Technical White Paper

Preboot Wireless Networking on HP Business PCs

January 2020

Table of contents

Disclaimer

The information contained within this whitepaper, including URL, other web site references, and other specification documents are subject to change without notice and are provided for informational purposes only. No licenses with respect to any intellectual property are being granted, expressly or impliedly, by the disclosure of the information contained in this document. Furthermore, neither HP Inc. nor any of its subsidiaries makes any warranties of any nature regarding the use of the information contained within this document, and thus the entire risk, if any, resulting from the use of information within this document is the sole responsibility of the user. In addition, the names of the technologies, actual companies, and products mentioned within this document may be trademarks of their respective owners. Complying with all applicable copyright and trademark laws is the sole responsibility of the user of this document. Without limiting any rights under copyright, 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. HP Inc. or its subsidiaries may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. Except where expressly provided in any written license from HP Inc. or its subsidiaries, the furnishing of this document, or any ideas contained within, does not grant any license to these ideas, patents, trademarks, copyrights, or other intellectual property.

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.

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
- Enterprise EAP-TLS
- Enterprise EAP-PEAP
- Enterprise EAP-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.

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 www.hp.com.

BIOS updates can be downloaded from www.hp.com

Preboot Wireless Networking - Base Requirements

WLAN Adapter

The platform must integrate a supported WLAN adapter in order to support preboot wireless networking. Supported WLAN adapters are:

- vPro™ - Intel® Wi-Fi 6 AX201 + BT5 (802.11ax 2x2)
- Non-vPro™ - Intel® Wi-Fi 6 AX201 + BT5 (802.11ax 2x2)
- vPro™ - Intel® Wireless-AC 9560 (802.11ac 2x2 Wi-Fi + BT5)
- Non-vPro™ - Intel ®Wireless-AC 9560 (802.11ac 2x2 Wi-Fi + BT5)

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 EAP-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 **EAP-PEAP** • Identity • CaCert • Password

	• AutoConnect • Scan Anyway (optional)
Enterprise EAP-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 **EAP-TTLS** • Identity • CaCert • Password • AutoConnect • Scan Anyway (optional)
Enterprise EAP-TLS	Just like PEAP and TTLS, EAP-TLS authenticates both client and server. However, EAP-TLS 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 settiungs are required: • SSID • Type must be **EAP-TLS** • 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 • EAP-TLS • EAP-PEAP • EAP-TTLS
AutoConnect	O	O	O	O	Connect automatically during pre-boot if the network is in range (and the priority allows).	N	Enumerated List	• Yes • No (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	• Yes • No (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 a 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": "No", "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.

Personal (PSK)

The fields for 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": "No", "Password":"P@55w0rd" }

Get/Read

Preboot Wi-Fi Profile 1

{ "SSID": "TestPSK", "Type": "Personal", "AutoConnect": "No", "Password":"******" }

EAP-TLS

The fields for enterprise EAP-TLS 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 EAP-TLS 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": "EAP-TLS", "AutoConnect": "Yes", "CaCert": "ABC…123", "ClientCert": "XYZ…321", "ClientPvtKey": "555…777", "ClientPvtKeyPassword": "P@ssw0rd", "Identity": "anonymous" }

Get/Read

Preboot Wi-Fi Profile 1

{ "SSID": "TestTLS", "Type": "EAP-TLS", "AutoConnect": "Yes", "CaCert": "ABC…123", "ClientCert": "XYZ…321", "ClientPvtKey": "******", "ClientPvtKeyPassword": "******", "Identity": "anonymous" }

EAP-PEAP & EAP-TTLS

The fields for enterprise EAP-PEAP and EAP-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 EAP-PEAP and Preboot Wi-Fi Profile 2 for an EAP-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": "EAP-PEAP", "AutoConnect": "Yes", "CaCert": "abc…123", "Identity": "SampleID", "Password":"" }

Preboot Wi-Fi Profile 2

{ "SSID": "TestTTLS", "Type": "EAP-TTLS", "AutoConnect": "No", "CaCert":"AAA…333",
"Identity":"SampleID1, "Password":"P@s5Word" }

Get/Read
Preboot Wi-Fi Profile 1
{ "SSID": "TestPEAP", "Type": "EAP-PEAP", "AutoConnect": "Yes", "CaCert": "abc…123",
"Identity": "SampleID", "Password":"******" }
Preboot Wi-Fi Profile 2
{ "SSID": "TestTTLS", "Type": "EAP-TTLS", "AutoConnect": "No", "CaCert":"AAA…333",
"Identity":"SampleID1, "Password":"******" }

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": "EAP-TLS", "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": "EAP-TLS", "AutoConnect": "No", "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": "EAP-TLS", "AutoConnect": "No", "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": "EAP-TLS", "AutoConnect": "No", "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": "EAP-TLS", "AutoConnect": "No", "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": "EAP-TLS", "AutoConnect": "No", "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 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	• Yes • No (default)

JSON Preboot Wi-Fi® Enrolment 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

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

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

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.

