Comp 4985 - Assignment 1

Nicole Jingco, A01001875

Overview

This document outlines the requirements, state diagram and pseudocode for the Lookup Application. This program allows the user to select the lookup option from the menu item and displays the converted output on the main window. The window will show the users original input as well as the reverted output. Conversion is done using the Winsock 2 API.

Requirements

The requirements for this program follows the design and functionality listed below.

Constraints

The program requirements must have the following:

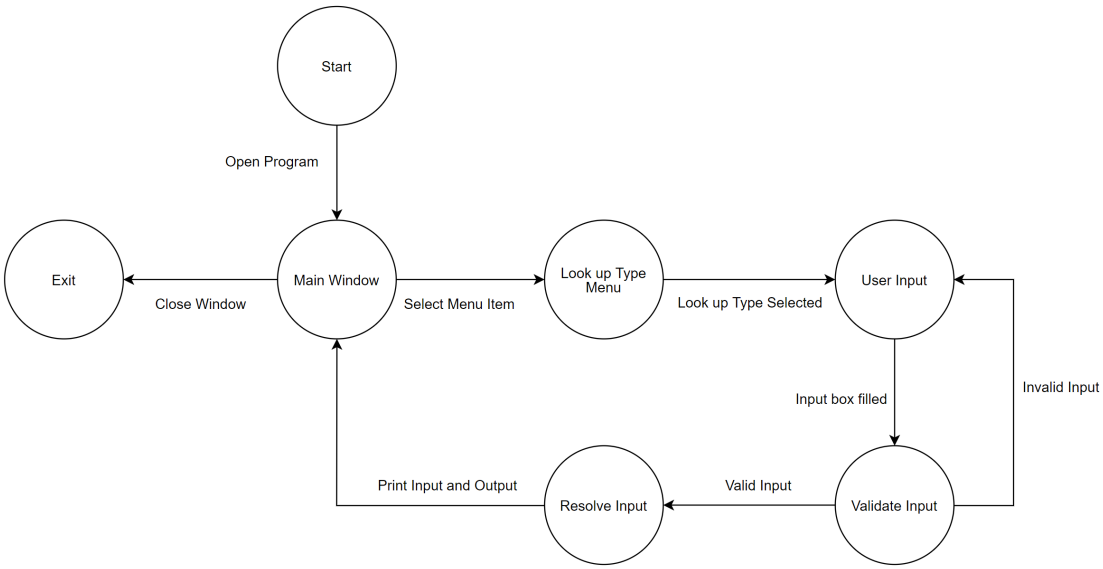
- Have a menu item to select from to do the following:
    1. Take a user specified host name and resolve it into a IP address.
    2. Take a user specified a IP address resolve it into an host name(s).
    3. Take a user specified service name/protocol and resolve it into its port number.
    4. Take a user specified port number/protocol and resolve it into its service name.
- Have a user input box
- Display the original user input and resolved output
- Must use the Winsock 2 API to resolve the lookup

Design

The design requirements must have the following:

- The design must be a Windows Menu-Driven Application

State Diagram



Pseudocode

```
// Exit - Closes the program
exit(){
    close program
}

// Main Window - Open the window
create_window(){
    create a window
    add menu items
    add input box
    add okay button
}

// Menu - sets how the input is getting converted
set_menu_item(){
    if menu item is "host name to ip"
        set lookup type to "host name to ip"
    else if menu item "ip to host name"
        set lookup type to "ip to host name"
    else if menu item "service to port"
        set lookup type to "service to port"
    else if menu item "port to service"
        set lookup type to "port to service"
    else
        Print "nothing selected from the menu item"
}

// User Input - Check is input box is filled
check_input_box(){
    if user input box is not empty
        return true
    Print "inupt box not filled"
    return false
```

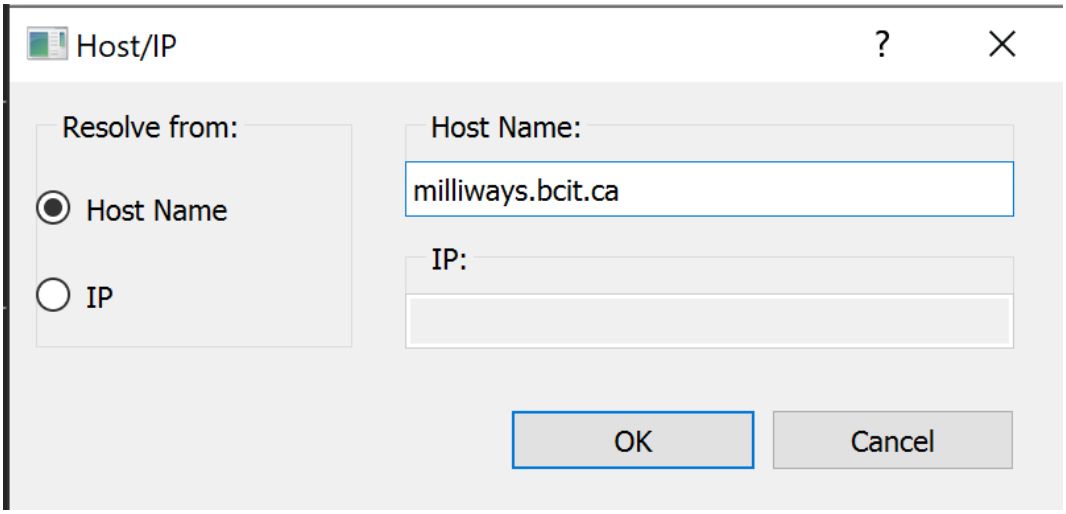
```
}

// Validate Input - Check if input is valid fro the chosen menu item
validate(input){
    if lookup type is "host name to ip"
        if input is a valid host name
            return true
    else if lookup type "ip to host name"
        if input is a valide ip
            return true
    else if lookup type is "service to port"
        if input is a valid service
            return true
    else if lookup type is "port to service"
        if input is a valid port
            return true
    else
        return false
}

// Resolve Input - Conver the input to the associated menu item
resolve_input(){
    if input is valid
        convert the input to its associated menu item
        print input
        print output
    else
        Print "invalid input"
}
```

