Technical White Paper

Preboot Wireless Networking on HP Business PCs

March 2024

Version: 4.34

Table of contents

Change Control 2
Summary 3
Overview of Preboot Wireless Networking Support for HP Business PCs 3
HP Sure Recover 3
Preboot Wireless Networking - Base Requirements 4
Description of Preboot Wireless Networking BIOS Settings 4
Security of Preboot Wireless Networking BIOS Settings 7
HP BIOS JSON-Based Configuration 8
JSON Preboot Wi-Fi Enrollment via HP CMSL 16
JSON Preboot Wi-Fi Enrollment via WMI 17
Preboot Wi-Fi Enrollment via BIOS 23
Additional Information 25
Appendix A – AMD Supported Wi-Fi modules 26
Appendix B – Intel Supported Wi-Fi modules 26
Appendix C – HP AMD Platforms with Pre-Boot Wi-Fi Support 28
Appendix D – HP Intel Platforms with Pre-Boot Wi-Fi Support 29

Change Control

Date	Reason for Change	Version
January 2020	• Initial Release	3.00VA
November 2023	• Update Scripts • Added tables to Appendix – indicating NIC-Platform support for pre-boot Wi-Fi	4.32VA
January 2024	• Update tables in Appendix – indicating removal of Qualcomm NIC support on AMD platforms	4.33VA
March 2024	• Feature Updates and Platform name corrections	4.34VA
	•	
	•	
	•	
	•	

© Copyright 2023, 2024 HP Development Company, L.P. All rights reserved. This Document is provided "as is." The information contained in this document, including URL, other web site references, and other specification documents are subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

HP Inc. or its subsidiaries may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. No licenses with respect to any intellectual property are being granted, expressly or impliedly, by the disclosure of the information contained in this document. No part of this document may be reproduced, stored, or transmitted in any form or by any means without the express written consent of HP Inc., however you may copy and use this document for your internal reference purposes.

Summary

This whitepaper describes the current support for BIOS-initiated preboot wireless networking on HP business PCs. It also briefly examines some of the business solutions that are made possible when the BIOS connects to a wireless network in the preboot environment. The use cases for this feature include but are not limited to the following:

- HP Sure Recover
- Wireless PXE boot
- BitLocker Network Unlock

Overview of Preboot Wireless Networking Support for HP Business PCs

Beginning with 2020 model year platforms, HP is delivering a preboot wireless networking capability whereby the BIOS itself can use the on-board Wireless LAN (WLAN or Wi-Fi®, used interchangeably) adapter to connect securely to wireless networks. Four different WLAN topologies are supported, each with its own specific connection requirements:

- WPA2-Personal (PSK)
- Enterprise EAP-TLS
- Enterprise PEAP
- Enterprise TTLS

In addition, the BIOS can be configured to connect to insecure, open wireless networks. One topology not currently supported is the so-called captive portal wireless connection mechanism, whereby an open wireless network responds to application-level HTTP/S requests with a secure login page to which the client can authenticate. Lack of support for captive portals in pre-boot Wi-Fi is an industry-wide limitation.

Once properly configured, the preboot Wi-Fi capability allows the BIOS to connect to an authenticated wireless network and use the WLAN interface in the same way that an ordinary wired Ethernet LAN adapter is used. Hence, this white paper will focus on the configuration requirements for securely connecting to wireless networks.

Some platforms will ship with preboot wireless networking support; others may have such support added in a field BIOS release. For specifics, please contact your HP sales representative, or view the technical documentation available for your platform on support.hp.com.

BIOS updates can be downloaded from support.hp.com

HP Sure Recover

A feature that can be leveraged with the use of Pre-Boot Wi-Fi is HP Sure Recover. HP Sure Recover can leverage the pre-boot networking capabilities when initially provisioning systems or in response to system drive corruption, destructive malware, or a scheduled recovery event. HP Sure Recover reduces downtime by enabling automated, policy-based reimaging without the need to create recovery media or contact the help desk. Recovery can be performed through connection to the network or through HP Sure Recover with Embedded Reimaging, an optional feature that includes additional storage mounted on the PC's system board for recovery without a network connection. HP Sure Recover is available on select commercial HP PCs.

HP provides Corporate-ready system images with optimized drivers and essential applications by default. However, an organization may also choose to host a custom corporate image either on the Internet or a corporate intranet. The recovery time will vary due to image size, network capability and congestion, etc.

Preboot Wireless Networking - Base Requirements

WLAN Adapter

The platform must integrate a supported WLAN adapter to support preboot wireless networking. Within this document, Appendices A through D indicate supported WLAN adapters and the platforms that support UEFI-Preboot wireless network features.

Platform BIOS

The platform BIOS must support the preboot Wi-Fi feature.
Once the base requirements are met, it is possible to configure the BIOS to connect successfully to a supported WLAN topology. Once a preboot Wi-Fi configuration is known to be working, it is possible to use the preboot Wi-Fi networking capability in the same way as wired Ethernet networking can be used. There is a single exception to this. Specifically, since preboot Wi-Fi requires the BIOS to authenticate itself to the WLAN infrastructure during POST, it is not possible to support a Wake-on-WLAN feature. This is because there is no properly authenticated WLAN connection until the platform powers on. Hence, magic packet wake-up through the WLAN interface is not supported.

Description of Preboot Wireless Networking BIOS Settings

HP supports six (6) different Wi-Fi preboot wireless profiles.
Each of these profiles can be configured with several different settings to allow for successful connection to a wireless networking environment. The names of each of these profiles follows a simple ordinal convention:

Preboot Wi-Fi Profile 1
Preboot Wi-Fi Profile 2
Preboot Wi-Fi Profile 3
Preboot Wi-Fi Profile 4
Preboot Wi-Fi Profile 5
Preboot Wi-Fi Profile 6

Each of these topologies requires specific configuration settings. Hence, each profile represents all the configuration settings required to successfully connect to a particular wireless network. Moreover, these profiles can be arranged in a priority order so that one profile is preferred over another when more than one wireless network is within range of the computer attempting to connect. BIOS might not be able to support all 6 profiles if any of the profile uses large amount of configuration data and Wi-Fi profile configurations are saved in BIOS non-volatile RAM(NVRAM).

In addition to these six profile names, HP supports several configuration settings that apply to all preboot wireless networking profiles:

Preboot Wi-Fi Auto Connect Profile Priority
Preboot Wi-Fi Profile Set Status
Preboot Wi-Fi Timeout
Preboot Wi-Fi Master Auto Connect

Each of these settings is explained in detail below.

Preboot Wi-Fi Wireless Networking Topology Settings

Since Wi-Fi settings often have to connect to multiple profiles, the HP preboot solution uses a JSON-encoded mechanism of setting as many as six (6) different preboot Wi-Fi profiles. In general, the following hold true for each Wi-Fi topology that might be enrolled in a particular profile.

WPA2-Personal	Security is based on the client presenting an SSID/password combination to the wireless access point. If the SSID/password combination authenticates, then the client is allowed to connect. The HP Preboot Wi-Fi solution requires the following settings: • SSID • Type must be **Personal** • Password • AutoConnect • Scan Anyway (optional)
Enterprise PEAP	Security is based on two factors. In this context, *server* refers to the Radius server that authenticates client and server when a client attempts to connect to a wireless network. • The client authenticates the server using a Certificate Authority certificate to verify that the server's certificate is trusted. • The server authenticates the client based on an identity and password that the client sends to the server. Because both client and server are authenticated, the HP preboot Wi-Fi solution requires the following settings: • SSID • Type must be **PEAP** • Identity • CaCert • Password • AutoConnect • Scan Anyway (optional)
Enterprise TTLS	As with Enterprise PEAP, security is based on two factors. The way these two factors interact in the networking infrastructure is different, but the fundamental security architecture is in fact quite similar: • The client authenticates the server using a Certificate Authority certificate to verify that the server's certificate is trusted. • The server authenticates the client based on an identity and password that the client sends to the server. Because both client and server are authenticated, the HP preboot Wi-Fi solution requires the following settings: • SSID • Type must be **TTLS** • Identity • CaCert • Password • AutoConnect • Scan Anyway (optional)
Enterprise EAPTLS	Just like PEAP and TTLS, EAPTLS authenticates both client and server. However, EAPTLS does not rely on a password for client authentication. Instead, it uses a certificate to perform client authentication. As such, each client requires both a client certificate and a private key. Hence, the following settings are required:

- SSID
- Type must be **EAPTLS**
- Identity
- CaCert
- ClientCert
- ClientPvtKey
- ClientPvtKeyPassword
- AutoConnect
- Scan Anyway (optional)

Preboot Wi-Fi Wireless Networking Settings Detailed Descriptions

The following discussion provides details of the meaning of each preboot Wi-Fi configuration setting. These settings apply individually to each of the six wireless networking profiles mentioned above, and to the on-demand setting.

