




Dell Lifecycle Controller 2 Remote Services

バージョン 1.00.00 ユーザーズガイド



メモ、注意、警告

-  **メモ:** コンピュータを使いやすくするための重要な情報を説明しています。
-  **注意:** 手順に従わない場合、ハードウェア損傷やデータ損失の可能性を示しています。
-  **警告:** 物的損害、けが、または死亡の原因となる可能性を示しています。

この文書の情報は、事前の通知なく変更されることがあります。

© 2012 すべての著作権は Dell Inc. にあります。

Dell Inc. の書面による許可のない複製は、いかなる形態においても厳重に禁じられています。

本書に使用されている商標 : Dell™、Dell ロゴ、Dell Precision™、OptiPlex™、Latitude™、PowerEdge™、PowerVault™、PowerConnect™、OpenManage™、EqualLogic™、Compellent™、KACE™、FlexAddress™、Force10™ および Vostro™ は Dell Inc. の商標です。Intel®、Pentium®、Xeon®、Core® および Celeron® は米国およびその他の国における Intel Corporation の登録商標です。AMD® は Advanced Micro Devices, Inc. の登録商標、AMD Opteron™、AMD Phenom™ および AMD Sempron™ は同社の商標です。Microsoft®、Windows®、Windows Server®、Internet Explorer®、MS-DOS、Windows Vista®、および Active Directory® は米国および/またはその他の国における Microsoft Corporation の商標または登録商標です。Red Hat® および Red Hat® Enterprise Linux® は米国および/またはその他の国における Red Hat, Inc. の登録商標です。Novell® および SUSE® は米国およびその他の国における Novell, Inc. の登録商標です。Oracle® は Oracle Corporation またはその関連会社、もしくはその両者の登録商標です。Citrix®、Xen®、XenServer® および XenMotion® は米国および/またはその他の国における Citrix Systems, Inc. の登録商標または商標です。VMware®、Virtual SMP®、vMotion®、vCenter® および vSphere® は米国またはその他の国における VMware, Inc. の登録商標または商標です。IBM® は International Business Machines Corporation の登録商標です。

商標または製品の権利を主張する事業体を表すために、その他の商標および社名が使用されていることがあります。それらの商標や会社名は、一切 Dell Inc. に帰属するものではありません。

2012 - 03

Rev. A00

目次

メモ、注意、警告.....	2
章 1: はじめに.....	9
iDRAC7 With Lifecycle Controller を使用するメリット.....	9
主な機能.....	9
Remote Services を使用する理由.....	10
Lifecycle Controller のライセンス可能機能.....	10
Web Services for Management.....	11
標準 DMTF.....	11
Dell 拡張.....	11
その他の必要マニュアル.....	12
デルへのお問い合わせ.....	13
章 2: Remote Services の使用.....	15
Remote Services 使用前の共通前提条件.....	15
Web サービスのセットアップ.....	15
WinRM クライアント.....	15
OpenWSMan クライアント.....	16
使用事例の使い方.....	16
使用事例の構造.....	16
使用事例の読み方.....	16
使用事例シナリオ.....	16
章 3: 自動検出とハンドシェイク.....	19
自動検出のための iDRAC の設定.....	19
プロビジョニングサーバーの文字列フォーマット.....	20
必要な場所でのプロビジョニングの設定.....	20
管理下システムの自動検出.....	21
DHCP または DNS の設定.....	21
前面パネルディスプレイでの検出ステータスの表示.....	21
新しい環境での自動検出の再開.....	22
章 4: ライセンスの管理.....	23
インストールされたライセンスの表示.....	23
ライセンス可能デバイスの表示.....	23
ライセンスのインストール.....	23
ライセンスのインストールのための参照資料.....	24
ライセンスの交換.....	24

ライセンスの削除.....	24
ライセンスのエクスポート.....	24
章 5: 証明書の管理.....	27
プロビジョニングサーバー用の Trusted Root クライアントカスタム証明書の作成.....	27
カスタムサーバー証明書の提供.....	27
カスタム証明書の削除.....	27
カスタムサーバーパブリックキーの削除.....	28
カスタムクライアント証明書の削除.....	28
ウェブサーバーまたは WS-Management 暗号化証明書とプライベートキーの PKCS 12 からの 変更.....	28
サーバー証明書の管理.....	28
サーバー証明書の管理のための参照資料.....	29
ディレクトリ CA 証明書の管理.....	29
ディレクトリ CA 証明書の管理のための参照資料.....	29
章 6: オペレーティングシステムの展開.....	31
オペレーティングシステムの展開.....	31
オペレーティングシステムの展開のための参照資料.....	32
リモートファイル共有の使用.....	33
サーバーメンテナンス中の ISO からの起動.....	34
サーバーメンテナンス中の ISO からの起動のための参照資料.....	35
ISO からの起動メソッドの比較.....	35
1 回限りの起動.....	36
ジョブ ID について.....	37
章 7: ジョブの管理.....	39
ジョブタイプ.....	39
ユーザー作成ジョブ.....	40
ジョブスケジュール.....	40
ジョブの削除.....	40
複数のアクションへの個別ジョブのスケジュール.....	40
複数のターゲットジョブの実行.....	41
開始時間と終了時間の指定.....	41
ジョブの自動削除.....	41
すべてのジョブをクリア.....	42
章 8: RAID 設定の管理.....	43
RAID コントローラの表示.....	43
スライスされた仮想ディスクの作成.....	43
RAID の設定.....	43
RAID のセットアップ - 設定後のシナリオ.....	46

RAID 設定のための参照資料.....	46
SATA ドライブの RAID モードから非 RAID 状態への変換.....	47
SATA ドライブの変換のための参照資料.....	47
章 9: ネットワークデバイスの管理.....	49
ネットワークデバイスインベントリの表示.....	49
ネットワークデバイス属性の表示.....	49
ネットワークデバイス属性の設定.....	49
保留中の値の削除.....	50
CNA のパーティションの有効化または無効化.....	50
CNA 用パーティションのパーソナリティと帯域幅の変更.....	51
パーソナリティの変更のための参照資料.....	52
仮想アドレス属性の設定.....	52
仮想アドレス属性の参照資料.....	53
起動ターゲット iSCSI および FCoE の設定.....	53
章 10: インベントリとログ.....	55
ハードウェアインベントリの取得.....	55
現在のハードウェアインベントリのエクスポート.....	55
ライフサイクルログ.....	55
ライフサイクルログのエクスポート.....	56
設定の削除とデフォルトへのリセット.....	56
章 11: リモートアップデート.....	57
リモートアップデートの使用.....	57
対応デバイス.....	57
URI からのリモートアップデート.....	58
リモートアップデートのスケジュール作成.....	58
前のバージョンへのロールバック.....	59
リモートファームウェアインベントリの使用.....	59
対応デバイス.....	59
ファームウェアインベントリの取得.....	60
リモートスケジュールのタイプ.....	60
即時アップデート.....	60
スケジュール指定アップデート.....	61
スケジュール再起動動作の設定.....	61
部品交換の管理.....	61
部品ファームウェアと設定アップデート属性の取得と設定.....	61
章 12: バックアップと復元.....	63
iDRAC vFlash カードまたはネットワーク共有へのサーバープロファイルのエクスポート.....	63
サーバープロファイルのエクスポートの機能とシステム動作.....	64

サーバープロファイルのエクスポートのための参照資料.....	65
iDRAC vFlash カードまたはネットワーク共有からのサーバープロファイルのインポート.....	65
復元後のシナリオ.....	67
復元後のシナリオの機能とシステム動作.....	67
サーバープロファイルのインポートのための参照資料.....	68
章 13: vFlash SD カードの管理.....	69
vFlash SD カードのインベントリの表示.....	69
vFlash SD カードのパーティションの表示.....	69
vFlash SD カードのパーティションの作成と修正.....	69
章 14: iDRAC 設定.....	71
iDRAC の属性の取得と設定.....	71
iDRAC 属性の取得と設定のための参照資料.....	71
iDRAC 属性.....	72
iDRAC ユーザーと役割の取得と設定.....	74
iDRAC ユーザーと役割の取得と設定のための参照資料.....	75
iDRAC の IP アドレス変更の報告.....	75
iDRAC IP アドレス変更の報告の機能とシステム動作.....	75
iDRAC IP アドレス変更の報告のための参照資料.....	76
章 15: BIOS と起動設定の管理.....	77
BIOS 属性のインベントリの表示.....	77
BIOS 属性の設定.....	77
1 回限りの起動.....	77
BIOS パスワードの設定、変更、削除.....	78
BIOS パスワードの設定、変更、削除のための参照資料.....	79
章 16: その他の使用事例のシナリオ.....	81
Remote Services ステータスの取得.....	81
Remote Services ステータスの取得のための参照資料.....	81
章 17: Remote Services プロファイル.....	83
オペレーティングシステム展開プロファイル.....	83
オペレーティングシステム展開メソッド.....	83
Lifecycle Controller 管理プロファイル.....	84
LC サービスメソッド.....	84
自動検出メソッド.....	84
エクスポートとインポートメソッド.....	84
ライフサイクルログメソッド.....	85
ハードウェアインベントリメソッド.....	85
シンプル NIC プロファイル.....	85

シンプル NIC メソッド.....	86
BIOS と起動管理プロファイル.....	86
BIOS と起動管理メソッド.....	87
永続ストレージプロファイル.....	87
vFlash SD カードメソッド.....	88
RAID プロファイル.....	88
RAID メソッド.....	89
ハードウェアインベントリプロファイル.....	90
ジョブ制御プロファイル.....	91
ジョブ制御メソッド.....	91
電源装置プロファイル.....	92
電源状態管理プロファイル.....	92
電源状態管理プロファイルメソッド.....	92
レコードログプロファイル.....	92
レコードログプロファイルメソッド.....	93
役割ベース認証プロファイル.....	93
役割ベース認証プロファイルメソッド.....	93
センサープロファイル.....	94
サービスプロセッサプロファイル.....	94
サービスプロセッサプロファイルメソッド.....	94
イベントフィルタプロファイル.....	94
イベントフィルタプロファイルメソッド.....	94
ライセンス管理プロファイル.....	95
ライセンス管理プロファイルメソッド.....	95
iDRAC カードプロファイル.....	95
iDRAC カードプロファイルメソッド.....	95
ベースサーバーと物理的資産プロファイル.....	96
ベースサーバーと物理的資産プロファイルメソッド.....	96
システム情報プロファイル.....	96
システム情報メソッド.....	96
シンプル ID 管理プロファイル.....	97
シンプル ID メソッド.....	97
章 18: トラブルシューティングとよくあるお問い合わせ (FAQ)	99
エラーメッセージ.....	99
自動検出 LCD メッセージ.....	99
よくあるお問い合わせ.....	100
章 19: スキーマ	103
ライフサイクルログスキーマ.....	103
章 20: 使いやすいシステムコンポーネント名	105

はじめに

Dell Lifecycle Controller は、高度な内蔵システム管理を提供します。iDRAC 機能に加え、システム管理機能が組み込まれた 1GB の管理下および永続ストレージが含まれます。

Dell Lifecycle Controller Remote Services はさらに、1 対多メソッドでのリモートシステム管理を可能にします。Remote Services は、iDRAC を介したリモートサーバープロビジョニングと管理のため、Web Service for Management (WS-Management) プロトコルをベースとした Web サービスインタフェースを使用して利用できます。このインタフェースは、リモートでのオペレーティングシステム (OS) 展開、リモートアップデート、およびリモートインベントリを含む多くのタスクの簡易化と、新規または展開済みの Dell システムのセットアップと設定のリモートでの自動化を目的としています。

Remote Services には、セキュアな Web サービスインタフェースを使用してネットワークでアクセスでき、アプリケーションやスクリプトによるプログラムでの活用が可能です。Remote Services では、管理コンソールから 1 対多のベアメタルサーバープロビジョニングを実行できます。ネットワークに接続されている Dell システムを識別して認証する新しい自動検出機能と、1 対多の管理コンソールの統合との組み合わせは、サーバープロビジョニングに必要な手動での手順を削減します。

iDRAC7 With Lifecycle Controller を使用するメリット

次のメリットが挙げられます。

- 可用性の向上 — 不具合発生からの復帰時間を短縮するために役立つ、エラーの可能性または実際のエラーの早期通知を行います。
- 生産性の向上および総所有コスト (TCO) の削減 — 遠隔地に多数存在するサーバーへの管理者の管理範囲を拡大は、交通費などの運用コストを削減しながら IT スタッフの生産性を向上させることができます。
- セキュアな環境 — リモートサーバーへのセキュアなアクセスを提供することにより、管理者はサーバーおよびネットワークのセキュリティを維持しながら、重要な管理作業を行うことができます。
- Lifecycle Controller による内蔵システム管理の強化 — ローカル展開においては Lifecycle Controller の GUI による展開および保守性の簡略化を提供し、リモート展開においては Dell OpenManage Essentials およびパートナーコンソールと統合された Remote Services (WS-Management) インターフェースを提供します。

iDRAC7 の詳細については、support.dell.com/manuals で、『*Integrated Dell Remote Access Controller ユーザーズガイド*』を参照してください。

主な機能

リモートサービスでは、Dell Management Console、Dell Modular Chassis Management Controller、パートナーコンソール、カスタマーの自作コンソール、およびスクリプトで次のようなシステム管理タスクをリモートで実行できるようになります。

- オペレーティングシステムおよびドライバのインストール
- ライセンスの管理
- BIOS ファームウェアのアップデートの実行
- 部品交換の管理
- コンポーネントファームウェアのアップデートの実行

- ハードウェアインベントリ情報の取得
- NIC/CNA および RAID 設定の取得と設定
- BIOS 設定および BIOS パスワードの取得と設定
- ライフサイクルログのエクスポート
- 現在および工場出荷時のハードウェアインベントリログのエクスポート
- vFlash SD カードパーティションの管理、接続、および起動
- ローカルキーを使用したコントローラーの暗号化の有効化と仮想ディスクのロック
- サーバープロファイルのエクスポートおよびインポート
- アップデートおよび設定ジョブのスケジュールとステータスの追跡

Remote Services を使用する理由

Remote Services は次のようなメリットと機能を提供します。

- 既存のコンソールを 1 対多サーバープロビジョニング用に活用します。
- 管理下システム上のオペレーティングシステムリソースを使用しません。
- 管理用にセキュアな通信パスを提供します。
- サーバーをプロビジョニングする間の手動操作を減らし、効率性を高めます。
- 設定の変更とアップデートのスケジュールを可能にすることにより、メンテナンスのためのシャットダウン時間を減らします。
- Windows および Linux コマンドライン (CLI) スクリプトを有効にします。
- WS-Management インタフェースを介したコンソールへの統合を可能にします。
- オペレーティングシステムにとらわれないソフトウェアのアップデートをサポートします。

Lifecycle Controller のライセンス可能機能

Lifecycle Controller 機能は、購入済みライセンスのタイプ (Basic Management、iDRAC7 Express、iDRAC7 Express for Blades、または iDRAC7 Enterprise) に応じて使用できます。Lifecycle Controller Web インターフェースでは、ライセンス取得された機能のみが使用可能になります。ライセンスの管理の詳細については、『iDRAC7 ユーザーガイド』を参照してください。次の表は、購入済みライセンスに応じて使用できる Lifecycle Controller の機能を示しています。

表 1. ライセンス可能機能

機能	IPMI 付き Base Management	iDRAC7 Express	iDRAC7 Express for Blades	iDRAC7 Enterprise
ファームウェアアップデート	あり	あり	あり	あり
オペレーティングシステムの展開	あり	あり	あり	あり
デバイス構成	あり	あり	あり	あり
診断	あり	あり	あり	あり
サーバープロファイルのエクスポートおよびインポート	-	-	-	あり
部品交換	-	-	-	あり
ローカルアップデート	あり	あり	あり	あり
ドライバパック	あり	あり	あり	あり

機能	IPMI 付き Base Management	iDRAC7 Express	iDRAC7 Express for Blades	iDRAC7 Enterprise
Remote Services (WSMAN 経由)		あり	あり	あり

Web Services for Management

WS-Management はシステム管理用に設計された Simple Object Access Protocol (SOAP) ベースのプロトコルです。WS-Management は、Distributed Management Task Force (DMTF) によって公開されているもので、デバイスがネットワークを介してデータを共有および交換するための相互運用可能なプロトコルを提供します。Lifecycle Controller Remote Services WS-Management 実装は、DMTF WS-Management 仕様バージョン 1.0.0 に準拠しています。

Dell Lifecycle Controller Remote Services は、WS-Management を使用して DMTF Common Information Model (CIM) ベースの管理情報を伝達します。CIM 情報は、管理下システムで操作可能なセマンティクスと情報の種類を定義するものです。Dell は WS-Management インタフェースを活用して、ハードウェアライフサイクル操作へのリモートアクセスを可能にします。

Dell の組み込みサーバープラットフォーム管理インタフェースはプロファイルで構成されており、各プロファイルは特定の管理ドメインや機能領域に固有のインタフェースを定義しています。さらに、デルではモデルやプロファイル拡張を多数定義することで、追加機能用のインタフェースを提供しています。WS-Management を通して使用できるデータやメソッドは、次の DMTF プロファイルおよび Dell 拡張プロファイルにマッピングされた、Lifecycle Controller Remote Services の計装インタフェースによって提供されます。

標準 DMTF

- ベースサーバー — ホストサーバーを表すための CIM クラスを定義します。
- ベースメトリクス — 管理下エレメント用にキャプチャされたマトリックスをモデリングし、制御する機能を提供するための CIM クラスを定義します。
- サービスプロセッサ — サービスプロセッサをモデリングするための CIM クラスを定義します。
- 物理的資産 — 管理下要素の物理的側面を表す CIM クラスを定義します。
- SM CLP 管理ドメイン — CLP の設定を表すための CIM クラスを定義します。
- 電源状態管理 — 電源制御処理のための CIM クラスを定義します。
- コマンドラインプロトコルサービス — CLP の設定を表現するための CIM クラスを定義します。
- レコードログ — 異なる種類のログを表現するための CIM クラスを定義します。
- 役割ベースの認証 — 役割を表すための CIM クラスを定義します。
- SMASH コレクション — CLP の設定を表現するための CIM クラスを定義します。
- プロファイル登録 — プロファイル実装を公開するための CIM クラスを定義します。
- シンプル ID 管理 — ID を表すための CIM クラスを定義します。

Dell 拡張

- Dell OS 展開 — オペレーティングシステム展開機能の設定を表現するための CIM および Dell 拡張クラスを定義します。
- Dell ソフトウェアアップデートプロファイル — BIOS、コンポーネントファームウェア、Lifecycle Controller ファームウェア、診断、およびドライバパックをアップデートするためのサービスクラスとメソッドを表す CIM および Dell 拡張を定義します。
- Dell ソフトウェアインベントリプロファイル — 現在インストールされている BIOS、コンポーネントファームウェア、診断、Lifecycle Controller、およびドライバパックのバージョンを表現するための CIM および Dell 拡張を定義します。また、ロールバックおよび再インストール用に Lifecycle Controller で利用できる BIOS およびファームウェアアップデートイメージのバージョン表現を提供します。

- **Dell ジョブ制御プロファイル**— アップデート要求によって生成されるジョブを管理するための CIM および Dell 拡張を定義します。ジョブは作成、削除、変更、および複数のアップデートを順序付けて一度の再起動で実行するためのジョブキューへの統合が可能です。
- **Dell Lifecycle Controller 管理プロファイル**— 自動検出、パーツ交換、ライフサイクルログの管理、およびハードウェアインベントリのエクスポートを管理するための属性の取得および設定するための CIM および Dell 拡張を定義します。
- **電源装置プロファイル**— システムの電源装置管理に関するプロパティとメソッドを定義します。
- **SMASH 集合体プロファイル**— システム管理コマンドラインプロトコル (SM-CLP) のターゲットアドレスリングをサポートする集合体を定義します。
- **Dell RAID プロファイル**— RAID ストレージの表現および設定のためのクラス、プロパティ、およびメソッドを説明します。
- **Dell Simple NIC プロファイル**— NIC および CNA ネットワークコントローラの表現および設定のためのクラス、プロパティ、およびメソッドを説明します。
- **Dell 永続ストレージプロファイル**— Dell プラットフォーム上の vFlash SD カードのパーティションを表現および管理するためのクラス、プロパティ、およびメソッドを説明します。
- **Dell BIOS および起動管理プロファイル**— システム BIOS セットアップの設定を表現し、システムの起動順序を管理するためのクラス、プロパティ、およびメソッドを説明します。
- **Dell CPU プロファイル**— プロセッサの管理に関するシステム管理タスクを実行するためのプロパティとインタフェースを説明します。
- **Dell ファンプロファイル**— ファンの管理に関するシステム管理タスクを実行するためのプロパティとインタフェースを説明します。
- **Dell iDRAC カードプロファイル**— iDRAC カードの基本的なプロパティの管理に関するシステム管理タスクを実行するためのプロパティとインタフェースを説明します。
- **Dell メモリ情報プロファイル**— メモリカード (DIMM) の管理に関するシステム管理タスクを実行するためのプロパティとインタフェースを説明します。
- **Dell PCI デバイスプロファイル**— システム内の PCI デバイスの管理に関するシステム管理タスクを実行するためのプロパティとインタフェースを説明します。
- **Dell システム情報プロファイル**— ホストシステムの管理に関するシステム管理タスクを実行するためのプロパティとインタフェースを説明します。
- **Dell ビデオプロファイル**— システム内のビデオコントローラの管理に関するシステム管理タスクを実行するためのプロパティとインタフェースを説明します。
- **Dell ライセンス管理プロファイル**— 管理下システム上の機能ライセンスを管理するためのクラス、プロパティおよびメソッドを説明します。
- **Dell イベントフィルタプロファイル**— イベントフィルタの表示、およびイベントの処置と通知を設定するためのクラス、プロパティおよびメソッドを説明します。
- **Dell センサープロファイル**— 管理下システムのセンサーを管理するためのクラス、プロパティおよびメソッドを説明します。
- **Dell 電源状態管理プロファイル**— システムの電源を管理するためのクラス、プロパティおよびメソッドを説明します。
- **レコードログ**— 異なる種類のログを表現するための CIM クラスを定義します。

Lifecycle Controller Remote Services WS-Management の実装は、転送セキュリティのためにポート 443 で SSL を使用し、基本的な認証をサポートします。Web サービスインタフェースは、Windows WinRM や Powershell CLI などのクライアントインフラストラクチャ、WS-MANCLI などのオープンソースユーティリティ、および Microsoft .NET などのアプリケーションプログラミング環境を活用して利用できます。

その他の必要マニュアル

このガイド以外にも、デルサポートサイト support.dell.com/manuals から次のガイドを入手できます。マニュアルページで、ソフトウェア → システム管理 をクリックします。右側の該当する製品リンクをクリックして、マニュアルにアクセスします。

- 『iDRAC7 バージョン 1.00.00 README』には Lifecycle Controller Remote Services の制限、既知の問題、およびその解決法などが記載されています。

- 『*Lifecycle Controller Web* サービスインターフェースガイド (Windows および Linux) 』には様々なメソッドを使用するためのサンプルが記載されています。
- 『*Dell Lifecycle Controller ユーザーズガイド*』には、GUI ベースのプレオペレーティングシステムコンソールの使用に関する情報が記載されています。
- 『*システム管理概要ガイド*』にはシステム管理タスクを実行するために使用できる様々なソフトウェアに関する簡潔な情報が記載されています。
- 『*Integrated Dell Remote Access Controller 7 (iDRAC7) ユーザーズガイド*』では、ラック型、タワー型、およびブレードサーバー用に iDRAC7 を設定し、ネットワーク経由でお使いのシステムとその共有リソースをリモートで管理および監視することについての情報を提供します。
- 『*Dell Repository Manager ユーザーガイド*』は、サポートされている Microsoft Windows オペレーティングシステムを実行するシステム用に、Dell Update Packages (DUP) で構成されるカスタムバンドルおよびリポジトリを作成する方法についての情報を提供します。
- 『*Dell システムソフトウェアサポートマトリックス*』の「*Lifecycle Controller 対応の Dell システムとオペレーティングシステム*」の項では、ターゲットシステムに展開できる Dell システムおよびオペレーティングシステムの一覧が掲載されています。
- 『*PERC H710, H710P, および H810 テクニカルガイドブック*』では、PERC H710、H710P、および H810 コントローラに関する仕様および設定関連の情報が記載されています。
- 『*Dell Systems Build and Update Utility (SBUU) ユーザーズガイド*』では、Dell システムの展開およびアップデートのための情報が記載されています。
- 「用語集」では、本書で使用されている用語が説明されています。


詳細については、次のシステムマニュアルを参照することができます。

- 『*iDRAC7 概要および機能ガイド*』では、iDRAC7 とそのライセンス可能機能、およびライセンスのアップグレードオプションに関する情報が記載されています。
- システムに付属している「安全にお使いいただくために」には安全や規制に関する重要な情報が記載されています。規制に関する詳細な情報については、dell.com/regulatory_compliance にある法規制の順守ホームページを参照してください。保証に関する情報は、このマニュアルに含まれているか、別の文書として同梱されています。
- ラックソリューションに付属の『*ラック取り付けガイド*』では、システムをラックに取り付ける方法について説明しています。
- 『はじめに』では、システムの機能、システムのセットアップ、および仕様の概要を説明しています。
- 『*オーナーズマニュアル*』では、システムの機能、システムのトラブルシューティング方法、およびシステムコンポーネントの取り付けまたは交換方法について説明しています。

追加の展開ガイド、ホワイトペーパー、プロファイル仕様、クラス定義 (.mof) ファイル、およびコードサンプルにも、次のアドレスからアクセスすることができます。

- Dell TechCenter の Lifecycle Controller ページ — delltechcenter.com/page/Lifecycle+Controller
- Lifecycle Controller WS-Management スクリプトセンター — delltechcenter.com/page/Scripting+the+Dell+Lifecycle+Controller
- MOF およびプロファイル — delltechcenter.com/page/DCIM.Library
- DTMF ウェブサイト — dmf.org/standards/profiles/
- *Lifecycle Controller Web* サービスインターフェースガイド — Windows および Linux

