

Manuel Prospéro II interface python-Qt

Josquin Debaz

13 avril 2015

Table des matières

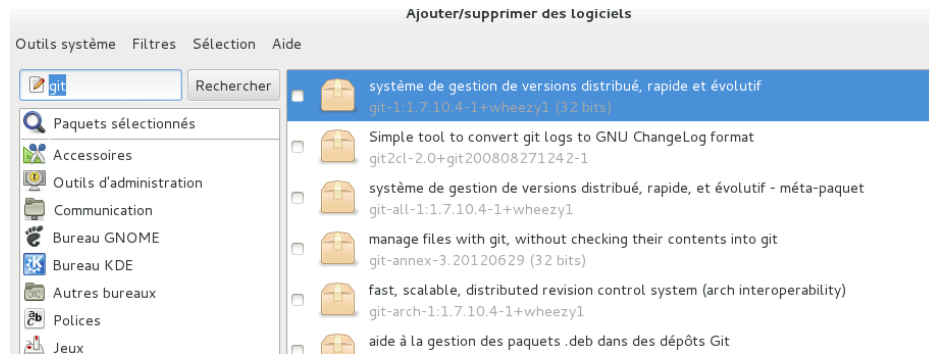
1	Récupérer Prospéro 2	2
1.1	Git pour Gnu/Linux	2
1.2	Git pour windows	2
1.3	Qt pour Gnu/Linux	3
1.4	Qt pour Windows	3
2	Mettre à jour la version de P2	4
2.1	Pour Windows	4
2.2	Pour Gnu/Linux	4
3	Démarrer Prospero 2	4
3.1	Pour Gnu/Linux et Mac	4
3.2	Pour Windows	4
3.3	Connecter l'interface au serveur	4
4	le cadran haut-gauche	5
4.1	Lexicon et Concepts	5
4.2	Search	5
4.3	Metadata	5
5	le cadran bas-gauche	6
6	le cadran bas-droit	6
6.1	Properties	6
6.2	Metadata	6
6.3	Text	6
7	le cadran haut-droit	6
7.1	Variant generation	6
8	le codex	8

1 Récupérer Prospéro 2

Il faut tout d'abord télécharger et installer git

1.1 Git pour Gnu/Linux

Installer git via votre gestionnaire de paquet



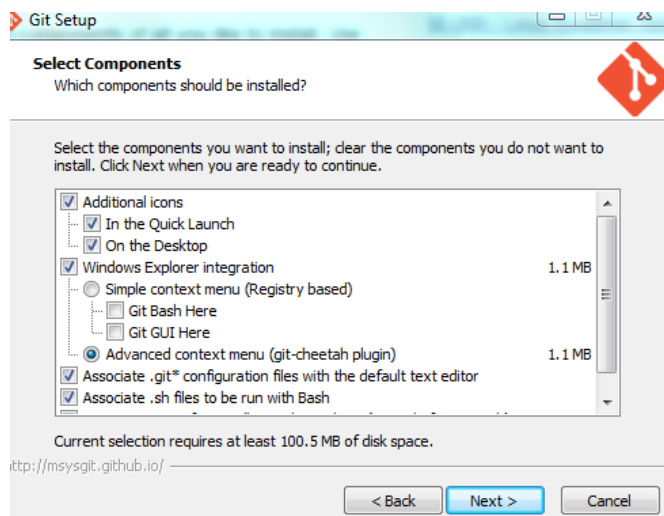
Dans un terminal, cloner le dépôt dans le répertoire voulu.

```
git clone https://github.com/josquindebaz/P2Qt.git
```

1.2 Git pour windows

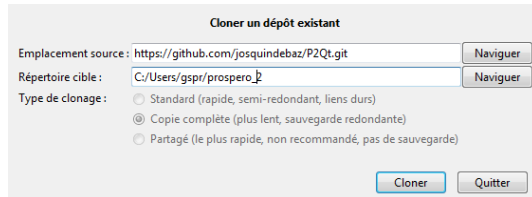
<http://msysgit.github.io/>

sélectionner l'ajout d'icône dans Quick Launch



Lancer Git Gui qui doit être dans le menu démarrer

Choisir cloner un dépôt



La source est <https://github.com/josquindebaz/P2Qt.git>

1.3 Qt pour Gnu/Linux

En principe Python est installé par défaut. Il faut également installer la librairie graphique Pyside, via votre gestionnaire de paquet en cherchant pyside et qtgui.



