



Hewlett Packard
Enterprise

HPE ProLiant Gen10 サーバーおよび HPE Synergy 用 UEFI 展開ガイド

摘要

このガイドでは、UEFI (Unified Extensible Firmware Interface) ベースの ProLiant Gen10 サーバーおよび Synergy コンピュートモジュールを起動する各種展開方法をどのように使用するかについて詳しく説明します。このガイドは、サーバーおよびストレージシステムのインストール、管理、トラブルシューティングの担当者を対象とし、コンピューター機器の保守の資格があり、高電圧製品の危険性について理解していることを前提としています。

部品番号: 881329-193
発行: 2019 年 3 月
版数: 1

ご注意

本書の内容は、将来予告なしに変更されることがあります。Hewlett Packard Enterprise 製品およびサービスに対する保証については、当該製品およびサービスの保証規定書に記載されています。本書のいかなる内容も、新たな保証を追加するものではありません。本書の内容につきましては万全を期しておりますが、本書中の技術的あるいは校正上の誤り、脱落に対して、責任を負いかねますのでご了承ください。

本書で取り扱っているコンピューターソフトウェアは秘密情報であり、その保有、使用、または複製には、Hewlett Packard Enterprise から使用許諾を得る必要があります。FAR 12.211 および 12.212 に従って、商用コンピューターソフトウェア、コンピューターソフトウェアドキュメンテーション、および商用製品の技術データ (Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items) は、ベンダー標準の商用使用許諾のもとで、米国政府に使用許諾が付与されます。

他社の Web サイトへのリンクは、Hewlett Packard Enterprise の Web サイトの外に移動します。Hewlett Packard Enterprise は、Hewlett Packard Enterprise の Web サイト以外の情報を管理する権限を持たず、また責任を負いません。

商標

Microsoft[®] および Windows[®] は、米国および/またはその他の国における Microsoft Corporation の登録商標または商標です。

Intel[®]、インテル、Itanium[®]、Pentium[®]、Intel Inside[®]、および Intel Inside ロゴは、インテルコーポレーションまたはその子会社のアメリカ合衆国およびその他の国における商標または登録商標です。

UEFI[®] は UEFI Forum, Inc. の登録商標です。

Linux[®] は、Linus Torvalds の米国およびその他の国における登録商標です。

目次

はじめに.....	6
UEFI の展開方法	6
UEFI ブート設定の構成.....	7
ブート設定.....	7
ブートモード比較 : UEFI およびレガシー BIOS	7
ブートモードの選択	7
UEFI 最適化ブートの有効化または無効化.....	8
UEFI ブート順序リストの変更	8
レガシー BIOS ブート順序リストの変更.....	9
UEFI ブート順序リストへのブートオプションの追加.....	9
ワнтаイムブートメニューオプション.....	10
ワнтаイムブートのオプションの選択.....	10
USB ブート.....	12
USB ブートオプション.....	12
汎用 USB ブート.....	12
Windows をインストールするためのブート可能な UEFI USB フラッシュドライブの作成	12
HPE Dual MicroSD デバイスからのブート.....	13
UEFI モードとレガシー BIOS モードの両方のデュアルブート USB キーの作成	13
PXE および iPXE ブート.....	14
PXE および iPXE 操作.....	14
PXE マルチキャストブート.....	14
PXE 構成	14
Linux 環境での UEFI ベースクライアント用の PXE サーバーの構成	14
ブートローダーの構成	15
Windows 環境でのレガシー BIOS ベースクライアント用の PXE サーバーの構成.....	16
Windows 環境での UEFI ベースクライアント用の PXE サーバーの構成	16
システム要件	16
Windows サーバーの構成	16
UEFI ネットワークブート用の VLAN の構成	16
システムユーティリティのネットワークオプションが提供するグローバルな VLAN 構成メニューの使用	17
特定の NIC アダプターが提供する構成メニューの使用	17
iPXE の構成.....	18
iPXE の構成	18
iPXE ファイルのダウンロード.....	18
iPXE のカスタマイズ.....	18
TFTP の構成.....	19
HTTP の構成.....	19
DHCP の構成.....	19
チェーンロードの構成	19
PXE または iPXE の起動.....	20

iSCSI ブート	21
iSCSI ソフトウェアイニシエーターの構成.....	21
iSCSI ソフトウェアイニシエーターの構成と iSCSI のブート	21
iSCSI ブート用のデバイスを選択し、ネットワークブート可能にする	21
iSCSI イニシエーター名の追加.....	22
iSCSI ブート試行の追加.....	22
iSCSI 接続の確認	23
iSCSI イニシエーターのブート順序の変更.....	23
HTTP ブート	24
ブート URL を指定するネットワークサービスの構成.....	24
必要なパッケージのインストール.....	25
ネットワークインターフェイスのセットアップ.....	25
DHCP サービスの構成.....	25
ルーター広告デーモンの構成.....	27
IPv4 および IPv6 用 DNS サービスの構成.....	28
HTTP サービスの構成 (apache2)	29
HTTP 基本認証の構成.....	30
apache2 での HTTPS (TLS) のセットアップ.....	30
クライアントマシンへの証明書のインストール.....	31
サポートされるブートローダーを使用した HTTP (S) ブートの実行.....	32
URL からの起動の構成.....	34
システムユーティリティによる URL からの起動の構成.....	34
内蔵 UEFI シェルの sysconfig コマンドによる URL からの起動の構成	34
RESTful インターフェイスツールによる URL からの起動の構成.....	34
プリブートネットワーク設定構成	36
システムユーティリティによるプリブートネットワーク設定の構成.....	36
プリブートネットワーク設定.....	36
内蔵 UEFI シェルの sysconfig コマンドによるプリブートネットワーク設定の構成	37
RESTful インターフェイスツールによるプリブートネットワーク設定の構成.....	38
内蔵 UEFI シェルのブート	39
ネットワーク展開用の内蔵 UEFI シェルコマンド	39
標準形式の出力 (SFO)	39
シェルスクリプトの起動	40
システムユーティリティの UEFI シェルスクリプト自動起動.....	40
シェルスクリプトの手動開始	40
シェルスクリプトの編集	40
ネットワークから起動スクリプトを読み取る内蔵 UEFI シェルの構成	40
起動スクリプトを読み取る内蔵 UEFI シェルの構成 (システムユーティリティ) ...	41
起動スクリプトを読み取る内蔵 UEFI シェルの構成 (sysconfig コマンド)	41
起動スクリプトを読み取る内蔵 UEFI シェルの構成 (RESTful インターフェイス ツール)	42
RAM ディスクの作成	42
展開とスクリプティング	42
内蔵 UEFI シェルのブートの構成 : サンプルの展開ソリューション	42
サンプル起動スクリプト	43
UEFI モードの FC/FCoE SAN ブート	46

FC/FCoE SAN ブート構成.....	46
FC/FCoE SAN からの起動.....	46
UEFI モードでのローカル HDD のブート	47
HDD ブート構成.....	47
内蔵 SATA のサポート	47
内蔵 SATA のサポートの有効化	47
トラブルシューティング.....	48
ネットワークブート URL のファイルをダウンロードできない	48
UEFI シェルスクリプトから展開できない	49
Web サイト.....	50
サポートと他のリソース.....	51
Hewlett Packard Enterprise サポートへのアクセス.....	51
アップデートへのアクセス.....	51
カスタマーセルフリペア (CSR)	52
リモートサポート (HPE 通報サービス)	52
保証情報.....	52
規定に関する情報.....	52
ドキュメントに関するご意見、ご指摘.....	53

はじめに

HPE ProLiant サーバーおよび Synergy コンピュートモジュールは、UEFI をベースにした業界標準のシステムファームウェアをサポートします。UEFI サーバーの展開オプションでは、拡張組み込みネットワークスタックと内蔵 UEFI シェル環境を利用することで、高い信頼性、セキュリティ、スケーラビリティを提供し、既存のネットワークベースの展開ソリューションよりも速いダウンロード速度を実現します。

UEFI の展開方法

次のような UEFI 展開方法があります。

- ・ **USB** - 任意の .efi タイプの UEFI アプリケーションを起動するブート可能な UEFI USB フラッシュドライブを作成、OS ランチャーを \efi\BOOT\BOOTX64.EFI に保存することによって起動、またはオペレーティングシステムのインストールのストレージとして USB フラッシュドライブのパーティションを使用します。
- ・ **PXE (Preboot Execution Environment) および iPXE** - PXE は、サーバーのネットワーク機能を使用して IP アドレスを取得し、サーバーからファイルをダウンロードして起動します。iPXE は HTTP/HTTPS ブートおよびその他のオプションを提供することで、PXE のブート方法を拡張します。
- ・ **iSCSI (Internet Small Computer System Interface)** - LUN 上、または iSCSI ストレージエリアネットワーク (SAN) に格納されているイメージから起動します。
- ・ **HTTP ブートおよび URL からの起動** - PXE ブートに使用される UDP 接続より信頼性の高い TCP 接続経由の HTTP/HTTPS を使用して、ファイルをネットワーク上 (URL) からダウンロードして起動します。
- ・ **内蔵 UEFI シェル** - より高速なネットワークベースのオペレーティングシステムを展開する自動スクリプトソリューションを提供する方法として、内蔵 UEFI シェルから起動します。
- ・ **SAN (FC および FCoE)** - ファイバーチャネルストレージエリアネットワーク (SAN) 上にあるディスクのオペレーティングシステムイメージから起動します。
- ・ **ローカル HDD** - ローカルハードドライブから起動します。

UEFI ブート設定の構成

ブート設定

iLO Web インターフェイス、UEFI システムユーティリティメニュー、または RESTful インターフェイスツールを使用すると、構成したブートオプションからのサーバーのブート試行順序など、ブート設定を表示して変更することができます。

以下のセクションでは、システムユーティリティメニューを使用してブート設定を構成する方法を説明します。

詳しくは、以下を参照してください。

HPE iLO ユーザーガイド : <http://www.hpe.com/info/ilo/docs>

RESTful インターフェイスツールのドキュメント : <http://www.hpe.com/info/restfulinterface/docs>

ブートモード比較 : UEFI およびレガシー BIOS

このサーバーには、2つのブートモード構成があります。UEFI モード（デフォルトのブートモード）とレガシー BIOS モードの2つのブートモード構成が用意されています。このガイドに記載されているブートオプションの一部は、特定のブートモードの選択を必要とします。

- ・ **UEFI モード** - デフォルトのブートモードです。UEFI と互換性のあるオペレーティングシステムを起動するようにサーバーを構成するモードで、セキュアブートや、USB、IPv6 PXE、iSCSI、HTTP/HTTPS FTP URL、内蔵ユーザーパーティション、HPE Smart アレイ S100i Gen10 SW RAID デバイスのブートなど、特定の UEFI システムユーティリティオプションを使用するために必要です。
- ・ **レガシー BIOS モード** - このブートモードでは、従来の BIOS ブート環境をエミュレートするようにサーバーが構成されます。状況によっては、レガシー BIOS モードでの操作が必要になる場合があります。たとえば、レガシーブートモードを使用してインストールしたカスタム OS イメージや、レガシー BIOS システムを使用して作成したカスタム OS イメージを起動する場合があります。

ブートモードの選択

このサーバーには、2つのブートモード構成があります。UEFI モードおよびレガシー BIOS モードです。一部のブートオプションでは、特定のブートモードを選択することが必要です。デフォルトでは、ブートモードは **UEFI モード** に設定されています。次の特定のオプションを使用するには、システムを **UEFI モード** で起動する必要があります。

- ・ セキュアブート、UEFI 最適化ブート、汎用 USB ブート、IPv6 PXE ブート、iSCSI ブート、および URL からのブート
- ・ ファイバーチャネル/FCoE スキャンポリシー

注記: 使用するブートモードはオペレーティングシステムのインストールと一致しなければなりません。そうでない場合、ブートモードを変更するとサーバーがインストール済みのオペレーティングシステムで起動する機能に影響を与える場合があります。

前提条件

UEFI モードで起動する場合は、**UEFI 最適化ブート**を有効なままにしてください。

手順

1. システムユーティリティ画面で、**システム構成 > BIOS/プラットフォーム構成 (RBSU) > ブートオプション > ブートモード**を選択します。
2. 設定を選択します。
 - ・ **UEFI モード** (デフォルト) —UEFI 互換性のあるオペレーティングシステムで起動するようシステムを設定します。
 - ・ **レガシー BIOS モード** - レガシー BIOS 互換モードで従来のオペレーティングシステムに起動するようシステムを構成します。
3. 設定を保存します。
4. サーバーを再起動します。

UEFI 最適化ブートの有効化または無効化

UEFI 最適化ブートを使用して、システム BIOS をネイティブの UEFI グラフィックドライバーを使用して起動するかどうかを制御します。UEFI 最適化ブートは、デフォルトで有効になっています。Windows Server 2008、Windows Server 2008 R2 または Windows 7 を使用している場合にのみ、UEFI 最適化ブートを無効にします。

前提条件

- ・ UEFI 最適化ブートを有効化するときは、ブートモードを UEFI モードに設定する必要があります。UEFI モードは、デフォルトのブートモードです。
- ・ 次の目的で、UEFI 最適化ブートを有効にする必要があります。
 - セキュアブートを有効にして使用する。
 - VMware ESXi を動作させる。

