Pyxmaker

-

Notice

GALODÉ Alexandre

Pyxmaker

Cette notice a été rédigée sous Libre Office et écrite avec des polices d'écriture libres sous Linux.

" La connaissance appartient à tout le monde "

Film Antitrust

Conçu et rédigé par Alexandre GALODÉ.

SOMMAIRE

FRANCAIS

Introduction

Pyxmaker (Python eXecutable MAKER) a été conçu à la suite d'un besoin personnel: déployer facilement des logiciels PYTHON, conçus sous Linux, sur Windows.

A partir d'informations basiques, et d'une structure standardisée de projet, il s'occupe de créer les différents scripts et s'interface avec les outils adéquats afin de générer directement deux fichiers: un standalone (.zip) et un setup d'installation.

Ce logiciel, exclusif windows, nécessite que soit installés, aux chemin par défaut, pygtk (en cas d'exécution directe du code), cx_freeze, et Inno Setup.

Pour résumer donc, Pyxmaker est un outil simple d'utilisation permettant un déploiement rapide sur windows de code python.

Pyxmaker est placé sous licence GPL V3.

1 Fonctionnement

Wikimedia Commons,
Applications-system.svg

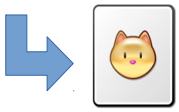
1.1 La structure standards à respecter

Pour que Pyxmaker fonctionne, cela impose de créer le code dans une structure bien définie, et d'utiliser un module prédéfini.

<NOM_PROJET>

***.py** Fichiers sources composants le logiciel

path_mkr.py Module fournissant le chemin courant adapté



icone.ico Icône du logiciel

Package(s) Package(s) potentiel(s) du logiciel

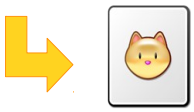
xx-<NOM> Dossier fichiers commencant par un nombre

_pxmk Dossier utilisé par Pyxmaker pour le setup

Licence.txt Texte de la licence

Before.txt Texte à afficher avant l'installation

After.txt Texte à afficher après l'installation



icone.ico Icone potentielle du setup

Chaque projet possède son dossier propre. Ce dossier doit être placé de préférence dans un dossier à la racine du C. Le chemin d'accès au projet ne doit en tout cas pas comporter d'espace (cela exclus d'office le bureau, mes documents, et program files entre autre). Personnellement, j'ai créé un dossier " C:\PJ_PYTHON ".

Ensuite à la racine, on disposera l'ensemble des fichiers python (.py) et des packages, ainsi que le fichier icône (ico) nommé " icone.ico " (vous pouvez créer ce fichier avec le logiciel GIMP à partir de n'importe quel autre format d'image).

Toujours au même niveau on trouvera le module ***path_mkr*** qui ne contient qu'une seul et unique fonction: ***cxf_get_path()***. Cette fonction ne prend aucun paramètre et renvoie un chemin. Ce chemin correspond au chemin d'exécution du code python (les *.py de la structure).

L'utilisation de ce module est rendu obligatoire par cx_freeze. La fonction détecte la plateforme (Linux, Windows, Mac), et dans le cas de windows s'il s'agit d'une exécution directe du code source ou de l'exécution du code converti en " .exe " .

