

QR Toolbox Installation Guide

Step 1. Download the QR Tool files

1. Download QR Tool files from https://github.com/USEPA/QR_Tool/tree/master by selecting the green “Code” button on the right side, then clicking “Download ZIP”.
2. Then, extract the ZIP files contents to the desired location. You now have all of the files necessary to run the software.

Step 2. Install Python and pip packages

The QR Toolbox requires Python 3+ (and Python 3.7.4 is preferred) and some Python packages to run, along with other setup commands. In order to install this software:

1. First locate the python-3.7.4-amd64.exe file in the QR_Tool/Setup folder, then double-click it to run it. The Python installer window will appear on your screen similar to **Figure 1**
2. Before starting the installation of Python, **select the option to “Add Python 3.7 to PATH”**:
 - a. **If you already have Python 3.7.4 installed, you can ignore this step.**



Figure 1 Python Installation Window – If Python not already installed

Note: EPA users may not have permission to install Python and need to ask IT support to give permission. The installation steps after Python is installed do not require IT support permission.

3. Once the Python installation is done, run the QR-Toolbox.py file in the QR_Tool folder, and it will automatically install the following required Python packages: pyzbar, imutils, qrcode, Pillow, opencv-python, and office365. This should look similar to the image below (**Figure 2**). **If this does not happen, or the command prompt window closes quickly/abruptly, or there is any other issue, you will need to restart your computer, and then perform the following actions once it has rebooted completely:**
 - a. Locate and open the QR_Tool folder and run the QR-Toolbox.py file once again. The file will then run the other commands to install the Python packages, and your screen should look similar to **Figure 2** below.

App Information

The app's information, including app id, secret, title, hosting url and redirect url.

Client Id:

Client Secret:

Title:

App Domain:
Example: "www.contoso.com"

Redirect URI:
Example: "https://www.contoso.com/default.aspx"

Click "Generate" button to generate Client Id and Client Secret. Write down Client Id and Client Secret to use in the next step. Enter a Title that is meaningful for you.

Title: your title

App Domain: www.localhost.com

Redirect URI: <https://www.localhost.com>

Click "Create" button to create.

Then run the following URL (example):

https://usepa.sharepoint.com/sites/Emergency%20Response/EOIncident/_layouts/15/AppInv.aspx

App Id and Title

The app's identity and its title.

App Id:

Title:

App Domain:
Example: "www.contoso.com"

Redirect URL:
Example: "https://www.contoso.com/default.aspx"

App's Permission Request XML

The permission required by the app.

Permission Request XML:

Enter the Client id and click 'Lookup' button.

In the 'Permission Request XML:' box, enter the following:

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
<AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web"
Right="FullControl"/>
</AppPermissionRequests>
```

Click the 'Create' button to finish.

Step 4. Modify settings.py

Locate and modify the settings.py file in QR_Tool/Setup using the Client Id and Client Secret generated from the last step and enter the SharePoint site URL for 'url'.

