
THINK BIOS CONFIGURATOR

Version 0.8

INTRODUCTION

The Think BIOS Configurator tool was developed using the WMI BIOS interface methods and the scripts located [at this website](#) to create a user friendly way of applying changes to the BIOS from within the Windows environment. This document will describe the ways the tool can be used and the various options it supports. The application needs no other files for execution. The application will read from WMI to provide the proper options for each of the available settings.

NOTE: Not all BIOS settings are exposed through the WMI interface. This tool will list only the settings that are configurable through WMI.

Standard execution of the application

This tool is in the HTML application format (hta) so it can be portable and be used in lightweight versions of Windows like WinPE without needing additional libraries to run. When launched by a user in Windows, the application will attempt to relaunch itself with administrative privileges which causes a User Account Control dialog to be displayed with the Program Name of "Microsoft (R) HTML application host" and the Verified Publisher as "Microsoft Windows". This occurs because access to the WMI BIOS Settings interface requires administrative privileges on the system and there is no "Run as Administrator" option for .HTA files. After approving the UAC prompt, the application window will appear. Below is an example of what a standard execution of the application will look like:

Think BIOS Configurator

File Actions:

Select a previously created .ini file of settings

Create an .ini file containing the current settings of the target machine in the working directory.

Security Actions:

☐ Supervisor password set on the target machine

☐ Use different credentials to connect to target machine

Targeting:

Accessing settings on **LOCALHOST (W550s)**

Setting	Value	Setting	Value
WakeOnLAN	ACOnly	EthernetLANOptionROM	Enable
IPv4NetworkStack	Enable	IPv6NetworkStack	Enable
UefiPxeBootPriority	IPv4First	USBBIOSupport	Enable
AlwaysOnUSB	Disable	USB30Mode	Automatic

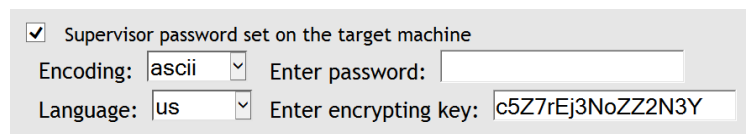
File Actions

At the top of the application screen is the File Actions section. On the left it is possible to browse to and select an .INI file that was previously generated by this program. Once selected, click **Apply Settings File** to make any changes detected in the .INI file. All the settings are read into the potential changes and then compared to the current settings of the machine. Only those that are different will be applied. A dialog box is displayed once the import is completed showing the settings that were successfully changed. This feature is also available via command line which is discussed in the Execution via Command Line section.

On the right, an .INI file of the current settings can be generated simply by clicking **Export Settings**. The file will be generated in the working directory the application is launched from. The file will be named **<model>Config.ini** where **<model>** is the text inside the parenthesis above the Settings Table. The settings that are exported are the current settings of the machine AND the pending changes to the machine. This is done to allow the user to create these files without actually having to commit changes to the machine. The created file has an optional first line if there is a supervisor password and the remaining lines are of the form **key,value**. If there is a supervisor password on the machine the application will include it based on the details provided in the Security Actions sections below.

Security Actions

If settings are being changed on a system where a BIOS Supervisor Password has been set, the Security Actions section provides the ability to provide the necessary password details. Select the check box next to **Supervisor password set on the target machine** to expand the user interface to show input fields for the password details as shown below.



☒ Supervisor password set on the target machine

Encoding: Enter password:

Language: Enter encrypting key:

Encoding can be either "ascii" or "scancode" and language is either "us", "fr", "gr". In most cases, the user should only have to supply the password and leave the other fields as they are. The password box is for the Supervisor password on the targeted machine. The encrypting key box is only used if the Export Settings function will be used to capture the settings of this targeted machine into an .INI file. A random 16 character string is provided or the user can create their own personal encryption key using characters from this set: a-z, A-Z, 0-9. Do not use any punctuation or special characters. This will allow the Supervisor password to be encrypted using [xxTEA](#) in the .INI file which is simply plain text. The encrypting key would be used later when the tool is used to apply the settings in the .INI file.

On the right side of the Security Actions section the user can select the check box next to **Use different credentials to connect to target machine** to expand the user interface to show additional input fields. When connecting to a remote machine over the network, a username and password for a local administrator account of the targeted machine must be supplied. On a domain network, the username may need to be in the format of "<domain>\<username>".

Note: the user CANNOT use different credentials when targeting the local machine.

