

CA Virtual Assurance for Infrastructure Managers

リファレンス ガイド

リリース 12.8



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。

CA の事前の書面による承諾を受けずに本ドキュメントの全部または一部を複写、譲渡、開示、変更、複本することはできません。本ドキュメントは、CA が知的財産権を有する機密情報です。ユーザは本ドキュメントを開示したり、

(i) 本ドキュメントが関係する CA ソフトウェアの使用について CA とユーザとの間で別途締結される契約または (ii) CA とユーザとの間で別途締結される機密保持契約により許可された目的以外に、本ドキュメントを使用することはできません。

上記にかかわらず、本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負います。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本ドキュメントの制作者は CA です。

「制限された権利」のもとでの提供: アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2013 CA. All rights reserved. 本書に記載された全ての製品名、サービス名、商号およびロゴは各社のそれぞれの商標またはサービスマークです。

CA Technologies 製品リファレンス

このマニュアルが参照している CA Technologies の製品は以下のとおりです。

- CA eHealth®
- CA Embedded Entitlements Manager (CA EEM)
- CA IT Asset Manager (CA ITAM)
- CA ITCM (CA IT Client Manager)
- CA NSM (CA Network and Systems Management)
- CA Patch Manager
- CA Server Automation
- CA Service Desk Manager (CA SDM)
- CA Spectrum®
- CA SystemEDGE
- CA Systems Performance for Infrastructure Managers
- CA Virtual Assurance for Infrastructure Managers
- CA Software Delivery

CA への連絡先

テクニカルサポートの詳細については、弊社テクニカルサポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。

目次

第 1 章: はじめに	15
本書の目的.....	15
対象読者.....	15
関連ドキュメント.....	16
規則.....	17
第 2 章: AutoShell	19
AutoShell について	19
マネージャ シェルとクライアント シェル	21
有効な AutoShell ユーザ	22
AutoShell へのアクセス	22
Common Information Model (CIM) オブジェクト	23
CIM クラスの取得	23
CIM オブジェクトの前提と依存の取得	24
CIM オブジェクトの取得	25
AutoShell の呼び出し	27
対話型モードの使用	29
リモート対話型モードの使用	30
バッチ モード.....	31
リモートバッチ モード.....	32
文字列化.....	33
コンソール (標準出力) へのデータの書き込み	35
キーボード (標準入力) からのデータの読み取り	36
オペレーティング システムへのアクセス	36
OSRedirect クラスの使用.....	37
スクリプトの実行.....	38
run-local Command の使用.....	39
run-remote コマンドの使用	41
AutoShell コア リファレンス.....	49
カテゴリ別コマンド、関数、クラス	50
AutoShell の一般コマンド	53
AutoShell 関数.....	132
AutoShell クラス	159
AutoShell ロード可能モジュールのコマンド リファレンス	191

CA VMware vCenter Server AutoShell コマンド	192
CA Microsoft Hyper-V AutoShell コマンド	237
CA IBM LPAR AutoShell コマンド	352
CA Oracle Solaris ゴースト AutoShell コマンド	383
展開とポリシー設定の AutoShell コマンド	407

第 3 章: コマンドライン スクリプト 449

コマンドライン命令 (CLI)	449
CA Cisco UCS コマンド	450
dpmucs associateserviceprofile コマンド -- サービス プロファイルとブレードの関連付け	452
dpmucs cloneserviceprofile コマンド -- サービス プロファイルのクローン作成	453
dpmucs createmacpool コマンド -- MAC アドレス プールの作成	455
dpmucs createorg コマンド -- 組織の作成	457
dpmucs createserverpool コマンド -- サーバ プールの作成	458
dpmucs createserviceprofile コマンド -- サービス プロファイルの作成	460
dpmucs createtemplatefromserviceprofile コマンド -- サービス プロファイルからのテンプレートの作成	462
dpmucs createuuidpool コマンド -- UUID プールの作成	463
dpmucs createwwpool コマンド -- WWNN プールの作成	465
dpmucs deletemacpool コマンド -- MAC アドレス プールの削除	467
dpmucs deleteorg コマンド -- 組織の削除	469
dpmucs deleteserverpool コマンド -- サーバ プールの削除	470
dpmucs deleteserviceprofile -- サービス プロファイルの削除	472
dpmucs deleteuuidpool コマンド -- UUID プールの削除	473
dpmucs deletewwpool コマンド -- WWNN プールの削除	475
dpmucs deployimage コマンド -- Cisco UCS イメージの展開	477
dpmucs disassociateserviceprofile コマンド -- サービス プロファイルの関連付け解除	479
dpmucs failoverserviceprofile コマンド -- フェールオーバーのためのサービス プロファイルとブレードの関連付け	481
dpmucs getblades コマンド -- ブレード情報の取得	483
dpmucs getchassis コマンド -- シャーシ情報の取得	484
dpmucs getpolicy コマンド -- UCS Manager からのポリシーの取得	486
dpmucs getserviceprofilestatus コマンド -- サービス プロファイル ステータスの取得	487
dpmucs getswitches コマンド -- スイッチ情報の取得	489
dpmucs gettopsystem コマンド -- システム情報の取得	490
dpmucs listserviceprofiles コマンド -- サービス プロファイルの一覧表示	492
dpmucs powercycle コマンド -- ブレードの電源サイクル操作	493
dpmucs renamepool コマンド -- プール名の変更	495
dpmucs serviceprofileupdate コマンド -- サービス プロファイル操作の実行	497

CA Hyper-V CLI コマンド	499
dpmhyperv AddVMNic コマンド -- VM にネットワーク アダプタ コントローラを追加する	499
dpmhyperv AddVMSCSIController コマンド -- VM の SCSI コントローラを作成する	501
dpmhyperv AgentGet コマンド -- エージェントの値を返す	503
dpmhyperv AgentSet コマンド -- エージェントの値を設定する	504
dpmhyperv ChangeVMState コマンド -- VM の状態を変更する	506
dpmhyperv CreateTemplateFromVM コマンド -- VM からテンプレートを作成する	508
dpmhyperv CreateVirtDisk コマンド -- 仮想ディスク イメージを作成する	510
dpmhyperv CreateVirtFloppy コマンド -- 仮想フロッピー ディスク イメージを作成する	512
dpmhyperv CreateVM コマンド -- VM を作成する	513
dpmhyperv CreateVMFromTemplate コマンド -- SCVMM サーバ テンプレートから VM を作成する	518
dpmhyperv CreateVMFromTemplate コマンド -- テンプレートから VM を作成する	523
dpmhyperv DeleteVM コマンド -- VM を削除する	530
dpmhyperv ExpandVirtDisk コマンド -- 仮想ディスクのサイズを拡張する	532
dpmhyperv ExportVM コマンド -- VM をエクスポートする	534
dpmhyperv ExportVMEx コマンド -- VM をエクスポートする	536
dpmhyperv GetHostSwitches コマンド -- ホストのスイッチを取得する	538
dpmhyperv GetJobErrorInfo コマンド -- ジョブのエラー情報を取得する	540
dpmhyperv GetJobInfo コマンド -- ジョブの情報を取得する	541
dpmhyperv GetJobState コマンド -- ジョブの状態を取得する	543
dpmhyperv GetServerInfo コマンド -- すべての管理対象サーバを取得する	544
dpmhyperv GetVersion コマンド -- PMM のバージョンを取得する	545
dpmhyperv GetVMProperties コマンド -- VM のプロパティを取得する	546
dpmhyperv GetVMState コマンド -- VM 状態を取得する	548
dpmhyperv ImportVM コマンド -- VM をインポートする	549
dpmhyperv ImportVMEx コマンド -- VM をコピーおよびインポートする	551
dpmhyperv SetClustered コマンド -- クラスタに対して VM を追加または削除する	553
dpmhyperv SetSysPrepProperties コマンド -- Sysprep プロパティを設定する	555
dpmhyperv SetVMDisk コマンド -- ドライブ コントローラにドライブまたはディスクを割り当てる	559
dpmhyperv SetVMProperties コマンド -- VM のプロパティを設定する	561
dpmhyperv SetVMVirtFloppy コマンド -- 仮想フロッピー イメージを VM フロッピー ドライブに割り当てる	565
dpmhyperv ShowClusterSharedVolumes コマンド -- クラスタ共有ボリュームをすべて表示する	566
dpmhyperv ShowDirectories コマンド -- Hyper-V Server のディレクトリのプロパティを表示する	568
dpmhyperv ShowLogicalDisks コマンド -- 論理ディスクをすべて表示する	569
dpmhyperv ShowPhysDisks コマンド -- ホストの物理ディスクを表示する	570
dpmhyperv ShowSCVMMHardwareProfiles コマンド -- SCVMM ハードウェア プロファイルをすべて表示する	572

dpmhyperv ShowSCVMMGuestOSProfiles コマンド -- ゲスト オペレーティング システムのプロ ファイルを表示する	573
dpmhyperv ShowTemplates コマンド -- VM テンプレートをすべて表示する	574
dpmhyperv ShowVirtFloppy コマンド -- 仮想フロッピー ドライブのリストを表示する	575
dpmhyperv ShowVMDisks コマンド -- 仮想ディスクをすべて表示する	577
dpmhyperv ShowVMNics コマンド -- ネットワーク インターフェース カードをすべて表示する	578
dpmhyperv ShowVMs コマンド -- VM をすべて表示する	579
dpmhyperv ShowVMSCSIControllers コマンド -- SCSI コントローラをすべて表示する	581
dpmhyperv ValidateAgentSNMPAccess コマンド -- エージェントへの SNMP アクセスを検証する	582
dpmhyperv ValidateServerAccess コマンド -- サーバアクセスを確認する	583
dpmhyperv DeleteTemplate コマンド -- テンプレートを削除する	585
CA IBM LPAR の CLI コマンド	586
dpmlpar cycle Command -- 論理パーティションのサイクル	587
dpmlpar delete コマンド -- 論理パーティションの削除	592
dpmlpar getresources コマンド -- LPAR リソースの取得	594
dpmlpar image コマンド -- IBM AIX LPAR を作成する	597
dpmlpar imgjobcheck コマンド -- IBM AIX イメージング ジョブ ステータスの取得	611
dpmlpar list コマンド -- HMC リソースのリスト表示	613
dpmlpar setresources コマンド -- メモリ リソースの追加	616
dpmlpar setresources コマンド -- プロセッサ リソースの追加	619
dpmlpar setresources コマンド -- メモリ リソースの削除	622
dpmlpar setresources コマンド -- プロセッサ リソースの削除	624
dpmlpar setresources コマンド -- メモリ リソースの更新	627
dpmlpar setresources コマンド -- プロセッサ リソースの更新	629
CA Microsoft Cluster Server CLI コマンド	632
dpmmcs getclusterinfo コマンド -- クラスタ情報の取得	633
dpmmcs getgroupresourcesinfo コマンド -- クラスタ グループのリソース情報の取得	634
dpmmcs getnetworkinterfacesinfo コマンド -- ネットワーク インターフェース情報の取得	635
dpmmcs getnetworksinfo コマンド -- ネットワーク情報の取得	637
dpmmcs getnodesinfo コマンド -- クラスタ ノード情報の取得	638
dpmmcs getresourcegroupsinfo コマンド -- クラスタ リソース グループ情報の取得	639
dpmmcs getresourcesdependenciesinfo コマンド -- リソースの依存関係情報の取得	640
dpmmcs getresourcetypesinfo コマンド -- クラスタ リソース タイプ情報の取得	642
dpmmcs getservicesinfo コマンド -- クラスタ サービス情報の取得	643
dpmmcs movegroup コマンド -- 別のノードへのリソース グループの移動	644
dpmmcs pauseservice コマンド -- クラスタ サービスの一時停止	646
dpmmcs bringgrouponline コマンド -- リソース グループのオンライン化	647
dpmmcs bringresourceonline コマンド -- リソースのオンライン化	648
dpmmcs resumeservice コマンド -- クラスタ サービスの再開	650
dpmmcs startservice コマンド -- クラスタ サービスの開始	651