手順

1. システムユーティリティ画面で、**システム構成 > BIOS/プラットフォーム構成 (RBSU) > ブートオプション > UEFI 最適化ブート**を選択します。
2. オプションを選択します。
 - ・ **有効** - UEFI モードに設定されている場合、ネイティブの UEFI グラフィックドライバーを使用して起動するようにシステム BIOS を構成します。
 - ・ **無効** - INT10 レガシービデオ拡張 ROM を使用して起動するようにシステム BIOS を構成します。ご使用のオペレーティングシステムとして、Windows Server 2008、Windows Server 2008 R2 または Windows 7 を使用している場合、この設定が必要です。
3. 設定を保存します。
4. サーバーを再起動します。

UEFI ブート順序リストの変更

UEFI ブート順序オプションを使用して、UEFI ブート順序リスト内のエントリーのブート順序を変更します。

手順

1. システムユーティリティ画面で、システム構成 > BIOS/プラットフォーム構成 (RBSU) > ブートオプション > UEFI ブート設定 > UEFI ブート設定 > UEFI ブート順序を選択します。
2. ブート順序リスト内を移動するには、ポインティングデバイスまたは矢印キーを使用します。
3. エントリーを選択し、そのエントリーのリスト内での順序を変更します。
 - ・ ブートリスト内でエントリーを上に移すには、+キーを押すか、またはエントリーをドラッグアンドドロップします。
 - ・ ブートリスト内でエントリーを下に移すには、-キーを押すか、またはエントリーをドラッグアンドドロップします。
4. 変更を保存します。

レガシー BIOS ブート順序リストの変更

前提条件

ブートモードがレガシー BIOS モードに設定されている。

手順

1. システムユーティリティ画面で、システム構成 > BIOS/プラットフォーム構成 (RBSU) > ブートオプション > レガシー BIOS ブート順序を選択します。
2. ブート順序リスト内を移動するには、ポインティングデバイスまたは矢印キーを使用します。
3. エントリーを選択し、そのエントリーのリスト内での順序を変更します。
 - ・ ブートリスト内でエントリーを上に移すには、+キーを押すか、またはエントリーをドラッグアンドドロップします。
 - ・ ブートリスト内でエントリーを下に移すには、-キーを押すか、またはエントリーをドラッグアンドドロップします。
4. 変更を保存します。

UEFI ブート順序リストへのブートオプションの追加

ブートオプションを追加を使用して、拡張子.EFI を持つ x64 UEFI アプリケーション (OS ブートローダーやその他の UEFI アプリケーションなど) を選択し、新しい UEFI ブートオプションとして追加できます。

新しいブートオプションは、UEFI ブート順序リストの最後に追加されます。ファイルを選択すると、ブートメニューに表示するブートオプションの説明と、.EFI アプリケーションに渡すデータ (オプション) を入力するよう求めるプロンプトが表示されます。

手順

1. FAT16/FAT32 パーティションを持つメディアを接続します。
2. システムユーティリティ画面で、システム構成 > BIOS/プラットフォーム構成 (RBSU) > ブートオプション > UEFI ブート設定 > ブートオプションを追加を選択します。
3. リスト内の .EFI アプリケーションを選択して **Enter** キーを押します。
4. 必要に応じて、**Enter** キーを押してメニューオプションをドリルダウンします。

5. ブートオプションの説明とオプションのデータを入力し、**Enter** キーを押します。

UEFI ブート順序リストに新しいブートオプションが表示されます。

6. 変更をコミットして終了を選択します。

ワンタイムブートメニューオプション

ワンタイムブートメニューを使用して、ワンタイムブートオーバーライドに UEFI ブートオプションを選択できます。

注記: このオプションを選択しても、事前定義済みのブート順序の設定は選択したオプションにより変更されません。iLO5 リモートコンソールで USB キーまたは仮想メディアを使用する場合、ワンタイムブートメニューを終了し、ワンタイムブートメニューに入り直してこのメニューを更新する必要があります。これにより、デバイスが表示されます。

以下のブートオプションがあります。

- ・ **Windows Boot Manager** などの OS ブートマネージャー - インストールされている OS のブートマネージャーをリストします。
- ・ **Generic USB Boot** - UEFI で起動可能な USB デバイスのプレースホルダーを提供します。このオプションのブート優先順位を設定し、今後取り付ける可能性がある USB デバイスと使用する際にこの優先度を保持できます。この優先順位を設定しても、**UEFI ブート順序**リスト内の個々の USB デバイスの優先順位設定には影響しません。

注記: このオプションは、UEFI モードでのみ使用できます。取り付けられた個々の USB デバイスのブート順序が低く構成されている場合でも、システムは **Generic USB Boot** エントリーで指定された順序ですべての UEFI でブート可能な USB デバイスのブートを試みます。

- ・ **内蔵 UEFI シェル** - ご使用の構成によっては、最初、このブートオプションがワンタイムブートメニューにない場合があります。UEFI システムユーティリティでこのブートオプションを追加できます。UEFI システムユーティリティユーザーガイドの「UEFI ブート順序リストへの内蔵 UEFI シェルの追加」を参照してください。
- ・ **ファイルシステムから UEFI アプリケーションを実行** - このオプションを使用すると、ファイルシステムから実行する UEFI アプリケーションを選択できます。システムで使用できるすべての FAT ファイルシステムを表示できます。x64 UEFI アプリケーション（拡張子.EFI）を選択して実行することもできます（OS ブートローダー、その他の UEFI アプリケーションなど）。
- ・ **レガシー BIOS ワンタイムブートメニュー** - レガシー BIOS ワンタイムブートメニューを起動、終了します。このメニューでは、このブートのみの特定の書き込みオプションを選択できます。このオプションでは、ご使用のブート順序やモードの設定は変更されません。
- ・ **デバイスブートオプション** - このオプションを使用すると、内蔵 Flexible LOM、内蔵 SATA ポート、内蔵 RAIDなどで起動できます。
- ・ **ファイルシステムから UEFI アプリケーションを実行** - このオプションを使用すると、ファイルから実行する UEFI アプリケーションを選択できます。

ワンタイムブートのオプションの選択

手順

1. システムユーティリティ画面で、ワンタイムブートメニューを選択します。
2. ワンタイムブートメニューオプションを選択します。

レガシー BIOS ワンタイムブートオプションを選択すると、システムが再起動します。

USB ブート

USB ブートオプション

- ・ [Windows をインストールするためのブート可能な UEFI USB フラッシュドライブの作成](#) (12 ページ)
- ・ [HPE Dual MicroSD デバイスからのブート](#) (13 ページ)
- ・ [UEFI モードとレガシー BIOS モードの両方のデュアルブート USB キーの作成](#) (13 ページ)

汎用 USB ブート

ワンタイムブートメニューの汎用 USB ブートエントリーは、UEFI でブート可能な USB デバイスのブレースホルダーです。このオプションのブート優先順位を設定し、今後取り付ける可能性がある USB デバイスと使用する際にこの優先度を保持できます。この優先順位を設定しても、UEFI ブート順序リスト内の個々の USB デバイスの優先順位設定には影響しません。新たに追加された USB デバイスは、デフォルトでリストの一番下に表示されますが、これらのエントリーはリスト内で移動し、そのエントリーから起動することもできます。

注記: 取り付けられた個々の USB デバイスのブート順序が低く構成されている場合でも、システムは **Generic USB Boot** エントリーで指定された順序ですべての UEFI でブート可能な USB デバイスのブートを試みます。

Windows をインストールするためのブート可能な UEFI USB フラッシュドライブの作成

前提条件

注記: Windows イメージおよび USB ドライブに関する情報は、[Microsoft Docs](#) を参照してください。

- ・ フラッシュドライブが FAT/FAT32 としてフォーマットされている。
- ・ OS ランチャーを起動する場合、ブートローダーが .efi タイプであり、デフォルトの場所 (\efi\BOOT\BOOTX64.EFI) に格納されている。
- ・ システムユーティリティの **USB オプション** が、USB ドライブからのブートをサポートするように構成されている。USB ドライブからのブートは、デフォルト構成です。詳しくは、UEFI システムユーティリティユーザーガイドの「USB オプション」を参照してください。

手順

1. Windows コンピューターに USB フラッシュドライブを接続して、コマンドプロンプトを開きます。
2. 以下のコマンドを一度に 1 つずつ入力して **Enter** キーを押します。

```
diskpart  
list disk
```
3. USB フラッシュドライブのディスク番号を記録します。
4. USB キーを右クリックしてフォーマットします。

5. OS インストール用の .iso ファイルからイメージファイルを抽出するには、以下の手順に従ってください。
 - a. ファイルエクスプローラーを開いて、.iso ファイルに移動し、右クリックして（コンテキストメニューを開き）マウントをクリックします。
.iso ファイルが開きます。
 - b. .iso ファイル内のすべてのファイルを選択し、送るおよび USB フラッシュドライブの取り外し可能ディスクを右クリックして選択します。
 6. 64 ビットの Windows UEFI USB フラッシュドライブを作成している場合、.iso イメージの内容を USB にコピーします。
 7. USB フラッシュドライブで、efi\microsoft\boot フォルダを 1 つレベルを上げて efi フォルダにコピーし、efi\boot にします
 8. アーカイブユーティリティを開き、USB フラッシュドライブの sources\install.wim\1\Windows\Boot\EFI\bootmgfw.efi に移動します。
-
- 注記:** install.wim ファイルはディレクトリではありません。これは Windows イメージ形式のファイルです。
-
9. bootmgfw.efi ファイルを選択し、これを USB フラッシュドライブの/efi/boot/フォルダにコピーします。
 10. OK をクリックし、アーカイブユーティリティを閉じます。
 11. 名前を bootmgfw.efi file から bootx64.efi に変更します。
 12. ブート可能な UEFI USB フラッシュドライブで UEFI を使用し、Windows のクリーンインストールを実行します。

HPE Dual MicroSD デバイスからのブート

手順

HPE Dual MicroSD は、USB ブートデバイスとよく似ています。HPE Dual MicroSD は、VMware ESXi など、オペレーティングシステムの冗長イメージを格納できます。詳しくは、http://www.hpe.com/support/8GBDualMicrosd_ug_en にある HPE Dual 8GB MicroSD Enterprise Midline USB User Guide を参照してください。

UEFI モードとレガシー BIOS モードの両方のデュアルブート USB キーの作成

レガシー BIOS（DOS など）と UEFI シェルの両方を起動できるデュアルブート USB デバイスを作成することができます。これは、たとえばオフラインの ROM アップグレードユーティリティを実行する場合に便利です。これには、次の操作を行います。

手順

1. レガシー BIOS モード MBR を含む FAT/FAT32 パーティションのある USB デバイスを接続します。
2. レガシーブート可能な USB に \efi\BOOT\BOOTX64.EFI ブートローダーを追加します。

PXE および iPXE ブート

PXE および iPXE 操作

- ・ PXE では、サーバーのネットワーク機能を使用してネットワーク接続経由でクライアントシステムを起動する方法が定義されます。DHCP や TFTP など、標準ネットワークプロトコルが必要です。
- ・ iPXE では、強化された PXE の実装を提供します。

DHCP サーバーの filename が PXE 環境用に構成されている場合は、標準の PXE が起動します。たとえば、Linux 環境で標準の PXE を起動するには、DHCP の filename を pxelinux.0 に設定します。DHCP サーバーの filename が ipxe.efi に設定されている場合に iPXE が起動します。

PXE マルチキャストブート

PXE マルチキャストでは、1 対多または多対多のディストリビューションを使用して、グループ通信のユニキャストとブロードキャストを結合します。これはクライアント側とサーバー側の両方の機能です。Microsoft Windows Server 2012 などのオペレーティングシステムでは、PXE マルチキャストモードでの展開がサポートされています。UEFI では、レガシー BIOS やオプション ROM ではサポートされない機能である、クライアント側での PXE マルチキャストブートが有効になります。UEFI ベースのシステムの PXE スタックは、マルチキャストアドレスで「リスン」します。これによってサーバーは複数の PXE ブートクライアントをマルチキャストグループの一部としてインストールできるため、時間と帯域幅が節約され、サーバー負荷が軽減されます。

PXE 構成

Linux 環境での UEFI ベースクライアント用の PXE サーバーの構成

最も一般的な UEFI PXE ブートローダーは、GRUB および GRUB2 です。

次の例は、既存の BIOS ベース PXE 構成を変更して、BIOS および UEFI ブートローダーの両方を起動するオプションを含める方法を示しています。例では、追加機能を太字で示しています。

変更された ISCdhcpd.conf ファイル

```
#/etc/dhcpd.conf
option domain-name "pxetest.org";
option domain-name-servers 192.168.100.10;
option routers 192.168.100.1;
ddns-update-style none;
subnet 192.168.100.0 netmask 255.255.255.0{
range 192.168.100.20 192.168.100.254;
default-lease-time 14400;
max-lease-time 172800;
next-server 192.168.100.10;
##filename "pxelinux.0"; #comment out for UEFI settings
##Added sections for UEFI
# In initial DHCP DISCOVER packet, PXE client sets option 93 to its arch.
# 0000 == IA x86 PC (BIOS boot)
# 0006 == x86 EFI boot
# 0007 == x64 EFI boot
option arch code 93 = unsigned integer 16; # RFC4578
class "pxe-clients" {
    match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
    if option arch = 00:07 {
        filename "EFI/bootx64.efi";
    }
}
```

```
        } else {
            filename "pxelinux.0";
        }
    }
}
subnet 192.168.100.0 netmask 255.255.255.0{
    range 192.168.100.20 192.168.100.254;
    default-lease-time 14400;
    max-lease-time 172800;
    next-server 192.168.100.10;
}
```