Field Name (Not Case Sensitive)	Required (R) Optional (O) Not Used (X)				Description	Write Only (private data)	Type	Possible values/properties
	PSK	TLS	TTLS	PEAP				
SSID	R	R	R	R	The name of the Wi-Fi network	N	UTF-8 String	Value required on every set
Type	R	R	R	R	Security or authentication type	N	Enumerated List	• Personal • EAPTLS • PEAP • TTLS
AutoConnect	O	O	O	O	Connect automatically during pre-boot if the network is in range (and the priority allows).	N	Enumerated List	• Enable • Disable (default)
Password	O	X	O	O	Hidden/private data	**Y**	UTF-8 String	
CaCert	X	R	R	R	Certificate Authority Certificate. See **NOTE** below.	N	UTF-8 String	Base64 encoded certificate
ClientCert	X	R	X	X	Client Certificate	N	UTF-8 String	Base64 encoded certificate
ClientPvtKey	X	R	X	X	Client Private Key	**Y**	UTF-8 String	Base64 encoded key
ClientPvtKeyPassword	X	O	X	X	Password related to the client private key	**Y**	UTF-8 String	
Identity	X	R	R	R	Unique identity for the client. When used in conjunction with client certificates, the identity must match the Subject CN= field of the client certificate.	N	UTF-8 String	
ScanAnyway	O	O	O	O	When an SSID is not broadcasting, this setting tells the preboot environment to attempt to connect to the configured SSID anyway.	N	Enumerated List	• Enable • Disable (default)

NOTE on CaCert. There are many different ways to securely issue a client certificate. The simplest way is for a *root certificate authority* – or root CA – to issue the client certificate. If this is the deployment environment for an HP preboot Wi-Fi solution, then the **CaCert** value should be the public key certificate of the root certificate authority. However, it is more common for one or more intermediate certificate authorities – or intermediate CAs – to be part of the deployment environment. If so, then **CaCert** must be the public key certificate of the intermediate CA that issues the client certificate. In a typical hierarchy, there is only one Intermediate CA:

- Root CA
  - ↳ Intermediate CA
    - ↳ Client Certificate

However, there might hypothetically be any number of intermediate CAs. If so, then the Intermediate CA that signs the client certificate must be provided as the **CaCert** value:

- Root CA
  - ↳ Intermediate CA 1
    - ↳ Intermediate CA 2
      - ↳ …
        - ↳ Intermediate CA *n*
          - ↳ Client Certificate

In the above example, Intermediate CA *n* must be provided as the **CaCert** value.

Security of Preboot Wireless Networking BIOS Settings

Any review of the configuration settings required for secure preboot Wi-Fi connections reveals that each contains sensitive information – like identity fields and passwords and private keys, among other data items. Many of these data fields are *write-only*, meaning that once written they will not be revealed to outside callers except where authorized. An example of an authorized outside caller is the preboot Wi-Fi connection manager that is part of HP BIOS. The preboot Wi-Fi connection manager must use these fields in order to successfully authenticate to the wireless networking infrastructure; hence it can gain access to these fields.

Nevertheless, these configuration settings must be stored somewhere, and since they *are stored* they must be protected. HP stores these configuration settings in the BIOS non-volatile RAM (NVRAM) region, on the Serial Programming Interface (SPI) chip that contains the HP BIOS and other firmware. These settings are stored as Universal Extensible Firmware Interface (UEFI) NVRAM variables. Unlike most other variables however, the values associated with these configuration settings are stored in an encrypted format. Each field is encrypted with a secure symmetric key. For those who are interested in such details, this secure symmetric key is derived from a separate secure symmetric key that is unique for each HP system that is shipped. This unique key is written permanently in the system hardware; access to the key is controlled by the HP Endpoint Security Controller; the HP Endpoint Security Controller never reveals the value of this key to any requestor. Instead, to authorized requestors the HP ECP device derives a key using an industry standard key derivation function. Then it delivers that key to the authorized requestor. The HP BIOS is one such authorized requestor; therefore HP BIOS has access to its own unique, derived key that it uses to encrypt and decrypt certain UEFI NVRAM variables, including the preboot Wi-Fi configuration settings.

HP BIOS JSON-Based Configuration

HP supports up to six (6) different wireless networking configuration profiles. Since configuring even one wireless networking profile can be quite complex, HP uses a JSON-encoded mechanism for enrolling these profiles. JSON is used to encode the data; then the data is enrolled into the BIOS via WMI.
Using a tag and data value format like JSON allows support of several profiles in a more consistent way and perhaps allow for better scalability (if needed). Supporting 6 profiles as above requires the following public WMI settings, mentioned above in brief and detailed here and below.

Preboot Wi-Fi Profile 1
Preboot Wi-Fi Profile 2
Preboot Wi-Fi Profile 3
Preboot Wi-Fi Profile 4
Preboot Wi-Fi Profile 5
Preboot Wi-Fi Profile 6

In addition, the HP preboot wireless networking solution supports a user-configurable wireless profile that allows end users to connect to a user-specific wireless network connection – such as a home wireless router – on a boot-by-boot basis. This connection must be manually selected by the end user each boot via F3 in order to connect.

Finally, the HP preboot wireless networking solution supports a few global configuration settings that apply to all preboot wireless network connections:

Preboot Wi-Fi Auto Connect Profile Priority
Preboot Wi-Fi Profile Set Status
Preboot Wi-Fi Timeout
Preboot Wi-Fi Master Auto Connect

In order to configure each of the six wireless profiles and the on-demand profile, a string is written to each profile setting that conforms to the following:

- Standard JSON rules should apply.
- Open and close curly braces are required.
- All names and values must be in double quote pairs.
- White space outside of the double quote pairs will be ignored.
- The string *null* without the double quotes is used to indicate that no value is present.
- The order of the name:value pairs does not really matter.

Preferably, all the JSON fields for a profile should be set in the same set/write command but if that isn't the case see Multiple Sets/WritesMultiple Sets/Writes below. Optional fields are not required to be set; however, all possible fields will be returned on a read/get.
For each profile for which nothing has been set/configured, the public WMI get/read will return nothing or an empty string just as for all other string class settings.

Base64 Encoded Data

The following fields contain binary values and in order to be used with our current tools, which assume strings, must be base64 encoded:

CaCert
ClientCert

ClientPvtKey

NOTE: The base64 strings corresponding to these are very long. In the examples below, they have been shortened (with an ellipsis …) for readability.

Fields with Hidden/Private Data

Some fields are essentially write-only or hidden/private data. The actual data will never be returned via the public WMI interfaces (JSON strings). If data has been set for the field, then a string of 6 asterisks "******" will be returned in the string on gets/reads. Since "ClientPvtKey" and "ClienPvtKeyPassword" are closely connected, if either one has been set, both will return "******".

ClientPvtKey
ClientPvtKeyPassword
Password

Unconfiguring / Deleting a profile

A profile is deleted in the same way as any other string class setting by setting an empty string value.

The following is an example of deleting the profile associated with Preboot Wi-Fi Profile 1.

Get/Read
Preboot Wi-Fi Profile 1
{ "SSID": "TestPSK", "Type": "Personal", "AutoConnect": "Disable", "Password":"******" }

Set/Write
Preboot Wi-Fi Profile 1

Get/Read
Preboot Wi-Fi Profile 1

Note that doing a read and checking for an empty string will tell you whether or not a profile is configured.

WPA2-Personal (PSK)

The fields for WPA2-Personal/PSK security/authentication:

- SSID
- Type
- AutoConnect
- Password
- ScanAnyway

The following are examples of what would be written and read when configuring Preboot Wi-Fi Profile 1 for a Personal PSK profile.

Set/Write
Preboot Wi-Fi Profile 1
{ "SSID": "TestPSK", "Type": "Personal", "AutoConnect": "Disable", "Password":"P@55w0rd" }

Get/Read

Preboot Wi-Fi Profile 1

```
{ "SSID":  "TestPSK", "Type": "Personal", "AutoConnect": "Disable", "Password":”******” }
```

EAPTLS

The fields for enterprise EAPTLS security/authentication are:

- SSID
- Type
- AutoConnect
- CaCert
- ClientCert
- ClientPvtKey
- ClientPvtKeyPassword
- Identity
- ScanAnyway

The following are examples of what would be written and read when configuring Preboot Wi-Fi Profile 1 for an Enterprise EAPTLS profile. Note that these should be written as a single line (line breaks are cause by Word or introduced for readability).

Set/Write

Preboot Wi-Fi Profile 1

```
{  "SSID":  "TestTLS", "Type":  "EAPTLS", "AutoConnect": "Enable", "CaCert": “ABC…123”,
"ClientCert": “XYZ…321”,  "ClientPvtKey": “555…777”, "ClientPvtKeyPassword": “P@ssw0rd”,
"Identity": “anonymous” }
```

Get/Read

Preboot Wi-Fi Profile 1

```
{  "SSID":  "TestTLS", "Type":  "EAPTLS", "AutoConnect": "Enable", "CaCert": “ABC…123”,
"ClientCert": “XYZ…321”,  "ClientPvtKey": “******”, "ClientPvtKeyPassword": “******”,
"Identity": “anonymous” }
```

PEAP & TTLS

The fields for enterprise PEAP and TTLS security/authentication are:

- SSID
- Type
- AutoConnect
- CaCert
- Identity
- Password
- ScanAnyway

The following are examples of what would be written and read when configuring Preboot Wi-Fi Profile 1 for an Enterprise PEAP and Preboot Wi-Fi Profile 2 for an TTLS profile. Note that these should be written as a single line (line breaks are cause by Word or introduced for readability).

