|  |  |
| --- | --- |
|  | Version  *1.3* |

QR Toolbox

U.S. Environmental Protection Agency

Text

Description automatically generatedUser’s Guide

u.s. Environmental protection agency

QR TOOLBOX

U.S. Environmental Protection Agency

Homeland Security Research Program

Research Triangle Park, NC 27711

**Disclaimer**

This tool was created by the U.S. Environmental Protection Agency through its Office of Research and Development’s Homeland Security Research Program (HSRP). The contents of this guide do not necessarily reflect the views of the Agency. Mention of trade names, products, or services does not convey official EPA approval, endorsement, or recommendation.

**Acknowledgements:**

Muhammad Karimi\*, Timothy Boe\*, Paul Lemieux\*, Jonathan Pettit\*\*, Catherine Young\*\*\*

\* EPA Office of Research and Development

\*\*EPA Office of Land and Emergency Management

\*\*\* EPA Region 1

Acronym/Abbreviation List

| **Acronym** | **Definition** |
| --- | --- |
| CESER | Center for Environmental Solutions and Emergency Response |
| CSV | Comma Separated Values |
| EPA | U.S. Environmental Protection Agency |
| HSRP | Homeland Security Research Program |
| URL  QR Code | Uniform Resource Locator  Quick Response Code |

**Table of Contents**

[Introduction 1](#_Toc56701138)

[How to Use This Guide 2](#_Toc56701139)

[Point of Contact 2](#_Toc56701140)

[Design 3](#_Toc56701141)

[Design Components 3](#_Toc56701142)

How to setup [SharePoint Online 4](#_Toc56701147)

[Run the Tool 8](#_Toc56701151)

[Starting the Tool 8](#_Toc56701152)

[Functions](#_Toc56701153) 9

[QR Reader 9](#_Toc56701154)

[QR Creator - Batch 13](#_Toc56701155)

[QR Creator - Single 14](#_Toc56701156)

Upload/[Consolidate 16](#_Toc56701157)

About/Credits 17

Exit 18

[Setup](#_Toc56701153) 19

[Change Camera Source 20](#_Toc56701154)

Change Storage Location 21

Ask Special Character Conversion 22

[Troubleshooting 25](#_Toc56701165)

**List of Figures**

[Figure 1. QR-Toolbox Design Components 3](#_Toc56701361)

Figure 2. SharePoint Client ID and Secret Generation 4

Figure 3. SharePoint Application Key Activation 6

Figure 4. settings.py file 7

FIgure 5. QR-Toolbox Icon 8

Figure 6. QR-Toolbox Startup 9

Figure 7. QR-Toolbox Reader 10

Figure 8. QR-Toolbox Upload Fail 11

Figure 9. QR-Toolbox New Session or Restart Session 12

Figure 10. QR-Toolbox Restart Session Example 12

FIgure 11. QR Creator - Batch Example 13

Figure 12. QR Creator - Batch Example 2 14

Figure 13. QR Creator - Single Example 15

Figure 14. QR Creator - Single Example 2 15

Figure 15. Upload/Consolidate Function 16

Figure 16. Upload Triggered but no data Example 16

Figure 17. Consolidate Records Function Example 17

Figure 18. About/Credits Button 18

Figure 19. Exit Button 19

Figure 20. Setup Options 20

Figure 21. Change Camera Source Options 21

Figure 22. Change Storage Location 22

Figure 23. Special Character Conversion Options 23

Figure 24. Special Character Conversion - Choosing Yes Example 23

Figure 25. Special Character Conversion - Choosing No Example 24

**List of Tables**

# Introduction



1

CHAPTER

Learn about the Quick Response (QR) Toolbox. A desktop-based tool for creating QR Codes one-by-one or en masse, along with the ability to read and log those codes to keep track of personnel or items/equipment.

EPA’s Center for Environmental Solutions and Emergency Response (CESER) within the Office of Research and Development (ORD) conducts applied, stakeholder-driven research and provides responsive technical support to help solve the Nation’s environmental challenges. The Center’s research focuses on innovative approaches to address environmental challenges associated with the built environment. CESER develops technologies and decision-support tools to help safeguard public water systems and groundwater, guide sustainable materials management, remediate sites from traditional contamination sources and emerging environmental stressors, and address potential threats from terrorism and natural disasters. A key element of oversight during an emergency response is the process of tracking equipment and resources. For large-scale incidents, this can often require tracking thousands of records across a large geographical area. This has historically been accomplished using paper or electronic methods; however, this process may entail a significant amount of manual effort to maintain, be error prone, or require the use of expensive proprietary software or hardware to facilitate.

