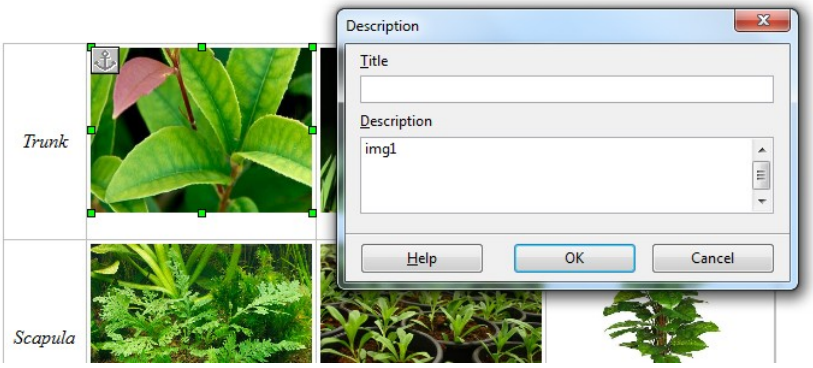
Report Generator 1.1 Tutorial (for users)

IMPORTANT! This tutorial is for users ONLY. The tutorial will not cover the development of any plug-in to create custom images or text generators, which is done in Python 2.7.

Requirements:
- Operative System: Windows XP / Vista / 7
- OpenOffice / LibreOffice (for editing templates / reports in ODT format)
- Demo data (available on the website)

1. **Download demo data.** The required files can be found in the folder *DemoDataUL*:
   - *ProcessedAndExportation / Subject1 / Session1 / bestCycles.xml*
   - *ReferenceData / RefData.xml*
   - *Reports / DescriptorsDemo.des*
   - *Reports / Template.odt*

2. **Have a look at the template first.** Open the file Template.odt with OpenOffice / LibreOffice. The pictures of different vegetations represent the placeholders for final images (data graphs). If you right-click on any picture, and choose *Description...* from the menu, a window will pop-up; the second field (*Description*) holds a custom name called *Image descriptor*. For instance, the image descriptor of the top-left image is *img1*. In the second page, you will see a table. Some cells have strings contained between << and >>. *These are Text descriptors*. The meaning of *Descriptors* will be clarified below (point 4).



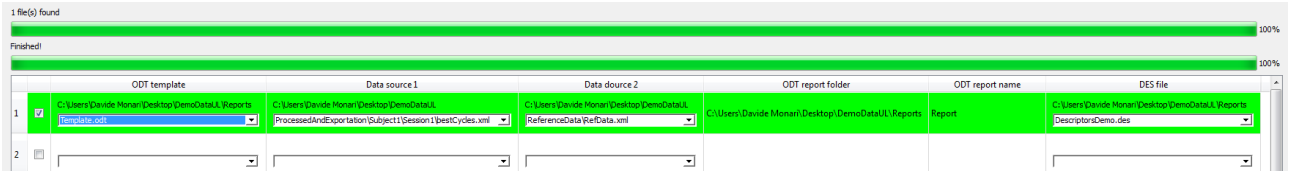
3. **Start the application.** Double-click on the executable *ReportGenerator.exe*. The main window will appear in a few seconds.

4. **Create / Edit descriptors file**. A descriptor is a string (associated to an image or placed directly into the template) that will be replaced with real data after processing the template.
   To edit descriptors, click on *Tools → Descriptors Editor*. A new window will pop-up. This will allow to create or edit descriptors files. A descriptor file (extension DES) is a file that will associate every descriptor name to a *generator* of output (image or text). You can click on *File → Load...* and select the file *DescriptorsDemo.des*. A list of descriptors will be desplayed. By selecting one,

the current options associated with it will be displayed on the left. It is possible to change descriptor name and the type of descriptor (image or text). The remaining options are delegated to plug-ins that can be coded in Python 2.7.
Report Generator is distributed with an Upper Limb plugin for managing descriptors for data coming from the U.L.E.M.A. software. The doc relative tothis plugin can be found in the folder *Plugins / UL / Doc / Guide.pdf*. You can edit the descriptors and then click on the button *Save*. If you want to save the file you can click on *File → Save* or *Save as...*and select the file path.

5. **Prepare for report creation in batch**. This can be done in the main window. Every row of the table represents the translation from a template to a final report. First of all, click on the check button on the left of the row, to activate it for later processing. Click on the *ODT template* cell and on the button *Fill from tree...,* select the folder *Reports* from the demo data folder. This will scan all subfolder from that level on to find ODT files. Select the file *Template.odt* in the pop-up menu (that has been filled in automatically with a list of found ODT files).
Repeat the same for the columns *Data source 1 / 2* and *DES file*, and now select *DemoDataUL* and *Reports*.
For *Data Source 1* and *2* select respectively the file *bestCycles.xml* and *RefData.xml*.
For *DES file* select *DescriptorsDemo.des*. Selecting the folder *DemoDataUL* for all the columns would have been ok as well.
Finally, click on the button *Choose report folder...* and select the folder where the report will be saved (e.g. *Reports*). Choose a file name for the report under the column *ODT report name*.

6. **Ready to go!** Do this by simply clicking on the button *Fill template for checked rows*. The row will first get a yellow background and after a few seconds, if everything went fine, a green one. If so, than you have your report file ready to be checked!



7. **Damn it! something is not working properly.** In the main window, you can click on *View → Log console*. The log console contains messages (and error descriptions for developers) that are generated during the usage of the software. These can be useful for later debugging. You can save the content of the log console by clicking on *File → Save ...*