

PyORTB VL

Outils pour OpenRails – Version Alpha 0.02 – 2016/07/24

Auteurs : BB25187 - <http://BB25187.eu>

PyORTB VL

Outils pour OpenRails

1 Sommaire

2	Le projet PyORTB.....	3
2.1	Constat et objectif	3
2.2	Une première version traitant les fichiers sonores « SMS »	3
2.3	La suite : les fichiers « WAG » et « ENG »	3
2.4	Les choix d'implémentation	3
2.5	Public visé	4
3	Installation.....	5
3.1	Installer Python 2.7 sur votre PC.....	5
3.2	Téléchargement.....	5
3.2.1	Installation.....	5
3.2.2	Positionnement des variables d'environnement	5
3.3	Installer PyORTB	5
4	Composition de l'outil PyORTB.....	6
4.1	Les fichiers de l'outil.....	6
4.1.1	pyortb_process_sms.pyc.....	6
4.1.2	pyortb_process_sms_tk.pyc.....	6
4.1.3	pyortb_guitxt.py.....	6
4.1.4	pyortb_process_sms_main.pyw.....	6
4.2	L'interface.....	7
4.2.1	Présentation générale	7
4.2.2	La zone de sélection	7
4.2.3	La zone d'information	8
4.2.4	La zone d'options.....	8
4.2.5	La zone d'actions	9
5	Utilisation de PyORTB.....	11
5.1	Sélectionner un répertoire ou un fichier.....	11
5.2	Choisir des options de traitement.....	12
5.3	Lancer et suivre le traitement	13

5.4	En cas d'erreur.....	14
5.4.1	Manifestation au niveau de l'interface graphique	14
5.4.2	Le fichier de log	15
5.4.3	Les erreurs émises dans les fichiers traités	17
6	Limitations connues de la version actuelle	17
7	Droits et copyrights	17
8	Contributions.....	17
9	Outils utilisés	18
10	Contact	18
11	Historique des versions	18

2 Le projet PyORTB

2.1 Constat et objectif

Je ne sais pas si vous avez remarqué, mais dans la construction, l'adaptation ou la modification de modèles, nous sommes amenés à manipuler pas mal de types de fichiers différents : fichiers ENG/WAG, fichiers SMS, fichier CON, ... Or, ceux-ci sont plus ou moins bien écrits et présentés. Souvent, ils utilisent des règles de mise en forme mal définies ou hétérogènes, ce qui les rend difficiles à lire, à comprendre, à modifier, à comparer, à corriger en cas de bugs. Quand on aime bosser sur de la matière bien propre, il est donc fréquent de devoir remettre les fichiers sous une forme lisible et systématique avant de pouvoir bosser dessus de manière sérieuse. C'est long et fastidieux à faire, et ce n'est pas très intéressant. Or, ce genre de remise en forme est typiquement le genre de tâche qui peut être automatisée sans trop de difficulté. C'est ce qu'on appelle couramment en anglais un « *pretty printer* », autrement dit un outil qui améliore et systématise la présentation d'un fichier.

Partant de là, on peut imaginer d'aller plus loin. En effet, si l'on sait mettre un fichier dans une forme propre, c'est qu'on a été capable de le lire et d'en extraire la structure. On n'est pas très loin alors de pouvoir l'analyser, de pouvoir détecter des erreurs ou des motifs indésirables en son sein, voire de l'enrichir, par exemple de commentaires automatiques ou de paramètres particuliers.

2.2 Une première version traitant les fichiers sonores « SMS »

Le premier outil proposé concerne les fichiers SMS, qui sont parmi les plus mal présentés et les plus illisibles de tout le petit monde MSTs/ORTS. La première version de l'outil se limite à la première étape évoquée ci-dessus, à savoir celle du simple "*pretty printer*". Il s'agit donc de mettre les fichiers dans une forme lisible et systématique au niveau des indentations, du bannissement des caractères de tabulation (qui ne sont absolument pas portables et ne garantissent pas la préservation de l'alignement initial), de la position des mots clefs, des parenthèses, ...

Ce premier outil permettra aussi de valider la méthode d'écriture et de diffusion envisagée.

2.3 La suite : les fichiers « WAG » et « ENG »

Viendront ensuite les fichiers ENG/WAG viendront ensuite. Là par contre, il y a plein de petits trucs sympas qu'on peut imaginer en matière de ré-ordonnancement des paramètres pour les mettre dans un ordre logique et cohérent, d'analyse, de détection de problèmes potentiels, d'enrichissement, ... voir peut-être plus.

2.4 Les choix d'implémentation

Pour des questions de facilité et de souplesse, c'est le langage Python qui a été retenu, car il présente plusieurs avantages :

- Il propose de nombreux modules d'extension
- Il repose sur une représentation orientée objet qui est très bien adaptée pour le traitement que j'envisage - surtout quand il va s'agir de traiter des fichiers aussi riches que les fichiers ENG
- Il est aisément portable d'une plateforme à une autre (x86, x86-64, Linux, ...)
- Il ne demande que l'installation préalable de Python sur votre PC, qui est simple et robuste

- Il rend les scripts concernés très faciles à appeler (sous réserve que les bonnes options aient été choisies lors de l'installation de Python !) : un double clic suffit.