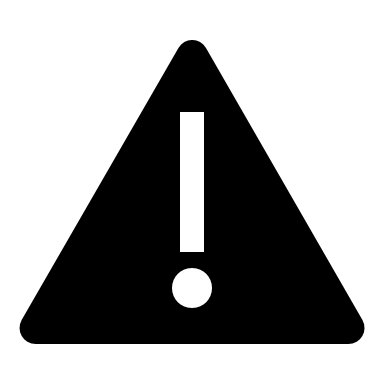
As part of the Homeland Security Research Program’s (HSRP’s) ongoing collaborative efforts with the EPA Regions and Office of Emergency Management, HSRP developed a system using commercial off-the-shelf webcams and open source software for generating or scanning quick response (QR) codes as a means for recognizing, recording, and sharing the location, duration, and status of resources. The system can easily be networked and communicate with satellite locations, maintaining a centralized database of records. The QR Toolbox serves as a free-to-use, customizable, and easily deployable solution capable for tracking assets in the field during emergency responses.

## How to Use This Guide

The purpose of this guide is to provide the necessary information to operate the tool. Described in this guide are the design and development details and choices for the program, as well as the information describing how to access and operate the tool. Appendix A provides a brief Quick Start Guide.

**icon keY**

 Valuable Tip

 Important Note

The “icon key” to the right contains symbols used throughout this guide to highlight important information and additional guidance.

## Point of Contact

Timothy Boe

US EPA Office of Research and Development

Center for Environmental Solutions and Emergency Response   
Systems Tools and Materials Management Branch  
919-541-2617 office  
919-541-0496 fax  
[boe.timothy@epa.gov](mailto:boe.timothy@epa.gov)

# Design



2

CHAPTER

*Understand the tool’s underlying design, methodology, and workflow*

The QR Toolbox is a Windows 10 Python-based application that uses a built-in or external webcam to scan QR codes.

## Design Components

The QR Toolbox was built using Python (3.7.4), Kivy, Office365 REST Python API, and SharePoint Online. The toolbox is capable of storing scanned QR codes locally (i.e., offline mode) or online using Microsoft SharePoint. The Toolbox uses the Office365 REST Python API to connect to the SharePoint Online Service to upload or download the files necessary to operate the application. **Figure 1** demonstrates the QR Toolbox design components.

Diagram

Description automatically generated

Figure 1. QR Toolbox Design Components

# Setting up SharePoint Online



3

CHAPTER

*How to setup the QR Toolbox online mode*

When operating in online mode, the QR Toolbox uses SharePoint to store and retrieve scanned QR data across an organization’s network. To use the online mode, a SharePoint Application Key is required. An Application Key can be generated by SharePoint through administrator access or by contacting the system administrator. **This process is only necessary if the QR Toolbox is being used external to EPA. EPA personnel should use the provided Application Key and continue to Chapter 4.**

* The instructions for setting up the application key are located here: <https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/security-apponly-azureacs>
* The SharePoint directory and setup will vary by organization and license. Example A shows the EPA Emergency Response SharePoint site URL, and Example B shows the Application Key creation link. Figure 2 shows the Application Key creation dialog.
  + Example 1: <https://usepa.sharepoint.com/sites/Emergency%20Response/EOCIncident/>
  + Example 2: <https://usepa.sharepoint.com/sites/Emergency%20Response/EOCIncident/_layouts/15/AppRegNew.aspx>

Graphical user interface

Description automatically generated

Figure 2. SharePoint Client ID and Client Secret Generation

* From the SharePoint Client ID and Client Secret Generation dialog, Click the “Generate” button to generate a Client Id and a Client Secret. Write down the Client Id and Client Secret for use in the next step. Enter a “Title” and click the “Create” button.

*Note: The following are examples and may be different depending on organization and SharePoint version.*

**Title**: insert title

**App Domain**: www.localhost.com

**Redirect URI**: https://www.localhost.com

* To grant the necessary permissions, take the URL from the last step and remove the “/\_layouts/15/AppRegNew.aspx”, and instead add “/\_layouts/15/AppInv.aspx” at the end, as shown in the example link below. Navigate to the URL, and a screen similar **Figure 3** should appear.
* The Application Key Activation dialog is used to enable the necessary permissions to read and write data to SharePoint remotely. At this screen, enter the Client ID (which was acquired from the last step) into the App ID field and click “Lookup”. Then fill out the remaining fields with the same information provided in the previous steps. Finally, copy the XML (shown below) into the Permission Request XML field as shown in **Figure 3**. This XML code describes the permissions needed by the application. Click the create button to finish.

Example URL of this step: <https://usepa.sharepoint.com/sites/Emergency%20Response/EOCIncident/_layouts/15/AppInv.aspx>



Figure 3. SharePoint Application Key Activation

Repeated Instructions from the last page:

* Enter the Client ID and click the ‘Lookup’ button.
* Fill out the other fields with the same information provided in the previous steps.
* 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.