デルへのお問い合わせ

 **メモ:** お使いのコンピュータがインターネットに接続されていない場合は、購入時の納品書、出荷伝票、請求書、またはデルの製品カタログで連絡先をご確認ください。

デルでは、オンラインまたは電話によるサポートとサービスのオプションを複数提供しています。サポートやサービスの提供状況は国や製品ごとに異なり、国/地域によってはご利用いただけないサービスもございます。デルのセールス、テクニカルサポート、またはカスタマーサービスへは、次の手順でお問い合わせいただけます。

1. **support.dell.com** にアクセスします。
2. サポートカテゴリを選択します。
3. 米国在住以外のお客様は、**support.dell.com** ページ下の国コードを選択してください。**All** を選択するとすべての選択肢が表示されます。
4. 必要なサービスまたはサポートのリンクを選択します。

Remote Services の使用

本項では、より優れた成果を挙げるため、Remote Services 機能の使用を開始し、新機能を効率的に使用するために役立つ前提条件の一部について説明します。

Remote Services 使用前の共通前提条件

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- Lifecycle Controller 2 バージョン 1.00.00 がインストールされている。
- iDRAC7 ファームウェアバージョン 1.00.00
- 最新バージョンの BIOS がインストールされている。Dell システムに関連する BIOS のバージョンの詳細については、『*iDRAC7 バージョン 1.00.00 Readme*』を参照してください。
- タスクを実行するための WS-Management 対応のユーティリティが利用可能。
- 最新の『*Lifecycle Controller Web サービスインターフェースガイド (Windows および Linux)*』をダウンロードしてください。詳細については、delltechcenter.com/page/Lifecycle+Controller を参照してください。
- Collect System Inventory on Restart (CSIOR) が有効になっている。

Web サービスのセットアップ

システムのセットアップ時には、次の条件が満たされていることを確認してください。

- Remote Services にアクセスするには次のツールを使用します。
 - オペレーティングシステムにインストール済みの Windows ベースのクライアントである WinRM。インストールされていない場合は、support.microsoft.com/kb/968930 からのダウンロードが可能です。
 - オープンソース OpenWSMan ベースの CLI などの Linux ベースのクライアント。詳細については、openwsman.org を参照してください。
 - オープンソースプロジェクト Wiseman などの Java ベースのクライアント。詳細については、wiseman.dev.java.net を参照してください。
- お使いのネットワークにおけるシステムの IP アドレスを把握しておいてください。iDRAC への接続が可能であることも必要です。詳細については、support.dell.com/manuals で iDRAC のマニュアルを参照してください。
- クライアントおよび管理下サーバーのネットワーク設定が適切であることを確認します。ping ユーティリティとの接続性を確認し、その後、クライアントとネットワークで HTTP および SSL プロトコルが使用できることを確認してください。

WinRM クライアント


Remote Services 機能を使用できるようにするには、コンソールに WinRM クライアントをインストールします。Microsoft Windows 7、Microsoft Windows Vista、および Microsoft Windows Server 2008 には、WS-Management と呼ばれる標準コンポーネントが装備されています。WinRM クライアントはこのコンポーネントに含まれています。Microsoft Windows XP および Microsoft Server 2003 の場合、support.microsoft.com/kb/968929 からこの

コンポーネントをダウンロードし、インストールすることができます。インストールにはローカルの管理者権限が必要です。

クライアントを接続のために設定する必要があります。詳細については、『*Lifecycle Controller Web* サービスインタフェースガイド-*Windows* および *Linux*』を参照してください。

OpenWSMan クライアント

OpenWSMan クライアントは、オープンソースプロジェクト *Openwsman* の一部である *WS-Management CLI* です。sourceforge.net から *WS-Management CLI* および *OpenWSMan* パッケージをダウンロードし、構築、インストール、および使用するには、openwsman.org でダウンロードリンクを参照してください。

 **メモ:** クライアントを接続のために設定する必要があります。設定詳細については、『*Lifecycle Controller Web* サービスインタフェースガイド-*Linux*』を参照してください。

使用事例の使い方

使用事例の構造

次の使用事例は参考としてご使用ください。

- **前提条件** — シナリオを実行する前に必要な条件を示します。
- **機能の説明** — シナリオの記述および機能の簡単な説明を提供します。
- **重要事項** — シナリオを実行する際の特別条件を示します。
- **機能またはシステム動作** — 機能およびシステム応答の機能性をリストします。
- **完了後の必要条件** — 実行終了後に、ユーザーまたはシステムによって行われるタスクをリストします。
- **参照資料** — 『*Lifecycle Controller Web* サービスインタフェースガイド-*Windows* および *Linux*』にある、手順の実行に関する詳細情報の記載場所を提供します。

使用事例の読み方

- シナリオを読み、内容を理解します。
- 必要なインフラをセットアップし、すべての前提条件タスクを完了します。
- 特別条件はいずれも指示通りに行います。
- 機能の動作とシステム反応のしくみを理解します。
- 『*Lifecycle Controller Web* サービスインタフェースガイド-*Windows* および *Linux*』バージョンにあるタスク詳細の記載場所を示す参照表と共に、プロファイル文書および *MOF* ファイルにある、メソッド、クラス、入力パラメータ、出力パラメータなどの追加情報を使用して、手順を実行します。

使用事例シナリオ

- [iDRAC vFlash カードまたはネットワーク共有へのサーバープロファイルのエクスポート](#)
- [iDRAC vFlash カードまたはネットワーク共有からのサーバープロファイルのインポート](#)
- [RAID の設定](#)
- [CNA 用パーティションのパーソナリティと帯域幅の変更](#)
- [仮想アドレス属性の設定](#)
- [起動ターゲットの設定 - iSCSI と FCoE](#)
- [iDRAC 属性の取得と設定](#)
- [iDRAC ユーザーと役割の取得と設定](#)

- [iDRAC IP アドレス変更の報告](#)
- [BIOS パスワードの設定、変更、削除](#)
- [Remote Services ステータスの取得](#)

自動検出とハンドシェイク

iDRAC の自動検出機能は、新しく設置されたサーバーがプロビジョニングサーバーをホストするリモート管理コンソールを自動的に検出することを可能にします。プロビジョニングサーバーは、管理コンソールが新しく設置された管理下システムを検出および管理できるように、iDRAC にカスタム管理ユーザー資格情報を提供します。

自動検出機能が **有効化** された Dell システム（工場出荷時のデフォルト設定は **無効**）を注文した場合、iDRAC は DHCP が有効で、ユーザーアカウントが無効化された状態で納品されます。自動検出機能が **無効** に設定されている場合は、**iDRAC7 設定ユーティリティ** を使用して自動検出を手動で有効化し、デフォルトの管理者アカウントを無効にできます。自動検出の詳細については、「[Lifecycle Controller 管理プロファイル](#)」を参照してください。


WS-Management を使用して、DCIM_LCService クラスの **SetAttribute()** メソッドを呼び出し、プロビジョニングサーバーの IP アドレスプロパティを設定できます。**SetAttribute()** 呼び出しの詳細については、delltechcenter.com/page/Lifecycle+Controller にある DCIM_LCManagement プロファイルまたは『*Lifecycle Controller インターフェースガイド (Windows および Linux)*』を参照してください。

サーバー上でリモート操作を正しく実行するために、次の条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- プロビジョニングサーバーに Dell 展開パックがインストールされている。
- Collect System Inventory on Restart (CSIOR) が有効になっている。

自動検出のための iDRAC の設定

自動検出機能を手動で有効にするには、次の手順を実行します。

1. 希望の場所にシステムを設置します。
 2. 管理下システムの電源を入れます。
 3. 起動中に **<F2>** を押します。
セットアップユーティリティメインメニューページが表示されます。
 4. **iDRAC 設定** をクリックします。
iDRAC 設定 ページが表示されます。
 5. 次の設定を指定します。
 - ネットワーク設定 — **NIC の有効化を有効** に設定します（ブレードサーバーのみ）。
 - 共通設定 — **ドメイン名の自動設定を有効** に設定します。
 - IPv4 設定 — **IPv4 の有効化を有効** に設定します。
-  **メモ:** インフラは IPv6 をサポートしていますが、自動検出の間は無効になっています。IPv6 はサーバーのプロビジョニング後に有効にすることができます。
- DHCP — **DHCP の有効化を有効** に設定し、**DHCP を使用して DNS サーバーアドレスを取得** を **有効** に設定します。
6. **戻る** をクリックしてから **ユーザー設定** をクリックします。
ユーザー設定 ページが表示されます。
 7. **ユーザーを有効にする** で **無効** を選択します。

これにより、デフォルトの管理者アカウントが無効になります。

8. **戻る** をクリックしてから **Remote Enablement** をクリックします。

Remote Enablement ページが表示されます。

9. **自動検出を有効にする** で **自動検出** を選択します。



メモ: 自動検出機能をアクティブにするには、管理者アカウントを無効にする必要があります。

10. **プロビジョニングサーバー** ボックスで、プロビジョニングサーバーの IP アドレス、またはホスト名の文字列を入力します。コマンドを使用して、プロビジョニングサーバーの IP アドレスまたはホスト名を設定するときは、次の条件が適用されます。

- `racadm racresetcfg` の発行時、または iDRAC7 のアップデート時には、iDRAC7 のデフォルトへのリセット中に **設定の保存** オプションが有効にするようにしてください。このオプションが無効になっていると、プロビジョニングサーバー IP またはホスト名が消去されます。
- 自動検出機能は、進行中のハンドシェイクに新しく設定したプロビジョニングサーバーの IP アドレスまたはホスト名を使用せず、次のハンドシェイクプロセス時からのみ使用します。

11. **次へ** をクリックしてから、**終了** をクリックします。

12. **はい** をクリックして変更を保存します。<Esc> を押して **セットアップユーティリティ** を終了します。

プロビジョニングサーバーの文字列フォーマット

自動検出機能は、次のフォーマットを使用した複数の IP アドレスおよび/またはホスト名の設定をサポートします。

- 文字列は、IP アドレス、および/またはホスト名、およびポートをコンマで区切ったリストです。
- ホスト名は修飾されています。
- IPv4 アドレスをホスト名と同時に指定する場合は、「[」で始まり、「]」で終わります。
- 各 IP アドレスまたはホスト名の後には、オプションで「:」とポート番号を付加できます。
- 有効な文字列の例：hostname、hostname.domain.com

必要な場所でのプロビジョニングの設定

必要な場所でプロビジョニングを設定するには、次の手順を行います。

1. 管理下システムの電源を入れます。
2. 起動中に、<F10> **Lifecycle Controller** を押します。
Lifecycle Controller ページが表示されます。
3. システムセットアップ → **詳細設定** → **iDRAC 設定** と移動します。
4. **次へ** をクリックして次のページに移動し、様々な設定を指定します。
 - ネットワーク設定 — **NIC の有効化** を **有効** に設定します。（ブレードサーバーのみ）
 - 共通設定 — **ドメイン名の自動設定を有効** に設定します。
 - IPv4 設定 — **IPv4 の有効化** を **有効** に設定します。



メモ: インフラは IPv6 をサポートしていますが、自動検出の間は無効になっています。IPv6 はサーバーのプロビジョニング後に有効にすることができます。

- **DHCP** — **DHCP の有効化** を **有効** に設定し、**DHCP** を使用して **DNS サーバーアドレスを取得** を **有効** に設定します。
5. 最後のページで、**適用** をクリックします。
 6. **終了** をクリックします。
 7. **終了して再起動** をクリックします。

管理下システムの自動検出

管理下システムを自動検出するには、次の手順を行います。

1. システムをネットワークに接続します。
2. 管理下システムの電源を入れます。
システムが次の操作を実行します。
 - iDRAC が起動し、プロビジョニングサーバーの IP アドレスまたはホスト名を DHCP/DNS から取得して、プロビジョニングサーバーに存在を知らせます。
 - プロビジョニングサーバーが、iDRAC からのセキュアなハンドシェイクセッションを検証して受け入れます。
 - プロビジョニングサーバーが、管理者権限のあるカスタムのユーザー資格情報を iDRAC に提供します。
 - iDRAC はセキュアなハンドシェイクを受信し、それを完了します。

管理下システムが検出された後、iDRAC は、リモートのオペレーティングシステム展開およびシステム管理タスクなどの操作を実行するための新しく取得された資格情報によって管理することができます。

DHCP または DNS の設定


システムをネットワークに追加したり、自動検出機能を有効にする前に、ダイナミックホスト設定プロトコル (DHCP) サーバー、またはドメインネームシステム (DNS) が設定されていることを確認してください。プロビジョニングサーバー IP アドレスまたはホスト名が **WS-Management** コマンドによって提供されていない場合や、F2 または F10 ベースのプロビジョニングサーバー入力を使用している場合は、次の DHCP または DNS ベースのメソッドのいずれかを使用して、DHCP または DNS を設定し、iDRAC がドメインネームやプロビジョニングサーバーのアドレスを検出できるようにします。

- DHCP サーバーがクラスのベンダースコープオプション 43、**LifecycleController**、オプション 1 を使用して、プロビジョニングサーバーの場所をコンマ区切りのリストで提供します。これらの場所はホスト名か IP アドレスであり、オプションとしてポートを含めることもできます。iDRAC は DNS ルックアップを使用して、管理コンソールのホスト名を IP アドレスに解読します。
- DNS サーバーは、IP アドレスに解読される **service option _dcimprovsrv_tcp** を指定します。
- DNS サーバーは、プロビジョニングサーバーの IP アドレスに解読される **DCIMCredentialServer** と命名された自動検出デフォルト「Host A」レコードを指定します。

DHCP と DNS の設定の詳細については、Dell エンタープライズテクノロジーセンター www.delltechcenter.com/page/Lifecycle+Controller で『*Lifecycle Controller* 自動検出ネットワークセットアップ仕様』を参照してください。

前面パネルディスプレイでの検出ステータスの表示

検出のステータスとハンドシェイクの進捗状況を前面パネルディスプレイに表示することができます。

 **メモ:** 前面パネルディスプレイはラックおよびタワー型のサーバーでのみ利用可能であり、ブレードサーバーの場合は、CMC の前面パネルディスプレイで表示する必要があります。

- 実行中
- 停止
- 一時停止
- 完了


検出プロセスを実行中の場合、最後の試行の進捗度に対応するプログレスコード（NICが無効なために検出およびハンドシェイクがブロックされたか、または管理者アカウントが有効化されているかどうかなど）を表示できます。タイムアウトまでの残り時間を表示することもできます。

新しい環境での自動検出の再開

システムが以前に自動検出を実行した場合でも、**Remote Services** を使用して自動検出を再開することができます。再開には次のオプションを使用します。

- 自動検出を即座に実行するか、次の AC パワーサイクル時に実行するかのいずれか。これは必須の入力です。
- プロビジョニングサーバーの IP アドレスまたはホスト名。これはオプションです。

例えば、管理下システムをひとつのデータセンターから別のデータセンターに移動するために自動検出を再開する必要があるとします。自動検出の設定は、検出に使用された認証情報と共に持続され、新しいデータセンターでシステムの電源を入れると自動検出は工場出荷時のデフォルト設定に応じて動作して、新しいデータセンター用の新しい iDRAC ユーザー認証情報を作成します。

 **メモ:** `Execute Server` コマンド権限を持つ DRAC 管理者または iDRAC ユーザーは、`WS-Management` コマンドを実行する必要があります。

自動検出を再開中、次の動作がデフォルトで実行されます。

- NIC の有効化（ブレードサーバー）
- IPv4 の有効化
- DHCP 有効
- デフォルトの「ルート」管理者アカウントである **User ID 2** を除いて、全ての管理者アカウントを削除
- **Active Directory** の無効化
- DHCP からの DNS サーバーアドレスの取得
- DHCP からのドメイン名の取得

ライセンスの管理

様々なシステム管理機能を有効化または無効化するためのライセンスの管理ができます。Remote Services では次の操作が可能です。

- インストール済みライセンスのリスト取得
- ライセンス可能デバイスのリスト取得
- ライセンスのインストールまたは削除
- ライセンスのエクスポート

インストールされたライセンスの表示

- DCIM_License クラスで列挙操作を実行して、システムにインストールされている全てのライセンスのインスタンスプロパティを表示します。
- 必要なライセンスの正しいインスタンス ID を使って DCIM_License クラスで取得操作を実行し、関連するプロパティを表示します。

ライセンス可能デバイスの表示

- DCIM_LicensableDevice クラスで列挙操作を実行し、システムに取り付けられている全てのライセンス可能デバイスのインスタンスプロパティを表示します。
- 必要なライセンス可能デバイスの正しいインスタンス ID を使って DCIM_LicensableDevice クラスで取得操作を実行し、関連するプロパティを表示します。

ライセンスのインストール

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- ネットワーク共有を使用している場合は、CIFS または NFS 共有をセットアップし、ネットワーク共有にライセンスをコピーします。

ライセンスをインストールするには、次の手順を行います。

1. DCIM_LicensableDevice クラスを列挙し、使用できるライセンス可能デバイスを表示します。ライセンスがインストールされているライセンス可能デバイスの FQDD を書き留めます。
2. LicenseList プロパティを参照して、ライセンス可能デバイスに現在ライセンスがインストールされていないことを検証します。LicenseList プロパティにはデバイスに現在インストールされているライセンスの資格 ID のリストが表示されます。インストールされているライセンスがある場合は、ライセンスの資格 ID を入力パラメータとして DeleteLicense() メソッドを使用し、ライセンスを削除します。代替方法として、ライセンス可能デバイスの FQDD を入力パラメータとして使用し、そのデバイスにインストールされているすべてのライセンスを削除します。
3. ImportLicense() メソッドまたは ImportLicenseFromNetworkShare() メソッドのどちらかを使用できます。
4. ImportLicense() メソッドを使用してインストールします。
 - ライセンスファイルを Base64 コード化します。

- コード化されたライセンスファイルおよびライセンス可能デバイスの FQDD を使用して、入力パラメータを準備します。
- **ImportLicense()** メソッドを呼び出します。

5. **ImportLicenseFromNetworkShare()** メソッドを使用してインストールします。

- ネットワーク共有パラメータおよびライセンス可能デバイスの FQDD を使用して、入力パラメータを準備します。
- **ImportLicenseFromNetworkShare()** メソッドを呼び出します。
- 返されたジョブ ID をインスタンス ID として使用して DCIM_LifecycleJob クラスで取得操作を実行し、ライセンスのインポートジョブのステータスを表示します。

ライセンスのインストールのための参照資料

詳細については、次のマニュアルを参照してください。

- プロファイル — Dell_LicenseManagement プロファイル
- MOF
 - DCIM_LicensableDevice.mof
 - DCIM_License.mof
 - DCIM_LicenseManagementService.mof
 - DCIM_LCElementConformsToProfile.mof
 - DCIM_LCRegisteredProfile.mof

詳細については、次を参照してください。

- [Remote Services 使用前の共通前提条件](#)
- [Lifecycle Controller のライセンス可能機能](#)

ライセンスの交換

ライセンスを交換するには、次の手順を行います。

1. DCIM_LicensableDevice クラスを列挙し、ライセンス可能デバイスの FQDD を取得します。
2. LicenseList プロパティを表示し、交換されるライセンスの資格 ID を書き留めます。
3. ライセンスファイルを Base64 コード化します。
4. コード化されたライセンスファイルを使用して、ライセンス可能デバイス FQDD と古いライセンスの資格 ID で入力パラメータを準備します。
5. **ReplaceLicense()** メソッドを呼び出し、ライセンスを交換します。

ライセンスの削除


単一のライセンスは、ライセンスの資格 ID を使用し、**DeleteLicense()** メソッドを呼び出すことによって削除できます。または、ライセンス可能デバイスの FQDD を使用し、**DeleteLicense()** メソッドを呼び出すことによって、ライセンス可能デバイスからすべてのライセンスを削除することもできます。

ライセンスのエクスポート

次の 4 通りのメソッドのうちのひとつを使用して、ライセンスをエクスポートできます。

- **ExportLicense()** — このメソッドは、資格 ID によって指定された単一のライセンスをエクスポートします。ライセンスはメソッドの出力パラメータであり、base64 コード化されています。

- **ExportLicenseToNetworkShare()** — このメソッドは、資格 ID によって指定された単一のライセンスを NFS または CIFS ネットワーク共有にエクスポートします。
- **ExportLicenseByDevice()** — このメソッドは、ライセンス可能デバイスにインストールされているすべてのライセンスをエクスポートします。ライセンスはメソッドの出力パラメータであり、base64 コード化されています。
- **ExportLicenseByDeviceToNetworkShare()** — このメソッドは、ライセンス可能デバイスにインストールされているすべてのライセンスを NFS または CIFS ネットワーク共有にエクスポートします。

 **メモ:** 単一のライセンス可能デバイスから複数のライセンスがエクスポートされる場合、ファイル名には **_0.xml**、**_1.xml**、**_2.xml** といった拡張子がつきます。

証明書の管理


iDRAC7 にカスタム定義の証明書を転送し、システムのサービスタグに基づいた固有の証明書を作成してセキュリティを強化するには、証明書の管理機能を使用します。システムの発注時、Dell 提供のカスタム工場出荷時インストール (CFI) を使用して、出荷前にご希望の証明書でシステムを事前設定するよう請求することもできます。

プロビジョニングサーバー用の Trusted Root クライアントカスタム証明書の作成

DCIM_LCService クラスの **DownloadClientCerts()** メソッドを呼び出して、カスタム署名済みの自動検出クライアント証明書を生成することができます。このメソッドは、認証局によって生成されたキー証明書、関連ハッシュ、およびパスワードパラメータを入力値として使用します。提供されるキー証明書は、システムサービスタグがコモンネーム (CN) として含まれる証明書の署名に使用されます。このメソッドは、自動検出クライアント証明書のダウンロード、生成、およびインストールが正常におこなわれたことを確認するために使用できるジョブ ID を返します。WinRM および WSMANCLI を使用したコマンドライン呼び出しの例は、『*Lifecycle Controller Web サービスインターフェースガイド (Windows および Linux)*』を参照してください。

カスタムサーバー証明書の提供

展開ネットワークのプロビジョニングサーバーすべてに署名するために使用される CA 証明書を転送するには、DCIM_LCService クラスの **DownloadServerPublicKey()** メソッドを呼び出すことができます。


 **メモ:** 信頼できる CA 証明書は、プロビジョニングサーバーすべての認証に使用されます。

プロビジョニングサーバー証明書を iDRAC に使用する前に、証明書が自己署名がされていることを確認します。

このメソッドは、CA 証明書、関連ハッシュ、およびハッシュタイプのパラメータを入力値として使用します。また、プロビジョニングサーバーのパブリックキーの処理とインストールの確認に使用できるジョブ ID を返します。WS-Management ユーティリティを使用したコマンドライン呼び出しの例は、『*Lifecycle Controller Web サービスインターフェースガイド (Windows および Linux)*』を参照してください。DCIM プロファイルの仕様および関連する MOF ファイルは delltechcenter.com/page/DCIM.Library で入手できます。

カスタム証明書の削除

管理下システムで作成またはアップロードされたカスタム証明書はいずれも削除することが可能です。この機能を使用することにより、必要時にいつでも、署名済みカスタム証明書のすべてをサーバーから一掃することができます。

 **メモ:** この機能は工場出荷時の証明書は削除しません。

カスタムサーバーパブリックキーの削除

DCIM_LCService クラスの **DeleteAutoDiscoveryServerPublicKey()** メソッドを使用して、サーバー証明書の検証または認証に使用される CA 証明書を削除します。

カスタムクライアント証明書の削除

DCIM_LCService クラスの **DeleteAutoDiscoveryClientCerts()** メソッドを使用して、クライアント証明書およびプライベートキーを削除します。

ウェブサーバーまたは WS-Management 暗号化証明書とプライベートキーの PKCS 12 からの変更

証明書およびキーを変更するには、次の手順を行います。

1. CSR およびプライベートキーを生成します。CSR には CA による署名が必要です。
2. 証明書とプライベートキーを組合せた後、PKCS#12 ファイルに暗号化します。
3. BASE64 は、PKCS#12 ファイルをバイナリからテキストに変換するために符号化して、WS-Management パラメータとして渡すことができるようにします。
4. アクティブな証明書のコンテンツを XML ファイルにコピーします。



サーバー証明書の管理

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- iDRAC の時間が正しく設定されている。

一部システムの証明書は有効期限が切れているため、新しくアップロードする必要があります。証明書は Web GUI、WS-Management、RACADM、Active Directory、および LDAP のセッションを認証します。

サーバー証明書を管理するには、次の手順を行います。

-  **メモ:** メソッドがすべての Web サービスを再起動し、アクティブなすべてのセッションを閉じます。
 -  **メモ:** 新しいサーバー証明書に署名した CA は、すべてのクライアントの信頼されたルート CA リストに追加する必要があります。
1. CSR およびプライベートキー（パスワード保護なし）を作成します — `openssl req -new -nodes`。
 2. 「`openssl ca`」を使用して CSR に署名するか、署名ウェブサーバーにアップロードします。
 3. ファイル（PEM file）に証明書とプライベートキーをコピーします — `cat cert.pem key.txt > cert_key.pem`。
 4. `cert_key.pem` を `pkcs12` に変換します — `openssl pkcs12 -export -in cert_key.pem -passin file:password.txt -out new.pfx`
 5. `pkcs12 base64` ファイルをコード化します — `openssl base64 -export -in new.pfx -out new_pfx.txt`
 6. `new_pfx.txt` のコンテンツを WS-Management コマンドへのパラメータとして使用します。
 7. 必要なパラメータで **SetCertificateAndPrivateKey()** メソッドを呼び出します。
サーバー証明書を設定すると、Web サービスが再起動します。すべてのセッションが閉じられ、新しい WS-Management コマンドは新しいサーバー証明書を受け入れる必要があります。

サーバー証明書の管理のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 2. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の記載箇所
-	12.10 — iDRAC 証明書とプライベートキーの設定
プロファイル	
DCIM_LCManagement プロファイル	
MOF	
DCIM_LCService.mof	


ディレクトリ CA 証明書の管理

Active Directory または LDAP を認証するには信頼されたルート CA 証明書をアップロードする必要があります。

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- iDRAC の時間が正しく設定されている。

ディレクトリ CA 証明書を管理するには、次の手順を行います。

 **メモ:** メソッドがすべての Web サービスを再起動し、アクティブなすべてのセッションを閉じます。

1. LDAP または AD サーバーから CA 証明書をダウンロードします。
2. openssl または別のツールを使用して証明書を base64 ベースのフォーマットにコード化します。
3. 必要なパラメータで **SetPublicCertificate()** メソッドを呼び出します。
サーバー証明書を設定すると、Web サービスが再起動します。すべてのセッションが閉じられ、新しい WS-Management コマンドは新しいサーバー証明書を受け入れる必要があります。