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 6t 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®.

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-Fi®Clear()
{
    $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 Profile On Demand", "", "")
    $bios.SetBiosSetting("Preboot Wi-Fi Master Auto Connect", "", "")
}

Wi-Fi®Clear
Write-Host "Cleared out six (6) preboot Wi-Fi® profiles and the on demand profile"
```

Setting Wireless Settings

Once wireless profile settings are cleared, they must be set to something if they are to be used at all. The following demonstrates a sample PowerShell script form that can accomplish this. Please note that this sample is for the purpose of *form* only. The certificate and private key values are actually base64-encoded lorem ipsum text. This enrolment won't succeed. The certificate and private key values must be replaced with the actual certificate and private key contents from a working infrastructure.

```
#
# (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 Wi-Fi®SetEapTlsProfile()
{
    $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": "No", ' +
            '"ScanAnyway": "Yes", "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, "")
}

Wi-Fi®SetEapTlsProfile
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
```

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 Wi-Fi®SetPeapProfile()
{
    $CaCert  = Get-Content -Path $CaCertPath -Encoding Byte
    $CaCert = [Convert]::ToBase64String($CaCert)

    $json = '{ "SSID": "' + $SSID + '", "Type":  "EAP-PEAP", "AutoConnect": "No", ' +
            '"ScanAnyway": "Yes", "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, "")
}


Wi-Fi®SetPeapProfile
Write-Host "Configured Preboot Wi-Fi Profile 2"
```

TTLS Example

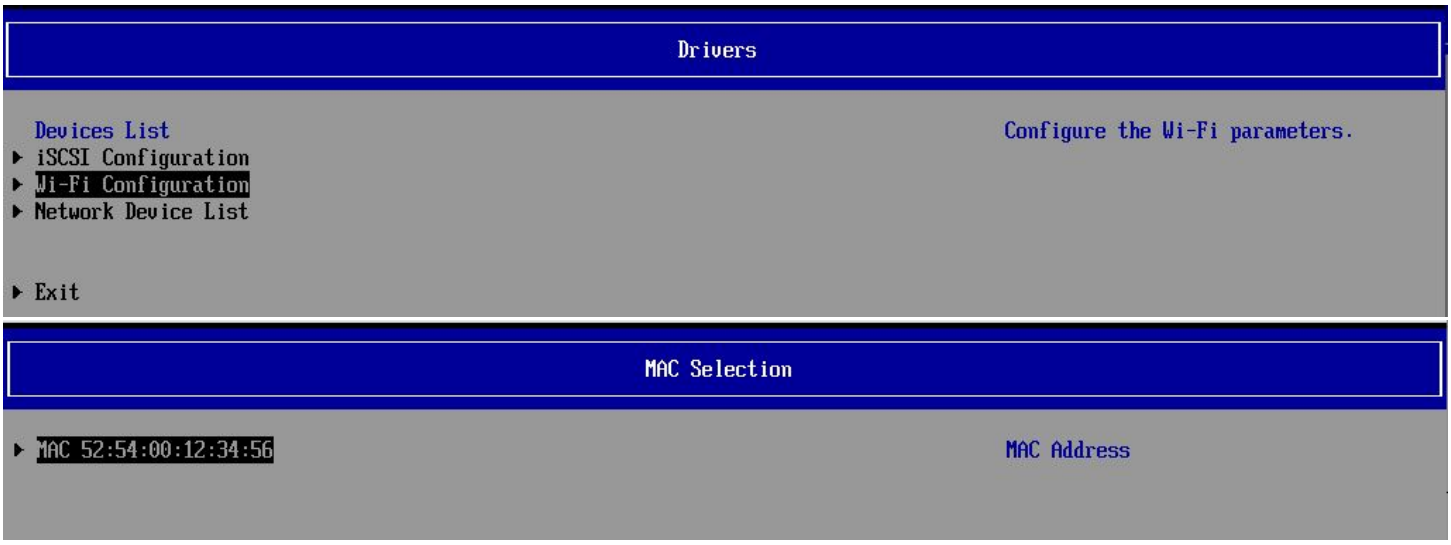
```
#
# (c) Copyright 2019 HP Development Company, L.P.
# This software and associated documentation (if any) is furnished under a license
# and may only be used or copied in accordance with the terms of the license.
# Except as permitted by such license, no part of this software or documentation may
# 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 Wi-Fi®SetTtlsProfile()
{
    $CaCert  = Get-Content -Path $CaCertPath -Encoding Byte
    $CaCert = [Convert]::ToBase64String($CaCert)

    $json = '{ "SSID": "' + $SSID + '", "Type":  "EAP-TTLS", "AutoConnect": "No", ' +
            '"ScanAnyway": "Yes", "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, "")
}


Wi-Fi®SetTtlsProfile
Write-Host "Configured Preboot Wi-Fi Profile 3"
```