**Modify settings.py**

The settings.py contains the SharePoint Client ID and Client Secret that were generated above, as well as the URL to the organization’s SharePoint site. The application uses the settings.py file to pass credentials to SharePoint in order to transfer and store data. It also contains other variables used for different purposes in the application. These will be further explained below:

* Locate and modify the settings.py file, which – if not changed – the default installation location during installation would be in “C:/Users/<user>/AppData/Local/Programs/QR-Toolbox/Setup”, where <user> is the user name of the computer the Toolbox is installed on. The settings.py file should look similar to **Figure 4** below. Replace the <Your URL>’ with the organization’s SharePoint site URL (keep the quotes). Replace the ‘<Your Client Id>’ with the generated Client Id, and the ‘<Your Client Secret>’ with the generated Client Secret.
* As for the other variables in the file:
  + listTitle = This variable contains the name of the SharePoint list the QR Tool uploads to when checking users in and out. The default name is ‘QR Timestamps’, however the name can be changed to what the user has defined their list as.
  + qrfolder = This variable contains the name of the folder that QR Codes are stored in on the SharePoint site. The name defaults to ‘QRCodes’ but can be changed.
  + bkcsvfolder = This variable contains the name of the folder that CSV Files are stored in on the SharePoint site. The name defaults to ‘HXWTEST’ but can be changed.
  + mainDirectory = This variable contains the name of the folder that is one level above and contains the qrfolder and bkcsvfolder (it must contain those folders). The name defaults to ‘EOC Documents’ but can be changed to whatever the upper level directory is named.
  + localQRBatchFile = This variable contains the name of the CSV file that the ‘QR Creator – Batch’ function pulls from when creating a batch of QR Codes. The name defaults to ‘names.csv’ but can be changed.
  + relative\_url = This variable contains the name of the CSV file that the online version of ‘QR Creator – Batch’ function pulls from when creating a batch of QR Codes. The path defaults to what is mentioned below but can be changed.

settings = {

'url': '<Your URL>',

'client\_id': '<Your Client Id>',

'client\_secret': '<Your Client Secret>',

‘listTitle’: ‘QR Timestamps’,

‘qrfolder’: ‘QRCodes’,

‘bkcsvfolder’: ‘HXWTEST’,

‘mainDirectory’: ‘EOC Documents’,

‘localQRBatchFile’: ‘names.csv’,

‘relative\_url’: ‘/sites/Emergency%20Response/EOCIncident/EOC%20Documents/QRCodes/names.csv’

}

Figure 4. settings.py file

*Note: The Client ID, Client Secret, or SharePoint URL must be updated prior to running the tool.*

# Run the Tool



4

CHAPTER

Understand the steps necessary to operate the tool and its features

## Access the Tool

To start the application, either double-click the shortcut named “QR-Toolbox” (pictured below) located on the desktop (if specified during installation) or go to the Start Menu and search for the QR-Toolbox application. **Figure 5** shows the QR-Toolbox icon.

A picture containing graphical user interface

Description automatically generated

Figure 5. QR-Toolbox Icon

## Startup

At startup, the user will be asked to choose a storage location (pictured below). Simply choose the preferred storage location (local or online). If online mode is chosen, confirm that the computer has an active internet connection and that the SharePoint Application Key process has been completed. If local mode is chosen, the user will be prompted to select a directory for storing files (see SETTINGS | *Choose Storage Location*).

As depicted in **Figure 6** below, the application consists of a single window, with a series of buttons at the bottom. Each button has its own function, except for the ‘Setup’ button, which opens an additional menu. All important information is displayed via the status screen that is located in the center of the application (above the buttons). The status screen is scrollable, either by using a mouse scroll wheel or by clicking the screen and dragging up or down (Note: laptop touchpad scrolling is somewhat difficult and finicky, it is better to use one of the other two methods).

Graphical user interface, application

Description automatically generated

Figure 6. QR-Toolbox Startup

## FUNCTIONS

## *QR Reader*

The QR Reader function uses the camera connected to your computer to read QR Codes, as seen below in **Figure 7**. The QR Codes, when read, are highlighted on the camera window with a red box around them. Upon scanning, the data stored in the code will be displayed on the screen, along with other information, such as the Desktop ID of the computer currently in use, the current time, and whether the data corresponding to that QR Code is ‘checking in’ or ‘checking out’.

*Note: QR Codes not generated by the QR Creator (described below) may be rejected by the system due to incompatibility issues*.

The QR Reader uses the ‘checking in’ and ‘checking out’ mechanism to keep track of the current status of personnel as well as items/equipment with QR codes attached. When a code is checked out, the system will also include the time elapsed duration. These data are stored in a .txt file, and at the end of the session, are then stored in a .CSV file. The CSV file is stored in 3 separate locations, depending on the systems settings:

1. Archive folder – this folder is located in the root folder of the program (i.e., C:/Users/<your user>/AppData/Local/Programs/QR-Toolbox/Archive). *The system will store all CSV and generated QR Codes here by default.*
2. If local mode – stored in the user specified directory; and
3. If online mode – stored on the SharePoint site.