dpmmscs stopservice コマンド -- クラスタ サービスの停止	652
dpmmscs takegroupoffline コマンド -- リソース グループのオフライン化	654
dpmmscs takesresourceoffline コマンド -- リソースのオフライン化	655
CA Solaris ゾーン CLI コマンド	656
dpmzone associateproject コマンド -- タスクとプロジェクトの関連付け	656
dpmzone clonezone コマンド -- ゾーンのクローン作成	658
dpmzone createandinstallzone コマンド -- ゾーンの作成およびインストール	659
dpmzone createpool コマンド -- リソース プールの作成	662
dpmzone createproject コマンド -- プロジェクトの作成	664
dpmzone createzone コマンド -- ゾーンの作成	665
dpmzone customcommand コマンド -- カスタム コマンドの実行	668
dpmzone deleteproject コマンド -- プロジェクトの削除	669
dpmzone deletezone コマンド -- ゾーンの削除	670
dpmzone installzone コマンド -- ゾーンのインストール	671
dpmzone movezone コマンド -- ゾーンの移動	673
dpmzone rebootzone コマンド -- ゾーンの再起動	674
dpmzone renamezone コマンド -- ゾーンの名称変更	675
dpmzone setpoolscheduler コマンド -- プール スケジューラの設定	676
dpmzone setprocessorset コマンド -- プロセッサ セットの CPU 最大数の設定	678
dpmzone showinterfaces コマンド -- ネットワーク インターフェースの表示	679
dpmzone showpools コマンド -- リソース プールの表示	680
dpmzone showprojects コマンド -- プロジェクトのプロパティの表示	681
dpmzone showresourceset コマンド -- リソース セットの表示	682
dpmzone showtasks コマンド -- プロジェクトのタスクのリスト表示	683
dpmzone showzoneinterfaces コマンド -- ゾーンのネットワーク インターフェースの表示	684
dpmzone showzoneprocess コマンド -- ゾーンのプロセスの表示	685
dpmzone showzoneprojects コマンド -- ゾーンのプロジェクトの表示	686
dpmzone showzones コマンド -- ゾーンの表示	687
dpmzone startzone コマンド -- ゾーンの起動	687
dpmzone stopzone コマンド -- ゾーンの停止	688
dpmzone uninstallzone コマンド -- ゾーンのアインストール	690
CA VMware vCenter Server の CLI コマンド	691
dpmvc addesxhost コマンド -- ESX ホストを追加する	691
dpmvc addvmvmdisk コマンド -- 仮想ディスクの追加	694
dpmvc addvmvnic コマンド -- 仮想 NIC の追加	697
dpmvc clone コマンド -- VM のクローン作成	700
dpmvc createdatastore コマンド -- データストアの作成	710
dpmvc cycle コマンド -- VM のサイクル	713
dpmvc datastore コマンド -- データ ストア プロパティの取得	717
dpmvc delete コマンド -- VM の破棄	720

dpmvc distributedswitch コマンド -- 分散仮想スイッチの管理	724
dpmvc entermaintenancemode コマンド -- 保守モードに設定する	730
dpmvc exitmaintenancemode コマンド -- 保守モードの終了	732
dpmvc faulttolerance コマンド -- フォールトトレラント操作の指定	734
dpmvc getavailablescsidisks コマンド -- 利用可能な SCSI ディスクの取得	736
dpmvc gethosthba コマンド -- ホストバスアダプタの取得	738
dpmvc getresources コマンド -- VM リソースの取得	739
dpmvc getshares コマンド -- VM 共有の取得	742
dpmvc getsnapshots コマンド -- VM スナップショットの取得	745
dpmvc imgjobcheck コマンド -- クローン作成ジョブステータスの取得	747
dpmvc migrate コマンド -- VM のマイグレート	750
dpmvc removedatastore コマンド -- データストアの削除	753
dpmvc removevmvmdisk コマンド -- 仮想ディスクの削除	755
dpmvc removevmvnic コマンド -- 仮想 NIC の削除	757
dpmvc setcpu コマンド -- VM の CPU 設定	759
dpmvc setmemory コマンド -- VM のメモリサイズの設定	761
dpmvc setresources コマンド -- VM リソースの設定	763
dpmvc setshares コマンド -- VM 共有の設定	767
dpmvc snapshot コマンド -- スナップショットを管理する	772
dpmvc templatetovm コマンド -- テンプレートを VM に変換する	778
dpmvc vapp コマンド -- vApp の管理	782
dpmvc virtualswitch コマンド -- 仮想スイッチの管理	786
dpmvc vmtotemplate コマンド -- VM をテンプレートに変換する	790
dpmovf import コマンド -- OVF パッケージのインポート	793
管理コマンドラインユーティリティ	795
dpmkpdb Command-- パフォーマンスデータのインポートまたはエクスポート	795
dpmutil の構文	796
dpmutil-amq コマンド--ActiveMQ Message Broker の設定	798
dpmutil-ecs コマンド--Windows スケジューラの設定	799
dpmutil-hd コマンド -- ヘルプデスクの設定	799
dpmutil-hmc コマンド -- HMC サーバまたは IVM サーバの設定	800
dpmutil-hyperv コマンド -- Hyper-V Server の設定	801
dpmutil-mgmtldb コマンド -- 管理データベースの設定	802
dpmutil-perfdb コマンド -- パフォーマンスデータベースの設定	803
dpmutil-remotevcaim コマンド--リモート VC AIM の設定	804
dpmutil-sc コマンド -- サービスコントローラの設定	804
dpmutil-scvmmserver コマンド -- SCVMM Server の設定	806
dpmutil-snmp コマンド -- SNMP の設定	807
dpmutil-systemcredential コマンド -- 外部システムの認証情報の設定	809
dpmutil-sysuser コマンド -- サービス起動時に使用されるローカル認証情報の設定	810

dpmutil vcdefault コマンド -- デフォルトの VMware vCenter 管理者ユーザの設定	811
dpmutil vcserver コマンド -- VMware vCenter の設定	812
dpmutil -vioserver コマンド -- VIO サーバの設定	813
Common Discovery コマンド	813
dpmcda cancelnetworkscan コマンド -- ネットワーク ディスカバリの取り消し	814
dpmcda deletenetwork コマンド -- ネットワークの削除	815
dpmcda discoverhost コマンド -- ホストを検出する	816
dpmcda discovernetwork コマンド -- ネットワークを検出する	817
収集エンジン コマンド	819
dpmce ceconfig -configuremetric コマンド -- メトリックを設定する	819
dpmce ceconfig -configuremetricfilter コマンド -- メトリック フィルタを設定する	822
dpmce ceconfig -disablemetric コマンド -- メトリックを無効にする	825
dpmce ceconfig -disablemetricforoverall コマンド -- メトリックを無効にし、全体的な計算から除 外する	826
dpmce ceconfig -enablemetric コマンド -- メトリックを有効にする	827
dpmce ceconfig -enablemetricforoverall コマンド -- メトリックを有効にし、全体的な計算に組み 込む	829
dpmce ceconfig -getglobal コマンド -- データ センターの間隔を取得する	830
dpmce ceconfig -getnode コマンド -- ホストの収集間隔を取得する	832
dpmce ceconfig -getmetriclist コマンド -- メトリック リストを取得する	833
dpmce ceconfig -refreshmetriclist コマンド -- メトリック収集を再試行する	834
dpmce ceconfig -setalias コマンド -- エイリアスを設定する	835
dpmce ceconfig -setglobal コマンド -- データ センターの間隔を設定する	837
dpmce ceconfig -setnode コマンド -- ホストの間隔を設定する	839
dpmce ceconfig -updatemetric コマンド -- メトリックを更新する	841
イベント コマンド	843
dpmeventmanager create_event コマンド -- イベントの作成	843
dpmeventmanager get_events コマンド -- イベントの取得	845
汎用シェル コマンド	848
caaipsecurity コマンド -- Web サービスのセキュリティ チェック用の認証情報の指定	849
createdpmdefaultgroup コマンド -- サービスの作成または更新	850
dpmcmd run コマンド -- スクリプトの実行	850
dpmrpt schedulereport コマンド -- レポートのスケジューリング	851
dpmssh run コマンド -- リモート シェル コマンドまたはスクリプトの実行	854
ヘルプ デスク コマンド	855
dpmhd addcomment コマンド -- チケットをコメントで更新する	855
dpmhd close コマンド -- ヘルプ デスク チケットのクローズ	856
dpmhd getattribute コマンド -- チケットの属性値の取得	857
dpmhd getstatus コマンド -- チケットのステータスの取得	859
dpmhd open コマンド -- ヘルプ デスク チケットのオープン	860