2.5 Public visé

Ces outils sont plus particulièrement destinés ceux qui mettent les mains dans le cambouis à l'occasion du développement de modèles de matériels roulants. Ils intéresseront cependant très probablement d'autres utilisateurs.

3 Installation

3.1 Installer Python 2.7 sur votre PC

3.2 Téléchargement

Il vous faudra télécharger l'installateur de Python 2.7 sur le site de Python :

<https://www.python.org/downloads/>.

Attention à bien télécharger et installer la version 2.7 et non la 3.5.

3.2.1 Installation

L'installation de Python est triviale, mais il est recommandé de laisser le chemin proposé par défaut pour l'installation, à savoir `C:\Python27`. Si vous essayer de l'installer dans un sous-répertoire des lieux habituels (`C:\Program Files(x86)` pour la version 32b ou `Programs` pour la version 64b), il est possible que vous rencontriez ensuite des problèmes à l'exécution avec des programmes multi-modules.

3.2.2 Positionnement des variables d'environnement

Attention : pour utilisateurs avertis seulement !

Pour garantir une exécution sans problème de n'importe quel programme Python, il est recommandé d'ajouter les répertoires d'installation de Python à la variable `PATH` de votre compte. Pour cela il faut suivre la démarche suivante :

- Ouvrir le panneau de configuration, puis Système et sécurité, puis Système, puis Paramètres système avancés.
- Dans le panneau qui s'ouvre, cliquer sur le bouton Variables d'environnement en bas à droite
- Dans la liste de variables de votre compte, sélectionner la variable `PATH`, puis cliquer sur *Modifier* (ou *Nouvelle* si `PATH` n'existe pas encore).
- Dans le panneau qui s'ouvre, ajouter à la variable `PATH` déjà définie les éléments suivants, en les plaçant en tête de chaîne : `C:\Python27;C:\Python27\Scripts`. Si la variable était vide avant modification, elle devrait apparaître comme suit après :

```
"C:\Python27";"C:\Python27\Scripts"
```

- Cliquer autant de fois que nécessaire sur les boutons de validation de vos modifications.

3.3 Installer PyORTB

L'installation des outils PyORTB se fait en décompressant l'archive et en plaçant le répertoire `PyORTB` à l'endroit de votre choix. Aucune modification n'est nécessaire dans la base de registre de votre PC.

4 Composition de l'outil PyORTB

4.1 Les fichiers de l'outil

L'outil de traitement des fichiers SMS est composé de plusieurs fichiers décrits ci-après.

4.1.1 `pyortb_process_sms.pyc`

C'est le fichier qui implémente les traitements proprement dits. Il est livré sous forme de fichier précompilé. A noter que ce script est accessible en ligne de commande, mais aucune documentation autre que l'aide en ligne n'est fournie en dehors de l'option d'aide en ligne. Les options sont toutefois très proches de celles qui permettent de contrôler les traitements au travers de l'interface graphique.

```
usage: pyortb_process_sms.pyc [-h] [-f FILE_SMS] [-v] [-n] [-k]

optional arguments:
  -h, --help            show this help message and exit
  -f FILE_SMS, --file_sms FILE_SMS
                        Specifies the file to process
  -v, --verbose          Dumps additional information on stdout
  -n, --nooutput         Does not write final output files (dryrun)
  -k, --keep            Keep intermediate files
```

4.1.2 `pyortb_process_sms_tk.pyc`

C'est le fichier qui définit la structure de l'interface graphique de l'outil. Il est livré sous forme de fichier précompilé. Cette interface s'appuie sur le module standard *Tkinter* de Python.

4.1.3 `pyortb_guitxt.py`

C'est le fichier qui définit les textes principaux de l'interface graphique. Il est livré sous forme de fichier source, ce qui permet d'adapter l'interface à différent langages. A noter toutefois que les textes du traitement proprement-dit ne sont pas modifiables !

4.1.4 `pyortb_main.pyw`

C'est le fichier racine de l'outil, celui que l'utilisateur doit donc lancer. Il est livré sous forme de fichier source.

4.2 L'interface

4.2.1 Présentation générale

L'interface est volontairement simple, et se présente comme illustré sur l'image ci-dessous.

Cette interface est réalisée en anglais, mais le nombre de termes utilisé est réduit et ne devrait pas poser de problème, surtout après avoir lu cette documentation ! Les textes de cette interface sont cependant modifiables en éditant le fichier `pyortb_guitxt.py` mentionné ci-dessus.

