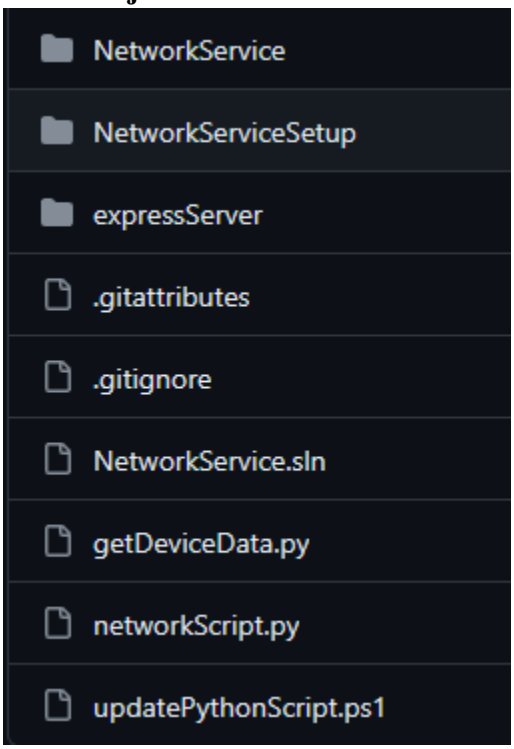
Ben Egloff
Prof. Aaron Striegel
SP24 Undergraduate Research

Introduction

For my research this semester I created a windows application that can be run as a background service that collects information about a device's current internet connection and information about the wifi neighborhood their device is currently in. The service is currently packaged and can be installed on any windows device by running the NetworkServiceSetup.msi file and following the install wizard setup. Currently, the output of the service is simply redirected to a localhost route, which can be established using the included 'expressServer' code included in the github deployment.

Project Folder Structure



The project contains 3 separate folders and various python scripts.

The NetworkService folder contains the basis for the C# service that can be loaded into Visual Studio. Within the NetworkService project Service1.cs drives the service execution, like picking which executable file to run, and how often to run said executable.

The NetworkServiceSetup folder contains the logic for packaging the NetworkService into an msi file that can be shared to install the service on any windows computer.

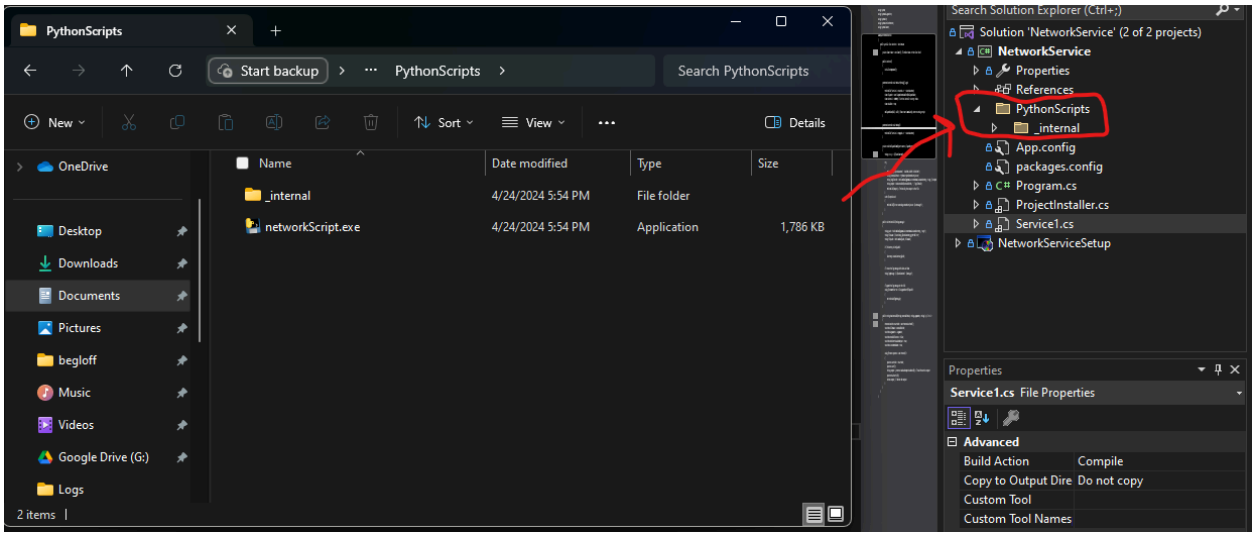
The expressServer folder contains a simple expressServer that sets up a post route at localhost:3000/api/v1/networkinfo that saves the json data sent over the network to a file.

The networkScript.py and getDeviceData.py files are the scripts that are executed every 10 minutes. networkScript.py is the main script, which calls getDeviceData internally.

UpdatePythonScript.ps1 is a powershell script that can be ran whenever networkScript.py or getDeviceData.py are updated to create a .exe file that is used in the service execution.