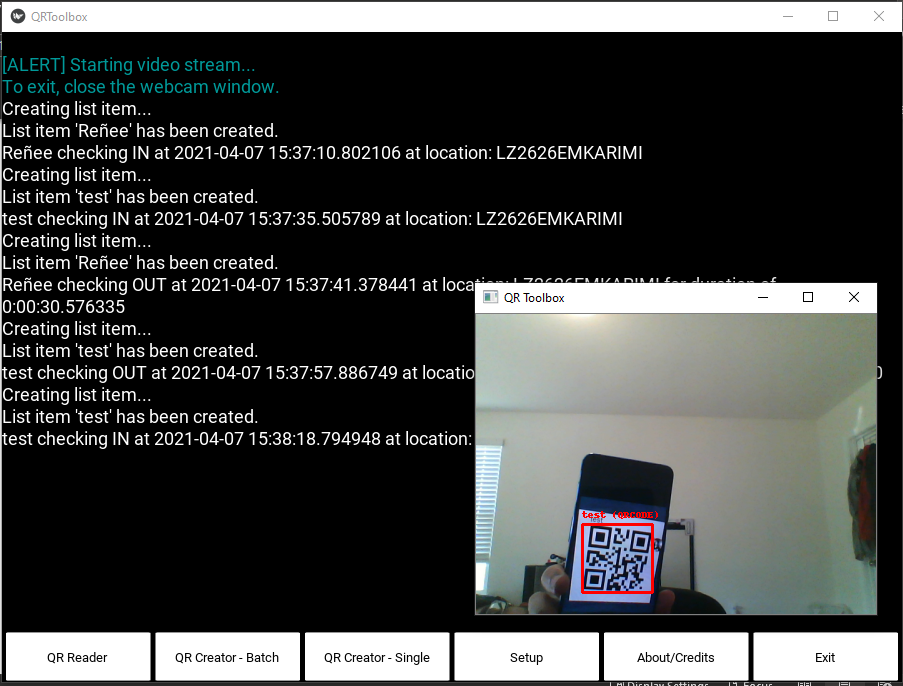


Figure 7. QR-Toolbox Reader

As demonstrated in **Figure 8**, when operating in the online mode and the system fails to upload the scanned QR code or the CSV file, the system will wait for 10 seconds and then try again, and if unsuccessful, the system will try again in 30 seconds. If the upload process continues to fail, the data will be backed up in a local file. The system will attempt to upload the data once an internet connection is successfully established (either in the QR Reader function, or in the QR Creator functions). The upload process can be triggered manually via the “Setup->Upload/Consolidate” function that is described later in this chapter.



Figure 8. QR-Toolbox Upload Fail

**Sessions**

Upon starting the QR Tool, the user will be notified as to whether they want to restart an old session or create a new one, as seen in **Figure 9 and 10** below. The QR Reader automatically saves the data from its last run or session (even in the event of a system crash), so that the user can restart the application and resume operation. A new session can be started by selecting the ‘New Session’ button. Note that this action will delete any previous unsaved records.

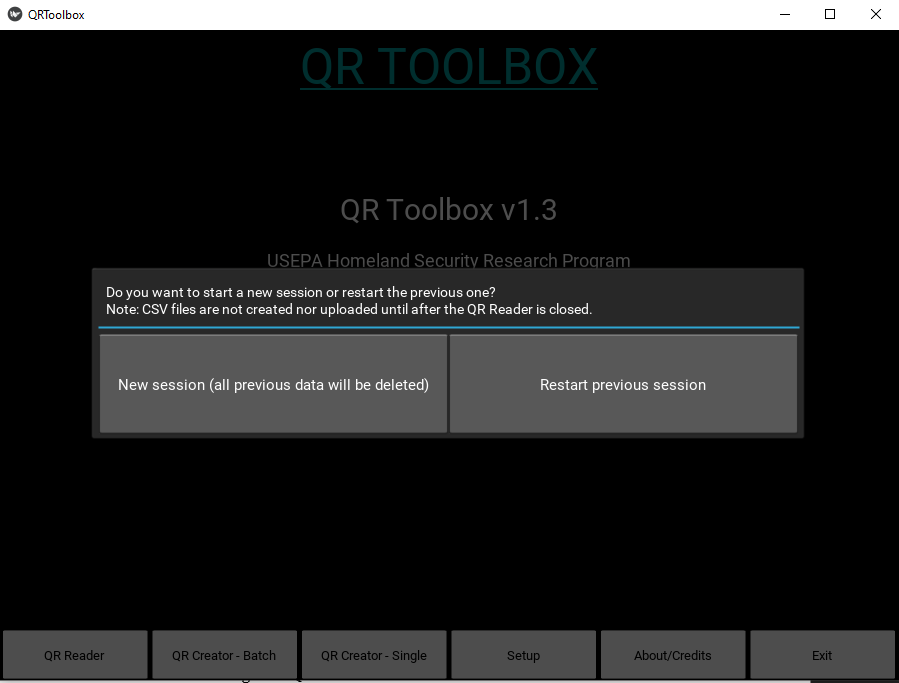


Figure 9. QR-Toolbox New Session or Restart Session



Figure 10. QR-Toolbox Restart Session Example

**Exiting the Session**

The QR Reader can be closed by clicking the ‘X’ close button on the QR Reader window.

