**Requirements Document: Subnet Discovery Tool v5.0**

**1.0 Core Objective**

The program, **DiscoverSubnet**, is a network discovery tool designed to identify specific MediaLinks hardware devices on a network. It will scan user-defined IP address ranges, identify devices using SNMP, and produce a summary report in both a GUI and a semicolon-delimited output file (either .txt or .csv).

**2.0 Technical & Environmental Requirements**

* **2.1 Platform:** The program must be a PowerShell script that runs correctly on **PowerShell 5.1**.
* **2.2 Portability:** The script must be self-contained and compilable into a standalone .exe file.
* **2.3 File System Awareness:** The program must automatically determine the directory it is being run from. All output files (logs, discovered device lists, settings) must be saved to this directory.
* **2.**4 The program file name will be DiscoverSubnet.ps1.
* **2.**5 A User configuration Persistence File will be created with the program defaults the first time the program is run and saved in the install folder. It will be read each time the program is ran and will save the user settings each time the user selects the button to scan the network.
* **2.**6 The file will contain a version number that will start with 2.0 and increase with each update and patch.
* **2.**7 The file name and version number will be shown in the GUIs and log file.
* **2.**8 The file will have a header similar to the following:

#requires -Version 5.1

########################################################################

##### DiscoverSubnet.ps1 Version 4.3 (Production). Copyright MediaLinks, Inc. 2025

##### Author: Gary Faubert

##### Last Modified: September 24, 2025

#####

##### Version 4.3 Changes:

##### - Fixed all syntax and architectural errors related to the background job creation.

##### - Implemented a stable, non-blocking message loop for a fully responsive GUI.

##### - All previous features and bug fixes are retained in this final stable version.

#####

##### Note: this PowerShell script can be compiled using the Visual Studio Code

##### command Win-PS2EXE

##### Note: In Win-PS2EXE select the source and uncheck all boxes.

########################################################################

**3.0 User Interface (UI)**

The program will feature two windows: a **Configuration Window** and a **Progress** & **Log Window**.

**3.1 Configuration Window**

This window is presented to the user on startup. It must contain the following configurable fields:

* **3.1.1 IP Address Ranges:** A text box for entering comma-separated IP ranges.
* **3.1.2 SNMP Community String:** A text box for the community string, defaulting to medialinks.
* **3.1.3 Ping/SNMP Retries:** A dropdown list allowing the user to select 0, 1, 2, or 3 retries, defaulting to 0.
* **3.1.4 Output File Name:** A text box for the base name of the output file, defaulting to DiscoveredDevices.
* **3.1.5 Output File Type:** A dropdown list for selecting the output file extension, with options for txt and csv.
* **3.1.6** Save **SNMP-Unresponsive Devices:** A checkbox, defaulting to unchecked. When checked, devices that respond to a ping but fail to provide SNMP data will be included in the final output file.
* **3.1.7 Max Parallel Scans:** A numeric up/down control allowing the user to set the number of concurrent scan threads, ranging from 1 to 100 and defaulting to 20.
* **3.1.8 Diagnostic Level:** A dropdown list with options Off, Standard, and Verbose to control the level of diagnostic detail written to the text log file (not the GUI).
* **3.1.9 "Start Discovery" Button:** A primary button to begin the scan. Clicking this button closes the Configuration Window and opens the Progress & Log Window.
* **3.1.10 User configuration Persistence File:** Whenever the program is run it will check for the existence of the configuration file in the install folder. If it is mussing it will be created with the defaults for each field. The program settings GUI will read the file and populate the fields with the last used settings. The user must be able to change any of the settings in the GUI. The file will be updated with the current settings when the user selects the discovery button on the GIU to start the network scan.

**3.2 Progress & Log Window**

This window appears after "Start Discovery" is clicked and remains for the duration of the scan.

* **3.2.1 Log Display:** A read-only text area that displays real-time status messages and results.
* **3.2.2 Progress Bar:** A progress bar that shows the overall completion percentage of the entire scan (all ranges combined).
* **3.2.3 "Save Results" Button:** Initially disabled. It becomes enabled only after the entire scan is complete.
* **3.2.4 "Cancel Scan" Button:** Initially enabled. Clicking it must immediately stop the ongoing scan.
* **3.2.5 "Close" Button:** Initially disabled. It becomes enabled after the scan is complete or has been cancelled.
* **3.2.6 GUI Responsiveness:** The Progress & Log Window **must remain fully responsive** at all times during the scan. The user must be able to move the window and click the "Cancel Scan" button instantly. This requires the main discovery work to be performed in a background job or thread.