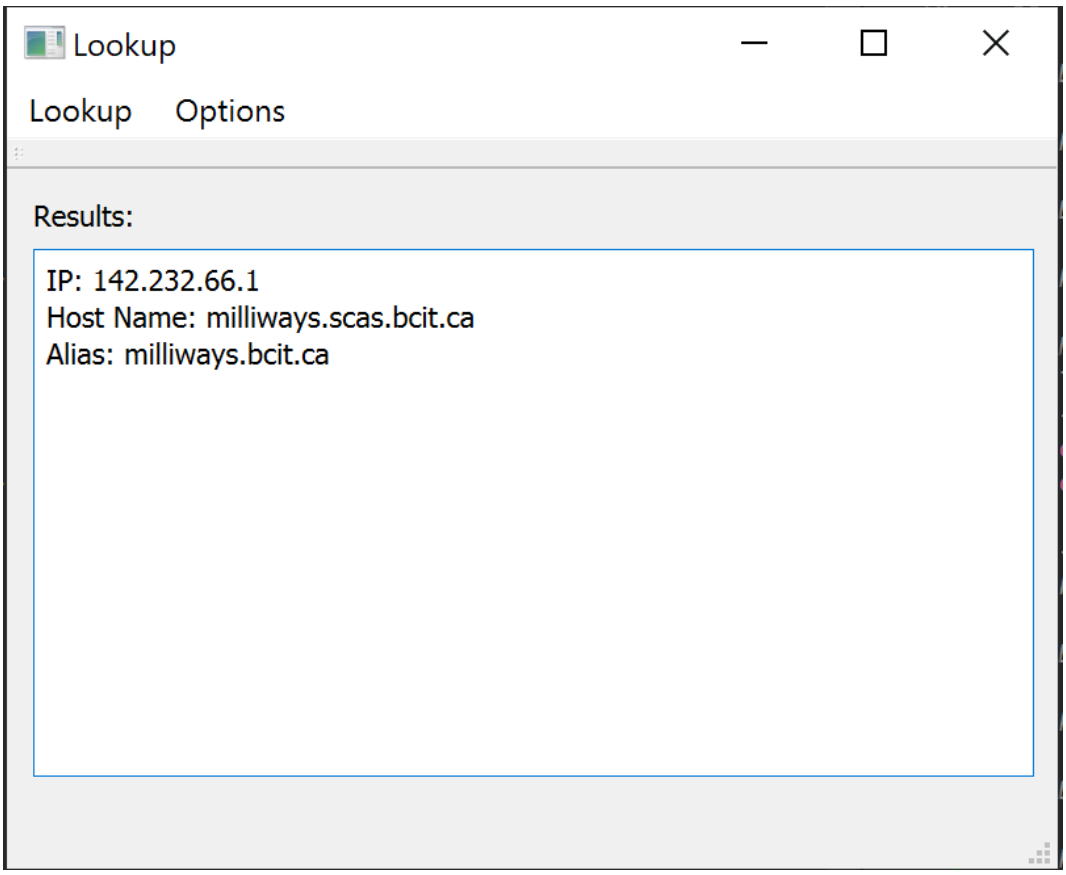
Testing

The following are test result to the listed conditions:

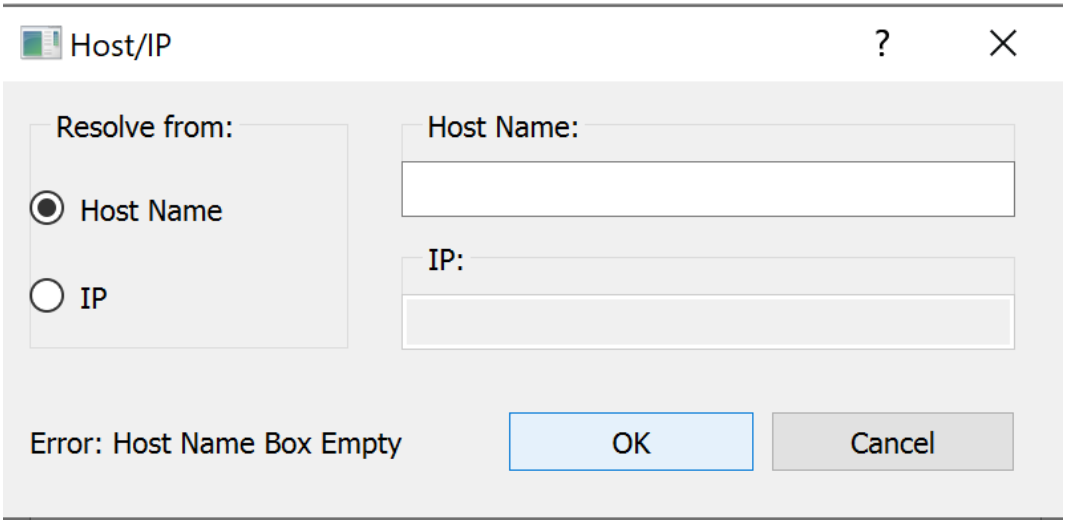
Host Name/IP

Host Name

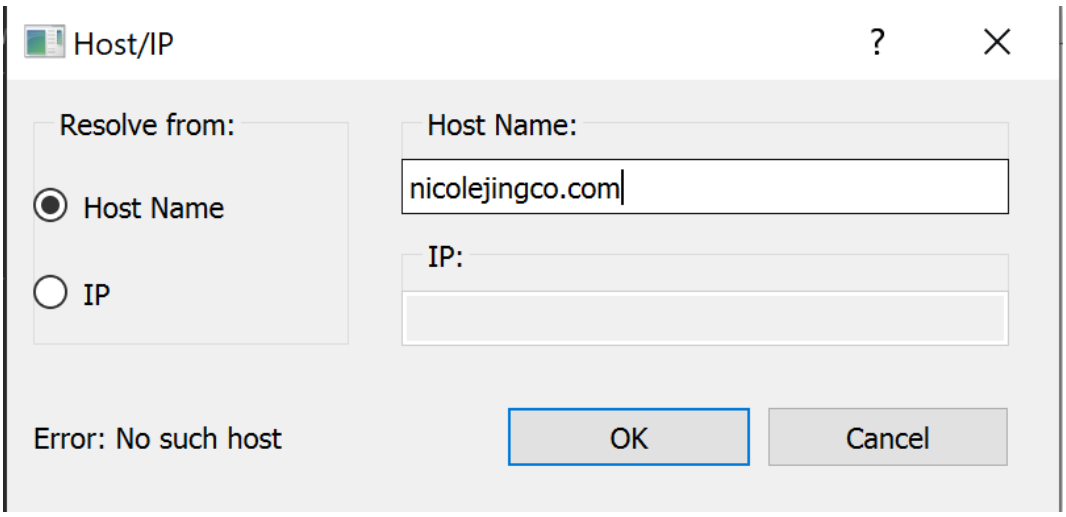
1. Host Name Found
    - 

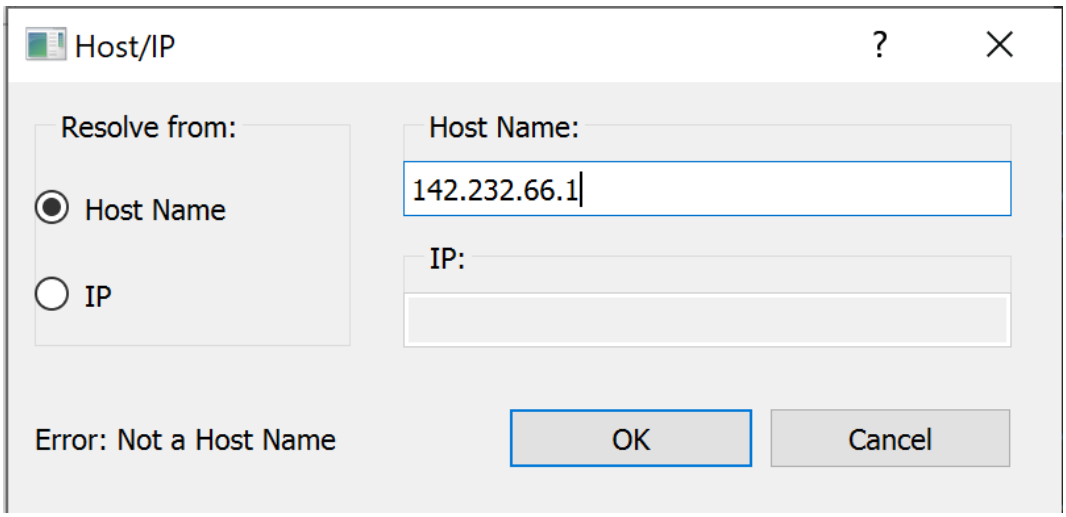
- 

2. Empty Host Name Box:

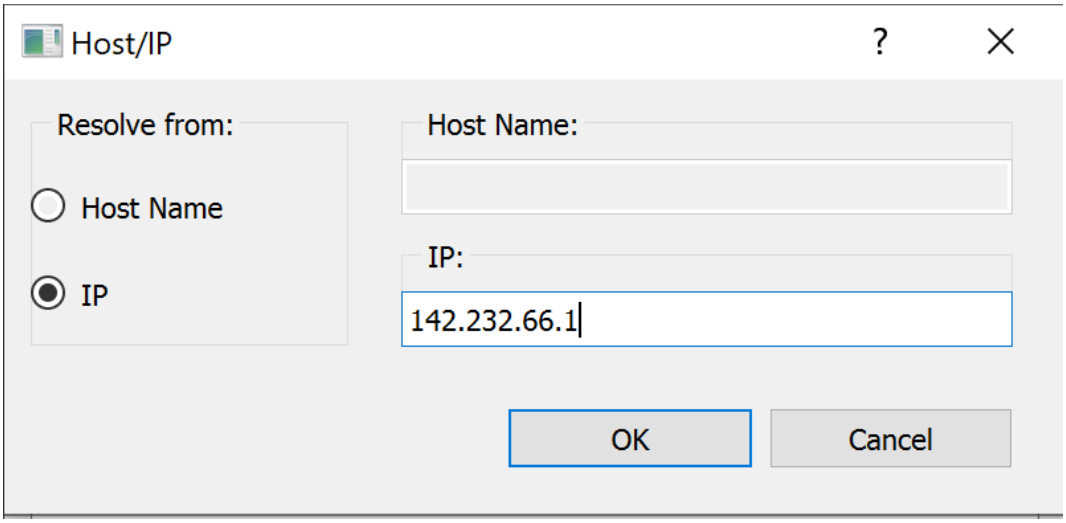
- 

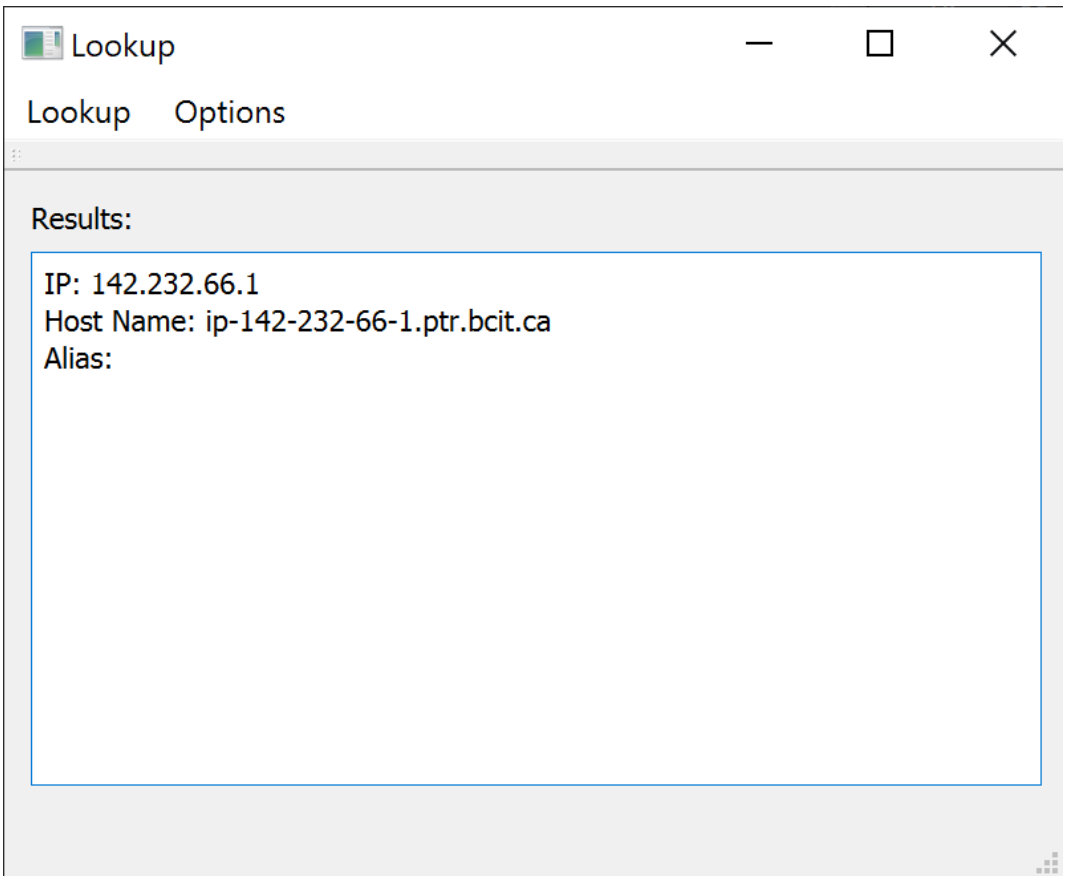
3. Host Name not found

- 

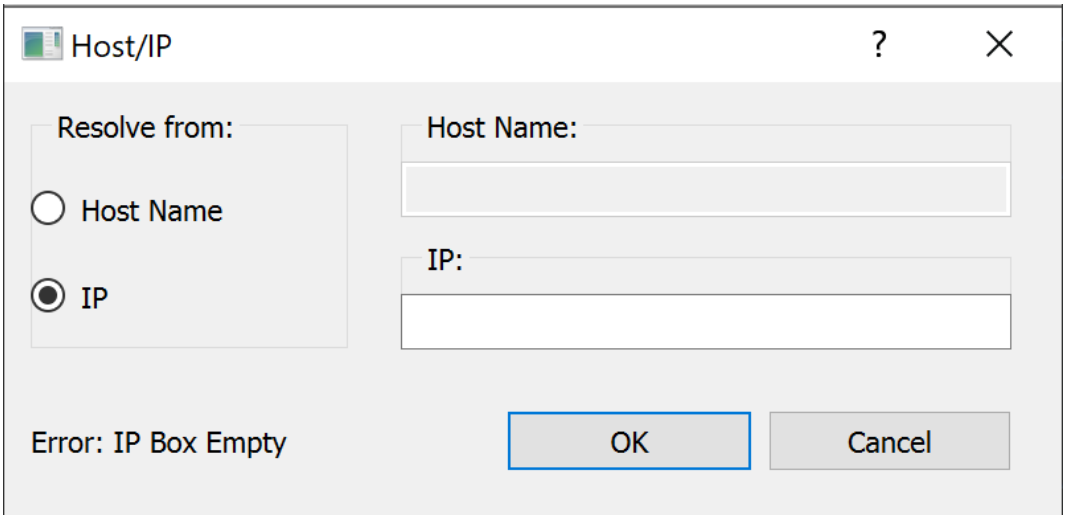
4. IP used in the Host Name Box
    - 

IP

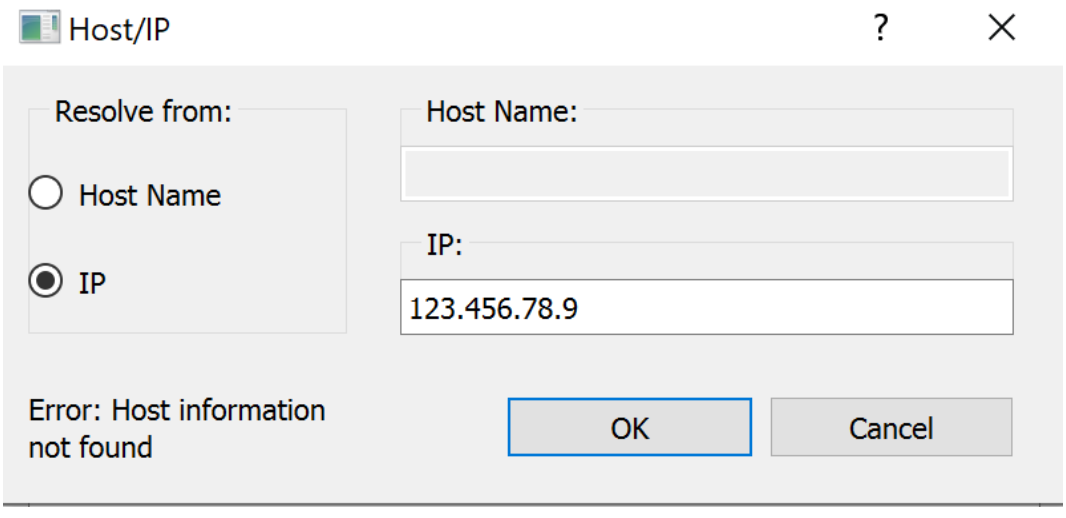