変更された dnsmasq.conf ファイル

```
#Configuration file for dnsmasq
#DHCP configuration
dhcp-option=option:domain-serch,foo.org
dhcp-boot=pxelinux.0,pxeserver,192.168.100.10
# UEFI IPv4 PXE
# currently using elilo boot file
dhcp-match=set:efi-x86_64,option:client-arch,7
dhcp-boot=tag:efi-x86_64,/EFI/bootx64.efi,pxeserver,192.168.100.10
dhcp-range=set:devnet,192.168.100.20,192.168.100.254,1h
dhcp-option=tag:devnet,121,0.0.0.0/0,192.168.100.1
```

注記: DHCP オプション 93 の arch codes については、RFC4578 を参照してください。

更新された TFTP ディレクトリ構造

```
/tftpboot
/tftpboot/pxelinux.0
/tftpboot/pxelinux.cfg/default
/tftpboot/EFI/
/tftpboot/EFI/bootx64.efi
/tftpboot/EFI/grub.cfg
/tftpboot/RHEL6.9
/tftpboot/RHEL7.4
/tftpboot/SLES11SP4
/tftpboot/SLES12SP3
/tftpboot/Ubuntu16.04.3
```

ブートローダーの構成

UEFI ベースのシステムでは、一般的なブートローダーは GRUB および GRUB2 です。正しい bootx64.efi ファイルを取得および構成する方法については、ディストリビューションのドキュメントを参照してください。GRUB2 は、UEFI PXE 構成における標準となっています。GRUB2 は、セキュアブートをサポートしている唯一のブートローダーです。

GRUB2 の使用

GRUB2 は現在セキュアブートをサポートしている唯一のブートローダーでもあります。

注記: GRUB2 と GRUB レガシーは同じものではないことに注意してください。

grub2.conf ファイルのサンプル

```
insmod gettext
insmod iso9660
insmod ntfs
insmod normal
```

```

insmod chain

menuentry 'RHEL 7.4' --class gnu-linux --class gnu --class os {
    echo 'Loading Kernel ...'
    linuxefi /RHEL-7.4Server/vmlinuz repo=http://192.168.100.10/RHEL-7.4Server/disc1
    echo 'Loading initial Ramdisk ...'
    initrdefi /RHEL-7.4Server/initrd.img
}
menuentry 'SLES 12 SP3 AHCI' --class gnu-linux --class gnu --class os {
    echo 'Loading Kernel ...'
    linuxefi /SLE12SP3Server/linux install=http://192.168.100.10/SLE12SP3Server/disc1
    echo 'Loading initial Ramdisk ...'
    initrdefi /SLE12SP3Server/initrd
}
menuentry 'Xenial 16.04.3' --class gnu-linux --class gnu --class os {
    echo 'Loading Kernel ...'
    linuxefi /Ubuntu-16.04.3/Linux
    echo 'Loading initial Ramdisk ...'
    initrdefi /Ubuntu-16.04.3/initrd.gz
}

```

Windows 環境でのレガシー BIOS ベースクライアント用の PXE サーバーの構成

Windows Deployment Service (WDS) などのソフトウェアを使用して、Windows のレガシー BIOS 用に PXE ブートを構成し、DHCP サーバーと TFTP サーバーを混在させるように構成することができます。構成手順は、使用するソフトウェアによって異なる場合があります。

Windows 環境での UEFI ベースクライアント用の PXE サーバーの構成

システム要件

Gen10 サーバーおよび HPE Synergy コンピュートモジュールによってサポートされている Windows のすべてのバージョンは、UEFI モードで起動することができます。

Windows XP や Server 2003 など、Windows の以前のバージョンは、レガシー BIOS モードでのみ起動できません。

Windows サーバーの構成

Windows Deployment Services (WDS) などのソフトウェアを使用して、Windows で UEFI の PXE ブートを構成できます。さらに、WDS を、Microsoft Deployment Toolkit (MDT) や Configuration Manager などの展開ソリューションと組み合わせて使用することができます。構成手順は、使用するソフトウェアに応じて異なる場合があります。

UEFI ネットワークブート用の VLAN の構成

システムユーティリティネットワークオプション > **VLAN 構成**メニューまたは RESTful インターフェイス ツールを使用して有効なネットワークインターフェイスにグローバルな VLAN 構成を設定できます。これには、PXE ブート、iSCSI ブート、FTP/HTTP ブートで使用されるものや、内蔵 UEFI シェルからのすべてのプリブートネットワークアクセスに対するものも含まれます。

NIC カードがサポートされている場合、システムユーティリティ > システム構成オプションの NIC 固有の構成メニューを使用して、ポートの VLAN 設定を設定することもできます。

-
- ❗ **重要:** PCIe スロットで、個々のカード固有の VLAN 構成をサポートしている NIC カードを使用する場合は、システムユーティリティネットワークオプションが提供するグローバルな VLAN 構成方法、または個々のカード固有の VLAN 構成方法のいずれかの方法のみを選択する必要があります。どのような状況でも、両方の VLAN 構成をアクティブにすることはありません。
-

システムユーティリティのネットワークオプションが提供するグローバルな VLAN 構成メニューの使用

手順

1. システムユーティリティ画面で、**システム構成 > BIOS/プラットフォーム構成 (RBSU) > ネットワークオプション > VLAN 構成**を選択します。
2. 以下の操作を実行します。
 - ・ **VLAN 制御** - 有効を選択すると、有効なすべてのネットワークインターフェイス上で VLAN タギングを有効にできます。この設定は、デフォルトでは無効になっています。
 - ・ **VLAN ID** - **VLAN 制御**が有効な場合、すべての有効なネットワークインターフェイスに、0~4094 のグローバル VLAN ID を入力します。
 - ・ **VLAN 優先順位** - **VLAN 制御**が有効な場合、VLAN タグ付きフレームに 0~7 の優先順位の値を入力します。
3. 変更を保存します。

特定の NIC アダプターが提供する構成メニューの使用

手順

1. システムのユーティリティ画面から**システム構成**を選択します。

システム構成画面には、**BIOS/プラットフォーム構成 (RBSU)** オプションと、NIC などの他の利用可能なデバイスの構成が表示されます。
2. ネットワークブートに使用する NIC ポートを選択します。

NIC ポートアダプターの構成オプションが表示されます。オプションのタイトルは、NIC カードによって異なります。
3. NIC の構成メニューオプション（たとえば、**MBA コンフィギュレーションメニュー**）を選択します。

構成メニューが表示されます。
4. **VLAN モード**については、**有効**を選択し、**Enter** キーを押します。
5. **VLAN ID (1 ~ 4094)**については、ネットワークの VLAN 設定と一致する VLAN ID を入力して、**Enter** キーを押します。
6. **Esc** キーを 2 回押して、**Y** を押して構成を保存および終了します。
7. サーバーを再起動します。

注記: NIC アダプター提供の VLAN 設定の構成には、RESTful インターフェイスツールは使用できません。

iPXE の構成

標準的な PXE クライアントは、TFTP を使用してデータを転送しますが、iPXE クライアントのファームウェアは、HTTP など、他のプロトコルを使用してデータを取得する機能を追加します。UEFI iPXE ブートを使用する場合、TFTP および DHCP の構成設定は、**PXE 構成** (14 ページ) に似ていますが、TFTP サーバーを介した iPXE のチェインロードなどの HTTP を構成するための手順を含むことがあります。チェインロードにより、各サーバーのネットワークカードを個別に再フラッシュすることなく、iPXE の機能を取得できます。

iPXE の構成

プロセスの概要

手順

1. iPXE をダウンロードします。
2. オプション : iPXE をカスタマイズします。
3. TFTP を構成します。
4. HTTP を構成します。
5. DHCP を構成します。

iPXE ファイルのダウンロード

手順

1. iPXE オープンソースブートファームウェアの Web サイト : <http://ipxe.org/download> で、次のいずれかのリンクを選択します。
 - ・ ビルト済みのバイナリ ISO イメージ
 - ・ カスタマイズ可能な iPXE ソースコード
2. ダウンロードを完了します。

iPXE のカスタマイズ

手順

1. iPXE のソースコードをコンパイルします。
2. グラフィックコンソールを有効にします。
 - ・ config/console.h ファイルで、次の行のコメントを解除します。

```
#define CONSOLE_FRAMEBUFFER /* Graphical framebuffer console */
```
 - ・ config/general.h ファイルで、次の行のコメントを解除します。

```
#define CONSOLE_CMD /* Console command */
```
3. time コマンドを有効にします。

```
config/general.h
```

 ファイルで、次の行のコメントを解除します。

```
#define TIME_CMD /* Time commands */
```

TFTP の構成

手順

1. ご使用のオペレーティングシステムで指示されている TFTP サーバーパッケージをインストールします。
2. `pxelinux` ファイルを TFTP ルートディレクトリに移動します。

HTTP の構成

手順

1. ご使用のオペレーティングシステムで指示されている HTTP サーバーパッケージをインストールします。
2. `Ipxe.efi` (または `snponly.efi`) ファイルを HTTP のルートディレクトリに移動します。
3. オペレーティングシステムのファイルを HTTP のルートディレクトリに移動します。

DHCP の構成

手順

1. ご使用のオペレーティングシステムで指示されている DHCP サーバーパッケージをインストールします。
たとえば、ISC `dhcpd` サービスをインストールします。
2. 構成ファイルを次のように変更します。
 - ・ PXE クライアントが TFTP サーバーから iPXE をロードできるようにするために、`filename` オプションとして、`ipxe.efi` を指定します。
 - ・ iPXE DHCP クライアントを処理するためのパラメーターを追加し、その `filename` として iPXE スクリプトの HTTP パスを指定します。
 - ・ HTTP ブートが検出された場合に、`ipxe.efi` の HTTP URL を送り返すためのパラメーターを追加します。

チェインロードの構成

iPXE を使用して起動できるようにしたいマシンが多数あるが各マシンのネットワークカードを再フラッシュしたくない場合は、既存の PXE ROM から iPXE をチェインロードすると便利です。この操作は、PXE クライアントが TFTP サーバーから iPXE をロードできるようにすることで実行できます。

チェインロードが有効になっている `dhcpd.conf` ファイル (サンプル)

```
# dhcpd.conf
#
option client-architecture code 93 = unsigned integer 16;
#
# Options common to all private networks
#
option domain-name "ipxedemo.net";
option domain-name-servers 16.110.135.52,16.110.135.51;
option ntp-servers 16.110.135.123;

default-lease-time 600;

max-lease-time 7200;
#
```

```
# Private network
#
subnet 192.168.0.0 netmask 255.255.0.0 {
range 192.168.1.100 192.168.1.200;
option routers 192.168.0.1;
option broadcast-address 192.168.255.255;

# iPXE is DHCP client
if exists user-class and option user-class = "iPXE" {
filename "http://192.168.0.1/ipxe-demo/menu.ipxe";

# UEFI HTTP Boot
# option client-architecture code 93 = unsigned integer 16;
} elsif option client-architecture = encode-int ( 16, 16 ) {
filename "http://192.168.0.1/ipxe.efi";
option vendor-class-identifier "HTTPClient";

# Chain-load iPXE from TFTP server
} else {
filename "ipxe.efi";
}
}
```

PXE または iPXE の起動

前提条件

iPXE および IPv6 PXE では、ブートモードを **UEFI モード** に設定します。

手順

1. サーバーを起動します。
2. POST 画面が表示されたら、次のいずれかを実行します。
 - a. **F12** (ネットワークブート) キーを押します。
 - b. **F11** キーを押してワンタイムブートメニューに進み、PXE または iPXE ブートターゲットを選択します。ワンタイムブートのオプションの選択(10 ページ)を参照してください。

iSCSI ブート

iSCSI ソフトウェアイニシエーターの構成

iSCSI ブートは、iSCSI ストレージエリアネットワーク (SAN) 内の LUN に保存されているブートイメージからのホストシステムのブートを意味します。iSCSI ブートは、iSCSI ソフトウェアイニシエーターを使用して実装することができます。iSCSI ソフトウェアイニシエーターでは、iSCSI ターゲットと呼ばれる Ethernet ネットワーク接続経由でのストレージデバイスへのアクセスが可能になります。これらのターゲットは、iSCSI がディスクへのアクセスをファイルベースではなく、ブロックレベルで提供するという点で、他のネットワークに接続されたストレージデバイスとは異なっています。

iSCSI ブートでは、ソフトウェアに実装されている iSCSI スタック、または iSCSI ホストバスアダプター (HBA) をブロックレベルストレージへの代替アクセス方法として実装されている iSCSI スタックでネットワークアダプターを使用して Windows または Linux ベースのシステムをブートできます。iSCSI ブートでは、各サーバーまたはブレードサーバーでローカルのストレージの必要性を排除することで、消費電力やシステムによる熱量発生を減らすことができます。

iSCSI ソフトウェアイニシエーターの構成と iSCSI のブート

プロセスの概要

前提条件

- ・ Windows 環境で iSCSI ブート用の VLAN を構成する前に、UEFI ネットワークブート用の VLAN の構成 (16 ページ)の手順を完了してください。
- ・ ブートモードを **UEFI モード**に設定したままにします。

手順

1. iSCSI ブート用のデバイスを選択し、これをネットワークブート可能にします。
2. iSCSI イニシエーターの名前を追加します。
3. iSCSI ブート試行を追加します。
4. iSCSI 接続を確認します。
5. iSCSI イニシエーターのブート順序を変更します。

iSCSI ブート用のデバイスを選択し、ネットワークブート可能にする

手順

1. システムユーティリティ画面で、システム構成 > BIOS/プラットフォーム構成 (RBSU) > ネットワークオプション > ネットワークブートオプションを選択します。
iSCSI に使用可能なネットワークデバイスとポートがネットワークブートオプション画面に表示されません。
2. iSCSI ターゲットが利用可能な SAN ネットワークに接続されているデバイスを選択します。
3. ネットワークブートを選択します。
4. 設定を保存します。

iSCSI イニシエーター名の追加

iSCSI イニシエーター名のオプションを使用して、IQN (iSCSI Qualified Name) 形式で名前を設定します。EUI フォーマットはサポートされません。このオプションは、イニシエーターに設定されたデフォルト名に置き換わります。