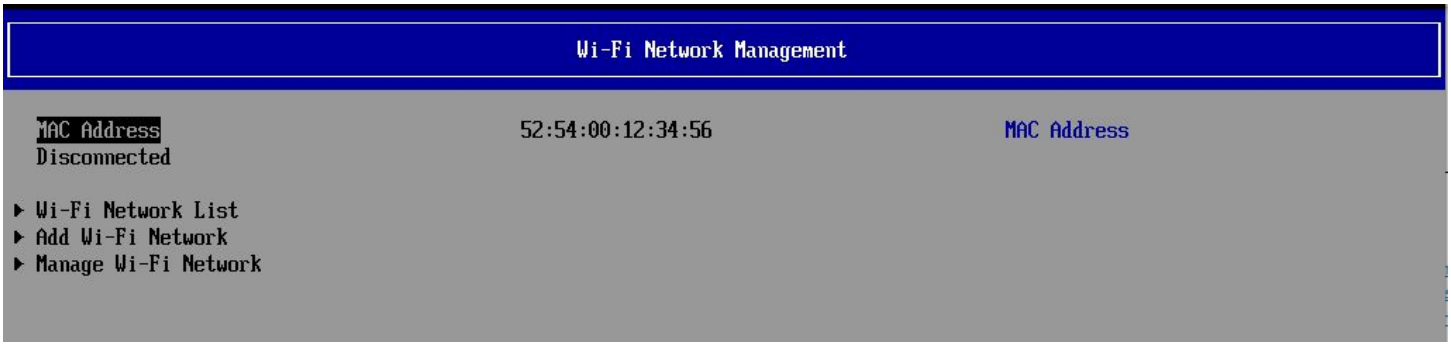
WPA2-Personal Example

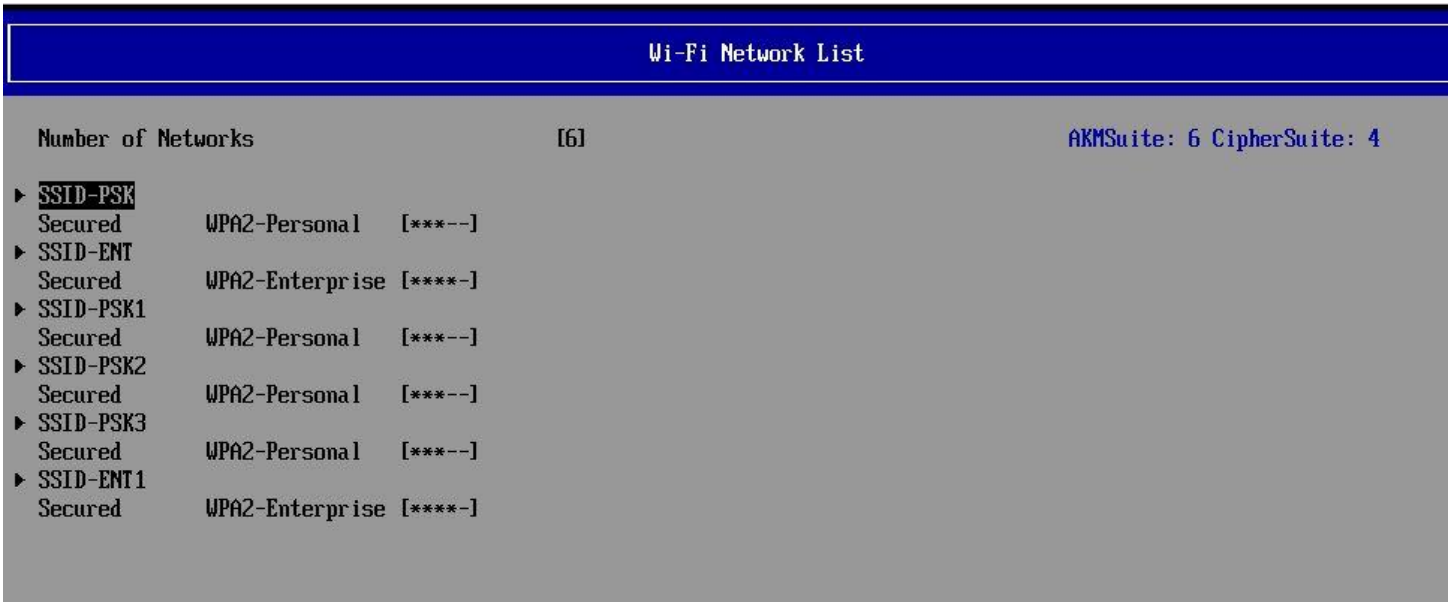
```
#
# (c) Copyright 2019 HP Development Company, L.P.
# This software and associated documentation (if any) is furnished under a license
# and may only be used or copied in accordance with the terms of the license.
# Except as permitted by such license, no part of this software or documentation may
# 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 Wi-Fi®SetPersonalProfile()
{
    $json = '{ "SSID": "' + $SSID + '", "Type":  "Personal", "AutoConnect": "No", ' +
            '"ScanAnyway": "Yes","', "Password": "' + $Password + '" } '

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


Wi-Fi®SetPersonalProfile
Write-Host "Configured Preboot Wi-Fi Profile 4"
```