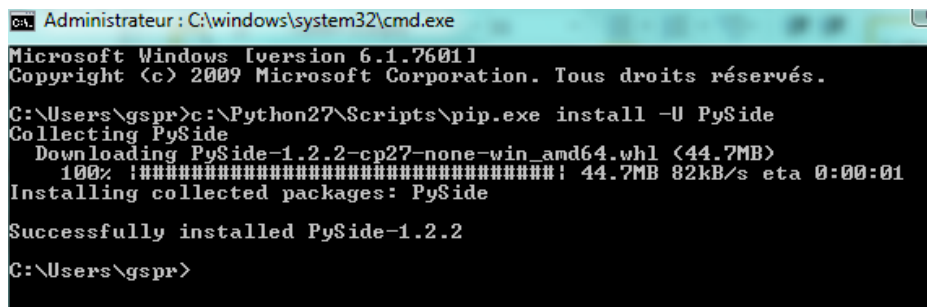
1.4 Qt pour Windows

Il vous faut installer Python : choisir la version 2.7 la plus récente, en 64bits en principe si votre machine n'est pas encore en 32 sinon prendre la version générale <https://www.python.org/downloads/windows/>

il faut également installer la librairie graphique : pour cela, on passe par pip, téléchargez <https://bootstrap.pypa.io/get-pip.py>, et exécutez le ou vrez ensuite un terminal (affichez le menu démarrer et tapez cmd, vous verrez apparaître l'icône cmd.exe) dans ce terminal, lancez la commande

```
c:\Python27\Scripts\pip.exe install -U PySide
```

si vous avez précisé un autre chemin pour l'installation de python ou de pip, il faudra évidemment adapter la commande



2 Mettre à jour la version de P2

On mettra à jour sa version grace à git.

2.1 Pour Windows

Pour cela cliquer droit dans le répertoire où est installé p2 et choisir Git Gui. A la première utilisation, utilisation il faut renseigner le champ adresse email dans Edition > Options

1. D'abord récupérer les dernières modifications : dans l'onglet Dépôt distant, choisir Récupérer de : origin
2. Puis fusionner dans le répertoire local : dans l'onglet Fusionner, choisir Fusion locale.

2.2 Pour Gnu/Linux

Dans un terminal, dans le répertoire voulu, taper :

```
git pull
```

3 Démarrer Prospero 2

3.1 Pour Gnu/Linux et Mac

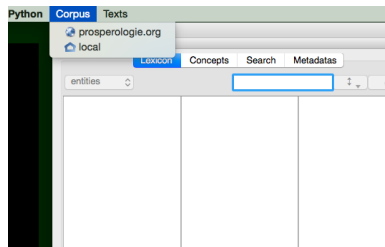
Dans un terminal, dans le répertoire voulu taper :

```
python p2gui.py
```

3.2 Pour Windows

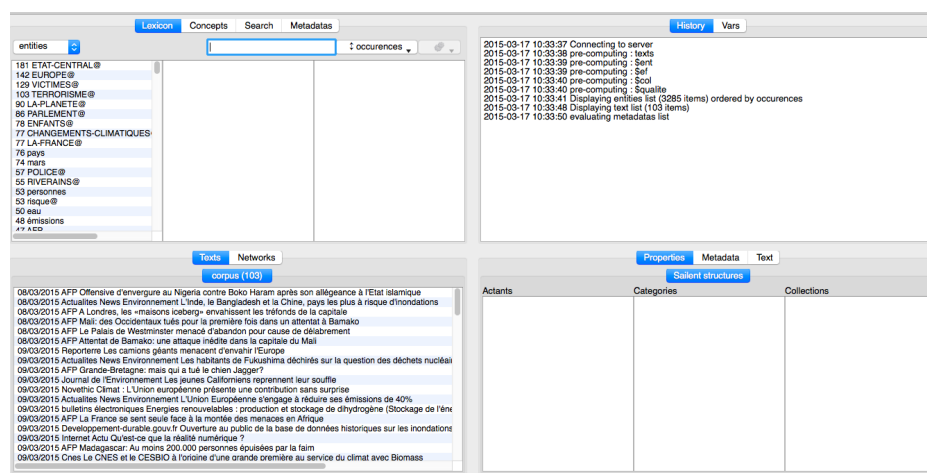
Pour utiliser Prospero 2, double cliquer sur le fichier p2gui.py

