

Tutorial: cx_Freeze

Mario Raul Freitas

27 de Agosto de 2016

Contents

1	O que é o cx_Freeze	1
2	Como instalar o cx_Freeze	2
3	Como utilizar o cx_Freeze	2
4	Arquivo setup.py	2
5	setup()	4
6	exe	5
7	opt	6
8	msi_data	7
9	Variáveis	8
10	Rodando o cx_Freeze	9
11	Modelos	9

1 O que é o cx_Freeze

O cx_Freeze é uma biblioteca de python capaz de transformar aplicações .py em executáveis .exe para Windows. Isso permite a distribuição dos programas sem a necessidade do usuário instalar o python previamente.

É possível criar uma pasta com todos os arquivos necessários para rodar o programa e distribuí-la como um arquivo compactado (.zip ou .rar, por

exemplo). Também é possível criar um instalador do windows .msi que irá instalar o programa no computador do usuário, de forma que ele se comporte tal qual um programa convencional.

2 Como instalar o cx_Freeze

Para instalar o cx_Freeze é necessário ter o python 3.4 (32-bits) instalado, assim como todas as bibliotecas utilizadas no programa que se deseja freeze. Com tudo isso instalado, basta utilizar o pip para instalar o cx_Freeze

```
pip install cx_Freeze
```

3 Como utilizar o cx_Freeze

Para rodar o cx_Freeze são necessárias duas coisas, o script que se deseja freeze e um arquivo setup.py. O arquivo setup.py contém as configurações de como será criado o executável ou o instalador.

4 Arquivo setup.py

O arquivo setup.py deve se parecer com o seguinte:

```
from cx_Freeze import setup, Executable

# executable options
script = 'T10_GUI.py'
base = 'Win32GUI'          # Win32GUI para gui's e None para console
icon = 'icon_64.ico'
targetName = 'CivCom Diagram Calculator.exe'

# build options
packages = ['matplotlib', 'atexit', 'PyQt4.QtCore', 'tkinter']
includes = []
include_files = ['icon_64.png', 'logo-90.png']

# shortcut options
shortcut_name = 'Diagram Calculator'

# bdist_msi options
```

```

company_name = 'CivCom UnB'
product_name = 'Diagram Calculator'
upgrade_code = '{66620F3A-DC3A-11E2-B341-002219E9B01E}'
add_to_path = False

# setup options
name = 'CivCom Diagram Calculator'
version = '0.1'
description = 'Bending moment diagram calculator'

"""
Edit the code above this comment.
Don't edit any of the code bellow.
"""

msi_data = {'Shortcut': [
    ("DesktopShortcut",          # Shortcut
     "DesktopFolder",          # Directory_
     shortcut_name,             # Name
     "TARGETDIR",               # Component_
     "[TARGETDIR]/{}".format(targetName), # Target
     None,                       # Arguments
     None,                       # Description
     None,                       # Hotkey
     None,                       # Icon
     None,                       # IconIndex
     None,                       # ShowCmd
     "TARGETDIR",               # WkDir
    ),

    ("ProgramMenuShortcut",      # Shortcut
     "ProgramMenuFolder",        # Directory_
     shortcut_name,              # Name
     "TARGETDIR",                # Component_
     "[TARGETDIR]/{}".format(targetName), # Target
     None,                       # Arguments
     None,                       # Description
     None,                       # Hotkey
     None,                       # Icon
     None,                       # IconIndex

```

```

        None,                                # ShowCmd
        "TARGETDIR",                          # WkDir
    )
]
}

opt = {
    'build_exe': {'packages': packages,
                  'includes': includes,
                  'include_files': include_files
                  },
    'bdist_msi': {'upgrade_code': upgrade_code,
                  'add_to_path': add_to_path,
                  'initial_target_dir': r'[ProgramFilesFolder]\%s\%s' % (company_name,
                                  product_name),
                  'data': msi_data
                  }
}

exe = Executable(
    script=script,
    base=base,
    icon=icon,
    targetName=targetName,
    # shortcutName=shortcut_name,
    # shortcutDir='DesktopFolder'
)

setup(name=name,
      version=version,
      description=description,
      options=opt,
      executables=[exe]
      )

```

A seguir, iremos descrever parte a parte o que cada trecho do código faz

5 setup()

O mínimo que o arquivo setup.py deve conter são os imports e a função setup(). A função setup é a única parte do código que efetivamente roda o

programa para converter a sua aplicação em um executável. São passados para ela os seguintes parâmetros:

- name: nome do programa (na descrição) [string]
- version: versão do programa [string]
- description: descrição do programa [string]
- options: opções de configurações [dictionary]
- executables: executáveis .exe que serão criados [list]

```
from cx_Freeze import setup, Executable
```

```
setup(name=name,  
      version=version,  
      description=description,  
      options=opt,  
      executables=[exe]  
    )
```

6 exe

No argumento 'executables' de setup passa-se um executável exe. Esse é definido como mostrado abaixo.

```
exe = Executable(  
    script=script,  
    base=base,  
    icon=icon,  
    targetName=targetName,  
    # shortcutName=shortcut_name,  
    # shortcutDir='DesktopFolder'  
)
```

- script: nome do script principal .py que será freezado [string]
- base: forma de execução do programa. deve-se usar 'Win32GUI' [string] se o programa for rodar em uma gui ou None [palavra reservada] se o programa for rodar no prompt de comando