ディレクトリ CA 証明書の管理のための参照資料


 **メモ:** 次の表で参照されている項に含まれる例は、一般的な例のみです。

表 3. 手順番号および参照箇所

手順番号	Lifecycle Controller Web Services Interface Guide (Windows および Linux) 内の箇所
-	12.9 — パブリック証明書の設定
プロファイル	
DCIM_LCService	
MOF	
DCIM_LCService.mof	

オペレーティングシステムの展開

オペレーティングシステム展開機能は、WS-Management Web サービスプロトコルと CIFS および NFS ネットワークファイル共有プロトコルを使用して、リモートでオペレーティングシステムを展開することを可能にします。詳細なインターフェース仕様およびクラス定義 (.mof) ファイルは、delltechcenter.com の Dell エンタープライズテクノロジセンターの Lifecycle Controller の項で参照できます。次の機能は外部メソッドという形での利用が可能で、サーバーでエンドツーエンドの OS 展開を実行するための使用事例に応じた様々な組み合わせで使用できます。

- 選択したオペレーティングシステム用の内蔵ドライバのローカル公開を、サーバーに列挙された USB デバイス（インストール中に自動でインストールされる）としてリモートでアクティブ化。
- 選択したオペレーティングシステムごとに、後ほどオペレーティングシステムの展開に使用できるの組み込みドライバを、CIFS または NFS ネットワーク共有にリモートで取得。
- ネットワーク共有にある ISO イメージから起動し、オペレーティングシステムのインストールを開始。
- ISO を vFlash SD カードにダウンロードし、カードから起動してオペレーティングシステムのインストールを開始。
- ネットワークからの ISO を仮想 USB CD-ROM デバイスとしてサーバーに連結し、サーバー再起動時には常に ISO からサーバーを起動。
- PXE から 1 回限りの起動。
- ハードディスクから 1 回限りの起動。

詳細については、「[オペレーティングシステム展開プロファイル](#)」を参照してください。


オペレーティングシステムの展開

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- サーバーにオペレーティングシステムをインストールするための起動ディスクがある。
- 新しいオペレーティングシステムおよび新しいデバイスのドライバが使用できるよう、最新のドライバパックがインストールされていることをお勧めします。
- WS-Management Web サービス要求やメソッド呼び出しを送信できるプロビジョニングコンソール、アプリケーション、または適切なスクリプト。

Lifecycle Controller を介し、サーバー上にローカルに連結されたドライバを使用してオペレーティングシステムをインストールします。

リモートのオペレーティングシステム展開を実行するには、次の手順を行います。

 **メモ:** カスタムオペレーティングシステムを使用するには、カスタムオペレーティングシステムイメージ (.iso フォーマット) を作成し、ネットワークで共有するか、DVD に ISO イメージを作成します。


1. **GetDriverPackInfo()** メソッドを呼び出し、サーバーでサポートされているオペレーティングシステム、および Lifecycle Controller にインストールされているドライバパックのバージョンを一覧にします。
2. **UnpackAndAttach()** メソッドを呼び出し、選択されたオペレーティングシステムのドライバを、Lifecycle Controller からサーバーに接続されている OEMDRV とラベル付けされた USB ベースの内蔵ドライブにコピーします。

デフォルトで、この OEMDRV ドライブは約 18 時間サーバーに公開され、その後自動的に切断されます。ただし、ドライブがサーバーに存在する必要がある期間（1 分から 18 時間まで）を指定するには、メソッドの呼び出し中にオプションのパラメータ **ExposeDuration** を使用します。

3. オペレーティングシステムがどこにホストされるかに応じて、次のいずれかのメソッドを使用し、ISO をローカルサーバーに接続して、そこから即時に再起動します。
 - **BootToNetworkISO()** — オペレーティングシステムのイメージ (.iso フォーマット) がネットワーク共有 (NFS または CIFS) でホストされている場合は、このメソッドを使用してネットワーク ISO を仮想 USB CD-ROM デバイスとしてサーバーに接続し、そこから即時に起動してオペレーティングシステムのインストールを開始します。
 - **BootToISOFromVFlash()** — オペレーティングシステムのイメージ (.iso フォーマット) が vFlash SD カードでホストされている場合は、このメソッドを使用してイメージをローカル USB CD-ROM デバイスとして接続し、そこから即時に起動してオペレーティングシステムのインストールを開始します。

 **メモ:** NFS、CIFS または TFTP 共有から .iso イメージを vFlash にコピーするために **BootToISOFromVFlash()** を実行する前に、その vFlash を後ほど起動に使用できるように、前提条件として **DownloadISOToVFlash()** メソッドを使用する必要があります。ただし、vFlash SD カードが取り付けられていてもフォーマットはされていない場合は、このメソッドがカードをフォーマットし、その後 ISO イメージをダウンロードします。

- **BootToPXE()** — オペレーティングシステムイメージが PXE でホストされている場合、このメソッドを使用してサーバーを再起動し、PXE から起動してオペレーティングシステムのインストールを開始します。
4. オペレーティングシステムのインストール完了後、ISO の接続方法に基づいて次のメソッドのひとつを使用し、ISO をホストサーバーから分離します。
 - **DetachISOImage()** — ISO が **BootToNetworkISO()** を使用して接続された場合は、このメソッドを使用してホストサーバーから ISO を分離します。
 - **DetachISOFromVFlash()** — ISO が **BootToISOFromVFlash()** を使用して接続された場合は、このメソッドを使用してホストサーバーから ISO を分離します。ISO が必要なくなった場合は、その後で **DeleteISOFromVFlash()** メソッドを使用して ISO を vFlash から削除することができます。
 5. **DetachDrivers()** メソッドを呼び出して、オペレーティングシステムのドライバがある OEMDRV ドライブを分離します。

 **メモ:** オペレーティングシステムのインストール中、ネイティブ OS インストーラが自動的に OEMDRV デバイスにあるドライバをインストールします。

オペレーティングシステムの展開のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 4. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の記載箇所
手順 1	11.3.1 — ドライバパック情報の取得
手順 2	11.3.2 — 選択されたドライバのアンパックと USB デバイスとしてのホスト OS への接続
手順 3	11.3.6 — ネットワーク ISO からの起動 11.3.11 — VFlashk の ISO からの起動 11.3.8 — PXE からの起動
手順 4	11.3.7 — ネットワーク ISO USB デバイスの分離 11.3.13 — VFlash からの ISO の分離
手順 5	11.3.3 — ドライバを含むエミュレートされた USB デバイスの分離

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の記載箇所
------	--------------------------------------------------------------------

プロファイル

DCIM_OSDeployment プロファイル

MOF

- DCIM_OSDeploymentService.mof
- DCIM_OSConcreteJob.mof
- DCIM_LCElementConformsToProfile.mof
- DCIM_LCRegisteredProfile.mof

ネットワーク共有への OS ドライバのコピー

Lifecycle Controller からネットワーク共有にオペレーティングシステムのドライバをコピーするには、次の手順を行います。

1. **GetDriverPackInfo()** メソッドを呼び出し、サーバー上の対応オペレーティングシステムと Lifecycle Controller にインストールされているドライバパックのバージョンをリスト化します。
2. **UnpackAndShare()** メソッドを呼び出し、選択したオペレーティングシステム用のドライバを Lifecycle Controller からネットワーク共有 (CIFS または NFS) にコピーします。

オペレーティングシステムの展開のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 5. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の記載箇所
手順1	11.3.1 — ドライバパック情報の取得
手順2	11.3.2 — 選択されたドライバのアンパックと USB デバイスとしてのホスト OS への連結

プロファイル

DCIM_OSDeployment プロファイル

MOF

- DCIM_OSDeploymentService.mof
- DCIM_OSConcreteJob.mof
- DCIM_LCElementConformsToProfile.mof
- DCIM_LCRegisteredProfile.mof

リモートファイル共有の使用

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」で説明されている前提条件が満たされていることを確認してください。

リモートファイル共有を使用してオペレーティングシステムを展開するには、次の手順を行います。

1. サーバーへのローカル CD-ROM デバイスとして列挙されているリモートファイル共有 (RFS) 上の ISO を連結する **ConnectRFSISOImage()** メソッドを呼び出します。RFS が iDRAC GUI または RACADM を使用して

連結されていること、または iDRAC 属性 AttachMode の値が Web サービスを使用して Attach (連結) に設定されていることを確認します。

2. **GetRFSISOImageConnectionInfo()** メソッドを呼び出して、RFS 接続情報を取得します。
3. **DisconnectRFSISOImage()** メソッドを呼び出して、ISO イメージをサーバーから分離します。

リモートファイル共有の使用のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 6. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の記載箇所
手順 1	11.3.18 - RFS ISO イメージの連結
手順 2	11.3.20 - RFS ISO イメージ連結情報の取得
手順 3	11.3.19 - RFS ISO イメージの分離
プロファイル	
DCIM_OSDeployment プロファイル	
MOF	
	<ul style="list-style-type: none">• DCIM_OSDeploymentService.mof• DCIM_OSConcreteJob.mof• DCIM_LCElementConformsToProfile.mof• DCIM_LCRegisteredProfile.mof


サーバーメンテナンス中の ISO からの起動

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」の項で説明されている前提条件が満たされていることを確認してください。

データセンターおよびエンタープライズ環境での仮想マシンと作業負荷のホストには、一般的に物理サーバーが使用されます。サーバーのメンテナンスが必要なときは (ハードウェアの交換、構成の変更、アップデートなど)、作業負荷は他の物理システムに移動され、メンテナンスが必要なサーバーはメンテナンスモードになります。このモードでは、サーバーはすべての問題が解決されるまで、ネットワーク共有から連結されている pre-OS 環境 (通常は ISO) から複数回起動します。OS 展開プロファイルを使用すると、この操作をより効率的に達成するために次のメソッドを使用できます。

サーバーメンテナンス中に ISO から起動するには、次の手順を行います。

1. **ConnectNetworkISOImage()** メソッドを呼び出し、ネットワーク共有 (CIFS または NFS) から ISO を仮想 CD-ROM デバイスとしてサーバーに公開します。管理下システムがメンテナンス中に再起動されたときは常に、ISO が **DisconnectNetworkISOImage()** メソッドを使用して分離されるまで、システムは起動順序に関係なくこの ISO から毎回起動します。

 **メモ:** iDRAC がリセットするとき、または電源が切断されたときは、ISO は再連結されます。

2. **GetNetworkISOImageConnectionInfo()** メソッドを呼び出し、**ConnectNetworkISOImage()** メソッドを使用して接続されたネットワーク ISO に関する詳細情報を取得します。このメソッドは、ISO がシステムから起動されたか否かも表示します。詳細については「*OSDeployment* プロファイル」、および関連する MOF を参照してください。
3. **DisconnectNetworkISOImage()** メソッドを呼び出して、ISO イメージをサーバーから分離します。
4. **SkipISOImageBoot()** メソッドを呼び出して、システムが次のサーバー起動時に、(**ConnectNetworkISOImage()** メソッドを使用して) 連結されている ISO から 1 度だけ起動しないようにし

まず、その起動後のサーバー再起動では、ISO が **DisconnectNetworkISOImage()** メソッドを実行して分離されるまで、BIOS は ISO からの起動を続行します。

サーバーメンテナンス中の ISO からの起動のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 7. 手順番号および参照箇所


手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 1	11.3.14 — ネットワーク ISO イメージの連結 11.3.15 — ネットワーク ISO イメージの分離
手順 2	11.3.17 — ネットワーク ISO イメージ連結情報の取得
手順 3	11.3.15 — ネットワーク ISO イメージの分離
手順 4	11.3.16 — ISO イメージ起動の省略
プロファイル	
DCIM_OSDeployment プロファイル	
MOF	
<ul style="list-style-type: none"> • DCIM_OSDeploymentService.mof • DCIM_OSConcreteJob.mof • DCIM_LCElementConformsToProfile.mof • DCIM_LCRegisteredProfile.mof 	


ISO からの起動メソッドの比較


表 8. ISO からの起動メソッドの比較

手順	BootToNetworkISO	BootToISOFromVFlash	ConnectNetworkISOImage	ConnectRFSISO
ネットワーク ISO に接続し、仮想 CD-ROM として連結	可	-	可	可
vFlash SD カード上の ISO に接続し、仮想 CD-ROM として連結	-	可	-	-
ホストサーバーを自動的に再起動	可	可	-	-
ISO イメージから即時起動	可	可	-	-
一回限りの起動	可	可	-	-
ホストサーバーに 18 時間 (または指定の期間) 連結	可	可	-	-

手順	BootToNetworkISO	BootToISOFromVFlash	ConnectNetworkISOIm age	ConnectRFSISO
Web サービスを使用してこのメソッドを実行した後、updates や config などのその他ジョブも実行可能	-	-	-	可

 **メモ:** 次回のホスト再起動では、デバイス (ISOIMG) が BIOS 起動リストで最初のデバイスとして設定されていない限り、自動的に ISO イメージからは起動されません。これは ISO が連結され、時間切れになっていないとき、または ISO が明確に分離されていない場合に該当します。

 **メモ:** ホストシステムが再起動するときは常に、BIOS はその起動順序に関係なくこのデバイス (ISOIMG) から起動します。


 **メモ:** RFS デバイスが、BIOS 起動リスト上の最初のデバイスであり、サーバーが再起動する場合は、BIOS は毎回連結されている ISO から起動します。

1 回限りの起動

サーバーを即時に再起動し、ISO、ハードディスクまたは PXE から起動するには 1 回限りの起動メソッドを使用します。サーバーのメンテナンス実行中、オペレーティングシステムのインストールの開始中、ハードディスクからの起動中、または PXE からの起動中に、起動前の ISO から 1 回限りの起動をするにはこれらのメソッドを使用します。

これには、次のメソッドを呼び出します。

- **BootToPXE()** — サーバーを即時再起動して、起動リストの起動順序に関係なく PXE から起動するには、このメソッドを呼び出します。
- **BootToHD()** — サーバーを即時再起動して、起動リストの起動順序に関係なくサーバーの最初のハードディスクから起動するには、このメソッドを呼び出します。

 **メモ:** あるいは、**ChangeBootOrderByInstanceID()** または **SetAttribute()** などの BIOS メソッドを使用して、1 回限りの起動を行います。

BootToHD メソッドの動作

メソッドが正確に動作しない可能性のあるインスタンスの一部を次に示します。

- システムに 2 個以上のハードディスクドライブがある場合、システムは起動順序で最初のハードディスクドライブを選択します。
- 起動順序に最初のデバイスと違うデバイスがある場合 (例えば、USB ドライブなど)、そのデバイスから起動します。
- システムが UEFI 起動モードになっている場合、オペレーティングシステム付きのハードディスクドライブも UEFI 起動モードで取り付けられている必要があります。BIOS 起動モードで取り付けられているオペレーティングシステム付きハードディスクドライブからの一回限りの起動は機能しません。

- システムにハードディスクドライブが取り付けられていない場合でもメソッドは実行されるので、メソッドを実行する前にサポートされているハードディスクドライブが取り付けられていることを確認してください。

ジョブ ID について

本書で説明されているいくつかのメソッドは、出力パラメーターとしてジョブ ID を返します。ジョブは、要求された処置が即時に実行できないときの追跡に役立ちますが、基盤となる技術の制約のため、標準的な Web サービス要求の応答タイムアウトよりも時間がかかります。返されたジョブ ID はその後、ジョブオブジェクトのインスタンスを取得するために **WS-MAN Enumerate** または **Get** 要求で使用できます。ジョブオブジェクトのインスタンスには、ジョブの状態、およびジョブが正常に完了したか、問題が発生して失敗したかを確認できるジョブステータスのプロパティが含まれています。ジョブの失敗が発生した場合、ジョブインスタンスには失敗の性質に関する詳細情報を提供するエラーメッセージプロパティも含まれます。他のプロパティには、対応言語へのエラーメッセージのローカライズ、エラーのより詳細な説明や、推奨される対応処置の説明を得るために使用できる、その他のエラー識別情報が含まれています。

本書で説明する **DCIM_OSDeploymentService** メソッドはすべて、メソッドが正常に実行されたか、エラーが発生したか、またはジョブが作成されたかどうかを示すエラーコードを返します。メソッド内で実行されている処置を即時に完了できない場合は、ジョブの作成が行われます。さらに、エラーが発生した場合、これらのメソッドは、エラーを対応言語にローカライズするために使用できるエラーメッセージ（英語）、およびその他のエラー識別子を含む出力パラメータも返します。エラー識別子は、**Dell Message Registry XML** ファイルへのインデックスおよび処理に使用できます。**Dell Message Registry** ファイルは、6 つの対応言語で利用できます（1 言語ごとに 1 ファイル）。翻訳されたエラーメッセージに加えて、**Message Registry** ファイルには、**Lifecycle Controller Remote Service** の Web サービスインタフェースから返される各エラーのための追加の詳細説明と推奨される対応処置も含まれています。**Dell Message Registry XML** ファイルをダウンロードするには、delltechcenter.com/page/Lifecycle+Controller/ を参照してください。

明確なジョブインスタンスを返すメソッドは次の通りです。

- **UnpackAndAttach()**
- **UnpackAndShare()**
- **BootToNetworkISO()**
- **DownloadISOToVFlash()**
- **BootToISOFromVFlash()**
- **ConnectNetworkISOImage()**
- **ConnectRFSISOImage()**

ジョブインスタンスは返さず、出力パラメータのみを返すメソッドは次の通りです。

- **GetDriverPackInfo()**
- **DetachDrivers()**
- **DetachISOImage()**
- **BootToPXE()**
- **BootToHD()**
- **GetHostMACInfo()**
- **DetachISOFromVFlash()**
- **DeleteISOFromVFlash()**
- **DisconnectNetworkISOImage()**
- **GetNetworkISOImageConnectionInfo()**
- **SkipISOImageBoot()**
- **DisconnectRFSISOImage()**

- `GetRFSISOImageConnectionInfo()`

ジョブの管理

Remote Services は次の機能によって Lifecycle Controller ジョブを管理します。

- ジョブの作成 — 設定を適用するために特定の種類のジョブを作成します。
- ジョブおよびジョブキューのスケジュール — DCIM_JobService クラスで **SetupJobQueue()** メソッドを使用して、複数のジョブを 1 回のシステム再起動で実行します。開始時刻を設定せずに **CreateTargetedConfigJob()** メソッドを使用してジョブを作成した場合は、**SetupJobQueue()** メソッドを使用してスケジュールおよび実行順序を設定します。開始時刻を **CreateTargetedConfigJob()** メソッドで設定した場合は、他のジョブと包括することはできず、ジョブは指定された時刻で実行されるようにセットアップされます。
- ジョブの削除 — DCIM_JobService クラスで **DeleteJobQueue()** メソッドを使用して、指定した既存のジョブを削除します。
- すべてのジョブの報告 — DCIM_LifecycleJob クラスを列挙してすべてのジョブを報告します。
- スケジュールされたジョブの報告 — JobStatus=Scheduled の選択フィルタと共に DCIM_LifecycleJob クラスを列挙し、スケジュールされたジョブのすべてのレポートを生成します。

ジョブコントロールの詳細については、「[ジョブコントロールプロファイル](#)」を参照してください。

ジョブタイプ

ジョブにはシステムによって作成されたジョブ（黙示的）とユーザーが作成したジョブ（明示的）の 2 種類があります。

- システム作成ジョブは、特定のリモートサービスタスクを実行するときに作成されます。例えば、ハードウェアインベントリのエクスポート、ライセンスのエクスポート、永続ストレージパーティションの作成などの Remote Services の機能はジョブを作成し、ジョブ ID を返します。ジョブステータスのポーリングは、タスクの完了ステータスを特定します。
- **CreateTargetedConfigJob**、**CreateRebootJob**、および **InstallFromURI** などのユーザー作成ジョブは、RAID、NIC、および BIOS などにユーザー設定を適用するために使用します。ユーザー作成ジョブは即時に実行するか、または決められた時間に実行するようにスケジュールすることができます。


システム作成ジョブ	ユーザー作成ジョブ
vFlash (初期化)	RAID 設定
vFlash (パーティションの作成)	BIOS 設定
vFlash (パーティションのフォーマット)	NIC 設定
vFlash (パーティションの連結)	iDRAC 設定
vFlash (パーティションの分離)	システム設定
vFlash (パーティションからのデータのエクスポート)	(BIOS、NIC、RAID 等) のソフトウェアアップデート
vFlash (イメージを使用したパーティションの作成)	再起動
ライフサイクルログのエクスポート	
ハードウェアインベントリのエクスポート	

ユーザー作成ジョブ

次のジョブはユーザーが作成したジョブです。

- **CreateTargetedConfigJob** — このメソッドは、RAID、NIC、BIOS、iDRAC およびシステムを設定するときに使用します。設定の変更を確定し、変更を適用するためにジョブを作成するには、このメソッドを使用します。
- **CreateRebootJob** — 再起動のジョブを作成するには、このメソッドを使用します。
- **InstallFromURI** — このメソッドを使用して、BIOS、RAID、NIC、iDRAC、PSU、Lifecycle Controller、OS ドライバパック、および診断のファームウェアをアップデートします。アップデートが成功すると、メソッドはジョブ ID を返します。ジョブ ID はそのエンティティでソフトウェアアップデートを実行します。

 **メモ: InstallFromURI()** メソッドは、BIOS、RAID、NIC、iDRAC および PSU ファームウェアのアップデートに使用され、ジョブ ID のスケジュールには **SetupJobQueue()** メソッドを使用します。

 **メモ: Lifecycle Controller、ドライバパック、および診断のアップデートに InstallFromURI()** メソッドを使用する場合は、ジョブをスケジュールしないでください。

ジョブスケジュール

ジョブ作成メソッドのうちのひとつを使用して返されたジョブ ID のために、ジョブをスケジュールすることができます。

SetupJobQueue — このメソッドは、ジョブ作成メソッドのいずれかによって返され、まだスケジュールされていないジョブ ID でのみ使用します。


ジョブの削除

JID_CLEARALL() メソッドを使用して、システムの現在のジョブをすべて削除できます。または、特定のジョブ ID を使って、そのジョブを削除することもできます。

複数のアクションへの個別ジョブのスケジュール


複数のアクションに個別のジョブをスケジュールするには、次の手順を行います（次の例では、BIOS および NIC/CNA のアップデートと NIC の設定）。

1. BIOS および NIC ファームウェアのアップデートパッケージのために、**InstallFromURI()** メソッドを呼び出します。
このメソッドで BIOS および NIC のアップデートがダウンロードされ、各デバイスのアップデートジョブにジョブ ID が作成されます。
2. NIC に NIC 属性（例えば、Integrated NIC 1）を設定し、このセットにターゲットのジョブを作成します。メソッドからジョブ ID が返されます。
3. ジョブが指定された開始時刻に指定された順序で実行されるように、これらのジョブ ID と **SetupJobQueue()** メソッドを使用してスケジュールを行います。

 **メモ:** スケジュールされた時間に iDRAC を自動的に再起動するには、再起動ジョブを作成（再起動のタイプをグレースフルまたはパワーサイクルに指定）し、**SetupJobQueue()** メソッド呼び出しに指定されたジョブ一覧に再起動ジョブ ID を追加します。再起動ジョブがジョブキューセットアップに含まれていない場合、ジョブはスケジュールされた開始時刻に実行される準備ができていますが、システムの再起動とジョブ実行の開始は、外部からの処置に依存します。

複数のターゲットジョブの実行

複数のターゲットジョブ（例えば複数の NIC での NIC 属性の設定）を一度に実行するには、次の手順を行います。

 **メモ:** ターゲットジョブは、POST またはセットアップユーティリティの実行中に作成できます。ジョブは、システムが POST を完了、またはセットアップユーティリティを終了するまで実行されません。

1. **Integrated NIC 1** の設定 : **Integrated NIC 1** に NIC 属性を設定し、**TIME_NOW** のスケジュール済み開始時刻でターゲット指定の設定ジョブを作成しますが、再起動はスケジュールしないようにしてください。
2. **Integrated NIC 2** の設定 : **Integrated NIC 2** に NIC 属性を設定し、**TIME_NOW** のスケジュール済み開始時刻でターゲット指定の設定ジョブを作成しますが、再起動はスケジュールしないようにしてください。
3. **Integrated NIC 3** に NIC 属性を設定し、**TIME_NOW** のスケジュール済み開始時刻で **Integrated NIC 3** にターゲット指定のジョブを作成して、再起動のタイプも指定します。
再起動のタイプによって定義されたメソッドにもとづいて iDRAC がシステムを再起動し、すべてのジョブが一度に実行されます。

開始時間と終了時間の指定

CreateTargetedConfigJob() および **SetupJobQueue()** メソッドは、**ScheduledStartTime**、**StartTimeInterval**、および **UntilTime** などの開始時間パラメータを受け入れます。パラメータのデータタイプは、**CIM** 日付 - 時刻です。**StartTime** パラメータが **null** の場合、処置は開始されません。日付 - 時刻のデータタイプは **YYYYMMDDhhmmss** の形式で定義されます。詳細は次の通りです。

- **yyyy** は年です
- **mm** は月です
- **dd** は日です
- **hh** は時間です
- **mm** は分です
- **ss** は秒です

例えば、**20090930112030** のようになります。すべての **Lifecycle Controller** アップデートのために日付および時刻をこの形式で入力し、属性、および異なるサービスクラスの **CreateTargetedConfigJob()** メソッドを設定します。**TIME_NOW** は、タスクをただちに実行を表現する特別な値です。

ジョブの自動削除

システム上のジョブの数が **DCIM_JobService** クラスの **StartAutoDeleteAtThreshold** プロパティ値を超えると、ジョブは自動的に削除されます。**DCIM_JobService** クラスの **DeleteOnCompletionTimeout** 値を超える時間で完了したジョブは、成功したかどうかに関わらず、すべてシステムから削除されます。

 **メモ:** **DeleteOnCompletionTimeout** 値は、**SetDeleteOnCompletionTimeout** メソッドを使用して変更します。

すべてのジョブをクリア

ジョブ ID にキーワード `JID_CLEARALL` と `DeleteJobQueue()` メソッドを使用して、すべてのジョブおよびそのジョブに関連する保留中の設定データすべてをクリアします。

