are one or more Components and Profiles with the 'Development' status by setting the status filter to 'Development'.
2. Verify that there are one or more Components and Profiles with the 'deprecated' status by setting the status filter to 'Deprecated'.
3. Find an existing deprecated Component or Profiles with a successor. If so, you should be able to **click through to the successor**.
4. Make sure you are logged in, then create a new private Component (you can open an existing one in the editor and use 'save as new' for this)
 - a. **Change** its status to deprecated.
 - b. Choose to **not** set a successor.
 - c. Go the deprecated items in the private space and **verify that it appears** in the list.
5. **Do the same** for a private Profile.
6. **Do the same** for a new Component that you moved to a team space after saving (the item should appear in the list of deprecated items of that team after deprecation).
7. **Do the same** for a new Profile that you moved to a team space after saving.
8. **Go to** the private Component that you just deprecated.
 - a. Click 'set successor' and **assign a random public Component as its successor**.
 - b. After reloading the Component (by **clicking another Component** first) a link to the successor **should** appear above the Component view (below the table).
 - c. This link **should** bring you to the selected successor.
9. Do the same for your deprecated private Profile, team Component and team Profile.
10. **Start editing** a new Component.
 - a. Choose the 'link existing Component' option and **insert a deprecated Component** that you know has a successor. Once it is added, the box for the Component should contain an option 'Replace with successor'.
 - b. Click it and **verify that the linked Component gets replaced with its successor**. You don't have to save the new Component.
11. Start editing an existing private Component or edit a public Component as new, save it and open the copy in the editor.
 - a. From the 'Publish' drop down menu, **choose the option 'Save as draft'** and click 'Yes'.
 - b. Go to the 'development' items in the public space and **verify** that the item you just published **is there with the development status**.

Component Editor enhancements and additions:

1. In the Component Editor, compare the CMDI 1.1 and CMDI 1.2 editing modes. Verify that the following functionalities are **not available** in CMDI 1.1 mode but are in CMDI 1.2 mode:
 - Setting of documentation language and/or adding additional documentation Elements to Components, Elements or Attributes (deletion of additional documentation Elements should be possible)
 - Setting the vocabulary type to 'open' and/or linking an external vocabulary In the 'Controlled vocabulary' tab of the type dialogue
 - Adding Automatic value expressions to Elements or Attribute (in the 'additional Element options' section)
 - Making an Attribute mandatory (via the 'Required' checkbox; unsetting this option should be possible in CMDI 1.1 mode).
2. **Verify that** multilingual documentation can be added to Components, Elements and Attribute
3. **Verify that** "Automatic value expressions" can be added to Elements and Attributes
4. **Verify that** regular expressions as Element/Attribute type ('Pattern' tab of the type dialogue of an Element or Attribute) are validated (by entering an invalid regular expression such as "this(is(invalid)")
5. **Verify that** it is **not possible** to save a Component or Profile without any Element or Attribute descendants.

Controlled vocabularies:

1. **Verify** that you can create an Element or Attribute with an open vocabulary as type. This requires linking a vocabulary; select any.
2. **Verify** that you can create an Element or Attribute with a closed vocabulary as type, both with and without a linked vocabulary (**try both**). It is always mandatory to enter one or more values in the table.
3. **Import** an external vocabulary into a closed vocabulary on an Element or Attribute. **Verify** that the values are inserted into the table, and that they are editable afterwards.
4. **Create** a new closed vocabulary Element or Attribute and add two or more values using the table. Then use the batch editing mode to change the order of the items in the vocabulary and change one or more values. Close the batch editing mode and **verify** that the changes have been applied to the table.

Component browser usability improvements/fixes:

1. **Verify** that a private Component can be **published** (either as draft or 'normal') from the Component Browser
2. **Verify** that a private Profile can be **published** (either as draft or 'normal') from the Component Browser
3. **Find** a Profile in the Component Browser with at least one linked Components (they can be recognised by their initial collapsed state).

- a. **Click the title** of one of the linked Components. This **should** change the state of the Component Browser such that that Component is now selected in the table and shown in the 'view' tab.
- b. The selected item **should be in view** without having to scroll to it manually.

Back end improvements and fixes:

1. Identify two different Components with the same name (they can be of any space or status).
 - a. **Create** a new private Profile and **link both Components** at its root level.
 - b. **Try to save** the Component and **verify** that this fails because the two linked Components have the same name.
 2. **Remove** one of the two 'duplicates' and **replace** it with an inline Component. Give it the same name (creating a new duplicate) and add an Element to make it valid.
 - a. Again, **try to save** the Component and **verify** that this **fails** because the linked and inline Components have the same name.
-

End of the test plan