

# How to use the Basket in Glozz

<http://www.glozz.org>

Version : **0.9**  
Date : June the 24<sup>th</sup>, 2014  
Author : Yann Mathet

*About the Glozz Basket.*

*The Basket was first introduced in Glozz within the GlozzQL request tool, to store a set of results. But in Glozz 2.0, it becomes a transverse tool, available everywhere in the platform and in the plugins.*

*Conception and development: Yann Mathet*

# Table of contents

1. Introducing the Glozz Basket.....	4
2. Getting started .....	4
2.1. Launching the Basket .....	4
2.2. Overview .....	5
3. Feeding the Basket .....	6
3.1. Manual feeding.....	6
3.2. Feeding the Basket with GlozzQL .....	7
3.3. Feeding with the Concordancer plugin .....	8
4. Adding or modifying features to the whole content of the Basket.....	8
5. Multiple Baskets.....	10

# 1. Introducing the Glozz Basket

The Basket is a place where the user can store any annotation of his choice, because they are of special interest for him, so that he can then observe, save or apply a treatment to these annotations. The Basket metaphor is as follows: once a corpus is loaded in Glozz, the user can consider annotations (units, relations, schemas) as items he may choose and add to his basket.

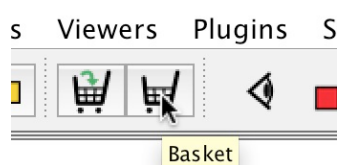
This feature is a transversal tool of Glozz, since it is available everywhere in the application (Glozz, GlozzQL, and Plugins).

## 2. Getting started

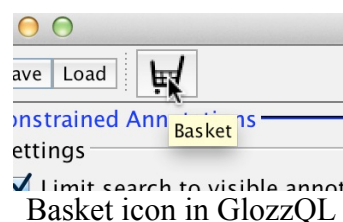
### 2.1. Launching the Basket

The Basket is generated as soon as Glozz is launched, is always available, and cannot be destroyed. In fact it is always present, but is either hidden or shown. So, there is no need to launch it, there is only to make it appear when necessary.

To make it appear on the screen, you may click on the icon shown below, either in the Glozz toolbar, or in the GlozzQL toolbar:



Basket icon in Glozz



Basket icon in GlozzQL

Note that if you close the Basket window, it only hides it, it does not close it. **When showing it again, the Basket will appear exactly as you left it.**

Besides, note that the Basket icon shows, if any, the number of annotations currently stored :