## *QR Creator - Batch*

This function allows the user to quickly generate a large number of QR Codes. Run the QR Creator – Batch function by selecting the QR Creator – Batch button. The system loads and reads data stored in the user supplied CSV file and generates a QR code for each record. This may take several minutes, depending on the size of the csv file (i.e., the number of records). Once complete, text will be displayed on the screen informing the user of a successful run, as displayed in **Figures 11 and 12**.

Setup

The following steps are required in order to successfully run batch mode:

* Local mode:
  + Locate the installation folder for the QR-Toolbox, if default it will be “C:/Users/<Your user>/AppData/Local/Programs/QR-Toolbox”
  + Create and place a CSV file called ‘names.csv’ in the root folder, i.e. QR-Toolbox. You can change the name of this file, but you must also change the name used for the ‘localQRBatchfile’ variable in the Setup/settings.py file to the same name. In this file, the first two columns (or both) should contain the data/name/information that will be converted to a QR Code. Each line represents a new code. There are no limitations on the number of codes that can be generated other than standard memory or file size limitations.
* Online mode:
  + This concept is similar to the above, except the ‘names.csv’ file is stored on the SharePoint site where the QR Codes are saved. The name for this file can also be changed, and you must then also change the value used for the variable ‘relative\_url’ in the Setup/settings.py file so that it reflects the same file name.