dpmhd setattribute コマンド -- チケットの属性の設定	861
dpmhd setstatus コマンド -- チケットのステータスの設定	863
イメージング コマンド	864
dpmimg image コマンド -- Solaris イメージを展開する	865
dpmimg imgjobcheck コマンド -- Solaris イメージング ジョブのステータスを取得する	870
ポリシー コマンド	872
dpmpolicy createdefaultactions コマンド -- デフォルト アクションを作成する	872
dpmpolicy exportaction コマンド -- アクションをファイルにエクスポートする	874
dpmpolicy exportrule コマンド -- ルールをファイルにエクスポートする	875
dpmpolicy findbestmachine コマンド -- ベスト サーバを検索する	876
dpmpolicy findcandidatemachine コマンド -- 候補サーバを検索する	877
dpmpolicy getruleexccount コマンド -- ルール実行回数を取得する	880
dpmpolicy importaction コマンド -- アクションをファイルからインポートする	882
dpmpolicy importrule コマンド -- ルールをファイルからインポートする	883
dpmpolicy resetruleexccount コマンド -- ルールの試行回数をリセットする	884
dpmpolicy runaction コマンド -- アクションを実行する	886
dpmpolicy setruleexclimit コマンド -- ルールの制限を設定する	889
ポリシー設定	891
caismutlity コマンド -- [ポリシー設定] UI への機能の追加	891
リソース マネージャ コマンド	893
dpmresourcemgr addbaremetal コマンド -- 物理コンピュータを追加する	894
dpmresourcemgr addcomputertonetwork コマンド -- ネットワークにサーバを追加する	895
dpmresourcemgr addipaddresspool コマンド -- IP アドレス プールを追加する	897
dpmresourcemgr addmachinestosome コマンド -- サービスにサーバを追加する	899
dpmresourcemgr addtocontainer コマンド -- コンテナへの内容物の追加	900
dpmresourcemgr createcontainer コマンド -- コンテナの作成	901
dpmresourcemgr createdefaultservice コマンド -- デフォルト サービスを作成する	902
dpmresourcemgr createipnetwork コマンド -- ネットワークを作成する	904
dpmresourcemgr createresourcegroup コマンド -- リソース グループの作成	906
dpmresourcemgr createresourcetemplate コマンド -- サーバ グループ テンプレートの作成	908
dpmresourcemgr createservice コマンド -- サービスを作成する	910
dpmresourcemgr deletebaremetal コマンド -- 物理コンピュータを削除する	911
dpmresourcemgr deletecontainer コマンド -- コンテナの削除	913
dpmresourcemgr deleteipnetworks コマンド -- ネットワークを削除する	914
dpmresourcemgr deletesystem コマンド -- システムを削除する	915
dpmresourcemgr getdiscoverednetworks コマンド -- 検出されたネットワークを取得する	916
dpmresourcemgr getipaddresspools コマンド -- IP アドレス プールをリスト表示する	917
dpmresourcemgr getipnetworks コマンド -- ネットワークをリスト表示する	919
dpmresourcemgr getserverlist コマンド -- サーバのリストを取得する	920
dpmresourcemgr getservicelist コマンド -- サービスのリストを取得する	921

dpmresourcemgr getsubnetdefaults コマンド -- サブネットのデフォルト値をリスト表示する	922
dpmresourcemgr removecomputerfromipnetwork コマンド -- ネットワークからサーバを削除する	924
dpmresourcemgr removefromcontainer コマンド -- コンテナの内容物の削除	925
dpmresourcemgr removeipaddresspool コマンド -- IP アドレス プールを削除する	926
dpmresourcemgr removemachinesfromservice コマンド -- サービスからサーバを削除する	928
dpmresourcemgr removeresourcegroup コマンド -- リソース グループの削除	929
dpmresourcemgr removeservice コマンド -- サービスを削除する	930
dpmresourcemgr removesourcetemplate コマンド -- リソース テンプレートの削除	932
dpmresourcemgr retrieveresourcegroup コマンド -- リソース グループの取得	933
dpmresourcemgr retrieveresourcetemplate コマンド -- リソース テンプレートの取得	934
dpmresourcemgr setcontainername コマンド -- コンテナ名の設定	936
dpmresourcemgr setcontainerservice コマンド -- サービスへのコンテナの設定	937
dpmresourcemgr setmanagementip コマンド -- ホストの管理 IP を設定する	939
dpmresourcemgr setmanagementstatus コマンド -- ホストの管理ステータスを設定する	940
dpmresourcemgr setserviceproperties コマンド -- サービス プロパティを設定する	941
dpmresourcemgr updateresourcegroup コマンド -- リソース グループの更新	943
dpmresourcemgr updateresourcetemplate コマンド -- リソース テンプレートの更新	945
dpmresourcemgr updateservice コマンド -- サービスを更新する	946
dpmresourcemgr updateipnetwork コマンド -- ネットワークを更新する	948
dpmresourcemgr updatesubnetdefaults コマンド -- ネットワークのデフォルト値を更新する	950
dpmresourcemgr updateipaddresspool コマンド -- IP アドレス プールを更新する	952
リモート モニタリング コマンド	953
rmonwatch add コマンド -- MIB テーブルに行を追加する	954
rmonwatch del コマンド -- MIB テーブルから行を削除する	957
rmonwatch getTable コマンド -- rmonWbem MIB テーブルを表示する	959
rmonwatch getVersion コマンド -- バージョン情報を表示する	961
SNMP の詳細パラメータ	962
ログ ファイル	964
利用可能なログ ファイル	965
ログ ファイル設定	967

第 1 章: はじめに

このセクションには、以下のトピックが含まれています。

[本書の目的](#) (P. 15)

[対象読者](#) (P. 15)

[関連ドキュメント](#) (P. 16)

[規則](#) (P. 17)

本書の目的

このガイドでは、AutoShell、CA Virtual Assurance CLI コマンド、およびパフォーマンス メトリックについて説明します。用語集では、仮想化テクノロジーで使用される特定の用語について説明します。

AutoShell は、繰り返し実行する複雑な管理タスクの自動化に使用できる、CA Virtual Assurance の中央コマンドラインインターフェースです。

AutoShell は、オブジェクト指向、XML、および正規表現を処理するために、スタンドアロンの JavaScript 実装を提供します。AutoShell では、標準バージョンの Mozilla Spidermonkey JavaScript インタープリタを使用します。このインタープリタは、Mozilla Firefox Web ブラウザにも JavaScript 機能を提供しています。このインタープリタは、AutoShell で JavaScript 構文を直接使用して、たとえばディレクトリ リストの表示を行うことを可能にします。

このガイドでは、すべての AutoShell コマンド、使用例、およびこのユーザインターフェースのベスト プラクティスを説明します。

対象読者

このガイドは、CA Virtual Assurance をインストールおよび設定して、仮想環境を管理するために使用する管理者を対象としています。読者が、お使用の環境で使用されるオペレーティング システム、仮想化技術、および SNMP に精通していることを前提とします。

関連ドキュメント

CA Virtual Assurance のマニュアルは、次のマニュアルで構成されています。

管理ガイド

ユーザの環境の仮想リソースを管理するために、CA Virtual Assurance を管理および使用する方法を調査します。

インストール ガイド

簡単なアーキテクチャ情報、さまざまなインストール方法、インストール後の設定情報、および導入時の手順が含まれます。

オンライン ヘルプ

CA Virtual Assurance ユーザ インターフェースを使用するためのウィンドウの詳細、および手順の説明を提供します。

リファレンス ガイド

AutoShell、CLI コマンド、および MIB 属性に関する詳細情報を提供します。

パフォーマンス メトリック参照

サポート対象プラットフォームのシステム パフォーマンスのモニタリングに利用可能なパフォーマンス メトリックについて説明します。

リリース ノート

オペレーティング システムのサポート、システム要件、発行済みの修正プログラム、各国語のサポート、既知の問題、およびドキュメントロードマップに関する情報を提供します。

サービス レスポンス モニタリング ユーザ ガイド

SRM のインストールおよび設定の詳細が記載されています。

SystemEDGE ユーザ ガイド

SystemEDGE のインストールおよび設定の詳細について説明します。

SystemEDGE リリース ノート

オペレーティング システムのサポート、システム要件、および機能に関する情報を提供します。

規則

このガイドでは、以下の規則を使用します。

大文字と小文字の区別

このガイドで言及されるクラス、コマンド、ディレクティブ、環境パラメータ、関数、プロパティの名前はすべて大文字と小文字を区別します。また、記載されているとおりに正確に入力する必要があります。システム コマンドと環境変数名は、オペレーティング システムの要件に応じて、大文字と小文字が区別される場合があります。

相互参照

他のガイド、またはこのガイドの他のセクション内の情報への参照は、以下の形式で表示されます。

ガイド名

別のガイドの名前を示します。

「章名」

このガイドまたは別のガイドの章の名前を示します。

同義語

属性、オブジェクト、オブジェクト識別子（OID）などの用語は、このドキュメントでは「変数」と同義です。

SystemEDGE エージェント、CA SystemEDGE などの用語は、このドキュメントでは SystemEDGE と同義です。

構文

構文とユーザ入力では、以下の形式を使用します。

斜体

実際の値を指定する必要がある変数名またはプレースホルダを示します。

{a|b}

オペランド **a** または **b** を選択する必要があることを示します。

[] または [[]]

オプションのオペランドを示します。

構文例

以下の例は、上記の規則を使用しています。

```
modify -t ZONE [-m zoneserver] -p psetname {-min mincpu|-max maxcpu} pset -session  
ssh
```

オペランド **-min** および **-max** は必須ですが、プロセッサセット内の CPU の最小数または最大数のどちらを定義するかに応じて、いずれか 1 つのみを使用します。オペランド **-m** は、指定しなくてもこのコマンドは機能します。コマンドのほかの部分は記載されているとおりに入力する必要があります。

デフォルト ディレクトリ

パス ステートメント内で使用する *CASYSEdge* は、SystemEDGE がインストールされているディレクトリを示します。デフォルト：
C:¥Program Files¥CA¥SystemEDGE.