手順

1. システムユーティリティ画面で、**システム構成 > BIOS/プラットフォーム構成 (RBSU) > ネットワークオプション > iSCSI ブート構成 > iSCSI イニシエーター名**を選択します。
2. iSCSI 修飾名 (IQN) フォーマットを使用して iSCSI イニシエーターの一意的名前を入力します。たとえば、`iqn.2001-04.com.example:uefi-13021088` です。

この設定は自動的に保存されます。

iSCSI ブート試行の追加

手順

1. システムユーティリティ画面で、**システム構成 > BIOS/プラットフォーム構成 (RBSU) > ネットワークオプション > iSCSI ブート構成 > iSCSI ブート試行を追加**を選択します。

このブート試行が、次にサーバーが再起動されるまで有効にならないことを示すメッセージが表示されません。

2. **Enter** を押します。
3. iSCSI 接続を試行するポートを選択します。
4. 構成設定を完了します。
 - ・ **iSCSI 試行名** - 名前を入力します。
 - ・ **iSCSI ブート制御** - **有効**を選択します。(デフォルト設定は**無効**です。)
 - ・ **IP アドレスタイプ** - アドレスタイプを選択します。
 - ・ **接続再試行カウント** - 0~16 の値を入力します。デフォルトの再試行回数は 3 回です。
 - ・ **接続タイムアウト** - 100~20000 の値 (ミリ秒単位) を入力します。デフォルトは、20000 (20 秒) です。
 - ・ **イニシエーター DHCP** - デフォルト設定です。イニシエーターについて静的 IP アドレスを構成する必要がある場合は、このオプションをオフにします。イニシエーターについて静的アドレスを構成する場合は、ターゲット名、IP アドレス、ポート、およびブート LUN も手動で構成する必要があります (ターゲット DHCP 構成を無効にします)。
 - ・ **ターゲット DHCP 構成** - デフォルト設定です。ターゲット設定を手動で構成する必要がある場合は、このチェックボックスをオフにして、ターゲット名、IP アドレス、ポート、およびブート LUN を入力します。
 - ・ オプション: **認証タイプ** - デフォルトは NONE です。必要な場合、**CHAP** を選択して CHAP エントリーを入力します。
5. **変更の保存**を選択します。
6. システムを再起動します。

iSCSI 接続の確認

手順

1. システムユーティリティ画面で、**システム構成 > BIOS/プラットフォーム構成 (RBSU) > ネットワークオプション > iSCSI ブート構成 > iSCSI 試行**を選択します。
iSCSI ブート試行の構成画面が表示されます。
2. **iSCSI ブート試行の追加**で構成したブート試行に対して正しい情報が一覧表示されていることを確認します。イニシエーター DHCP を選択した場合、新たに割り当てられた IP アドレスが記載されていることを確認します。
3. システムユーティリティ画面で、**内蔵アプリケーション > 内蔵 UEFI シェル**を選択します。
4. 内蔵 UEFI シェルプロンプトで、`map -r` コマンドを入力します。
マッピングテーブルが表示されます。
5. iSCSI ターゲットデバイスがファイルシステムとして識別されることを確認します。たとえば、次のように入力します。FS0
6. `exit` コマンドを入力して、内蔵 UEFI シェルを終了します。
7. iSCSI ターゲットデバイスが**ワンタイムブートメニュー**に一覧表示されていることを確認します。**ワンタイムブートメニューオプション**(10 ページ)を参照してください。
8. Windows または Linux ブートディスクまたはライブイメージで起動した場合は、マップ済み iSCSI ターゲットデバイスがインストールターゲットとして一覧表示されていることを確認します。

iSCSI イニシエーターのブート順序の変更

iLO 5 Web インターフェイス、システムユーティリティメニュー、または RESTful インターフェイスツールを使用して、iSCSI イニシエーターのブート順序を変更できます。システムユーティリティの使用方法については、**UEFI ブート順序リストの変更**(8 ページ)を参照してください。

HTTP ブート

HTTP ブートは、ネットワークブートシステムの業界標準の新機能で、TFTP ベースの PXE ブートの後継となる、より高速かつ安全な機能です。DHCP、DNS、および HTTP プロトコルを組み合わせ、ネットワーク上のシステムの展開および構成を実現します。HTTP ブートは、PXE ブートと同様に動作しますが、HTTP または HTTPS を使用してイメージをダウンロードします。HTTP ブートは UEFI モードでのみサポートされません。

PXE のように、HTTP ブートは、EFI ローダーやアプリケーションのようなネットワークブートストラッププログラム (NBP) のダウンロードおよび起動をサポートします。さらに、HTTP ブートは、ISO イメージをダウンロードして、そこから起動することもサポートしています。これは HTTP ブートの新機能で、PXE ではサポートされません。URL は、IPv4 または IPv6 のアドレスか、完全修飾ホスト名となります。ホスト名が URL で指定された場合、DNS 名前解決が使用されます。

HTTP ブートは、次の 2 つの方法のどちらかでを行います。

1. PXE ブートと同じように、DHCP を使用してブートファイル (EFI アプリケーション/ブートローダーまたは ISO イメージ) を検出します。URL が DHCP サーバーによって指定されている場合、システムは URL によってポイントされるファイルをダウンロードし、そこから起動します。

HPE ProLiant Gen10 サーバーおよび HPE Synergy コンピュートモジュールで、**RBSU > ネットワークオプション > ネットワークブートオプション > HTTP サポートオプション**において、**自動 (デフォルト)、HTTP のみ**、または **HTTPS のみ**を設定すると、HTTP ブートオプションが作成され、ネットワークブートが有効になっている各ポートの UEFI ブートメニューに表示されます。作成された HTTP ブートオプションを **ワнтаイムブートメニュー**から選択すると、システムはそのネットワークデバイス上で HTTP ブートの実行を試みます。

HTTP ブートを実行するサーバー側の構成のセットアップについて詳しくは、[ブート URL を指定するネットワークサービスを構成する](#)を参照してください。

2. 静的に指定された URL (EFI アプリケーションまたは ISO イメージをポイントしている) から起動します。静的に指定された URL からの起動をサポートするために、DHCP サーバーで特別な構成を行う必要はありません。DHCP サーバーは、この URL をダウンロードできるようにシステムに必要な IP アドレス構成のみ指定します。

HPE ProLiant Gen10 サーバーおよび HPE Synergy コンピュートモジュールでは、ユーザー指定の URL を最大 4 つまで構成することができます。1 つまたは複数の特定の URL からの起動を構成するには、**RBSU > ネットワークオプション > プリブートネットワーク設定**において、URL 1、2、3、または 4 から起動の設定を使用します。各 URL は、ブート可能な ISO または EFI ファイルとなります。HTTP または HTTPS のいずれかの形式で、IPv4 または IPv6 のサーバーアドレスまたはホスト名を使用して URL を入力します。たとえば、URL を次のいずれかの形式にすることができます。http://192.168.0.1/file/image.iso、http://example.com/file/image.efi、https://example.com/file/image.efi、http://[1234::1000]/image.iso。URL が構成されると、この URL が UEFI ブートメニューにブートオプションとして表示されます。

これらの設定を行うには、RESTful インターフェイスツールまたは内蔵 UEFI シェルの sysconfig コマンドを使用することもできます。その後、ブートメニューからこのオプションを選択し、指定されたファイルをシステムメモリにダウンロードして、そのファイルからシステムをブートできるようにすることができます。**RBSU > ネットワークオプション > プリブートネットワーク設定 > プリブートネットワークインターフェイス**で設定された値に基づいて、この URL をダウンロードおよび起動するネットワークインターフェイスが選択されます。

ブート URL を指定するネットワークサービスの構成

この一連のタスクを使用して、ネットワークサーバー側の構成をセットアップします。ここで示したすべての例は、Ubuntu/Debian で行われるものです。ご使用の環境で同じ構成を実行するには、ご使用の OS のユーザーおよび管理ガイドを参照してください。

手順

1. 必要なパッケージのインストール
2. ネットワークインターフェイスのセットアップ
3. DHCP サービスの構成
4. ルーター広告デーモンの構成
5. IPv4 および IPv6 用 DNS サービスの構成
6. HTTP サービスの構成
7. HTTP 基本認証の構成
8. apache2 での HTTPS (TLS) のセットアップ
9. クライアントマシンへの証明書のインストール

必要なパッケージのインストール

この表は、Debian/Ubuntu システムに推奨されるサービスを示したものです。

DNS	bind9
DHCP IPv4	isc-dhcp-server
DHCP IPv6	isc-dhcp-server および radvd
Web サーバー	apache2

手順

ご使用の環境に必要なパッケージをインストールします。ご使用の OS でのパッケージのインストールについては、ご使用の OS のドキュメントを参照してください。

ネットワークインターフェイスのセットアップ

手順

DHCP および他のサービスがクライアントの要求をリスンするネットワークインターフェイスをセットアップします。ネットワークインターフェイスのセットアップについては、ご使用の OS のドキュメントを参照してください。

DHCP サービスの構成

以下について、DHCP サービスを構成します。

- ・ ブートファイルまたは ISO イメージを求める HttpBoot クライアント要求用の IPv4 および IPv6 アドレス割り当て
- ・ startup.nsh スクリプトを求める HPE 内蔵 UEFI シェルクライアント要求

手順

1. IPv4 経由で DHCP をセットアップするには、これらの行を `/etc/dhcp/dhcp.conf` に追加します。

```

option arch code 93 = unsigned integer 16;
option user-class code 77 = { integer 8, string };

subnet 192.168.111.0 netmask 255.255.255.0 {

    # Common response to all client requests on this subnet. IP, GW and DNS.
    range 192.168.111.100 192.168.111.120;
    option routers 192.168.111.1;
    option domain-name-servers 192.168.111.3;

    # This is to respond to requests from HttpBoot clients with what they require.
    class "http-clients" {
        match if substring (option vendor-class-identifier, 0, 10) = "HTTPClient";
        option vendor-class-identifier "HTTPClient";
        if option arch = 00:10 { # x86-64 architecture systems.
            option bootfile-name "https://192.168.111.4/ISO/sles12sp2_x64.iso";
            option bootfile-name "http://webserv.example.com/boot/grub2/x86_64-efi/grub.efi";
        }
    }

    # This is to respond to requests from Embedded UEFI Shell asking for a script URL.
    class "uefishell-clients" {
        match if substring (option user-class, 1, 9) = "UEFIShell";
        option user-class 9 "UEFIShell";
        if option arch = 00:10 {
            option bootfile-name "https://webserv.example.com/SCRIPT/start.nsh";
        }
    }
}

```

注記:

「サブネット」宣言は、DHCP サービスがリスンするネットワークインターフェイスのプレフィックス（アドレスおよびネットマスク）と一致する必要があります。

注記: 他のアーキテクチャタイプについては、<http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml> で「Processor Architecture Types」を参照してください。

2. IPv6 用の DHCP をセットアップするには、次の操作を行います。

- a. 新しいファイルを 2 つ作成します：`/var/lib/dhcp/dhcpd6.leases`（空のファイル）および`/etc/dhcp/dhcpd6.conf`。
- b. これらの設定を`/etc/dhcp/dhcpd6.conf` ファイルに追加します。

```

ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
log-facility local7;

option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = { integer 16, string };
option dhcp6.vendor-class code 16 = { integer 32, integer 16, string };

subnet6 1235::/64 {

    # Common response for all client requests on this subnet. Supply IP and DNS.