Set/Write

Preboot Wi-Fi Profile 1

{ "SSID": "TestPEAP", "Type": "PEAP", "AutoConnect": "Enable", "CaCert": “abc…123”, "Identity": “SampleID”, "Password":”” }

Preboot Wi-Fi Profile 2

{ "SSID": "TestTTLS", "Type": "TTLS", "AutoConnect": "Disable", "CaCert":”AAA…333”, "Identity":”SampleID1, "Password":”P@s5Word” }

Get/Read

Preboot Wi-Fi Profile 1

{ "SSID": "TestPEAP", "Type": "PEAP", "AutoConnect": "Enable", "CaCert": “abc…123”, "Identity": “SampleID”, "Password":”******” }

Preboot Wi-Fi Profile 2

{ "SSID": "TestTTLS", "Type": "TTLS", "AutoConnect": "Disable", "CaCert":”AAA…333”, "Identity":”SampleID1, "Password":”******” }

Working with Certificates

When copying the base64-encoded certificate content, open it in a text editor and remove the -----BEGIN CERTIFICATE----- header and -----END CERTIFICATE----- footer, and also remove the line breaks in the base64-encoded certificate file.

Here’s an example.
The original base64-encode certificate file looks like:

```
 -----BEGIN CERTIFICATE-----
MIIDqTCCApGgAwIBAgIUGzBRxvA/JLj81fp+nF+XEM3FnP8wDQYJKoZIhvcNAQEL
BQAwZDELMAkGA1UEBhMCVVMxDjAMBgNVBAgMBVN0YXRlMQ0wCwYDVQQHDARDaXR5
MRAwDgYDVQQKDAdDb21wYW55MQwwCgYDVQQLDANPcmcxFjAUBgNVBAMMDVNQTS1F
Sy1DZXJ0IzEwHhcNMjIwNDA3MTkwNjAzWhcNMzIwNDA0MTkwNjAzWjBkMQswCQYD
VQQGEwJVUzEOMAwGA1UECAwFU3RhdGUxDTALBgNVBAcMBENpdHkxEDAOBgNVBAoM
B0NvbXBhbnkxDDAKBgNVBAsMA09yZzEWMBQGA1UEAwwNU1BNLUVLLUNlcnQjMTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALO2rkigeXCMLpfUy9/p/Uc6
r60wyLqYEarcUiR+Xl1O1I9ROeHpPDRlMZTgks4MpVbPBcGG4fvZkZSqFmJe8e5e
Lf9ugE7uaH2ZVY9szfJNbyHy/xXcVKbISrQ4LaQk761CQf/MvVCaevX+b9bKqy6a
w826Ut4WSe0D0tvRDPyP2FWqbUQlq2nf0+khbIEZ7wCpccrUeYeag6r0m0xk1fsM
OvS4DvzOezi+CdayzrtTsIRx78kns3UVCiFQHTe3U1Z4hwwRr3S6mc4D2nxq1R+H
7ljGogvgq7QhdLUO9I66Hxs2fdjEewMF2VkVgmtVWbGx5VtBpzCJ2j3plGLJB+MC
AwEAAaNTMFEwHQYDVR0OBBYEFHH085yBTBuqnhPp4bbNvdDDVtLtMB8GA1UdIwQY
MBaAFHH085yBTBuqnhPp4bbNvdDDVtLtMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZI
hvcNAQELBQADggEBAKI1n26k4lMjwz2Fym8TkclZI5ckJIc/6nipsOrl9kicAuG4
eHv5WnUxds0sMyQZdt0GLS6O+Ie0wWB7Ertr+dBhmqucqNxcV/fyWTP39JMpEDGI
v7kEohWNRiKBH6EoI8p1OzkGfrZtNQw+qT5OUQnRkXCpRXzFaqkomL7ppxgyOgZ0
RJvsBcnTlvOqg3oC5Jcnn9itVG3wF3h0NzqnFMEjPrZfJeZT9hVbllRGy9zCDq2A
qCKo3JK0kt6O03Cf2dENsodG5bheEe2MfuIouX6pviOn1hy6p3VSlNjJ/VeDGMUG
c16UdxEttsHaeBq48BaTwMWzLmhJOKZYWbgGXlc=
-----END CERTIFICATE-----
```

After the edit it should look like:

```
MIIDqTCCApGgAwIBAgIUGzBRxvA/JLj81fp+nF+XEM3FnP8wDQYJKoZIhvcNAQELBQAwZDELMAkGA1UEBh
MCVVMxDjAMBgNVBAgMBVN0YXRlMQ0wCwYDVQQHDARDaXR5MRAwDgYDVQQKDAdDb21wYW55MQwwCgYDVQQL
DANPcmcxFjAUBgNVBAMMDVNQTS1FSy1DZXJ0IzEwHhcNMjIwNDA3MTkwNjAzWhcNMzIwNDA0MTkwNjAzWj
BkMQswCQYDVQQGEwJVUzEOMAwGA1UECAwFU3RhdGUxDTALBgNVBAcMBENpdHkxEDAOBgNVBAoMB0NvbXBh
bnkxDDAKBgNVBAsMA09yZzEWMBQGA1UEAwwNU1BNLUVLLUNlcnQjMTCCASIwDQYJKoZIhvcNAQEBBQADgg
EPADCCAQoCggEBALO2rkigeXCMLpfUy9/p/Uc6r60wyLqYEarcUiR+Xl1O1I9ROeHpPDRlMZTgks4MpVbP
BcGG4fvZkZSqFmJe8e5eLf9ugE7uaH2ZVY9szfJNbyHy/xXcVKbISrQ4LaQk761CQf/MvVCaevX+b9bKqy
6aw826Ut4WSe0D0tvRDPyP2FWqbUQlq2nf+khbIEZ7wCpccrUeYeag6r0m0xk1fsMOvS4DvzOezi+Cdayz
rtTsIRx78kns3UVCiFQHTe3U1Z4hwwRr3S6mc4D2nxq1R+H7ljGogvgq7QhdLUO9I66Hxs2fdjEewMF2Vk
VgmtVWbGx5VtBpzCJ2j3plGLJB+MCAwEAAaNTMFEwHQYDVR0OBBYEFHH085yBTBuqnhPp4bbNvdDDVtLtM
B8GA1UdIwQYMBaAFHH085yBTBuqnhp4bbNvdDDVtLtMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEL
BQADggEBAKI1n26k4lMjwz2Fym8TkclZI5ckJIc/6nipsOrl9kicAuG4eHv5WnUxds0sMyQZdt0GLS6O+I
e0wWB7Ertr+dBhmqucqNxcV/fyWTP39JMpEDGIv7kEohWNRiKBH6EoI8p1OzkGfrZtNQw+qT5OUQnRkXCp
RXzFaqkomL7ppxgyOgZ0RJvsBcnTlvOqg3oC5Jcnn9itVG3wF3h0NzqnFMEjPrZfJeZT9hVbllRGy9zCDq
2AqCKo3JK0kt6O03Cf2dENsodG5bheEe2MfuIouX6pviOn1hy6p3VSlNjJ/VeDGMUGc16UdxEttsHaeBq4
8BaTwMWzLmhJOKZYWbgGXlc=
```

Multiple Sets/Writes

Preferably, all the JSON fields for a profile should be set in the same set/write command. However, while multiple writes can be done the last write must contain the hidden/private data and any other fields that may need to be updated. This is a security requirement so that someone cannot redirect to a malicious MITM access point and leverage credentials meant for use with a different AP.

This implies that the software component that sets a field that contains hidden/private data (private key, private key password, password) needs to be able to set all the hidden/private data for that profile.

The following is an example sequence setting Preboot Wi-Fi Profile 3 where only partial information is set. Note that these should be written as a single line (line breaks are cause by Word or introduced for readability).

Get/Read
Preboot Wi-Fi Profile 3

Set/Write *// client items not set, optional fields not present*
Preboot Wi-Fi Profile 3
{ "SSID": "TestTLS", "Type": "EAPTLS", "CaCert": “ABZ…123”, "ClientCert": null, "ClientPvtKey": null, "Identity": “anonymous” }

Get/Read *// read & prepare to set client items, optional fields reported on Get/Read*
Preboot Wi-Fi Profile 3
{ "SSID": "TestTLS", "Type": "EAPTLS", "AutoConnect": "Disable", "CaCert": “ABZ…123”, "ClientCert": null, "ClientPvtKey": null, "ClientPvtKeyPassword": null, "Identity": “anonymous” }

Set/Write *// set client & hidden/private items, change identity*
Preboot Wi-Fi Profile 3
{ "SSID": "TestTLS", "Type": "EAPTLS", "AutoConnect": "Disable", "CaCert": “ABZ…123”, “ClientCert": “XYZ…321”, "ClientPvtKey": “zzz…888”, "ClientPvtKeyPassword": “Pa55word”, "Identity": “specialID” }

Get/Read *// hidden/private items return “******”*

Preboot Wi-Fi Profile 3

```
{ "SSID":  "TestTLS", "Type":  "EAPTLS", "AutoConnect": "Disable", "CaCert": “ABZ…123”,
"ClientCert": “XYZ…321”,  "ClientPvtKey": “******”, "ClientPvtKeyPassword": “******”,
"Identity": “specialID” }
```

The following is an example of what will be read back if you do a set without specifying hidden/private data.