- icon: nome ou diretório do ícone que será usado no .exe da aplicação [string]
- targetName: nome do .exe que será criado
- shortcutName: nome do atalho criado na instalação [string] (está comentado porque existe um método melhor)
- shortcutDir: nome da pasta em que o atalho será colocado [string] (está comentado porque existe um método melhor)

7 opt

Outro argumento de setup é o options, para o qual é passado o dicionário opt. Esse foi definido como a seguir:

```
opt = {
    'build_exe': {'packages': packages,
                  'includes': includes,
                  'include_files': include_files
                 },
    'bdist_msi': {'upgrade_code': upgrade_code,
                  'add_to_path': add_to_path,
                  'initial_target_dir': r'[ProgramFilesFolder]\%s\%s' % (company_name,
                                product_name),
                  'data': msi_data
                 }
}
```

- build_exe: opções relativas ao processo de freezagem da aplicação
- bdist_msi: opções relativas ao processo de criação do instalador
- packages: bibliotecas de python incluídas na distribuição [list de strings]
- includes: módulos de python incluídos na distribuição [list de strings]
- include_files: outros arquivos incluídos na distribuição, como imagens, dados e outros [list de strings]
- upgrade_code: código de upgrade para atualizações do software; é o que garante que o instalador irá reconhecer se o programa já foi instalado no computador do usuário e qual a versão ele está utilizando [string]

- `add_to_path`: adicionar ou não o programa ao path do usuário [True ou False]
- `initial_target_dir`: diretório onde será instalado o programa (C:/Arquivos de programas/Nome da Empresa/Nome do Programa)[string]
- `data`: conjunto de dados para o instalador (usado para criar atalhos para o programa no Desktop e no Menu Iniciar) [dictionary]

8 msi_data

Essa variável contém os dados para que o instalador crie os atalhos no computador do usuário. Utilizando esse método é possível adicionar diversas opções que não estão disponíveis pelo método mostrado na variável `exe`. Por exemplo, por esse método é possível fazer com que ao abrir a aplicação pelo atalho na Área de Trabalho, que o programa entenda que ele está sendo aberto no diretório em que o programa foi instalado. Isso é necessário para que as imagens e dados possam ser carregados ao abrir o programa.

```
msi_data = {'Shortcut': [
    ("DesktopShortcut",          # Shortcut
     "DesktopFolder",           # Directory_
     shortcut_name,             # Name
     "TARGETDIR",               # Component_
     "[TARGETDIR]/{}".format(targetName), # Target
     None,                      # Arguments
     None,                      # Description
     None,                      # Hotkey
     None,                      # Icon
     None,                      # IconIndex
     None,                      # ShowCmd
     "TARGETDIR",               # WkDir
    ),

    ("ProgramMenuShortcut",      # Shortcut
     "ProgramMenuFolder",        # Directory_
     shortcut_name,             # Name
     "TARGETDIR",               # Component_
     "[TARGETDIR]/{}".format(targetName), # Target
     None,                      # Arguments
```

```

        None,                # Description
        None,                # Hotkey
        None,                # Icon
        None,                # IconIndex
        None,                # ShowCmd
        "TARGETDIR",         # WkDir
    )
]
}

```

9 Variáveis

Todos os trechos de código acima são padrão para freezar qualquer aplicação. Para facilitar a utilização, todos eles foram escritos em função de variáveis que são definidas no começo do código. Portanto, para editar o arquivo `setup.py` e configurá-lo para uma outra aplicação basta mudar o valor dessas variáveis:

```

# executable options
script = 'T10_GUI.py'
base = 'Win32GUI'          # Win32GUI para gui's e None para console
icon = 'icon_64.ico'
targetName = 'CivCom Diagram Calculator.exe'

# build options
packages = ['matplotlib', 'atexit', 'PyQt4.QtCore', 'tkinter']
includes = []
include_files = ['icon_64.png', 'logo-90.png']

# shortcut options
shortcut_name = 'Diagram Calculator'

# bdist_msi options
company_name = 'CivCom UnB'
product_name = 'Diagram Calculator'
upgrade_code = '{66620F3A-DC3A-11E2-B341-002219E9B01E}'
add_to_path = False

# setup options
name = 'CivCom Diagram Calculator'

```



```
version = '0.1'
description = 'Bending moment diagram calculator'
```

10 Rodando o cx_Freeze

Para rodar o cx_Freeze abra uma janela do prompt de comando no diretório que contém o script a ser freezado e o arquivo setup.py e digite o seguinte comando:

```
python setup.py build
```

Isso irá criar uma pasta build, que contém o executável e todos os arquivos necessários para que ele rode. Essa pasta pode ser compactada e distribuída para os usuários.

Para criar um instalador utilizando o cx_Freeze abra uma janela do prompt de comando no diretório que contém o script a ser freezado e o arquivo setup.py e digite o seguinte comando:

```
python setup.py bdist_msi
```

11 Modelos

Nesse tutorial estão inclusos os arquivos setup.py já preenchido conforme o tutorial, uma versão comentada do mesmo arquivo e dois arquivos .bat, build.bat e bdist_msi.bat. Os arquivos .bat simplesmente rodam o comando para criar uma build e criar um instalador, respectivamente.