```

```

range6 1235::2000 1235::2100;
option dhcp6.name-servers 1235::1003;

# This is to respond to requests from HttpBoot clients with what they require.
class "http-clients" {
    match if substring (option dhcp6.vendor-class, 6, 10) = "HTTPClient";
    option dhcp6.vendor-class 0 10 "HTTPClient";
    if option dhcp6.client-arch-type = 00:10 { # x86-64 architecture systems.
        option dhcp6.bootfile-url "https://[1235::1004]/ISO/sles12sp2_x64.iso";
        #option dhcp6.bootfile-url "http://webserv.example.com/boot/grub2/x86_64-efi/grub.efi";
    }
}

# This is to respond to requests from Emb UEFI Shell asking for a script URL.
class "uefishell-clients" {
    match if substring (option dhcp6.user-class, 2, 9) = "UEFIShell";
    option dhcp6.user-class 9 "UEFIShell";
    if option dhcp6.client-arch-type = 00:10 {
        option dhcp6.bootfile-url "http://[1235::1004]/SCRIPT/start.nsh";
    }
}
}

```

注記: 起動に使用する任意の.iso または.efi を URL に設定できます。.iso または.efi は、HTTP サーバーのルートフォルダーの下に配置する必要があります。詳しくは、[HTTP サービスの構成](#)を参照してください。

3. IPv4 と IPv6 の両方で DHCP サービスを起動または再起動します。

ルーター広告デーモンの構成

自動 IPv6 アドレス構成用のルーターアドレスおよびネットワークプレフィックスを探しているクライアントからの IPv6 ICMP ルーター要請メッセージに応答するためのルーター広告デーモン (radvd) を構成します。

手順

1. /etc/radvd.conf ファイルを作成します。

```

interface eth0 {
    AdvSendAdvert on;
    AdvManagedFlag on;
    prefix 1235::/64 {
    };
};

```

注記: このインターフェイスは、クライアントが IPv6 ルーターアドレスの要求を送信する場所と一致する必要があります。

2. /etc/sysctl.conf ファイルを編集することによって、radvd サービスが実行されているルーター (ホスト) での IPv6 パケット転送を有効にします。

```

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1

```

注記: この情報が/etc/sysctl.conf がない場合は、ファイルに追加します。

3. radvd サービスを再起動します。

IPv4 および IPv6 用 DNS サービスの構成

クライアント要求を特定の IP アドレスでのみリスンするように DNS サービス (named) を構成します。そうしないと、サービスはデフォルトで、必要ない場合でもすべての NIC インターフェイスでリスンします。

手順

1. /etc/bind/named.conf.options を編集して、オプション{}セクションの中にこれらの行を追加し、その他の listen コマンドを削除またはコメントアウトします。

```
listen-on { 192.168.1.3; };
listen-on-v6 { 1235::1003; };
```

注記: アドレスは、DNS サービスが検索要求をリスンするローカル NIC インターフェイスの IP アドレスと一致する必要があります。

2. /etc/bind/named.conf.local を編集します。

```
zone "abc.com" {
    type master;
    file "/etc/bind/db.abc.com";
};
```

3. /etc/bind/db.abc.com を編集して、これらの行を追加すると、正引き (ホストから IP) が構成され、dnssrv. および root.dnssrv. に、DNS サービスが実行中のサーバーのホスト名および IP アドレスが設定されます。サービス名の末尾にある '!' は、意図的なものです。構成を変更して DNS サービスを再起動するたびに、「Serial」フィールドがインクリメントされる必要があります。

```
;
; BIND data file for interface 192.168.111.3/1235::1003
;
$TTL      604800
@         IN      SOA      dnssrv.      root.dnssrv. (
                        3              ; Serial
                        604800         ; Refresh
                        86400          ; Retry
                        2419200        ; Expire
                        604800 )       ; Negative Cache TTL
;
@         IN      NS       dnssrv.
@         IN      A        192.168.111.3
@         IN      AAAA     1235::1003

; other hosts in this network. webserv.abc.com is the only one for now.
webserv  IN      A        192.168.111.4
webserv  IN      AAAA     1235::1004
```

オプションで逆引き (IP からホスト) を構成することもできますが、用途上、必要ありません。

4. DNS サービスを再起動します。

HTTP サービスの構成 (apache2)

このタスクによって、apache2 での HTTP の GET および PUT が有効になります。

このタスクを完了すると、`netstat antep |grep apache2` コマンドで構成されたポートの IPv4 および IPv6 の両方のアドレスで apache2 がリスンしているか確認できます。

```
# netstat -antep | grep apache2
tcp      0      0 192.168.111.4:80    0.0.0.0:*    LISTEN   0      36264      1897/apache2
tcp6     0      0 1235::1004:80      :::*        LISTEN   0      36262      1897/apache2
```

前提条件

URL からの起動に使用するファイル (.iso/.efi) を、ルートディレクトリ /var/www/html/ にコピーする必要があります。

手順

1. /etc/apache2/ports.conf を編集して、サーバーが使用するポートおよび IP アドレスをセットアップします。

apache2 がクライアント要求をリスンするすべてのアドレスを指定します。次の例では、apache2 が実行されているシステムのネットワークインターフェイスが、IPv4 アドレス 192.168.1.4 および IPv6 アドレス 1235::1004 で構成されています。

```
Listen [1235::1004]:80
Listen 192.168.111.4:80
```

```
<IfModule ssl_module>
    Listen 443
</IfModule>
```

```
<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

注記: 一部のシステムには、ports.conf ファイルがない場合があります。/etc/apache2/apache2.conf に「Include ports.conf」の行がある場合、ports.conf ファイルが存在し、設定の例を ports.conf に追加することができます。そうでない場合、apache2.conf 自体に前述の例の行を追加する必要があります。

2. /etc/apache2/apache2.conf を次のように編集します。

- a. これらの行を追加します。

```
LoadModule dav_module /usr/lib/apache2/modules/mod_dav.so
LoadModule dav_fs_module /usr/lib/apache2/modules/mod_dav_fs.so
LoadModule dav_lock_module /usr/lib/apache2/modules/mod_dav_lock.so
DavLockDB /var/www/DavLock
```

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    AuthType None
    Require all granted
    Dav On
</Directory>
```

b. ファイル内の他の<Directory> </Directory>ブロックをすべてコメントアウトします。

3. KeepAlive をオンに設定します (まだ設定されていない場合)。

```
KeepAlive On
```

4. apache2 を再起動します。

HTTP 基本認証の構成

apache2 で HTTP 基本認証を有効にする場合にのみ、このタスクを実行します。

手順

1. 次の行を/etc/apache2/sites-enabled/000-default.conf に追加します。

```
<Directory "/var/www/html">
    AuthType Basic
    AuthName "Restricted Content"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>
```

2. パスワードを生成して/etc/apache2/.htpasswdに格納するには、htpasswd コマンドを実行します。任意の適切なユーザー名を設定します。このユーザー名は、クライアントが HTTP 要求をサーバーに送信する際に指定する必要があるユーザー名とパスワードになります。この例では、httpuser がユーザー名です。

```
htpasswd -c /etc/apache2/.htpasswd httpuser
```

3. apache2 を再起動します。

apache2 での HTTPS (TLS) のセットアップ

手順

1. mkdir および openssl コマンドを使用して、apache2 サーバー用の自己署名証明書を作成します。例は次のとおりです。

```
mkdir /etc/apache2/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

openssl コマンドの後に Enter キーを押すと、いくつか質問されます。

2. プロンプトが表示されたら、質問に答えます。

```
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Your State
Locality Name (eg, city) []:Your City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:YourCompany
Organizational Unit Name (eg, section) []:Department of Kittens
Common Name (e.g. server FQDN or YOUR name) []:abc.com
Email Address []:your_email@abc.com
```

キーおよび証明書が作成され、`/etc/apache2/ssl` ディレクトリに配置されます。

3. SSL を有効化します。

```
a2enmod ssl
```

4. `/etc/apache2/sites-available/default-ssl.conf` を編集して、仮想ホストの設定が構成されていることを確認します。

```
<VirtualHost _default_:443>
  ServerAdmin admin@abc.com
  ServerName abc.com
  ServerAlias www.abc.com
  DocumentRoot /var/www/html
  SSLCertificateFile /etc/apache2/ssl/apache.crt
  SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

5. 仮想ホストを有効化します。

```
a2ensite default-ssl.conf
```

6. apache2 を再起動します。

クライアントマシンへの証明書のインストール

手順

1. 次のいずれかを実行します。

- ・ 証明書 (`/etc/apache2/ssl/apache.crt`) を HTTP 経由でサーバーから HPE ProLiant Gen10 システムに取得し、ローカルファイルシステムに配置します。これを行うには、証明書を HTTP サーバーのルートディレクトリ `/var/www/html/` にコピーして、SUT 内蔵 UEFI シェルで以下を実行します。

```
FS0:\> webcli -g http://websrv.abc.com/apache.crt
```

注記: 証明書をダウンロードする前に、Gen10 システムに使用可能なローカルストレージオプションがない場合は、RAM ディスクを作成することもできます。

```
Shell:> ramdisk -c -s 128
```

- ・ USB デバイスを使用します。サーバーの証明書/etc/apache2/ssl/apache.crt を USB キーにコピーし、その USB キーを Gen10 システムに接続します。

2. `tlsconfig` コマンドを使用して証明書をインストールします。

```
tlsconfig -e -f apache.crt
```

注記: iLO RESTful インターフェイスツールを使用することもできます。詳しくは、<https://hewlettpackard.github.io/ilo-rest-api-docs/ilo5/#https-boot-tls-configuration> を参照してください。

サポートされるブートローダーを使用した HTTP (S) ブートの実行

HTTP (S) ブートまたは URL からの起動は、次の 2 種類の方法で実行できます。

1. ISO イメージ (mini/net インストーラーイメージまたはフルメディアイメージが可能) をダウンロードして起動します。**ブート URL を指定するネットワークサービスの構成**にあるすべてのタスクを完了すると、この方法でシステムが起動するよう構成されます。
2. EFI ブートローダー (たとえば、`grub.efi`) などのネットワークブートストラッププログラムをダウンロードします。これは、PXE ブートに似ています。

GEN10 リリースの場合、SLES12 SP2、SP3、SLES 15、および RHEL 7.6 は、両方の HTTP (S) ブート方法をサポートします。

	SUSE Linux Enterprise Server 12 SP2、SP3	SUSE Linux Enterprise Server 15	Red Hat Enterprise Linux 7.6
HTTP URL を使用する方 法 1	対応	対応	対応
HTTP URL を使用する方 法 2	対応	対応	対応
HTTPS URL を使用する 方法 1	対応	対応	対応
HTTPS URL を使用する 方法 2	X	対応	対応

サポートされる OS インストールイメージからブートローダーを使用することで、サポートされる別の OS の HTTP (S) ブートを実行することができます。

このタスクは、サポートされるブートローダーをセットアップするための手順を示しています。

手順

1. サーバーにインストールメディアの内容をコピーします。たとえば、HTTP ルートディレクトリ (`/var/www/html/`) にフォルダー (`/sles12sp3`) を作成します。SLES12 SP3 インストールメディア

(ISO イメージまたは CD/DVD ドライブ) をマウントし、ファイルをコピーして (またはシンボリックリンクを作成して) /var/www/html/sles12sp3/ にコピーします。

2. HTTP ルートディレクトリに /boot/grub2/x86_64-efi/ フォルダを作成し、grub.efi をインストールメディアから、新たに作成したフォルダにコピーします。たとえば、grub.efi を /var/www/html/sles12sp3/EFI/BOOT/ から /var/www/html/boot/grub2/x86_64-efi/ にコピーします。
3. /var/www/html/boot/grub2/ に grub.cfg を作成して、次の例のように構成します。

注記: /var/www/html/ に grub.cfg を、また /var/www/html/x86_64-efi/ に grub.efi を置くことができます。

```
set timeout=60
terminal_output console
menuentry 'SLES 12 SP3' --class os {
echo 'Loading Linux ...'
linuxefi /sles12sp3/boot/x86_64/loader/linux install=http://192.168.111.4/sles12sp3/
echo 'Loading initial ramdisk ...'
initrdefi /sles12sp3/boot/x86_64/loader/initrd
}
menuentry 'SLES 15 (HTTPS)' --class os {
echo 'Loading Linux ...'
linuxefi (https,192.168.111.4)/sles15/boot/x86_64/loader/linux install=http://192.168.111.4/sles15/
echo 'Loading initial ramdisk ...'
initrdefi (https,192.168.111.4)/sles15/boot/x86_64/loader/initrd
}
menuentry 'RHEL 7.6' --class os {
echo 'Loading Linux ...'
linuxefi linuxefi /images/pxeboot/vmlinuz inst.stage2=http:// 192.168.111.4/EFI/BOOT/grubx64.efi inst.noverifyssl quiet
echo 'Loading initial ramdisk ...'
initrdefi /images/pxeboot/initrd.img
}
menuentry 'RHEL 7.6 (HTTPS)' --class os {
echo 'Loading Linux ...'
linuxefi linuxefi /images/pxeboot/vmlinuz inst.stage2=http:// 192.168.111.4/EFI/BOOT/grubx64.efi inst.noverifyssl quiet
echo 'Loading initial ramdisk ...'
initrdefi /images/pxeboot/initrd.img
}
```

注記: ブートローダーが、HTTP サーバーに接続されているネットワークポートをデフォルトのネットワークポートとして使用していない場合、インストールソースの選択メニューでブートローダーが停止することがあります。この場合、ブートローダーがすべてのネットワークポートを通るように URL (http://192.168.111.4/sles12sp3/) を手動で入力して、インストールプロセスを続行することができます。

4. ブート URL が grub.efi をポイントするように、DHCP サーバーの dhcpd.conf を構成します。
 - a. 次のいずれかを入力します。

- ・ HTTP ブートの場合 :

```
option bootfile-name "http://192.168.111.4/boot/grub2/x86_64-efi/
grub.efi";
```

- ・ HTTPS ブートの場合 :

```
option bootfile-name "https://192.168.111.4/boot/grub2/x86_64-efi/
grub.efi";
```

- b. isc-dhcp-server サービスを再起動します。

注記: クライアントシステム (HPE ProLiant Gen10 または HPE Synergy コンピュートモジュール) が、HTTP ブートではなく URL からの起動を使用する場合、「URL x から起動」オプションに URL 例を直接入力することもできます。

URL からの起動の構成

前提条件

システムユーティリティによる URL からの起動の構成

手順

1. システムユーティリティ画面で、システム構成 > BIOS/プラットフォーム構成 (RBSU) > ネットワークオプション > プリブートネットワーク設定を選択します。
2. 別の NIC ポートを使用するようにプリブートネットワークインターフェイスを変更するには、プリブートネットワークインターフェイスオプションを選択および構成します。デフォルトは「自動」で、ネットワーク接続されている最初のネットワークインターフェイスポートを自動的に使用します。
3. URL を設定するには、URL 1 から起動オプションを選択し、ブート可能な ISO または EFI ファイルをポイントする有効な HTTP または HTTPS で構成します。

内蔵 UEFI シェルの sysconfig コマンドによる URL からの起動の構成

手順

1. システムユーティリティ画面で、内蔵アプリケーション > 内蔵 UEFI シェルを選択します。
2. 内蔵 UEFI シェルプロンプトから、次のいずれかを実行します。
 - ・ プリブートネットワークインターフェイスを指定し、URL 1 からの起動を IPv4 アドレスで構成します。例：

```
Shell> sysconfig -s PreBootNetwork=Slot1NicPort1  
UrlBootFile=http://192.168.1.1/bootx64.efi
```