1. IP Found
    - 



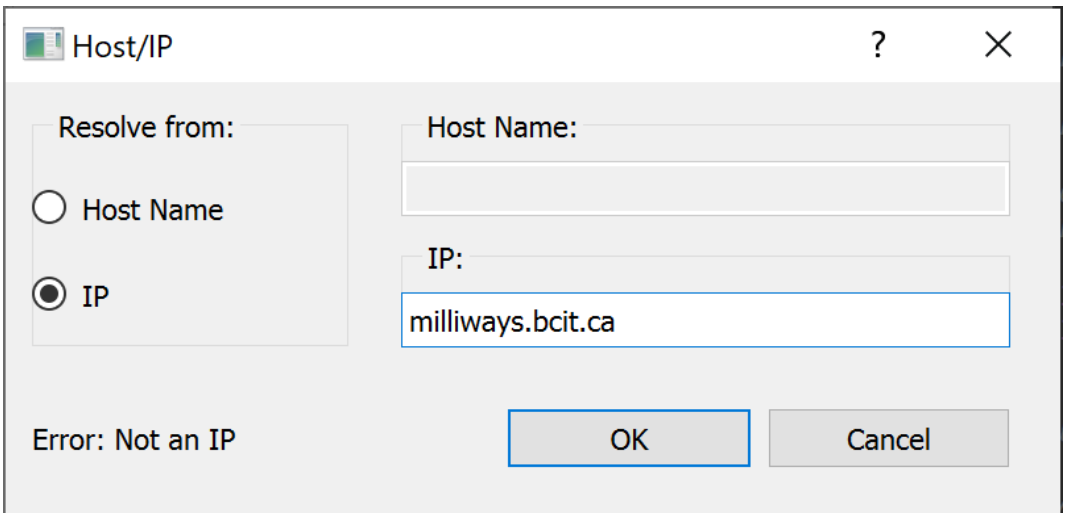
2. Empty IP Box



3. IP Not Found



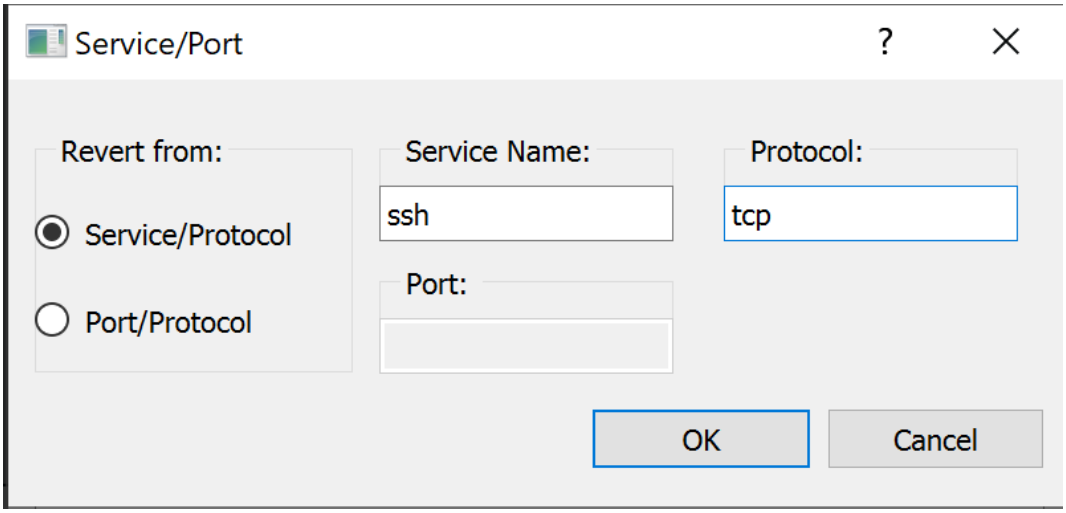
4. Host Name used in the IP Box

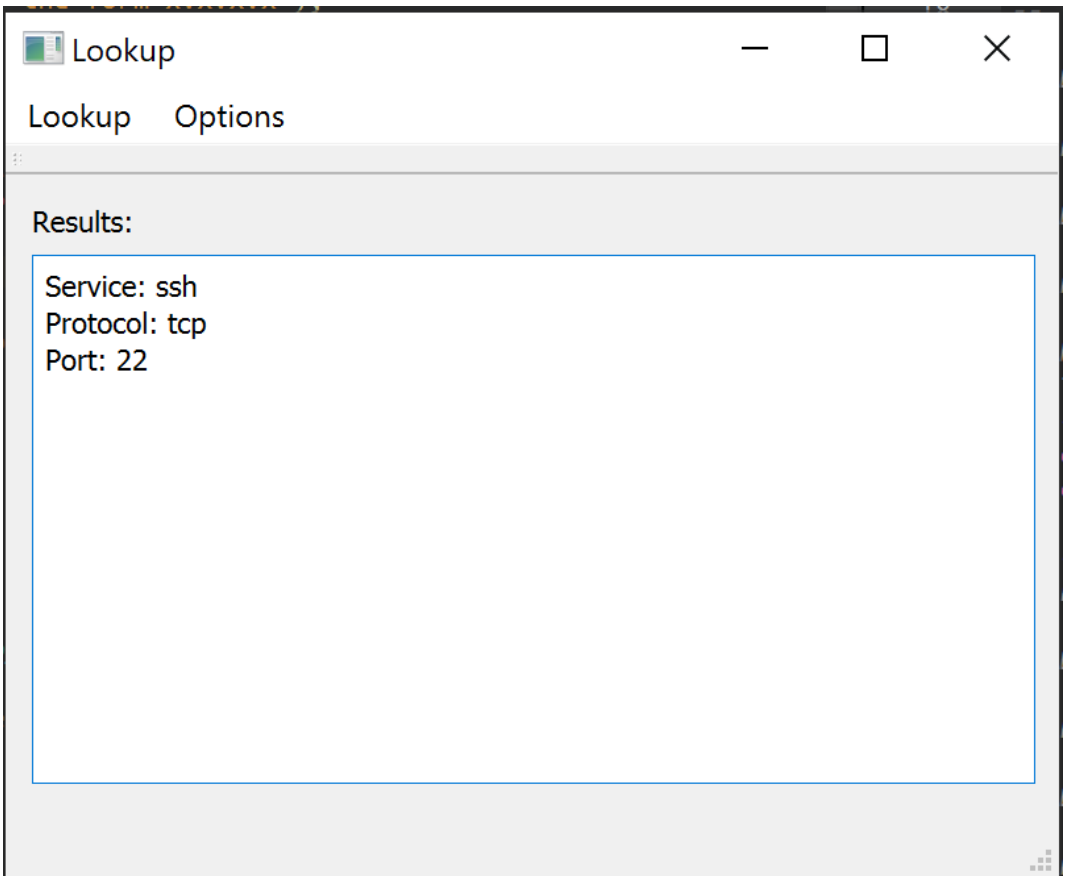