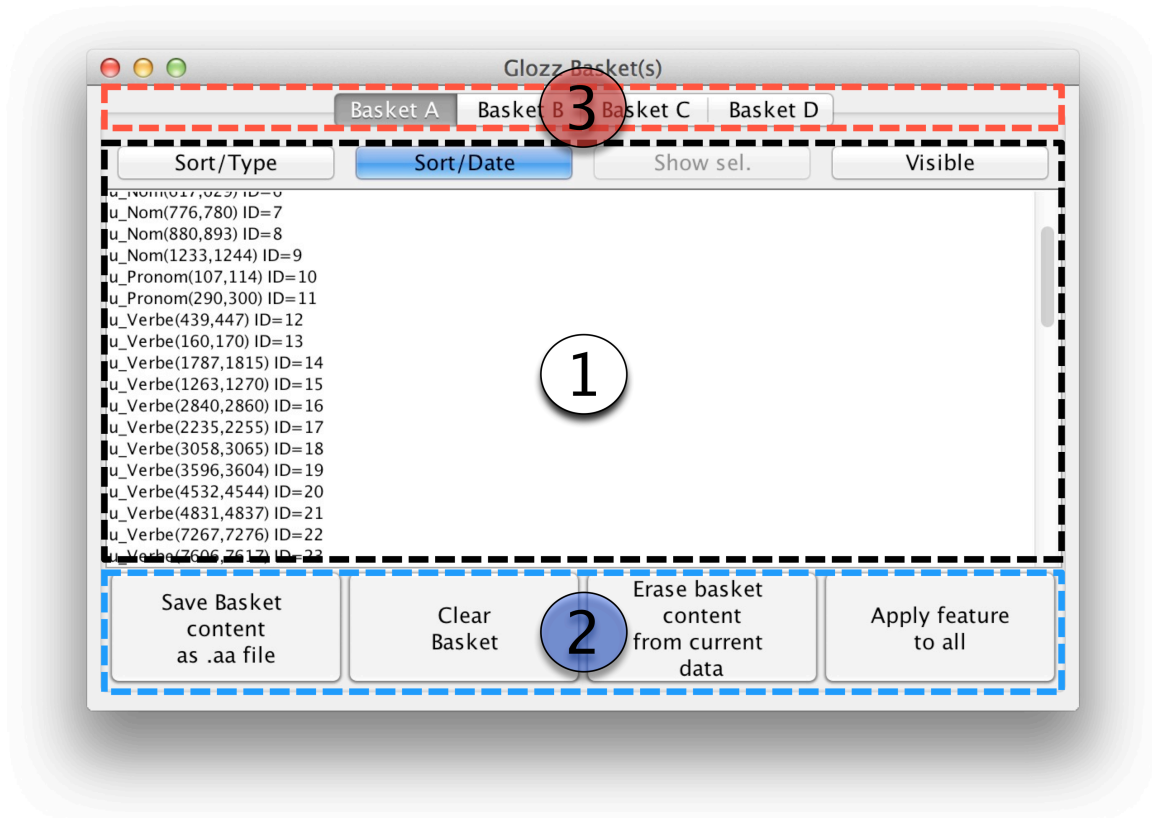
Empty Basket



Basket containing 62 annotations

## 2.2. Overview

The Basket includes 3 panels:



1) The main panel, in the center, contains the annotations currently stored in the Basket. It is exactly the same component as we find in the "annotation as text" view of Glozz, or in GlozzQL when you click on a given ConstrainedAnnotation to get its results. So it is a panel you may be already used to.

It comes with 4 buttons :

- "Sort/Type" sorts the content of the Basket depending on their type, in the following order: units, then relations, then schemas
- "Sort/Date" sorts the content according to creation date, from older to younger.
- "Show Selected" scrolls the list so that the current selected annotation, if any, appears in the list (in the case it is currently hidden, due to current scroll position)
- "Visible" makes the annotations currently hidden appearing again. Indeed, it is possible to **double-click** on any annotation of the list to hide it in Glozz (then it appears crossed out in the list), and then this button cancels this action.

Please see the main Glozz manual for more details and illustrations.

2) The panel at the bottom provides 4 possible actions on the Basket:

- **"Save Basket as .aa file"** : this saves the content of the Basket as an independant Glozz annotatin file. Then it will be possible to load the text with this .aa file to get only what was in the Basket.
- **"Clear Basket"** : it clears the Basket. It is the only way to clear the Basket, since closing the window will not really close the Basket, just hive it.
- **"Erase basket content from current data"** : this will remove IN GLOZZ all the annotations currently stored in the Basket. Be carefull, this action cannot be cancelled once accepted.
- **"Apply feature to all"** : this action will be shown further in a special section. It basically consists in creating or modifying a feature for each annotation currently stored in the Basket.

3) Current Basket chooser:

In fact, there is not one Basket, but 4 Baskets (namely "A", "B", "C", and "D") which can be used at the same time, to store different kinds of annotations, if needed. This will be explained further, but for most usages, only the default Basket (namely, Basket A) will be used. So, there is no need to read the specific section if you intend to use only the default Basket, and so, you won't have to click on the other tabs (B, C, and D) in this case.

### 3. Feeding the Basket

At the time this manual is written, there are 3 ways to feed the Basket. Other ways may appear in the future since any new plugin may be designed to do so.

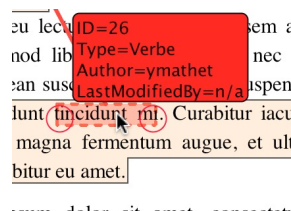
#### 3.1. Manual feeding

First of all, you must choose the selection mode of a Unit, of a Relation, or of a Schema, depending on what kind of annotation you wish to add to the Basket.

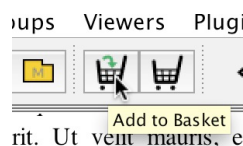
In the following example, we choose to add a Unit, so we click on the dedicated button:



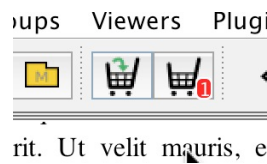
Then we click on the Unit we want to select in the main view of Glozz :



Once it is selected, we click on the "Add to Basket" icon:



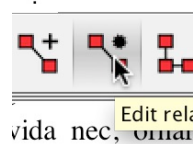
We view that the Basket icon now shows "1" as the number of its current annotations :



And, if we click on this button, we can check in the Basket that the Unit is correctly stored :