Get/Read

Preboot Wi-Fi Profile 4

Set/Write *// private key set but no private key password*

Preboot Wi-Fi Profile 4

```
{ "SSID":  "TestTLS4", "Type":  "EAPTLS", "AutoConnect": "Disable", "CaCert": “124…acb”,
“ClientCert": “321…xyz”,  "ClientPvtKey": “zzz…888”, "Identity": “IDen” }
```

Get/Read *// hidden/private items return “******”*

Preboot Wi-Fi Profile 4

```
{ "SSID":  "TestTLS4", "Type":  "EAPTLS", "AutoConnect": "Disable", "CaCert": “124…acb”,
"ClientCert": “321…xyz”,  "ClientPvtKey": “******”, "ClientPvtKeyPassword": “******”, "Identity": “IDen” }
```

Preboot Wi-Fi Auto Connect Profile Priority

This setting is an ordered list which the BIOS uses to determine which profile to connect to if more than one Wi-Fi network is visible/available. All profiles are included and the IT admin can change the order as needed. The BIOS implementation may need to use a separate NVRAM variable and map to/from Intel priority variable.

Preboot Wi-Fi Auto Connect Profile Priority
    Preboot Wi-Fi Profile 1
    Preboot Wi-Fi Profile 2
    Preboot Wi-Fi Profile 3
    Preboot Wi-Fi Profile 4
    Preboot Wi-Fi Profile 5
    Preboot Wi-Fi Profile 6

If a profile doesn’t have autoconnect enabled, then it is ignored. Also, HP BIOS features generally uses wired connection if available and uses Wi-Fi only if it’s not available except for features that has unique policy on Wi-Fi priority over wired connections.

Preboot Wi-Fi Profile Set Status

The setting “Preboot Wi-Fi Profile Set Status” is a read-only string that does not persist across boots. It may return various strings indicating the status of the last set. It will be an empty string if no “set/write” operation has been done on the current boot. Some examples are below:

- “The following field(s) are missing: “ followed by a list of fields that are missing but are required such as CaCert, ClientCert, ClientPvtKey, and Identity.
- “SSID and Type are required on all sets.”
- “Complete: all required fields for the specified ‘Type’ are present.”

Preboot Wi-Fi Profile Connection Timeout

The setting “Preboot Wi-Fi Profile Connection Timeout” is used to inform the firmware how long it should wait before assuming the connection will fail. If autoconnect is enabled, firmware will wait this amount of time even if SSID isn’t visible or until a connection is established.

Preboot Wi-Fi Master Auto Connect

This setting controls whether or not we want anything to autoconnect during POST versus if we are trying to establish a Wi-Fi connection and the profile says autoconnect then it will be a candidate. Normally this setting would be disabled and enabled in cases where someone wanted to do BitLocker Network unlock, HP Sure Recover, or other situation where you know it needs the network.

Preboot Wi-Fi Wireless Networking Global Settings Detailed Descriptions

The following table details the meaning of each of the global wireless networking settings discussed previously.

Field Name (Not Case Sensitive)	Description	Write Only (private data)	Type	Possible values/properties
Preboot Wi-Fi Auto Connect Profile Priority	An ordered list of the six supported wireless connection profiles. The order represents the priority order of connection when more than one wireless network is in range of the computer.	N	UTF-8 String List, with each item in the list separated by a newline	To connect in profile priority order 5, 2, 1, 4, 6, 3, use the following value: Preboot Wi-Fi Profile 5 Preboot Wi-Fi Profile 2 Preboot Wi-Fi Profile 1 Preboot Wi-Fi Profile 4 Preboot Wi-Fi Profile 6 Preboot Wi-Fi Profile 3
Preboot Wi-Fi Profile Set Status				
Preboot Wi-Fi Timeout	This is a numeric value that defines the number of seconds that the boot will wait for the wireless connection’s link status to be ready before proceeding with the rest of the boot. The default is 60. If the wireless networking connection’s link status is ready prior to the expiration of the timeout, then the boot proceeds as soon as the wireless link status is established It’s available in BIOS setup – Main->Update System BIOS -> Network Configuration Settings	N	UTF-8 String representation of a number	• 0-65535

Preboot Wi-Fi Master Auto Connect	This can be enabled or disabled. When enabled, all wireless profiles that might attempt to connect in a particular priority order will attempt to connect automatically every boot. When disabled, the auto connect value of each wireless profile will be asserted. It’s available in BIOS setup – Main->Update System BIOS -> Network Configuration Settings	N	Enumerated List	• Enable (default) • Disable

JSON Preboot Wi-Fi Enrollment via HP CMSL

HP supports a tool called Client Management Script Library to facilitate the management of HP PC settings via PowerShell scripts. This includes HP BIOS settings.

- HP Client Management Script Library (CMSL): https://developers.hp.com/hp-client-management/doc/client-management-script-library
- HP CMSL BIOS and Device scripts: https://developers.hp.com/hp-client-management/doc/bios-and-device

There are several available CMSL scripts that are very useful for managing preboot Wi-Fi settings using PowerShell. These include:

Working with HP BIOS Settings directly

Function	Description
Get-HPBIOSSetting	Retrieves an HP BIOS Setting object by name on the current device unless specified for another platform
Get-HPBIOSSettingValue	Retrieves the value of an HP BIOS Setting on the current device unless specified for another platform
Get-HPBIOSSettingsList	Retrieves all BIOS settings on the current device unless specified for another platform
Set-HPBIOSSettingValue	Sets the value of a BIOS setting on the current device unless specified for another platform
Set-HPBIOSSettingValuesFromFile	Sets one or more BIOS settings from a file on the current device unless specified for another platform
Set-HPBIOSSettingDefaults	Resets the BIOS settings to shipping defaults on the current device unless specified for another platform

Among the most useful scripts for setting preboot Wi-Fi settings using PowerShell is Set-HPBIOSSettingsValuesFrom File (https://developers.hp.com/hp-client-management/doc/Set-HPBIOSSettingValuesFromFile). This script allows for continued list of HP BiosConfigUtility (BCU) formatted text files when managing preboot Wi-Fi settings. The specific examples cited above can be used in the context of these HP CMSL scripts to manage HP BIOS settings related to preboot Wi-Fi settings.

JSON Preboot Wi-Fi Enrollment via WMI

HP supports a WMI interface between the operating system and the BIOS. This allows for any supported HP platform to configure and manage preboot Wi-Fi settings using WMI scripts. This section will provide some sample scripts that can be used to manage preboot Wi-Fi configuration settings.

HP InstrumentedBIOS WMI Namespace

```
$bios = Get-WmiObject -Namespace root/HP/InstrumentedBIOS -Class HP_BIOSSettingInterface
```

HP supports an HP-specific WMI namespace whose hierarchy is root/HP/InstrumentedBIOS. To connect to this namespace in PowerShell requires a single statement.

After successfully execution of the above statement in PowerShell, $bios is instantiated as an instance of the HP_BIOSSettingInterface WMI class that has three methods:

- FireTestEvent
- SetBIOSSettings
- SetSystemDefaults

SetBIOSSettings is the method we will use throughout these examples.

In the later script examples, the $bios object is the on instantiated by this command.

Clear Wi-Fi Settings

One of the most important operations is clearing out any existing Wi-Fi settings before enrolling new Wi-Fi settings. A clear operation can be useful as an initial step even when no preboot Wi-Fi configuration settings exist on the unit being configured. This is because in some environments – where hundreds or even thousands of computers are under management – it's not always evident whether specific configuration settings already exist. With the HP solution, one can always query the system, but if the goal is to completely reconfigure all Wi-Fi settings, an arbitrary *clear* operation can make sense.

The section above, entitled Unconfiguring / Deleting a Profile, provides details on how to clear out one of the supported 6 Wi-Fi profiles. To clear them all out requires that each of the 6 settings be cleared out in succession. The following PowerShell script can be used to clear out the Wi-Fi settings arbitrarily on any HP platform that supports preboot Wi-Fi.

> **Note**: Be sure to disable *Intel Active Management Technology (AMT)* and *Fast Boot* before configuring Wi-Fi settings on Intel systems. For AMD systems disable *AiMT* and *AMD DASH* before configuring Wi-Fi settings.