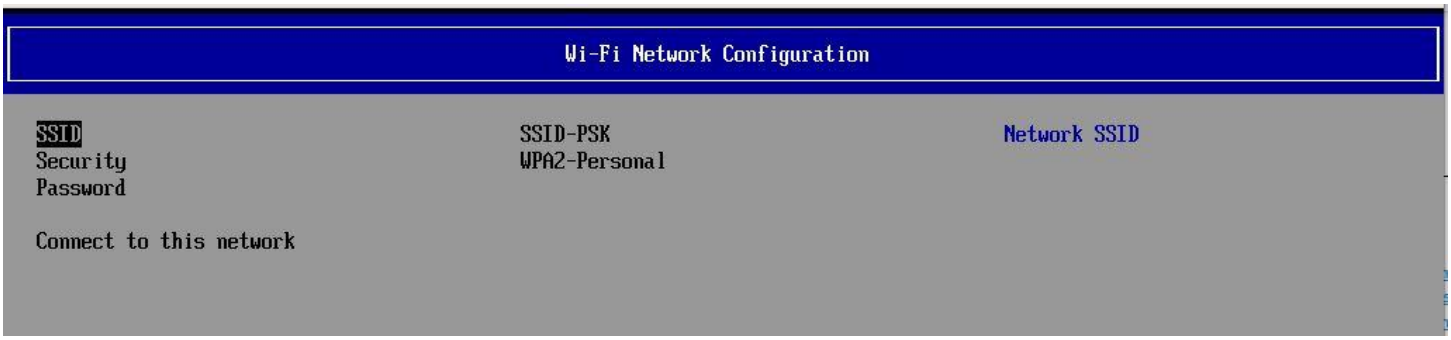
Preboot Wi-Fi® Enrollment via BIOS

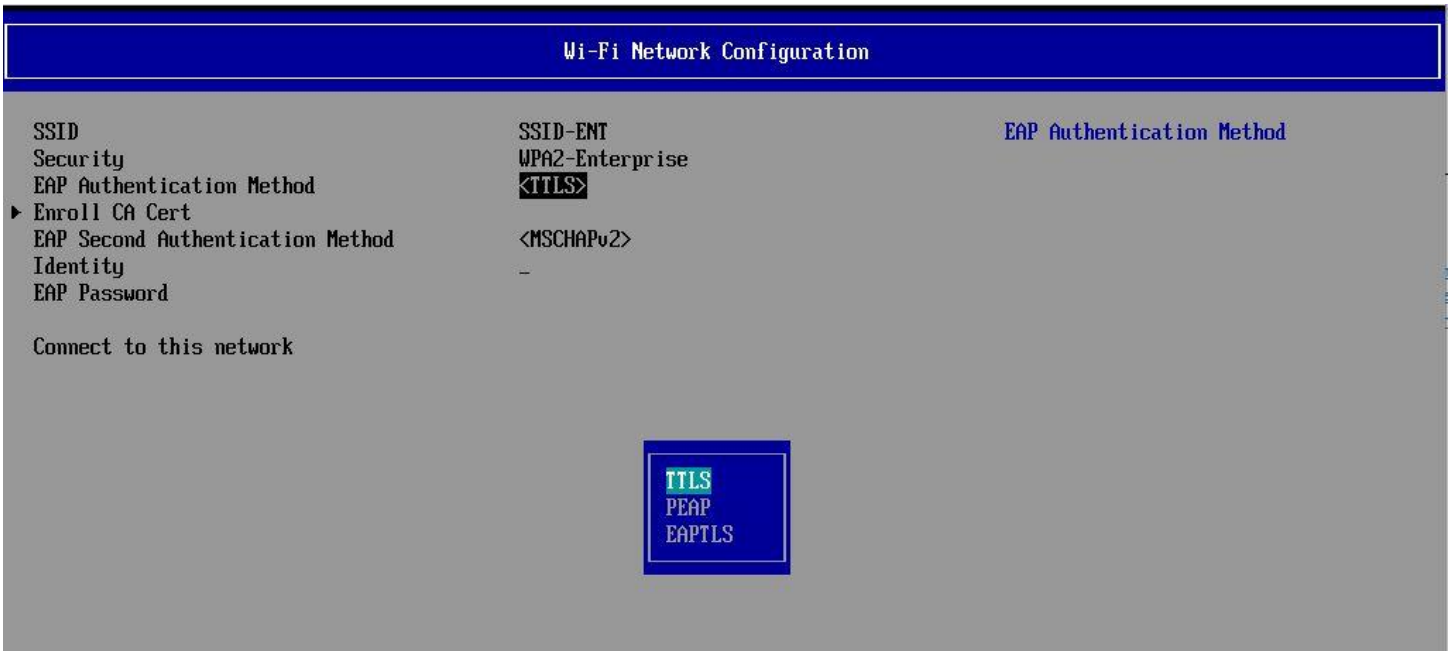
This section describes steps to configure preboot Wi-Fi® thru BIOS setup which is accessible during system boot-up by using the ‘F3’ function key. On entering F3 menu, ‘Wi-Fi Configuration’ link will provide a link to MAC address of Wi-Fi® controller which in turn provides the connection status, list 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 varaible.




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



Additional Information

LAN / WLAN Auto Switching BIOS setting is primarily used in OS and it will not impact preboot Wi-Fi functionalities.

Use of AMT WI-FI® and pre-boot WI-FI® are mutually exclusive; only one can be enabled.

Fast Boot needs to be disabled to use pre-boot WI-FI®.Wi-Fi®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*

Notes:

* step is not OS-specific (Linux, WinPE, WinRE, etc.) just indicates transfer occurs/completes successfully
** assumes proper DHCP & PXE Server configuration for wired & wireless network access
*** Must configure with Autoconnect enabled

**** PXE boot support applies to both IPv4 and IPv6

Disclaimer

The information contained within this whitepaper, including URL, other web site references, and other specification documents are subject to change without notice and are provided for informational purposes only. No licenses with respect to any intellectual property are being granted, expressly or impliedly, by the disclosure of the information contained in this document. Furthermore, neither HP Inc. nor any of its subsidiaries makes any warranties of any nature regarding the use of the information contained within this document, and thus the entire risk, if any, resulting from the use of information within this document is the sole responsibility of the user. In addition, the names of the technologies, actual companies, and products mentioned within this document may be trademarks of their respective owners. Complying with all applicable copyright and trademark laws is the sole responsibility of the user of this document. Without limiting any rights under copyright, 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. HP Inc. or its subsidiaries may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. Except where expressly provided in any written license from HP Inc. or its subsidiaries, the furnishing of this document, or any ideas contained within, does not grant any license to these ideas, patents, trademarks, copyrights, or other intellectual property.

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