|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Table of contents  Launching the Application from Command Line 3  SFTPPipeline Dashboard 4  Active Upload Widget 5  Admin Login Widget 6  Object Management: 7  QtDesigner: Installation & Main Window 10  Adding UI Files to the Python Application 12  Packaging Into An Executable 13   Database Schema 15 | | | | | | | |  | |  |  | SFTP Pipeline User Guide |  |
|  |  |  | |
| Urban Science | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Extra Functionality | | | QtDesigner | Pyuic5 | | |  | QtDesigner .ui can be loaded directly by the constructor or converted to .py files using the following command, as discussed in the **pyuic5** section: *pyuic5 -x example.ui -o example.py* | | PyQt5-tools | | |  | Contains QtDesigner. Install with: *pip install pyqt5-tools* Located in your python folder: *[python folder] / Lib / site-packages / qt5\_applications / Qt / bin/ designer.exe* | | PyInstaller | | |  | Allows packaging the SFTP tool into a single executable. Install with: *pip install pyinstaller* | | 5 | | |  |  | Module Dependencies | |
|  |  | PyQt5 | |
|  | GUI framework in which the User Interface is developed. Can be used to implement new UI elements in future development. |
| MySQL Connector | |
|  | Self-contained python driver for communicating with MySQL servers. |
| Hash Lib | |
|  | Contains cryptographic and hashing functions that are used in the application. |

|  |
| --- |
| **Launching the application From Commandline**  **Open your File Explorer, and navigate to the ‘application/src’ directory…  Inside the ‘/src’ directory, you see the SFTPPipeline.py file.  Open a Terminal window in this location, and then execute the following command: ‘ *python SFTPPipeline.py ’* The application window should appear.**  **Note: Errors may occur in the absence of modules the application depends upon, and may be reconciled with the execution of the following command:  *‘ pip install [API/Module Name] ‘*** |

|  |
| --- |
| **SFTPPipeline Dashboard** |
| **The SFTPPipeline Dashboard initially consists of 5 main widgets:**   1. Displays user-created **Jobs** consisting of one or more **Presets**   **D**   1. Displays user-created **Presets** consisting of one or more **Files** 2. Displays connections to any number of SFTP servers   **A**  **B**  **C**   1. Allows authorized users to navigate to the **Administrative Login** portal. 2. **Active Uploads**:  Each **Job**/**Preset** will display upload progress with respect to individual files.   **E** |
| **Active Uploads Widget** |
| Consider the **Active Uploads** widget below.  Deliverables can be selected in the form of **Jobs** and **Presets**.  Multiple **Jobs**/**Presets** can be selected simultaneously, and their individual files will be added to the **Active Uploads** pane.  Upon clicking **Upload**, their progress is displayed incrementally, with respect to individual files.  **Important Note:** New **Users**, **Presets**, **Jobs**, and SFTP **Connections** can only be created by authorized users who have been authenticated through the **Admin Login** widget, as depicted in the following section. |
| **Admin Login Widget** |
| Authorizedusers can click the **Admin** button in the top right corner of their dashboard.  Users can return to the previous window by clicking the **Home** button in the upper right region.  The following window possesses multiple Tabs, allowing **Admin** users greater control.  Each of these tabs allow users to create, manage, and delete **Users**, **Presets**, **Jobs**, and **Connection** objects from the online database; affecting what other users see and have access to.  The window will change to display an administrative login form as depicted above.  The environment depicted in this documentation uses example credentials, in which the passwords have been hashed and salted. |
| **Object Management: Users List** |
| In this example, this affects Users’ ability to use the Administrative widgets  The **Users List** allows Admins to add and delete Users from the database. |
| **Object Management: Presets List**  **C** |