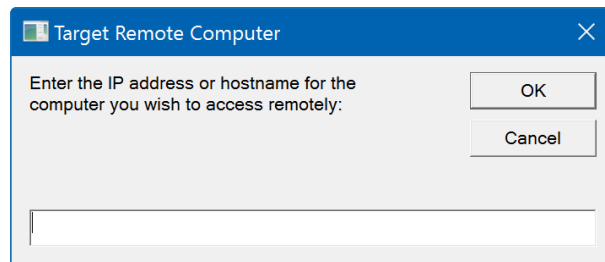
Targeting

By default the application will attempt to load the settings of the local machine when it launches. The address and model of the machine whose settings are being displayed will be displayed above the Settings table, such as:

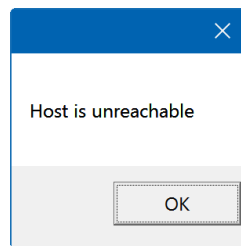
Accessing settings on *LOCALHOST (W550s)*

Page | 4

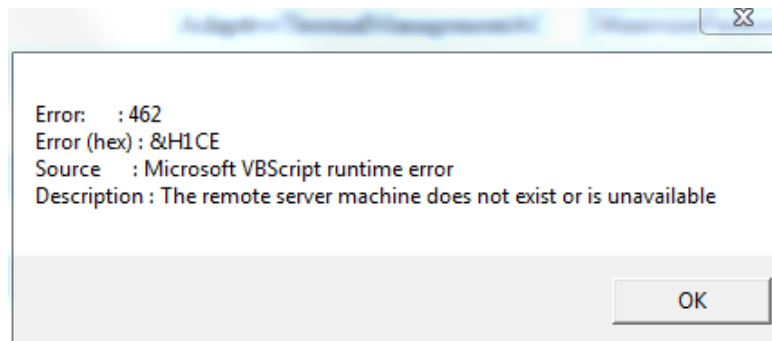
If you would like to target another remote machine on the network, click **Target Remote** and the dialog below will be displayed



If the host is unreachable, the application will gracefully display that information.



If the WMI service is unavailable on the target machine or the application is denied access another message box will come up and provide an error message similar to below. Once this dialog is acknowledged, the application will refresh with the information on the local machine.



Once connected to the remote computer, the screen will refresh with the data that is on that machine. To switch back to the local machine, just click the 'Target Local' button, which is only enabled if the application is accessing a remote machine.

Settings Table

Accessing settings on **LOCALHOST (W550s)**

Save Changed Settings

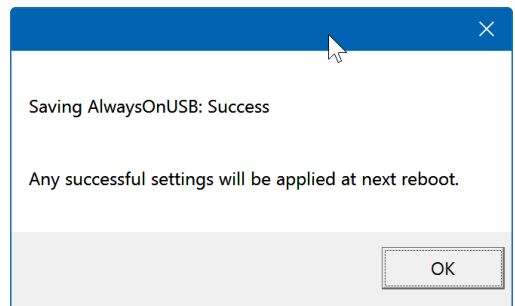
Restore BIOS Defaults

Reset to Current Settings

Setting	Value	Setting	Value
WakeOnLAN	ACOnly	EthernetLANOptionROM	Enable
IPv4NetworkStack	Enable	IPv6NetworkStack	Enable
UefiPxeBootPriority	IPv4First	USBBIOSupport	Enable
AlwaysOnUSB	Enable	USB30Mode	Automatic
TrackPoint	Automatic	TouchPad	Automatic
FnCtrlKeySwap	Disable	FnSticky	Disable
FnKeyAsPrimary	Disable	BootDisplayDevice	ICD

The Settings Table displays the current settings of the target machine. As seen above, each setting has an associated value column which can be changed easily. Most settings are in the form of dropdown boxes but options like ‘BootOrder’ have more complex options to select the new value of the setting. If the user makes any changes, the text of the setting name will be changed to red (as seen above with the AlwaysOnUSB setting) to signify that it is a value different than what is currently set on the computer.

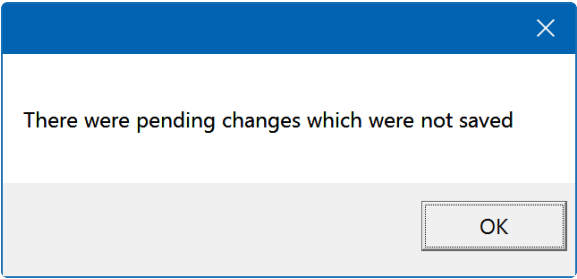
Once all the required changes have been set, click **Save Changed Settings** and the application attempts to commit the changes to the machine. Only the settings that are pending change are attempted to be saved to save time and eliminate any issues with trying to commit a setting will the same value. This is what will be displayed after the button is clicked:



As seen above, the application was successful in setting one setting. Error messages are provided in the case of a failure. More than likely, the error users will see will be ‘Access Denied’. **NOTE: All settings will only take affect once the computer is restarted.**

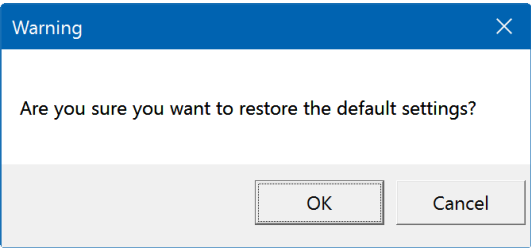
If settings have been changed in the application but not applied to the machine and the user wants to revert back to the original settings, click **Reset to Current Settings**. The user will see the items that were previously marked in red text will be reverted back to black text and the values reset to their original settings. This is a quick way for the user to create multiple configuration files and revert back to the settings they had already saved.