RAID 設定の管理

RAID 設定機能を使用して、システムに取り付けられている RAID コントローラ、物理ディスク、およびエンクロージャのプロパティを取得します。利用可能なメソッドを使用して、物理ディスクと仮想ディスクの異なる属性を設定することができます。

RAID コントローラの表示

- DCIM_ControllerView クラスで列挙操作を実行して、システムに取り付けられているすべての RAID コントローラのインスタンスプロパティを表示します。
- 必要な RAID コントローラの正しいインスタンス ID を使って DCIM_ControllerView クラスで取得操作を実行し、関連するプロパティを表示します。

スライスされた仮想ディスクの作成

スライスされた仮想ディスクを作成するには、次の手順を行います。

1. DCIM_RAIDService クラスで **GetRAIDLevels()** メソッドを使用し、システム内の RAID 構成を調べます。
2. DCIM_RAIDService クラスで **GetAvailableDisks()** メソッドを使用して収集した ID を元に、仮想ディスクを作成する必要がある物理ディスクを選択します。
3. DCIM_RAIDService クラスで **CheckVDValues()** メソッドを使用して、必要な RAID レベルおよび物理ディスクで利用可能なサイズ、および仮想ディスクのデフォルトパラメータをチェックします。
4. **CreateVirtualDisk()** メソッドを呼び出す前に、入力パラメータを構成してください。
5. **CreateVirtualDisk()** メソッドを呼び出します。
6. 選択したメソッドの出力パラメータ（戻りコード値）をチェックします。保留中の仮想ディスクのインスタンス ID は出力パラメータで、メソッドが正常に実行されると戻りコード値が返されます。例えば、メソッドが正常に実行されると、コード 0 が返されます。
7. **CreateTargetedConfigJob()** メソッドを呼び出す前に、入力パラメータ（例えば Target、RebootType、ScheduledStartTime、UntilTime、など）を構成し、コントローラの正しい FQDD（完全修飾デバイスディスクリプタ）を使用します。
8. **CreateTargetedConfigJob()** メソッドを呼び出し、保留中の値を適用します。
9. ジョブ制御プロファイルメソッドを使用して、ジョブ ID 出力のステータスをクエリします。
10. DCIM_VirtualDiskView クラスを列挙して、作成された仮想ディスクを表示します。

RAID の設定

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- PERC コントローラ、およびローカルキー管理をサポートするファームウェア
- SED ハードディスクドライブ

次のハードウェアリソースで RAID のセットアップと設定を行います。

- ストレージコントローラ — PERC
- 物理ディスク (SED) — 4 台
- 各物理ディスクのサイズ — 1 TB

次の RAID 構成を作成します。

- 仮想ディスクのサイズ — 10 GB (10240MB)
- 仮想ディスクの数 — 10
- RAID レベル — 5
- 専用ホットスペア — 1 台
- コントローラの暗号化を有効にして、ローカルキーを作成します。

RAID を設定するには、次を行います。

1. システムに接続されたストレージコントローラのリストと、コントローラのプロパティを取得します。後ほど使用するため、次のコントローラパラメータのステータスを検証し、書き留めてください。
 - コントローラの FQDD (完全修飾デバイスディスクリプタ)
 - セキュリティステータス
 - 暗号化モード
 - キー ID
2. コントローラに接続されている物理ディスクの FQDD と値を取得します。
3. 次の表の項目で正しい値を設定した後、**CreateVirtualDisk()** メソッドを実行します。




表 9. RAID セットアップ用の値


パラメータ	値
FQDD	コントローラおよび接続されている物理ディスクの FQDD
RAID レベル	RAID レベルを 5 に設定します。 RAID 5 は、物理ディスク間でデータをストライプし、パリティ情報を使用して冗長データを維持します。物理ディスクが 1 台故障した場合、パリティ情報を使用してデータが再構築されます。RAID 5 は、読み取りパフォーマンスとデータの冗長性が優れていますが、書き込みパフォーマンスは遅くなります。
スパン深度	値を 1 に設定します。
スパン長	値を 3 に設定します。 スパン長の値は、各スパンに含まれる物理ディスクの数を指します。この値は、物理ディスクの数を、スパン深度の値で割って計算されます。
サイズ	各仮想ディスクを 10240 MB に設定します。
開始 LBA	開始 LBA の値は、既存の仮想ディスクに基づいて算出します。512 バイトのブロックで次の開始 LBA を計算するには、次の式を使用します。



メモ: この操作は、スライスされた仮想ディスクがあるときのみ必要です。

- RAID0 — 直前の開始 LBA + ((サイズ ÷ ドライブ数) ÷ 512)
- RAID1 — 直前の開始 LBA + (サイズ ÷ 512)
- RAID5 — 直前の開始 LBA + ((サイズ ÷ ドライブ数 - 1)) ÷ 512)
- RAID6 — 直前の開始 LBA + ((サイズ ÷ ドライブ数 - 2)) ÷ 512)
- RAID10 — 直前の開始 LBA + ((サイズ ÷ 2) ÷ 512)
- RAID50 — 直前の開始 LBA + ((サイズ ÷ (スパンあたりのドライブ数 - 1)) ÷ 512)

パラメータ	値
	<ul style="list-style-type: none"> - RAID60 — 直前の開始 LBA + ((サイズ ÷ (スパンあたりのドライブ数 - 2)) ÷ 512)
	<p> メモ: このかわりに、開始 LBA を「0xFFFFFFFFFFFFFFF」に設定すると、仮想ディスクスライスの開始部分は、最後のスライスの終了直後になるように自動的に計算されます。</p>
ストライプサイズ	<p>ストライプエレメントのサイズとは、そのストライプにある各物理ディスク上でストライプが消費するディスク容量です。ビット単位で次の値を設定できます。</p> <p> メモ: S110 および H310 コントローラは、64KB のストライプのみをサポートします。</p> <ul style="list-style-type: none"> - 64KB = 128 - 128KB = 256 - 256KB = 512 - 512KB = 1024 - 1MB = 2048
読み取りポリシー	<ul style="list-style-type: none"> - 先読みなし - 先読み - 適応先読み
書き込みポリシー	<ul style="list-style-type: none"> - ライトスルー - ライトバック - ライトバックの強制
ディスクキャッシュポリシー	<ul style="list-style-type: none"> - 有効 - 無効
仮想ディスクの名前	<p>オプションで、仮想ディスクに名前を付けることができます。最长で 115 文字までの英数字を使用できます。</p>
	<ol style="list-style-type: none"> 10 個の仮想ディスクを作成するには、このメソッドを上表にリストされているものと同じ値でさらに 9 回実行します。 仮想ディスクが作成されたことを確認してください。 次の値を指定して、EnableControllerEncryption() メソッドを呼び出します。 <ul style="list-style-type: none"> - コントローラの FQDD (完全修飾デバイスディスクリプタ)。 - 暗号化モード — ローカルキー暗号化。 - キー ID。 - パスフレーズ — 有効なパスフレーズは、8~32 文字で構成されます。文字列中には、大文字、小文字、数字、記号を含める必要があり、スペースは使用できません。 スペアとして使用される物理ディスクの FQDD を使用して、AssignSpare() メソッドを呼び出します。 <p> メモ: 専用ホットスペアが必要な場合は、コントローラの FQDD を使用するのではなく、関連する仮想ディスクの FQDD をターゲットとして使用します。</p> CreateTargetedConfigJob() メソッド用に入力パラメータ (例えば、Target、RebootType、ScheduledStartTime など) を構成します。サポートされている入力パラメータの一覧は、delltechcenter.com/page/DCIM.Library で、RAID プロファイルマニュアルを参照してください。 CreateTargetedConfigJob() メソッドを呼び出し、保留中の値を適用します。このメソッドが成功すると、システムには作成された設定タスクにジョブ ID を返す必要が生じます

 **メモ:** タスクを実行するには、システムを再起動する必要があります。

RAID のセットアップ - 設定後のシナリオ

1. 先行して生成したジョブ ID を使用して、ジョブステータスを取得します。
2. RAID 設定およびローカルキーベースのコントローラ暗号化の有効化が正しく行われたことをチェックするには、システムが自動的に Lifecycle Controller から起動し、RAID 設定の変更およびローカルキーを適用することを検証する必要があります。
3. 先行して生成され、ジョブが正常に終了しました というステータスメッセージが返されたジョブ ID を使って、ジョブステータスを取得します。
4. [手順 1](#) と [手順 2](#) を繰り返して、変更が適用されたことを確認します。

RAID 設定のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 10. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 1	16.7 — RAID インベントリのリスト - ControllerView クラス
手順 2	16.9 — RAID インベントリのリスト - PhysicalDiskView クラス
手順 3	16.18.5 — 仮想ディスクスライスの作成 - CreateVirtualDisk
手順 4	16.18.5 — 仮想ディスクスライスの作成 - CreateVirtualDisk
手順 5	16.10 — RAID VirtualDiskView インベントリのリスト
手順 6	16.17.3 — キーによるコントローラのロック - EnableControllerEncryption
手順 7	16.16.2 — ホットスペアの割り当て - AssignSpare()
手順 8	16.14 — 保留中の値の RAID への適用 - CreateTargetedConfigJob
手順 9	16.14 — 保留中の値の RAID への適用 - CreateTargetedConfigJob

プロファイル

DCIM-SimpleRAIDProfile

MOF

- DCIM_ControllerView.mof
- DCIM_EnclosureView.mof
- DCIM_PhysicalDiskView.mof
- DCIM_RAIDAttribute.mof
- DCIM_RAIDEnumeration.mof
- DCIM_RAIDInteger.mof
- DCIM_RAIDService.mof

- DCIM_RAIDString.mof
- DCIM_VirtualDiskView.mof

SATA ドライブの RAID モードから非 RAID 状態への変換

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- 非 RAID モードをサポートする PERC S110 または H310 コントローラ。
- SATA または SSD ハードディスクドライブ

RAID ドライブを非 RAID SATA ドライブに変換するには、次の手順を行います。

1. システムに接続されたストレージコントローラのリストと、コントローラのプロパティを取得します。
2. FQDD、コントローラの値、コントローラに接続している物理ディスクを取得します。
3. **ConvertToNonRAID()** メソッドを呼び出し、変換を開始します。
4. **CreateTargetedConfigJob()** メソッドを呼び出し、保留中の値を適用します。このメソッドが成功するなら、システムは作成された設定タスクにジョブ ID を返す必要が生じます。

SATA ドライブの変換のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 11. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順1	16.7 — RAID インベントリのリスト - ControllerView クラス
手順2	16.9 — RAID インベントリのリスト - PhysicalDiskView クラス
手順3	16.21 — 物理ディスクの非 RAID への変換 - ConvertToNonRAID()
手順4	16.14 — 保留中の値の RAID への適用 - CreateTargetedConfigJob

プロファイル

DCIM-SimpleRAIDProfile

MOF

- DCIM_ControllerView.mof
- DCIM_EnclosureView.mof
- DCIM_PhysicalDiskView.mof
- DCIM_RAIDAttribute.mof
- DCIM_RAIDEnumeration.mof
- DCIM_RAIDInteger.mof
- DCIM_RAIDService.mof
- DCIM_RAIDString.mof

- DCIM_VirtualDiskView.mof

ネットワークデバイスの管理

ネットワーク管理機能を使用して、システム内の次のネットワークデバイスの詳細リストを取得し、各デバイスの属性を設定します。

- ネットワークインタフェースカード (NIC)
- 統合型ネットワークアダプタ (CNA)
- マザーボードに搭載の LAN (LOM)
- ネットワークドーターカード (NDC)
- メザニンカード (ブレードサーバーのみ)

シンプル NIC プロファイルの詳細については、「[シンプル NIC プロファイル](#)」を参照してください。

ネットワークデバイスインベントリの表示

- DCIM_NICView クラスで列挙操作を実行し、システム内のネットワークデバイスすべての (Broadcom および Intel) インスタンスプロパティを表示します。
- 必要なネットワークデバイスの正しいインスタンス ID を使用してクラスで取得操作を実行し、関連するプロパティを表示します。


ネットワークデバイス属性の表示

- DCIM_NICAttribute クラス (DCIM_NICEnumeration, DCIM_NICInteger, DCIM_NICString) のうちいずれか1つで列挙操作を実行し、システム内のネットワークデバイスすべての使用可能な属性と可能な値を表示します。
- DCIM_NICAttribute クラスのいずれかで取得操作を実行して、ネットワークデバイスの属性を表示します。特定のサブクラスの属性情報を表示するには、サブクラスにリストされている属性名と共に、正しいインスタンス ID を使用してください。

ネットワークデバイス属性の設定

属性を設定するには、次の手順を実行してください。

1. 該当するインスタンス ID を識別し、インスタンス情報をメモします。
2. IsReadOnly フィールドが、false に設定されていることを確認します。
3. インスタンス情報を使用して、入力パラメータを準備します。
4. **SetAttribute()** または **SetAttributes()** メソッドを呼び出します。
5. 属性で取得コマンドを実行し、保留中フィールドにアップデートされた値を表示します。
6. **CreateTargetedConfigJob()** メソッドを呼び出す前に、入力パラメータ (例えば Target、RebootType、ScheduledStartTime、UntilTime など) を構築し、ターゲット用のネットワークデバイスの正しい FQDD (完全修飾デバイスディスクリプタ) を使用します。サポートされている入力パラメータのリストは、delltechcenter.com/page/DCIM.Library でシンプル NIC プロファイルのマニュアルを参照してください。
7. **CreateTargetedConfigJob()** メソッドを呼び出し、保留中の値を適用します。このメソッドが成功すると、システムには作成された設定タスクにジョブ ID を返す必要が生じます


 **メモ:** 1つまたは複数の属性を設定するタスクを実行するには、システムを再起動してください。

8. ジョブ制御プロファイルメソッドを使用して、ジョブ ID 出力のステータスをクエリします。
9. 手順 5 を繰り返し、メソッドが正しく実行されたことを確認します。

保留中の値の削除


保留中の値を削除するには、次の手順を実行します。

1. DCIM_JobService クラスで **DeletePendingConfiguration()** メソッドを呼び出す前に、入力パラメータを構成し、ネットワークデバイスの正しい FQDD (完全修飾デバイスディスクリプタ) を使用します。

 **メモ:** 保留中データを削除できるのは、ターゲットジョブの作成前のみです。ターゲットジョブが作成された後は、このメソッドを実行することはできません。必要に応じて **DeleteJobQueue()** メソッドを呼び出し、ジョブを削除して保留中の値をクリアすることができます。ただし、システムが再起動し、ジョブの実行が開始された場合は、メソッドは機能しません。


2. **DeletePendingConfiguration()** メソッドを呼び出します。
3. 戻されたメソッドの戻りコード値によって削除を確認できます。

CNA のパーティションの有効化または無効化


 **メモ:** NicPartitioning プロパティや PartitionState プロパティを無効にしても、パーティション 1 を無効にすることはできません。

CNA のパーティションを有効または無効にするには、次の手順を実行します。


1. DCIM_NICEnumeration クラスを列挙し、NicMode、iScsiOffloadMode を持つクラスのインスタンスの現在の値、およびそれらの FQDD プロパティを識別します。
2. 識別されたパーティションについて、FQDD プロパティを使用して **SetAttribute()** メソッドを呼び出し、パーティションを有効または無効にします。

 **メモ:** モードのいずれかがアクティブになっていても、パーティションは有効になっています。

3. 属性で取得コマンドを実行し、保留中フィールドにアップデートされた値を表示します。
4. **CreateTargetedConfigJob()** メソッドを呼び出す前に、入力パラメータ (Target、RebootJobType、ScheduledStartTime、UntilTime など) を構成します。
ポートの複数のパーティションに設定の変更がある場合は、**RebootJobType** および **ScheduledStartTime** を指定しないでください。ジョブ制御プロファイルメソッドを使用してジョブをスケジュールします。手順 6 に進んでジョブを作成します。サポートされているすべてのパラメータのリストは、delltechcenter.com/page/DCIM.Library でシンプル NIC プロファイルのマニュアルを参照してください。
5. **CreateTargetedConfigJob()** メソッドを呼び出し、保留中の値を適用します。このメソッドが成功すると、システムは作成された設定タスクにジョブ ID を返します。

 **メモ:** 1つまたは複数の属性を設定するタスクを実行するには、システムを再起動してください。

6. **CreateRebootJob()** で再起動ジョブを作成し、**SetupJobQueue()** を使用してすべてのパーティションジョブおよび再起動ジョブをスケジュールします。

 **メモ:** パーティションジョブが同時に実行されるようにスケジュールされない場合、パーティションで保留中の変更は失われます。

7. ジョブ制御プロファイルメソッドを使用して、ジョブ ID 出力のステータスをクエリします。
8. 手順 1 を繰り返し、メソッドが正しく実行されたことを確認します。

CNA 用パーティションのパーソナリティと帯域幅の変更

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」で説明されている前提条件が満たされていることを確認してください。

10 Gb のイーサネットリンクを備え、複数のパーソナリティをサポートする統合型ネットワークアダプタカード (CNA) カード上で、ポートのパーティショニング、およびパーソナリティと帯域幅の割り当てを行います。


次のパーソナリティと帯域幅をセットアップする必要があります。

表 12. パーソナリティと帯域幅

パーソナリティの数	2
各パーティションのパーソナリティ	帯域幅
iSCSI	50
FCoE	50

CNA のパーティションのパーソナリティを変更し、帯域幅を設定するには、次の手順を行います。

1. DCIM_NICEnumeration クラスを列挙し、AttributeName=NicMode/FCoEOffloadMode/iScsiOffloadMode を持つクラスのインスタンスの現在値、およびそれらの FQDD のプロパティを識別します。
2. 識別されたパーティションについて、特定のパーソナリティを有効化し、その他のパーソナリティを無効化するため、FQDD プロパティを使用して **SetAttribute()** メソッドを呼び出します。


 **メモ:** パーティションでは複数のパーソナリティがサポートされているため、複数のパーソナリティを同時に有効化または無効化することができます。異なる CNA カード上でのパーソナリティ設定に関する制限については、『[iDRAC7 バージョン 1.00.00 Readme](#)』、または delltechcenter.com/page/DCIM.Library でシンプル NIC プロファイルのマニュアルを参照してください。

3. **手順 6**に進んで、残りの手順を完了します。
4. DCIM_NICInteger クラスを列挙し、AttributeName=MaxBandwidth または MinBandwidth を持つクラスのインスタンスの現在値、またはそれらの FQDD プロパティを識別します。帯域幅の最大値と最小値です。
 - 20~30
 - 30~40
 - 25~35

異なる CNA カード上での帯域幅設定に関する制限については、『[iDRAC7 バージョン 1.00.00 Readme](#)』、または delltechcenter.com/page/DCIM.Library でシンプル NIC プロファイルのマニュアルを参照してください。

5. 識別されたパーティションについて、FQDD を使用して **SetAttribute()** メソッドを呼び出し、帯域幅を変更します。
6. 属性の保留中フィールドで、更新された値を確認します。
7. **CreateTargetedConfigJob()** メソッドを呼び出す前に、入力パラメータ (Target、RebootJobType、ScheduledStartTime、UntilTime など) を構成します。

ポートの複数のパーティションに設定の変更がある場合は、RebootJobType および ScheduledStartTime を指定しないでください。ジョブ制御プロファイルメソッドを使用してジョブをスケジュールします。手順 9 に進んでジョブを作成します。サポートされているすべての入力パラメータのリストは、delltechcenter.com/page/DCIM.Library でシンプル NIC プロファイルのマニュアルを参照してください。
8. **CreateTargetedConfigJob()** メソッドを呼び出し、保留中の値を適用します。このメソッドが成功するなら、システムは作成された設定タスクにジョブ ID を返す必要が生じます。

-  **メモ:** 1つまたは複数の属性を設定するタスクを実行するには、システムを再起動してください。
9. **CreateRebootJob()** を使用して再起動ジョブを作成し、**SetupJobQueue()** を使用してすべてのパーティションジョブと再起動ジョブをスケジュールします。パーティション上の保留中の変更は、同時に実行されるようにスケジュールされていない場合は失われます。
 10. ジョブ制御プロファイルメソッドを使用して、ジョブ ID の出力のステータスをクエリすることができます。
 11. 手順 4 を繰り返し、メソッドが正しく実行されたことを確認します。

パーソナリティの変更のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。


表 13. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 1	15.1 — CNA インベントリのリスト - Enumeration クラス
手順 2	15.14 — CNA LAN モードの設定
手順 4	15.3 — CNA インベントリのリスト - Integer クラス
手順 5	15.11 — MaxBandwidth 属性の設定
手順 6	15.3 — CNA インベントリのリスト - Integer クラス
手順 7	15.7 — 保留中の値の CNA への適用 - CreateTargetedConfigJob()
手順 8	15.5 — 保留中の値の CNA への適用 - CreateTargetedConfigJob()
手順 9	7.8 — CreateRebootJob() 10.2.1 — ジョブキューの設定
手順 10	10.2.3 — ジョブストア内のジョブのリスト
プロファイル	
delltechcenter.com/page/DCIM.Library のシンプル NIC プロファイルマニュアル	
MOF	
<ul style="list-style-type: none"> • DCIM_NICView • DCIM_NICString • DCIM_NICEnumeration • DCIM_NICInteger • DCIM_NICAttribute • DCIM_NICService 	

仮想アドレス属性の設定

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」で説明されている前提条件が満たされていることを確認してください。

CNA カードの仮想アドレス属性を変更します。

 **メモ:** システムが AC 電源接続から切断されると、すべての仮想アドレス属性がデフォルトにリセットされます。

次の各仮想アドレス属性それぞれに適切な値を設定します。

- VirtMacAddr
- VirtIscsiMacAddr
- VirtFIPMacAddr
- VirtWWN
- VirtWWPN

仮想アドレス属性の参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 14. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
-	15.12 — VirtMacAddr 属性の設定
プロファイル	
DCIM_SimpleNIC プロファイル	
MOF	
<ul style="list-style-type: none"> • DCIM_NICView • DCIM_NICString • DCIM_NICEnumeration • DCIM_NICInteger • DCIM_NICAttribute • DCIM_NICService 	

起動ターゲット iSCSI および FCoE の設定

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」で説明されている前提条件が満たされていることを確認してください。

CNA カードの iSCSI および FCoE の属性を変更します。

iSCSI および FCoE 属性を変更するには、次の手順を行います。

- iSCSI イニシエータ属性を設定するには、次のそれぞれについて適切な値を設定します。
 - IscsiInitiatorIpAddr
 - IscsiInitiatorSubnet
 - IscsiInitiatorSubnetPrefix
 - IscsiInitiatorGateway
 - IscsiInitiatorPrimDns
 - IscsiInitiatorSecDns
 - IscsiInitiatorName
 - IscsiInitiatorChapId
 - IscsiInitiatorChapPwd
- iSCSI の最初のターゲットを設定するには、次のそれぞれについて適切な値を設定します。
 - ConnectFirstTgt
 - FirstTgtIpAddress

- FirstTgtTcpPort
- FirstTgtBootLun
- FirstTgtIscsiName
- FirstTgtChapId
- FirstTgtChapPwd
- FCoE 起動ターゲットを設定するには、次のそれぞれについて適切な値を設定します。
 - MTUParams
 - ConnectFirstFCoETarget
 - FirstFCoEWWPNTarget
 - FirstFCoEBootTargetLUN
 - FirstFCoEFCFVLANID

インベントリとログ

インベントリとログの機能を使用して、次の操作を行います。

- 現在および工場出荷時のインベントリの取得とエクスポート
- ライフサイクルログの取得とエクスポート
- システムのリセット

ハードウェアインベントリの取得

Remote Services を使用して、システムのハードウェアインベントリを即時に取得できます。インベントリにはシステムに取り付けられているすべてのハードウェアデバイスのリストがあります。

ハードウェアインベントリの情報は、**Lifecycle Controller** の永続ストレージにキャッシュされ、**iDRAC** および **UEFI** アプリケーションで利用可能です。


ハードウェアのインベントリを取得するには、ファン、電源装置、**iDRAC**、ビデオコントローラ、**CPU**、**DIMM**、および **PCI/PCIe** などの異なるシステムハードウェアのプロパティを表示するため、それらの表示クラスを列挙する必要があります。

異なるハードウェアのプロファイルの詳細については、「[ハードウェアインベントリプロファイル](#)」を参照してください。

ハードウェアコンポーネントの使いやすい名前前の詳細については、「[使いやすいシステムコンポーネント名](#)」を参照してください。

現在のハードウェアインベントリのエクスポート

- 現在のハードウェアインベントリを **XML** ファイルにエクスポートするには、**DCIM_LCService** クラスで **ExportHWInventory()** メソッドを呼び出します。
- 管理下ノードの工場出荷時デフォルト値のコピーを保存するには、**DCIM_LCService** クラスで **ExportFactoryConfiguration()** メソッドを呼び出します。

 **メモ:** XML ファイルを USB デバイス、ネットワーク共有、または両方の場所に保存します。


ライフサイクルログ

ライフサイクルログは、次の情報を表示します。

- **iDRAC** 設定の変更
- リモート書き込み操作およびユーザー認証エラーすべてのログ
- デバイス、バージョン、日付別のファームウェアアップデート履歴。
- **BIOS** および **NIC** 設定の変更。
- **RAID** 設定の変更。
- エラーメッセージの ID。詳細については、support.dell.com/manuals のエラーメッセージレジストリを参照してください。
- 重要度、カテゴリ、日付別のイベント（アップデートおよび設定のみ）

 **メモ:** 設定変更の詳細は表示されません。

- 日付別のカスタマーコメント

 **メモ:** ライフサイクルログは、オペレーティングシステムがシステムにインストールされていなくても利用可能で、システムの電源状態にも依存しません。


ライフサイクルログのエクスポート


この機能を使用して、ライフサイクルログの情報を XML ファイルにエクスポートします。XML ファイルは USB デバイス、ネットワーク共有、あるいはそれら両方に保存します。

ライフサイクルログをエクスポートするには、DCIM_LCService クラスで **ExportLCLog()** メソッドを呼び出します。スキーマについての詳細は、「[ライフサイクルログスキーマ](#)」を参照してください。