| **C**  **A**  **Presets** can be effortlessly deleted by selecting one or more **Presets** in the **Presets** pane followed by clicking the button **Delete Selected Preset.**  **F**  Lastly, a **Preset Name** must be provided in the insertion field, and the **Add Preset** button clicked in order for the new **Preset** to be created.  **E**  After doing so, one or more Servers can be selected, designating them to the **Preset**. Click a server a 2nd time if you wish to deselect it and disassociate it from the **Preset** you are creating.  **E**  New Presets can be created by clicking the **File Selector** window opens a new window. users can then select multiple files to add to the **Files List**.  Existing **Presets** can be selected in this window, causing the associating files to populate the **Files List** .  **D**  **A**  **B**  **F**  **E**  **D**  **C**  **B**  **A**  The Presets List has a considerably more intricate interface.  [Cite your source here.] |
| **Object Management: Connections List** |
| **Connections** can be removed by selecting them in the list and then clicking **Delete Selected Connection(s)**.  Users can create new **Connections** from this window, by providing the following, and then clicking **Add Connection**  • A **Server Name** • A **Username** • A **Password** • A valid URL for the **Remote Directory**. |
| **QtDesigner: Installation** |
| The first step is installing QtDesigner. Open a Command Prompt and execute the command  *pip install pyqt5-tools*    You can find your python folder by executing the command: *python -c “import sys; print(sys.path)”*  Then navigating to the folder:  *… / [Pythonxxx] / Lib / site-packages / qt5\_applications / Qt / bin/ designer.exe*    Right-click your executable and **Pin To Start**, **Pin to Taskbar**, or **Create [a] Shortcut** to save yourself time in the future.  Launch the application.  [ This does not require a visual demonstration. ] |
| **QtDesigner: Main Window** |
| **5wi6**  From here, you can insert **Buttons**, **Labels**, **Widgets**, and a myriad of other objects to your Main Window from your **Widget Box** (left). Any object created can be selected via the **Object Inspector** (top right), and its properties viewed and edited in the **Property Editor** further down.   For a more in-depth tutorial on QtDesigner creating complex graphical user interfaces, consider watching the PyQt5 QtDesigner tutorial linked below: *https://www.youtube.com/playlist?list=PLzMcBGfZo4-lB8MZfHPLTEHO9zJDDLpYj*  QtDesigner’s visibility may depend on your device’s resolution. The new **MainWindow** will appear in the middle pane.  The main window should open with a smaller, **New Form** window displayed on top of it. The primary two elements you will likely be concerned with are of course **Main Window** and **Widget**.You will first want to **create** a **Main Window**.Note: Keep the **Device** and **Screen Size** properties in mind when designing UI windows and widgets, as they may impact performance on a diverse variety of machines. |
| **QtDesigner: PYUIC5** |
| *~/SFTPPipeline/src/views/MainWindow.py*  As mentioned on pg. 2, line 33 of our code demonstrates the python constructor of our code incorporates .ui files into the application directly.  However, the API **pyuic5** can optionally convert .ui files to .py files which can then be used as **Views** in your python application.  [Cite your source here.]  Optional Features |
| **Packaging Into An Executable** |
| Text  Description automatically generated  **You can install PyInstaller with: *pip install pyinstaller”***  **Then, navigate to your ‘~/SFTPPipeline/src ‘ folder:** |
| Note: ***Pyinstaller will produce the folder and executable depicted below. Ignore the external executable. The SFTPPipeline folder contains all the assets, as well as the EXE it needs.***  **Execute the following command:** *pyinstaller --add-data "views;views" --add-data "assets;assets" --onefile SFTPPipeline.py*  **A very intimidating wall of output should flood your command line.** |
| **Database Schema** |
| **Note that this executable successfully launches the application’s main page so long as the user has a working connection to their company’s database. For the sake of this example, the tool is connecting to a MySQL database for testing purposes.**  **You can see several folders and files in this directory, including the SFTPPipeline.exe file that will launch this program.  This ‘SFTPPipeline’ folder is self-containing. It can be moved out of the ‘*/src/*dist’ folder, and into another location, such as moving it to the C drive, such that its location would become ‘C:\Program Files\SFTPPipeline’.  Note that launching this application from the SFTPPipeline directory depicted will successfully launch and display the tool’s main page so long as the user has a working connection to their database. For the sake of this example, the tool is connecting to a MySQL database.** |
| **Database Schema** |
|  |