インストール パス

パス ステートメント内で使用する *Install_Path* は、CA Virtual Assurance または CA Virtual Assurance コンポーネントがインストールされているディレクトリを示します。

デフォルト

- Windows x86 : C:¥Program Files¥CA
- Windows x64 : C:¥CA、C:¥Program Files (x86)¥CA、または C:¥Program Files¥CA
- UNIX、Linux : /opt/CA

第 2 章: AutoShell

AutoShell は、複雑な反復タスクと管理タスクを自動化するために使用できる、コマンドラインおよびスクリプト環境です。

この章では、AutoShell コマンドについて詳しく説明し、数多くの使用例と、このユーザ インターフェースのベスト プラクティスを紹介します。

このセクションには、以下のトピックが含まれています。

[AutoShell について](#) (P. 19)

[マネージャ シェルとクライアントシェル](#) (P. 21)

[有効な AutoShell ユーザ](#) (P. 22)

[AutoShell へのアクセス](#) (P. 22)

[Common Information Model \(CIM\) オブジェクト](#) (P. 23)

[AutoShell の呼び出し](#) (P. 27)

[文字列化](#) (P. 33)

[コンソール \(標準出力\) へのデータの書き込み](#) (P. 35)

[キーボード \(標準入力\) からのデータの読み取り](#) (P. 36)

[オペレーティング システムへのアクセス](#) (P. 36)

[スクリプトの実行](#) (P. 38)

[AutoShell コア リファレンス](#) (P. 49)

[AutoShell ロード可能モジュールのコマンド リファレンス](#) (P. 191)

AutoShell について

AutoShell は、スクリプト言語とコマンドライン シェルを組み合わせたもので、一般的に JavaScript と呼ばれている、標準化されたスクリプト言語の ECMA-Script を基にしています。JavaScript は Web ページ上で使用されるスクリプト言語としてよく知られていますが、必ずしもブラウザで実行する必要があるわけではありません。オブジェクト指向、XML、正規表現処理のサポートを実装した、スタンドアロンのスクリプト言語です。

AutoShell では、Mozilla Firefox Web ブラウザに JavaScript 機能も提供する Mozilla Spidermonkey JavaScript インタープリタの標準装備のバージョンを使用します。これにより、AutoShell で JavaScript 構文を直接使用して、たとえばディレクトリ リストの表示を行うことを可能にします。

通常は、文字列を引用符で囲み、特殊文字をエスケープして、関数呼び出しを丸かっこで囲む必要があります。この方法で命令を入力することは効率的ではありません。そのため、AutoShell では、JavaScript 言語の上に、一般的なシェルタイプのコマンドと必須およびオプションの引数に対応する独自のコマンド変換レイヤを実装しています。これらのコマンドでは、文字列を引用符で囲まなくても、ベースとなる JavaScript 構文に変換されます。

主な AutoShell の機能は以下のとおりです。

- オペレーティング システムにアクセスして子プロセスを実行し、出力を取得する。
- リモート スクリプトを並列して実行し、集中して実行できないタスクを効率的に処理する。
- リモート ファイル システムにアクセスし、コンピュータ間でファイルを直接コピーすることで、高度なプロビジョニングを実現する。
- 通常はライブラリ呼び出し、Java、COM などの管理 API で使用する、共通のアプリケーションプログラミング インターフェース (API) メカニズムをサポートし、これらの API にスクリプトからアクセスできるようにする。
- リモート レジストリ アクセス (Windows 固有)。
- 自分自身のクライアント シェルおよび任意のインストール パッケージをリモートの Windows システムにインストール配信する。

注: JavaScript 言語リファレンスと例については、Web サイト <https://developer.mozilla.org/ja/JavaScript> を参照してください。AutoShell では、JavaScript バージョン 1.7 を使用しています。

関連項目

[マネージャ シェルとクライアント シェル](#) (P. 21)

[AutoShell の呼び出し](#) (P. 27)

[文字列化](#) (P. 33)

[スクリプトの実行](#) (P. 38)

[AutoShell コア リファレンス](#) (P. 49)

マネージャ シェルとクライアント シェル

AutoShell は、マネージャ シェルとクライアント シェルの 2 つの部分で構成されます。コマンドの入力やスクリプト実行の開始を行うには、マネージャ システム上でマネージャ シェルを使用してローカル タスクを実行します。クライアント シェルは、AutoShell リモート実行インフラストラクチャの一部と見なされます。リモート実行やリモート ファイル システム アクセスを可能にするには、管理対象のすべてのリモート ノードにクライアント シェルをインストールする必要があります。このほかにクライアント シェルを使用する必要はありません。マネージャ シェルは、安全な SSH 接続を通して、リモート操作を行うクライアント シェルを起動します。そのため、「AutoShell」という用語は、常にマネージャ シェルを表します。

マネージャ シェル

対話形式でコマンドを入力、スクリプトを実行、マネージャ コンピュータ上でローカル タスクを実行することができます。

クライアント シェル

SSH (Secure Shell) 接続を使用してリモート実行やリモート ファイル システム アクセスを可能にします。また、管理対象のすべてのサーバ リモート ノードにインストールする必要があります。

AutoShell は、対話型モードでもスクリプト (バッチ) モードでも起動できます。

有効な AutoShell ユーザ

CA Virtual Assurance のインストール時に、インストール ウィザードのネイティブ セキュリティ ユーザ情報画面で CA Embedded Entitlements Manager (Embedded Entitlements Manager) のユーザ ID とパスワードを定義します。認証情報は Embedded Entitlements Manager データベースに格納されます。ユーザは CA Virtual Assurance 管理者グループに割り当てられ、CA Virtual Assurance ユーザ インターフェースおよび AutoShell マネージャへのログインに使用できます。

Embedded Entitlements Manager を使用する CA Virtual Assurance コンポーネントが、ローカルまたはリモート システムにインストールされている場合、AutoShell マネージャは常に Embedded Entitlements Manager データに対するログイン認証情報を検証します。そうでない場合、AutoShell マネージャは、Windows 認証に対するログイン認証情報を検証します。

AutoShell へのアクセス

CA Virtual Assurance サーバから AutoShell にアクセスできます。

AutoShell にアクセスする方法

1. Windows エクスプローラを開き、以下のディレクトリに移動します。
C:\Program Files\CA\SC\AutoShellClient
2. caaipaomautoshellclient.exe をダブルクリックします。
AutoShell のコマンドプロンプト ウィンドウが表示されます。
3. 有効な CA Virtual Assurance ユーザのユーザ名とパスワードを入力します。

AutoShell コマンドプロンプトが表示されます。

関連項目

[有効な AutoShell ユーザ](#) (P. 22)

Common Information Model (CIM)オブジェクト

CA Virtual Assurance では、CIM ベースのオブジェクト モデルを使用して、管理対象オブジェクトの情報を保存します。管理対象オブジェクトは、AutoShell スクリプト環境内でネイティブ オブジェクトとして取得、問い合わせ、操作できます。CIM の詳細については、Distributed Management Task Force (DMTF) によって公表された CIM の仕様

(<http://www.dmtf.org/standards/cim/>) を参照してください。DMTF CIM のドキュメント形式および機能仕様については、http://www.dmtf.org/standards/published_documents/DSP0202_1.0.0.pdf を参照してください。

インストール時にオプションの AutoShell CIM 機能を有効にするには、AutoShell CIM ALM (AutoShell ロード モジュール) を選択します。

重要: AutoShell は、CA Virtual Assurance オブジェクト ストアへの完全な書き込みアクセスを提供します。不適切な AutoShell コマンドやスクリプトを使用すると、このストアの整合性がおかしくなり、ほかの CA Virtual Assurance コンポーネントの誤動作を引き起こす結果となります。AutoShell 内部からは、常に読み取り専用でオブジェクト ストアを操作してください。

CIM クラスの取得

AutoShell で CIM クラスを取得できます。

DMTF CIM クラスを取得するには、以下のコマンドを使用します。

```
get-autoShellClasses CIM_*
```

CA 定義された CIM 拡張クラスを取得するには、以下のコマンドを使用します。

```
get-autoShellClasses CA_*
```

CIM オブジェクトの前提と依存の取得

`getAntecedents()` および `getDependents()` メソッドを使用して、CIM オブジェクトの前提と依存を取得できます。

CIM オブジェクトの前提を取得するには、`getAntecedents()` メソッドを呼び出します。

```
// すべてのコンピュータ システムを取得
arr=CA_ComputerSystem.getInstances()
// 返された最初のコンピュータ システムで処理を続行
cs=arr[0]
// 前提を取得
arrAnt = cs.getAntecedents()
for(i=0; i<arrAnt.length; i++)
{
    show-cimObject arrAnt[i]
}
```

CIM クラスの前提インスタンスを取得するには、`getAntecedents()` メソッドを呼び出します。

```
os = cs.getAntecedents("CIM_OperatingSystem");
show-cimObject os
```

CIM オブジェクトの依存を取得するには、`getDependents()` メソッドを呼び出します。

```
// すべてのコンピュータ システムを取得
arr=CA_ComputerSystem.getInstances()
// 返された最初のコンピュータ システムで処理を続行
cs=arr[0]
// 依存を取得
arrAnt = cs.getDependents()
for(i=0; i<arrAnt.length; i++)
{
    show-cimObject arrAnt[i]
}
```

CIM クラスの依存インスタンスを取得するには、`getDependents()` メソッドを呼び出します。

```
fs = cs.getDependents("CIM_FileSystem");
show-cimObject fs
```


CIM オブジェクトの取得

各 CIM クラスには、クラス オブジェクトの配列を返す静的な `getInstances()` メソッドがあります。

CA Virtual Assurance で管理されているシステムをすべて取得するには、以下のコマンドを使用します。

```
aObjs = CA_ComputerSystem.getInstances()  
? "Query returned", oObjs.length
```

CA Virtual Assurance で管理されているシステムのサブセットを取得するには、以下のフィルタ表現のいずれかを使用します。

```
aObjs = CA_ComputerSystem.getInstances("ElementName='Everest.local'")
```

または

```
aObjs = CA_ComputerSystem.getInstances("ElementName like 'Ev*')"
```

システム名と説明を取得するには、以下のコマンドを使用します。

```
aObjs = CA_ComputerSystem.getInstances();  
for(i=0; i<aObjs.length; i++)  
    ? aObjs[i].getElementName(),aObjs[i].getDescription()
```

CIM オブジェクトのプロパティと値を取得するには、以下のコマンドを使用します。

```
aObjs = CA_ComputerSystem.getInstances("ElementName='Debian1'")  
show-cimObject aObjs[0]
```