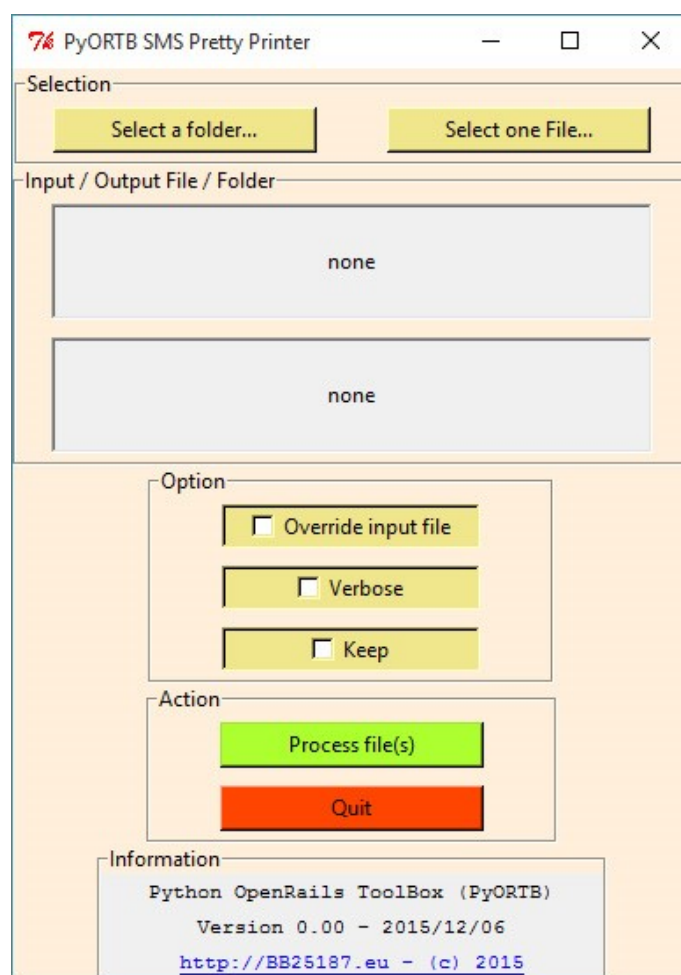


Image 1: L'interface de PyORTB

4.2.2 La zone de sélection

En haut se trouve la zone de sélection des fichiers à traiter. Deux boutons sont proposés :

- La sélection d'un répertoire (*Select a Folder*). Tous les fichiers SMS contenus dans le répertoire sélectionné sont traités.
- La sélection d'un fichier unique (*Select a file*). Un seul fichier est traité.

Le choix de traiter un répertoire est exclusif de celui de traiter un fichier.

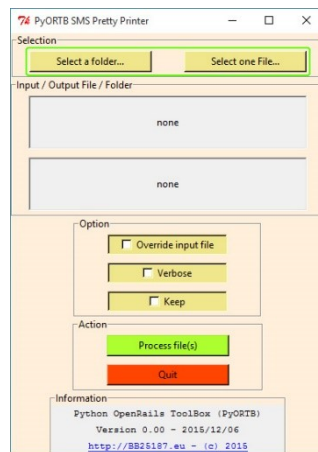


Image 2: La zone de sélection des répertoires et fichiers

4.2.3 La zone d'information

En dessous de la zone de sélection des répertoires et des fichiers se trouve une zone d'information. Celle-ci contient deux panneaux d'affichage textuel qui fournissent des informations sur les fichiers d'entrée et de sortie du traitement.

Après la sélection, ces panneaux indiquent le répertoire ou le fichier choisi.

Durant le traitement, le nom des fichiers traités est affiché au fur et à mesure.

Après le traitement, ces panneaux fournissent des informations sur son bon déroulement mais peuvent également contenir des informations concernant d'éventuelles erreurs.

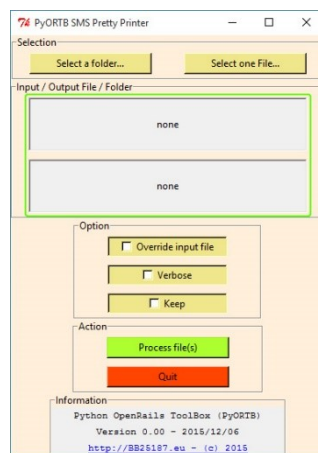


Image 3: La zone d'information

4.2.4 La zone d'options

Le traitement est contrôlé par un nombre réduit d'options.

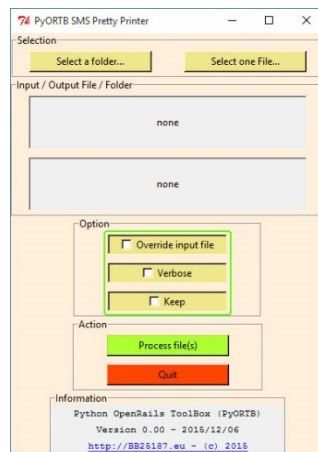


Image 4: La zone d'options

4.2.4.1 L'écrasement des fichiers d'entrée (*Override input file*)

Durant le traitement, les fichiers traités sont enregistrés par défaut avec un nom différent des fichiers d'origine par une extension additionnelle *new*. Par exemple : le fichier *EnginCab.sms* est enregistré dans *EnginCab.sms.new* après traitement. Les fichiers d'origine sont alors préservés. Cette méthode peut toutefois être fastidieuse lors du traitement de fichiers par lots.

La sélection de l'option d'écrasement des fichiers d'entrée (*Override input file*) permet de recopier le fichier traité dans le fichier d'origine à la fin du traitement. Le fichier d'origine est alors perdu !