```
settings = {
    'url': '<Your URL>',
    'client_id': '<Your Client Id>',
    'client_secret': '<Your Client Secret>'
}
```

The following lines in QR-Toolbox.py may need to be modified:

- *listTitle = "QR Timestamps" (this is the SharePoint list name for writing scanned entry to)*
- *qrfolder = "QR Codes" (this is the folder name to store QR Code image files)*
- *bkcsvfolder = "HXWTEST" (this is the folder name to store backup csv files)*
- *qrbatchfile = "names.csv" (this is the file name for batch generate QR Code images)*
- *relative_url =
"/sites/Emergency%20Response/EOCIncident/EOC%20Documents/QR Codes/names.csv" (this is
the location that names.csv resides)*

QR Toolbox User Guide

Run the QR-Toolbox

To run the QR-Toolbox, simply double-click the QR-Toolbox.py to run.

** If the user has two versions of Python installed (Python 2 and Python 3), they may need to specify running QR-Toolbox.py in Python 3.

The following is one way to do that:

1. Create a .bat file to run the QR-Toolbox by opening a new notepad file and saving it as a .bat. Add the following line from bullet 'a' below (put it all on one line) to the .bat file (you will need to fill in the locations of each executable file, as exemplified in bullet 'b'):
 - a. `<python 3 exe location>/python.exe <QR_Tool location>/QR-Toolbox.py`
 - b. Example: `C:\Users\<user>\AppData\Local\Programs\Python\Python37\python.exe C:\Users\<user>\Desktop\QR_Tool\QR-Toolbox.py`
2. (Optional) Create a shortcut of the .bat file on the desktop by right-clicking on it and selecting "Create Shortcut", and then dragging that shortcut to the desktop or anywhere else.
3. Double-click the .bat file or the shortcut to run the QR-Toolbox.

Startup Questions

There is one question that is asked when the QR-Toolbox is started:

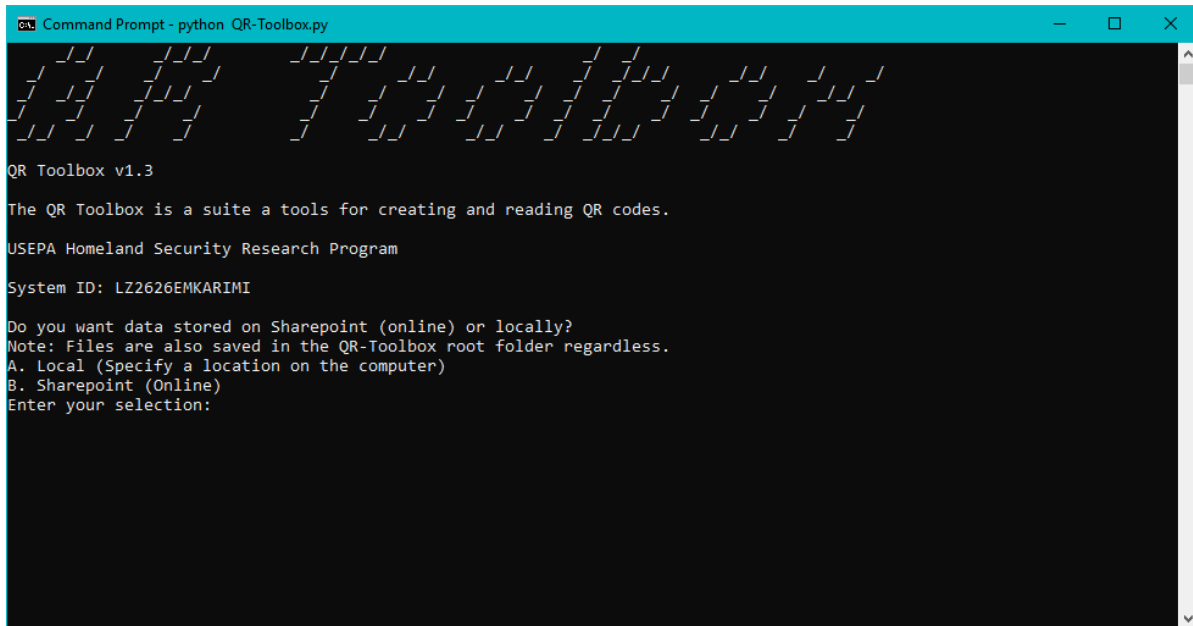
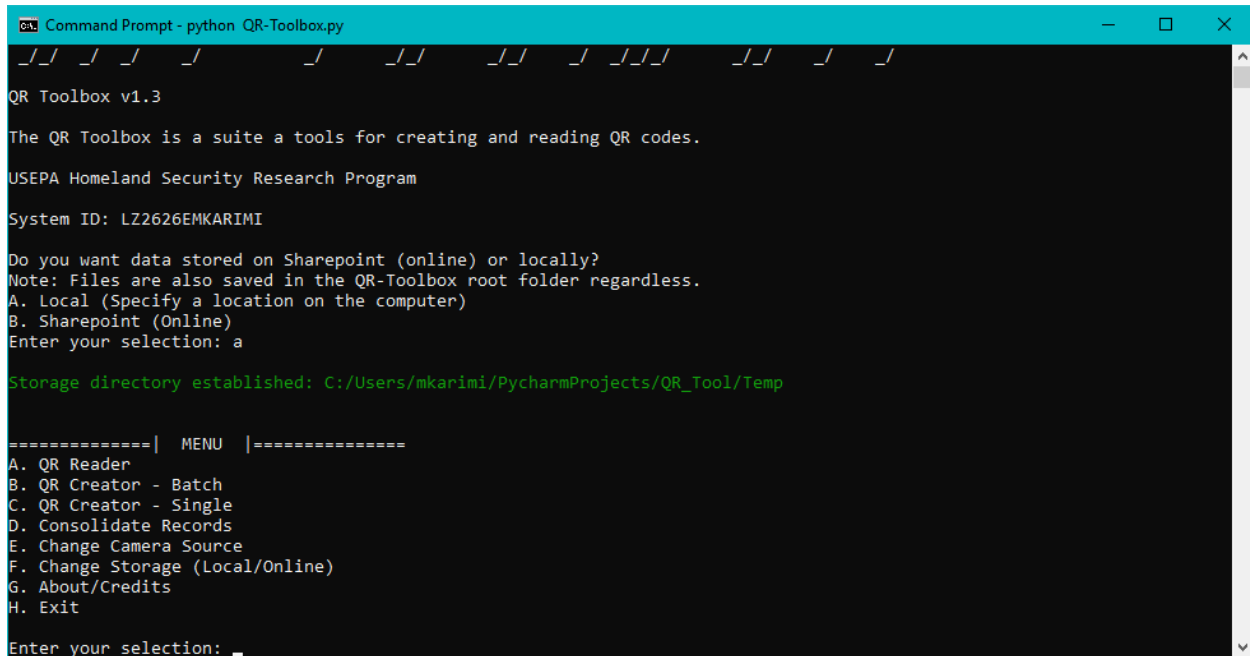


Figure 3 Startup Question

1. Do you want data stored on SharePoint (online) or locally?
 - a. When any data is created by the QR-Toolbox, it is stored in the same folder as the QR-Toolbox.py by default. However, the user has the option of also telling the software to store that data either online on SharePoint, or somewhere else on their computer (locally).
 - b. You can also change this from the main menu, as mentioned below.

Functions in the QR-Toolbox

There are 8 functions in each version of the QR-Toolbox (the version shown to the user depends on the answer given for the startup question), with only a slight difference between the two versions:



```
Command Prompt - python QR-Toolbox.py

QR Toolbox v1.3

The QR Toolbox is a suite a tools for creating and reading QR codes.

USEPA Homeland Security Research Program

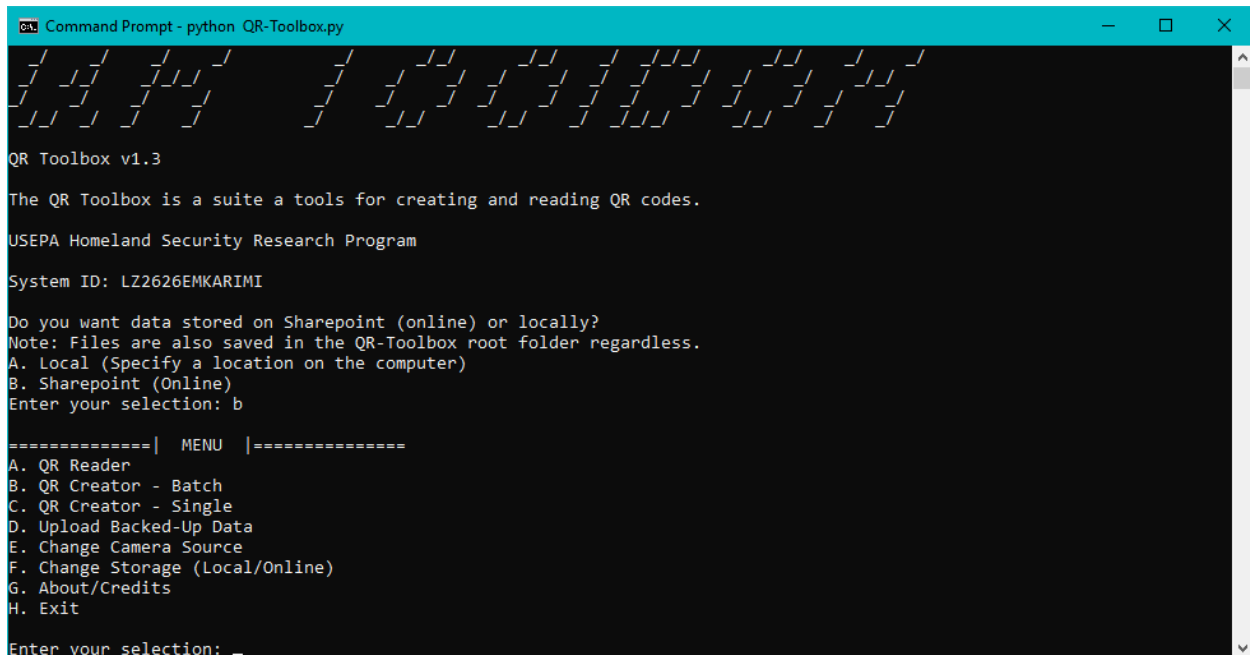
System ID: LZ2626EMKARIMI

Do you want data stored on Sharepoint (online) or locally?
Note: Files are also saved in the QR-Toolbox root folder regardless.
A. Local (Specify a location on the computer)
B. Sharepoint (Online)
Enter your selection: a

Storage directory established: C:/Users/mkarimi/PycharmProjects/QR_Tool/Temp

=====| MENU |=====
A. QR Reader
B. QR Creator - Batch
C. QR Creator - Single
D. Consolidate Records
E. Change Camera Source
F. Change Storage (Local/Online)
G. About/Credits
H. Exit
Enter your selection: _
```

Figure 4 Local Version Menu



```
Command Prompt - python QR-Toolbox.py

QR Toolbox v1.3

The QR Toolbox is a suite a tools for creating and reading QR codes.