Service/Port

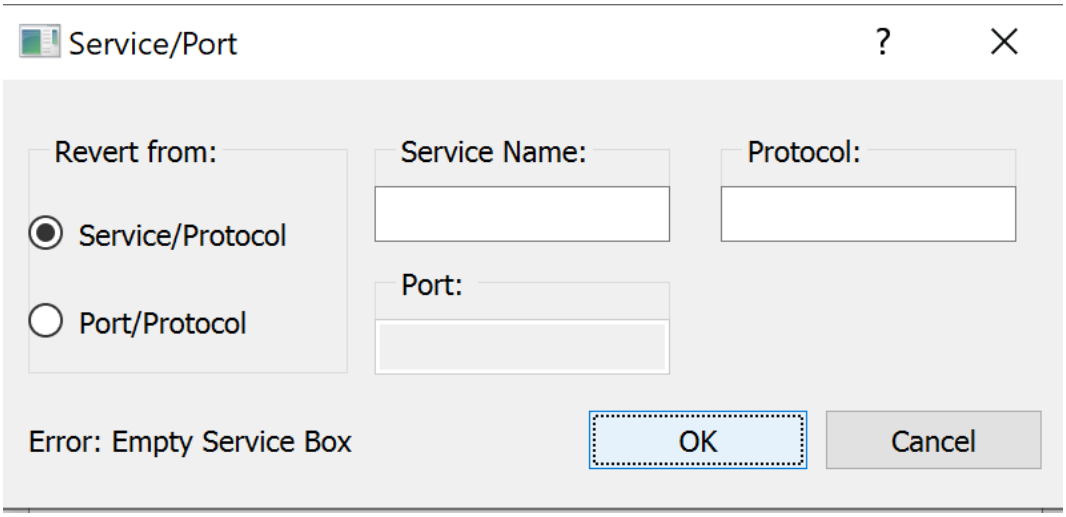
Service

1. Service Found

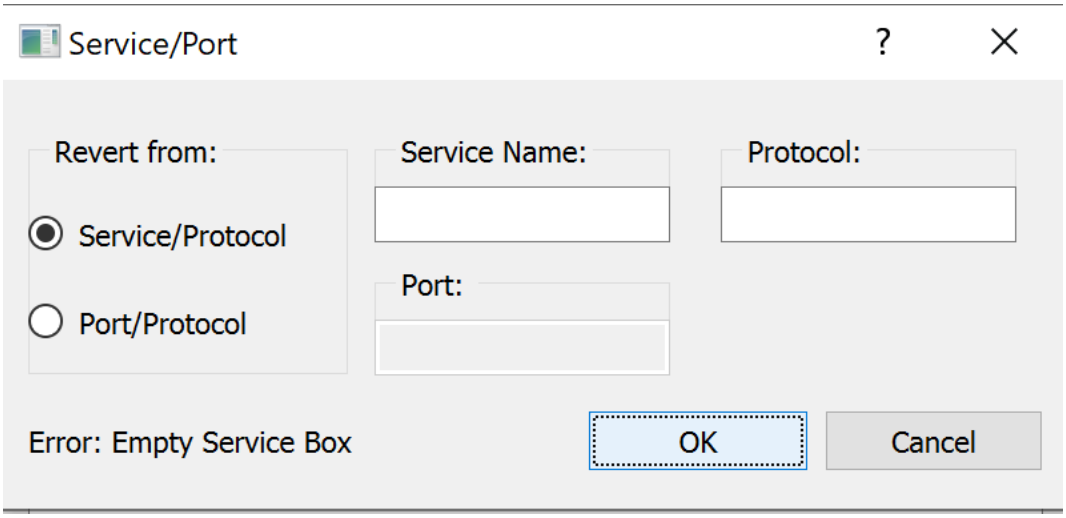


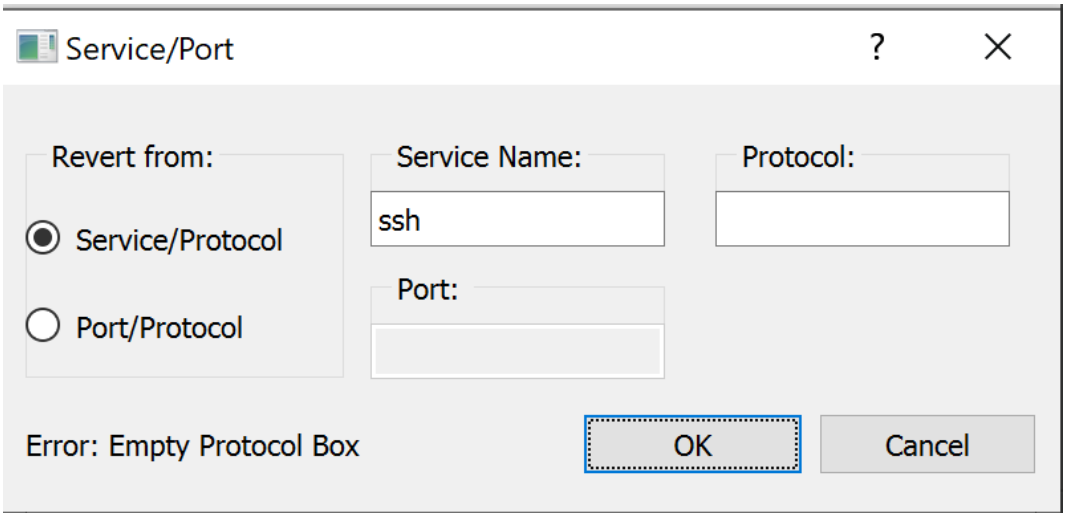


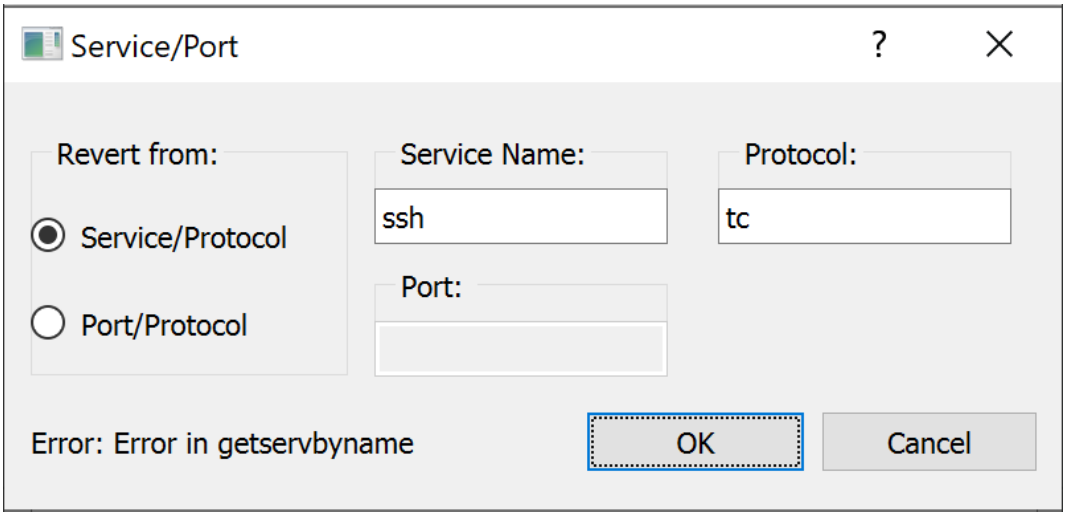