設定の削除とデフォルトへのリセット

この機能を使って、管理対象ノードの回収、管理対象ノードの別のアプリケーションでの再利用、管理対象ノードのセキュアでない場所への移動等が必要となった場合に機密データおよび設定関連情報を削除します。


 **注意:** この機能は iDRAC を工場出荷時のデフォルトにリセットし、すべての iDRAC ユーザー資格情報および IP アドレスの設定を削除します。また、すべての変更イベント、ファームウェアアップグレード、ユーザーコメント、証明書、ExportedFactoryConfiguration 情報、ファームウェアロールバックファイル、およびライセンスファイルすべての履歴が含まれたライフサイクルログを削除します。この機能を使用する前に、ライフサイクルログを安全な場所にエクスポートすることを推奨します。この操作後、システムを手動でシャットダウンしてから、システムに電源を投入します。

 **メモ:** 設定を削除する前に、ライフサイクルログおよび ExportedFactoryConfiguration の情報をバックアップしてください。

設定を削除して工場出荷時のデフォルト値にリセットするには、DCIM_LCService で **LCWipe()** メソッドを呼び出します。

Lifecycle Controller リセット後のハードウェアインベントリの表示とエクスポート

Lifecycle Controller のリセット後に、誤ったインベントリデータが表示されたり、XML ファイルにエクスポートされたりします。Lifecycle Controller をリセットした後に、正確なハードウェアインベントリデータを表示、またはエクスポートするには、次の手順を実行します。

 **メモ:** Lifecycle Controller のリセット後、システムを手動でシャットダウンします。

1. システムに電源を投入して、iDRAC が機能し始めるまで数分待ちます。
2. 電源コードを外し、30 秒待ちます。
3. 電源コードを再度接続し、システムを起動して DCIM_LCService クラスで **ExportHWInventory()** メソッドを呼び出します。

リモートアップデート

リモートアップデートおよびファームウェアインベントリ機能を使用して、オペレーティングシステムに依存しないアップデートを実行し、ファームウェアインベントリを取得します。

リモートアップデートの使用


帯域外アップデートまたはオペレーティングシステムに依存しないプラットフォームアップデートとしても知られるリモートアップデートでは、オペレーティングシステムの状態に関係なく、システムをアップデートできます。ファームウェアアップデートは、システム状態（電源がオンまたはオフ）に関係なく開始できます。


オペレーティングシステムに依存しないプラットフォームアップデートでは、システムでオペレーティングシステムが稼動している必要はありません。アップデートを実行するために、**Lifecycle Controller** システムサービスからの正常な再起動またはパワーサイクル再起動と共に同時に複数のアップデートをスケジュールできます。アップデートは **BIOS** の中間再起動を伴う場合がありますが、アップデートが完了するまでは、**Lifecycle Controller** がそれらを自動的に処理します。

この機能は、次の 2 通りのアップデート方法をサポートしています。

- **URI (Uniform Resource Identifier) からのインストール**— この方法では、URI を使用したホストプラットフォームへのソフトウェアのインストールまたはアップデートの **WS-Management** 要求を可能にします。URI は、ネットワーク上のリソースの識別および名前付けに使用される文字列で構成されます。URI は、**Lifecycle Controller** にダウンロードしてインストールできる **Dell** アップデートパッケージイメージのネットワーク上の場所を指定するために使用します。
- **ソフトウェア ID からのインストール**— この方法では、**Lifecycle Controller** で既に使用可能なバージョンへのアップデートまたはロールバックができます。

リモートアップデートには、**WS-Management** 対応のアプリケーション、スクリプト、またはコマンドラインユーティリティを使用できます。アプリケーションやスクリプトは、リモートアップデートインタフェースメソッドの 1 つを使用して **WS-Management** 呼び出しメソッド要求を実行します。実行後、**iDRAC** はネットワーク共有（ローカルネットワーク共有、**CIFS**、**NFS**、**FTP**、**TFTP**、**http**) URI からファームウェアをダウンロードし、指定した時間に、指定した正常、パワーサイクル、または強制システム再起動タイプを活用してアップデートが実行されるようにステージングします。

 **メモ:** システムのドライバパックでリモートアップデートを実行すると、現在のドライバパックが置き換えられます。置き換えられたドライバパックは、使用できなくなります。

 **メモ:** 英数字のパス名のみがサポートされます。

対応デバイス

リモートアップデートは、次のデバイスおよびコンポーネントでサポートされています。

- **iDRAC7**
- **RAID シリーズ 6 および 7**
- **NIC**、**LOM**、**NDC**、および **CNA** (**Broadcom**、**Intel**、および **QLogic**)
- 電源装置
- **BIOS**


- オペレーティングシステムのドライバパック
- Lifecycle Controller
- 診断

URI からのリモートアップデート


URI を使用してリモートアップデートを実行するには、次の手順を行います。


1. iDRAC の IP アドレスにメソッド呼び出し要求を送信するには、適切な **WS-Management** クライアントを使用します。**WS-Management** コマンドには、DCIM_SoftwareInstallationService の **InstallFromURI()** メソッド、および iDRAC が Dell アップデートパッケージ (DUP) をダウンロードする場所が含まれます。サポートされているダウンロードプロトコルは、FTP、HTTP、CIFS、NFS、および TFTP です。

コマンドが正しく呼び出されると、ジョブ ID が返されます。

 **メモ:** 別のアップデートジョブを作成するために、WS-Management を使用して追加の **InstallFromURI()** メソッド呼び出しの要求を送信することもできます。

2. DCIM_SoftwareInstallationService で **CreateRebootJob()** メソッドを呼び出して再起動ジョブを作成し、目的の再起動タイプ (シャットダウンを強制しない正常再起動、パワーサイクル、シャットダウンを強制する正常再起動) を指定します。
3. アップデートと再起動のジョブ ID を使用することにより、Dell ジョブ制御プロファイルを使ってこれらのジョブを即座に実行、または後で実行するように日時をスケジュールできます。また、ジョブ ID を使用してジョブのステータスのクエリや、ジョブのキャンセルを行うこともできます。
4. すべてのジョブは成功とマークされますが、ダウンロード中またはアップデート中にエラーが発生した場合は失敗とマークされます。失敗したジョブについては、失敗に関するエラーメッセージとエラーメッセージ ID をジョブ情報から利用できます。

 **メモ:** DUP を正常にダウンロードして解凍した後、ダウンロードがジョブのステータスをダウンロード済みにアップデートし、ジョブをスケジュールできるようになります。署名が無効な場合や、ダウンロードまたは解凍に失敗した場合は、ジョブのステータスが失敗に設定され、該当するエラーコードが付きます。


 **メモ:** アップデートされたファームウェアのバージョンを表示するには、ファームウェアアップデートジョブの完了後にファームウェアインベントリを列挙します。

リモートアップデートのスケジュール作成


ファームウェアアップデートは、今すぐ、または後日実行するようスケジュール (ステージング) することができます。診断、Lifecycle Controller、および OS ドライバパックのアップデートは、ステージングを行わずに直接実行することができます。これらのアップデートはダウンロード後にすぐに適用されるため、ジョブスケジュールは不要です。その他のリモートアップデートはすべてステージングされたアップデートで、異なるスケジュールオプションを使用してスケジュールする必要があります。DUP は Lifecycle Controller にダウンロードされてステージングが行われ、その後システムが UEFI システムサービスから再起動して、実際のアップデートを実行します。

アップデートのスケジュールには、複数のオプションがあります。

- 任意の時間に任意のコンポーネントのアップデートを実行する。
- 再起動ジョブ ID を取得するために、再起動コマンドを実行する。
- ジョブのステータスはいずれも、DCIM_SoftUpdateConcreteJob インスタンスを列挙し、JobStatus プロパティ値を確認することによってチェックします。
- DCIM_JobService で **SetupJobQueue()** メソッドを使用してジョブをスケジュールする。


 **メモ:** Remote Services バージョン 1.3 でのリモートアップデートでは、**SetupJobQueue()** メソッドのみの使用が可能です。

- DCIM_JobService で **DeleteJobQueue()** メソッドを使用して既存のジョブを削除する。

 **メモ:** アップデートがスケジュールされているときにシステムをシャットダウンしたり、15分以上シャットダウン状態にする場合は、スケジュールされているジョブを削除することをお勧めします。

前のバージョンへのロールバック

Lifecycle Controller に保存されている前のバージョンのファームウェアから再インストールするには、**InstallFromSoftwareIdentity()** メソッドを使用します。DUP をダウンロードする代わりに、**InstallFromSoftwareIdentity()** はジョブを作成してジョブ ID を返します。

 **メモ:** Lifecycle Controller、診断、およびドライバパックのアップデートはロールバックできません。

リモートファームウェアインベントリの使用

リモートファームウェアインベントリ機能は、管理下システムの様々なデバイスに現在インストールされているファームウェアのインベントリを取得します。また、ロールバックに使用できるバージョン (N および N-1 バージョン) も取得できます。

リモートファームウェアインベントリ機能では、システムの状態およびオペレーティングシステムの状態に依存しないインベントリを実行できます。取り付けられているデバイスのファームウェアのほか、ロールバックと再インストールに使用できるファームウェアのリストも取得できます。

 **メモ:** WS-Management 要求の認証に使用される iDRAC ユーザー資格情報は、ファームウェアと内蔵ソフトウェアのインベントリを要求するためにログイン権限を必要としますが、この機能はシステム管理者に限定されるものではありません。

対応デバイス

リモートでの即時ファームウェアインベントリは、次のデバイスおよびコンポーネントでサポートされています。


- iDRAC7
- ストレージコントローラ (RAID シリーズ 7 および 8)
- NIC および LOM (Broadcom、Intel、および QLogic)
- 電源装置
- BIOS
- オペレーティングシステムのドライバパック
- Lifecycle Controller
- 診断
- 複合プログラム可能論理デバイス (CPLD)
- 物理ディスク
- エンクロージャ

インスタントファームウェアインベントリクラスは、次のファームウェアインベントリ情報を提供します。

- 対応デバイスにインストールされているファームウェア。
- 各デバイスにインストールできるファームウェアバージョン。

ファームウェアインベントリの取得


Dell Software Inventory プロファイルは、サーバーにインストール済み、またはインストール可能なファームウェアと埋め込みソフトウェアのバージョンを表す Dell CIM データモデル拡張を定義します。ファームウェアインベントリには、WS-Management Web サービスプロトコルを使用してアクセスできます。


-  **メモ:** 過去に取り付けられ、その後取り外されたハードウェアに DCIM_SoftwareIdentity インスタンスが存在する場合があります。CSIOR が実行されていない場合、そのようなインスタンスは使用可能としてリストされます。

Windows WS-Management を使用してファームウェアインベントリを取得するには、次の手順を実行してください。

1. DCIM_SoftwareIdentity クラスを列挙して、システムのインベントリを取得します。


インベントリは、インストール済みおよび使用可能な CIM インスタンスとして収集されます。

-  **メモ:** システムのファームウェアと組み込みソフトウェアのインベントリは、システム管理者権限またはサーバーコマンドの実行権限を持つユーザーが取得できます。

-  **メモ:** インベントリのインスタンスは、システムオフまたはシステムオンの両方の状態でシステムから取得されます。

- コンポーネントに現在インストールされているソフトウェアは、インストール済みソフトウェアインスタンスとしてリストされます。このインスタンスのステータス値はインストール済みとして表現されます。
- 持続的なストレージで利用できるソフトウェアは、利用可能なソフトウェアインスタンスとしてリストされます。このインスタンスの主なプロパティ値である InstanceID は、DCIM:AVAILABLE:<COMPONENTTYPE>:<COMPONENTID>:<Version> として表現され、同インスタンスのステータス値は「利用可能」となります。現在インストールされているソフトウェアのインスタンスも、利用可能なソフトウェアのインスタンスとして表現されます。

2. インベントリのインスタンスは、アップデートおよびロールバック操作のための入力値を提供します。アップデート操作を実行するには、インストール済みインスタンスである DCIM:INSTALLED:<comptype>:<compid>:<version> から InstanceID 値を選択します。ロールバック処理の場合は、利用可能なインスタンスの DCIM:AVAILABLE:<comptype>:<compid>:<version> から InstanceID 値を選択します。InstanceID 値は編集できません。

-  **メモ:** 利用可能なソフトウェアインスタンスのバージョン文字列のプロパティ値が、インストール済みソフトウェアインスタンスと同じ場合は、ロールバック操作にその利用可能なソフトウェアインスタンスの InstanceID 値を使用しないでください。

リモートスケジュールのタイプ

スケジュールには2つの方法があります。

- 即時アップデート
- スケジュール指定アップデート

即時アップデート

コンポーネントファームウェアを即座にアップデートするには、開始時間を TIME_NOW にして、アップデートと再起動ジョブをスケジュールします。Lifecycle Controller、診断、および OS ドライバパックなどの Lifecycle Controller コンポーネントへのアップデートには、再起動やアップデートのスケジュールは不要です。これらのコンポーネントのアップデートは即座に行われます。

スケジュール指定アップデート

SetupJobQueue() メソッドを使用して1つまたは複数のジョブの開始時間を指定するには、**StartTimeInterval** パラメータへの日時の指定が必要です。オプションで**UntilTime** パラメータに日時を指定することもできます。

UntilTime を指定すると、定められた時間内にアップデートを実行するメンテナンス枠が定義されます。時間枠を過ぎてアップデートが完了していない場合は、現在実行されているアップデートジョブは完了するまで続行されますが、スケジュールされた開始時間を過ぎた未処理のジョブは失敗します。

スケジュール再起動動作の設定

CreateRebootJob() メソッドは、次の再起動タイプのひとつを入力パラメータとして使用し、再起動ジョブ ID が出力パラメータとして返されます。再起動ジョブ ID は **SetupJobQueue()** メソッドの **JobArray** パラメータの最後のジョブ ID として、他のアップデートジョブ ID と共に使用されます。

- **再起動 1 - パワーサイクル**— システムの電源を切ってからシステムに電源を投入する管理下サーバーのパワーサイクルを実行します。これは正常な再起動ではありません。システムは、オペレーティングシステムにシャットダウン要求を送信することなく電源を切ります。システムが切れた状態で A/C 電源が引き続き供給されているという場合、システムに電源投入するのは再起動タイプ 1 のみです。
- **再起動 2 - シャットダウンを強制しない正常再起動**— 管理下サーバーの正常なシャットダウンを実行します。パワーサイクルの待機時間内にシステムの電源がオフになった場合は、システムに再び電源が投入され、再起動ジョブが **再起動完了** とマークされます。パワーサイクルの待機時間内にシステムの電源がオフにならなかった場合、再起動ジョブは失敗とマークされます。
- **再起動 3 - シャットダウンを強制する正常再起動**— 管理下サーバーの正常なシャットダウンを実行します。パワーサイクルの待機時間内にシステムの電源がオフになった場合は、システムに再び電源が投入され、再起動ジョブが **再起動完了** とマークされます。パワーサイクルの待機時間内にシステムの電源がオフにならなかった場合、システムはパワーサイクルを行います。

部品交換の管理

部品交換機能は、RAID コントローラ、NIC、または電源装置などの新しく交換されたコンポーネントのファームウェアまたは設定、もしくはこれら両方の自動アップデートを行い、それらが元の部品と同じになるようにします。これはライセンスが必要な機能で、iDRAC7 エンタープライズのライセンスがインストールされている場合に利用可能になります。コンポーネントの交換時に部品交換機能が有効になっている場合は、**Lifecycle Controller** の処置がシステムモニタにローカルに表示されます。

部品交換に関連するプロパティは、**WS-management** 対応の様々なユーティリティを使用して設定できます。詳細については、『*Lifecycle Controller Web サービスインターフェースガイド (Windows または Linux)*』を参照してください。DCIM プロファイル仕様および関連 MOF ファイルは、**DCIM 拡張ライブラリ領域 (delltechcenter.com)** の Dell TechCenter wiki から入手できます。

- 📌 **メモ:** SAS カードでは、ファームウェアのアップデートのみがサポートされています。SAS カードでは属性値を設定できないため、設定のアップデートはサポートされません。
- 📌 **メモ:** 部品交換は様々なメーカーの多くのサーバーコンポーネントでサポートされています。完全な対応リストは、support.dell.com/manuals にある『*iDRAC7 バージョン 1.00.00 Readme*』を参照してください。

部品ファームウェアと設定アップデート属性の取得と設定

WS-Management を使用して現在の **部品ファームウェアアップデート** および **再起動時のシステムインベントリ収集** プロパティの値を取得するには、列挙コマンド要求を送信して **DCIM_LCEnumeration** クラスのインスタンスを取得することができます。属性ごとに **DCIM_LCEnumeration** インスタンスオブジェクトが返され、このオブジェクトの **AttributeName** 文字列プロパティには、**Part Firmware Update** といった部品交換に関

連するプロパティの名前が含まれています。CurrentValue プロパティには、プロパティの現在の設定が含まれています。特定の属性名と値については、**Dell Lifecycle Controller 管理プロファイル仕様**を参照してください。属性名と値には次のようなものがあります。

- AttributeName - Part Configuration Update
- PossibleValues - Disabled, Apply always, Apply only if firmware match
- AttributeName - Part Firmware Update
- PossibleValues - Disable, Allow version upgrade only, Match firmware of replaced part

部品交換に関連するプロパティの値の設定には、**WS-Management** ウェブサービスプロトコルを使用して、設定および適用の処置が要求されます。

DCIM_LCService クラスで **SetAttribute()** メソッドを呼び出し、属性を設定します。**SetAttribute()** メソッドはプロパティ名と値を入力パラメータとして受け入れます。この表には、部品ファームウェアと設定アップデートの値がリストされています。

オプション	値
部品ファームウェアアップデート	
Allow version upgrade only	CurrentValue の入力値が Allow version upgrade only の場合、新しい部品のファームウェアバージョンが元の部品より古い場合に、交換部品のファームウェアアップデートが実行されます。
Match firmware of replaced part	CurrentValue の入力値が Match firmware of replaced part の場合、新しい部品のファームウェアは元の部品のファームウェアバージョンにアップデートされます。
Disable	入力値が Disable の場合、ファームウェアのアップグレード処置は実行されません。
部品設定のアップデート	
Apply always	部品が交換されると、現在の設定が適用されます。
Apply only if firmware match	現在の設定は、現在のファームウェアが交換した部品のファームウェアと一致する場合にのみ適用されます。
Disabled	部品が交換されても、現在の設定は適用されません。

DCIM_LCService クラスで **CreateConfigJob()** メソッドを呼び出し、値を適用します。**CreateConfigJob()** メソッドはスケジュールされた開始時間 (TIME_NOW も可)、およびフラグが必要な場合は再起動をパラメータとして受け入れます。ジョブ ID がパラメータとして返され、ジョブの完了ステータスの確認に使用できます。


バックアップと復元

エクスポートとインポート機能を使用して、サーバープロファイルのバックアップ、エクスポート、復元を行います。

iDRAC vFlash カードまたはネットワーク共有へのサーバープロファイルのエクスポート

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- iDRAC7 Enterprise のライセンスがインストールされている。cccc
- サーバーに有効な 7 桁のサービスタグがある。
- エクスポート先に読み取りおよび書き込みアクセスがある。
- iDRAC vFlash カード：
 - 取り付け、有効化および初期化が完了している。
 - 少なくとも 384 MB の空き容量が利用可能。
- ネットワーク共有：
 - ネットワーク共有を持つシステムとの通信を行うための、iDRAC に関する許可とファイアウォール設定が行われている。
 - 正しい機能ライセンスがインストールされている。
 - 少なくとも 384 MB の空き容量が利用可能。

 **メモ: BackupImage()** メソッドを呼び出すとバックアップイメージファイルがネットワーク共有に作成されます。そのサイズは、システム構成に応じて 30~384 MB になります。

- iDRAC では、サーバーコマンド実行権限が利用できます。

サーバーファームウェアおよび設定のバックアップを作成して、それを iDRAC vFlash カードまたはネットワーク共有にエクスポートします。バックアップファイルは、パスワードによってセキュア化されています。次のバックアップを行います。

- BIOS、LOM、Lifecycle Controller 対応のアドイン NIC カード、およびストレージコントローラ (RAID レベル、仮想ディスク、コントローラの属性) などの、ハードウェアおよびファームウェアのインベントリ。
- サービスタグ、システムタイプなどのシステム情報。
- Lifecycle Controller のファームウェアイメージ、システム設定、および iDRAC ファームウェアと設定。

重要なメモ

- エクスポート実行中は、ファームウェアアップデート、オペレーティングシステムの展開、およびファームウェアの設定などの操作が実行されないようにしてください。オペレーティングシステムの展開が Lifecycle Controller を使用して行われる場合は、エクスポートを実行する前に iDRAC をリセット、または Lifecycle Controller をキャンセルしてください。
- Lifecycle controller を使用したオペレーティングシステムの展開後、OEMDRV が 18 時間オープン状態になります。これは Lifecycle Controller にオペレーティングシステムインストールのステータスがない

ためです。オペレーティングシステム展開後に、アップデート、設定、復元などの操作を実行する必要がある場合は、**OEMDRV** パーティションを削除してください。パーティションを削除するには、**iDRAC** をリセットするか、**Lifecycle Controller** をキャンセルします。

- **BIOS** アップデート、**NIC** 属性の設定などの、他のリモートサービスジョブをスケジュールしないでください。
- **ScheduledStartTime** パラメータを使用しない場合、ジョブ ID は返されますが、スケジュールはされません。ジョブをスケジュールするには、**DCIM_JobService** クラスで **SetupJobQueue()** メソッドを呼び出します。
- エクスポートジョブは、**DCIM_JobService** クラスの **DeleteJobQueue()** メソッドを使用してジョブの開始前にキャンセルできます。ジョブの開始後、**POST** 中に **F2** を押し、**Lifecycle Controller** のキャンセルを選択します。または **iDRAC** をリセットします。これにより、リカバリ処理が開始され、システムを以前の既知の状態に戻します。リカバリ処理の時間は 5 分以内です。リカバリが完了したことを確認するには、**WS-Management** コマンドを使用してそのエクスポートジョブをクエリするか、**iDRAC RAC** またはライフサイクルログをチェックします。
- **WS-Management** を使用してネットワーク共有にエクスポートする場合、イメージ名に使用できる文字数は 64 文字のみです。
- エクスポートの実行中、およびエクスポート後に、バックアップファイルが改ざんされることがないようにしてください。

サーバープロファイルをエクスポートするには、次の手順を実行します。

1. バックアップイメージファイルの保存先が **iDRAC vFlash** カードまたはネットワーク共有 (**CIFS** または **NFS**) かに応じて、入力パラメータを構築します。
2. **DCIM_LCService** で **BackupImage()** メソッドを呼び出します。ジョブ ID (例: **JID_001291194119**) が画面上に返されます。
3. ジョブステータス、またはジョブの完了率を取得するには、このジョブ ID で必要な **WS-Management** コマンドを実行します。

様々なスクリプト言語を使ったジョブ ID のクエリのほか、**iDRAC** ログでもジョブステータスの進行状況をチェックします。ジョブステータスが完了と表示されたら、ライフサイクルログですべてのエクスポートエントリをチェックします。ログを表示するには、**DCIM_LCService** クラスで **ExportLCLog()** メソッドを使用してログをエクスポートするか、**Unified Server Configurator-Lifecycle Controller** が有効になっている GUI でログを表示します。

 **メモ:** エクスポートに失敗したときは、ジョブステータスが失敗としてマークされ、失敗の理由を説明するメッセージが表示されます。エラーメッセージ ID および推奨される処置の詳細については、support.dell.com/manuals で『*Dell Lifecycle Controller Remote Services* エラーメッセージおよびトラブルシューティングリスト』を参照してください。

サーバープロファイルのエクスポートの機能とシステム動作

- インポート中、**Lifecycle Controller** は利用できません。
- エクスポート実行中、次の操作のうちのひとつが行われます。
 - **SRVCNF** というラベル名のパーティションが **iDRAC vFlash** カード上に自動的に作成され、バックアップファイルが作成されて、このパーティションに保存されます。**iDRAC vFlash** カード上に **SRVCNF** というラベル名のパーティションがすでに存在する場合は、上書きされます。
 - バックアップファイルが作成され、ネットワーク共有に保存されます。
- エクスポートは、サーバーの設定に応じて完了に最長 45 分かかる場合があります。
- エクスポートは、対象となるすべてのコンポートネットを 1 回の操作でバックアップします。1 つのコンポーネント (例えば、**LOM** ファームウェアと設定のみ) をバックアップすることはできません。
- エクスポートは、ドライバパックや診断パッケージの情報はバックアップしません。
- セキュリティを強化するため、パスフレーズを使用してバックアップイメージファイルをロックします。

- 可変共有タイプの値を指定しない場合は、リモートサービスはその値を0として読み取り、イメージをNFS共有上にバックアップしようとします。
- エクスポート中、Lifecycle Controller 対応デバイス（BIOS、iDRAC、NIC およびストレージコントローラ）の現在のファームウェアバージョンのみがバックアップされます。ロールバックのファームウェアバージョンはバックアップされません。
例：現在インストールされている BIOS ファームウェアのバージョンが 2.1 であり、ロールバックがバージョン 2.0（2.0 は 2.1 をインストールする前の旧バージョン）である場合、エクスポート後、現在インストールされている BIOS ファームウェアのバージョン 2.1 がバックアップされます。

サーバープロファイルのエクスポートのための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 15. 手順番号および参照箇所


手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 1	18.1 — サーバープロファイルのエクスポート
手順 2	18.1.1 — iDRAC vFlash カードへのサーバープロファイルのエクスポート - BackupImage() 18.1.2 — NFS 共有へのサーバープロファイルのエクスポート - BackupImage() 18.1.2 — CIFS 共有へのサーバープロファイルのエクスポート - BackupImage()
手順 3	18.1.4 — エクスポートステータスの監視
プロファイル	
DCIM-LCManagementProfile	
MOF	
DCIM_LCService.mof	

iDRAC vFlash カードまたはネットワーク共有からのサーバープロファイルのインポート

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- iDRAC7 エンタープライズのライセンスがインストールされている。
- サーバーのサービスタグが空白、またはバックアップ時と同じ。
- iDRAC vFlash カード：
 - 取り付けおよび有効化済みで、SRVCNF パーティションが存在する。
 - 少なくとも 384 MB の空き容量が利用可能。
- iDRAC vFlash カードからインポートする場合は、カードが取り付け済みで、SRVCNF パーティションにバックアップイメージがあることを確認してください。このイメージは、インポートしているプラットフォームと同じプラットフォームからのものです。
- ネットワーク共有からインポートする場合は、バックアップファイルが保存されているネットワーク共有が引き続き利用可能であることを確認してください。
- インポートの実行前にマザーボードを交換した場合は、そのマザーボードに最新の iDRAC および BIOS がインストールされていることを確認してください。