例: show-cimObject コマンドの出力

```
CA_ComputerSystem
=====
Caption: null
Description: null
ElementName: Debian1
Facets: 44
InstallDate: null
Name: 47332da1-e782-0308-d614-c2e99de989bc
OperationalStatus: undefined
StatusDescriptions: undefined
Status: null
HealthState: 5
PropagatedHealthState: null
IsPropagated: true
PrimaryStatus: null
DetailedStatus: null
OperatingStatus: null
CommunicationStatus: null
ManagementStatus: null
EnabledState: null
OtherEnabledState: null
RequestedState: null
EnabledDefault: null
TimeOfLastStateChange: null
AvailableRequestedStates: undefined
TransitioningToState: null
CreationClassName: CA_ComputerSystem
NameFormat: null
PrimaryOwnerName: null
PrimaryOwnerContact: null
Roles: undefined
OtherIdentifyingInfo: undefined
IdentifyingDescriptions: undefined
Dedicated: undefined
OtherDedicatedDescriptions: undefined
ResetCapability: null
PowerManagementCapabilities: undefined
HostName: 192.168.1.119
IpAddress: null
SerialNumber: null
Version: null
CpuCount: 1
TotalVisibleMemorySize: 256
PatchLevel: null
BiosVersion: null
Fans: null
Chassis: null
```

```
DrCapable: null
VendorId: null
MacAddress: null
ArchTypeUID: null
Available: null
LastUpdate: null
SnapshotType: null
TotalStorageCapacity: null
Model: null
CpuSpeed: null
SystemImageId: null
HardwareClassId: null
```

AutoShell の呼び出し

以下のコマンドを使用して、コマンドライン インターフェースから AutoShell を起動できます。

```
caaiapaautoshell.exe [-U username] [-P password] [-h host] [[-f] scriptfile]
[scriptargs] [-e expr] [-i]
```

-U *username*

有効な CA Virtual Assurance ユーザを指定します。また、CAASUSER 環境変数を使用してユーザ名を設定することもできます。

-P *password*

ユーザのパスワードを指定します。また、CAASSPASS 環境変数を使用してパスワードを設定することもできます。

-f *scriptfile* [*scriptargs*]

(オプション) 指定したスクリプト ファイルをオプションのスクリプト引数と共にロードします。このパラメータは複数回指定できます。

-e *expr*

(オプション) 指定された式を評価します。

-i

(オプション) スクリプトをロードした後、対話型モードに入ります。

-locale=*locale*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。 コマンドプロンプトのロケールを使用するには、"native" を指定します。

例

hello.js スクリプトを実行します。

```
caaipaomautoshell.exe -U vaimuser -P topsecret -f hello.js
```

4*3 の値を計算します。

```
caaipaomautoshell.exe -U vaimuser -P topsecret -e 4*3
```

ユーザ名とパスワードの環境変数を設定し、 $\sin(\pi/2)$ の値を計算します。

```
set CAASUSER=vaimuser  
set CAASSPASS=topsecret  
caaipaomautoshell.exe -e Math.sin(Math.PI/2)
```

関連項目

[有効な AutoShell ユーザ](#) (P. 22)

[対話型モードの使用](#) (P. 29)

[リモート対話型モードの使用](#) (P. 30)

[バッチモード](#) (P. 31)

[リモートバッチモード](#) (P. 32)

対話型モードの使用

対話型モードを使用して、ローカルまたはリモート ホスト上で単一の操作を実行したり、スクリプト コマンドシーケンスを作成するときに小規模なコードフラグメントをテストしたりします。

対話型モードを使用する方法

1. 以下のいずれかの方法で次の **AutoShell** 実行可能ファイルを実行します。

```
caaipaomautoshell.exe
```

AutoShell からユーザ名とパスワードの入力を求められます。その後、**AutoShell** は、認証確認を実行するか、リモート クライアントを認証するなど将来の使用に備えて認証情報を保存します。この判断は、製品統合のタイプによって決まります。

```
caaipaomautoshell.exe -U bob -P mypassword
```

-U および **-P** スイッチを使用してユーザ名とパスワードを指定します。

2. (オプション) 環境変数 **CAASUSER** および **CAASSPASS** を以下のように設定して、認証情報を指定します。

```
set CAASUSER=bob  
set CAASSPASS=mypassword
```

正常にログオンした後、**AutoShell** プロンプトが表示されます。

```
asmgr::->
```

```
asmgr
```

ローカル ホスト名を指定します。

3. AutoShell はベースのスクリプト言語として JavaScript を使用しているため、プロンプトで任意の有効な JavaScript 式を入力します。

例：

2+3 は 5 を表示します。

AutoShell では、式が単純な場合はコマンドプロンプトで出力コマンドを使用せず、自動的に結果表示します。ただし、スクリプト内で式の結果を表示する場合は、出力コマンドを使用する必要があります。

[run-local コマンド](#) (P. 39) も参照してください。

4. AutoShell の対話型セッションを終了するには、以下のいずれかのコマンドを入力します。

```
quit  
exit
```

リモート対話型モードの使用

AutoShell クライアントがインストールされている管理対象のシステムとリモート対話型セッションを開始できます。AutoShell では、マネージャシェルおよびクライアントシェルと SSH サーバシステムとの通信で SSH を使用し、ログイン認証情報としてユーザ名とパスワードを使用します。この場合、利用できるコマンドは AutoShell コマンドではなく、リモートシステムで設定されている SSH のコマンドとなります。

リモート対話型モードを使用する方法

1. -h スイッチを使用してリモート ホスト名を指定します。

```
caaipaomautoshell.exe -h ascll
```

対話型コマンドプロンプトが表示されます。対話型コマンドプロンプトには、指定されたリモート ホスト名前が含まれます。プロンプトで入力したコマンドは、リモートシステムのクライアントシェルで実行されます。結果はマネージャシェルに返され、対話型セッションで表示されます。

注：スクリプトが対話型リモートセッション内で実行するように指定されている場合、そのスクリプトはリモートシステム上に存在する必要があります。

2. AutoShell の対話型セッションを終了するには、以下のいずれかのオプションを入力します。

```
quit  
exit
```

バッチ モード

AutoShell スクリプトは対話型モードから起動できます。ただし、通常は手動ではなく自動で実行します。

バッチ モードを使用する方法

`-f` スイッチと実行するスクリプトのパス名を指定して **AutoShell** を起動します。

```
caaipaomautoshell.exe -f hello.js
```

AutoShell はスクリプトを実行し、スクリプトの実行が完了すると終了します。スクリプトが完了すると、対話型モードが開始されます。スクリプトを処理した後、対話型モードに入るには、コマンド最後に `-i` を指定します（指定する順序が重要です）。

```
caaipaomautoshell.exe -f hello.js -i
```

注: これらの例では、`-U/-P` スイッチで認証情報を指定するか、`CAASUSER/CAASSPASS` 環境変数を設定しています。認証情報が指定されていない場合、**AutoShell** はスクリプト実行を開始する前に認証情報を要求します。

引数をスクリプトに渡すこともできます。

```
caaipaomautoshell.exe -f hello.js 1 abc "x y z"
```

スクリプトに `1`、`abc`、`"x y z"` が渡されます。これらの引数には、標準的な JavaScript メカニズムの引数配列を通してアクセスできます。

複数のスクリプトを指定するには、**-f** スイッチを複数回使用します。

```
caaipaomautoshell.exe -f hello.js -f world.js
```

これらのスクリプトは、指定された順序で実行されます。引数を指定する場合、以下のようにスクリプトごとに一連の引数を指定します。

```
caaipaomautoshell.exe -f hello.js 1 -f world.js 2
```

この場合、**hello.js** スクリプトには引数 **1**、**world.js** には引数 **2** が渡されます。

式を評価するには別の形式を使用します。スクリプトに単純な式を追加して **-f** を使用して実行する代わりに、**-e** スイッチを使用してコマンドライン上で指定できます。

```
caaipaomautoshell.exe -e Math.sqrt(2)
caaipaomautoshell.exe -e "for(i=1;i<11;i++)qout(i)"
```

リモートバッチ モード

リモートバッチ モードを備えたリモート システム上で **1** つまたは複数のスクリプトを実行できます。

リモートバッチ モードを使用する方法

スクリプト ファイルを指定する前に **-h** スイッチを使用して、ホスト名を指定します。たとえば、以下のコマンドを入力します。

```
caaipaomautoshell.exe -h ascl1 -f hello.js -f world.js
```

スクリプト ファイルはマネージャ システム上に存在し、実行するリモートクライアントに転送されます。

文字列化

文字列化は、文字のシーケンスを受け取り、JavaScript リテラル文字列に変換します。たとえば、AutoShell の `get-help` コマンドは、オプションのパラメータとしてフィルタ文字列を受け取ります。

```
get-help help
```

AutoShell は標準装備の JavaScript インタープリタを使用しているため、プロンプトまたはスクリプトファイルで指定されたコマンドは、有効な JavaScript 構文に変換されます。AutoShell のコマンド変換レイヤは、コマンドを JavaScript 関数呼び出しに変換します。

```
help( "help" );
```

JavaScript 内のリテラル文字列は、常に引用符で囲む必要があります。そのため、コマンド変換レイヤはコマンドで指定された文字列の前後に引用符を自動的に配置します。引用符が文字列の前後にある場合、文字列化は行われません。

JavaScript のリテラル文字列では、特殊文字をエスケープする必要があります。特に、引用区切りの文字列内の円記号と引用符には注意してください。エスケープするには、これらの特殊文字の前に円記号を付けます。AutoShell `dir` コマンド (1) の関数呼び出し (2) への変換について考えてみましょう。

```
(1) dir c:¥Program Files¥CA¥*. *  
(2) ca.aip.direct.directory("c:¥¥Program Files¥¥CA¥¥*. *" , "", false, false, false, false);
```

スペースが含まれている文字列や 1 つのオプションに複数の入力パラメータが指定されている文字列を処理する場合、引用符が必要な場合があります。スペースが含まれている文字列を引用符なしで指定した場合、すべての入力パラメータが照合されるまで、スペースは引数の区切りとして解釈されます。この動作を防ぐには、引用符を使用します。

引用符のないコマンドを使用する (1) 代わりに、円記号をエスケープしてパスを引用符で囲み、AutoShell が正しく引数を識別できるようにします (2)。

- (1) `copy c:¥Program Files¥CA¥*.¥ c:¥temp`
- (2) `copy "c:¥¥Program Files¥¥CA¥¥*.¥" c:¥temp`