Clearing Wireless Settings

```
#
# (c) Copyright 2019 HP Development Company, L.P.
# This software and associated documentation (if any) is furnished under a license
# and may only be used or copied in accordance with the terms of the license.
# Except as permitted by such license, no part of this software or documentation may
# be reproduced, stored in a retrieval system, or transmitted in any
# form or by any means without the express written consent of HP Development Company.
#


$bios = Get-WmiObject -Namespace root/HP/InstrumentedBIOS -Class HP_BIOSSettingInterface

function Wi-FiClear()
{
    $bios.SetBiosSetting("Preboot Wi-Fi Profile 1", "", "")
    $bios.SetBiosSetting("Preboot Wi-Fi Profile 2", "", "")
    $bios.SetBiosSetting("Preboot Wi-Fi Profile 3", "", "")
    $bios.SetBiosSetting("Preboot Wi-Fi Profile 4", "", "")
    $bios.SetBiosSetting("Preboot Wi-Fi Profile 5", "", "")
    $bios.SetBiosSetting("Preboot Wi-Fi Profile 6", "", "")
    $bios.SetBiosSetting("Preboot Wi-Fi Master Auto Connect", "", "")
}

Wi-FiClear
Write-Host "Cleared out six (6) preboot Wi-Fi profiles”
```

EAP-TLS Example

```
#
# (c) Copyright 2019 HP Development Company, L.P.
# This software and associated documentation (if any) is furnished under a license
# and may only be used or copied in accordance with the terms of the license.
# Except as permitted by such license, no part of this software or documentation may
# be reproduced, stored in a retrieval system, or transmitted in any
# form or by any means without the express written consent of HP Development Company.
#
param(
    $SSID,
    $Identity,
    $CaCertPath,
    $ClientCertPath,
    $ClientPvtKeyPath,
    $PvtKeyPassword
)

$bios = Get-WmiObject -Namespace root/HP/InstrumentedBIOS -Class HP_BIOSSettingInterface

function Set-WiFiProfileEAPTLS()
{
    $CaCert  = Get-Content -Path $CaCertPath -Encoding Byte
    $CaCert = [Convert]::ToBase64String($CaCert)

    $ClientCert  = Get-Content -Path $ClientCertPath -Encoding Byte
    $ClientCert = [Convert]::ToBase64String($ClientCert)

    $ClientPvtKey  = Get-Content -Path $ClientPvtKeyPath -Encoding Byte
    $ClientPvtKey = [Convert]::ToBase64String($ClientPvtKey)


    $json = '{ "SSID": "' + $SSID + '", "Type":  "EAP-TLS", "AutoConnect": "Disable", ' +
            '"ScanAnyway": "Enable", "CaCert": "' + $CaCert + '", “ClientCert": "' +
            $ClientCert + '",  "ClientPvtKey": "' + $ClientPvtKey +
            '", "ClientPvtKeyPassword": "' + $PvtKeyPassword + '", "Identity": “' +
            $Identity + '” } '

    # Note the below "preboot wi-fi profile 1" is not case sensitive.
    $bios.setbiossetting("preboot wi-fi profile 1", $json, "")
}

Set-WiFiProfileEAPTLS
Write-Host "Configured Preboot Wi-Fi Profile 1"
```

If the above sample code is saved as PowerShell script `EapTlsSet`, it might be invoked thus. Note that by reading in the certificate and private key files as `Byte`, and then encoding them in base 64 format, it does not matter what format these files are written. So long as they are written in a valid, industry-standard format – such as DER or PEM – the code above will work:

```
PS C:\Wi-Fi®CertEnrollment> .\EapTlsSet.ps1 -SSID SSID -Identity machine.domain.com -CaCertPath
C:\temp\testcer\MyCA_Base64.cer -ClientCertPath C:\temp\testcer\MyClientCer_Base64.cer -
ClientPvtKeyPath C:\temp\testcer\MyPrivateKey_Base64.pvk -PvtKeyPassword password
```

TTLS Example

```
#
# (c) Copyright 2019 HP Development Company, L.P.
# This software and associated documentation (if any) is furnished under a license
# and may only be used or copied in accordance with the terms of the license.
# Except as permitted by such license, no part of this software or documentation may
# be reproduced, stored in a retrieval system, or transmitted in any
# form or by any means without the express written consent of HP Development Company.
#
param(
    $SSID,
    $Identity,
    $CaCertPath,
    $TtlsPassword
)

$bios = Get-WmiObject -Namespace root/HP/InstrumentedBIOS -Class HP_BIOSSettingInterface

function Set-WiFiProfileTTLS()
{
    $CaCert  = Get-Content -Path $CaCertPath -Encoding Byte
    $CaCert = [Convert]::ToBase64String($CaCert)

    $json = '{ "SSID": "' + $SSID + '", "Type":  "EAP-TTLS", "AutoConnect": "Disable", ' +
            '"ScanAnyway": "Enable", "CaCert": "' + $CaCert +
            '", "Password": "' + $TtlsPassword + '", "Identity": “' +
            $Identity + '” } '

    # Note the below "preboot wi-fi profile 3" is not case sensitive.
    $bios.setbiossetting("preboot wi-fi profile 3", $json, "")
}


Set-WiFiProfileTTLS
Write-Host "Configured Preboot Wi-Fi Profile 3"
```

PEAP Example

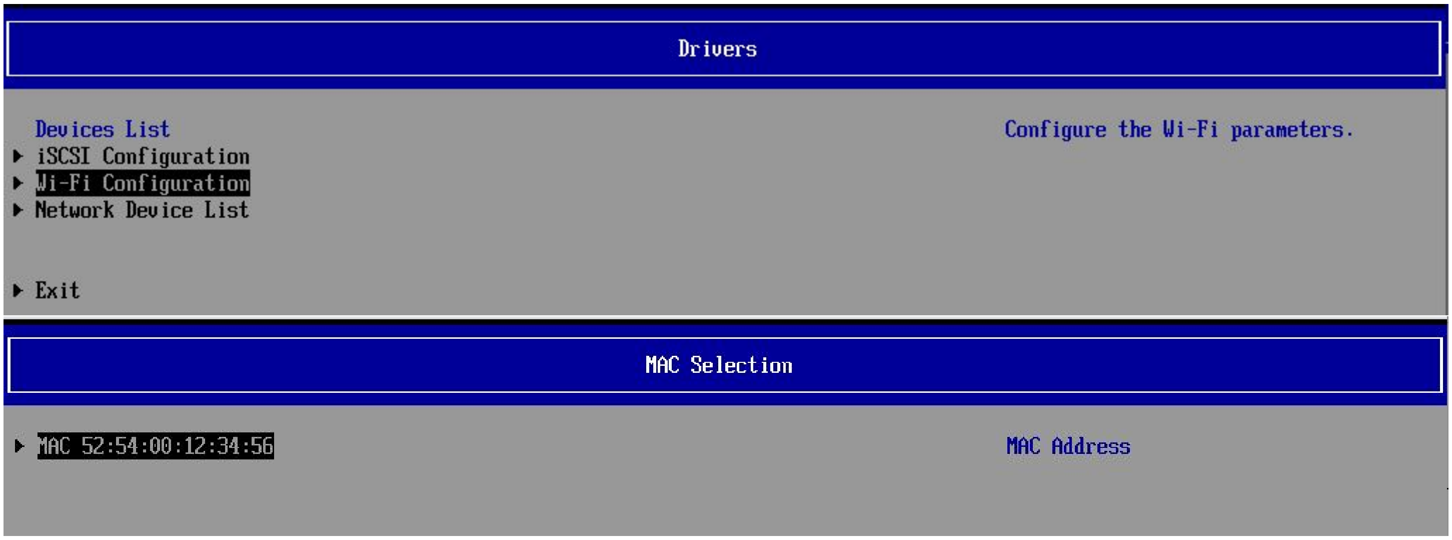
```
#
# (c) Copyright 2019 HP Development Company, L.P.
# This software and associated documentation (if any) is furnished under a license
# and may only be used or copied in accordance with the terms of the license.
# Except as permitted by such license, no part of this software or documentation may
# be reproduced, stored in a retrieval system, or transmitted in any
# form or by any means without the express written consent of HP Development Company.
#
param(
    $SSID,
    $Identity,
    $CaCertPath,
    $PeapPassword
)

$bios = Get-WmiObject -Namespace root/HP/InstrumentedBIOS -Class HP_BIOSSettingInterface

function Set-WiFiProfilePEAP()
{
    $CaCert  = Get-Content -Path $CaCertPath -Encoding Byte
    $CaCert = [Convert]::ToBase64String($CaCert)

    $json = '{ "SSID": "' + $SSID + '", "Type":  "EAP-PEAP", "AutoConnect": "Disable", ' +
            '"ScanAnyway": "Enable", "CaCert": "' + $CaCert +
            '", "Password": "' + $PeapPassword + '", "Identity": “' +
            $Identity + '” } '

    # Note the below "preboot wi-fi profile 2" is not case sensitive.
    $bios.setbiossetting("preboot wi-fi profile 2", $json, "")
}


Set-WiFiProfilePEAP
Write-Host "Configured Preboot Wi-Fi Profile 2"
```

