How to use MT-Server Toolbox :

Have you ever dreamed of being able to control your own remote server from a simple and ergonomic tool ?
If so, this tool is for you.
However, it requires a minimum of coding knowledge to be able to use it.
This tool is still in development and this guide is succinct.
Do not hesitate to share your questions, problems or improvements on Github.

Summary :
- Presentation
- Installation
- Game Server
- Usage

Presentation :

For a long time, I was looking for a simple, modifiable and free tool to start, stop, send commands, chat with Discord, etc... on a remote or local machine.
Having found nothing that suited me and being a developer in my spare time, I decided to start making my own and share it with you.

Not wanting Linux, and for maximum compatibility, everything is done in Powershell code with a few plugins.

This tool was developed for Windows and on Windows.
Note: Windows can sometimes detect the .exe or .ps1 as a threat. And the only threat is yourself.
I assure you that no Virus is present and no script will alter your System.

Throughout this Guide we will rely on my personal deployment of this tool.
Let's get started.

Usage environment :
- Client = Windows 10
- Server = Windows Server 2019

Installation :

Server Side :

1 : You will need the miniupnp tool to dynamically open ports in UPnP
You can find it in ‘Installation tools\upnpc-exe-win32-20220515.7z’ in the 'MT-Server ToolBox Server' folder.
Check if your router supports the UPnP IGD2 protocol and is able to receive UPnP port opening commands.
In my case I use a Freebox
Link to Author Website : http://miniupnp.free.fr/index_fr.html

Installing miniupnp:

1 : Copy the following files to the C:\Windows folder
- miniupnpc.dll
- miniupnpc.lib
- upnpc.exe

That's all.

Installing MT-Server ToolBox Server:

1 : Copy the ‘MT-Server ToolBox Server’ folder in C:\

2 : In the ‘MT-Server ToolBox Server\Installation tools' folder :
- Copy ‘MT-Server-Toolbox_Launcher.ps1’
- Place it in: C:\Windows\System32\GroupPolicy\Machine\Scripts\Startup

3 : To automatically launch the Toolbox Script when the server starts we need to specify it in Windows.

We need to go to gpedit.msc :
- Windows + R, type gpedit.msc
- Go to ‘Computer Configuration’
- Windows Settings
- Scripts (Startup/Shutdown)
- Startup
- Powershell Scripts
- Add the 'MT-Server-Toolbox_Launcher.ps1' that we just placed
- Apply everything and that's all.

Configuring the ServerConfig.json:

1 : Open the 'ServerConfig.json' in the 'MT-Server ToolBox Server' folder
- In "ip_Net" define the public IP of your router.
- In "tcpport" define the listening port of the ToolBox that you want to use.
- In "Discord" define the Webhook of your Discord Server if you want the Server to send information to Discord

Tuto : https://www.svix.com/resources/guides/how-to-make-webhook-discord/

Port forwarding:

To be able to start the server remotely we must redirect a router port.

To be able to do this, I'll let you discover how to do it with your equipment.
By default the WOL port on Windows is ‘9’ in UDP.

As destination IP it is preferable to define all the machines.
This can be done by putting: 192.168.1.255

Windows Firewall:

Previously we defined a listening port in ServerConfig.json.
We need to add it as an inbound rule in the Firewall.

Create a new TCP inbound rule with the port you defined in the **ServerConfig.json** and allow all connections on it.

That’s all for the Server Side.

To see if everything works, you can run 'Main.ps1'
Warning: Only one script can listen a TCP port. If you have errors that scroll continuously, check in the task manager that it is not already running.

So normally at each startup Windows will launch the script 'MT-Server-Toolbox_Launcher.ps1' in hidden mode which will launch the 'Main.ps1' which will ask to open in UPnP the port "tcpport" defined in the .json and will be ready for any action requested.
For everything else I'll let you look in the script and folder.
Comments are noted everywhere and don't hesitate to ask me questions.

Client Side :

For the client, place the 'MT-Server ToolBox' folder wherever you want.
The client-side script can work both on a local network and on an internet network.

Configuring the ServerConfig.json:

Open the 'ServerConfig.json' in the 'MT-Server ToolBox' folder and fill in the chosen parameters.
Everything is written inside to help you.

Warning:

- In "WOLport" define the same UDP port that we redirected in Router

- In "tcpport" define the same TCP port as the one we created in the Server Firewall

That’s all for the Client Side.

Now you can run the 'Main.ps1' or the 'MT-Server ToolBox.exe'

Game Server :

To define and add game servers.
You must correctly fill out the 'GameConfig.json' on the server side, as well as the "StartMethod" and "StopMethod" scripts.
The 'GameConfig.json' on the client side must be identical to that of the server to work except the "StartMethod" and "StopMethod" lines are not necessarily required on the client side.

Note: If you use the Discord Webhook you can send the Custom 'Game_Server_Config' command in the options section of the ToolBox to retrieve and copy its already filtered content.

I gave you three examples. I'll let you see and explore how they work.
You can add as many "GameServer..." as you want.They will be loaded automatically.

Usage :

The MT-Server Toolbox allows you to start a Local or Remote server, to be able to stop it, to be able to launch game server instances, to send custom commands, etc...

This tool is under MIT License, so you can use it, explore it and modify it according to your needs while retaining the name of the original author.
I strongly invite you to explore it to see how my functions work, I have written as many comments as possible inside to help you.

Settings Panel :

Local Mode :

Local mode allows you to define whether you use the Launcher on the same network as your server or not.

Translation :

A translation system is also available.
Just go to the 'resources\Stringtable.json' folder and create one in your langage.
Done like the other languages already listed.
Feel free to share them on Github.

In .json, don't forget the commas. It will save your life.

Custom command :

You can send custom TCP commands from the ToolBox tool panel

Here is the one already included and available:

- ‘Game_Server_Config’ sending the contents of the 'GameConfig.json' to the server side via the Discord Webhook.

- ‘Admin_Dev_Action’ opens FTP ports to send files to the server and ports to connect to it using Windows Remote Desktop.

Debug :

The server-side and client-side launcher will display everything it does during execution
in 'Main.ps1' on the Client side on line 11 you can deactivate Debug mode.
Warning: This will also activate the Login panel at launch.
The Password is set on line 55 to $LoginPassword.
By default is 'zz' .