2. Service and Protocol Boxes Empty



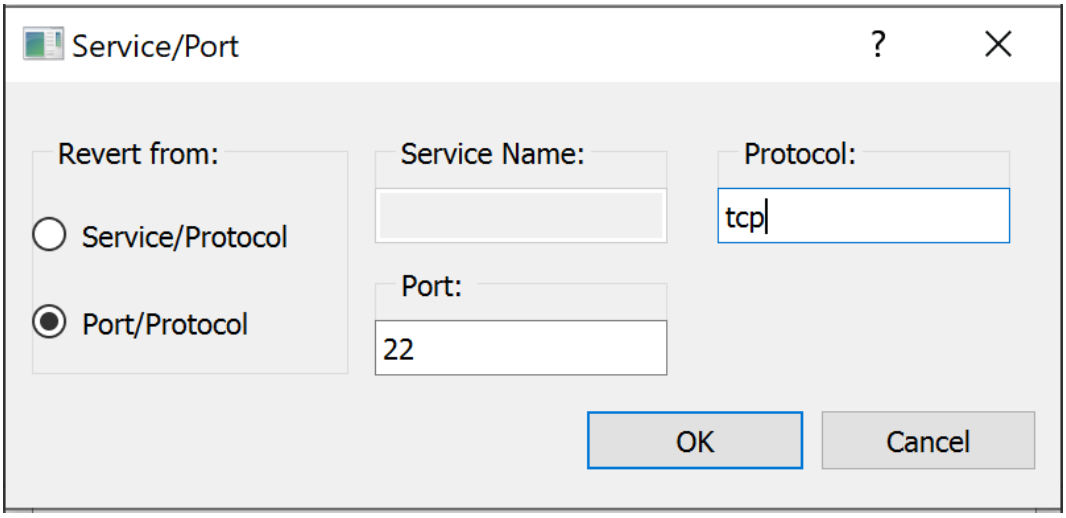
3. Service Box Empty

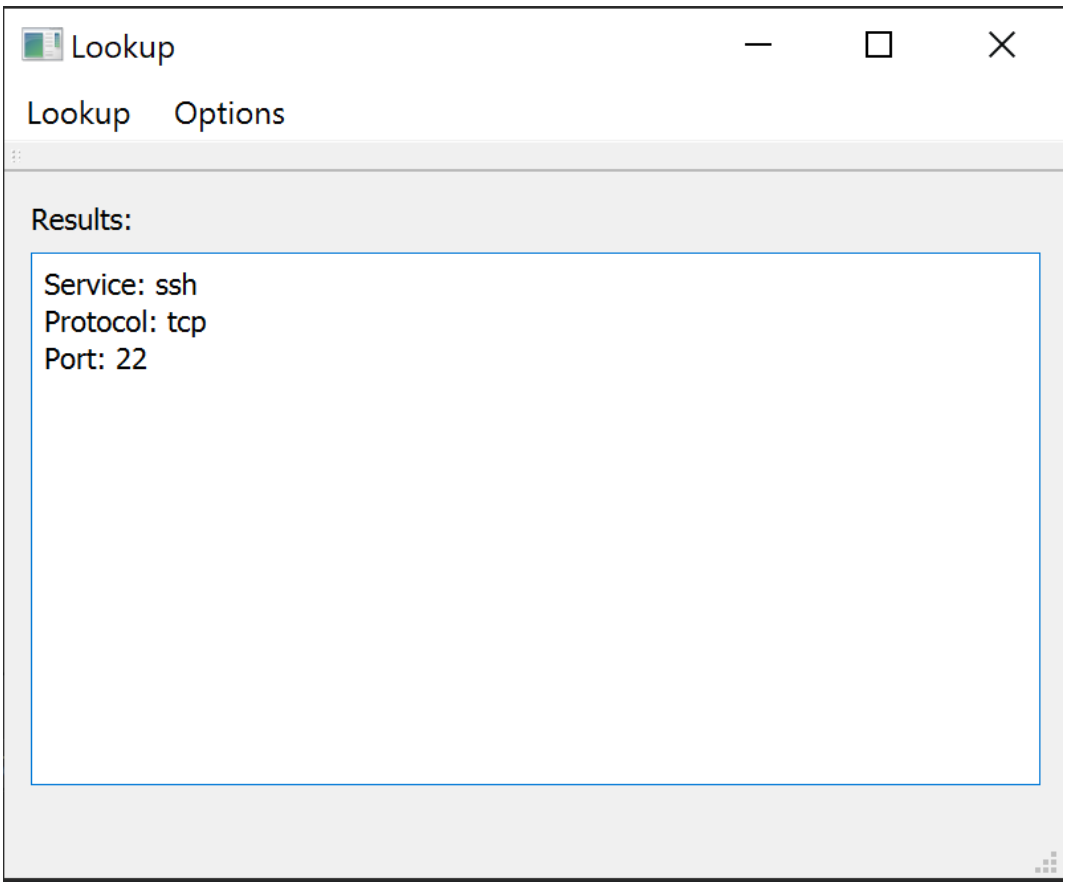


4. Protocol Box Empty
    - 

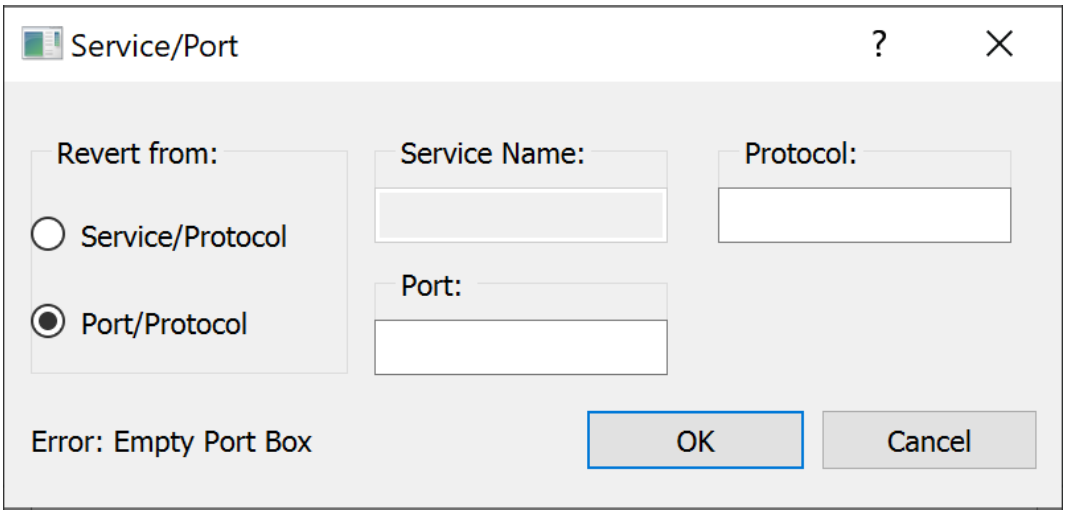
5. Service Not Found