WPA2-Personal Example

```
#
# (c) Copyright 2019 HP Development Company, L.P.
# This software and associated documentation (if any) is furnished under a license
# and may only be used or copied in accordance with the terms of the license.
# Except as permitted by such license, no part of this software or documentation may
# be reproduced, stored in a retrieval system, or transmitted in any
# form or by any means without the express written consent of HP Development Company.
#
param(
    $SSID,
    $Password
)

$bios = Get-WmiObject -Namespace root/HP/InstrumentedBIOS -Class HP_BIOSSettingInterface

function Set-WiFiProfilePersonal()
{
    $json = '{ "SSID": "' + $SSID + '", "Type":  "Personal", "AutoConnect": "Disable", ' +
            '"ScanAnyway": "Enable",", "Password": "' + $Password + '” } '

    # Note the below "preboot wi-fi profile 4" is not case sensitive.
    $bios.setbiossetting("preboot wi-fi profile 4", $json, "")
}


Set-WiFiProfilePersonal
Write-Host "Configured Preboot Wi-Fi Profile 4"
```

Preboot Wi-Fi Enrollment via BIOS

This section describes steps to configure preboot Wi-Fi thru BIOS setup which is accessible during system boot-up by using the ‘F3’ function key. On entering F3 menu, ‘Wi-Fi Configuration’ link will provide a link to MAC address of Wi-Fi controller which in turn provides the connection status, list of access points and allow users to connect to it by entering credentials for various types of Wi-Fi topology. File explorer will be available to select the certificate files from media and unique GUID value is required for each certificate since it’s stored as part of same variable.



Drivers

Devices List
▸ iSCSI Configuration
▸ Wi-Fi Configuration
▸ Network Device List

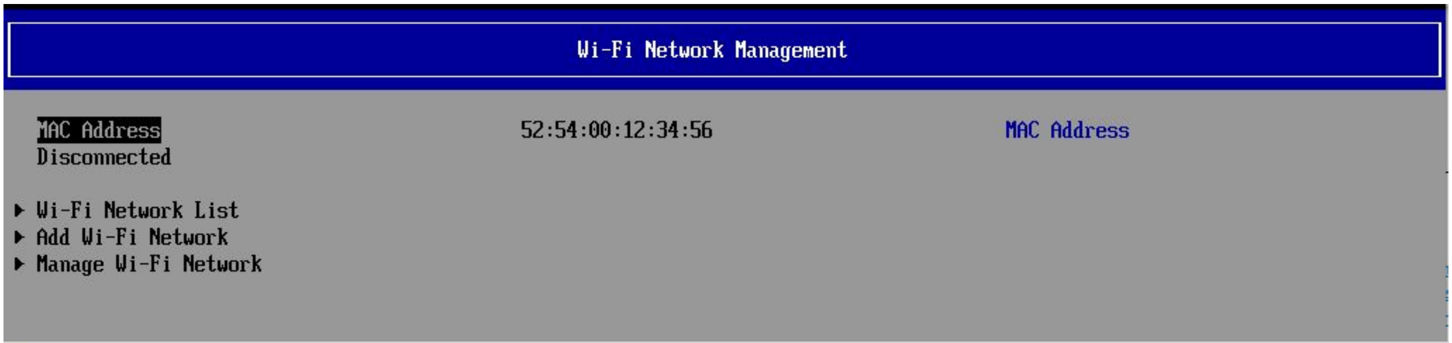
▸ Exit

Configure the Wi-Fi parameters.