3.3 Connecter l'interface au serveur



Pour se connecter à un serveur distant, choisir prosperologie.org dans le menu Corpus. Si le serveur et le réseau fonctionnent, la connexion depuis l'interface

devrait se réaliser rapidement. Après quelques secondes de précalculs (la barre de progression en bas à droite indique leur avancée), l'interface affichera la liste des entités dans le cadran haut-gauche et la liste des textes dans le cadran bas-gauche.



4 le cadran haut-gauche

Dans ce cadran, 4 onglets sont proposés : Lexicon, Concepts, Search et Metadata.

4.1 Lexicon et Concepts

Le classement des éléments lexicaux et des concepts se fait par nombre d'occurrence par défaut, sauf pour les collections pour lesquelles il se fait par déploiement. Un menu déroulant permet de changer ce classement.

Une recherche dans la liste principale (celle de gauche) est possible via le champ de saisie. On entre le motif (sensible à la casse), puis on passe de match en match avec la touche Entrée. Il est à noter que les regex fonctionnent.

Après avoir sélectionné un élément, vous pouvez avoir accès à la liste des textes et à son réseau, via le bouton action (petits engrenages à droite du cadran), le résultat est donné dans le cadran bas-gauche.

4.2 Search

Permet une recherche de motifs dans les textes.

4.3 Metadata

Donne accès au contenu des CTX du corpus, champ par champ.

5 le cadran bas-gauche

Il donne accès à différentes entrées par onglets :

- les textes : soit la liste intégrale des textes du corpus (conservée), soit une liste liée à un élément d’une liste (au fur et à mesure les onglets se rajoutent)
- les réseaux : générés au fur et à mesure des demandes de calcul

6 le cadran bas-droit

Affiche les éléments d’un texte sélectionné dans le cadran bas-gauche. Il comporte 3 onglets : Properties, Metadata, Text

6.1 Properties

Affiche les structures émergentes : Actants, Catégories et Collections. Pour l’instant, toutes les Collections d’un texte sont affichées (scorée par leur déploiement).

En double-cliquant sur un élément déployable, on en affiche les sous-éléments s’ils sont présents.

6.2 Metadata

Le contenu du .ctx du texte.

6.3 Text

Le contenu du .txt

7 le cadran haut-droit

7.1 Variant generation

Cet onglet est une aide à la génération de texte incluant des variantes issues des fichiers de Marlowe.

Dans la partie droite, le champ supérieur permet de tester une expression, le résultat est affiché dans la liste juste en-dessous :

- une expression textuelle renverra les /Var dans lesquels elle est contenue
- un /Var renverra la liste des items qui le composent

en double-cliquant sur un item de la liste générée, on le fait passer dans le champ de test

Dans la partie gauche, la phrase entrée dans le champ supérieur va être transformée. Suite au clic sur le bouton Identify, une proposition de remplacements des éléments de la phrase par des /Var (tirés aléatoirement si plusieurs possibilités existent) va être intégrée au champ moyen. Il est alors possible de

FIGURE 1 – Exemple avec [argument]

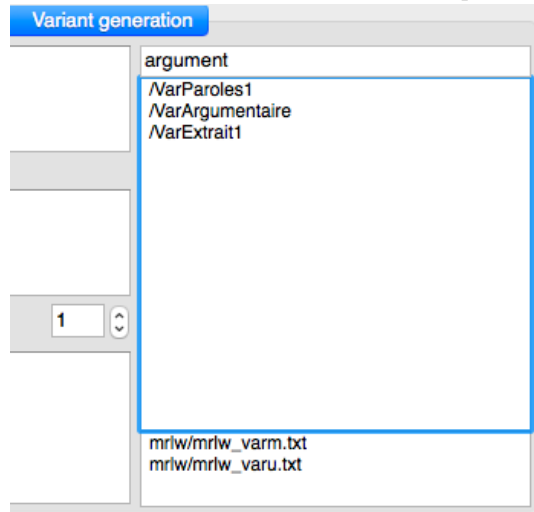
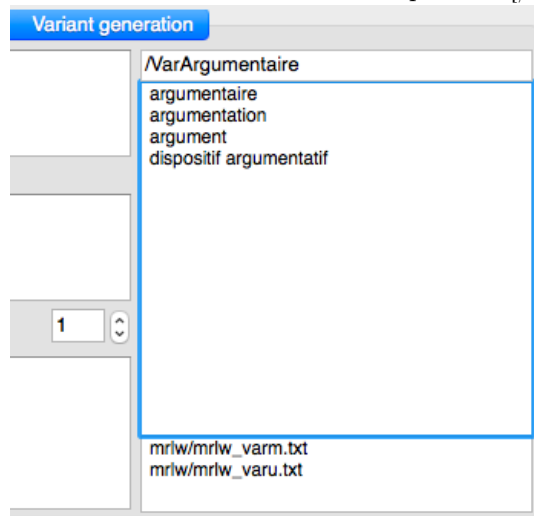