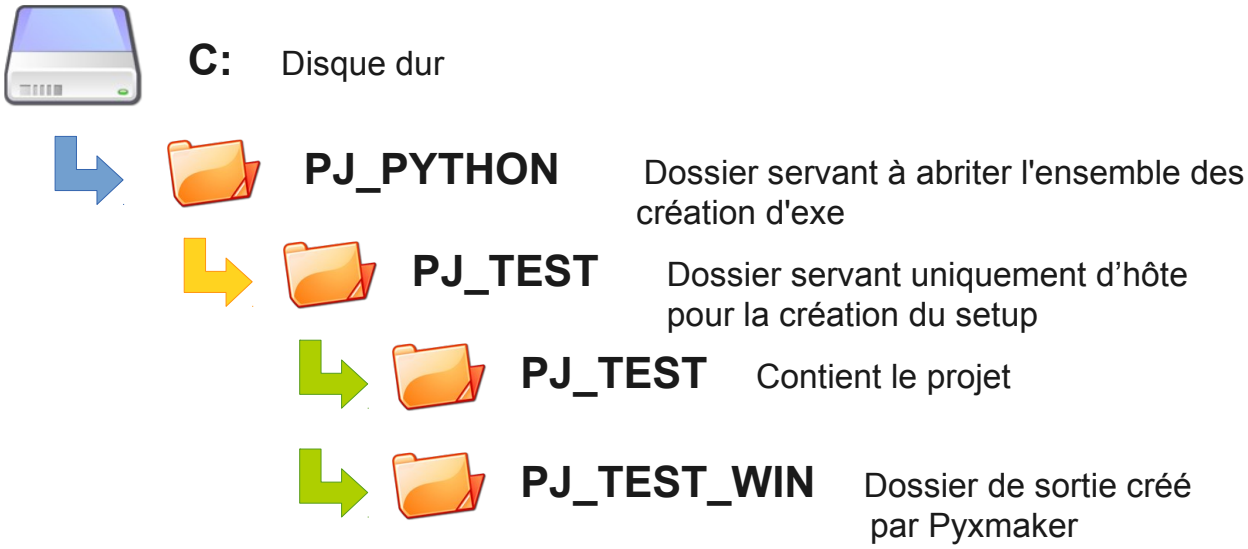
Toujours à la racine du dossier projet, nous devons trouver les potentiels " dossiers package ". Ils doivent absolument commencer par une lettre (tout comme le préconise les PEP).

L'ensemble des fichiers annexes nécessaires au bon fonctionnement du logiciel doivent être placés dans un/des dossier(s) dont le nom commencera par un nombre (ex: 00-DOC, 01-BDD, 02-IMG, …). Ce nombre doit comporté au moins un chiffre, sans limite maximum. Cela permet à Pyxmaker de différencier aisément les packages des autres dossiers, et également de structurer proprement les projets.

Le dernier élément que l'on trouvera à la racine est le dossier " _pxmk ". Ce dossier contiendra des éléments nécessaires à la création du setup d'install. Trois fichiers textes (tous obligatoires, même s'ils sont vides) et potentiellement un fichier icône. Les trois fichiers texte permettent de contenir le texte relatif à la licence à afficher dans le setup, et du texte à afficher avant et après l'installation. L'icone, au format ico, au nom libre est pour le setup. Elle n'est pas obligatoire et vous êtes libre d'en choisir une autre sur votre disque.

Pyxmaker créé en sortie un dossier " <NOM_PROJET>_WIN " au même niveau que le dossier du projet. Pour cela, il peut être préférable de créer un dossier pour la conversion.

Prenons un exemple: un projet s'appelant " PJ_TEST "

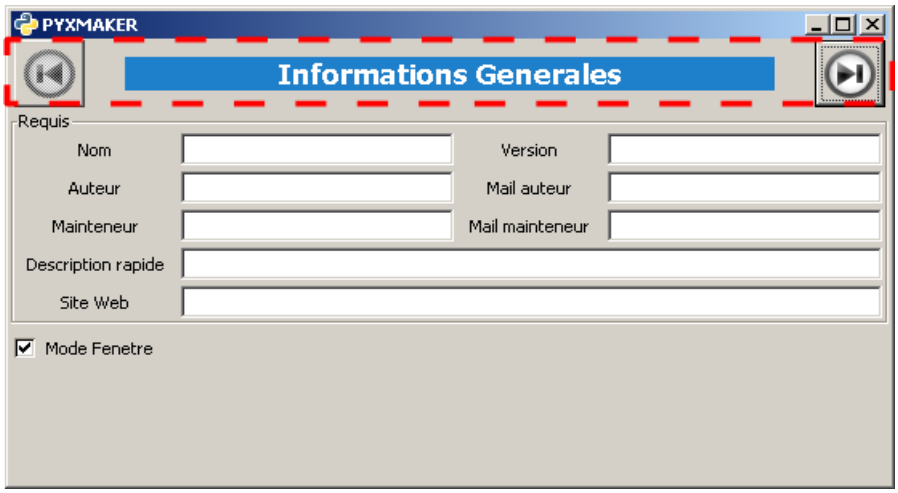


Dans cet exemple, nous avons pris pour exemple le disque C. Cela peut bien sur etre un disque D, E, …

