Tux Gadget Format (TGF) Specifications

1. Tux Gadget Format

Tux gadget format (TGF) is the gadget format of Tux Droid gadget manager. The files named with this extension can be inserted in the TAF (Tux droid applications Framework). This format is inspired by the ODF (Open Document Format). The resulting file is a compressed structured document containing a sub files and sub directories. (The used compression is 'zip')

1.1. TGF to gadget philosophy

The TGF format is the reflective structure of the tux gadget philosophy. Each part of the file is a gadget side (audio, visual, settings, internationalization, notification, resources, etc ...). A TGF file is an offered functionality to the end user like a common gadget.

2. Directory structure

```
Root
    settings.xml
    about.xml
    strings.xml
    Strings
        <lang_country>.xml
    Scripts
        Python
            init.pyp
            main.pyp
            notify.pyp
            GUI
                conf
                    other.glade
                    other.pyp
                widget
                    other.glade
                    other.pyp
    Pictures
        Icones
            gadget.png
        GUI
    Sounds
    Data
```

3. settings.xml

3.1. Function

This file contains the parameters of the gadget. These parameters can be modified during the execution of the gadget.

3.2. Xml structure

```
<general>
    <language>en_US</language>
    <pitch>100</pitch>
    <notified>true</notified>
    <notify_delay>5</notify_delay>
    <menu_active>true</menu_active>
</general>
<parameters>
    <item>
        <name>server</name>
        <value>pop.c2me.be</value>
    </item>
    <item>
        <name>login</name>
        <value>remi.jocaille@c2me.be</value>
    </item>
</parameters>
```

4. about.xml

4.1. Function

This file contains the informations about the gadget.

4.2. Xml structure

<gadget_name>weather</gadget_name>

<gadget_author>Rémi Jocaille</gadget_author>

<gadget_version>0.0.1<gadget_version>

<gadget_description>Weather forecast gadget for Tux Droid</gadget_description>

5. Strings

5.1. strings.xml

a) Function

This file contains the canvas of the gadget strings and a list of the supported languages.The message values can be a string or a list of strings. When the value is a list of strings, the taken value on use will be selected randomly into it.

5.2. Xml structure

<strings>

<name_to_read type='str'>Tuxgi</name_to_read>

<test_simple_string type='str'>My string</test_simple_string>

<test_list_string type='list'>[“My string”, “Your string”]</test_list_string>

</strings>

5.3. Strings/<lang_country>.xml

a) Function

This file contains the string translations for a specified language. For example, for French, this file will be named 'fr_FR.xml'. The items in the file is in relation with the canvas specifications in 'strings.xml' file.

5.4. Xml structure

```
<strings>
    <name_to_read type='str'>Tuxgi</name_to_read>
    <test_simple_string type='str'>Ma chaine</test_simple_string>
    <test_list_string type='list'>[“Ma chaine”, “Ta chaine”]</test_list_string>
</strings>
```

6. Scripts/Python

6.1. init.pyp

Init.pyp (PYthon Part) contains the initializing code of the gadget. This script is executed when the gadget object is created.

6.2. main.pyp

Main.pyp contains the main python code of the gadget. It's the 'voiceget' side of the gadget. In this script you must describe the voice part behavior of the gadget. You also can define a function named “update_informations” which will overload the same function in the gadget object.

6.3. notify.pyp

This file contains the python code of the notification system. You must define two functions in this file: “notify_checker” and “notify_actuator”. The “notify_checker” function is executed like a timer. The delay of the timer is defined by the general setting named “notify_delay”. The return of this function must be a boolean which will indicate if the “notify_actuator” will be executed.

7. Scripts/Python/GUI

7.1. Function

The 'GUI' directory contains the needed graphical user interfaces by your gadget. A GUI is defined in a sub directory which contains two files: 'other.glade' and 'other.pyp'. The glade file is the graphical declaration and the pyp file is the python script which controls the graphical parts of the GUI. Two GUI must be implemented by default. The 'conf' GUI which controls the gadget settings and the 'widget' GUI which defines the visual side of gadget.

Comment: don't interact directly with the tux resources (Tux droid and TTS) on the script part of a GUI because it would be an error in the gadget philosophy.

8. Pictures/Icones

8.1. gadget.png

This image is the icon showed in the manager and the gadget GUI.

9. Pictures/GUI

This directory contains the needed pictures at the gadget GUI.

10. Sounds

This directory contains the external sounds played by the gadget.

11. Data

This directory contains the miscellaneous data needed by the gadget. For example, you can put into it the list of checked emails, user playlists for an audio player, etc ...