Noter cependant que si une erreur est détectée dans le fichier durant le traitement, alors cette recopie n'est pas effectuée. Le fichier d'origine reste alors inchangé, et le fichier modifié reste disponible pour analyse sous le nom utilisé lorsque l'option d'écrasement n'est pas sélectionnée (extension additionnelle *new*)

4.2.4.2 Le mode verbeux (*Verbose*)

Cette option n'a en principe pas d'utilité pour un usage normal. Elle affiche en effet des informations détaillées sur le déroulement du traitement dans la fenêtre de l'interpréteur Python, qui n'apparaît que si vous renommez le fichier *pyortb_process_sms_tk.pyw* en *pyortb_process_sms_tk.py*. Sans ce renommage, l'option n'a pas d'effet particulier.

4.2.4.3 La préservation des fichiers temporaires (*Keep*)

Le traitement est décomposé en plusieurs étapes, qui utilisent des fichiers temporaires. Ceux-ci sont enregistrés dans le répertoire dans lequel se trouve le ou les fichiers à traiter. Ils sont normalement détruits au fur et à mesure que le traitement progresse. Il est toutefois possible de les conserver au moyen de l'option de préservation des fichiers temporaires (*Keep*).

Cette préservation peut être utile pour analyser des problèmes rencontrés durant le traitement.

4.2.5 La zone d'actions

En bas de l'interface se trouve la zone d'action. Celle-ci comporte deux boutons qui permettent :

- De lancer le traitement (*Process*)

- De quitter l'outil (*Quit*)

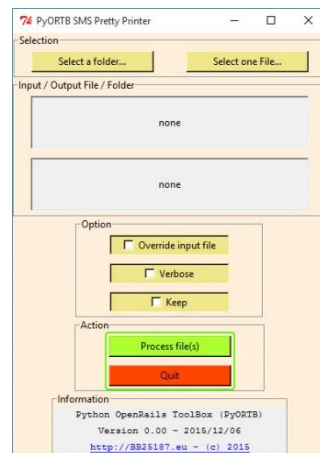


Image 5: La zone d'actions

Notez que ces boutons n'ont aucun effet si aucun répertoire ni aucun fichier n'ont été sélectionnés au préalable. Par ailleurs le fait de cliquer sur le bouton de lancement du traitement invalide certains éléments de l'interface (boutons et options).

5 Utilisation de PyORTB

5.1 Sélectionner un répertoire ou un fichier

Il faut cliquer soit sur le bouton de sélection d'un répertoire (*Select a folder...*), soit sur le bouton de sélection d'un fichier (*Select a file...*). Un menu s'ouvre alors, qui permet de sélectionner le répertoire ou le fichier à traiter.