J'utilise ici un dossier PJ_PYTHON car je créé régulièrement des petits softs Open Source. Le dossier PJ_PYTHON me sert à abriter l'ensemble de ces conversions. De même, afin d'éviter tout mélange de dossier, je créé un sous dossier par projet qui abrite le dossier des sources, ainsi que le dossier que génère Pyxmaker. Si vous ne faite cela qu'occasionnellement, vous pourrez alors supprimer le dossier " PJ_TEST " de plus haut niveau.

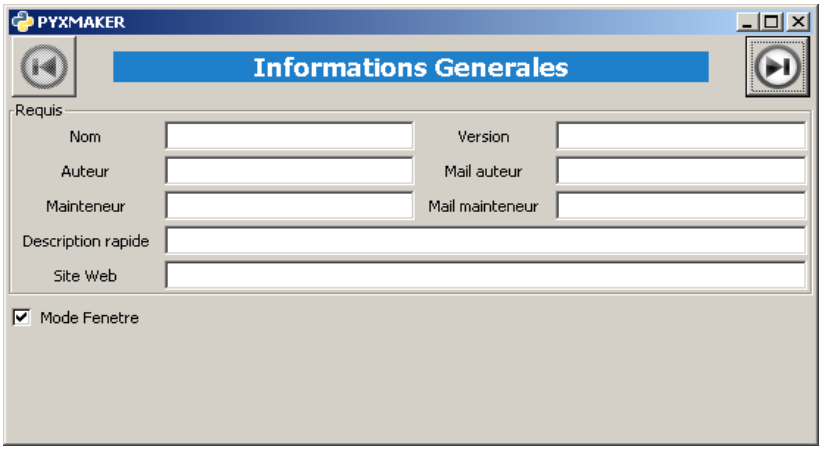
Ici, ***cxf_get_path()*** renverra " C:\PJ_PYTHON\PJ_TEST\PJ_TEST\ " comme chemin.

1.2 La toolbar



La toolbar est commune à tous les écrans. Elle vous permettra de commuter entre les différents écrans. Dans l'ordre: Informations Generales, Executable, Configuration.

1.3 Écran " Informations Generales "



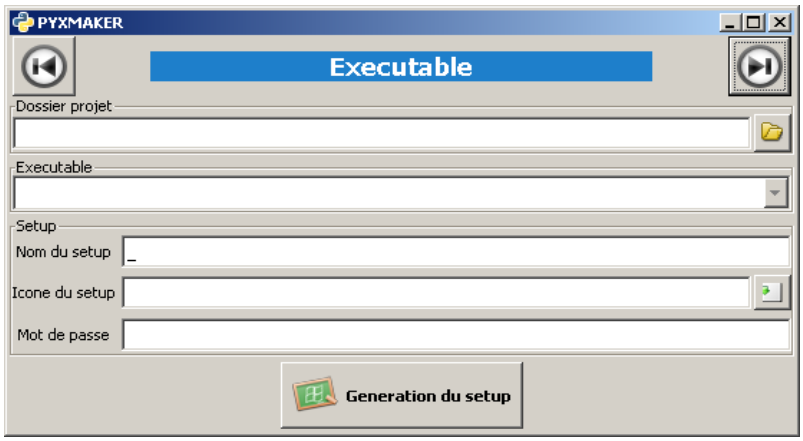
L'écran " Informations Generales " se compose de deux zones

Dans la première, on renseigne les informations textuelles du projet (elles sont toutes obligatoires):

=>Nom: zone texte
=>version: uniquement des nombres, séparés ou non par des " . " ou des " - "
=>Auteur: zone texte
=>Mail auteur: zone texte régie par REGEX, adresse mail valide requise
=>Mainteneur: zone texte
=>Mail mainteneur: zone texte régie par REGEX, adresse mail valide requise
=>Description rapide: zone texte
=>Site web: zone texte régie par REGEX, adresse web valide (http://...)