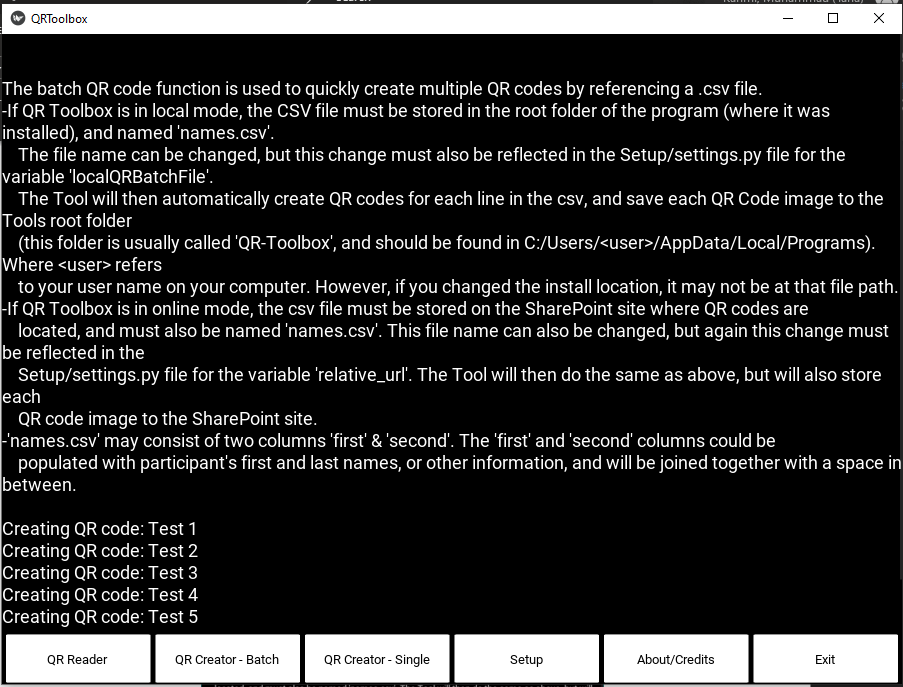


Figure 11. QR Creator – Batch Example

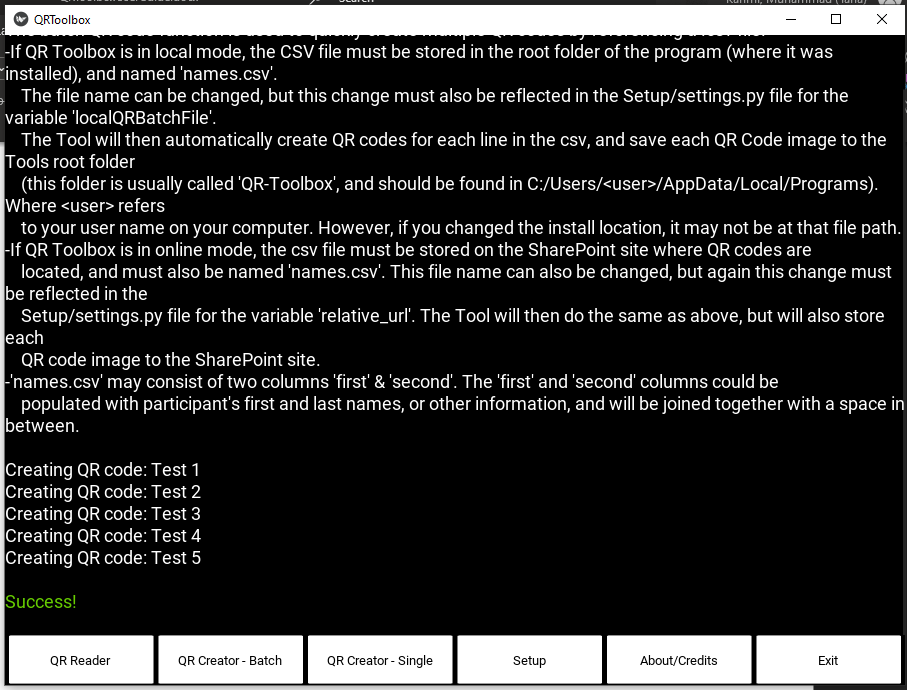


Figure 12. QR Creator – Batch Example 2

## *QR Creator - Single*

This function enables the creation of a single QR Code. To do this, click the ‘QR Creator – Single’ button on the application window, and then enter the text that the QR code will represent (Note: Only English and Latin characters are accepted.). Once the text has been entered, press the “OK’ button. The system will create a new QR code and notify the user, as shown in **Figure 13 and 14**.

When operating in online mode, confirm that the SharePoint and the settings.py are set up properly, as demonstrated in Chapter 3. The generated QR Code will then be stored in the Archive folder as well as the QR Codes folder on SharePoint. In local mode, the Code will be stored in the Archive folder as well as in the storage location specified by the user.

Graphical user interface

Description automatically generated

Figure 13. QR Creator – Single Example

A picture containing timeline

Description automatically generated

Figure 14. QR Creator – Single Example 2

## *Upload/Consolidate*

This function, as shown in **Figure 15** in the Setup menu, either uploads backed-up data or consolidates all of the CSV files in the chosen location into a single CSV file. The function of this button changes depending on the connection status (i.e., online or offline mode).

Graphical user interface

Description automatically generated

Figure 15. Upload/Consolidate Function

**Online mode**: the system will upload any data that failed to upload in a previous session as shown in **Figure 16**. Take care not to delete or modify anything contained in the System\_Data folder.

*Note: if the upload fails, the system will attempt to reestablish a connection in 10 seconds and again at 30 seconds, the system will back up the resulting data and (automatically) attempt the upload at a later time. This process may be initited by the user by clicking the “Upload/Consolidate” button.*

A picture containing text

Description automatically generated

Figure 16. Upload Triggered, no data example

**Local mode**: the system collects all of the CSV files located in the local folder (chosen by the user), and consolidates them into a single CSV file, as shown in **Figure 17.** The system only does this with CSV files that start with “QRT-R-”, contain “\_”, and end with “.csv”.



Figure 17. Consolidate Records Function Example

## *About/Credits*

This function (as seen in **Figure 18** below) shows information about the QR-Toolbox and contributions. Note: this process will clear the screen.

Text, timeline

Description automatically generated

Figure 18. About/Credits Button

## *Exit*

As shown in **Figure 19** below, this function brings up the Exit dialog, which prompts the user to exit the software. The ‘X’ Close button in the top right corner will also initiate this dialog. The dialog confirms that the user wants to close the program, as closing the program while running other functions may cause a loss of data (*Note: If the QR Reader function is running and the software is closed, data will be lost*).