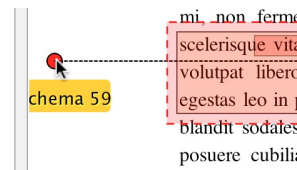


It is almost the same for Relations:



...enim possere, in  
vitae magna. Aliquam  
nim at fermentum mol  
nisl eu lectus. Fusce v  
uismod libero eu eni  
Aenean suscipit nulla i  
incididunt tincidunt mi.  
elis magna fermentun  
turabitur eu amet

And for Schemas:



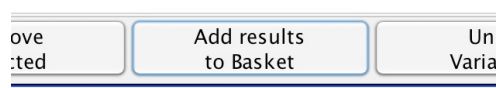
### 3.2. Feeding the Basket with GlozzQL

Please refer to the main manual of Glozz to get a comprehensive presentation of how to use GlozzQL, including how to add results to the Basket.

To recap, GlozzQL enables to query Glozz annotations with various constraints, including :

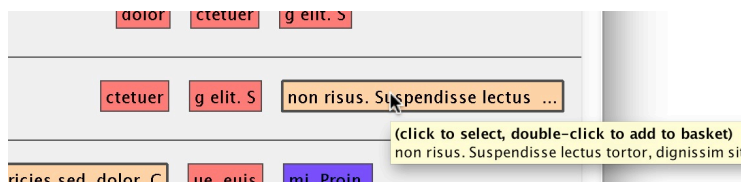
- name of author
- text content of a unit (regular expressions or full text)
- the fact that a unit of a given type covers a unit of a given type
- the fact that a relation targets a given type of annotation
- and so on...

Once you have created some ConstrainedAnnotations (which basically mean requests...) in GlozzQL, you can click on one of them to make its results appearing, and then click on the special button shown below to have them added to the Basket:



### 3.3. Feeding with the Concordancer plugin

While observing the concordancer, we can double-click on any unit shown in it to add it instantly into the Basket :

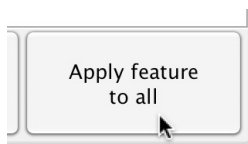


## 4. Adding or modifying features to the whole content of the Basket

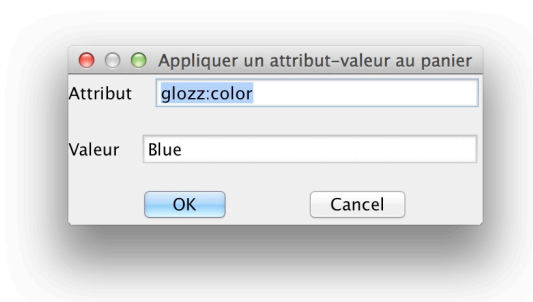
In most cases, if different annotations are added to the same Basket, it is because they do share something (either manually observed in Glozz or in the Concordancer, either automatically found by GlozzQL).

Then, it may be interesting to concretize that by adding or modifying a feature in the same way for all these annotations.

This is provided by the Basket through the following button:



Then, a dialog window asks for the feature to create or modify, and the corresponding value to apply (shown in french below, "Attribut" means "Feature", and "Valeur" means "Value"):

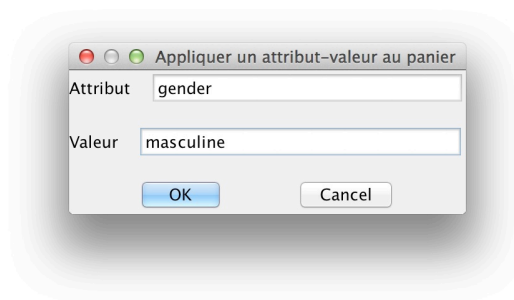


You may be surprised that the Feature name is already filled with "glozz:color", and that the Value name is already filled with "Blue". This is because this is very convenient in many cases, since Glozz is designed to automatically interpret the "glozz:color" attribute, if present, as the color to give to annotations when the special color mode is chosen. We will illustrate it further.