La seconde zone permet d'indiquer si notre code s'exécute en mode graphique/fenêtre ou en mode console (décoche). Cette option est cochée par défaut.

1.4 Écran " Executable "



L'écran de conversion est composée de plusieurs zones.

Tout d'abord, la partie dossier projet. Via le bouton à la droite, on selectionne le dossier qui contient le projet.

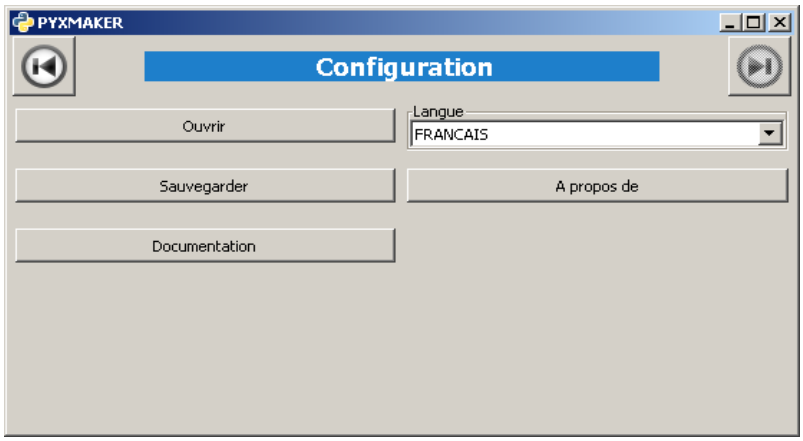
Une fois ce dossier renseigné, on sélectionne le fichier python (.py) qui correspond à l'exécutable.

Enfin, la zone setup permet de configurer un peu plus finement la création du setup d'install:
=>Le nom à donner au setup (par défaut: <nom_appli>_<version>).
=>L'icone désirée pour le setup (.ico uniquement), non obligatoire
=>Mot de passe pour le setup, non obligatoire

Pour ces deux derniers paramètres, si rien n'est saisie dans leurs champs, ils ne seront alors pas pris en compte.

Enfin, le bouton en bas de l'écran permet de lancer la chaine complete, qui aboutira à la création d'un fichier standalone (.zip) et d'un setup d'install.

1.5 Écran " Configuration "



L'écran " Configuration " permet de gérer un fichier de configuration pour le projet (boutons ouvrir et sauvegarder), de choisir la langue désirée pour le logiciel (effective au redémarrage du logiciel, francais par defaut), d'accéder à la fenêtre about, ou encode d'afficher cette notice d'utilisation.

1.6 Écran A propos de

Cet écran vous permet simplement d'accéder à des informations sur le logiciel même (licence, nom, version, ...)

2 Côté technique

Wikimedia Commons,
Crystal_Clear_app_Internet_Connection_Tools.png

2.1 Composition du logiciel

Le logiciel est constitué de plusieurs modules:
>ihm_pyxmaker: gère la couche graphique
>path_mkr: gère le chemin d'exécution
>bdd_pyxmaker: gère l'accès à la BDD
>cxf_pyxmaker: gère l'interfce avec cx_freeze
>setup_pyxmaker: gère l'interface avec Inno Setup
>save_load_pyxmaker: gère le fichier de configuration du projet
>pyxmaker: établi l'interfaçage entre les autres modules et pilote le logiciel même

2.2 Documentation technique du code source

Une documentation technique complète, type javadoc, est dispo dans:
PYXMAKER/00-DOC/00 - SRC

2.3 BDD

La BDD utilisée est une BDD SQLite.

Elle est constituée de 3 tables
>ABOUT: contient l'ensemble des données pour l'écran " A propos de "
>CONFIG: contient l'ensemble des éléments texte pour l'IHM et les messages
>MENU: contient l'ensemble du texte pour les menus

ENGLISH

Introduction

Pyxmaker (Python eXecutable MAKER) was written due to a personal needed: Easy windows deploy of my python software.