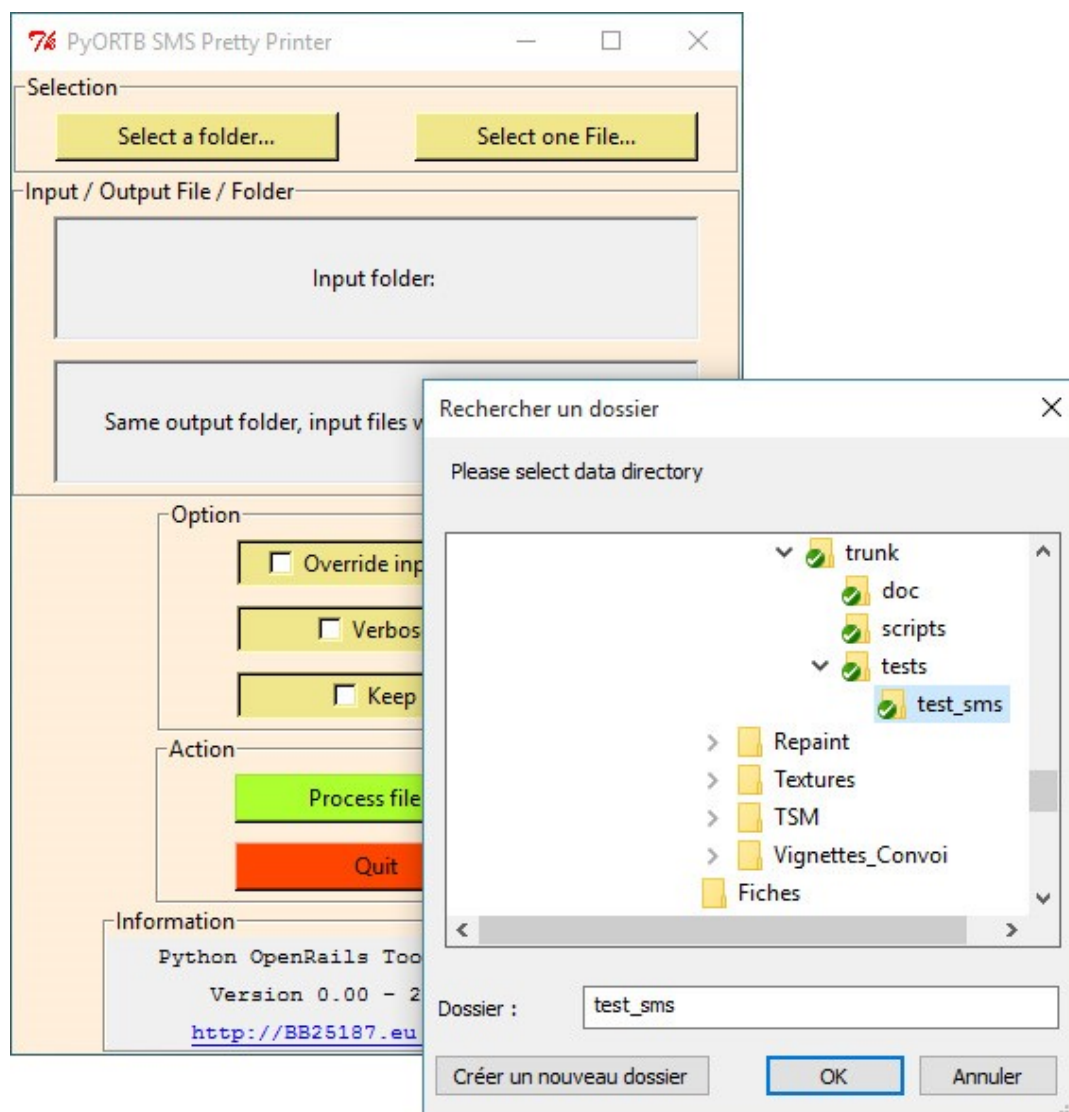


Image 6: Sélection d'un répertoire

Il faut alors valider le choix (*OK*).

Le nom de l'élément sélectionné apparaît alors dans la fenêtre d'information supérieure. Des informations complémentaires peuvent être affichées dans la fenêtre d'information inférieure. Ces informations dépendent des options sélectionnées.

5.2 Choisir des options de traitement

Il suffit de cocher les options choisies dans la partie de l'interface réservée à cet effet.

A noter qu'il est possible de sélectionner les options de traitement avant ou après avoir sélectionné le répertoire ou le fichier à traiter, et avant de valider le lancement du traitement.

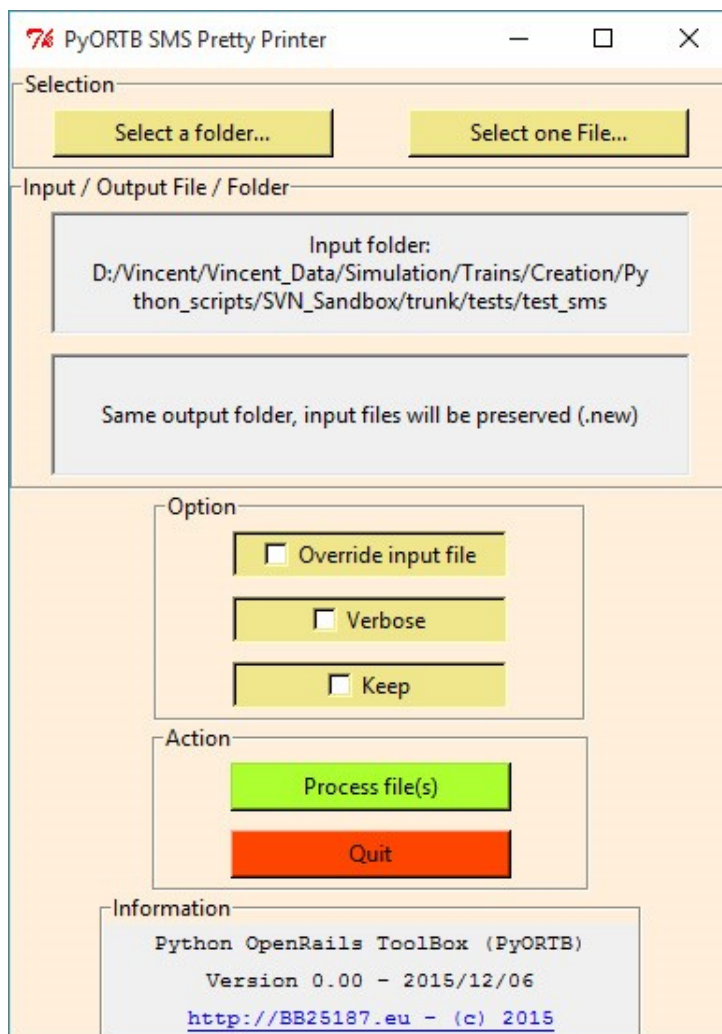


Image 7: Après sélection d'un répertoire

5.3 Lancer et suivre le traitement

Le lancement du traitement s'effectue en cliquant simplement sur le bouton correspondant (*Process file(s)*). Les informations dynamiques de l'interface permettent de suivre le déroulement du traitement, et notamment le nom du fichier en cours de traitement. Un affichage exhaustif n'est toutefois pas garanti.