- ・ ネットワーク接続されている最初のネットワークインターフェイスから起動する URL を設定します (プリブートネットワークインターフェイス = 自動)。例：

```
Shell> sysconfig -s UrlBootFile=http://server.domain.com/boot.iso
```

- ・ FlexibleLOM から起動する IPv6 の URL を設定します。例：

```
Shell> sysconfig -s PreBootNetwork=FlexLom1Port1  
UrlBootFile=http://[1235::1000]/bootx64.efi
```

3. システムユーティリティ画面で、ワンタイムブートメニューを選択します。
4. URL ファイルオプションを選択します。

RESTful インターフェイスツールによる URL からの起動の構成

<http://www.hpe.com/info/redfish-ja> で、RESTful インターフェイスツールの場所を確認します。

手順

1. RESTful インターフェイスツールを実行します。
2. コマンドラインプロンプトで、次のいずれかを実行します。
 - ・ プリブートネットワークインターフェイスを指定し、URL 1 からの起動を IPv4 アドレスで構成します。
`ilorest set PreBootNetwork=Slot1NicPort1 UrlBootFile=http://192.168.1.1/bootx64.efi`
 - ・ デフォルトのインターフェイスから起動する URL を設定します（プリブートネットワークインターフェイス = 自動）。
`ilorest set PreBootNetwork=Auto UrlBootFile=https://server.domain.com/boot.iso`
 - ・ FlexibleLOM から起動する IPv6 の URL を設定します。
`ilorest set PreBootNetwork=FlexLom1Port1 UrlBootFile=http://[1235::1000]/bootx64.efi`
3. 変更を有効にするために、次のいずれかを実施します。
 - ・ システムを実行している場合は再起動します。
 - ・ システムの電源がオフの場合、電源をオンにします。

プリブートネットワーク設定構成

システムユーティリティ、内蔵 UEFI シェル、または RESTful インターフェイスツールを使用して、プリブートネットワークインターフェイスと関連する設定を構成できます。

注記: システムの実行時に RESTful API を使用して構成を変更した後、その変更を反映するためにシステムを再起動する必要があります。RESTful API を使用した構成の時点でシステムの電源を切った場合は、システムの電源投入後に設定が表示されます。システムユーティリティまたは UEFI 埋め込み Shell の `sysconfig` コマンドを使用して行った変更は、すぐに適用されます。

システムユーティリティによるプリブートネットワーク設定の構成

手順

1. システムユーティリティ画面で、システム構成 > BIOS/プラットフォーム構成 (RBSU) > ネットワークオプション > プリブートネットワーク設定 を選択します。
2. プリブートネットワーク設定 を選択および構成します。

プリブートネットワーク設定

- ・ **起動前ネットワークインターフェイス** - 起動前のネットワーク接続に使用するネットワークインターフェイスを指定します。
 - **自動 (デフォルト)** - システムは、ネットワーク接続されている最初の使用可能なポートを使用します。
 - **特定のポートの選択** - システムは、選択された NIC ポートを使用します。
- ・ **DHCPv4** - 内蔵 UEFI シェルからのネットワーク操作のために、DHCP サーバーからのプリブートネットワーク IPv4 構成の取得を有効または無効にします。
 - **有効** - DHCPv4 ネットワークアドレス構成を有効にします。個別の設定は使用できません。
 - **無効** - DHCPv4 のアドレス構成を無効にします。したがって以下の静的 IP アドレス設定を手動で構成する必要があります。
 - IPv4 アドレス
 - IPv4 サブネットマスク
 - IPv4 ゲートウェイ
 - IPv4 プライマリ DNS
 - IPv4 セカンダリ DNS
- ・ **プリブートネットワークプロキシ** - プリブートネットワークプロキシを指定します。これが設定されている場合、プリブートネットワークインターフェイスのネットワーク操作は構成済みのプロキシ経由で試行されます。プロキシは HTTP URL 形式である必要があり、`http://IPv4_address:port`、`http://[IPv6_address]:port` または `http://FQDN:port` として指定することができます。
- ・ **IPv6 構成ポリシー**

- **自動** - 内蔵 UEFI シェルからのネットワーク操作のために、プリブートネットワーク IPv6 構成を自動的に取得することができます。個別の設定は使用できません。
- **手動** - 静的 IP アドレス設定を個別に構成することができます。
- ・ **URL 1、2、3、または 4 から起動** - ブート可能な ISO または EFI ファイルのネットワーク URL を指定します。HTTP または HTTPS のいずれかの形式で、IPv4 または IPv6 のサーバーアドレスまたはホスト名を使用して URL を入力します。たとえば、URL を次のいずれかの形式にすることができます。http://192.168.0.1/file/image.iso、http://example.com/file/image.efi、https://example.com/file/image.efi、http://[1234::1000]/image.iso。構成すると、この URL が UEFI ブートメニューにブートオプションとして表示されます。その後、ブートメニューからこのオプションを選択し、指定されたファイルをシステムメモリにダウンロードして、そのファイルからシステムをブートできるようにすることができます。

注記: URL からの起動は、「DHCPv4」および「IPv6 構成ポリシー」設定に依存しません。

ISO ファイルからの起動には、予備の OS 環境イメージ（WinPE やミニ Linux など）または完全な OS インストールイメージの起動のみを含めることができます（ただし、OS が HTTP ブート機能をサポートする場合。古い OS バージョンは ISO ファイルまたは OS インストールイメージからの起動をサポートしない可能性があります）。使用している OS のドキュメントで HTTP ブート機能がサポートされるかどうかを確認してください。

- ❗ **重要:** 同じインターフェイス上で `webclient` または `ftp` を実行する予定の場合、ネットワークインターフェイス上で内蔵 UEFI シェル `ifconfig` コマンドを使用する必要はありません。システムユーティリティ内で構成されている**プリブートネットワーク設定**は、これらのインターフェイスを自動的に選択します。

`ftp` および `webclient` によって使用されるインターフェイスが `ifconfig` によって構成される場合、その設定は消去されます。代わりに、システムユーティリティ**プリブートネットワーク設定**メニューは、コマンドが実行されるとインターフェイスに適用されます。

内蔵 UEFI シェルの `sysconfig` コマンドによるプリブートネットワーク設定の構成

手順

1. システムユーティリティ画面で、**内蔵アプリケーション > 内蔵 UEFI シェル**を選択します。
2. 内蔵 UEFI シェルプロンプトから、次のいずれかを実行します。

- ・ DHCP を使用して、特定の NIC ポートの IPv4 アドレスを取得します。例：

```
Shell> sysconfig -s PreBootNetwork=EmbNicPort1 Dhcpv4=Enabled
```

- ・ 特定の NIC ポートの IPv6 自動構成を使用します。例：

```
Shell> sysconfig -s PreBootNetwork=FlexLom1Port1 Ipv6ConfigPolicy=Automatic
```

- ・ システムが、ネットワーク接続されているインターフェイスを選択しながらも、静的 IPv4 アドレスを使用するようにします。例：

```
Shell>sysconfig -s PreBootNetwork=Auto Dhcpv4=Disabled Ipv4Address=192.168.1.66  
Ipv4SubnetMask=255.255.255.0 Ipv4Gateway=192.168.1.1 Ipv4PrimaryDNS=192.168.1.1  
Ipv4SecondaryDNS=192.168.1.2
```

- ・ システムが、ネットワーク接続されているインターフェイスを選択しながらも、静的 IPv6 アドレスを使用するようにします。例：

```
Shell>sysconfig -s PreBootNetwork=Auto Ipv6ConfigPolicy=Manual  
Ipv6Address=1235::1022 Ipv6Gateway=1235::1000 Ipv6PrimaryDNS=1235::1000
```

3. `webcli` または `ftp` コマンドを使用して、ファイルをダウンロードまたはアップロードします。

RESTful インターフェイスツールによるプリブートネットワーク設定の構成

<http://www.hpe.com/info/redfish-ja> で、RESTful インターフェイスツールの場所を確認します。

手順

1. RESTful インターフェイスツールを実行します。

```
ilorest login proliantssystem1.hpe.com -u username -p password  
ilorest select Bios.v1_0_0
```

2. コマンドラインプロンプトで、次のいずれかを実行します。

- ・ DHCP を使用して、特定の NIC ポートの自動 IPv4 アドレス構成を取得します。

```
ilorest set PreBootNetwork=EmbNicPort1 Dhcpv4=Enabled
```

- ・ 特定の NIC ポートの IPv6 自動構成を使用します。例：

```
ilorest set PreBootNetwork=FlexLom1Port1 Ipv6ConfigPolicy=Automatic
```

- ・ システムが、ネットワーク接続されている Dhcpv4 インターフェイスを選択しながらも、静的 IPv4 アドレスを使用するようにします。

```
ilorest set PreBootNetwork=Auto Dhcpv4=Disabled Ipv4Address=192.168.1.66  
Ipv4SubnetMask=255.255.255.0 Ipv4Gateway=192.168.1.1 Ipv4PrimaryDNS=192.168.1.1  
Ipv4SecondaryDNS=192.168.1.2
```

- ・ システムが、ネットワーク接続されている IPv6 インターフェイスを選択しながらも、静的アドレスを使用するようにします。

```
ilorest set PreBootNetwork=Auto Ipv6ConfigPolicy=Manual Ipv6Address=1235::1022  
Ipv6Gateway=1235::1000 Ipv6PrimaryDNS=1235::1000 Ipv6SecondaryDNS=1235::1001
```

3. 変更を有効にするために、次のいずれかを実施します。

- ・ システムを実行している場合は再起動します。
- ・ システムの電源がオフの場合、電源をオンにします。

内蔵 UEFI シェルのブート

システム BIOS には ROM に内蔵 UEFI シェルが含まれています。内蔵 UEFI シェル環境には、**UEFI Shell Specification** に基づいて、スクリプティング、ファイル操作、システム情報の取得に使用できる API と CLI が用意されています。内蔵 UEFI シェルは、他の UEFI アプリケーションも実行します。これらの機能により、UEFI システムユーティリティの機能が強化されます。内蔵 UEFI シェルへのアクセスは、デフォルトで有効になっています。

ここでは、内蔵 UEFI シェルの CLI を使用してシステム BIOS を構成し、OS の展開とインストールを自動化する方法について説明します。

ネットワーク展開用の内蔵 UEFI シェルコマンド

次の表に、ネットワーク展開に使用できる内蔵 UEFI シェルコマンドを示します。各コマンドについて詳しくは、UEFI シェルユーザーガイドを参照してください。

コマンド	説明
ramdisk	RAM ディスクを作成および削除します。
restclient	ローカルの RESTful API サービスと対話します。
sysconfig	システム BIOS 設定を表示または構成します。
webclient	HTTP または FTP からファイルをダウンロードし、ISO ファイルシステムをマウントします。
tlsconfig	TLS 接続の設定と証明書を表示および修正します。

標準形式の出力 (SFO)

汎用的な表ベースの標準出力形式 (SFO) コマンドオプションを使用すると、内蔵 UEFI シェルコマンドの出力処理を容易にすることができます。標準形式の出力を使用するシェルコマンドでは、解析コマンドを使用して解析できるカンマ区切りデータの行と列を使用するコマンドを除き、通常が表示と同じ情報が表示されます。最初の列には、行のデータのタイプを示す C 形式の識別子が必ず含まれます。この識別子は、テーブル名と呼ばれます。 '_' 文字で始まるテーブル名は、実装ごとに異なります。

2 番目以降の列は、実際の内蔵 UEFI シェルコマンドデータを含む引用符付きの C 形式の文字列です。それぞれの内蔵 UEFI シェルコマンドでは、列の形式と意味は列番号とテーブル名によって異なります。

-sfo オプションをサポートする内蔵 UEFI シェルコマンドでは、常にテーブル名 ShellCommand が生成されます。2 番目の列には、拡張を含まない内蔵 UEFI シェルコマンドの名前が含まれています。例：
ShellCommand, "ls"。

例

```
FileInfo, "fs0:/efi/boot/winloader.efi", "45670", "arsh"  
FileInfo, "fs0:/efi/boot/timsfile.txt", "1250", "a"  
FileInfo, "fs0:/efi/boot/readme.txt", "795", "a"
```

拡張構文

以下の構文では、識別子は、英字またはアンダースコアで始まる C 形式の識別子です。引用符で囲まれた文字列は、二重引用符 (") 文字、続いて 0 個以上の文字、最後に二重引用符 (") 文字で終わります。文字列内の引用符は、^文字を使用してエスケープする必要があります (^"など)。^文字は^^を使用して挿入できます。

```
sfo-format := sfo-row
sfo-row <EOL> <sfo-row>
sfo-row := sfo-table-name, sfo-columns
sfo-table-name := identifier
sfo-columns := sfo-column |
sfo-columns, | sfo-column
sfo-column := quoted-string |
<empty>
```

例

```
VolumeInfo, "TimsVolume", "400000000", "32000000", "16000000"
FileInfo, "fs0:/efi/boot/winloader.efi", "45670", "arsh"
FileInfo, "fs0:/efi/boot/timsfile.txt", "1250", "a"
FileInfo, "fs0:/efi/boot/readme.txt", "795", "a"
```

シェルスクリプトの起動

UEFI シェルスクリプトは、次の 2 つの方法のいずれかを使用して起動することができます。

システムユーティリティの UEFI シェルスクリプト自動起動

起動スクリプトを使用すると、RAM ディスクの作成、ネットワークからのファイルのダウンロード、データの収集、ネットワークへの結果のアップロードを行った後、システムを再起動せずに OS を起動できます。ローカルメディア上にスクリプトファイルを保存したり、ネットワーク上からスクリプトファイルにアクセスしたりできます。