From basical information, and a standard project structure, software create alone the different scripts and call the good tools in order to generate a standalone (.zip) and a setup install.

This software, working exclusively under windows, need to work, pygtk (only if use from source code), cx_freeze, and Inno Setup

To summer, Pyxmaker is a simple tool that allows a fast windows deploy of python software

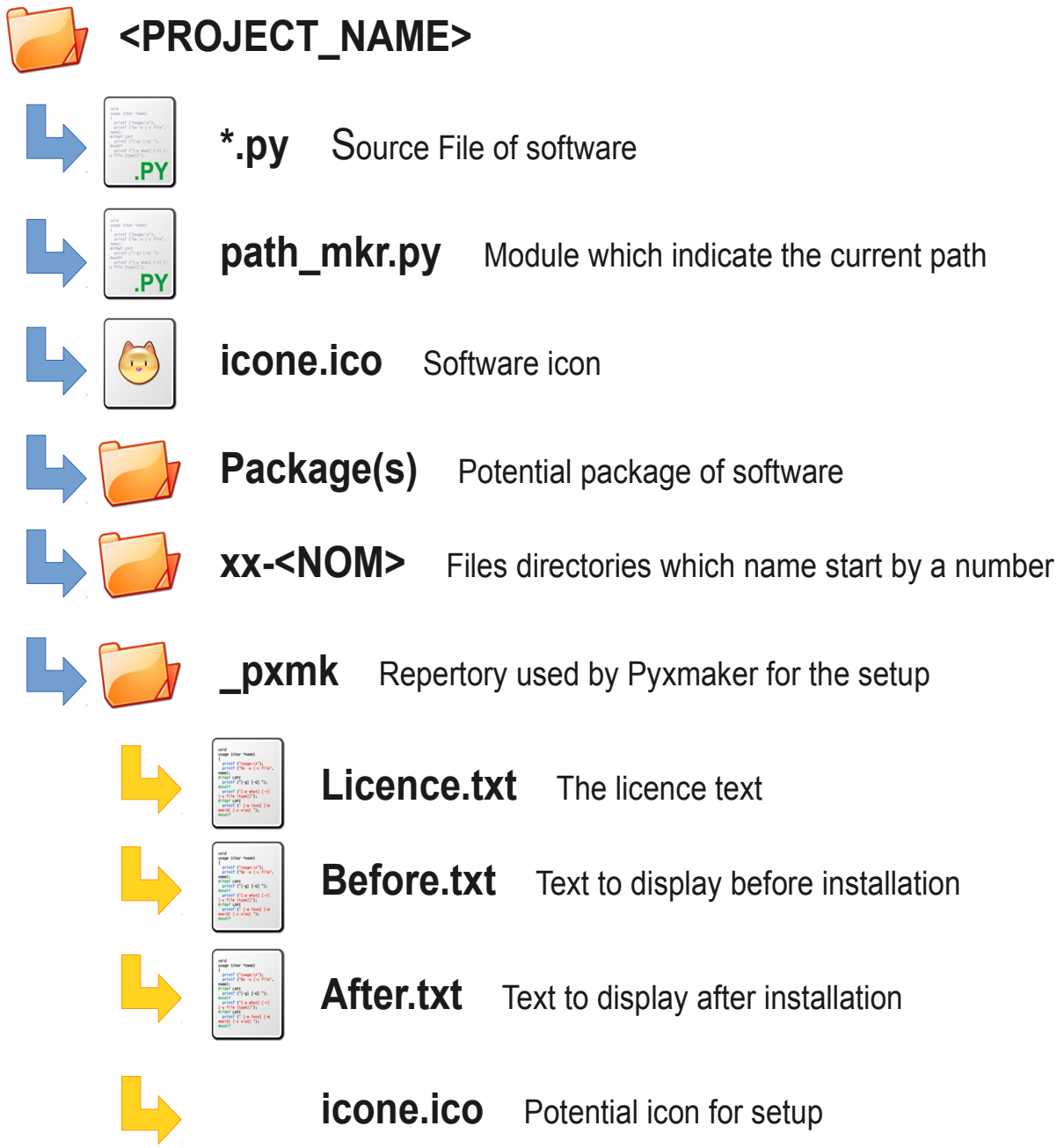
Pyxmaker is under GPL V3 licence.