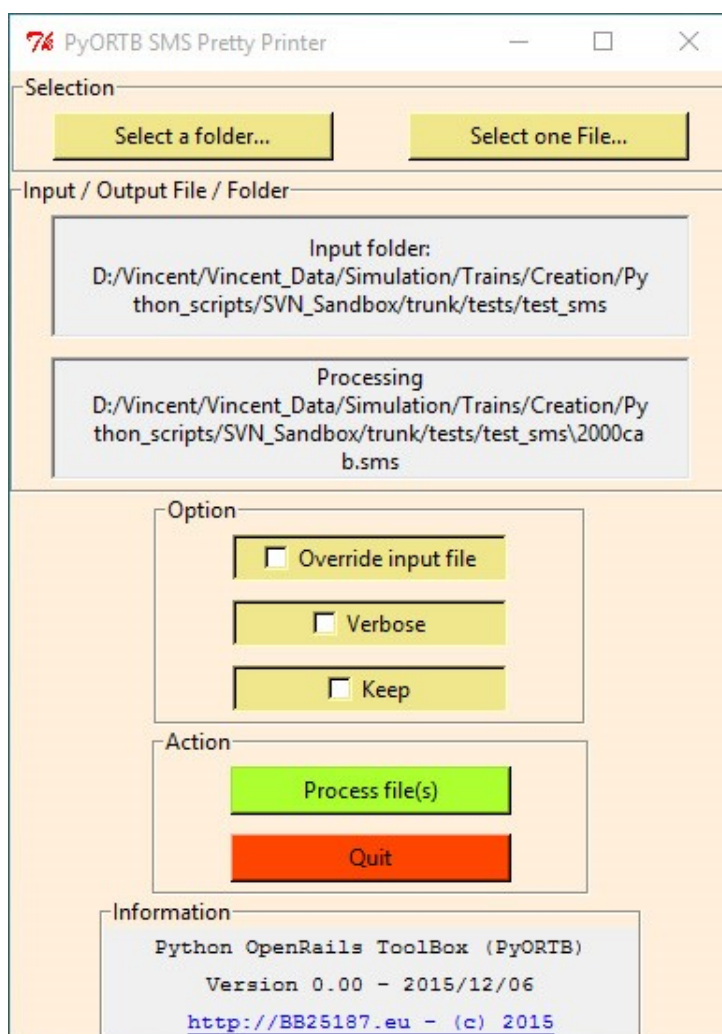


Image 8: En cours de traitement

5.4 En cas d'erreur

5.4.1 Manifestation au niveau de l'interface graphique

En cas d'erreur ou de problème, une fenêtre d'alerte spécifique s'affiche et un son est émis à la fin du traitement. Il faut valider cliquer sur le bouton de validation de cette fenêtre pour pouvoir continuer. Le code affiché est un code interne qui peut être utile pour comprendre la cause du problème.