FIGURE 2 – Exemple avec [/VarArgumentaire]



générer (via le bouton Generate) des exemples de phrases utilisant des items de ces /Var tirés aléatoirement. On peut fixer le nombre de phrase générées avec le nombre indiqué par la spinbox (en français, zone de sélection numérique) de droite.

FIGURE 3 – Exemple avec [un argument important du corpus]

8 le codex

Le codex permet de générer automatiquement des champs du .ctx à partir du nom du fichier .txt.

On accède au codex via le menu Corpus. Une nouvelle fenêtre s'ouvre.

Elle comporte deux lignes, la première sert à générer les .ctx, la seconde à l'édition du codex.

Dans la première case, en haut à gauche, on peut glisser déposer les fichiers .txt à partir d'un gestionnaire de fichiers. La possibilité de générer un ctx correspondant est testée et le résultat apparaît dans la case de droite (la liste des champs et de leur contenu est visualisée en passant le pointeur sur la ligne du résultat). On peut vider la liste, retirer un fichier ou retester la liste via le menu contextuel (clic droit). En cochant [get titles], la première ligne du .txt sera utilisée pour informant le champ [title] du ctx.

En cliquant sur save CTX, on génère les .ctx dans le même répertoire que les .txt correspondant. Si un .ctx existe déjà, il n'est pas écrasé, sauf si [replace] est coché.

Les noms de fichiers .txt sont analysés suivant cette forme : ABREVVyymaaD.txt, avec ABREV un radical unique, yymaa la date codant l'année sur

deux chiffres, le mois sur un chiffre (septembre = 9, octobre = A, novembre = B, décembre = C), et le jour sur deux chiffres, suivant éventuellement par un détrompeur pour différencier d'éventuels doublons, consistant généralement en une lettre majuscule. Par exemple, JDLE11B04A.txt désigne un article du *Journal de l'Environnement* du 4 novembre 2011. Sous cette forme, les dates d'avant 2000 sont codées à partir de 50 et celles d'après 2000 jusqu'à 50. Si la date de votre texte est antérieure à 1950 ou postérieure à 2050, vous utiliserez la forme yyyymmaa pour la date.

La seconde ligne de la fenêtre du codex permet de consulter et d'éditer celui-ci. Les radicaux sont listés dans la seconde colonne. On peut effectuer une recherche, globalement ou pour un seul champ via la colonne de gauche.

En sélectionnant le radical voulu, on visualise les champs du codex dans la colonne de droite. L'ajout ou l'effacement d'une entrée se fait via le menu contextuel, la modification par un double-clic. En ajoutant un nouveau radical, plusieurs champs sont proposés par défaut, il n'est pas nécessaire de tous les remplir, il est toutefois conseillé de toujours renseigner le champ [author]. Le bouton [save codex] enregistre les modifications dans le fichier codex.xml du répertoire de l'interface python.

Si les radicaux sont uniques, un même auteur peut être présent plusieurs fois (avec des radicaux différents).

La dernière colonne sert à la fusion du codex avec les codex.cfg de P1 et les support.publi de Tirésias. Les nouveaux radicaux sont directement renseignés, les nouveaux champs de radicaux déjà présent également. Cependant, la fusion n'écraiera pas les champs existant, la liste des éléments non pris en compte du fichier sélectionné apparaît dans la case dédiée.