3 Operation

Wikimedia Commons,
Applications-system.svg

3.1 Standard structure to use

In order Pyxmaker works correctly, you have to use a precised project structure.



Each project has it s own repertory. This repertory must be placed in another one at the root of C HDD. In all case, path don't have to use space (so not desktop, program files, ...). Personally, i use " C:\PJ_PYTHON ".

Then, at root, we'll have all .py files, packages and "icone.ico" 'use GIMP to create it from PBG file.

Always at the root, there is ***path_mkr*** module which has only one function: ***cxf_get_path()***. This function doesn't take any parameters and let know the execution path of .py files.

The use of this module is mandatory because of the use of cx_freeze. The function detects the OS type (Linux, Windows, Mac), and under Windows, if we execute directly the source code, or if we used an created executable(".exe").

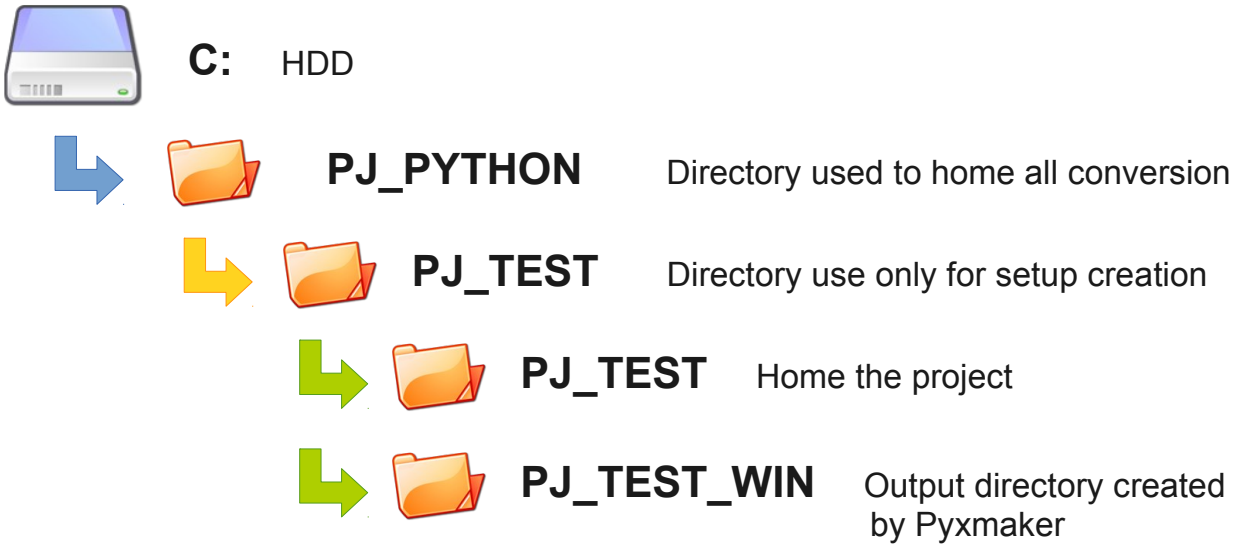
The potential package repertory must start by a letter as preconise in PEP.

All others files needed for software must be placed in directories which name start by a number (one or more digit) without maximum (ex: 00-DOC, 01-BDD, 02-IMG, …). This allow to Pyxmaker to easily make difference between package and data directories, and structure the project.

The last element is " _pxmk " directory. It will contain the mandatory files for setup creation. Licence.txt will contain the licence text which will be display at the installation. Before.txt and After.txt will contain the text to display before and after the installation. This three files must exist, but can be empty if there is nothing to display. In our example, we place in this folder the setup ico. But you can choose another one in the path you desire.