Updating Python Scripts

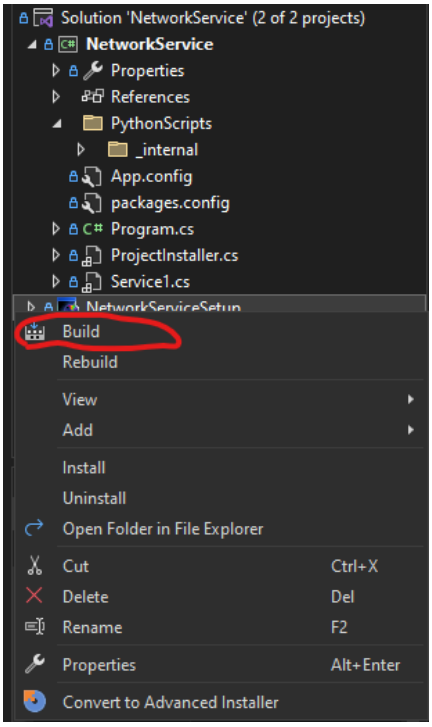
Updating the python scripts is a relatively simple process. Simply run ./updatePythonScript.ps1 to run the necessary steps to create the new .exe file and update the file structure. To ensure that your changes are reflected correctly on Visual Studio, make sure to delete the existing _internal directory and networkScript.exe that already exist in the visual studio project. You will need to copy the newly generated _internal directory and networkScript.exe files from your local filesystem to NetworkService/PythonScripts WITHIN VisualStudio.



As shown in the picture make sure you copy to the PythonScripts folder within the VisualStudio project.

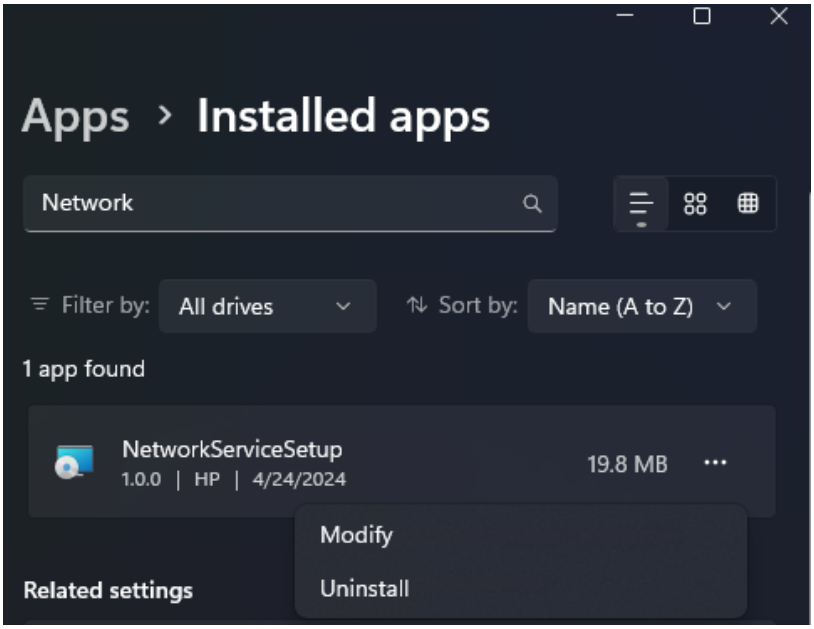
Updating C# Scripts and Propagating Changes

After any C# Script editing, Python script editing or other configuration changes, it is necessary to build the NetworkServiceSetup project to implement changes in the msi file and NetworkService windows service as well. By right clicking on ‘NetworkServiceSetup’ and clicking build, both the service and installer will be updated.

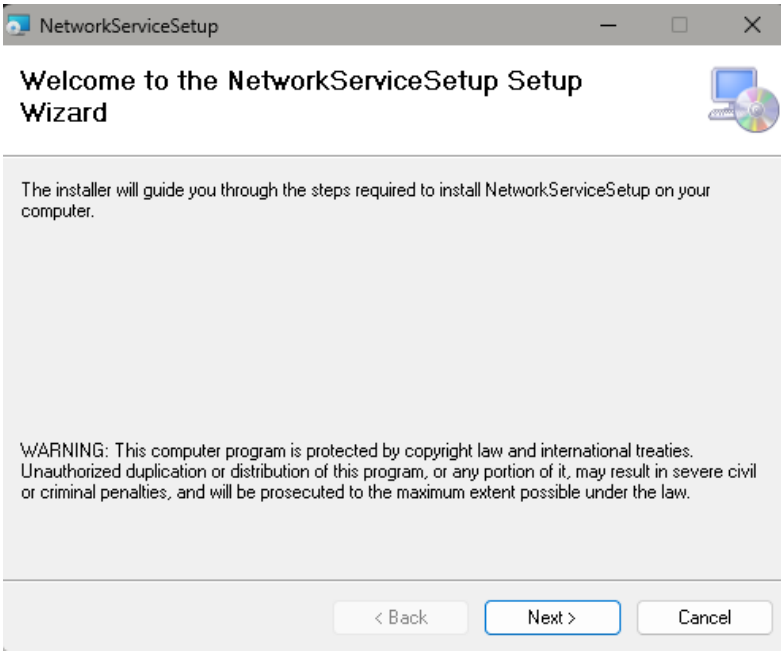


Running Service After Build

After building the NetworkServiceSetup project, you can distribute the msi file found in NetworkServiceSetup/Debug to any windows device for install. If you have previously installed the service and want to install a new version, you must first uninstall the application by navigating to ‘Add or remove programs’ on your computer. After searching NetworkService, you should be able to uninstall the existing service.



Following an uninstall if necessary, running the msi file mentioned above and proceeding through the install wizard will result in the service being installed on your computer.



Once you have a successful installation, you can run the service by opening the ‘Services’ application and starting the service called ‘NetworkService’. This service will run in the background and ship network information every 10 minutes to the expressServer or some other route if specified.