If a user closes the application with changes pending, a message will be displayed that there were pending changes. Unfortunately, there is a bug in Microsoft's code that makes it so the closing of the application cannot be stopped once it is started.



As seen above, the application was successful in setting one setting. Error messages are provided in the case of a failure. More than likely, the error users will see will be 'Access Denied'. **All settings will only take affect once the computer is restarted.**

The **Restore BIOS Defaults** button applies factory default values to the settings. NOTE: Not all BIOS settings are affected by the BIOS defaults change and varies by product. If the button is clicked, a warning prompt will be shown to confirm the user wishes to do that.



T

Execution via Command Line

This was a very important feature when developing this application. By having command line execution of the application, the user can use this in deployment situations. When used in this fashion, the application will flash briefly and suppress all notifications. All command line parameters must be enclosed in quotation marks and have the following format: **"switch=key,value"**. There are no spaces around the equals sign. There are several switches that are supported. Order does not matter but parameters are processed from left to right. Therefore if for some reason the user provides any switch more than once the last one will be the one the application uses. If the user passes both the 'config' and 'file' switch in one command, the 'file' switch will take precedence over the 'config'.

Page | 7

File switch

To apply a configuration file via the command line, the user would want to use the 'file' switch. The configuration file can be described either by a full path or just the name if it is located in the same folder as the application itself. Just like the 'Apply config file', all the settings will be processed and applied if they are different than the current settings on the machine.

```
Eg. ThinkBiosConfig.hta "file=C:\W550sConfig.ini"
```

Config switch

To apply a single configuration change, the user would want to use the "config" switch. This switch is mainly used for when the user wants to change just one setting and using a configuration file would be too much overhead.

```
Eg. ThinkBiosConfig.hta "config=USB30Mode,Disable"
```

Pass switch

This switch is used in conjunction with both 'file' and 'config'. If there is a supervisor password on the target machine, the application needs this information to apply the settings.

In the case of the 'file' switch, the application will be expecting the encryption key that was used when creating the configuration file if the password was encrypted and stored in the file or the supervisor password itself. If there is an encrypted string at the beginning of the file, the application will attempt to decrypt the first line of the configuration file to retrieve the supervisor password that is stored in the file.

```
Eg. ThinkBiosConfig.hta "file=C:\W550sConfig.ini" "pass=myEncryptionKey"
```

In the case of the 'config' switch, the application will be expecting the supervisor password itself. This will be passed to the application to save the setting properly.

```
Eg. ThinkBiosConfig.hta "config=USB30Mode,Disable" "pass=mySupervisorPassword"
```

Also if working with a different language than 'US' the user can enter their language along with the password in the following format (example demonstrates French password):

```
Eg. ThinkBiosConfig.hta "config=USB30Mode,Disable" "pass=mySupervisorPassword,ascii,fr"
```

Remote switch

The remote feature allows the user to set the target computer before the application displays the first settings. This can be used in two ways: set the target computer and open the GUI or set the target computer and apply a command line switch to it. To set the target computer for the GUI, only include the 'remote' switch.

Eg. `ThinkBiosConfig.hta "remote=<IP_or_Hostname>"`

To apply a command line switch to the remote machine, just use the previously described switches. Remember order of the switches does not matter.

Eg. `ThinkBiosConfig.hta "remote=<IP_or_Hostname>" "file=C:\W550sConfig.ini"`

Help switch

Should show a helpful dialog but isn't implemented yet.

Page | 8

Eg. `ThinkBiosConfig.hta "help"`

Deployment Situations

This application can be used in a deployment environment like MDT or SCCM provided support for HTA/HTML is enabled (MDT boot images provide this by default, SCCM needs to have the feature enabled). The application must be called after the disk has been partitioned. In testing, placing the command during the Post-Install step seems like a good place. Also remember that a restart is required to apply the settings. Example code for a task sequence step:

```
cmd.exe /c PathToApp\ThinkBiosConfig.hta "config=BootOrder,HDD0:PCILAN"
```

Note: The 'cmd.exe /c' is needed 9 out of 10 times when running as a Local System account. In some cases it isn't necessary but for consistency it is recommended.

Troubleshooting

If a supervisor password exists and you know you are typing in the correct password but are receiving 'Access Denied' errors, restart the machine. You may have exceeded the allowed attempts.