Trusted Platform Module (TPM)

TPM 2.0 module with support for TPM 1.2 is included.

This is a security device designed to secure the system using integrated cryptographic keys.

Support and requirements:

- Only supported for OS installed as UEFI boot. Legacy boot is not supported.
- Secure Boot must be configured and enabled for full TPM functionality to be available.
- TPM 2.0 is not natively supported by Windows 7, but hotfix to add support for Bitlocker Drive Encryption with TPM 2.0 is available here:
  https://support.microsoft.com/en-us/help/2920188/update-to-add-support-for-tpm-2-0-in-windows-7-and-windows-server-2008
  Alternatively search for **KB2920188** to find the hotfix using your preferred search engine.

BIOS settings:

TPM settings can be managed in BIOS.

- Enter BIOS by pressing Del key during boot
- Go to Advanced tab and select Trusted Computing

Available options:

BIOS Setting	Options	Default	Description
Security Device Support	Disable/Enable	Enable	Enable/disable TPM Security Device
SHA-1 PCR Bank	Disabled/Enabled	Enabled	
SHA256 PCR Bank	Disabled/Enabled	Enabled	
Pending operation	None/TPM Clear	None	Option to schedule an operation. TPM Clear will clear all information stored on the Security Device
Platform Hierarchy	Disabled/Enabled	Enabled	
Storage Hierarchy	Disabled/Enabled	Enabled	
Endorsement Hierarchy	Disabled/Enabled	Enabled	
TPM2.0 UEFI Spec Version	TCG_1_2/TCG_2	TCG_2	**TCG_1_2:** Compatible mode for Win8/Win10 **TCG_2:** Support new TCG2 protocol and event format for Win10 or later
Physical Presence Spec Version	1.2/1.3	1.3	Select PPI Spec Version supported by OS.
Device Select	TPM 1.2/TPM 1.3/Auto	Auto	**TPM 1.2** will restrict support to 1.2 devices **TPM 2.0** will restrict support to 2.0 devices **Auto** will support both with default set to 2.0. TPM 1.2 devices will be enumerated-

To save any changes made and exit BIOS, press F4 button.
For detailed usage info please refer to official documentation for your selected operating system.

Secure Boot

Secure Boot is a security feature that can protect the system from running unauthorized boot loaders and avoid loading non-signed drivers during boot process. The feature is only supported for boot devices in UEFI mode.

CSM Support (BIOS Legacy boot and Option ROM support) is not compatible with this security feature and must be disabled in BIOS before enabling Secure Boot.

Basic Procedure:

-Set UEFI boot type in BIOS (set by default)
- Enter BIOS by pressing Del key during boot.
- Navigate to Boot tab in BIOS
- Make sure Boot mode select is set to: UEFI Boot Type
- Press F4 to save and Exit

- Disable CSM Support
- Enter BIOS by pressing Del key during boot.
- Go to Advanced tab and Select “CSM Configuration”.
- Set CSM Support to Disabled
- Press F4 to save and exit (unit must be restarted before continuing)

- Enable Secure Boot
- Enter BIOS by pressing Del key during boot.
- Go to Security tab and Select “Secure Boot”.
- Set Attempt Secure Boot to Enabled
- Secure Boot Mode is set to standard as default.
- Customized option will enable Key Management menu below with advanced settings.
- Press F4 to save and exit

Choose Boot Device

By factory default, press F7 to choose boot device at startup of unit.

Battery Status and Alarm

CMOS battery voltage is monitored during each boot and the following warning will show if it's below set low limit:

WARNING!!!
The CMOS battery capacity too low.

If this error is displayed, it is recommended to schedule service of the unit for replacement of CMOS battery.

When the CMOS battery is dead, or voltage is too low, it will not keep track of time/date when power is disconnected. BIOS settings will still be remembered, but bootup takes longer time and some BIOS related functions may not work optimally without battery.

The low limit is set to 2.4V by default. This is the minimum voltage required by main board for full functionality and to keep track of time and date while powered off.