注: コマンド変換レイヤはパスを文字列化しないため、文字列内の特殊文字は手動でエスケープする必要があります。

スペースを含む文字列をまとめて、特殊文字の自動エスケープを行うには、`_S()` マクロも使用できます。

```
copy _S(c:¥Program Files¥CA¥*.¥) c:¥temp
```

AutoShell コマンドを使用してタスクを自動化する場合、コマンドの一部で JavaScript 変数からの入力を受け取る必要がある場合が少なくありません。この場合は、文字列化が行われないようにします。文字列化が行われると、変数名がリテラル文字列に変換されます。また、コマンドは予期された結果を生成しません。式の文字列化を防ぐには、丸かっこの中に式を入れます。

```
var topic = "help";  
get-help (topic)
```

コンソール(標準出力)へのデータの書き込み

スクリプトで結果や診断メッセージを表示することはよくあることです。標準出力に任意の種類の出力を書き込むには、`?` および `??` コマンドを使用します。標準出力は、リダイレクトが指定されていなければコンソール画面が出力先となります。式の中から標準出力に出力を書き込むには、`qout()` および `qqout()` 関数を使用します。関数 `qout()` は `?` と同じ、`qqout()` は `??` と同じです。両方のコマンドと関数で、引数のリストを受け取ります。

`?, qout`

リストの各引数を文字列表現で標準出力に書き込みます。表示される各引数の間には、スペースが 1 個自動的に配置されます。出力は改行文字で終了します。

`??, qqout`

リストの各引数を文字列表現で標準出力に書き込みます。表示される各引数の間には、スペースが 1 個自動的に配置されます。

注: `?` および `??` 出力コマンドは行の先頭でのみ使用できます。

例

`?` コマンドは、指定された値に続けて 1 個の改行文字を出力します。

```
? "Hello World"
Hello World
```

`?` コマンドは、カンマ区切りの引数リストも受け取ります。表示される引数は、1 個のスペース区切りで出力されます。

```
? "Hello", "World"
Hello World
```

`??` コマンドは `?` のように動作しますが、出力の最後に改行を追加しません。一部の `??` コマンドは、1 行の出力を構築するスクリプト内で使用できます。

```
?? "Hello"
?? " "
?? "World"
```

どちらのコマンドも、任意のデータ型とその組み合わせを受け取ります。

```
? "今日の日付: ", new Date
? "2 の平方根: ", Math.sqrt(2)
```

1 から 10 までの数を表示します。

```
for(i=1;i<11;i++)qout(i);
```

キーボード(標準入力)からのデータの読み取り

スクリプトの実行時に入力が必要な場合、以下のいずれかのコマンドを使用します。

wait

キーが入力されるまで待機します。オプションで入力された文字を編集に格納します。

accept

Enter キーが押されるまで、キーボード(標準入力)から文字列を読み取ります。

例

任意のキーが押されるまで待機します。

```
wait "何かキーを押してください..."
```

キーが 1 つ押されるまで待機し、入力された文字を変数 **ch** に格納します。

```
wait "'何かキーを押してください'" to ch
```

Enter キーが押されるまで文字列を読み取り、変数 **str** に格納します。

```
accept to str
```

「金額を入力してください:」というメッセージを表示し、Enter キーが押されるまで文字列を読み取り、変数 **str** に格納します。

```
accept "金額を入力してください: " to str
```

オペレーティング システムへのアクセス

管理シェルではよく、オペレーティング システムのコマンドやほかのネイティブ ユーティリティを呼び出してシステム関連タスクを実行します。対話型 **AutoShell** プロンプトまたはスクリプトの内部から、先頭に感嘆符を付けて **OS** コマンドを実行します。感嘆符は常に入力行の最初の非空白文字である必要があります。

デフォルトでは、AutoShell は、OS コマンドの出力を AutoShell コンソールに直接書き込みます。ただし、AutoShell コンソールに書き込まれる出力に AutoShell が直接アクセスして処理することはできません。子プロセスの出力を処理するアプリケーションでは通常、出力をファイルにリダイレクトし、子プロセスが終了してからファイルを読み取ります。AutoShell では、JavaScript オブジェクトに子プロセスの出力を直接割り当てることができます。以下の例を参照してください。

OS アクセスを使用して、AutoShell から任意のアプリケーションを起動することができます。OS コマンドを起動する場合、AutoShell は常にコマンドが完了するまで待機してから、コマンドプロンプトに戻るか、スクリプトの実行を続行します。

例

現在のディレクトリにあるファイルをリスト表示し、稼働している OS のバージョンを出力して、メモ帳を起動します。これらのサンプル コマンドは AutoShell との対話なしで実行されます。以下のコマンドの最初のコマンドでは、Autoshell の組み込みコマンドの `dir` ではなく、オペレーティング システムの `dir` コマンドが使用されます。AutoShell は、メモ帳アプリケーションが終了するまで待機します。

```
! dir
! ver
! notepad
```

アンパサンド `&` を使用してコマンドを連結し、単一の子プロセスで実行します。

```
! dir >t.txt & type t.txt
```

OSRedirect クラスの使用

子コマンドを起動する際、AutoShell では、事前定義されている OSRedirect クラスを使用して出力をリダイレクトします。出力をリダイレクトして子プロセスのリターンコードを取得するには、OSRedirect オブジェクトを作成し、コマンドを呼び出す際にオプションの `-output` 節を使用して渡します。

```
out = new OSRedirect();
! dir -output out
```

AutoShell コンソールでこのコマンドを対話形式で実行した場合、情報はコンソールに表示されません。指定されたオブジェクトに出力はリダイレクトされます。コマンド完了後、指定したオブジェクトのメソッドを呼び出し、コマンド実行に関する情報を取得します。

```
? "コマンドの実行時にエラーが発生しました: ", out.errorOccurred()
? "コマンドの完了状況: ", out.hasCompleted()
? "コマンドで生成された出力: ", out.output()
? "コマンドのリターン コード (エラー レベル) : ", out.result()
```

メソッドの `output()` と `result()` で必要な情報を取得できます。`output()` メソッドは、完全なディレクトリ リストを 1 つの文字列で返します。標準的な JavaScript の文字列関数を使用してこの出力をさらに処理できます。たとえば、行を解析して配列に格納し、一行ずつ表示します。

```
lines=out.output().split("\n");
for(i=0;i<lines.length;i++)out(lines[i]);
```

`dir` オペレーティング システム コマンドを ! AutoShell コマンドで起動する場合、! コマンドを発行する前に `OSRedirect` オブジェクトを割り当てます。この手順を省略するために、!! コマンドを使用して `OS` コマンドを起動することもできます。!! コマンドは `OSRedirect` オブジェクトを作成し、指定された変数に `OSRedirect` オブジェクトが割り当てられていない場合は、この `OSRedirect` オブジェクトを割り当てます。

以下のコマンドは、`OSRedirect` オブジェクトが明示的に割り当てられていなくても、現在のディレクトリ リストを変数 `out` に出力します。

```
!! dir -output out
```

!! コマンドを オプションの `-output` 節を指定せずに発行すると、ディレクトリ情報は変数 `$$stdout` に自動的に書き込まれます。

```
!! dir
? $$stdout.output()
```

スクリプトの実行

このセクションでは、ローカル ホスト、および複数のリモート ホスト上でのスクリプトの実行について説明します。

関連項目

[run-local Command の使用](#) (P. 39)

[run-remote コマンドの使用](#) (P. 41)

run-local Command の使用

run-local コマンドを使用して、スクリプトの式を評価したり、スクリプトファイルをローカルに実行したりします。**run-local** コマンドでは、スクリプト コードにパラメータを渡すことができます。

式を評価するには、たとえば、以下のような式を指定します。

```
run-local 1+1
```

式のパラメータは文字列化されます。したがって、引用符を使用した式と引用符を使用していない式を指定できます。

スクリプトを実行するには、オプションの **-file** 節でスクリプトのパス名を指定します。

```
run-local -file scripts\primes.js
```

実行中のスクリプトで別のスクリプトを起動するただ 1 つの方法は、**-file** オプションを指定して **run-local** コマンドを実行することです。JavaScript ファイルのパス名を指定してスクリプトを起動した場合にのみ、対話型プロンプトで機能します。パス名の指定は、スクリプトファイル内ではサポートされていません。スクリプトからスクリプトを実行する場合は、以下の例を参考にしてください。

```
// スクリプト ファイル hello.js の開始
function hello()
{
    ? "Hello World!"
}
// スクリプト ファイルの終了
```

このスクリプトを起動するには、以下のコマンドを使用します。

```
run-local -file hello.js
```

このコマンドは、現在のスクリプト コンテキストで使える `hello()` 関数を作成しますが、実行しません。したがって、コンソールには何も表示されません。`run-local` コマンドが戻ってきたら、新たに定義した関数を呼び出しスクリプトで起動し、予期されるあいさつを出力できます。

```
hello();
```

スクリプト ファイルを呼び出す際に実行できるのは、`run-local` コマンドを実行するとき周囲のスコープ（関数など）に入っていないコードのみです。以下の例にスクリプトを変更すると、`run-local` コマンドが戻る前にあいさつが出力されます。

```
// スクリプト ファイル hello2.js の開始
? "Hello World!"
// スクリプト ファイルの終了
```

呼び出すことができ、さらにコードも実行できる関数を作成するには、スクリプト ファイル内で関数自身を呼び出します。

```
// スクリプト ファイル hello3.js の開始
hello();
function hello()
{
    ? "Hello World!"
}
// スクリプト ファイルの終了
```

スクリプト ファイル内の実行可能コードを、関数のスコープ外の任意の場所で直接表示できます。前の例のように最初の関数宣言の前に実行可能コードを配置することは、必須ではありません。

オプションの `-with` 節を使用して、`run-local` コマンドで実行する式やスクリプトにパラメータを渡すことができます。パラメータには、標準的な JavaScript 引数配列を通して式やスクリプトでアクセスできます。

```
// スクリプト ファイル args.js の開始
var i, l;
l = arguments.length;
for(i=0; i < l; i++)
{
    ? arguments[i]
}
// スクリプト ファイルの終了
```


このスクリプトは、渡された引数をすべて出力します。たとえば、以下のコマンドは「1」、「abc」、および現在の日付と時間を出力します。

```
run-local -file args.js -with 1, "abc", new Date()
```

関連項目

[オペレーティング システムへのアクセス](#) (P. 36)

[OSRedirect クラスの使用](#) (P. 37)