Pyxmaker create, in output, a " <NOM_PROJET>_WIN " folder, at the same level than the project directory.

An example to illustrate the explanations: a " PJ_TEST " project

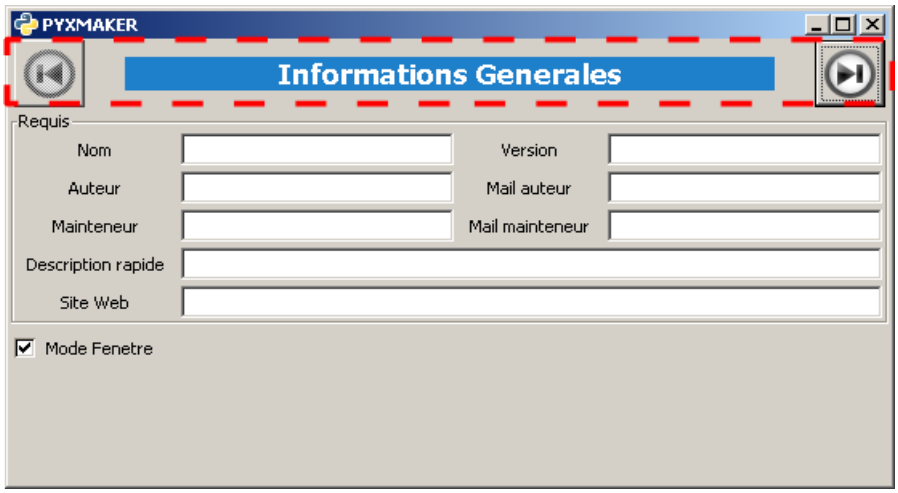


In this example, we have C HDD. Of course, you can use D, E, …, HDD

Here i used a " PJ_PYTHON " directory because i create regularly small Open Source software. This directory lets me home all my convert project. As well, i use a sub directory, with project name, to home every convert project. If you make this occasionally you can delete the high level " PJ_TEST " directory.

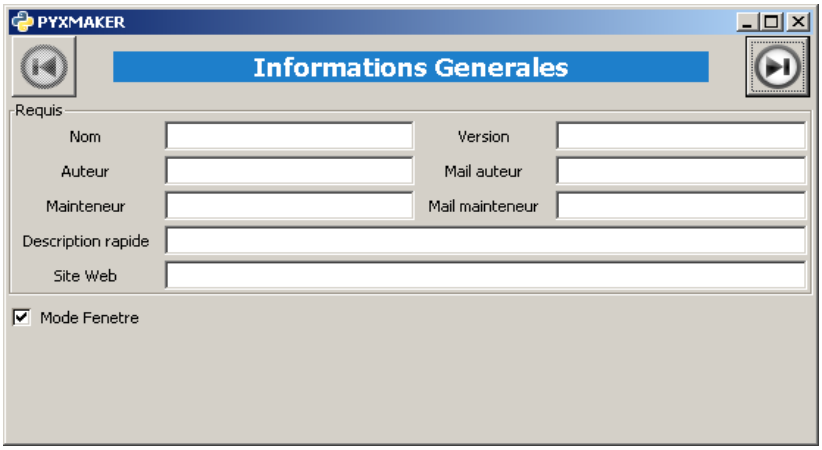
Here, ***cxf_get_path() will return*** " C:\PJ_PYTHON\PJ_TEST\PJ_TEST\ " as path.

3.2 Toolbar



Toolbar appears on all screens. It allows to change screens. In order: " General Informations ", " Executable ", " Configuration ".