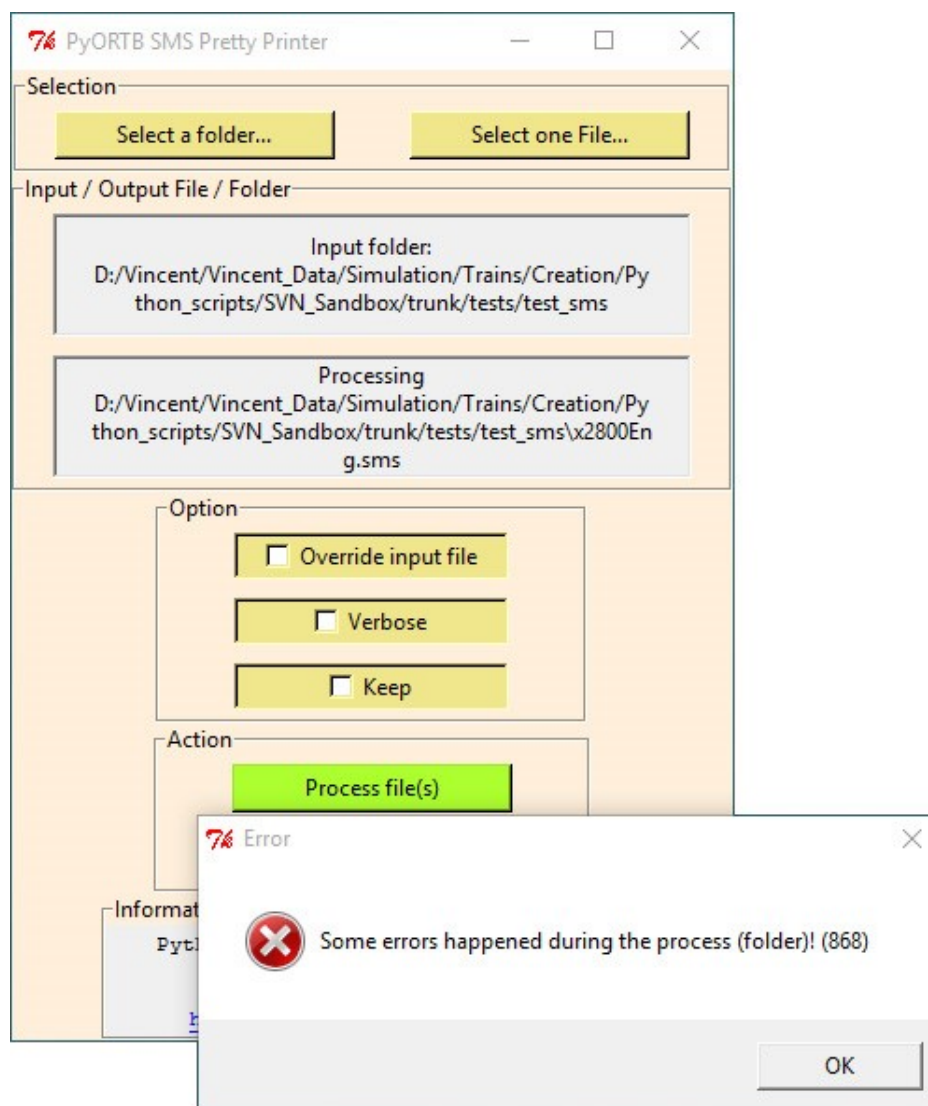


Image 9: Cas d'erreur

Après avoir cliqué sur le bouton de la fenêtre d'alerte, le processus se termine, et des informations complémentaires sont affichées sur l'interface. On peut y trouver la liste des fichiers qui ont provoqué une erreur de traitement.

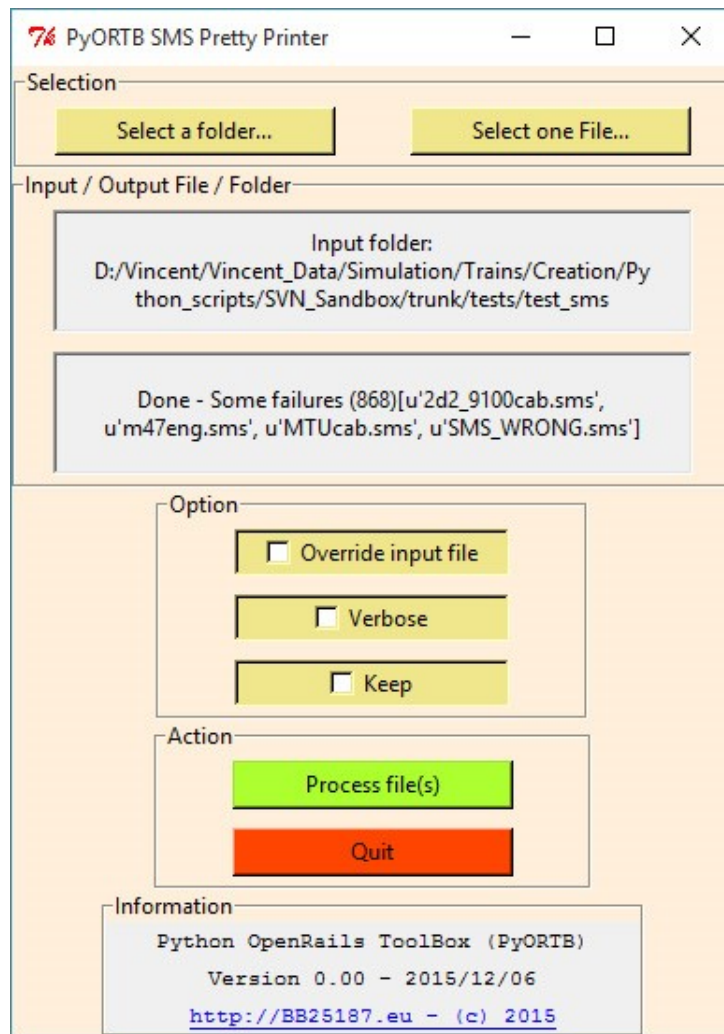


Image 10: En fin de traitement

Par ailleurs, deux moyens complémentaires permettent d'analyser ces erreurs. Il s'agit du fichier de log et des messages d'erreur insérés dans les fichiers traités.

5.4.2 Le fichier de log

Le fichier de log est créé dans le même répertoire que celui de l'outil. Il contient des informations détaillées sur le déroulement du traitement.

5.4.2.1 L'entête

L'entête du fichier rappelle le nom du répertoire ou du fichier traité, ainsi que les options choisies.

```
INFO:root:*****
INFO:root:* Pretty printer for MSTs / ORTS SMS sound files *
INFO:root:* (c) http://BB25187.eu 11/2015 - Version 0.00 *
INFO:root:*****
INFO:root:directory D:/Vincent/.../tests/test_sms
INFO:root:verbose=0
INFO:root:killinput=1
INFO:root:keepfiles=0
INFO:root:file=None
INFO:root:Input file
```

Exemple 1 : Entête du fichier de log

Ces valeurs doivent correspondre à celles entrées sur l'interface graphique en début de processus.