ファームウェアおよび設定（サーバーおよびファームウェア）のバックアップをインポートして、バックアップが取得されたものと同じシステムにバックアップを復元します。

 **メモ:** マザーボードの交換を行った場合には、各ハードウェアを以前と同じ場所に再度取り付けるようにしてください。例えば、NIC PCI カードは、バックアップ時に使用されたスロットと同じスロットに取り付けます。

オプションで、現在の仮想ディスクの設定を削除して、バックアップイメージファイルから設定を復元することもできます。

重要なメモ

- バックアップイメージファイルには、ユーザーデータは含まれていません。設定を削除すると、ユーザーデータは削除されます。
- インポート実行中は、ファームウェアアップデート、オペレーティングシステムの展開、およびファームウェアの設定などの操作が実行されないようにしてください。オペレーティングシステムの展開が Lifecycle Controller を使用して行われる場合は、インポートを実行する前に iDRAC をリセット、または Lifecycle Controller をキャンセルする必要があります。
- Lifecycle controller を使用したオペレーティングシステムの展開後、OEMDRV が 18 時間オープン状態になります。オペレーティングシステム展開後に、アップデート、設定、復元などの操作を実行する必要がある場合は、OEMDRV パーティションを削除してください。パーティションを削除するには、iDRAC をリセットするか、Lifecycle Controller をキャンセルします。
- WS-Management のインポートコマンドでは、ScheduledStartTime パラメータを使用しない場合、ジョブ ID は返されますが、スケジュールはされません。ジョブをスケジュールするには、SetupJobQueue() メソッドを呼び出します。
- インポートジョブは、DeleteJobQueue() メソッドを使用してジョブの開始前にキャンセルできます。ジョブの開始後、POST 中に F2 を押し、iDRAC の設定 → Lifecycle Controller の順に移動し、Lifecycle Controller 処置のキャンセルで はい を選択、または iDRAC のリセットをします。これにより、リカバリ処理が開始され、システムを以前の既知の正常稼働状態に戻します。リカバリ処理が 5 分以上かかることはありません。リカバリ処理が完了したことを確認するには、WS-Management コマンドを使用してそのインポートジョブをクエリするか、iDRAC RAC またはライフサイクルログをチェックします。
- マザーボードを交換した場合は、インポートを開始する前に、RestoreImage() メソッドを呼び出せるようにするため、POST 中に Ctrl-E を押し、ネットワークの IP アドレスを設定する必要があります。このメソッドの呼び出し後、バックアップイメージファイルからサービスタグが復元されます。

 **メモ:** インポート操作中、電源装置のファームウェアはアップデートされません。電源装置のファームウェアアップデート中は、PSU への電源が切断されます。ファームウェアは、インポート完了後に適用できます（必要な場合）。

サーバープロファイルをインポートするには、次の手順を実行します。

1. バックアップイメージファイルの場所が iDRAC vFlash カードまたはネットワーク共有（CIFS または NFS）かに応じて、入力パラメータを構築します。
2. RestoreImage() メソッドを呼び出します。ジョブ ID（例えば、JID_001291194119）が画面に返されます。
3. ジョブの完了率のステータスを取得するには、このジョブ ID で必要な ID コマンドを実行します。様々なスクリプト言語を使ったジョブ ID のクエリのほか、iDRAC ログでもジョブステータスの進行状況をチェックします。ジョブステータスが完了と表示されたら、ライフサイクルログですべてのエクスポートエントリをチェックします。ログを表示するには、DCIM_LCService クラスで ExportLCLog() メソッドを使用してログをエクスポートするか、Lifecycle Controller GUI でログを表示します。

 **メモ:** インポートが失敗したときは、ジョブステータスが失敗としてマークされ、失敗が発生した理由を説明するメッセージが表示されます。エラーメッセージ ID および推奨アクションに関する詳細は、support.dell.com/manuals で『Dell Lifecycle Controller Remote Services エラーメッセージおよびトラブルシューティングリスト』を参照してください。

復元後のシナリオ

- 次の操作が実行されます。
 - システムの電源がオンの場合、オフになります。オペレーティングシステムが実行中の場合は、正常なシャットダウンが試行されます。正常にシャットダウンできない場合は、15分後に強制シャットダウンが実行されます。
 - システムが **Lifecycle Controller** の内容をすべて復元します。
 - システムの電源がオンになり、サポートされるデバイス (BIOS、ストレージコントローラ、およびアドイン NIC カード) にファームウェア復元を行うタスクを実行するため、**Lifecycle Controller** で起動します。
 - システムが再起動し、**Lifecycle Controller** が起動して、サポートされるデバイス (BIOS、ストレージコントローラ、およびアドイン NIC カード) のファームウェアの検証、設定の復元を行うためのタスクが実行されます。さらに、実行されたすべてのタスクの最終的な検証が行われます。
 - システム電源がオフになり、**iDRAC** 設定およびファームウェア復元が行われます。完了後、**iDRAC** がリセットされて、システムの電源がオンになります。
 - システム電源がオンになり、復元処理が完了します。復元処理の全エントリについては、**iDRAC** ログまたはライフサイクルログを確認してください。
- インポート後、**Lifecycle Controller GUI** からライフサイクルログをチェック、または **WS-Management** を使用して **LC** ログをネットワーク共有にエクスポートします。ログには、**BIOS**、**ストレージコントローラ**、**LOM**、およびアドイン **NIC** カード (サポートされる場合) の設定およびファームウェアアップデートに関するエントリが収められています。これらの各デバイスに対して複数のエントリが存在する場合は、そのエントリ数は、**Remote Services** が復元を試行した回数と同じになります。

復元後のシナリオの機能とシステム動作

- インポート中は、**Lifecycle Controller** は利用できません。
- インポートにより、バックアップされた内容がすべて復元されます。
- インポートは、サーバーの設定に応じて最長 60 分かかる場合があります。
- インポートは、診断およびドライバパックの情報を復元しません。
- インポートは、デフォルトで現在の仮想ディスク設定を維持します。



メモ: 現在の仮想ディスク設定を削除して、バックアップイメージファイルからの設定を復元したい場合は、値を 0 にした **PreserveVDConfig** パラメータを使用します。これは、バックアップ時の仮想ディスクの内容 (たとえば、オペレーティングシステム) は復元せず、空の仮想ディスクの作成と属性の設定のみが行われます。

- タスクの実行中、追加の再起動が行われます。これは、システムがタスクの再実行を試行するデバイスの設定を行おうとしているためです。どのデバイスが失敗したかに関する情報は、ログを確認してください。
- **iDRAC** ユーザーが **RestoreImage()** メソッドを呼び出すには、サーバーコマンドの実行権限を持っている必要があります。
- コントローラは、仮想ディスクがない場合でもグローバルホットスペアの作成を許可しますが、これらはシステムの再起動後に削除されます。仮想ディスクなしでホットスペアを作成した場合、**SAS** コントローラ上で復元操作が試行され、復元が可能でない場合にはエラーが報告されます。サポートされていない **RAID** レベルが使用されている場合は、**SAS** コントローラ上での復元操作が失敗することがあります。
- サーバードプロファイルのインポート後は、現在インストールされているファームウェアのバージョンがロールバックバージョンとなります。

例 1: 現在インストールされている **BIOS** ファームウェアのバージョンが 2.2 で、エクスポート時にバージョン 2.1 がインストールされました。インポート後、バージョン 2.1 がインストールされているバージョンとなり、2.2 がロールバックバージョンとなります。

例 2 : 現在インストールされている BIOS ファームウェアのバージョンが 2.1 で、エクスポート時にバージョン 2.1 がインストールされました。インポート後、バージョン 2.1 がインストールされているバージョンとなり、2.1 がロールバックバージョンとなります。

サーバープロファイルのインポートのための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 16. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 1	18.2 — サーバープロファイルのインポート
手順 2	18.2.1 — iDRAC vFlash カードからのサーバープロファイルのインポート - RestoreImage() 18.2.2 — NFS 共有からのサーバープロファイルのインポート - RestoreImage() 18.2.3 — CIFS 共有からのサーバープロファイルのインポート - RestoreImage()
手順 3	18.2.4 — インポートステータスの監視
プロファイル	
Dell_LCManagement プロファイル	
MOF	
DCIM_LCService.mof	

vFlash SD カードの管理

vFlash サービスはライセンスを必要とする機能です。vFlash SD カードは、管理下システムの vFlash SD カードスロットに差し込むセキュアなデジタル (SD) カードです。最大 16GB の容量のカードを使用することができます。カードの挿入後、パーティションの作成や管理をするには、vFlash サービスを有効にする必要があります。

vFlash SD カードの詳細については、「[永続ストレージプロファイル](#)」を参照してください。

vFlash SD カードのインベントリの表示

PDCIM_VFlashView クラスで列挙操作を実行して、次のような vFlash SD カードのプロパティをすべて表示します。

- 利用可能なサイズ
- 容量
- ライセンスの有無
- 正常性
- 有効または無効状態
- 初期化状態
- 書き込み保護状態

vFlash SD カードのパーティションの表示

DCIM_OpaqueManagementData クラスで列挙操作を実行して、すべてのパーティションおよびパーティション ID、サイズ、データフォーマットなどのプロパティを表示します。

vFlash SD カードのパーティションの作成と修正

vFlash SD カードのパーティションを作成および修正するには、次の手順を行います。

1. DCIM_OpaqueManagementData クラスで列挙操作を実行して、現在のパーティションのリストを取得します。
2. DCIM_PersistentStorageService クラスで **CreatePartition()** または **CreatePartitionUsingImage()** メソッドを呼び出す前に、入力パラメータを構成します。
3. **CreatePartitionUsingImage()** メソッドを呼び出し、起動可能なイメージを作成します。この操作では、NFS、CIFS、および FTP などのサーバー共有に保存されているイメージから起動可能なパーティションが作成されます。そのかわりに、**CreatePartitionUsingImage()** を呼び出し、ISO イメージから起動可能なパーティションを作成します。
ジョブが正しく作成されると、コード 4096 が返されます。
4. ジョブ制御プロファイルメソッドを使用して、ジョブ ID の出力のステータスを照会します。
5. 手順 1 を繰り返し、メソッドが正しく実行されたことを確認します。
6. **ModifyPartition()** メソッドを呼び出し、パーティションのアクセスタイプを読み取り専用または読み取り / 書きこみに変更します。


iDRAC 設定

この機能を使用して、iDRAC の属性を設定します。

iDRAC の属性の取得と設定

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」の項で説明されている前提条件が満たされていることを確認してください。

iDRAC の属性を取得および設定をするには、次の手順を行います。

 **メモ:** iDRAC 設定後の再起動は不要です。

1. DCIM_iDRACCardAttribute クラスを列挙し、このクラスの現在のインスタンス（すべての iDRAC 設定属性）をすべて識別します。
2. 必要な属性を取得するには、InstanceID プロパティとクラス名を使用して特定のインスタンスを取得します。
3. 属性を設定するため、FQDD プロパティ、AttributeName および AttributeValue を使用して DCIM_iDRACCardService クラスの **ApplyAttributes()** メソッドを呼び出します。
ジョブ ID（例：JID_001291194119）が画面上に返されます。
4. ジョブの完了率のステータスを取得するには、このジョブ ID で必要な ID コマンドを実行します。
5. 変更内容を検証するには、属性の InstanceID プロパティを使用してインスタンスを取得し、属性値を検証して、属性が設定されていることを確認します。

iDRAC 属性の取得と設定のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 17. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 1	19.1 — iDRAC カードインベントリのリスト - Enumeration クラス 19.5 — iDRAC カードインベントリのリスト - Integer クラス 19.7 — iDRAC カードインベントリのリスト - String クラス
手順 2	19.2 — iDRAC カードの列挙インスタンスの取得
手順 3	19.4.1 — iDRAC 値の変更 - ApplyAttributes()(Immediate)
手順 4	19.4.2 — ジョブ完了のポーリング
手順 5	19.4.3 — 属性検証の設定
プロファイル	
DCIM_iDRACCardProfile	
MOF	

- DCIM_iDRACCardEnumeration.mof
- DCIM_iDRACCardInteger.mof
- DCIM_iDRACCardService.mof
- DCIM_iDRACCardString.mof
- DCIM_iDRACCardView.mof

iDRAC 属性

Remote Services を使用して、次の表にリストされる iDRAC の属性を設定することができます。

表 18. LAN 属性

属性	説明	値
VLAN 有効	操作の VLAN モードとパラメータを示します。VLAN を有効にすると、一致する VLAN ID のトラフィックのみが許可されます。無効にすると、VLAN ID と VLAN 優先度は利用できなくなり、それらのパラメータに設定された値はいずれも無視されます。	有効または無効
VLAN ID	VLAN ID の値を設定します。有効値は、IEEE 801.11g 仕様によって定義されています。	1~4094
VLAN 優先度	VLAN ID 優先度の値を設定します。有効値は、IEEE 801.11g 仕様によって定義されています。	0~7
自動ネゴシエーション	自動ネゴシエートがオンのときは、最も近いルーターまたはハブと通信して、iDRAC が二重モードとネットワーク速度の値を自動設定するかどうかを決定します。自動ネゴシエートがオフのときは、二重モードおよびネットワーク速度の値を手動で設定する必要があります。	オンまたはオフ
LAN 速度	ユーザーのネットワーク環境と一致するように、ネットワークの速度を設定します。自動ネゴシエーションがオンの場合、このオプションは使用できません。	10 MB または 100 MB
LAN デュプレックス	ユーザーのネットワーク環境と一致するように、二重モードを設定します。自動ネゴシエーションがオンの場合、このオプションは使用できません。	フルまたはハーフ

表 19. LAN ユーザー設定

属性	説明	値
自動検出	サーバーの自動検出です。	有効または無効
プロビジョニングサーバーアドレス	プロビジョニングサーバーのアドレスを入力します。	IPV4、IPV6、またはホスト名
アカウントアクセス	アカウントアクセスを無効にすると、LAN ユーザー設定の他のフィールドもすべて無効になります。	有効または無効

属性	説明	値
アカウントユーザー名	iDRAC ユーザー名の変更を可能にします。	最大 16 文字の印字可能な ASCII 文字
パスワード	システム管理者が、iDRAC ユーザーのパスワード (暗号化済み) を指定または編集できるようにします。	最大 20 文字
パスワードの確認	確認のために、iDRAC ユーザーのパスワードを再入力します。	最大 20 文字
アカウント権限	IPMI LAN チャンネルでのユーザーの最大権限をユーザーグループに割り当てます。	管理者、オペレータ、ユーザー、アクセスなし
スマートカード認証	iDRAC ログイン用のスマートカード認証です。有効にした場合は、iDRAC にアクセスするためのスマートカードを取り付けます。	有効、無効、または RACADM で有効

表 20. 仮想メディアの接続モード

モード	説明
連結	現在の操作環境において、仮想メディアデバイスを使用することができます。仮想メディアはシステムのプロッピーイメージ、プロッピードライブ、または CD/DVD ドライブを有効化して、それらのプロッピーイメージやドライブがローカルシステムに存在 (連結または接続) しているかのように、管理下システムのコンソールで使用できるようにします。
分離	仮想メディアデバイスにアクセスすることはできません。
自動連結	仮想メディアデバイスは、ユーザーが物理的にメディアを接続するたびに、サーバーに自動でマップされます。

表 21. IPv4 設定

属性	説明	値
IPv4	iDRAC NIC IPv4 プロトコルのサポート。IPv4 を無効にすると、制御が非アクティブ化されます。	有効または無効
RMCP+ 暗号キー	RMCP+ 暗号化キー設定 (空白不可) です。デフォルト設定は、すべてゼロ (0) です。	0~40 の 16 進法の文字列
IP アドレスソース	DHCP サーバーから IPv4 アドレスを取得する iDRAC NIC の機能です。 IP アドレスソース を無効にすると、イーサネット IP アドレス、およびユーザー設定のその他 制御が非アクティブ化されます。	有効または無効
DHCP からの DNS サーバーの取得	iDRAC はダイナミックホスト設定プロトコル (DHCP) サーバーから DNS を取得します。	はいまたはいいえ
DNS サーバー 1 (プライマリ DNS サーバー)	iDRAC はダイナミックホスト設定プロトコル (DHCP) サーバーから DNS サーバー 1 の IP アドレスを取得します。	最大値は 255.255.255.255
DNS サーバー 2 (セカンダリ DNS サーバー)	iDRAC はダイナミックホスト設定プロトコル (DHCP) サーバーから DNS サーバー 2 の IP アドレスを取得します。	最大値は 255.255.255.255

表 22. IP 設定の属性

属性	説明	値
iDRAC 名の登録	iDRAC 名を、ドメイン名システム (DNS) に登録します。	はいまたはいいえ
iDRAC 名	DNS の登録に使用される iDRAC 名の表示または編集を行います。 名前 の文字列には、最大 63 文字までの印刷可能な ASCII 文字を使用できます。 iDRAC 名の登録 が いいえ に設定されているときには、 名前 の文字列を編集できます。	有効または無効
DHCP からのドメイン名	iDRAC は DHCP サーバーからドメイン名を取得します。 いいえ に設定した場合は、手動でドメイン名を入力する必要があります。	はいまたはいいえ
ドメイン名	使用する iDRAC ドメイン名が DHCP から取得されていない場合に、そのドメイン名の表示または編集を行います。 DHCP からのドメイン名 が いいえ に設定されているときには、ドメイン名を指定できます。	有効または無効
ホスト名文字列	iDRAC に関連するホスト名の指定または編集を行います。 ホスト名 の文字列には、最大 62 の印刷可能な ASCII 文字を使用できます。	有効または無効

iDRAC ユーザーと役割の取得と設定

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [Remote Services 使用前の共通前提条件](#)
- [iDRAC 属性の取得と設定](#)

iDRAC のユーザー名、パスワードのセットアップ、およびユーザーへの役割の割り当てを行います。

iDRAC のユーザーおよび役割の取得と設定をするには、次の手順を行います。

1. DCIM_iDRACCardAttribute を列挙して、変更する属性を識別します。
2. 次の属性のプロパティを取得します。
 - FQDD (例: iDRAC.Embedded.1)
 - GroupID (例: Users.3)
 - AttributeName (例: UserName、Privilege、IpmiSerialPrivilege または IpmiLanPrivilege)
3. 属性を設定するため、FQDD プロパティ、AttributeName および AttributeValue を使用して DCIM_iDRACCardService クラスの **ApplyAttributes()** メソッドを呼び出します。
 - Target — FQDD プロパティの値
 - AttributeName[] — GroupID プロパティおよび AttributeName プロパティの値 - GroupID#AttributeName (例: Users.3#UserName または Users.3#Password)
 - AttributeValue[] — 属性に設定する値

ジョブ ID (例: JID_001291194119) が画面上に返されます。

4. 管理者ユーザー名の新しい値を確認します (CurrentValue が新しい値に変更されます)。

iDRAC ユーザーと役割の取得と設定のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 23. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 2	5.2.1 アカウントと機能 (iDRAC 属性を使用)
手順 3	5.3.1 ユーザー名の変更 (iDRAC 属性を使用)
手順 4	5.2.1 アカウントと機能 (iDRAC 属性を使用)
プロファイル	
DCIM_iDRACCardProfile	
MOF	
<ul style="list-style-type: none">• DCIM_iDRACCardEnumeration.mof• DCIM_iDRACCardInteger.mof• DCIM_iDRACCardService.mof• DCIM_iDRACCardString.mof• DCIM_iDRACCardView.mof	

iDRAC の IP アドレス変更の報告

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」の項で説明されている前提条件が満たされていることを確認してください。

IP アドレスの変更を、iDRAC から SCCM に報告します。iDRAC の IP アドレスの変更を示す SOAP (Simple Object Access Protocol) メッセージが送信されます。この機能は、サービスタグに関連付けられているシステムの iDRAC IP アドレスが変更されたことを、プロビジョニングサーバーに通知します。

iDRAC の IP アドレス変更を報告するには、システム管理者アカウントを使用して、IPChangeNotification 属性を設定します。オプションで、プロビジョニングサーバーのアドレスも設定します。

iDRAC の IP アドレスの変更が、手動での介入または DHCP リースの期限切れのいずれかによる場合、iDRAC はプロビジョニングサーバーに対して、そのサーバーのサービスタグおよび iDRAC の新しい IP アドレスを通知します。その後、プロビジョニングサーバーはそのサービスタグを使用してそのサーバーの古いエントリーを検索し、アップデートすることができます。

iDRAC の IP アドレス変更があってもこの通知がなされない場合、プロビジョニングサーバーは、そのサーバーに対する制御を失います。

iDRAC IP アドレス変更の報告の機能とシステム動作

- プロビジョニングサーバーの iDRAC 属性が設定されている場合は、その設定値が使用されます。設定されていない場合は、DHCP ベンダー、DNS SRV レコード、またはデフォルトのプロビジョニングサーバーのホスト名のいずれかのオプションを使用して、プロビジョニングサーバーが特定されます。
- この機能はデフォルトでは無効になっています。
- この機能は、自動検出が無効になっている場合でも、IP アドレスが変更されたことをプロビジョニングサーバーに通知します。
- プロビジョニングサーバーは、IP アドレス変更の通知を要求する必要があります。

- 複数のプロビジョニングサーバーの通知をサポートします。

iDRAC IP アドレス変更の報告のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 24. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
-	19.9.1 — 現在の iDRAC IPChange 状態の取得 19.9.2 — iDRAC IPChange 通知の設定 - SetAttribute()
プロファイル	
DCIM_iDRACCardProfile	
MOF	
<ul style="list-style-type: none"> • DCIM_iDRACCardEnumeration.mof • DCIM_iDRACCardInteger.mof • DCIM_iDRACCardService.mof • DCIM_iDRACCardString.mof • DCIM_iDRACCardView.mof 	

BIOS と起動設定の管理

BIOS および起動設定機能を使用して BIOS のプロパティを設定し、起動ソースや起動順序の変更などの操作を実行します。詳細については、「[BIOS と起動管理プロファイル](#)」を参照してください。


BIOS 属性のインベントリの表示

DCIM_BIOSInteger、DCIM_BIOSEnumeration、DCIM_BIOSPassword、および DCIM_BIOSString クラスのいずれかで列挙操作を実行し、システム内の BIOS 属性の利用可能インスタンスすべてを表示します。

BIOS 属性の設定

属性を設定するには、次の手順を実行してください。

1. `AttributeName` を使用してターゲット属性を識別します。
2. `IsReadOnly` フィールドが、`false` に設定されていることを確認します。
3. `SetAttribute()` または `SetAttributes()` メソッドを呼び出す前に、手順 1 で取得したインスタンス情報をメモし、入力パラメータを準備します。
4. `SetAttribute()` または `SetAttributes()` メソッドを呼び出します。
5. 出力パラメータを確認します。
6. `CreateTargetedConfigJob()` メソッドを呼び出す前に、入力パラメータ（例えば `RebootJobType`、`ScheduledStartTime`、`UntilTime`、ジョブ、など）を準備し、正しい BIOS FQDD を使用します。
7. `CreateTargetedConfigJob()` メソッドを呼び出します。

 **メモ:** 1 つまたは複数の属性を設定するタスクを実行するには、システムを再起動してください。


8. ジョブ制御プロファイルメソッドを使用して、ジョブ ID 出力のステータスをクエリします。
9. 手順 1 を繰り返し、メソッドが正しく実行されたことを確認します。

1 回限りの起動

起動管理メソッドを使用して、BIOS 起動デバイスから 1 回限りの起動を実行します。連結されていない vFlash パーティションから 1 回限りの起動を試行すると、Remote Services によってこのパーティションが自動的に連結され、ジョブ ID が返されます。この ID を使ってジョブをクエリすることができます。

1 回限りの起動を設定するには、次の手順を実行します。

1. DCIM_BootConfigSetting クラスで列挙操作を実行し、`BootSeq` および対応する `InstanceID` を含む `ElementName` フィールドを識別します。
2. DCIM_BootSourceSetting クラスで列挙操作を実行し、起動ソースの `InstanceID` を識別します。各インスタンスの `CurrentEnabledStatus` 属性は、インスタンスが有効であるか無効であるかを識別します。
3. `ChangeBootOrderByInstanceID()` メソッドを呼び出す前に、手順 1 および手順 2 で取得したインスタンス情報をメモし、入力パラメータを準備します。
4. `ChangeBootOrderByInstanceID()` メソッドを呼び出します。
5. 出力パラメータを確認します。

6. **CreateTargetedConfigJob()** メソッドを呼び出す前に、入力パラメータ（例えば **RebootJobType**、**ScheduledStartTime**、**UntilTime**、ジョブ、など）を準備し、正しい **BIOS FQDD** を使用します。
 7. **CreateTargetedConfigJob()** メソッドを呼び出します。
-  **メモ:** 1 つまたは複数の属性を設定するタスクを実行するには、システムを再起動してください。
8. ジョブ制御プロファイルメソッドを使用して、ジョブ ID 出力のステータスをクエリします。
 9. 手順 2 を繰り返し、メソッドが正しく実行されたことを確認します。



BIOS パスワードの設定、変更、削除

サーバー上でリモート操作を正しく実行するために、次の前提条件が満たされていることを確認してください。

- [リモートサービス使用前の共通の前提条件](#)。
- iDRAC の管理者権限。
- 現在の BIOS パスワードのローカルステータス。
- パスワード状態はロックになっている必要があります。

BIOS パスワードの設定、変更、削除を行うには、次の手順を行います。

1. **DCIM_BIOSPassword** クラスで列挙を行い、パスワード状態をチェックします。パスワード状態に可能な値は次のとおりです。
 - 0 - 状態が不明
 - 2 - パスワードが設定済み
 - 3 - パスワードが未設定
 - 4 - パスワードがジャンパの使用により無効化
 2. **DCIM_BIOSService** クラスの **ChangePassword()** メソッドを、次の操作のための関連パラメータで呼び出します。
 - パスワードの設定
 - パスワードの変更
 - パスワードの削除

