**Rcspy开发文档  
Version:0.1**

**The Rcspy Developer**

**(wangwsong@fomxial.com)**

**2018.5**

# 一、开发环境

|  |  |
| --- | --- |
| 系统 | Windows/linux/mac均可，建议Ubuntu14.04/16.04LTS |
| Anaconda版本 | 5.0.1x86\_64 |
| Python版本 | 2.7.x |
| IDE | JetBrains Pycharm 2018.03 |
| PyQt版本 | PyQt5 |
| Matplotlib版本 |  |

简要配置教程：

1. 登录清华镜像站<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/> 针对机器平台选择合适的镜像下载并安装。
2. 配置完成Anaconda环境后，执行pip install obspy安装obspy框架。
3. 安装pycharm ：登录<http://www.jetbrains.com/pycharm/download/> 选择合适的版本下载安装
4. 拉取源代码，登录<https://github.com/Veasonwang/Rcspy>下载源代码或者输入命令拉取源代码（需要[安装git](https://www.cnblogs.com/shelly01-zhou/p/7736610.html)）

git clone https://github.com/Veasonwang/Rcspy.git

1. 打开pycharm，打开源代码文件夹。
2. 运行Rcspy.py 即可。

# 二、QtDesigner的使用

添加界面时，需要使用QtDesigner绘制界面，将生成的.ui文件转化成.py文件。QtDesigner安装及使用步骤如下：

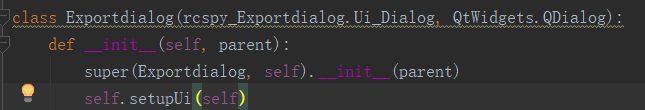
1. 安装Qt，Windows平台登录 <https://www.qt.io/> 下载Qt。Linux下运行Sudo apt-get install Qtcreator命令完成安装。
2. 打开QtDesigner，创建一个新的窗口，另存为.ui文件。执行命令：Pyuic5 –o xxx.ui xxx.py可生成py源码文件
3. 将.py文件移动到main/ui\_package中，在Sub\_windows\_support.py中添加from ui\_package import xxxx即可完成该窗口类的导入
4. 创建该窗口对应的class，完成构造函数，主要需要继承创建的窗口类。示例如图1：

图1

1. 调用。需要创建该窗口时，直接创建该类的实例即可。

# 三、源代码文件结构

1. 根目录。根目录下是GPL2.0开源协议、readme、以及版本信息。
2. ./main目录。该目录是程序主目录，源码位置在此，该目录下有四个源码文件。作用如下：

* Rcspy.py：程序的入口，里面包含程序主窗口类——Rcspy类，负责程序启动、主窗口逻辑调用、变量初始化，是功能的入口。
* datamanager.py：数据管理框架。包含Files、File、Station、Channel类。以及Source类、ChennelVisible类、Phasetime类等其他类。
* util.py：主要是绘图模块（MplCanvas类），以及其他继承自Qt的控件类。
* Sub\_windows\_support.py：所有的子窗口类在此文件中。

# 四、主要类的成员变量与成员函数

1. Rcspy类成员变量：

* self.qml：程序唯一MplCanvas类的变量，与QWidget控件绑定
* self.qmlcanvas：Rcspy的QWidget控件
* self.fig：qml变量唯一的figure，唯一画布。
* self.zoomswi：缩放开关
* self.Files：一般Files类实例
* self.ondrawstations：待绘制的台站
* self.sac\_files：为了兼容绘图类的以三通道为单位的绘图接口，转为sac文件设置的files类，要求sac文件的输入数量必须是3的倍数，否则出错。
* self.qmldrawswi：左右拖动开关。

1. Rcspy类的方法：

* Eventconnect()：连接在QtDesigner中未完成连接的事件
* initdrawstation()：当打开文件时，确保一开始有一个台站是可见的
* draw()：完成画布的更新。
* update\_ondraw\_stations()：同步待绘制台站。当更改台站是否可见属性后，需要执行此函数同步待绘制台站。
* onqwidghtsizechangeed(QRectevent)：保证qml的大小与qmlcanvas的大小一致
* pick(event,phase)：event是MplCanvas变量的鼠标事件，phase是一个字符。用于拾取震相。调用对应通道的getpick()方法完成震相的拾取，记录鼠标所在位置的时间和选择的震相完成。
* \_initStationTree()：初始化列表树控件
* poptreemenu()：列表树菜单的弹出实现
* Input\_Source\_info(items)：为制定文件输入震源信息
* Export\_calculation\_phase\_file(items)：输出指定文件的理论到时信息
* Export\_Pick\_phase\_file(items)：输出指定文件拾取的震相
* attach\_event\_file(items)：为指定文件关联事件文件
* export\_xml(items)：导出指定文件所包含的事件
* Onstationtreeselectedchange()：当列表树控件选定的项目改变时，改变状态栏的状态。
* Set\_selected\_Invisible(selectedList)：使选中的台站不可见
* Set\_selected\_Visible(selectedList)：使选中的台站不可见
* \_initVistblebtn()：初始化可见的通道。
* \_changeStationVisibility()：对选中的台站可见状态取反