USEPA Homeland Security Research Program

System ID: LZ2626EMKARIMI

Do you want data stored on Sharepoint (online) or locally?
Note: Files are also saved in the QR-Toolbox root folder regardless.
A. Local (Specify a location on the computer)
B. Sharepoint (Online)
Enter your selection: b

=====| MENU |=====
A. QR Reader
B. QR Creator - Batch
C. QR Creator - Single
D. Upload Backed-Up Data
E. Change Camera Source
F. Change Storage (Local/Online)
G. About/Credits
H. Exit
Enter your selection: _
```

Figure 5 SharePoint Version Menu

1. QR Reader: Scan the QR Code and the entry will be stored.
 - a. SharePoint version:

- i. QR Code entry is stored in the same folder as QR-Toolbox.py, and on the SharePoint site.
 - ii. User will be prompted to restart the last session, if there was one (allowing the user to pick up where they left off, if something happened (e.g. computer crashed, ran out of battery, etc.) or if they want to continue from a past session despite no error.
 - iii. SharePoint checks for internet connectivity, and if it is lost or another error occurs during network-related actions, Program will attempt the action again after a 10 and 30 second interval, and if those fail as well, the data will be stored locally. The user then has the option to upload that backed up data from the main menu (when the user has determined a network connection has been reestablished), or the Program will automatically do so after the next successful upload.
 - b. Local version:
 - i. QR Code entry is stored in root, and in the location the user specified.
 - ii. User will be prompted to restart the last session, if there was one (allowing the user to pick up where they left off, if something happened (e.g. computer crashed, ran out of battery, etc.) or if they want to continue from a past session despite no error.
 - c. *Notes: Program will reject QR codes created by other sources when it detects a compatibility issue.*
2. QR Creator – Batch: Generate QR Code image file for each entry defined in names.csv.
- a. SharePoint version:
 - i. Example entries in names.csv:
 - 1. Lastname.firstname@epa.gov
 - 2. *FirstName LastName*
 - ii. The names.csv must be available in the SharePoint site defined in the QRToolbox.py.
 - 1. *relative URL =
"/sites/Emergency%20Response/EOCIncident/EOC%20Documents/QRCodes/names.csv"*
 - 2. The user can modify the above line in QRToolbox.py to point to another SharePoint site.
 - iii. *Note: if there are any special characters (ex. 'È', 'è', etc.), the program cannot store these on SharePoint without replacing those characters with their regular equivalents ('è' = 'e'). The program will ask the user if they would like to do that,*

and if they respond 'No', that text/entry will be skipped. This will occur for every entry that has special characters.

- b. Local version:
 - i. Same examples as above
 - ii. The names.csv must be in the same folder as the QR-Toolbox.py file.
 - iii. The generated QR Codes will be stored in the root folder and in the folder the user specified.
3. QR Creator – Single: Generate QR Code for single entry. This option will prompt to enter email or other ID to generate QR Code image file for the entered value.
 - a. SharePoint version: The QR Code image file will be stored in the SharePoint Site defined in settings.py and the folder name as defined in QR-Toolbox.py: qrfolder = "QRCodes", and also stored in the same folder as QR-Toolbox.py.
 - i. *Note: if there are any special characters (ex. 'È', 'è', etc.), the program cannot store these on SharePoint without replacing those characters with their regular equivalents ('è' = 'e'). The program will ask the user if they would like to do that, and if they respond 'No', that text/entry will be skipped.*
 - b. Local version: QR Code image file stored in the root folder and in the location the user specified.
4. Consolidate Records (Local version only): This option is used to consolidate records into one file.
5. Upload Backed-Up Data (Online version only): This option triggers the Program to attempt to upload any backed-up data.
6. Change Camera Source
 - a. Determines which camera the QR-Toolbox should use. The user can choose from the following options:
 - i. Integrated webcam (built in camera, such as in a laptop)
 - ii. Separate webcam (camera the user connects via USB port or similar)
 - iii. PiCamera (a camera specifically for Raspberry Pis)
 - b. *Note: Integrated webcam is the default*
 - c. *Note: These mappings do assume that the computer has default settings regarding webcams. If those settings are different/changed or it is not a Windows computer that is being used, these mappings above may not be accurate. Example: Had plugged in USB Logitech on one laptop, where Option A (which should be integrated webcam) was actually the separate webcam/Logitech, and Option B (which should be the separate webcam) was actually the integrated webcam, thus in such a case, one would simply flip their choice.*
7. Change Storage (Local/Online)
 - a. When any data is created by the QR-Toolbox, it is stored in the same folder as the QR-Toolbox.py by default. However, the user has the option of also telling the software to

store that data either online on SharePoint, or somewhere else on their computer (locally).

- b. This function allows the user to change that secondary storage location.
- 8. About/Credits: Display information about the QR Code Tool.
- 9. Exit: Exit the QR Code Tool

Access the QR Records in SharePoint

A SharePoint workflow will automatically aggregate and consolidate the QR data from all input devices.