5.4.2.2 Déroulement du traitement pour un fichier ne provoquant pas d'erreur

Pour un fichier qui ne provoque pas d'erreur, seuls sont consignés les événements de recopie de fichiers.

```
D:/Vincent/.../tests/test_sms\1018Cab.sms renamed as
D:/Vincent/.../tests/test_sms\1018Cab.sms.back
INFO:root:Output file D:/Vincent/.../tests/test_sms\1018Cab.sms.new
renamed as D:/Vincent/.../tests/test_sms\1018Cab.sms
INFO:root:Process done. Output file is
D:/Vincent/.../tests/test_sms\1018Cab.sms
```

Exemple 2 : Contenu du fichier de log pour le fichier 1018Cab.sms ne causant pas d'erreur

5.4.2.3 Détection d'erreurs dans des fichiers traités

En cas d'erreur dans des fichiers traités, des messages d'avertissement et/ou d'erreur sont émis dans le fichier de log. Ces messages permettent de retrouver le nom du fichier incriminé, ainsi que la première ligne qui cause ce problème. Les trois exemples ci-dessous illustrent de tels cas.

```
WARNING:root:Line 197: Level discrepancy Triggers 3!=4
INFO:root:Process done. Output file is
D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\2d2_9100cab.sms.new
ERROR:root:Some errors during execution
(D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\2d2_9100cab.sms - 197)
```

Exemple 3 : Erreur dans le fichier 2d2_9100cab.sms, ligne 197

```
WARNING:root:Line 570: Level discrepancy ScalabiltyGroup 2!=1
INFO:root:Process done. Output file is
D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\m47eng.sms.new
ERROR:root:Some errors during execution
(D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\m47eng.sms - 570)
```

Exemple 4 : Erreur dans le fichier m47eng.sms, ligne 570

```
WARNING:root:Line 100: Level discrepancy Stream 1!=3
INFO:root:Process done. Output file is
D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\MTUcab.sms.new
ERROR:root:Some errors during execution
(D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\MTUcab.sms - 100)
```

Exemple 5 : Erreur dans le fichier MTUcab.sms, ligne 100

```
WARNING:root:The file does not seem to be a legal sms file:
D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\SMS_WRONG.sms
ERROR:root:Some errors during execution
(D:/Vincent/Vincent_Data/Simulation/Trains/Creation/Python_scripts/PYORTB_
Sandbox/trunk/tests/test_sms\SMS_WRONG.sms - 1)
```

Exemple 6 : Erreur dans le fichier SMS_WRONG.sms, non identifié comme fichier SMS

5.4.3 Les erreurs émises dans les fichiers traités

Lorsqu'une première erreur est détectée dans un fichier, l'outil insère un message d'erreur afin de faciliter l'analyse. La ligne d'insertion correspond au numéro de ligne mentionné dans l'erreur du fichier de log. Ce fichier ne peut alors fonctionner correctement dans les simulateurs. Toutefois, il faut noter que les erreurs trouvées empêchent en général leur bon fonctionnement, et cela même sans ajout de message d'erreur.

Le message prend la forme suivante :

```
Skip (** First error found in line below >>> **)
```

Il est en général assez simple de trouver les causes de l'erreur en inspectant le fichier autour de cette ligne. Le plus souvent, il s'agit d'erreurs de parenthèse.

6 Limitations connues de la version actuelle

La version actuelle ne traite pas les commentaires multi-lignes, qui conduisent à l'émission d'une erreur en cours de traitement.

7 Droits et copyrights

Petits rappels, jamais superflus, relatifs, entre autres, à l'attachement affectif d'un auteur pour ses créations :

- L'usage de cet outil est libre. Les scripts, documentations, éléments d'installation qu'il contient sont fournis à titre entièrement gratuit. Ils ne peuvent en aucun cas être vendus ni faire l'objet de négociations de nature commerciale.
- Merci de ne pas modifier, réutiliser totalement ou partiellement les éléments livrés avec cet outil ou l'outil lui-même sans mon accord explicite.
- La diffusion de cet outil sur d'autres sites que <http://BB25187.eu> n'est pas autorisée sans mon accord explicite.

Bref, ne confondons pas Freeware et foire à la brocante ! C'est d'ailleurs ce que suggère également une excellente source, qu'il convient de lire avec toute l'attention requise :

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/>

Les auteurs de l'outil déclinent toute responsabilité en cas de dommage causé par l'installation ou l'utilisation du présent contenu sur le matériel, le système d'exploitation, les logiciels ou un quelconque élément de l'ordinateur des utilisateurs. Je vous rassure cela reste très improbable dans le cadre d'un usage raisonnable !

8 Contributions

Le tableau ci-dessous résume les contributions des différents auteurs.

Tâche / étape	Auteurs et contributeurs
Ecriture des scripts	BB25187
Documentation	BB25187
Installeur	BB25187

Tableau 1: Contributions

9 Outils utilisés

Les outils utilisés pour la réalisation de ce modèle sont mentionnés dans le tableau ci-dessous.

Tâche / étape	Outil
Scripts	Python 2.7
Edition de texte	Notepad++ (Don Ho / Notepad)
Edition de texte – Différences	CompareIt (GrigSoft)
Documentation	MS Office (Microsoft)
Installeur	Install Creator Pro (ClickTeam)

Tableau 2: Outils utilisés

10 Contact

Voir sur <http://BB25187.eu/>, ou par message privé (MP) le forum [Activity Simulator World \(ASW\)](#).

11 Historique des versions

Version	Date	Commentaires
Alpha-01	2016/04/10	Première diffusion.
Alpha-02	2016/07/24	Ajout du contrôle Dist_Min_Max

Tableau 3: Historique des versions