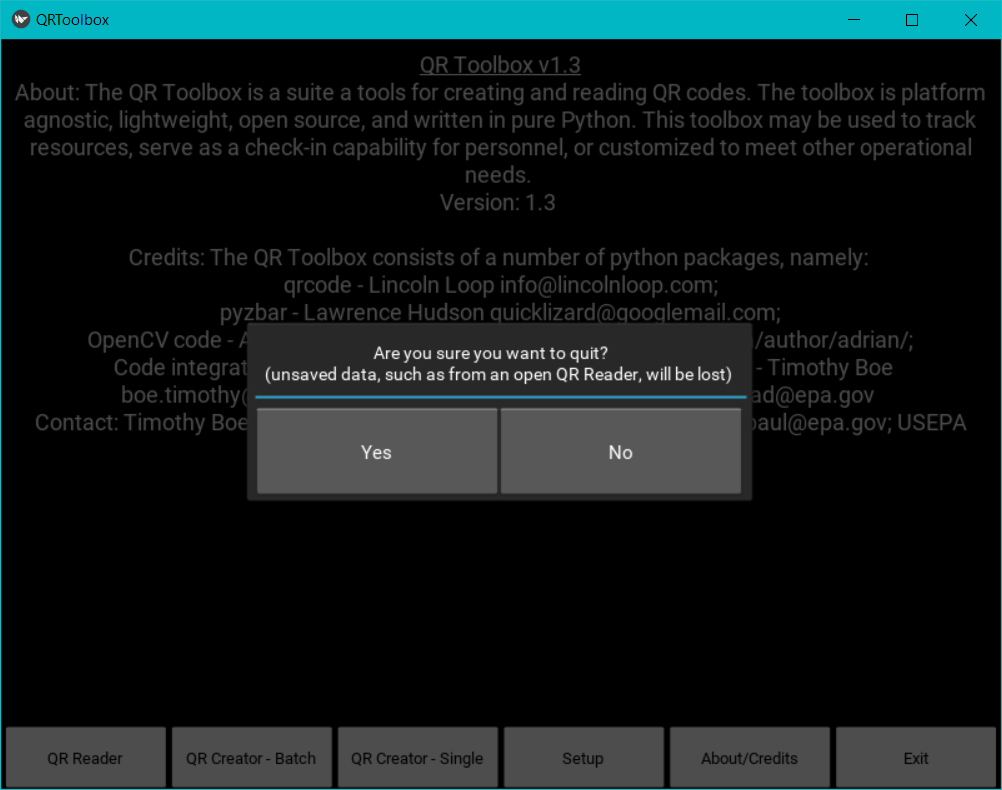


Figure 19. Exit Button

## SETUP

As shown in **Figure 20** below, clicking the “Setup Button” provides access to the following options: Upload/Consolidate (described above), Change Camera Source, Change Storage, and Special Character Conversion.

Graphical user interface, website

Description automatically generated

Figure 20. Setup Options

## *Change Camera Source*

The “Change Camera Source” settings determines which camera the QR Reader uses when reading QR Codes. The source may be 1 of the following 3 options (also demonstrated in **Figure 21** below):

1. The built-in camera (if equipped)
2. An externally connected camera
3. A Raspberry Pi Camera/PiCamera (*button is* *currently disabled; it was left as a disabled choice to serve as a placeholder for the future*)

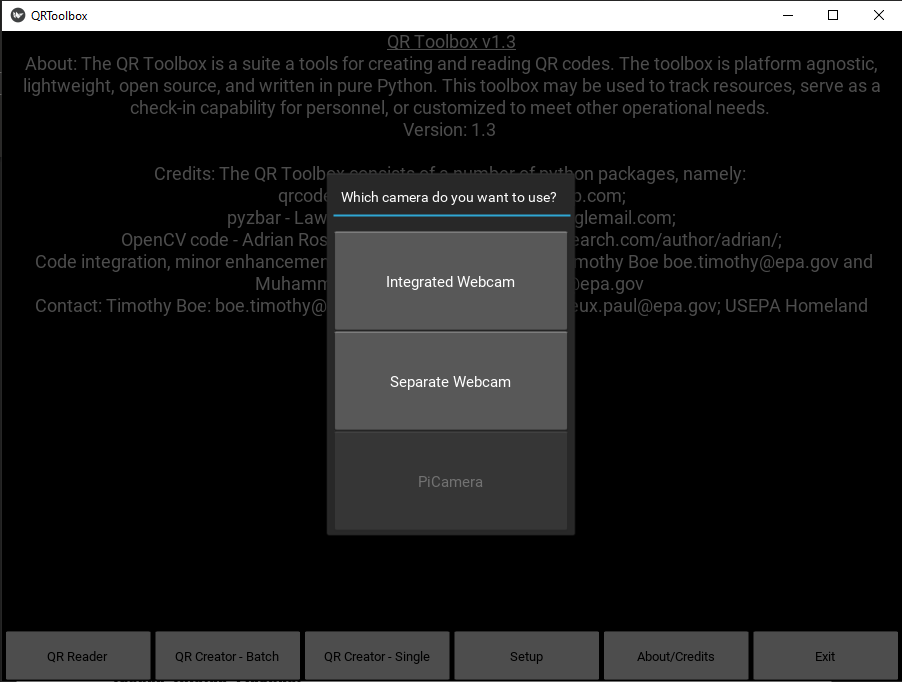


Figure 21. Change Camera Source Options

## *Change Storage Location*

This setting allows the user to specify where QR Codes will be saved (either online [SharePoint] or local mode [user chooses a folder]), as shown in **Figure 22** below.

*Note: Files are saved in the Archive folder regardless (which is in the installation folder of the QR-Toolbox).*

Graphical user interface, application

Description automatically generated

Figure 22. Change Storage Location

## *Special Character Conversion*

This setting allows the user to specify how the application manages special characters (i.e., convert to regular characters or skip them entirely) as seen in **Figures 23, 24, and 25** below. Special characters are those that contain ASCII punctuation & symbols. Generally, the application cannot read QR Codes that contain these special characters. To address this issue, when the system generates QR codes, it converts any special characters into regular or English equivalent characters (e.g., é becomes e). In offline mode, these codes are converted back when the system writes or displays the code in the status box. However, when operating in online mode, special characters do not write properly and must be (permanently) converted to regular characters and saved to SharePoint. By selecting the “No (those QR codes will be skipped)” option, the application will skip all objects/names that contain special characters. By default, the program will convert special characters. The user also has the option to toggle the setting back on, allowing the program to convert codes with special characters to regular characters.