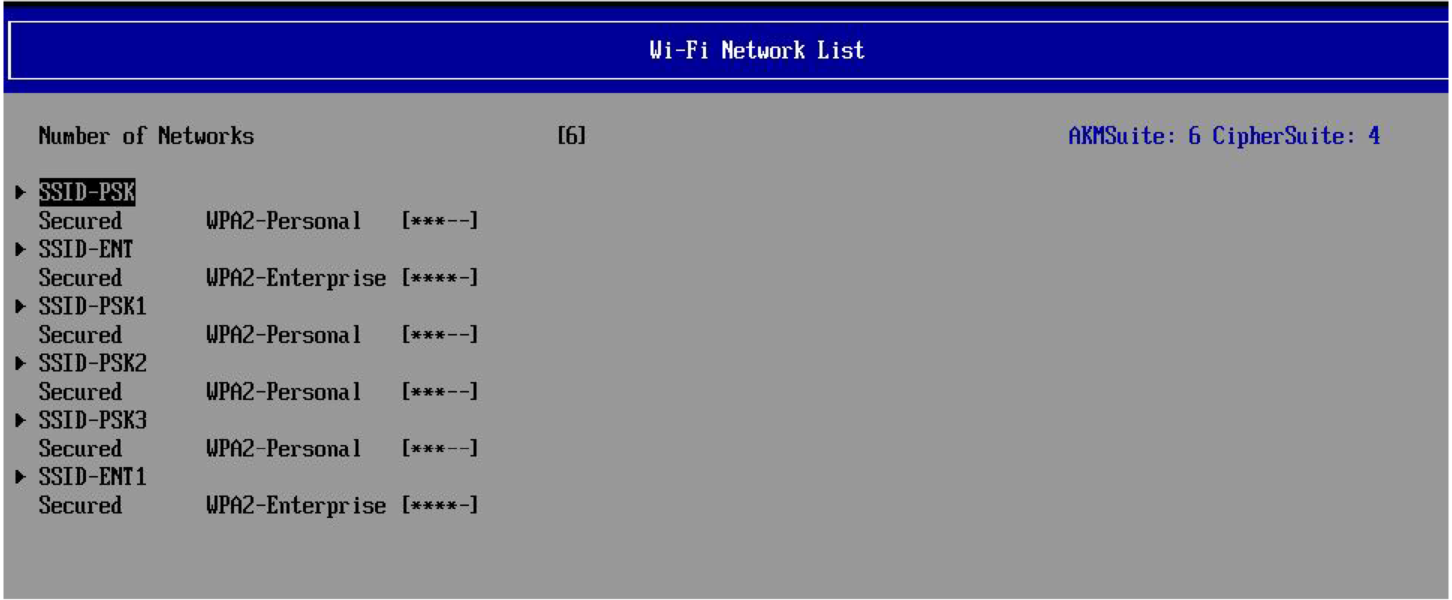
MAC Selection

▸ MAC 52:54:00:12:34:56

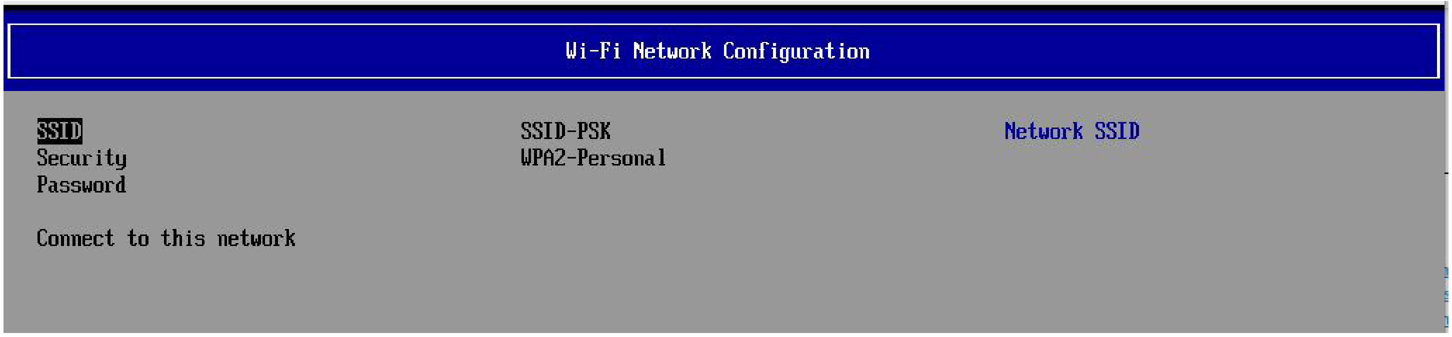
MAC Address



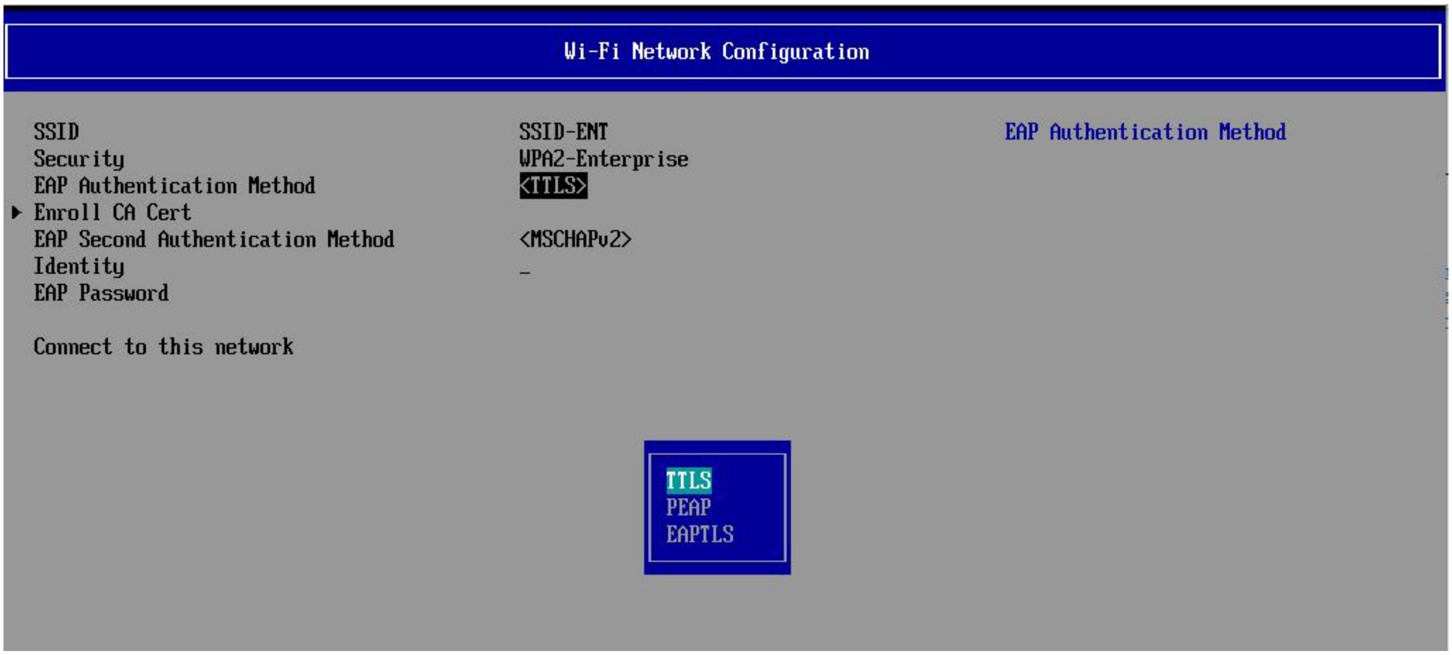
```
                    Wi-Fi Network Management

MAC Address                  52:54:00:12:34:56          MAC Address
Disconnected

▶ Wi-Fi Network List
▶ Add Wi-Fi Network
▶ Manage Wi-Fi Network
```



```
                       Wi-Fi Network List

Number of Networks                 [6]              AKMSuite: 6 CipherSuite: 4

▶ SSID-PSK
  Secured      WPA2-Personal   [***--]
▶ SSID-ENT
  Secured      WPA2-Enterprise [****-]
▶ SSID-PSK1
  Secured      WPA2-Personal   [***--]
▶ SSID-PSK2
  Secured      WPA2-Personal   [***--]
▶ SSID-PSK3
  Secured      WPA2-Personal   [***--]
▶ SSID-ENT1
  Secured      WPA2-Enterprise [****-]
```



```
                  Wi-Fi Network Configuration

SSID                         SSID-PSK                   Network SSID
Security                     WPA2-Personal
Password

Connect to this network
```



Wi-Fi Network Configuration

```
SSID                                 SSID-ENT                    EAP Authentication Method
Security                             WPA2-Enterprise
EAP Authentication Method            <TTLS>
▶ Enroll CA Cert
EAP Second Authentication Method     <MSCHAPv2>
Identity                             _
EAP Password

Connect to this network
```

Additional Information

LAN / WLAN Auto Switching BIOS setting is primarily used in OS, and it will not impact preboot Wi-Fi functionalities.
On Intel systems, use of AMT Wi-Fi and pre-boot Wi-Fi are mutually exclusive; only one can be enabled at one time. Intel systems have AMT enabled by default, so to use pre-boot Wi-Fi AMT must be disabled first.
For AMD systems, use of DASH (Desktop and mobile Architecture for System Hardware) and pre-boot Wi-Fi are mutually exclusive; only one can be enabled at one time. AMD systems have pre-boot Wi-Fi enabled by default, so to use DASH pre-boot Wi-Fi must be disabled first.

Fast Boot needs to be disabled to use pre-boot Wi-Fi. IPV4 and IPV6 boot entries will have 'Wi-Fi' string appended to those entries to differentiate between wired and Wi-Fi boot entries.

Comparison between wired PXE boot and Wi-Fi PXE boot	Wired PXE Boot **	Wi-Fi PXE Boot **
1	Boot PC	same
2	Enter F10	same
3	Promote Wired Network device to 1st boot option****	Promote Wi-Fi Network device to 1st boot option****
4	Connect RJ45 network cable	n/a
5	Save F10 settings & reboot	same
6	n/a	Enter F3
7	n/a	Enter / save Wi-Fi network settings & reboot***
8	System should connect to network and download pxe image*	same*

Appendix A – AMD Supported Wi-Fi modules

Below is a table of Wi-Fi modules (NICs or, Network Interface Cards) supported on AMD products.

Module Name	Comments	Notes
MediaTek RZ616 Wi-Fi 6E +BT 5.3 M.2 2230 AIM-T 160MHz PCI-e+USB WW WLAN	1. HP Pulsar Part Number: N22392-001 a. HP ZBook Power G10 A b. HP EliteBook 645, 655 G10 c. HP EliteBook 835, 845, 865 G10 d. HP ZBook Firefly 14 G10 A e. HP ProBook 445, 455 G10 f. HP Pro x360 435 G10	

Appendix B – Intel Supported Wi-Fi modules

Below is a table of Wi-Fi modules (NICs or, Network Interface Cards) supported on Intel products.

Module Name	Comments	Notes
Intel AX211 Wi-Fi 6E +BT 5.3 M.2 2230 160MHz CNVi WW WLAN	HP Pulsar Part Number: M53266-001	
Intel AX211 Wi-Fi 6E +BT 5.3 M.2 2230 vPro 160MHz CNVi WW WLAN	HP Pulsar Part Number: 53264-001	
Intel AX211 Wi-Fi 6E +BT 5.3 M.2 1216 160MHz CNVi WW WLAN	HP Pulsar Part Number: M53270-001	
Intel AX211 Wi-Fi 6E +BT 5.3 M.2 1216 vPro 160MHz CNVi WW WLAN	HP Pulsar Part Number: M53268-001	
Intel Wi-Fi 6 AX201 ax 2x2 MU-MIMO +BT 5 M.2 2230 non-vPro 160MHz MIPI+BRI WW 2Ant	HP Pulsar Part Number: L55990-002	
Intel Wi-Fi 6 AX201 ax 2x2 MU-MIMO +BT 5 M.2 1216 vPro 160MHz MIPI+BRI WW 2Ant	HP Pulsar Part Number: L59946-002	
Intel Wi-Fi 6 AX201 ax 2x2 MU-MIMO +BT 5 M.2 1216 non-vPro 160MHz MIPI+BRI WW 2Ant	HP Pulsar Part Number: L33637-002	
Intel Wi-Fi 6 AX201 ax 2x2 MU-MIMO +BT 5 M.2 1216 non-vPro 160MHz MIPI+BRI WW 2Ant	HP Pulsar Part Number: L33637-001	
Intel Wi-Fi 6 AX200 ax 2x2 MU-MIMO +BT 5 M.2 2230 non-vPro 160MHz PCI-e+USB WW 2Ant	HP Pulsar Part Number: L35108-001	
Intel 9560 ac 2x2 MU-MIMO +BT 5 M.2 2230 non-vPro 160MHz MIPI+BRI WW	HP Pulsar Part Number: 937262-001	
Intel 9260 ac 2x2 +BT 5 M.2 2230 non-vPro 160MHz PCI-e+USB WW	HP Pulsar Part Number: 920686-001	
Intel 8265 ac 2x2 +BT 4.2 M.2 non-vPro PCI-e+USB WW 2Ant	HP Pulsar Part Number: 851593-001	
Intel 9560 ac 2x2 MU-MIMO +BT 5 M.2 2230 non-vPro 160MHz MIPI+BRI BRZL	HP Pulsar Part Number: 937262-201	
Intel AX411 Wi-Fi 6E +BT 5.3 M.2 2230 160MHz CNVi WW WLAN	HP Pulsar Part Number: M53274-001	DT Only
Intel AX411 Wi-Fi 6E +BT 5.3 M.2 2230 vPro 160MHz CNVi WW WLAN	HP Pulsar Part Number: M53272-001	DT Only

Appendix C – HP AMD Platforms with Pre-Boot Wi-Fi Support

Below is a table of HP notebook (NB) platforms with AMD host processors that support UEFI (Pre-Boot) Wi-Fi. AMD supports Pre-Boot Wi-Fi on notebook platforms. (Desktops are not supported)

Cycle	Chipset/Platform HW	Marketing Name (Short Name)	Platform
2023	AMD Phoenix	HP ZBook Power 15.6 inch G10 A Mobile Workstation PC	NB
2023	AMD Phoenix	ZBook Firefly 14 G10 A	NB
2023	AMD Phoenix	HP EliteBook 835 13 inch G10 Notebook PC	NB
2023	AMD Phoenix	HP EliteBook 845 14 inch G10 Notebook PC	NB
2023	AMD Phoenix	HP EliteBook 865 16 inch G10 Notebook PC	NB
2023	AMD BCL-R	HP ProBook 445 14 inch G10 Notebook PC	NB
2023	AMD BCL-R	HP ProBook 455 15.6 inch G10 Notebook PC	NB
2023	AMD BCL-R	HP Pro x360 435 G10	NB
2023	AMD BCL-R	HP EliteBook 645 14 inch G10 Notebook PC	NB
2023	AMD BCL-R	HP EliteBook 655 15.6 inch G10 Notebook PC	NB

Appendix D – HP Intel Platforms with Pre-Boot Wi-Fi Support

Below is a table of HP platforms with Intel host processors that support UEFI (Pre-Boot) Wi-Fi.
Intel supports Pre-Boot Wi-Fi on notebook (NB) and desktop (DT) platforms.