3.3 " General Information " screen

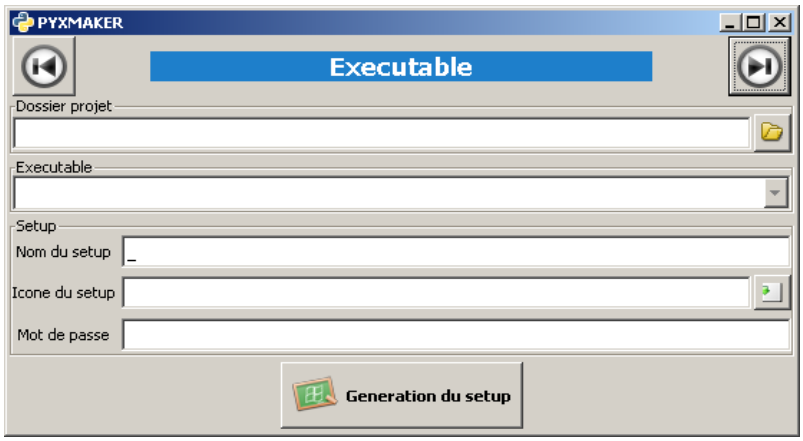


" General informations " screens is composed of two areas.

In the first one, the project textual data (all mandatories):
=>Name: text area
=>Version: only numbers, separated by " . " or " - "
=>Author: text area
=>Author mail: text area, with REGEX, valid e-mail required
=>Maintainer: text area
=>Maintainer mail: text area, with REGEX, valid e-mail required
=>Fast description: text area
=>Site web: text area, with REGEX, valid web adress required (http://...)

The second area allow to indicate if ours scripts are launched in console mode (not check) or windows mode (checked, by default).

3.4 " Executable " screen



The " Executable " screen is composed of several area.

First, the project directory part. Via the right button, you can choose the directory that contains the project to convert.

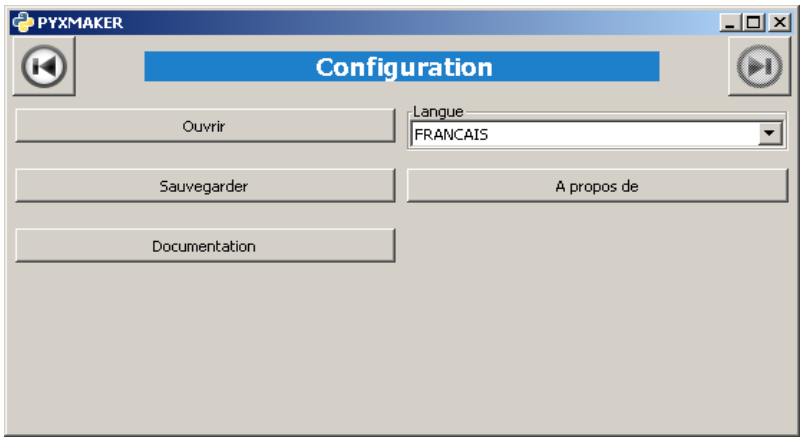
Then, you have to select the .py file which correspond to the executable

To finish, the setup area, which allows to config better the setup creation:
=>The name for the setup (default: <software_name>_<version>).
=>the desired icon for the setup(.ico only), no mandatory
=>Setup password, no mandatory

For this last two parameters, it won't be used if they are empty.

The back button allow to launch the entire process to obtain standalone and setup install files.

3.5 " Configuration " screen



" Configuration " screen allow to manage a project configuration file (Open & Close buttons), to choose the desired language (need to relaunch the software, french as default), to display the about windows, or to display this documentation.

3.6 About screen

This screen lets you access to software informations (licence, name, version, …).

4 Technical data

Wikimedia Commons,
Crystal_Clear_app_Internet_Connection_Tools.png

4.1 Software composition

The software is composed of several modules:
>ihm_pyxmaker: manage the HMI
>path_mkr: manage the execution path
>bdd_pyxmaker: manage the database
>cxf_pyxmaker: manage the communication with cx_freeze
>setup_pyxmaker: manage the communication with Inno Setup
>save_load_pyxmaker: manage the project configuration file
>pyxmaker: manage the communication between precedetn module and software

4.2 Technical documentation of source code

A complete technical documentation is readable in:
PYXMAKER/00-DOC/00 - SRC

4.3 BDD

The used database is an SQLite one.

She's composed of 3 tables:
>ABOUT: contains the data for about windows
>CONFIG: contains the configuration for software
>MENU: contains the text for the menu of software