[run-local コマンド -- ローカル システム上でスクリプトを実行 \(Funclet\)](#) (P. 108)

run-remote コマンドの使用

run-remote コマンドは、1 つ以上のリモート **AutoShell** クライアントシステム上で式またはスクリプト ファイルを実行します。リモート実行を行うには、リモート ターゲット ノードに **AutoShell** クライアントをインストールして設定する必要があります。

run-remote は **run-local** コマンドと同じ方法で、式、スクリプト、引数を指定できます。スクリプト ファイルはマネージャ システム上に保存し、実行するターゲット システムに転送する必要があります。さらに、リモート クライアントを節で指定する必要があります。**run-remote** コマンドを最も単純に起動する方法は、サーバ名を指定することです。

```
run-remote "1+2" on "ascl1"
```

注: ホスト名の文字列化はサポートされていません。

リモート実行で複数のターゲットを指定するには、リストを使用します。

```
run-remote "1+2" on "ascl1", "ascl2"
```

このようにして **run-remote** を使用した場合、実行したリモート ジョブに関する情報は取得されません。ジョブは成功または失敗しますが、結果は示されません。リモート コマンドは非同期で実行されます。つまり、リモート コマンドがまだ実行中であっても、**run-remote** コマンドは戻ってきます。

リモート コマンドが処理されるかエラーが発生するまで待機するには、`-wait` オプションを付けて `run-remote` を呼び出します。

```
run-remote "1+2" on "ascl1" -wait
```

リモート実行をより詳細に制御するために、事前定義済みの `RemoteTarget` `AutoShell` クラスが用意されています。

関連項目

[RemoteTarget クラスの使用](#) (P. 42)

[run-remote コマンド -- リモート システム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

RemoteTarget クラスの使用

以下の方法で、`RemoteTarget` クラスと共に `run-remote` コマンドを使用します。

- `run-remote` コマンドから返された `RemoteTarget` オブジェクトを処理します。
- `RemoteTarget` オブジェクトを作成し、`run-remote` コマンドに渡します。

`run-remote` コマンドは、常に `RemoteTarget` オブジェクトの配列を返します。この配列の長さは、節で指定したリモート ターゲットの数と同じです。配列要素は、同じ位置のリモート ターゲットにそのまま対応します。

```
arrRT = run-remote -file hello.js on "ascl1", "ascl2"
```

`run-remote` コマンドは、2 つの `RemoteTarget` オブジェクトから成る配列を返します。

- `arrRT[0]` には、リモート ホスト `ascl1` でのスクリプト `hello.js` の実行に対応する `RemoteTarget` オブジェクトが含まれます。
- `arrRT[1]` は `ascl2` での同じスクリプトの実行に対応します。

RemoteTarget オブジェクトは、スクリプトの実行プロセス、出力、結果に関するステータス情報を利用可能にします。以下の **RemoteTarget** メソッドを使用して、このステータス情報を取得します。

hasCompleted()

コマンドの処理が成功 (**true**) したか失敗 (**false**) したかを示します。オプションの **-wait** スイッチを指定せずに **run-remote** を起動すると、コマンドは非同期で実行されます。コマンドが完了したかどうか、および結果を利用できるかどうかを確認するには、このメソッドを呼び出します。コマンドが正常に完了した場合、メソッド **hasCompleted()** は **true** を返します。その後の処理の妨げとなるエラーが発生した場合は、**false** を返します。

戻り値： **true** または **false**

errorOccurred()

コマンドが正常に終了し、出力と結果を確認できる場合は **false** を返し、エラーが発生した場合は **true** を返します。

戻り値： **true** または **false**

output()

指定されたスクリプトを対応するリモートシステム上で実行し、生成された出力をキャプチャして格納した文字列を返します。

戻り値： 文字列

result()

指定されたスクリプトを対応するリモートシステム上で実行し、XML にエンコードされた値を返します。

戻り値： *XML*

RemoteTarget クラスのその他のメソッドについては、「[AutoShell クラス \(P. 159\)](#)」を参照してください。

例

前に紹介した「Hello World!」を出力する `hello.js` スクリプトで、両方の `RemoteTarget` オブジェクトの `output()` メソッドを使用してこの文字列を返すとします。より大きな視点で見ると、以下のスクリプトはこの種の処理を行います。

```
var t;
var arrRT;
var i, l;

arrRT = run-remote -file hello.js on "ascl1", "ascl2"
l = arrRT.length;

// 完了を待機。上のコマンドで -wait 節を使用することもできますが、
// 以下の方法では独自にタイムアウトを設定して退出できます。
t = 0;

while(t<60)
{
    for(i=0; i<l; i++)
    {
        if(!arrRT[i].hasCompleted())
            break;
    }
    // 呼び出しがすべて完了したら、ループを中断
    if(i==l)
        break;
    sleep(500);
    t++;
}
```

```

for(i=0; i<l; i++)
{
    if(arrRT[i].hasCompleted())
    {
        if(arrRT[i].errorOccurred())
        {
            ? "次のホストでエラーが発生しました: ", arrRT[i].getHostName()
        }
        else
        {
            ? arrRT[i].getHostName()," から返された出力: "
            ? arrRT[i].output()
        }
    }
    else
    {
        arrRT[i].abort();
    }
}

```

OS コマンドの呼び出しとリモート実行を組み合わせると、リモートシステム上の OS サービスやデータにアクセスできます。たとえば、以下のコードは、サーバ **ascl1** 上で稼働しているオペレーティングシステムの OS バージョン文字列を問い合わせ返します。

```

arrRT = run-remote ! ver on "ascl1" -wait
? arrRT[0].output()

```

前の例では、**AutoShell** はリモート コマンドの出力のみを表示します。また、リモートで処理される式やスクリプトで値を返すこともできます。

```

arrRT = run-remote 2*21 on "ascl1" -wait

```

この場合、式「**2*21**」がサーバ **ascl1** に送信され、リモートで計算されます。コマンドが完了すると、対応する **RemoteTarget** オブジェクトの **result()** メソッドで結果を使用できます。

result() メソッドは値を直接返すのではなく、XML 表現で値を返します。

```

? arrRT[0].result()
<value type="number">42</value>

```

この XML 文字列を標準の JavaScript 値に変換するには、**get-remoteResult()** を使用します。

```

v = get-remoteResult(arrRT[0])
? typeof v, v

```

また、AutoShell のシリアル化は、配列やオブジェクトなどの複雑な型を返すリモート スクリプトにも適用されます。シリアル化により、構造データを失わずに、リモート サーバ間で大量の構造化されたデータを転送できます。以下の例は、3 つの要素から成る配列を返すリモート実行を示しています。

```
arrRT = run-remote [1,"abc",new Date()] on "ascl1" -wait
v = get-remoteResult(arrRT[0])
? typeof v, v
```

RemoteTarget オブジェクトを使用して作業するには、実際のリモート コマンドを呼び出す前に **RemoteTarget** オブジェクトを作成する方法もあります。**RemoteTarget** オブジェクトでは、オブジェクトを作成するときにターゲット ホストの名前を指定して構築します。

```
var rt1 = new RemoteTarget("ascl1");
var rt2 = new RemoteTarget("ascl2");
```

複数のコマンドやスクリプトを同じリモート ホスト上で実行する場合は、明示的に **RemoteTarget** オブジェクトを作成することをお勧めします。こうすることで、**run-remote** が実行されるたびに同じ **RemoteTarget** オブジェクトを再利用できます。明示的に作成しない場合、**run-remote** は呼び出されるたびに **RemoteTarget** オブジェクトを作成します。

run-remote で **RemoteTarget** オブジェクトを使用するには、ホスト名の代わりに **RemoteTarget** オブジェクトを指定します。

```
run-remote -file hello.js on rt1, rt2 -wait
? rt1.output()
? rt2.output()
run-remote Math.log(Math.E) on rt1, rt2
? get-remoteResult(rt1)
? get-remoteResult(rt2)
```

RemoteTarget オブジェクトは、**run-remote** から返された配列を格納しなくても使用できます。配列には、最初にコマンドに渡した **RemoteTarget** オブジェクトへの参照のみが格納されます。したがって、明示的に作成した **RemoteTarget** オブジェクトに直接アクセスして実行結果を問い合わせることができます。

リモート ターゲットは、前の例のようなリストや配列で指定できます。

```
arrRT = [new RemoteTarget("ascl1"), new RemoteTarget("ascl2")];  
run-remote -file hello.js on arrRT
```

RemoteTarget オブジェクトは、後続の **run-remote** コマンド呼び出しで再利用できます。また、リモート クライアント セッションの作成を管理する方法を提供します。

AutoShell で **run-remote** を実行すると、指定したリモート システムに **SSH** を介して接続します。デフォルトでは、リモート システム上の **SSH** サーバは、クライアント **AutoShell** の新規インスタンスを開始します。新しく作成された **AutoShell** プロセスは渡されたスクリプトを評価し、評価が完了するとプロセスは破棄されます。その後、複数のコマンドを呼び出すと、リモート システム上で複数のクライアント **AutoShell** プロセスが作成および破棄されます。また、スクリプトを実行する際に作成されたコンテキスト情報はすべて失われるため、後のリモート スクリプトで利用することはできません。

RemoteTarget クラスは、リモート ホスト上でのクライアント **AutoShell** セッション作成を管理する、使いやすいメカニズムを実装しています。この **RemoteTarget** オブジェクト用にリモート ホスト上でクライアント **AutoShell** の永続セッションを作成するには、**RemoteTarget** オブジェクトの **createRemoteContext()** を呼び出します。

```
var rt = new RemoteTarget("ascl1");  
rt.createRemoteContext()
```

後続の **run-remote** 呼び出しでは、リモート ホスト上にクライアント **AutoShell** の一時セッションは割り当てられず、**createRemoteContext()** で作成した永続セッションが使用されます。また、スクリプトを実行する際に作成されたコンテキスト情報は、後で実行されるリモート スクリプトですべて利用できます。**RemoteTarget** オブジェクトを回収すると、リモート ホスト上のクライアント **AutoShell** の永続セッションは自動的に破棄されます。このセッションは、対応するオブジェクトの **destroyRemoteContext()** を呼び出して明示的に破棄することもできます。

`createRemoteContext()` を使用する効果は、以下の例で簡単に確認できます。

```
var rt = new RemoteTarget("ascl1");
rt.createRemoteContext()
run-remote X=5 on rt -wait
run-remote X on rt -wait
? get-remoteResult(rt)
rt.destroyRemoteContext()
run-remote X=5 on rt -wait
run-remote X on rt -wait
? get-remoteResult(rt)
```