The low voltage limit can be changed or disabled in BIOS by following the steps:

- Enter BIOS by pressing Del key during boot.
- Go to Advanced tab and select RTC Battery Alarm Configuration submenu
- Select option: RTC Battery Alarm Voltage
    - Default value is 2.4V
    - Limit can be set to any value between 2.0V and 3.0V (0,1V steps) or disabled to not show any warning.
    - Press F4 to save and exit

To check the current CMOS Battery voltage in BIOS go to: Advanced -> Hardware Monitor -> VBAT value
If the battery is dead or not connected the voltage will be reported as <1V.

Nominal CMOS battery voltage is between 3.0-3.2V when new and will decrease gradually over time.

Once voltage is below 2.4V it will no longer function properly, and there is risk of time/date loss.

The expected lifetime of CMOS battery is 5+ years. Depending on usage and storage conditions, the actual lifetime may vary between 3-10 years.

LAN Teaming

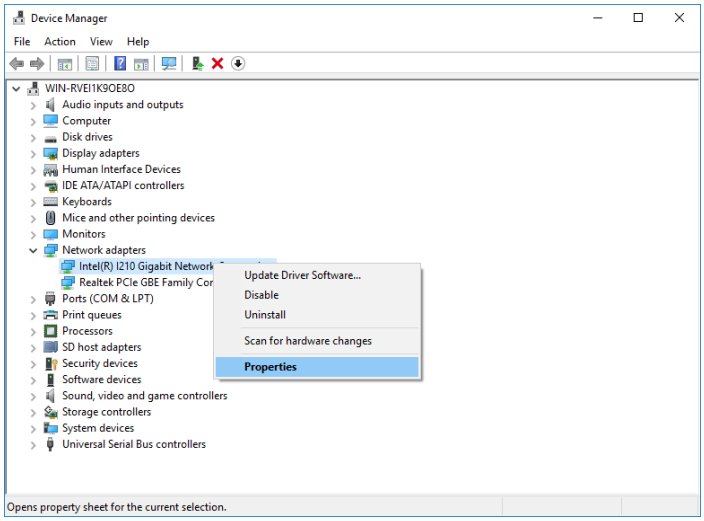
The network ports support VLAN and Teaming functionality.

For Windows 10, Intel Network Connections version 23.2.x or newer is required.
For Linux OS, adapter teaming is implemented using the native Linux Channel bonding module.

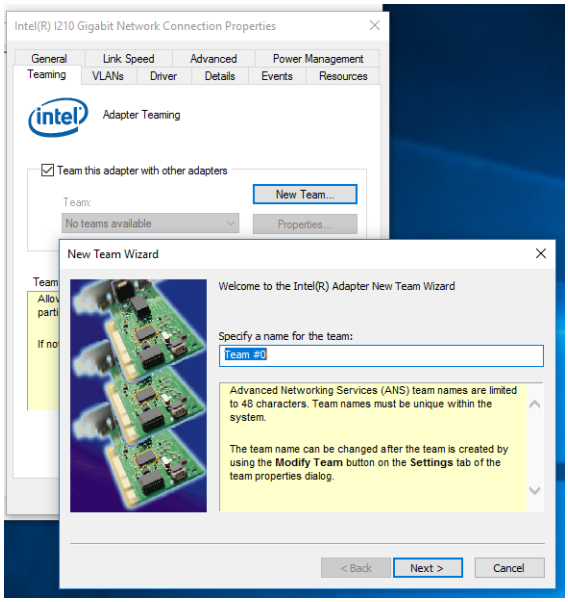
Latest Intel network driver/software is available at:
https://downloadcenter.intel.com/product/64402/Intel-Ethernet-Controller-I210-IT

Procedure to set up teaming in Windows 10:

- Open Device Manager and expand Network adapters
- Right-click on intel port and select properties.