Of course, you are free to fill the feature name and the value as you wish. Note that from one use to the next one, the feature name and the feature value are stored by this window, to make

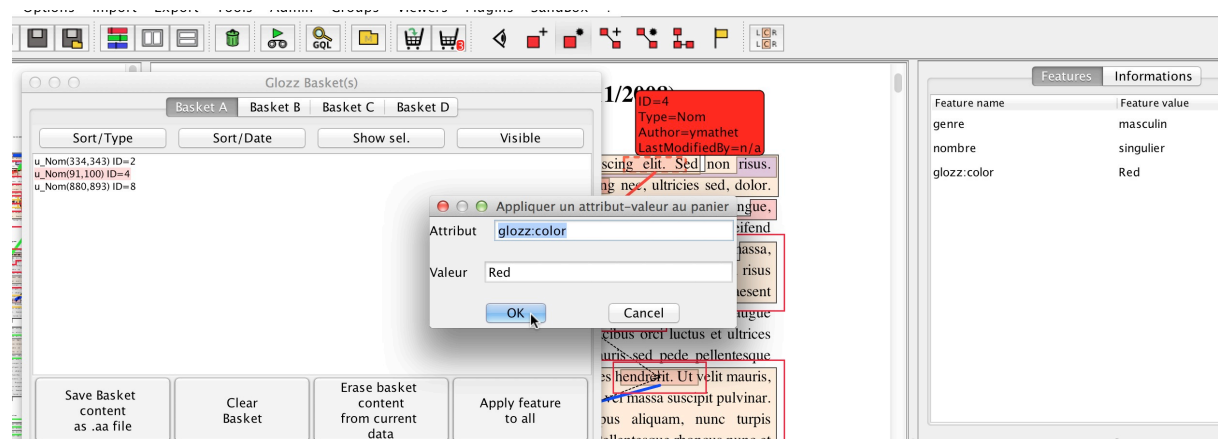


its use faster in most cases. For instance, we want to add/modify the feature "gender" to "masculine" for all the Basket this way:

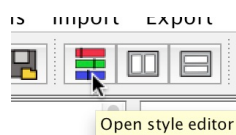


**Policy:** for a given annotation (Unit, Relation or Schema), **either the feature already exists**, in which case its value is updated with this new one, **or the feature does not already exist**, in which case the feature is created for that annotation, and filled with the given value.

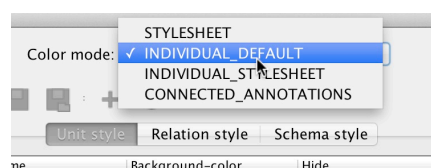
As a demonstration, we have manually added 3 units to the Basket (you can see them listed in the main Basket view, and see "3" appearing on the Basket button), then we have clicked on "Apply feature to all", then filled it with "glozz:color" and "Red". You can see the new feature "glozz:color" with value "Red" appears in the features of the currently selected unit, in the right panel of Glozz.



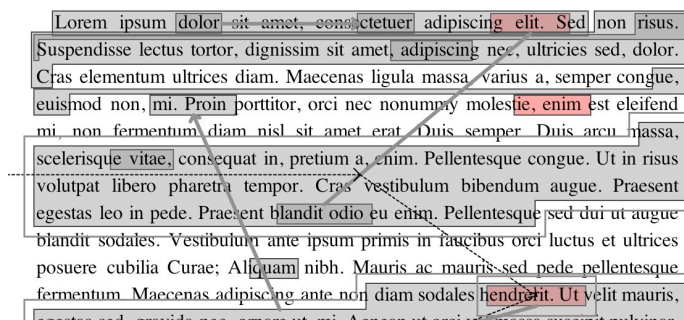
Once applied, we open the "style editor" with the following button of the toolbar of Glozz:



In the "style editor", we choose the "INDIVIDUAL\_DEFAULT" mode:



As a result, each annotation now appears with its individual color stored in its "glozz-color" feature, if this feature exists for this annotation, or in the default color otherwise. We can see that the 3 units we had selected now appear in Red color, thanks to the action of the Basket:



## 5. Multiple Baskets

Sometimes, it is more convenient to work with several Baskets at the same time, because we want to store annotations related to different phenomena in different places. In fact, this is what the Basket offers out of the box.

There are simultaneously 4 independant baskets embedded in the Basket, named "A", "B", "C", and "D".

By default, hence most of the time, the Basket works with the instance "A".

But at any moment, we can switch to another Basket, by clicking on its tab, for instance "B" as shown below:



We can see that the Basket icon now shows "B" character to indicate that we are no longer currently working with instance "A", but with instance "B" of the Basket:



Now, everything added to the Basket, whatever how it is added (manually, by GlozzQL or by the Concordancer) is added to basket "B". For instance, if we manually add one annotation into the Basket, we can see that the icon is updated as follows:



And if we click on "Basket A" tab in the Basket, then the button is updated as follows, since there are already 3 annotations in this Basket from our previous demonstration (note that "A" is not shown, since it is considered as the default Basket) :



So, at this moment, Basket A contains 3 annotations, Basket B contains 1 annotation, and C and D are empty.