**4.0 Program Flow & Logic**

* **4.1 Startup:**
  + The program loads settings from DiscoverSubnet.settings.json (if it exists).
  + The Configuration Window is displayed, populated with the loaded settings or the defaults if no settings file is found.
* **4.2 Validation:**
  + Users must be able to enter text in the text fields and select options from the drop menus.
  + One or more ranges can be entered separated by “,”.
  + Ranges can be in the format of 4 fields separated by “.”.
  + The first 3 fields must be an integer between 1 and 254.
  + The fourth filed can be a 0 or a range from 2 to 254 separated by “-“.
  + If the fourth field is “0” it must be interpreted as “2-254”.
  + Any white space or non-numeric entries will be stripped.
  + Any violation of a legal IP address range will result in an error asking the user to correct the entry.
  + Community strings must be 1 to 32 characters in length and can consist of Uppercase and lowercase letters (a-z, A-Z), Numbers (0-9), and Special characters (e.g., @, #, $, %, &, \*) otherwise post an error asking the user to correct the entry.
  + Report file names must meet valid windows standards otherwise post an error asking the user to correct the entry.
* **4.3 Scan Initiation:**
  + Upon clicking "Start Discovery," the current settings from the GUI are saved to DiscoverSubnet.settings.json in the install folder.
  + The Progress & Log Window is displayed.
  + The main discovery process is launched in a **background job** to ensure the GUI remains responsive.
* **4.4 Scan Execution:**
  + The program will parse the user's input into a list of IP ranges.
  + It will process each range **sequentially, one at a time**.
  + Within a single range, the individual IP addresses will be scanned **in parallel**, using the number of threads specified by "Max Parallel Scans."
* **4.5 Reporting Flow:**
  + Before scanning a range, a "Starting scan for range..." message is displayed in the GUI.
  + After a range is complete, a "Scan complete for range..." message is displayed, followed by a summary of the devices found within that specific range.
  + This process repeats for all specified ranges.
  + After all ranges have been scanned, a final "Overall Discovery Complete" message is displayed summarizing the devices discovered.

**5.0 Device Discovery & Identification Logic**

The process for identifying a device at a single IP address is as follows:

* **5.1 Ping:** The IP address is first checked for reachability using a ping. The number of retries is based on the user's selection.
* **5.2 SNMP Query:**
  + If the ping is successful, the script performs SNMP GET requests for the following OIDs:
    - 1.3.6.1.2.1.1.2.0 (sysObjectID)
    - 1.3.6.1.2.1.1.5.0 (sysName)
    - 1.3.6.1.2.1.1.6.0 (sysLocation)
  + If any of these SNMP requests fail, the device is classified as **SNMP-unresponsive**. The key for this device will be [No Name Found];[No Location Found];UNKNOWN.
* **5.3 Device Type Refinement:** If all initial SNMP requests succeed, the type is determined based on the sysObjectID value:
  + **5.3.1 General Mapping:**
    - 1.3.6.1.4.1.17186.1.10 -> MD8000
    - 1.3.6.1.4.1.21839.1.2.17 -> MDX2040
    - 1.3.6.1.4.1.17186.1.24 -> MDP3020
    - 1.3.6.1.4.1.17186.3.1.1.1.0 -> MDX-32C/48X6C
    - Any other OID -> UNKNOWN
  + **5.3.2 MDX-Series Refinement:** If the base type is MDX-32C/48X6C, the sysName is checked.
    - If the name contains "32C", the final type is **MDX-32C**.
    - If the name contains "48X", the final type is **MDX-48X6C**.
  + **5.3.3 MD8000-Series Refinement:** If the base type is MD8000, a follow-up SNMP GET is performed on OID 1.3.6.1.4.1.17186.1.10.1.1.3.0.
    - If the value is 1, the type is **MD8000EX**.
    - If the value is 2, the type is **MD8000SX**.
    - Otherwise, the type remains **MD8000**.

**6.0 Device Multiple IP Addresses Detection**

* **6.1 Device types, such as the MD8000 family have more than one IP Addresses**
  + When an IP Address successfully pings and the device Name and Location can be determined, check the list of devices that have already been detected from any IP range and if the Name and Location match, append the newly discovered IP address to the already discovered addresses for that device.
  + Multiple addresses for a device must be separated by a comma,

**7.0 Output and Reporting Formats**

**6.1 GUI Reporting Format**

* For each range, discovered devices must be displayed in the following format: Name=CNN Location=ANHMCARDNA2 Type=MD8000EX Address=192.168.12.170, 192.168.12.171
* SNMP-unresponsive devices for the range should be summarized on a single line: SNMP Unresponsive: 192.168.12.21, 192.168.12.22

**6.2 File Output (DiscoveredDevices.txt or .csv)**

* The file must start with the header line: Name;Location;Type;IPs
* The file will contain all devices that were successfully identified via SNMP (including those with an "UNKNOWN" type from an unrecognized OID).
* If the "Save SNMP-unresponsive devices" checkbox is checked, the [No Name Found];[No Location Found];UNKNOWN entries will also be appended to the file.

**7.0 Logging**

* A new timestamped log file (e.g., DiscoveryLog\_YYYY-MM-DD\_HH-mm-ss.log) is created for each run.
* The first line of the log file must be the program's version number.
* All messages displayed in the GUI are also written to the text log file.
* Diagnostic messages (Standard or Verbose) are written **only** to the text log file, as controlled by the GUI