パスワードを変更または削除するには、新パスワードとともに正しい旧パスワードを使用する必要があります。誤ったパスワードを使用した場合、ターゲットの設定と作成ジョブは機能しますが、ジョブは失敗し、パスワードの変更または削除は行われません。
-  **メモ:** セットアップパスワードを変更または削除するには、旧セットアップパスワードを使用する必要があります。ただし、システムパスワードを変更または削除するには、旧システムパスワードかセットアップパスワードのどちらでも使用することができます。
3. **CreateTargetedConfigJob()** メソッドを呼び出す前に、入力パラメータ（例えば **RebootJobType**、**ScheduledStartTime**、**UntilTime**、ジョブ、など）を準備し、正しい **BIOS FQDD** を使用します。
-  **メモ:** 属性を設定するタスクを実行するには、システムを再起動する必要があります。
4. **CreateTargetedConfigJob()** メソッドを呼び出します。
 5. ジョブステータス、またはジョブの完了率を取得するには、このジョブ ID で必要な ID コマンドを実行します。
 6. BIOS パスワードがシステム上でローカルに設定されていることを確認します。

BIOS パスワードの設定、変更、削除のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 25. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順 1	17.10 — BIOS インベントリのリスト - Password クラス
手順 3 および 手順 4	17.9.2 — ターゲット指定の設定ジョブの作成
手順 5	17.9.3 — BIOS パスワードの設定ステータスの監視
プロファイル	
Dell_BIOSandBootManagement	プロファイル
MOF	
DCIM_BIOSService.mof	

その他の使用事例のシナリオ

本項にはその他いくつかの使用事例が記載されています。

Remote Services ステータスの取得

サーバー上でリモート操作を正しく実行するために、「[Remote Services 使用前の共通前提条件](#)」の項で説明されている前提条件が満たされていることを確認してください。

Remote Services の操作（たとえば、NIC の管理、RAID 構成の管理、インベントリ、など）を行う前に、Remote Services が実行中で、最新にアップデートされ、データの送信が可能であることを確認してください。Remote Services ステータスの取得機能を使って、次の操作を行います。

- 準備完了、準備中、再ロード中 など、Remote Services の現在のステータスを取得する。
- Remote Services が利用可能であることを調べるため、継続してポーリングを行う。

Remote Services ステータスを取得するには、次を行います。

1. **GetRSStatus()** メソッドを呼び出します。
Message、MessageID、および ReturnValue とともに、ステータスが返されます。
2. 準備完了ステータスが返されるまで、間隔をあけてメソッドの実行を継続します。
準備完了ステータスは、Lifecycle Controller の操作準備が整っていることを示します。

Remote Services ステータスの取得のための参照資料


 **メモ:** 次の表で参照されている項に記載される例は、一般的な例のみです。

表 26. 手順番号および参照箇所

手順番号	Lifecycle Controller Web サービスインタフェースガイド (Windows および Linux) 内の箇所
手順1	20.1 — リモートサービスステータスの取得
プロファイル	
DCIM-LCManagementProfile	
MOF	
DCIM_LCService.mof	

Remote Services プロファイル

本項には、各プロファイルと、そのクラスおよびメソッドに関する高レベルの情報が記載されています。プロファイルと関連する MOF の詳細については、delltechcenter.com/page/DCIM.Library を参照してください。WinRM および WS-Management コマンドラインの呼び出し例は、次を参照してください。

- delltechcenter.com/page/Lifecycle+Controller
- *Lifecycle Controller Web* サービスインタフェースガイド – Windows および Linux

オペレーティングシステム展開プロファイル

次の表では、オペレーティングシステム展開プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 27. オペレーティングシステム展開プロファイル

クラス名	操作	メソッド
DCIM_OSDeploymentService	Get Enumerate Invoke	「オペレーティングシステム展開メソッド」 参照してください。
DCIM_OSConcreteJob	Get Enumerate	-

オペレーティングシステム展開メソッド

- **GetDriverPackInfo()** メソッドは、Dell ライフサイクルコントローラで利用できる、内蔵デバイスドライバを使用してサーバーにインストールできる、オペレーティングシステムのリストを返します。
- **UnpackAndAttach()** メソッドは、指定された時間にサーバーにローカルで接続された USB デバイスへ、選択されたオペレーティングシステムのためのドライバを抽出します。
- **DetachDrivers()** メソッドは、ホストサーバーからのドライバを含む USB デバイスの接続を外します。
- **UnpackAndShare()** メソッドは、選択されたオペレーティングシステム用のドライバを抽出し、それらを指定されたネットワーク共有へコピーします。
- **BootToNetworkISO()** メソッドは、CIFS または NFS ネットワーク共有に存在する ISO イメージからシステムを起動させるために使用します。
- **DetachISOImage()** メソッドは、ホストサーバーから ISO イメージを分離します。
- **BootToPXE()** メソッドは、Preboot Execution Environment (PXE) メカニズムを利用してサーバーを起動するために使用します。
- **DownloadISOToVFlash()** メソッドは、pre-OS ISO イメージを vFlash SD カードにダウンロードするために使用します。
- **BootToISOFromVFlash()** メソッドは、すでにダウンロードされている vFlash pre-OS イメージから起動するために使用します。
- **DetachISOFromVFlash()** メソッドは、ホストサーバーから ISO イメージを分離します。
- **DeleteISOFromVFlash()** メソッドは、vFlash SD カードから ISO イメージを削除します。

Lifecycle Controller 管理プロファイル

次の表では、Lifecycle Controller 管理プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 28. Lifecycle Controller 管理プロファイル

クラス名	操作	メソッド
DCIM_LCService	Get Enumerate Invoke	SetAttribute() SetAttributes() GetRemoteServicesAPIStatus() 「自動検出メソッド」 、「 ライフサイクルログメソッド 」、 および「 ハードウェアインベントリメソッド 」も参照し てください。
DCIM_LCString	Get Enumerate	-
DCIM_LCEnumeration	Get Enumerate	-

LC サービスメソッド

次のメソッドは、自動検出、部品交換、および IO の ID に関連した属性の設定に使用されます。

- **SetAttribute()** メソッドは、単一の属性を設定するために使用します。
- **SetAttributes()** メソッドは、複数の属性を設定するために使用します。
- **CreateConfigJob()** メソッドは、**SetAttribute()** および **SetAttributes()** メソッドで設定された保留中の値を適用するために使用します。
- **GetRemoteServicesAPIStatus()** メソッドは Lifecycle Controller Remote Services が Web サービス要求の受け入れ準備完了かどうかを知るために使用します。

自動検出メソッド

- **ReInitiateDHS()** メソッドは、プロビジョニングサーバー検出およびハンドシェイクを再開するために使用します。
- **ClearProvisioningServer()** メソッドは、プロビジョニングサーバー値をクリアするために使用します。
- **DownloadServerPublicKey()** メソッドは、Lifecycle Controller (LC) にサーバーパブリックキーをダウンロードするために使用します。
- **DownloadClientCerts()** メソッドは、クライアントのプライベート証明書、パスワード、およびルート証明書を Lifecycle Controller にダウンロードするために使用します。
- **DeleteAutoDiscoveryClientCerts()** メソッドは、以前にダウンロードされた自動検出クライアント証明書およびプライベートキーを削除するために使用します。
- **SetCertificateAndPrivateKey()** メソッドは、PKCS#12 ファイルの内容を利用して iDRAC 証明書およびプライベートキーのペアをアップデートするために使用します。
- **SetPublicCertificate()** メソッドは、iDRAC におけるパブリック SSL 証明書をアップデートするために使用します。
- **DeleteAutoDiscoveryServerPublicKey()** メソッドは、以前にダウンロードされた自動検出サーバーパブリックキーを削除するために使用します。

エクスポートとインポートメソッド

- **BackupImage()** メソッドは、vFlash SD カードまたはネットワーク共有上にあるファームウェア、ファームウェアインベントリ、およびサーバーコンポーネント設定をバックアップまたはエクスポートします。

- **RestoreImage()** メソッドは、サーバープロファイルをインポートし、サーバーを以前の設定に復元します。
- **GetRSStatus()** メソッドは、Remote Services のステータスを取得するために使用されます。
- **GetRemoteServicesAPIStatus()** メソッドは、ホストサーバーステータス、Lifecycle Controller ステータス、およびプロビジョニングのタスクがその時点で実行可能かどうかを示す全体的なステータスを取得します。

ライフサイクルログメソッド

- **LCWipe()** メソッドは、システムを破棄する前に、ライフサイクルコントローラから設定を全てクリアするために使用します。
- **ExportLifecycleLog()** メソッドは、ライフサイクルコントローラからリモート共有のファイルへログをエクスポートするために使用します。
- **InsertCommentInLCLog()** メソッドは、ライフサイクルコントローラログへ追加のユーザーコメントを挿入するために使用します。

ハードウェアインベントリメソッド

- **ExportHWInventory()** メソッドは、Lifecycle Controller からリモート共有のファイルへハードウェアインベントリをエクスポートするために使用します。
- **ExportFactoryConfiguration()** メソッドは、ライフサイクルコントローラからリモート共有のファイルへ工場出荷時の設定をエクスポートするために使用します。

シンプル NIC プロファイル

次の表では、シンプル NIC プロファイルのクラス、機能、操作、およびメソッドをリストします。


表 29. シンプル NIC プロファイル

クラス名	機能	操作	メソッド
DCIM_NICService	これはセントラルクラスで、NIC、FCOE、およびiSCSI 属性を変更するために呼び出されます。	Get Enumerate Invoke	「 シンプル NIC メソッド 」を参照してください。
DCIM_NICView	このクラスを使用して、システムの LOM、アドイン NIC、および CNA のインスタンス ID とその他プロパティを表示します。	Get Enumerate	-
DCIM_NICAttribute — このクラスは、次の BIOS サブクラスの出力を表示します。			
• DCIM_NICEnumeration	このサブクラスを使用して、NIC 列挙インスタンスのプロパティを表示します。	Get Enumerate	SetAttribute() SetAttributes()
• DCIM_NICInteger	このサブクラスを使用して NIC 整数インスタンスのプロパティを表示します。	Get Enumerate	SetAttribute() SetAttributes()
• DCIM_NICString	このサブクラスを使用して NIC ストリングインスタンスのプロパティを表示します。	Get Enumerate	SetAttribute() SetAttributes()

シンプル NIC メソッド

これらのメソッドは、システムのマザーボードに搭載の LAN、アドイン NIC、および CNA に、NIC、FCOE、および iSCSI 属性を適用するために使用します。各メソッドには、それぞれの入力および出力パラメータがあります。メソッドには特定のリターンコード値があります。NIC サービスクラスには 4 つの異なるメソッドがあります。

- **SetAttribute()** メソッドは、NIC 属性の値を設定または変更するために使用します。
- **SetAttributes()** メソッドは、属性群の値の設定に使用されます。
- **CreateTargetedConfigJob()** メソッドは、**SetAttribute** および **SetAttributes** メソッドで作成された保留中の値を適用するために使用します。このメソッドが正常に実行されると、保留中の属性値を適用するためのジョブが作成されます。

 **メモ:** 最初の **CreateTargetedConfigJob()** メソッド呼び出しの後の **CreateTargetedConfigJob()** メソッド呼び出しは、最初のジョブが完了するまではエラーが発生します。**CreateTargetedConfigJob()** メソッドを複数回呼び出すと、古いリクエストは上書き、または失われます。

- **DeletePendingConfiguration()** メソッドは、設定ジョブが **CreateTargetedConfigJob()** で作成される前に行われた、保留中の設定 (**SetAttribute** および **SetAttributes** メソッドで作成) 変更をキャンセルします。

BIOS と起動管理プロファイル

次の表では、BIOS および起動管理プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 30. BIOS と起動管理プロファイル

クラス名	機能	操作	メソッド
BIOS 管理			
DCIM_BIOSService	このセントラルクラスを使用して BIOS 属性を変更します。	Get Enumerate Invoke	「BIOS および起動管理メソッド」 を参照してください。
DCIM_BIOSEnumeration	このサブクラスを使用して BIOS 列挙インスタンスのプロパティを表示します。	Get Enumerate	SetAttribute() SetAttributes()
DCIM_BIOSInteger	このサブクラスを使用して BIOS スtring インスタンスのプロパティを表示します。	Get Enumerate	SetAttribute() SetAttributes()
DCIM_BIOSString	このサブクラスを使用して、BIOS 整数インスタンスのプロパティを表示します。	Get Enumerate	SetAttribute() SetAttributes()
DCIM_BIOSPassword	このサブクラスを使用して、BIOS パスワードを管理します。	Get Enumerate	ChangePassword()
DCIM_BootConfigSetting	このクラスには次の起動リストインスタンスがあります。 <ul style="list-style-type: none">• IPL• BCV• UEFI• vFlash	Get Enumerate Invoke	ChangeBootSourceState() ChangeBootOrderByInstanc eID()

クラス名	機能	操作	メソッド
------	----	----	------

- OneTime


起動管理

DCIM_BootSourceSetting	このクラスを使用して、起動ソースおよび関連デバイスの起動順序を変更します。	Get Enumerate	-
------------------------	---------------------------------------	---------------	---

BIOS と起動管理メソッド

BIOS および起動管理メソッドはシステムの属性を適用し、起動設定を変更するために使用されます。各メソッドには、それぞれの入力および出力パラメーター式があります。メソッドには特定の戻り値があります。BIOS および起動管理では次のメソッドが使用されます。

- **SetAttribute()** メソッドは、BIOS 属性の値を設定または変更するために使用します。
- **SetAttributes()** メソッドは、属性群の値の設定または変更で使用されます。
- **ChangeBootSourceState()** メソッドは、起動ソースの **EnabledState** を有効から無効、または無効から有効へ変更するために使用します。
- **ChangeBootOrderByInstanceID()** メソッドは、起動リストインスタンス (IPL、BCV、UEFI) からの起動ソースの起動順序を変更するために使用されます。このメソッドでは起動ソースインスタンスをひとつのリストからのみ受け入れるため、複数のインスタンスの起動順序を変更するには、このメソッドを異なる起動リストのインスタンスで複数回呼び出します。
- **CreateTargetedConfigJob()** メソッドは、**SetAttribute()** および **SetAttributes()** メソッドによって作成された、保留中の値を適用するために使用されます。このメソッドが正常に実行されると、保留中の属性値を適用するためのジョブが作成されます。このメソッドは、起動順序、ソース状態、および一回限りの起動デバイスの設定にも使用されます。

 **メモ:** 最初の **CreateTargetedConfigJob()** メソッド呼び出しの後の **CreateTargetedConfigJob()** メソッド呼び出しは、最初のジョブが完了するまではエラーが発生します。ただし、現在のジョブを削除し、**CreateTargetedConfigJob()** を使って新しいジョブを作成することができます。

- **DeletePendingConfiguration()** メソッドは、設定ジョブが **CreateTargetedConfigJob()** で作成される前に行われた、保留中の設定 (**SetAttribute** および **SetAttributes** メソッドで作成) 変更をキャンセルします。
- **ChangePassword()** メソッドは、BIOS パスワードを変更します。

永続ストレージプロファイル

次の表では、永続ストレージプロファイルのクラス、機能、操作、およびメソッドをリストします。

表 31. 永続ストレージプロファイル

クラス名	機能	操作	メソッド
DCIM_PersistentStorageService	このセントラルクラスを使って外部メソッドを定義します。	Get Enumerate Invoke	「vFlash SD カードメソッド」 を参照してください。
DCIM_VFlashView	このクラスを使って、システムに接続されたすべての vFlash SD カードの、異なるインスタンス ID および関連プロパティを表示します。	Get Enumerate	-
DCIM_OpaqueManagementData	このサブクラスを使って、特定の vFlash SD カード上	Get Enumerate	-

クラス名	機能	操作	メソッド
	で利用可能なパーティションを表示します。		

vFlash SD カードメソッド

- **InitializeMedia()** メソッドは、vFlash SD カードをフォーマットするために使用します。
- **VFlashStateChange()** メソッドは、vFlash SD カードを有効化または無効化するために使用します。
- **CreatePartition()** メソッドは、vFlash SD カードに新規パーティションを作成するために使用します。
- **CreatePartitionUsingImage()** メソッドは、イメージファイル (.img または .iso フォーマットがあります) を利用して新規パーティションを作成するために使用します。
- **DeletePartition()** メソッドは、vFlash SD カードのパーティションの削除に使用します。
- **FormatPartition()** メソッドは、選択した vFlash SD カードのパーティションのフォーマットに使用します。
- **ModifyPartition()** メソッドは、vFlash のパーティションを変更するために使用します。これは、パーティションの種類 (フロッピー、ハードディスク、または CD) に応じて異なります。
- **AttachPartition()** メソッドは、1つ、または複数のパーティションを仮想 USB マスストレージデバイスとして接続するために使用します。
- **DetachPartition()** メソッドは、仮想 USB マスストレージデバイスとして使用されている1つ、または複数のパーティションの接続を外すために使用します。
- **ExportDataFromPartition()** メソッドは、vFlash SD カードのパーティションの内容を、.img または .iso フォーマットのイメージファイルとしてローカル、またはリモートのロケーションにコピー、またはエクスポートするために使用します。

RAID プロファイル

次の表では、RAID プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 32. RAID プロファイル


クラス名	機能	操作	メソッド
DCIM_RAIDService	これはセントラルクラスで、外部メソッドを定義します。	Get Enumerate Invoke	「 RAID メソッド 」を参照してください。
DCIM_ControllerView	このクラスを使用して、システムに接続されたコントローラの異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-
DCIM_PhysicalDiskView	このクラスを使用して、システムに接続された物理ディスクの異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-
DCIM_VirtualDiskView	このクラスを使用して、作成された仮想ディスクの異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-
DCIM_EnclosureView	このクラスを使用して、システムに接続されたエンクロージャの異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-

クラス名	機能	操作	メソッド
DCIM_ControllerBatteryView	このサブクラスを使用して、コントローラバッテリーのプロパティを表示します。	Get Enumerate	-
DCIM_EnclosureEMMView	このクラスを使用して、EMM ファームウェアがあるエンクロージャの異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-
DCIM_EnclosurePSUView	このクラスを使用して、エンクロージャの PSU の異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-
DCIM_EnclosureFanSensor	このクラスを使用して、エンクロージャファンの異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-
DCIM_EnclosureTemperatureSensor	このクラスを使用して、エンクロージャファンの異なるインスタンス ID と関連プロパティを表示します。	Get Enumerate	-


RAID メソッド

RAID メソッドは、異なる RAID コンポーネントに属性を適用するために使用します。各メソッドには、それぞれの入力および出力パラメータがあります。メソッドには特定の戻りコード値があります。RAID サービスクラスの異なるメソッドは次の通りです。

- **AssignSpare()** メソッドは、物理ディスクを仮想ディスクの専用ホットスペア、またはグローバルホットスペアとして割り当てるために使用します。
- **ResetConfig()** メソッドは、仮想ディスクのすべてを削除し、ホットスペア物理ディスクすべての割り当てを解除します。既存の仮想ディスクにあるデータはすべて失われます。


 **メモ:** インポートされていない外部物理ディスクにある仮想ディスクは削除されません。

- **ClearForeignConfig()** メソッドは、外部物理ディスクをローカル設定に追加するために外部物理ディスクの準備を行うために使用します。

 **メモ:** 外部物理ディスク上のデータはすべて失われます。

- **DeleteVirtualDisk()** メソッドは、ターゲットのコントローラから単一の仮想ディスクを削除するために使用します。このメソッドが正常に実行されると、この仮想ディスクに削除のマークが付きます。
- **CreateVirtualDisk()** メソッドは、ターゲットのコントローラに単一の仮想ディスクを作成するために使用します。このメソッドが正常に実行されると、保留中であるが未作成の仮想ディスクが発生します。
- **GetDHSDisks()** メソッドは、識別された仮想ディスクの専用ホットスペアにするドライブの候補を検索するために使用します。
- **GetRAIDLevels()** メソッドは、仮想ディスクを作成するための RAID レベルの候補を検索するために使用します。物理ディスクのリストが提供されていない場合、このメソッドは接続されたすべてのディスクで動作します。
- **GetAvailableDisks()** メソッドは、仮想ディスクを作成するためのドライブの候補を検索するために使用します。
- **CheckVDValues()** メソッドは、指定された RAID レベルとディスク数に応じた、仮想ディスクのサイズとデフォルト設定を検索するために使用します。

- **SetControllerKey()** メソッドは、ドライブの暗号化をサポートするコントローラにキーを設定します。
- **LockVirtualDisk()** メソッドは、識別された仮想ディスクを暗号化します。この仮想ディスクは、暗号化が有効化されている状態の暗号化対応物理ディスクに存在する必要があります。
- **CreateTargetedConfigJob()** メソッドは、他のメソッドで作成された保留中の値を適用するために使用します。このメソッドが正常に実行されると、保留中の属性値を適用するためのジョブが作成されます。

 **メモ:** 最初の **CreateTargetedConfigJob()** メソッド呼び出しの後の **CreateTargetedConfigJob()** メソッド呼び出しは、最初のジョブが完了するまではエラーが発生する原因となります。

- **DeletePendingConfiguration()** メソッドは、設定ジョブが **CreateTargetedConfigJob()** によって作成される前に行われた保留中の設定（他のメソッドにより作成）の変更をキャンセルします。
- **RemoveControllerKey()** メソッドは、コントローラの暗号化キーを消去します。暗号化されたすべての仮想ディスクは、そのデータと共に消去されます。
- **ReKey()** メソッドは、ターゲットコントローラのローカルキー管理の暗号化キーを変更するために使用します。
- **EnableControllerEncryption()** メソッドは、コントローラにローカルキー暗号化（LKM）を適用します。
- **SetAttribute()** メソッドは、RAID 属性の値を設定または変更するために使用します。
- **SetAttributes()** メソッドは、属性群の設定または変更で使用されます。
- **CreateVirtualDisk()** メソッドは、ターゲットコントローラに仮想ディスクを作成するために使用します。このメソッドは次の操作実行するためにも使用できます。
 - スライスされた仮想ディスクの作成。これらの仮想ディスクは、**CreateVirtualDisk()** のサイズに入力されたパラメータ値が、物理ディスクセットの合計サイズよりも少ない場合に作成されます。追加のスライスされた仮想ディスクは、最初の仮想ディスクの作成に使用されたものと同じ物理ディスクセットと RAID レベルを使用して作成することができます。
 - ターゲットのコントローラでの **CacheCade** 仮想ディスクの作成。このメソッドは、内部的に RAID-0 仮想ディスクを作成します。作成方法はスライスされた仮想ディスクの作成と同じです。
- **UnassignSpares()** メソッドは、物理ディスクを仮想ディスクの専用ホットスペア、またはグローバルホットスペアとしての割り当てから解除するために使用します。

ハードウェアインベントリプロファイル

次の表では、管理下ノード上の異なるハードウェアにおけるクラス、機能、操作、およびメソッドを一覧表示します。

表 33. ハードウェアインベントリプロファイル

クラス名	機能	操作	メソッド
CPU プロファイル			
DCIM_CPUView	このクラスを使用して、システム内のすべての CPU、およびそれに関連する利用可能なキャッシュのインスタンス情報を取得します。	Get Enumerate	-
ファンプロファイル			
DCIM_FanView	このクラスを使用して、システム内で利用可能なファンすべてのインスタンス情報を取得します。	Get Enumerate	-
iDRAC プロファイル			
DCIM_IDRACCardView	このクラスを使用して、システム内で利用可能な iDRAC カードすべてのイ	Get Enumerate	-

クラス名	機能	操作	メソッド
	インスタンス情報を取得します。		
メモリプロファイル			
DCIM_MemoryView	このクラスを使用して、システム内で利用可能なメモリモジュールすべてのインスタンス情報を取得します。	Get Enumerate	-
PCI プロファイル			
DCIM_PCIDeviceView	このクラスを使用して、システム内で利用可能な PCI デバイスすべてのインスタンス情報を取得します。	Get Enumerate	-
ビデオプロファイル			
DCIM_VideoView	このクラスを使用して、システム内で利用可能なビデオコントローラすべてのインスタンス情報を取得します。	Get Enumerate	-
電源装置プロファイル			
DCIM_PowerSupplyView	このクラスを使用して、システム内で利用可能な電源装置すべてのインスタンス情報を取得します。	Get Enumerate	-
システムビュープロファイル			
DCIM_SystemView	このクラスを使用してシステムの製造元、モデル、サービスタグ、総メモリ数、BIOS バージョン、システム ID、アセットタグ、電源状態などのシステムに関する一般詳細を取得します。	Get Enumerate	-

ジョブ制御プロファイル

次の表では、ジョブ制御プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 34. ジョブ制御プロファイル

クラス名	操作	メソッド
DCIM_JobControlService	Get Enumerate	「 ジョブ制御メソッド 」を参照してください。
DCIM_LifecycleJob	Get Enumerate	-

ジョブ制御メソッド

このメソッドは、ジョブキューの設定、およびジョブキューからのジョブの削除のために使用します。

- **SetupJobQueue()** メソッドは、キュー内で特定の順序で実行される 1 つ、または複数のジョブを含むジョブキューを作成するために使用します。
- **DeleteJobQueue()** メソッドは、ジョブキューからジョブを削除するために使用します。

電源装置プロファイル

次の表では、電源装置プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 35. 電源装置プロファイル

クラス名	操作	メソッド
DCIM_PowerSupplyView	Get Enumerate	-
DCIM_PowerSupply	Get Enumerate	-
DCIM_PowerRedundancySet	Get Enumerate	-

電源状態管理プロファイル

次の表では、電源状態管理プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 36. 電源状態管理プロファイル

クラス名	操作	メソッド
DCIM_CSPowerManagementService	Get Enumerate Invoke	「電源状態管理プロファイルメソッド」 を参照してください。
DCIM_CSPowerManagementCapabilities	Get Enumerate	-
DCIM_CSAssociatedPowerManagementService	Get Enumerate	-