デフォルトでは、UEFI シェルスクリプト自動起動がシステムユーティリティで無効になっており、すべての利用可能な FAT16 または FAT32 ファイルシステムでシェルが startup.nsh ファイルを探すように構成できます。これらの設定は、接続されているメディア上の特定のファイルシステムまたはネットワーク上の特定の場所でシェルが起動スクリプトを探すように変更することができます。ネットワーク上の場所を探すように構成した場合、startup.nsh ファイルの場所として HTTP または FTP 形式の URL を指定するか、またはシステムが DHCP を使用してスクリプトの場所 (URL) を取得できるようにすることができます。詳しくは、UEFI システムユーティリティユーザーガイドの「UEFI シェルスクリプト自動起動」を参照してください。

シェルスクリプトの手動開始

手順

- ・ 次のいずれかを実行します。
 - .nsh スクリプトファイルの場所に移動して、スクリプトの名前を入力します。
 - スクリプトのフルパス名を指定します。

シェルスクリプトの編集

edit コマンドを使用してスクリプトファイルをオフラインまたはシェルで編集できます。type コマンドを使用して、スクリプトを画面に出力することもできます。

ネットワークから起動スクリプトを読み取る内蔵 UEFI シェルの構成

ネットワークから起動スクリプトを読み取る内蔵 UEFI シェルを構成できます。UEFI システムユーティリティ、sysconfig コマンド、または RESTful インターフェイスツールを使用して構成することができます。

起動スクリプトを読み取る内蔵 UEFI シェルの構成（システムユーティリティ）

手順

1. プリブートネットワーク構成を完了します。ただし、**URL から起動**フィールドにデータを入力しないでください。このフィールドは、ISO イメージまたは EFI アプリケーションから起動する場合にのみ必要です。[システムユーティリティによるプリブートネットワーク設定の構成](#)を参照してください。
2. UEFI シェルスクリプト自動起動を有効にします。この設定は、デフォルトで無効になっています。UEFI システムユーティリティユーザーガイドの「UEFI シェルスクリプト自動起動」を参照してください。
3. 自動起動のロケーションをネットワークに設定します。UEFI システムユーティリティユーザーガイドの「シェル自動起動スクリプトロケーション」を参照してください。
4. 内蔵 UEFI シェルスクリプトのパスが含まれている場所（URL）を設定します。UEFI システムユーティリティユーザーガイドの「シェル自動起動スクリプトのためのネットワーク上の場所」を参照してください。または、「DHCP を使用したシェル自動起動スクリプトの検出」を有効にして、シェルが DHCP を使用してスクリプトの URL を検出できるようにすることができます。詳しくは、HPE ProLiant Gen10 サーバーおよび HPE Synergy 用 UEFI システムユーティリティガイドで「DHCP を使用した、シェル自動起動スクリプトの検出の有効化または無効化」を参照してください。

注記: 内蔵 UEFI シェルスクリプト（.nsh）をポイントするように URL を静的に設定します。その他のファイルはサポートされません。

注記: HTTPS の URL を使用する場合は、サーバー証明書がシステムに登録されていることを確認します。tlsconfig シェルコマンドまたは **RBSU > サーバーセキュリティ > TLS (HTTPS) オプション**を使用します。

注記: URL が HTTP プロキシ経由でアクセスできる場合、**RBSU > ネットワークオプション > プリブートネットワーク設定**の下にある**プリブートネットワークプロキシオプション**を設定します。

起動スクリプトを読み取る内蔵 UEFI シェルの構成（sysconfig コマンド）

手順

1. システムユーティリティ画面で、**内蔵アプリケーション > 内蔵 UEFI シェル**を選択します。[sysconfig コマンドによるプリブートネットワーク設定の構成](#)を参照してください。
2. 次の例に示すように、シェルプロンプトでいずれかの sysconfig コマンドを入力します。

- ・ 静的な URL を設定するには、以下を入力します。

```
Shell> sysconfig -s UefiShellStartup=Enabled
UefiShellStartupLocation=NetworkLocation
UefiShellStartupUrl=https://deploy.server.com/scripts/startup.nsh
```

- ・ DHCP を使用して URL を検出するには、以下を入力します。

```
Shell> sysconfig -s UefiShellStartup=Enabled
UefiShellStartupLocation=NetworkLocation
UefiShellStartupUrlFromDhcp=Enabled
```

起動スクリプトを読み取る内蔵 UEFI シェルの構成 (RESTful インターフェイスツール)

手順

1. RESTful インターフェイスツールを実行します。

```
ilorest login proliantsystem1.hpe.com -u username -p password
ilorest select Bios.v1_0_0
```

2. プリブートネットワーク設定をセットアップします。RESTful インターフェイスツールによるプリブートネットワーク設定の構成を参照してください。
3. コマンドラインプロンプトで、次のいずれかを入力します。

```
· ilorest UefiShellStartupLocation=NetworkLocation
  UefiShellStartupUrl=https://deploy.server.com/scripts/startup.nsh

· ilorest UefiShellStartupLocation=NetworkLocation
  UefiShellStartupUrlFromDhcp=Enabled
```

4. 変更を有効にするために、次のいずれかを実施します。

- ・ システムを実行している場合は再起動します。
- ・ システムの電源がオフの場合、電源をオンにします。

RAM ディスクの作成

RAM ディスクとは、任意のストレージデバイスのようにアクセス可能な(ブロック/ストレージデバイスが RAM でエミュレートされる)「メモリ内」ファイルシステムです。

ボリュームラベル MYRAMDISK 付きの 512 MB の FAT32 RAM ディスクを作成するには、内蔵 UEFI シェルプロンプトで次のコマンドを入力します。

```
Shell> ramdisk -c -s 512 -v MYRAMDISK -t F32
```

詳細については、`help ramdisk`, `or ramdisk -?`と入力します。

このコマンドについて詳しくは、UEFI シェルユーザーガイドを参照してください。

展開とスクリプティング

構成手順が完了して内蔵 UEFI シェルから起動すると、シェルは構成されている接続メディアまたはネットワークの場所にある `startup.nsh` ファイルを自動的に見つけます。これはネットワークからスクリプトをダウンロードする RAM ディスクをプロビジョニングします。次に、起動スクリプトは自動的にシステム構成を実行し、OS の展開とインストールのプロセスを開始します。

内蔵 UEFI シェルのブートの構成：サンプルの展開ソリューション

手順

1. システムユーティリティ、`sysconfig` コマンド、または RESTful インターフェイスツールを使用して、プリブートネットワーク設定を構成します。

システムユーティリティによるプリブートネットワーク設定の構成、sysconfig コマンドによるプリブートネットワーク設定の構成、または RESTful インターフェイスツールによるプリブートネットワーク設定の構成を参照してください。

2. システムユーティリティ、sysconfig コマンド、または RESTful インターフェイスツールを使用して、起動スクリプトを読み取る内蔵 UEFI シェル設定を構成します。

起動スクリプトを読み取る内蔵シェルの構成 (システムユーティリティ)、起動スクリプトを読み取る内蔵 UEFI シェルの構成 (sysconfig コマンド)、または起動スクリプトを読み取る内蔵 UEFI シェルの構成 (RESTful インターフェイスツール) を参照してください。

3. 次のいずれかを実行します。
 - a. ワンタイムブートメニューから内蔵 UEFI シェルを選択します。ワンタイムブートのオプションの選択 (10 ページ)を参照してください。
 - b. 内蔵 UEFI シェルを常に最初に起動する場合は、UEFI ブート順序リストの一番上に移動します。UEFI ブート順序リストの変更 (8 ページ)を参照してください。

サンプル起動スクリプト

RESTful インターフェイスツールを使用して内蔵 UEFI シェルのブートを構成した場合、システムが起動し、必要な設定でシステム自体を構成して再起動します。システムは次に内蔵 UEFI シェルを起動し、このシェルによって構成済みの URL から起動スクリプトをダウンロードし、実行します。

システムユーティリティを使用して内蔵 UEFI シェルのブートを構成した場合は、すぐに内蔵シェルを起動できます。内蔵 UEFI シェルは URL から起動スクリプトをダウンロードし、実行します。

```
@echo -off

#
# Setup the environment variables. All of them are created as volatile.
#

#
# The volume label for the RAMDISK.
#
set -v VolumeLabel MYRAMDISK

#
# Variable to store the file system index that will be looped
# to determine the FS<x> number for the RAMDISK that is created.
#
set -v FsIndex 0

#
# Variable to store the output string of the ramdisk -c command.
# Successful creation of RAMDISK will give the following output:
# "RAM disk 'FSx:' created successfully." where x=0,1,2,...
#
set -v RamDiskStr 0

#
# Size of the RAMDISK in MegaBytes (MB).
#
set -v RamDiskSize 512

#
# Server URL hosting the OS loader and images.
# Can be HTTP or FTP. Names or IP addresses are allowed.
# Ensure DNS service is available and configured (see pre-requisites)
# when server names are used.
#
set -v Url https://192.168.1.1
```

```

#
# Files to be downloaded
#
set -v DownloadFile1 efilinux.efi
set -v DownloadFile2 deploy.kernel
set -v DownloadFile3 deploy.ramdisk

#
# Step 1. Create RAMDISK to store the downloaded OS programs.
#
echo "Creating a RAM Disk to save downloaded files..."
ramdisk -c -s %RamDiskSize% -v %VolumeLabel% -t F32 >v RamDiskStr
if %lasterror% ne 0x0 then
    echo "Cannot create a RAMDISK of size %RamDiskSize%."
    goto EXITSCRIPT
endif
echo "RAM Disk with Volume Label %VolumeLabel% created successfully."

#
# Step 2. Check each word in the output (RamDiskStr) and see if it matches
# the FSx: pattern. The newly created RAMDISK will be FS1: or higher.
# Here the check goes up to FS3: (the inner for loop), but a larger limit
# may be used in case many other file systems already exist before
# the creation of this RAMDISK. The FS for the RAMDISK is found when the
# FsIndex matches the FS<x> in RamDiskStr. Change the working directory
# to FS<FsIndex>:, so all downloads get saved there.
#
# FS0: is ignored. In the worst case, when no other usable
# file system is present, FS0: will map to the file system
# that this script is executing from.
#
#
for %a in %RamDiskStr%
    for %b run (1 10)
        set -v FsIndex %b
        if 'FS%FsIndex%:' == %a then
            FS%FsIndex%:
            goto RDFOUND
        endif
    endfor
endfor

#
# The following message appears if the newly created RAMDISK cannot be found.
#
echo "RAMDISK with Volume Label %VolumeLabel% not found!"
goto EXITSCRIPT

#
# The following message appears if the RAMDISK FS<x> has been found and you are in the
# RAMDISK's root folder.
#
:RDFOUND
echo "RAMDISK with Volume Label %VolumeLabel% found at FS%FsIndex%:."

#
# Step 3: Download the required files into the RAMDISK.
#
echo "Downloading %Url%/deploy/%DownloadFile1% (File 1 of 3...)"
webclient -g %Url%/deploy/%DownloadFile1% -o %DownloadFile1%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile2% (File 2 of 3...)"
webclient -g %Url%/deploy/%DownloadFile2% -o %DownloadFile2%
if %lasterror% ne 0x0 then

```

```

goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile3% (File 3 of 3...)"
webclient -g %Url%/deploy/%DownloadFile3% -o %DownloadFile3%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

#
# Step4: Launch the boot loader.
#
echo "Starting the OS..."
%DownloadFile1% -f %DownloadFile2% initrd=%DownloadFile3%

#
# You reach here only if the downloads and booting failed.
#
:EXITSCRIPT
echo "Exiting Script."