    - 

Port

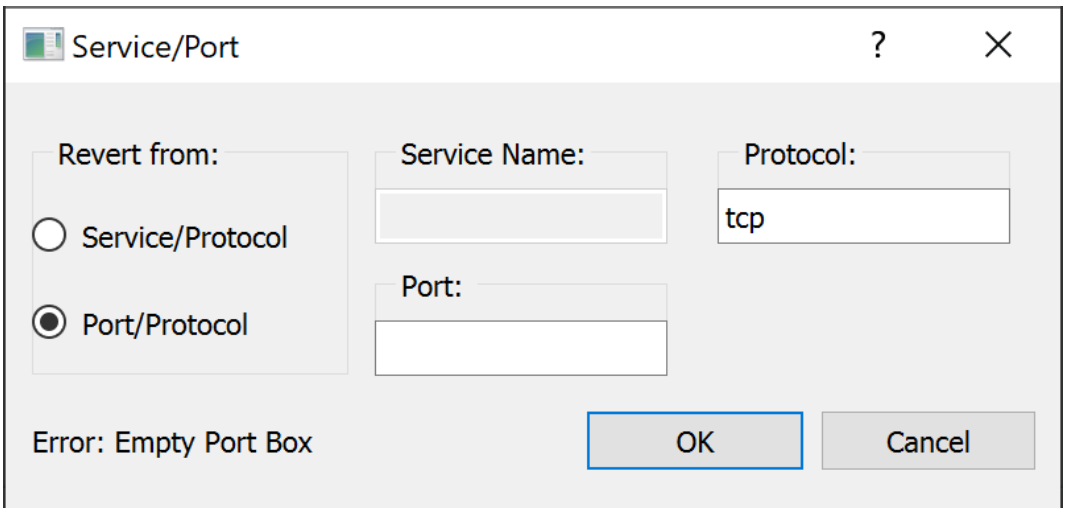
1. Port Found
    - 

- 

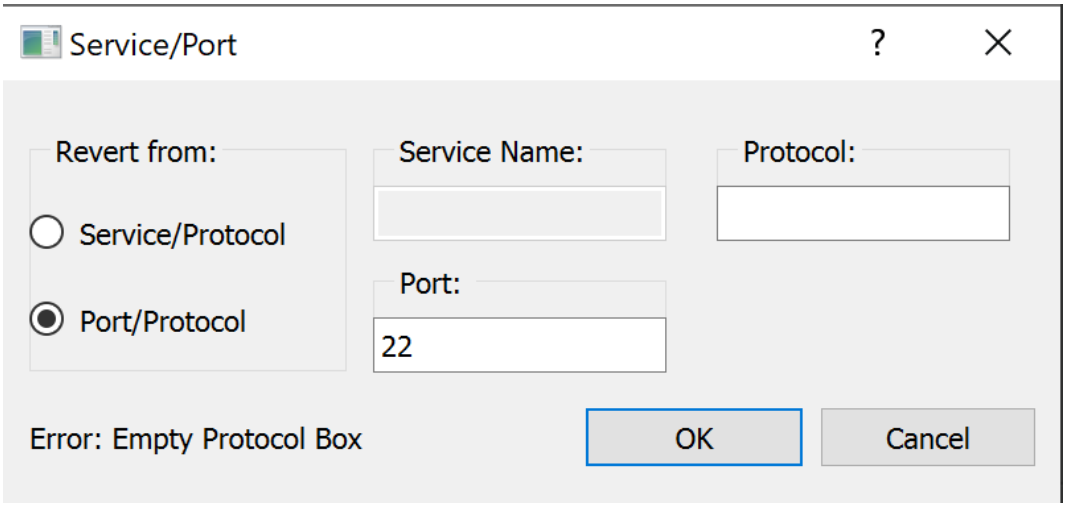
2. Port and Protocol Boxes Empty

- 

3. Port Box Empty

- 

4. Protocol Box Empty



5. Port Not Found