電源状態管理プロファイルメソッド

このメソッドは電源状態の変化ステータスを取得するために使用します。

- 保留中の電源状態の変更を取得するには **RequestPowerStateChange()** メソッドを呼び出します。

レコードログプロファイル

次の表では、レコードログプロファイルのクラス、機能、操作、およびメソッドをリストします。

表 37. レコードログプロファイル

クラス名	操作	メソッド
DCIM_LCRecordLog	Get Enumerate Invoke	「レコードログプロファイルメソッド」 を参照してください。
DCIM_LCRecordLogCapabilities	Get Enumerate	-
DCIM_LCLogEntry	Get Enumerate Invoke	-
DCIM_SELRecordLog	Get Enumerate Invoke	-
DCIM_SELRecordLogCapabilities	Get Enumerate	-
DCIM_SELLogEntry	Get Enumerate Invoke	-

レコードログプロファイルメソッド

このメソッドは、システムで生成されるログを管理するために使用します。

- **ClearLog()** メソッドは、SSL レコードログ内のすべてのエントリを削除するために使用します。0 の戻りコード値は、ログエントリの消去が正しく開始されたことを意味します。
- **GetConfigResults()** メソッドは、特定のログエントリに関連する設定結果を取得する機能を提供します。

役割ベース認証プロファイル

次の表では、役割ベース認証プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 38. 役割ベース認証プロファイル

クラス名	操作	メソッド
DCIM_LocalRolePrivilege	Get Enumerate Set	-
DCIM_CLPPrivilege	Get Enumerate	-
DCIM_Role	Get Enumerate	-
DCIM_IPMIRole	Get Enumerate	-
DCIM_IPMISOLRole	Get Enumerate	-
DCIM_CLPRole	Get Enumerate	-
DCIM_LocalRoleBasedAuthorizationService	Get Enumerate	-
DCIM_IPMIRoleBasedAuthorizationService	Get Enumerate	「役割ベース認証プロファイルメソッド」 を参照してください。
DCIM_CLPRoleBasedAuthorizationService	Get Enumerate	「役割ベース認証プロファイルメソッド」 を参照してください。
DCIM_LocalRoleBasedManagementCapabilities	Get Enumerate	-
DCIM_IPMIRoleBasedManagementCapabilities	Get Enumerate	-
DCIM_CLPRoleBasedManagementCapabilities	Get Enumerate	-

役割ベース認証プロファイルメソッド

このメソッドはシステムへの役割ベースのアクセスを管理するために使用します。

- **DCIM_IPMIRoleBasedAuthorizationService** クラスの **AssignRoles()** メソッドは、**DCIM_IPMIIdentity** のインスタンスによって表現されるセキュリティプリンシパルを、**DCIM_IPMIRole** のインスタンスによって表現される 0、または複数の役割に割り当てるために使用されます。
- **DCIM_CLPRoleBasedAuthorizationService** クラスの **AssignRoles()** メソッドは、**DCIM_CLPIIdentity** のインスタンスによって表現されるセキュリティプリンシパルを、**DCIM_CLPRole** のインスタンスによって表現される 0、または複数の役割に割り当てるために使用されます。

センサープロファイル

次の表では、センサープロファイルのクラス、機能、操作、およびメソッドをリストします。

表 39. センサープロファイル

クラス名	操作	メソッド
DCIM_PSNumericSensor	Get Enumerate Set	-
DCIM_NumericSensor	Get Enumerate Set	-
DCIM_Sensor	Get Enumerate	-

サービスプロセッサプロファイル

次の表では、サービスプロセッサプロファイルのクラス、機能、操作、およびメソッドをリストします。

表 40. サービスプロセッサプロファイル

クラス名	操作	メソッド
DCIM_SPComputerSystem	Get Enumerate Invoke	「サービスプロセッサプロファイルメソッド」 を参照してください。
DCIM_TimeService	Get Enumerate Invoke	「サービスプロセッサプロファイルメソッド」 を参照してください。

サービスプロセッサプロファイルメソッド

このメソッドを使用してサービスプロセッサを管理します。

- **RequestStateChange()** メソッドは、RequestedState パラメータで指定された値に iDRAC 状態をリセットするために使用します。
- **ManageTime()** メソッドは、サービスプロセッサ時間をクエリするために使用します。

イベントフィルタプロファイル

次の表では、イベントフィルタプロファイルのクラス、機能、操作、およびメソッドをリストします。

表 41. イベントフィルタプロファイル

クラス名	操作	メソッド
DCIM_EFConfigurationService	Get Enumerate Invoke Set (カテゴリ別)	「イベントフィルタプロファイルメソッド」 を参照してください。
DCIM_EventFilter	Get Enumerate	-

イベントフィルタプロファイルメソッド

このメソッドは、イベントフィルタの管理に使用されます。

- **SetEventFilterByCategory()** メソッドは、特定のカテゴリ、サブカテゴリ、および重要度に属するすべてのイベントフィルタへの処理と通知を設定するために使用します。
- **SetEventFilterByInstanceIDs()** メソッドは、特定の InstanceID セットに属するすべてのイベントフィルタへの処理と通知を設定するために使用します。

ライセンス管理プロファイル

次の表では、ライセンス管理プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 42. ライセンス管理プロファイル

クラス名	操作	メソッド
DCIM_LicenseManagementService	Get Enumerate Invoke	「 ライセンス管理プロファイルメソッド 」を参照してください。
DCIM_LicensableDevice	Get Enumerate	-
DCIM_License	Get Enumerate	-

ライセンス管理プロファイルメソッド

このメソッドは、ライセンスの管理に使用します。

- **ImportLicense()** メソッドは、ライセンスファイルをライセンスマネージャにインポートするために使用します。
- **ImportLicenseFromNetworkShare()** メソッドは、共有の場所にあるライセンスをインポートするために使用します。
- **DeleteLicense()** メソッドは、割り当てられているライセンスを削除するために使用します。
- **ExportLicense()** メソッドは、iDRAC からライセンスファイルをエクスポートするために使用します。
- **ExportLicenseByDevice()** メソッドは、iDRAC からライセンスファイルをエクスポートするために使用します。
- **ExportLicenseToNetworkShare()** メソッドは、iDRAC からライセンスファイルをエクスポートするために使用します。
- **ExportLicenseByDeviceToNetworkShare()** メソッドは、ライセンスをデバイスから外部の場所にエクスポートするために使用します。
- **ReplaceLicense()** メソッドは、ライセンスマネージャのライセンスファイルを交換するために使用します。

iDRAC カードプロファイル

次の表では、iDRAC カードプロファイルのクラス、機能、操作、およびメソッドをリストします。

表 43. iDRAC カードプロファイル

クラス名	操作	メソッド
DCIM_IDRACCardView	Get Enumerate	-
DCIM_IDRACCardEnumeration	Get Enumerate	-
DCIM_IDRACCardString	Get Enumerate	-
DCIM_IDRACCardInteger	Get Enumerate	-
DCIM_IDRACCardService	Get Enumerate Invoke	「 iDRAC カードプロファイルメソッド 」を参照してください。

iDRAC カードプロファイルメソッド

本メソッドは iDRAC を管理するために使用します。

- **SetAttribute()** メソッドは、iDRAC 属性値の設定または変更のために使用します。

- **SetAttributes()** メソッドは、iDRAC 属性群の値の設定または変更のために使用します。
- **CreateTargetedConfigJob()** メソッドは、**SetAttribute** および **SetAttributes** メソッドで作成された保留中の値を適用するために使用します。
- **DeletePendingConfiguration()** メソッドは、**SetAttribute** および **SetAttributes** メソッドで作成された保留中の値をキャンセルするために使用します。
- **ApplyAttributes()** メソッドは、iDRAC 属性値の設定または変更のために使用します。
- **SendTestEmailAlert()** メソッドは電子メールテスト通知を送信するために使用します。

ベースサーバーと物理的資産プロファイル

次の表では、ベースサーバーおよび物理的資産プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 44. ベースサーバーと物理的資産プロファイル

クラス名	操作	メソッド
DCIM_ComputerSystem	Get Enumerate Invoke	「ベースサーバーと物理的資産プロファイルメソッド」 を参照してください。
DCIM_ComputerSystemPackage	Get Enumerate	-
DCIM_CSEnabledLogicalElementCapabilities	Get Enumerate	-
DCIM_Chassis	Get Enumerate	-

ベースサーバーと物理的資産プロファイルメソッド

このメソッドは、基本的なサーバー管理タスクを実行するために使用されます。

- **RequestStateChange()** メソッドは、コンポーネントの状態を、有効、無効、リセットのいずれかの値に変更するために使用します。

システム情報プロファイル

次の表では、システム情報プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 45. システム情報プロファイル

クラス名	操作	メソッド
DCIM_SystemEnumeration	Get Enumerate	-
DCIM_SystemString	Get Enumerate	-
DCIM_SystemInteger	Get Enumerate	-
DCIM_SystemManagementService	Get Enumerate Invoke	「システム情報メソッド」 を参照してください。

システム情報メソッド

このメソッドは、基本的なシステム情報を取得するために使用します。

- **SetAttribute()** メソッドは、システム属性値の設定または変更で使用されます。
- **SetAttributes()** メソッドは、属性群の値の設定または変更で使用されます。

- **CreateTargetedConfigJob()** メソッドは、**SetAttribute** および **SetAttributes** メソッドで作成された保留中の値を適用するために使用します。
- **DeletePendingConfiguration()** メソッドは、**SetAttribute** および **SetAttributes** メソッドで作成された保留中の値をキャンセルするために使用します。
- **ShowErrorsOnLCD()** メソッドは、LCD のエラーを非表示または表示するために使用します。
- **IdentifyChassis()** メソッドは、シャーシ上の LED をオンやオフにして、シャーシを識別するために使用します。

シンプル ID 管理プロファイル

次の表では、シンプル ID 管理プロファイルのクラス、機能、操作、およびメソッドをリストします。

表 46. シンプル ID 管理プロファイル

クラス名	操作	メソッド
DCIM_Account	Get Enumerate Set Invoke	「 シンプル ID メソッド 」を参照してください。
DCIM_EnabledLogicalElementCapabilities	Get Enumerate	-
DCIM_LocalUserIdentity	Get Enumerate	-
DCIM_LANIdentity	Get Enumerate	-
DCIM_SerialIdentity	Get Enumerate	-
DCIM_CLPIdentity	Get Enumerate	-
DCIM_LocalUserAccountManagementService	Get Enumerate	-
DCIM_IPMIAccountManagementService	Get Enumerate	-
DCIM_CLPAccountManagementService	Get Enumerate	-
DCIM_LocalUserAccountManagementCapabilities	Get Enumerate	-
DCIM_IPMICLPAccountManagementCapabilities	Get Enumerate	-
DCIM_RegisteredProfile	Get Enumerate	-
DCIM_LCRegisteredProfile	Get Enumerate	-

シンプル ID メソッド

RequestStateChange() メソッドは、DCIM_Account インスタンスで表現されるアカウントを有効または無効にします。

トラブルシューティングとよくあるお問い合わせ (FAQ)

エラーメッセージ

エラーメッセージ ID および推奨処置に関する詳細は、support.jp.dell.com/manuals で『Dell Lifecycle Controller Remote Services エラーメッセージおよびトラブルシューティングのリスト』を参照してください。エラーメッセージとそれに関連する情報を表示するには、**エラーメッセージ ID** ドロップダウンリストからエラーメッセージ ID を選択します。さらに、delltechcenter.com/page/Lifecycle+Controller から詳細なエラーメッセージレジストリをダウンロードすることも可能です。

自動検出 LCD メッセージ

次の表では、自動検出操作の実行中に表示される LCD メッセージがリストされています。

表 47. 自動検出メッセージ

メッセージ 1	メッセージ 2
Stopped	-
Running	「 自動検出メッセージおよび解決方法 」を参照してください。
Suspended	「 自動検出メッセージおよび解決方法 」を参照してください。
Complete	-

次の表では、LCD メッセージと解決法がリストされています。これらのメッセージは「[自動検出メッセージ](#)」にリストされるメッセージと組み合わせて表示されています。例えば、自動検出が動作中で管理者アカウントが有効になっているときに、実行中およびブロックされており、管理アカウントが有効のメッセージが前面パネルディスプレイに表示されます。

表 48. 「自動検出メッセージおよび解決方法」を参照してください。

メッセージ 2	解決方法
Stopped (デフォルト)	-
Started	-
Auto Discovery disabled	自動検出を有効化します。
Blocked Admin Account Enabled	管理アカウントのすべてを無効化します。
Blocked Active Directory Enabled	Active Directory を無効化します。
Blocked IPv6 Enabled	IPv6 を無効化します。
Blocked No IP on NIC	NIC を有効化します。
No Provisioning Server Found	BIOS で psinfo 値をチェックします。 BIOS で psinfo が設定されていない場合は、DHCP オプションが有効化されているかどうか、および/または DNS サーバー設定が有効であるかを確認します。

メッセージ2	解決方法
Blocked Provisioning Server Unreachable/Invalid address	BIOS で psinfo 値をチェックします。
No Service Tag	サーバーを起動してください。問題が解決しない場合は、テクニカルサポートにお問い合わせください。
SSL connection failed no service at IP/port	BIOS で psinfo 値をチェックするか、DHCP サーバーのベンダーオプションをチェックします。
SSL Connection refused	BIOS で psinfo 値をチェックするか、DHCP サーバーのベンダーオプションをチェックします。
SSL connection failed (サーバー認証)	サーバー証明書が無効であるか、iDRAC にインストールされている信頼できるサーバーの CA 証明書によって署名されていません。プロビジョニングサーバー証明書を交換するか、iDRAC に新しいサーバー証明書をアップロードします。
SSL connection failed (クライアント認証)	iDRAC クライアント証明書はプロビジョニングサーバーが信頼する CA によって署名されていません。iDRAC CA を信頼リストに追加するか、iDRAC で新しい証明書を生成します。
SSL connection failed (その他)	BIOS を介してルートアカウントを有効化し、iDRAC トレースログを取得します。問題が解決しない場合は、テクニカルサポートにお問い合わせください。
SOAP failure	プロビジョニングサーバーは <code>getCredentials()</code> SOAP の呼び出しをサポートしません。プロビジョニングサーバーが自動検出をサポートすること、およびプロビジョニングサーバー情報が DHCP ベンダーオプション、DNS SRV レコード、または BIOS で正しく設定されていることを確認します。
No credentials returned	プロビジョニングサーバーでの既知のサーバーリストにサービスタグがリストされていることをチェックします。
Failed to create account	16 のすべての iDRAC アカウントがすでに使用されていないことを確認します。

よくあるお問い合わせ

本項では、Remote Service のユーザーからのよくあるお問い合わせに回答しています。

1. Lifecycle Controller とは何ですか?

Lifecycle Controller は、カスタマーが診断、オペレーティングシステム (OS) の展開、ファームウェアのアップデート、および設定を実行する際に役立つ内蔵システム管理ソリューションです。Remote Services とは、一般的な用語で、ユーザーがターゲットサーバーにリモートで接続し、スクリプトベースのシステム管理操作を行うことを可能にする機能を指す総括的な用語です。

2. Remote Services の使用のためにクライアントと管理下サーバー間の検証するにはどうすればよいですか?

クライアントと管理下サーバー間の接続の検証には ping ユティリティを使用します。クライアントとネットワークで HTTP および SSL プロトコルが許可されることを確認してください。

3. 部品交換とは何ですか?

部品交換とは、取り付けられた、または交換されたハードウェアコンポーネントのファームウェア、設定、あるいはその両方を、システムが自動的にアップデートできるようにする機能です。

4. CSIOR とは何ですか? なぜ CSIOR が有効になっているのですか?

CSIOR とは Collect System Inventory on Restart (再起動時のシステムインベントリの収集) の略です。この機能を有効にすることで、システム起動中にファームウェアとハードウェアが自動的にアップデートされます。システムは、工場出荷時に CSIOR が有効化された状態で出荷されます。

5. BIOS、RAID、または NIC の属性のいずれかにローカルで変更を行った場合、どのようにシステムインベントリ情報を最新の状態にしておけますか?

システム起動中に手動で <F10> を押すか、CSIOR 属性を有効にして、システム起動ごとにシステムインベントリおよび設定属性情報を収集します。

特定のコンポーネントには、**LastSystemInventoryTime** および **LastUpdateTime** プロパティの値を表示するために **DCIM_SystemView** クラスを列挙します。

6. Lifecycle Controller または Remote Service を使用して管理下システムをアップデートする方法は?

Lifecycle Controller では、起動中に <F10> を押します。Lifecycle Controller GUI では、プラットフォームのアップデートをクリックして、アップデートするデバイスを選択します。Remote Services の詳細については、『Lifecycle Controller Web サービスインタフェースガイド - Windows および Linux』を参照してください。

7. 赤い画面が表示される致命的なエラーが発生したらどうすればよいですか?

赤い画面が表示されたら、システムのコールドリブート (AC パワーサイクル) を実行してください。

8. Lifecycle Controller または Remote Service にアクセスするには、オペレーティングシステム (OS) のインストールが必要ですか?

Lifecycle Controller または Remote Service へのアクセスには、いずれも OS は必要ありません。

9. UEFI バージョンは 32 ビットと 64 ビットのどちらをサポートしますか?

UEFI は 64 ビットをサポートします。

10. システムが Broadcom または Intel NIC を使用しているにもかかわらず、NIC インベントリが何も検出しないのはなぜですか?

お使いのシステムにインストールされている NIC はデル対応ではありません。

11. WS-Management 機能を使用してシステムをリモートで再起動できますか?

はい。システムは、**DCIM_ComputerSystem** クラスの **RequestStateChange()** メソッドを使用して再起動することができます。再起動は、**DCIM_JobService** クラスの **CreateRebootJob()** メソッドを使用して再起動ジョブを作成し、その後ジョブ制御サービスの **SetupJobQueue()** メソッドを使用して再起動ジョブをスケジュールします。

12. DIMM を交換しても LastUpdateTime が変わらないのはなぜですか?

同じスロットで DIMM の取り外しと取り付けが行われても、ビューでの LastUpdateTime は変更されません。

13. WinRM を使用した DCIM_iDracCardAttribute 取得の応答時間を改善する方法はありますか?

はい。次のコマンドを実行して WinRM 設定を設定することにより、PCIDeviceView 列挙にかかる時間を減らすことができます。

```
#winrm set winrm/config @{MaxBatchItems="75"}
```

14. ジョブをクリアする方法は?

DCIM_LifecycleJob を列挙して、Lifecycle Controller 内のジョブをすべてリストし、**DeleteJobQueue()** メソッドを使用して特定のジョブを削除します。

15. すべてのジョブをクリアする方法は?

JID_CLEARALL のジョブ ID で **DeleteJobQueue()** メソッドを呼び出します。

16. HII に対してローカルで変更が行われた場合、WS-Management を介して変更が反映されるのはいつですか?

Lifecycle Controller の終了後、WS-Management インタフェースは約 2 分で使用可能な情報をアップデートします。

17. CreateTargetedConfigJob() メソッドの呼び出しを正常に行うためには、システムはどのような状態であるべきですか?

CreateTargetedConfigJob() メソッドが正常に実行されるには、システムの電源が切れている、BIOS POST 後である (例として、BIOS または UEFI 起動マネージャ)、または OS から起動されている必要があります。ただし、システムが POST を終了するか、セットアップユーティリティを終了するまで、ジョブは実行されません。

18. CreateTargetedConfigJob() メソッドを使用して作成されたジョブを削除するにはどうしたらよいですか?

CreateTargetConfigJob() メソッドを呼び出すと、システムがジョブを実行するために Lifecycle Controller から起動できるようにする追加の再起動ジョブが作成されます。ジョブを削除する場合は、再起動ジョブも削除する必要があります。すべてのジョブを列挙して、該当するジョブを削除に選択するか、**JID_CLEARALL** ですべてのジョブを削除します。

19. クアッドコアプロセッサ用 ProcCore 設定の違いは何ですか?

クアッドポートプロセッサは、属性 ProcCore の値を 4 セットに設定すると、現行値がすべてになります。

20. ジョブの完了後、NIC の LED の点滅属性が常に NULL に設定されるのはなぜですか?

LED の点滅の NIC 属性は一回限り設定です。ただし、SSIB タスクの完了後、タスクが現行値を null に設定します。この属性の目的は、特定の時間（秒）NIC LED を点滅させることにです。

21. SetAttribute() メソッドを介して設定できる属性はいくつありますか?

SetAttribute() メソッドで設定できる属性はひとつだけです。ひとつのメソッドの呼び出しで 2 つ以上の属性を設定するには、設定するコンポーネントのサービスに **SetAttributes()** メソッドを使用してください。

22. 別の属性を設定しているのに、他の属性も設定されているのはなぜですか?

BIOS および NIC の属性には依存性のある属性がいくつかあります。特定の属性を設定する時、依存する属性のすべてが依存性に基づいて変更されます。これは予期される動作です。

- BIOS 依存性 — TPM、電源管理、AC 電源リカバリ、および内蔵 NIC。
- NIC 依存性 — VLAN モードおよび WakeONLAN 属性。

23. 同じタスクで VlanMode および VlanID を設定できますか?

依存性を伴う VlanMode と VlanID 属性を同じタスクで設定することはできません。まず親属性 (VlanMode) を最初の設定操作として設定し、子属性 (VlanID) を 2 番目の設定操作として設定してから、ジョブを確定します。

スキーマ

本項では、一般的なライフサイクルログスキーマを記載します。

ライフサイクルログスキーマ

```
<?xml version="1.0" encoding="UTF-8"?>

<?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/
2001/XMLSchema" xmlns:dm="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified"> <xs:element
name="Description" type="xs:string"/> <xs:element name="MessageID"
type="xs:string"/> <xs:element name="Arg" type="xs:string"/> <xs:element
name="MessageArguments"> <xs:complexType> <xs:sequence minOccurs="0">
<xs:element ref="dm:Arg" minOccurs="0"/> </xs:sequence> </xs:complexType> </
xs:element> <xs:element name="Event"> <xs:complexType> <xs:sequence
minOccurs="0"> <xs:element ref="dm:Description"minOccurs="0"/> <xs:element
ref="dm:MessageID" minOccurs="0"/> <xs:element
ref="dm:MessageArguments" minOccurs="0"/> </xs:sequence> <xs:attribute
name="TimeStamp" type="xs:string" use="required"/> <xs:attribute name="AgentID"
type="xs:integer" use="required"/> <xs:attribute name="Severity"
type="xs:integer" use="required"/> <xs:attribute name="s"
type="xs:string" use="required"/> </xs:complexType> </xs:element> <xs:element
name="Events"> <xs:complexType> <xs:sequence minOccurs="0"> <xs:element
ref="dm:Event" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence>
<xs:attribute name="lang" type="xs:string" use="optional"/> <xs:attribute
name="schemaVersion" type="xs:string" use="optional"/> <xs:attribute
name="timeStamp" type="xs:dateTime" use="optional"/> </xs:complexType> </
xs:element> </xs:schema>
```


使いやすいシステムコンポーネント名

以下の表は、システムコンポーネントの FQDD（完全修飾デバイスディスクリプタ）およびそれに相当する使いやすい名前を示しています。

システムコンポーネント名の FQDD	使いやすい名前
RAID.Integrated.1	内蔵 RAID コントローラ
RAID.Embedded.1-1	内蔵 S110 RAID コントローラ
RAID.Slot.1-1	スロット 1 の RAID コントローラ
NIC.Mezzanine.1B-1	メザニン内の NIC
NIC.Mezzanine.1C-1	
NIC.Mezzanine.1C-2	
NIC.Mezzanine.3C-2	
NIC.Integrated.1	Integrated NIC 1
NIC.Integrated.2	Integrated NIC 2
NIC.Integrated.1-1	内蔵 NIC 1 ポート 1
NIC.Integrated.1-1	内蔵 NIC 1 ポート 1 パーティション 1
NIC.Slot.1-1	スロット 1 ポート 1 の NIC
NIC.Slot.1-2	スロット 1 ポート 2 の NIC
Video.Embedded.1-1	内蔵ビデオコントローラ
HostBridge.Embedded.1-1	内蔵 ホストブリッジ 1
ISABridge.Embedded.1-1	内蔵 ISA ブリッジ 2
P2PBridge.Embedded.1-1	内蔵 P2P ブリッジ 3
P2PBridge.Mezzanine.2B-1	メザニン 1 の内蔵ホストブリッジ (ファブリック B)
USBUHCI.Embedded.1-1	内蔵 USB UHCI 1
USBOHCI.Embedded.1-1	内蔵 USB OHCI 1
USBEHCI.Embedded.1-1	内蔵 USB EHCI 1
Disk.SATAEmbedded.A-1	内蔵 SATA ポート A のディスク
Optical.SATAEmbedded.B-1	内蔵 SATA ポート B の光学ドライブ
TBU.SATAExternal.C-1	外付け SATA ポート C のテープバックアップ
Disk.USBFront.1-1	前面 USB 1 に接続されたディスク
Floppy.USBBack.2-1	背面 USB 2 に接続されたフロッピードライブ
Optical.USBFront.1-1	前面 USB 1 に接続された光学ドライブ
Disk.USBInternal.1	内蔵 USB 1 に接続されたディスク
Optical.iDRACVirtual.1-1	仮想接続された光学ドライブ

システムコンポーネント名の FQDD	使いやすい名前
Floppy.iDRACVirtual.1-1	仮想接続されたフロッピードライブ
Disk.iDRACVirtual.1-1	仮想接続されたディスク
Floppy.vFlash.<string>	vFlash SD カードパーティション 2
Disk.vFlash.<string>	vFlash SD カードパーティション 3
iDRAC.Embedded.1-1	iDRAC
System.Embedded.1-1	システム
HardDisk.List.1-1	ハードディスクドライブ C:
BIOS.Embedded.1-1	システム BIOS
BIOS.Setup.1-1	システム BIOS セットアップ
PSU.Slot.1	電源装置 1
Fan.Embedded.1	ファン 1
Fan.Embedded.2	ファン 2
System.Chassis.1	ブレードシャーシ
LCD.Chassis.1	LCD
Fan.Slot. 1	ファン 1
Fan.Slot. 2	ファン 2
...	...
Fan.Slot. 9	ファン 9
MC.Chassis.1	シャーシ管理コントローラ 1
MC.Chassis.2	シャーシ管理コントローラ 2
KVM.Chassis.1	KVM
IOM.Slot.1	IO モジュール 1
...	...
IOM.Slot.6	IO モジュール 6
PSU.Slot.1	電源装置 1
...	...
PSU.Slot.6	電源装置 6
CPU.Socket.1	CPU 1
System.Modular.2	ブレード 2
DIMM.Socket.A1	DIMM A1