```

このサンプルスクリプトは、以下の操作を行います。

1. ダウンロードしたブートローダー、OS カーネル、ファイルシステム、およびブートローダーとカーネルを初期化してネットワーク経由でインストールを続行するために必要なすべての構成ファイルを保存するための一時的な RAM ディスクを作成します。
2. 作成した RAM ディスクの FS<x> ID を決定します。
3. 作業ディレクトリを RAM ディスクのルートに設定します。例 : FS1:\.
4. OS の起動に必要なファイル（ブートローダー、OS カーネル、OS カーネルのメモリ内ファイルシステム）をダウンロードします。
5. 次のいずれかを実行します。
 - a. ダウンロードが成功した場合、ブートローダーを起動します。OS カーネルファイルへのパス、そのインメモリファイルシステム、および（起動の際にブートローダーがカーネルに引き渡す）OS カーネルの引数が、コマンドライン引数としてブートローダーに渡されます。
 - b. ダウンロードが失敗した場合は、クリーンアップを実行し、スタートアップスクリプトを終了します。

内蔵 UEFI シェルとプリブートスクリプトの役割は、ここで終了し、OS は、そのメモリ内ファイルシステムに組み込まれている OS 固有の展開スクリプトを使用して、単独で展開を続行できます。

UEFI モードの FC/FCoE SAN ブート

FC/FCoE SAN ブート構成

システムユーティリティのデフォルトでは、ターゲットのデバイス設定でインストール済みの FC/FCoE アダプターがあらかじめ構成されているブートターゲットのみスキャン可能です。ブートポリシーの設定は変更することができます。変更後、UEFI モードまたはレガシー BIOS モードのいずれかで FC/FCoE SAN から起動することができます。UEFI モードでのブート方法を以下に説明します。

注記: ブート時間が長くなるように、使用可能なすべてのターゲットをスキャンする FC スキャンポリシーを設定しないようにしてください。お使いの特定のアダプターまたはシステムのユーティリティでグローバルスキャンポリシーを設定することで、UEFI システムユーティリティのオプションでこの設定を無効にできます。

FC/FCoE SAN からの起動

システムユーティリティを使用して FC/FCoE SAN から起動するには、次の操作を行います。

前提条件

- ・ EFI システムパーティション (ESP) とブートローダーファイルがデフォルトの場所 (`\efi\BOOT\BOOTX64.EFI`) に保存されている。
- ・ SAN ブートターゲットが GPT (GUID パーティションテーブル) 構成である。
- ・ ブートモードが **UEFI モード** に設定されている。

手順

1. 必要に応じて、以下のいずれかを実行してブートポリシーを変更します。
 - a. PCIe スロットにないファイバーチャネルコントローラーについては、HPE ProLiant Gen10 サーバーおよび HPE Synergy 用 UEFI システムユーティリティユーザーガイドの「ファイバーチャネル/FCoE スキャンポリシー」の説明に従って設定を変更してください。
 - b. PCIe スロットにあるファイバーチャネルコントローラーについては、HPE ProLiant Gen10 サーバーおよび HPE Synergy 用 UEFI システムユーティリティユーザーガイドの「PCIe ストレージブートポリシー」の説明に従って設定を変更してください。
2. **UEFI ブート順序リストの変更**の説明に従って、ブート順序リストの一番上にこのブートオプションを移動します。
3. サーバーを再起動します。

UEFI モードでのローカル HDD のブート

HDD ブート構成

UEFI モードまたはレガシー BIOS モードのいずれかで、内蔵 SATA または内蔵 (SAS ベース) Smart アレイ コントローラーに接続されているローカルハードディスクドライブを含む、ローカルハードディスクドライブから起動することができます。以下で、UEFI モードのローカル HDD ブートについて説明します。

内蔵 SATA のサポート

内蔵 SATA に接続されているローカルの SATA ハードディスクドライブから起動するには、インストールされている OS に対して適切な SATA コントローラーを有効にする必要があります。ご使用のサーバーモデルに応じて、SATA AHCI または Dynamic Smart アレイ RAID サポートのいずれかがデフォルトで有効化されています。

△ 注意: ご使用のブートモードがレガシー BIOS モードに構成されているときは、Dynamic Smart アレイはサポートされません。Dynamic Smart アレイ RAID を有効にする場合、既存の SATA ドライブでのデータの消失またはデータの破損を防止するために UEFI モードを使用する必要があります。このオプションを有効にする前にすべてのドライブをバックアップしてください。

SATA AHCI サポートを有効にする前に、ご使用のオペレーティングシステムのドキュメントを参照して、ベースメディアのドライバーがこの機能をサポートしていることを確認します。

内蔵 SATA のサポートの有効化

前提条件

2.2 TB を超えるパーティションでは、ブートモードは UEFI モードに設定する必要があります。

手順

1. システムユーティリティ画面で、**システム構成 > BIOS/プラットフォーム構成 (RBSU) > システムオプション > SATA コントローラーオプション > 内蔵 SATA 構成**を選択します。
2. SATA オプション用に正しい AHCI または RAID システムドライバーを使用しているか確認します。
3. 設定を選択します。
 - a. **SATA AHCI サポートを有効** - AHCI 用の内蔵チップセット SATA コントローラーを有効化します。
 - b. **HP Dynamic Smart Array RAID サポートを有効** - Dynamic Smart アレイ RAID 用の内蔵チップセット SATA コントローラーを有効化します。
4. 設定を保存します。
5. UEFI モードでは、**UEFI ブート順序リストの変更**の説明に従って、UEFI ブート順序リストの一番上にこの UEFI ブートオプションを移動します。
6. サーバーを再起動します。

トラブルシューティング

ネットワークブート URL のファイルをダウンロードできない

症状

ネットワークブート用として指定した URL のファイルをダウンロードしようとする時、エラーメッセージが表示されます。

解決方法 1

原因

静的構成時に指定したネットワーク接続設定が正しくありません。

アクション

1. 内蔵 UEFI シェルの ping コマンドを使用してネットワーク接続をチェックします。UEFI シェルユーザガイドの「Ping」を参照してください。
2. 静的なネットワーク接続設定を変更し、URL にあるファイルをもう一度ダウンロードします。

解決方法 2

原因

DHCP サーバーが応答しませんでした。

アクション

1. DHCP サーバーが利用可能で動作可能か確認してください。
2. URL のファイルをもう一度ダウンロードします。

解決方法 3

原因

選択した NIC ポートにケーブルが接続されていません。

アクション

1. ケーブル接続があることを確認します。
2. URL のダウンロードを再試行します。

解決方法 4

原因

ファイルが正しくないかサーバーに存在しない、または必要な権限がないためダウンロードできません。ファイル名をチェックし、サーバーに存在していることを確認します。サーバーからダウンロードする権限があることを確認します。

アクション

1. ファイルが存在し、正しいファイル名を使用し、ダウンロードに十分な権限があるか確認します。
2. URL のファイルをもう一度ダウンロードします。

解決方法 5

原因

HTTP サーバーが停止しているか、または応答していません。

アクション

1. 指定した HTTP サーバーが使用可能であり、動作中であることを確認します。
2. URL のファイルをもう一度ダウンロードします。

UEFI シェルスクリプトから展開できない

症状

UEFI シェルスクリプトを使用して OS を展開しようとする、展開が失敗したことを示すエラーメッセージが表示されます。

原因

構成設定が正しくありません。

アクション

1. 以下を確認します。
 - a. 内蔵 UEFI シェルインターフェイスが **UEFI ブート順序リスト** または **ワンタイムブートメニュー** に追加されている。
 - b. **UEFI ブート順序リスト** に追加すると、**UEFI ブート順序リスト** で内蔵 UEFI シェルインターフェイスが最初のオプションになるため、他のブートオプションよりも優先してロードされる。
 - c. UEFI シェルスクリプト自動起動が有効になっている。
 - d. 接続されているメディアの `startup.nsh` スクリプトファイルの場所またはネットワーク上の場所が正しく指定されている。接続されているメディア内の場合は、`startup.nsh` スクリプトが `fs X:\` ディレクトリ内になければなりません。
 - e. `.nsh` スクリプトに、サポートされているコマンドのみが含まれている。
 - f. 使用しているシステムに、自動スクリプトの実行中に RAM ディスクを作成するための十分な RAM メモリがある。
 - g. `.nsh script` を使用して起動された OS ブートローダーや診断アプリケーションの UEFI 環境での実行がサポートされている。
2. 展開をやり直します。

Web サイト

全般的な Web サイト

Hewlett Packard Enterprise Information Library

<http://www.hpe.com/info/EIL>

Single Point of Connectivity Knowledge (SPOCK) ストレージ互換性マトリックス RESTful インターフェイスツールのドキュメント

<http://www.hpe.com/storage/spock>

ストレージのホワイトペーパーおよびアナリストレポート

<http://www.hpe.com/storage/whitepapers>

UEFI の仕様

<http://www.uefi.org/specifications>

UEFI の学習資料

http://www.uefi.org/learning_center

RESTful API ツール

<http://www.hpe.com/info/redfish-ja>

Hewlett Packard Enterprise Worldwide の連絡先

<http://www.hpe.com/assistance>

サブスクリプションサービス/サポートのアラート

<http://www.hpe.com/support/e-updates-ja>

Software Depot

<http://www.hpe.com/support/softwaredepot>

カスタマーセルフリペア

<http://www.hpe.com/support/selfrepair>

Insight Remote Support

<http://www.hpe.com/info/insightremotesupport/docs>

上記以外の Web サイトについては、[サポートと他のリソース](#)を参照してください。

サポートと他のリソース

Hewlett Packard Enterprise サポートへのアクセス

- ・ ライブアシスタンスについては、Contact Hewlett Packard Enterprise Worldwide の Web サイトにアクセスします。

<http://www.hpe.com/info/assistance>

- ・ ドキュメントとサポートサービスにアクセスするには、Hewlett Packard Enterprise サポートセンターの Web サイトにアクセスします。

<http://www.hpe.com/support/hpesc>

ご用意いただく情報

- ・ テクニカルサポートの登録番号（該当する場合）
- ・ 製品名、モデルまたはバージョン、シリアル番号
- ・ オペレーティングシステム名およびバージョン
- ・ ファームウェアバージョン
- ・ エラーメッセージ
- ・ 製品固有のレポートおよびログ
- ・ アドオン製品またはコンポーネント
- ・ 他社製品またはコンポーネント

アップデートへのアクセス

- ・ 一部のソフトウェア製品では、その製品のインターフェイスを介してソフトウェアアップデートにアクセスするためのメカニズムが提供されます。ご使用の製品のドキュメントで、ソフトウェアの推奨されるソフトウェアアップデート方法を確認してください。
- ・ 製品のアップデートをダウンロードするには、以下のいずれかにアクセスします。

Hewlett Packard Enterprise サポートセンター

<http://www.hpe.com/support/hpesc>

Hewlett Packard Enterprise サポートセンター：ソフトウェアのダウンロード

<http://www.hpe.com/support/downloads>

Software Depot

<http://www.hpe.com/support/softwaredepot>

- ・ eNewsletters およびアラートをサブスクライブするには、以下にアクセスします。

<http://www.hpe.com/support/e-updates-ja>

- ・ お客様の資格を表示したりアップデートしたり、契約や保証をお客様のプロファイルにリンクしたりするには、Hewlett Packard Enterprise サポートセンターの **More Information on Access to Support Materials** ページにアクセスします。

<http://www.hpe.com/support/AccessToSupportMaterials>

- ❶ **重要:** 一部のアップデートにアクセスするには、Hewlett Packard Enterprise サポートセンターからアクセスするときに製品資格が必要になる場合があります。関連する資格を使って HPE パスポートをセットアップしておく必要があります。

カスタマーセルフリペア (CSR)

Hewlett Packard Enterprise カスタマーセルフリペア (CSR) プログラムでは、ご使用の製品をお客様ご自身で修理することができます。CSR 部品を交換する必要がある場合、お客様のご都合のよいときに交換できるよう直接配送されます。一部の部品は CSR の対象になりません。Hewlett Packard Enterprise もしくはその正規保守代理店が、CSR によって修理可能かどうかを判断します。

リモートサポート (HPE 通報サービス)

リモートサポートは、保証またはサポート契約の一部としてサポートデバイスでご利用いただけます。リモートサポートは、インテリジェントなイベント診断を提供し、ハードウェアイベントを Hewlett Packard Enterprise に安全な方法で自動通知します。これにより、ご使用の製品のサービスレベルに基づいて、迅速かつ正確な解決が行われます。ご使用のデバイスをリモートサポートに登録することを強くおすすめします。

ご使用の製品にリモートサポートの追加詳細情報が含まれる場合は、検索を使用してその情報を見つけてください。

リモートサポートおよびプロアクティブケア情報

HPE 通報サービス

<http://www.hpe.com/jp/hpalert>

HPE プロアクティブケアサービス

<http://www.hpe.com/services/proactivecare-ja>

HPE プロアクティブケアサービス：サポートされている製品のリスト

<http://www.hpe.com/services/proactivecaresupportedproducts>

HPE プロアクティブケアアドバンスドサービス：サポートされている製品のリスト

<http://www.hpe.com/services/proactivecareadvancedsupportedproducts>

保証情報

ご使用の製品の保証に関する情報を表示するには、以下のリンクを参照してください。

HPE ProLiant と IA-32 サーバーおよびオプション

<http://www.hpe.com/support/ProLiantServers-Warranties>

HPE Enterprise および Cloudline サーバー

<http://www.hpe.com/support/EnterpriseServers-Warranties>

HPE ストレージ製品

<http://www.hpe.com/support/Storage-Warranties>

HPE ネットワーク製品

<http://www.hpe.com/support/Networking-Warranties>

規定に関する情報

安全、環境、および規定に関する情報については、Hewlett Packard Enterprise サポートセンターからサーバー、ストレージ、電源、ネットワーク、およびラック製品の安全と準拠に関する情報を参照してください。

<http://www.hpe.com/support/Safety-Compliance-EnterpriseProducts>

規定に関する追加情報

Hewlett Packard Enterprise は、REACH（欧州議会と欧州理事会の規則 EC No 1907/2006）のような法的な要求事項に準拠する必要に応じて、弊社製品の含有化学物質に関する情報をお客様に提供することに全力で取り組んでいます。この製品の含有化学物質情報レポートは、次を参照してください。

<http://www.hpe.com/info/reach>

RoHS、REACH を含む Hewlett Packard Enterprise 製品の環境と安全に関する情報と準拠のデータについては、次を参照してください。

<http://www.hpe.com/info/ecodata>

社内プログラム、製品のリサイクル、エネルギー効率などの Hewlett Packard Enterprise の環境に関する情報については、次を参照してください。

<http://www.hpe.com/info/environment>

ドキュメントに関するご意見、ご指摘

Hewlett Packard Enterprise では、お客様により良いドキュメントを提供するように努めています。ドキュメントを改善するために役立てさせていただきますので、何らかの誤り、提案、コメントなどがございましたら、ドキュメントフィードバック担当 (docsfeedback@hpe.com) へお寄せください。この電子メールには、ドキュメントのタイトル、部品番号、版数、およびドキュメントの表紙に記載されている刊行日をご記載ください。オンラインヘルプの内容に関するフィードバックの場合は、製品名、製品のバージョン、ヘルプの版数、およびご利用規約ページに記載されている刊行日もお知らせください。