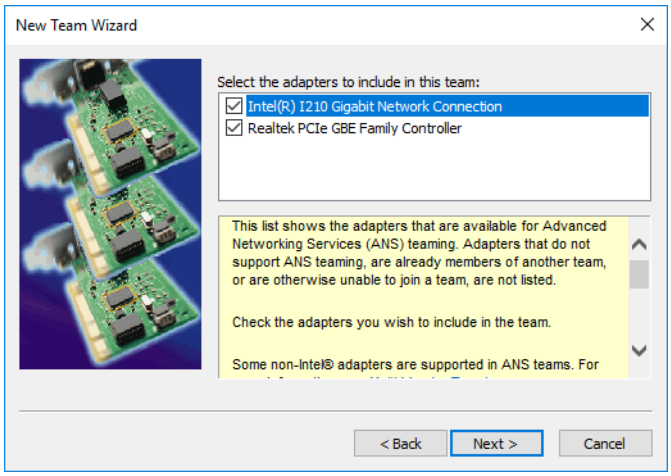


- Open the Teaming tab and check “Team this adapter with other adapters”
  - If Teaming tab is not shown, you need to update Intel network driver/software.
- Click on “New Team…” button, specify team name and press Next.

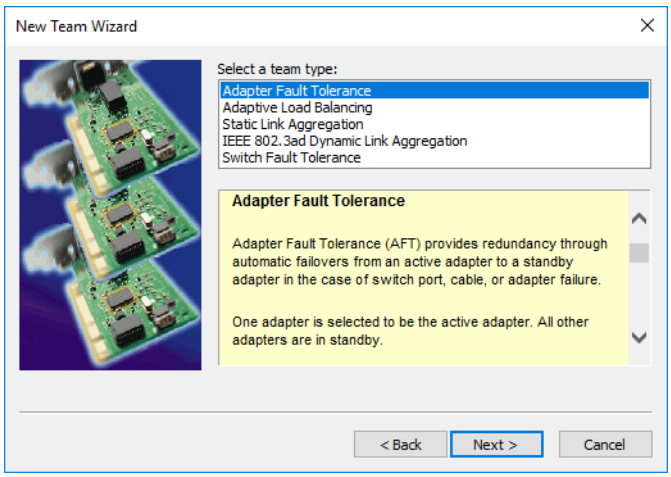


- Select the network ports to be added to the team then press Next.

Additional features overview/details



- Select team type from list, then press Next.



- Select Next/Finish to confirm and create team.
- If notification about jumbo packet limitation is shown, press OK.

Reading internal temperature

SCOM* command “TMP” can be used to read out internal temperature sensor value.

*If you are not familiar with the SCOM protocol, please review the Hatteland Technology’s Serial Remote Control Interface (SCOM) protocol document can be downloaded from:
https://www.hattelandtechnology.com/hubfs/pdfget/inb100018-7.htm

COM Ports Numbering

Default numbering on a factory standard model (example: Microsoft® Windows® 10) are as follow:

COM 1	= Internal Port: SCOM (COM)
COM 2	= Physical Port: RS-232/RS-485/RS-422, DB9M, non-isolated
COM 3	= Internal debug port on motherboard
COM 4-5	= Physical Port: CAN (via HD IO-Board)
COM 6-9	= Physical Port: NMEA (via HD IO-Board)

COM ports are not reserved nor locked by BIOS. COM Port numbering may differ depending on OS used and customized solutions. Use the Operating System functions to determine actual port numbering, if problems arise.

Using the eMMC as a backup/recovery device when SSD is used

Solution is pending implementation.

- Available for Microsoft® Windows® 10 only.
- eMMC will be used as boot and recovery device.
- The feature requires an SSD for the Operating System.

The backup is configurable, and can take periodic snapshots of a running system, and restore the unit to this state or factory settings. It will be possible to replace the SSD and restore the unit to the last backup with ease.

In the event of a drive failure or replaced SSD, the system will boot to recovery mode, where the system can be restored.

Install Operating System (OS)

For units without factory installed OS:
To install any OS on the unit, follow the instructions given by the manufacturer.

Install Microsoft® Windows® 10 with recovery option

1: Boot to Windows 10 installer
2: Select language settings
3: Click “Repair your computer”
4: Click “Troubleshoot”
5: Click “Command Line” and type the following indicated in a black box below.

type `diskpart`
type `list disk`
Take note of which disk number the SSD and eMMC is. This will usually be eMMC=0 and SSD=1

type `exit`

Step 1 - Prepare disks
==========
The contents of the script files will be listed on the end of this section (next page in this manual).
Refrence: "Diskpart_Pre-install.txt" and "Diskpart_Post-install.txt"

type `diskpart /s \Diskpart_Pre-install.txt`

Step 2 - Apply install image
==========
type `DISM /Apply-Image /ImageFile:\Image\W10LTSC2019.wim /Index:1 /ApplyDir:W:\`