Cycle	Chipset/Platform HW	Marketing Name (Short Name)	Platform (DT or NB)
2021	TGL-U	HP ProBook 430 G8 Notebook PC	NB
2021	TGL-U	HP ProBook 440 G8 Notebook PC	NB
2021	TGL-U	HP ProBook 450 G8 Notebook PC	NB
2021	TGL-U	HP ZHAN 66 Pro 14 G4 Notebook PC	NB
2021	TGL-U	HP ProBook 630 G8 Notebook PC	NB
2021	TGL-U	HP ProBook 640 G8 Notebook PC	NB
2021	TGL-U	HP ProBook 650 G8 Notebook PC	NB
2021	TGL-U	HP ZBook Power G8 Mobile Workstation	NB
2021	TGL-U	HP EliteBook x360 830 G8 Notebook PC	NB
2021	TGL-U	HP EliteBook 830 G8 Notebook PC	NB
2021	TGL-U	HP EliteBook 840 G8 Notebook PC	NB
2021	TGL-U	HP EliteBook 840 Aero G8 Notebook PC	NB
2021	TGL-U	HP EliteBook 850 G8 Notebook PC	NB
2021	TGL-U	HP EliteBook x360 1030 G8 Notebook PC	NB
2021	TGL-U	HP EliteBook x360 1040 G8 Notebook PC	NB
2021	TGL	HP Elite x2 G5 Tablet	NB
2021	TGL-U	HP Elite Dragonfly Max Notebook PC	NB
2021	TGL-U	HP Elite Dragonfly G2	NB
2021	TGL	HP ZBook Fury 17 inch G8 Mobile Workstation PC	NB
2021	TGL	HP ZBook Fury 15 inch G8 Mobile Workstation PC	NB
2021	TGL	HP ZBook Studio 15 inch G8 Mobile Workstation PC	NB
2021	TGL	HP Engage Go 10 Mobile System	NB
2021	RKL	HP ProDesk 600 G8 Desktop Mini PC	DT
2021	RKL	HP EliteDesk 800/880 G8 Tower PC	DT
2021	RKL	HP Z1 Entry Tower G8	DT
2021	RKL	HP EliteDesk 800 G8 Small Form Factor PC	DT
2021	RKL	HP EliteDesk 800 G8 Desktop Mini PC	DT
2021	RKL	HP ProDesk 600 G8 SFF	DT
2021	RKL	HP ProDesk 600/680 G8 Microtower PC	DT
2021	RKL	HP Z2 G8 Tower (WKS)	DT
2021	RKL	HP Z2 G8 SFF (WKS)	DT
2021	RKL	HP ProDesk 600 G8 Desktop Mini PC	DT
2021	RKL	HP EliteOne 800 G8 24/27 All-in-One PC	DT
2021	RKL	HP EliteDesk 800/880 G8 Tower PC	DT
2021	RKL	HP EliteDesk 800 G8 Small Form Factor PC	DT
2021	RKL	HP EliteDesk 800 G8 Desktop Mini PC	DT
2021	RKL	HP ProDesk 600 G8 SFF	DT
2021	CML	HP EliteOne 1000 G2 All-in-One	DT

2022 / 2023 / 2024	ADL / RPL / RPL-R	HP Z2 G9 Tower Workstation Desktop PC	DT
2022 / 2023 / 2024	ADL / RPL / RPL-R	HP Z2 G9 SFF Workstation Desktop PC	DT
2022 / 2023 / 2024	ADL / RPL / RPL-R	HP Z2 G9 Mini Workstation Desktop PC	DT
2022 / 2023 / 2024	ADL / RPL / RPL-R	HP Z1 G9 Tower Desktop PC	DT
2022	ADL-P	HP EliteBook 830 G9 Notebook PC	NB
2022	ADL-P	HP EliteBook 840 G9 Notebook PC	NB
2022	ADL-P	HP EliteBook 1040 G9 Notebook PC	NB
2022	ADL-P	HP EliteBook 860 G9 Notebook PC	NB
2022	ADL-P	HP Elite x360 1040 G9 2-in-1 Notebook PC	NB
2022	ADL-P	HP Elite x360 830 G9 2-in-1 Notebook PC	NB
2022	ADL-P	HP ZBook Studio 16 inch Mobile Workstation G9	NB
2022	ADL-S BGA	HP ZBook Fury G9 Mobile Workstation PC	NB
2022	ADL-P	HP ZBook Power 15.6 inch G9 Mobile Workstation PC	NB
2022	ADL-P	HP ZBook Firefly 16 inch G9 (DIS)	NB
2022	ADL-P	HP ZBook Firefly 16 inch G9	NB
2022	ADL-P	HP ZBook Firefly 14 inch G9 (DIS)	NB
2022	ADL-P	HP ZBook Firefly 14 inch G9	NB
2022	ADL-P	HP Elite Dragonfly Folio 13.5 inch G3 2-in-1-Notebook PC	NB
2022	ADL-P	HP Elite Dragonfly 13.5” G3 Notebook PC	NB
2022	ADL-P	HP ProBook 440 G9 Notebook PC	NB
2022	ADL-P	HP ProBook 450 G9 Notebook PC	NB
2022	ADL-P	HP EliteBook 630 G9 Notebook PC	NB
2022	ADL-P	HP EliteBook 640 G9 Notebook PC	NB
2022	ADL-P	HP EliteBook 650 G9 Notebook PC	NB
2022	ADL-P	HP ZHAN 66 Pro 14 G5 Notebook PC	NB
2022	ADL-P	HP Pro mt440 G3 Mobile Thin Client	NB
2022	TGL	HP Engage Go Mobile System G3	NB
2022	ADL/RPL	HP Elite Tower 600/680 G9 Desktop PC	DT
2022	ADL/RPL	HP Elite SFF 600 G9 Desktop PC	DT
2022	ADL/RPL	HP Elite Mini 600 G9 Desktop PC	DT
2022	ADL/RPL	HP Elite Mini 800 G9 Desktop PC	DT
2022	ADL/RPL	HP Elite SFF 800 G9 Desktop PC	DT
2022	ADL/RPL	HP Elite Tower 800/880 G9 Desktop PC	DT
2022	ADL/RPL	HP EliteOne 840, 870 G9 All-in-One Desktop PC	DT
2022	ADL/RPL	HP Presence 23.8”, 27” All-in-One Conferencing PC	DT
2022	ADL/RPL	HP Mini Conferencing PC with Zoom Rooms	DT
2023	RPL	HP ZBook Studio 16 inch Mobile Workstation G10	NB
2023	RPL	HP ZBook Fury G10 Mobile Workstation PC	NB
2023	RPL	HP ZBook Firefly 14 inch G10 Mobile Workstation PC	NB
2023	RPL	HP ZBook Firefly 16 inch G10 Mobile Workstation PC	NB

2023	RPL	HP ZBook Firefly 14 inch G10 Mobile Workstation PC	NB
2023	RPL	HP ZBook Firefly 16 inch G10 Mobile Workstation PC	NB
2023	RPL	HP Elite Dragonfly 13.5 inch G4 Notebook PC	NB
2023	RPL	HP EliteBook 830 13 inch G10 Notebook PC	NB
2023	RPL	HP EliteBook 840 14 inch G10 Notebook PC	NB
2023	RPL	HP EliteBook 860 16 inch G10 Notebook PC	NB
2023	RPL	HP Elite x360 830 13 inch G10 2-in-1 Notebook PC	NB
2023	RPL	HP EliteBook 1040 14 inch G10 Notebook PC	NB
2023	RPL	HP Elite x360 1040 14 inch G10 2-in-1 Notebook PC	NB
2023	RPL	HP ProBook 440 14 inch G10 Notebook PC	NB
2023	RPL	HP ProBook 450 15.6 inch G10 Notebook PC	NB
2023	RPL	HP EliteBook 630 13 inch G10 Notebook PC	NB
2023	RPL	HP EliteBook 640 14 inch G10 Notebook PC	NB
2023	RPL	HP EliteBook 650 15.6 inch G10 Notebook PC	NB

Sign up for updates **hp.com/go/getupdated**

Intel and vPro are trademarks of Intel Corporation in the U.S. and other countries. AMD is a trademark of Advanced Micro Devices, Inc.. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Wi-Fi® is a registered trademark of Wi-Fi Alliance®.