

Hi krita developers,

This is the idea of tagging implementation in krita. Look at it once and correct me if I'm wrong .

What is Tagging ?

Tagging means adding some tags to the resources in krita and call them according to your use. In Tagging, there are three main functions

- 1). Adding a Tag – here, we will create a tag and add it to the resource.
- 2). Deleting a Tag – we can delete a tag for resource.
- 3). Search for resource – By using tags, we can search for resources which contains particular resource.

How we are implementing ?

1. UI Description

In the UI, we thought of placing a text field and three radio buttons for adding, deleting and searching. How UI works for three functions is described below

For Add functionality,



Whenever we click on the resource, the tags associated with them, will display in the text field separated with comma ' , ' like this " round, dark, green, "

If we want to add a tag

- 1) we click on the add radio button.
- 2). Next, we select the resource for which we want to add the tag.
- 3). Whenever we select the resource, the tags associate with the resource will be displayed. At the end of the list, the cursor will appear
- 4) Now we can enter the tag name and press enter to add.

For Delete functionality,



round, thick,

ADD Del Search

If we want to delete a tag

- 1) we click on the delete radio button.
- 2). Next, we select the resource for which we want to delete the tag.
- 3). Whenever we select the resource, the tags associate with the resource will be displayed.
- 4) we can delete the tag which we don't want and press enter.

For Search functionality,



Enter Tag here ...

ADD Del Search

- 1). we need to click on the search radio button
- 2). we enter the tag in the text field and press enter
- 3). we will provide the autocompletion option

2. Code Description

For implementing tagging, the first thing we need to do is to store the tags with resources. For this, we are using QmultiHash object in koResourceServer.h file.

QMultiHash <Qstring,key > - Qstring is ResourceFilename and key is tag

QmultiHash would make the lookup for a tag much faster ..

Now Functions :

void addTagObject (koResource, string);

void deleteTagObject (koResource, string);

koResourceList filterTagObjects (string);

whenever a tag is created for an object, we call the function addTagObject(koResource, string)

- we will check for whether this tag is valid or is already added to the resource.
- If it is valid and it is not added to the resource, we will add the tag using insert (key,value) of QmultiHash object.

whenever a tag is searched, we call the function filterTagObjects (string)

- we will check the resource and tag pair exists in the QmultiHash object using find(key,value) function.
- Resources names can be retrieved from m_resources list in koResourceServer
- If resource and tag pair found, we will display that resource.

whenever a tag is deleted, we will call the function deleteTagObjects(koResource, string)

- we will check for the resource and tag pair as in filterTagObjects() .
- we will delete that tag using remove function of QmultiHash object.

We also add two other functions

writeToXML() - this one will be called at the end of the program, which saves the data in the QmultiHash object in a XML file in gimp tags.xml format.

ReadFromXML() - this one will be called at the start up to retrieve the tags and store them in QmultiHash object.

I think for providing auto completion, we need to store tags in QstringList.

This the idea of the tagging and its implementation details. Please give me your feed back