*Note: This function is disabled when the user is in Local Mode, as it has no effect in Local Mode.*

Graphical user interface

Description automatically generated

Figure 23. Special Character Conversion Options

A picture containing text

Description automatically generated

Figure 24. Special Character Conversion – Choosing Yes Example

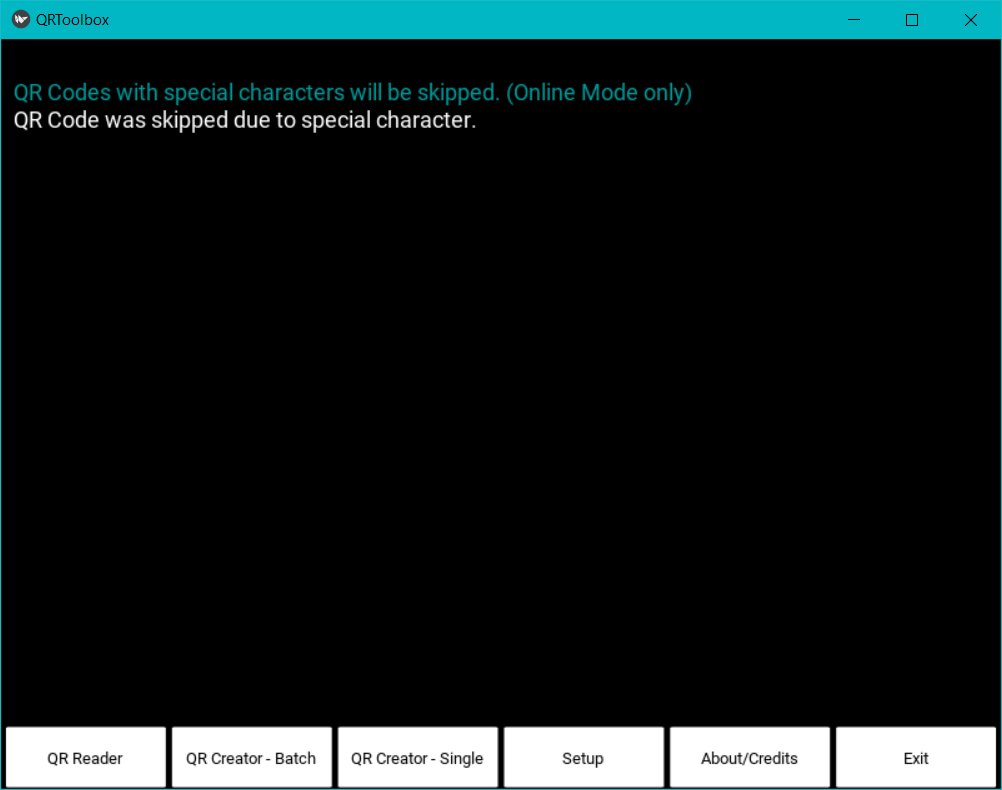


Figure 25. Special Character Conversion – Choosing No Example

# Troubleshooting



5

CHAPTER

Read about issues and resolutions to common problems

This chapter provides solutions to commonly found problems that users may encounter when using QR-Toolbox.

**Table 1. Troubleshooting Guide**

| **Problem** | **Cause** | **Remedy** |
| --- | --- | --- |
| Program doesn’t run | If program doesn’t run after installation, it’s likely due to a missing Microsoft Visual Code Redistributable. | In the installation folder of the QR-Toolbox locate the ‘Setup’ folder. The folder contains two files, ‘vcredist\_x64.exe’ and ‘vcredist\_x86.exe’. To figure out which one of those to use, go to the Start Menu and search for ‘System Information.’ In the ‘System Information” locate the ‘System Info’ line. If the value for ‘System Info’ contains ‘x64’, then run and install the ‘vcredist\_x64.exe’, and if the value has ‘x32’ or ‘x86’, then run and install the ‘vcredist\_x86.exe’. |
| Online doesn’t work | This can occur if SharePoint has not been setup, or due to an issue with internet connection. If neither of these things are the issue, then there could be an issue with the code. | Confirm the SharePoint (Chapter 3) instructions were correctly followed and that the system has an active internet connection. |
| QR Reading fails or causes a crash | Likely due to trying to read a QR Code that was not generated by the QR-Toolbox or had characters that aren’t supported. Those kinds of QR Codes may or may not be compatible with the QR-Toolbox. | Generate the QR Code using the QR-Toolbox and rescan. |
| QR Creation fails | Not Latin or English characters. | Ensure only English and Latin characters are being used. |
| Built-in Camera doesn’t start | A common error on Windows computers. | Restart the computer and check system drivers if the problem persists |
| External Camera doesn’t work when user does not have a built-in camera | If the system lacks a built-in camera or the user selects an external camera that does not exist. | If the user does not have a built-in webcam, they should select the ‘Integrated Webcam’ option, as that is the default source for their computer. |

This concludes the User’s Guide. Technical or troubleshooting questions may be emailed to the point of contact listed in Chapter 1.

U.S. Environmental Protection Agency

Homeland Security Research Program

Research Triangle Park, NC 27711