Step 3 - Prepare boot sector
==========
type `cd "W:\Windows\System32"`

type `bcdboot W:\Windows /s S:`

Step 4 - Prepare recovery
==========
type `md R:\Recovery\WindowsRE`

type `xcopy /h W:\Windows\System32\Recovery\Winre.wim R:\Recovery\WindowsRE`

Step 5 - Finalize disk setup
==========
type `diskpart /s \Diskpart_Post-install.txt`

Step 6 - Finalize installation
==========
Reboot the computer and complete normal Windows setup
Activate Windows
Update Windows and install missing drivers
Set up Windows and software to default state

Step 7 - Set up backup
==========
Aquire Wimlib binaries from https://wimlib.net
Windows 64bit: https://wimlib.net/downloads/wimlib-1.13.0-windows-x86_64-bin.zip

Copy Wimlib files to D:\

Set up a scheduled script to run backup, or manually create snapshots.
See below reference for details.

Wimlib reference scripts:

```
REM ======================================================
REM ==== Initial backup example                       ====
REM ======================================================

REM ======================================================
REM ==== Get current time and date                    ====
REM ==== independent of locale                       ====
REM ======================================================

for /f "tokens=2 delims=.=" %%i in ('wmic os get localdatetime ^/value') do set tmp=%%i & set
td=%tmp:~0,4%-%tmp:~4,2%-%tmp:~6,2%_%tmp:~8,2%.%tmp:~10,2%.%tmp:~12,2%
echo %td%

REM ======================================================
REM ==== Capture initial complete base image          ====
REM ======================================================

D:\WimLib\wimcapture.cmd C:\ D:\Backup\Base.wim

**********************************************************************
**********************************************************************
REM ======================================================
REM ==== Differential backup example                  ====
REM ======================================================

REM ======================================================
REM ==== Backup changed data to separate file         ====
REM ======================================================

D:\WimLib\wimappend.cmd C:\ D:\Backup\%td%.wim --delta-from=D:\Backup\Base.wim
```

"DiskPart_Pre-install.txt" script contents

```
REM ====================================================
REM ===== DiskPart_Pre-install.txt                  =====
REM =====                                            =====
REM ===== These commands are used with DiskPart     =====
REM ===== to create four partitions for a           =====
REM ===== UEFI/GPT-based PC.                        =====
REM =====                                            =====
REM ===== Adjust the partition sizes to fill        =====
REM ===== the drive as necessary.                   =====
REM =====                                            =====
REM ===== This script assumes eMMC is disk 0        =====
REM ===== and SSD is disk 1                         =====
REM ====================================================

REM ===== 1 Prepare disk 0 (eMMC) ======================
select disk 0
clean
convert gpt

REM ===== 1.1 System partition ========================
create partition efi size=100
format quick fs=fat32 label=”System”
assign letter=”S”

REM ===== 1.2 Recovery tools partition ================
create partition primary size=1024
format quick fs=ntfs label=”Recovery tools”
assign letter=”R”
set id=”de94bba4-06d1-4d40-a16a-bfd50179d6ac”

REM ===== 1.3 Data partition ===========================
create partition primary
format quick fs=ntfs label=”Recovery data”
assign letter=”D”

REM ===== 2 Prepare disk 1 (SSD) =======================
select disk 1
clean
convert gpt

REM ===== 2.1 Microsoft Reserved (MSR) partition =======
create partition msr size=16

REM ===== 2.2 Windows partition =======================
create partition primary
format quick fs=ntfs label=”Windows”
assign letter=”W”
```

"DiskPart_Post-install.txt" script contents

```
REM ================================================
REM ===== DiskPart_Post-install.txt            =====
REM ===== Post install script                 =====
REM =====                                      =====
REM ===== This should be run after copying    =====
REM ===== WinRE to recovery partition         =====
REM ================================================

REM ================================================
REM ===== 1 Setting hidden flag for            =====
REM ===== Recovery partition                  =====
REM ================================================
Select disk 0
select partition 2
gpt attributes=0x8000000000000001
```