最初の呼び出しで作成された変数 **X** がクライアント **AutoShell** の永続セッションにまだ存在するため、2 番目のコマンド呼び出しでは **5** が返されます。リモートクライアントセッションが破棄された、残りの 2 つの呼び出しでは、2 つの一時クライアント **AutoShell** の一時セッションが作成されます。変数 **X** は別のセッションで作成され、すでに破棄されているため、2 番目のコマンドでは空の結果が返されます。

関連項目

[RemoteTarget クラス](#) (P. 173)

AutoShell コア リファレンス

このセクションでは、AutoShell コア固有のコマンド、関数、クラスについて説明します。プラットフォーム固有の拡張期の属するコマンドについては、「[AutoShell ロード モジュールのコマンド リファレンス \(P. 191\)](#)」を参照してください。

AutoShell コマンドは `cmdlet` と `funclet` に分けられます。

Cmdlet

行の先頭を空白以外の文字で始める必要があるコマンド定義を指定します。この制限があるため、`cmdlet` は、より幅広い JavaScript の式の一部としてではなく、スタンドアロンで使われます。特に、右辺値（代入演算子の右側のオペランド）として使用することはできません。？は、AutoShell `cmdlet` の例です。

Funclet

オプションの節や文字列化などを備えた、コマンドに似た冗長な構文を持ちます。また、値を返すこともできます。多くの場合、`funclet` は `cmdlet` のように、つまりスタンドアロンとして 1 行で使われます。ただし、返される値は、より幅広い式の一部として処理できます。

カテゴリ別コマンド、関数、クラス

AutoShell は、以下の CA Virtual Assurance 固有のコマンド(cmdlet、funclet)、関数、クラスを提供します。これらのコマンドを使用して適切な AutoShell アプリケーションを作成し、仮想環境を管理します。

Cmdlet

!、!!、?、??、accept、arrdump、cat、cd、DBG_PROMPT、DBG_PROMPT_LINE、exit、external、get-autoShellClassInfo、get-webServiceInfo、help、mkdir、objdump、pwd、quit、set alternate to、set console、set result disp、start-java、type、wait、weak external

Funclet

add、ASSERT、associate、chdir、clone、copy、cp、create、custom、del、dir、get-help、get-remoteResult、install、ls、modify、move、mv、new-SSHSession、new-ZONESession、new-comObject、new-webService、push-client、push-winRemote、push-winRemote、PW_GET、query-service、reboot、rebuild、rem-client、remove、ren、rename、reset、rm、rmdir、run-SSHCommand、run-SSHShell、run-ZONECommand、run-client、run-local、run-remote、run-winRemote、show、shutdown、start、start-service、status、stop、stop-service、uninstall

関数

base64Decode、base64Encode、curDir、delete、disable、enable、gete、memoRead、memoWrit、platform、pute、qout、qqout、regCreateKey、regCreateSubkeys、regDeleteKey、regDeleteVal、regGetKeyValues、regGetSubKey、regGetVal、regIsKey、regIsVal、reSetKeyValues、regSetVal、setProcExitCode、shellType、typeName

クラス

OSRedirect、RemoteTarget

AutoShell のコマンド、関数、クラスは、以下の基準で分類されます。

配列

arrdump

AutoShell の展開

exit、quit、push-client、rem-client、run-client、setProcExitCode、shellType

クラスとオブジェクト

new-comObject、get-autoShellClassInfo、get-remoteResult、objdump、OSRedirect、RemoteTarget、run-local、run-remote、typeName

コマンド実行

!、!!、get-remoteResult、OSRedirect、push-winRemote、RemoteTarget、run-local、run-remote、run-SSHCommand、run-SSHShell、run-winRemote、run-ZONECommand

データ型

typeName

暗号化

base64Decode、base64Encode

環境変数

gete、pute

ファイルとディレクトリ

cat、cd、chdir、copy、cp、curDir、del、dir、ls、mkdir、mv、pwd、ren、rm、rmdir、type

ヘルプ

get-help、help

入力と出力

?, ??、accept、DBG_PROMPT、DBG_PROMPT_LINE、PW_GET、qout、qqout、set alternate to、set console、set result disp、wait

数式

ASSERT

その他

external、start-java、weak external

プラットフォーム

push-winRemote、run-LPARCommand、run-winRemote、run-ZONECommand、platform

レジストリ

regCreateKey、regCreateSubkeys、regDeleteKey、regDeleteVal、
regGetKeyValues、regGetSubKey、regGetVal、regIsKey、regIsVal、
regSetKeyValues、regSetVal

リモート アクセス

get-remoteResult、push-winRemote、RemoteTarget、run-remote、
run-SSHCommand、run-SSHShell、run-winRemote

サービス

get-webServiceInfo、new-webService、query-service、start-service、
stop-service

セッション

new-SSHSession、new-ZONESession

Solaris ゾーン

associate、clone、create、custom、delete、disable、enable、install、modify、
move、reboot、rename、show、start、status、stop、uninstall

文字列

accept、PW_GET、memoRead、memoWrit、typeName

AutoShell の一般コマンド

このセクションでは、一般的な AutoShell コア コマンド (cmdlet、funclet) の詳細についてアルファベット順に説明します。これらの一般コマンドは、LPAR または Solaris ゾーン プラットフォームに固有のものではありません。AutoShell では、対応するタスクの起動を簡略化する一連のコマンドが定義されています。

以下の標準的な JavaScript の関数呼び出しは、C:¥ の詳細なディレクトリ リストを取得します。

```
ca.aip.direct.directory("C:¥*. *" , "",false,false,false,true)
```

対応する AutoShell コマンドは、以下のようになります。

```
dir C:¥*. * -l
```

例

コンソール内の現在のディレクトリのコンテンツをリスト表示します。

```
dir *.*
```

ディレクトリ リストを配列に割り当てます。

```
arr = dir *.* -retval
```

!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 (Cmdlet)

オペレーティング システム コマンドを実行するか、子プロセスとしてアプリケーションを開始します。ユーザ指定の **OSRedirect** オブジェクトまたはデフォルトのシステム変数 (**\$\$stdout**) で、子プロセスの出力を自動的にキャプチャします。

コマンドの構文は、以下のとおりです。

```
!! cmd [-output osRedirect]
```

cmd

実行するコマンドまたは起動するアプリケーション。 **&&** シーケンスを使用して複数のコマンドを連結できます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

osRedirect

(オプション) 子プロセスの出力を受け取り、指定された変数にキャプチャする、既存の **OSRedirect** オブジェクト。

デフォルト: **\$\$stdout**

例

Windows OS に関する DOS メモリ情報を取得し、呼び出し前に初期化されていない変数 **out** に出力をリダイレクトします。

```
!! mem -output out
// 一行ずつ出力
aLines = out.output().split(String.fromCharCode(10));
arrdump aLines
```

DNS ルックアップを実行し、出力を自動的にキャプチャして **\$\$stdout** に送ります。

```
!! nslookup ca.com
? $$stdout.output()
```

関連項目:

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)
[OSRedirect クラス](#) (P. 160)

! コマンド -- コマンドまたは子プロセスの呼び出し (Cmdlet)

オペレーティング システム コマンドを実行するか、子プロセスとしてアプリケーションを開始します。子プロセスによって生成された出力は、デフォルトでは **AutoShell** コンソールに書き込まれます。オプションで、**AutoShell OSRedirect** オブジェクトに出力をリダイレクトします。

コマンドの構文は、以下のとおりです。

```
! cmd [-output osRedirect]
```

cmd

実行するコマンドまたは起動するアプリケーションを指定します。
&& シーケンスを使用して複数のコマンドを連結できます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式を丸かっこで囲み、式が自動的に引用されないようにしてください。

osRedirect

(オプション) 子プロセスの出力を受け取る、既存の **OSRedirect** オブジェクト。

デフォルト: ""

例

Windows OS のバージョン文字列を表示します。

```
! ver
```

Windows OS のバージョン文字列と現在のディレクトリを表示します。

```
! ver && cd
```

ネットワーク上のシステムに対して **PING** を実行し、出力をキャプチャします。

```
out=new OSRedirect();
! ping 192.168.0.100 -output out
if(out.result()==0)
{
    ? out.output();
}
else
{
    ? "PING 失敗"
}
```

関連項目

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect クラス](#) (P. 160)

?? コマンド -- リストを標準出力に書き込む (Cmdlet)

リストの各引数を文字列表現で標準出力に書き込みます。表示される各引数の間には、スペースが 1 個自動的に配置されます。?、??、qout()、または qqout() を使用して続きの出力を呼び出し、元の出力の直後に出力を書き込みます。そのため、通常は ?? または qqout() を複数回呼び出して、出力行を構成します。

コマンドの構文は、以下のとおりです。

?? [x]

x

(オプション) 出力する引数のリスト。

例

スペース区切りで 3 つの列見出しを出力します。

```
?? "列 1"
```

```
?? " "
```

```
?? "列 2"
```

```
?? " "
```

```
? "列 3"
```

3 番目のラベルは ? コマンドを使用して出力し、次の行の先頭にカーソルを移動させます。

関連項目

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect クラス](#) (P. 160)

? コマンド -- 出力を標準出力に書き込む (Cmdlet)

このコマンド (`cmdlet`) は、リストの各引数を文字列表現で標準出力に書き込みます。各引数の間には、スペースが 1 個自動的に配置されます。`set console` および `set alternate` コマンドを使用すると、出力のリダイレクトや抑制を行うことができます。出力は改行文字で終了します。

コマンドの構文は、以下のとおりです。

```
? [x]
```

```
x
```

(オプション) 出力する引数のリスト。リストが空の場合、改行文字のみが出力されます。

例

Print "Hello World!":

```
? "Hello World!"
```

数のリストを出力します。

```
? 1,2,3
```

空行を出力します。

```
?
```

1 から 10 までの数字を出力します。? コマンドは行の先頭でのみ使用できます。式の中からテキストを出力するには、対応する `qout()` 関数を使用する必要があります。

```
for(i=1;i<11;i++) qout(i);
```

関連項目

[?? コマンド -- リストを標準出力に書き込む \(Cmdlet\)](#) (P. 56)

[qqout -- stdout に出力を書き込む \(関数\)](#) (P. 141)

[qout -- 出力に続いて改行を stdout に書き込む \(関数\)](#) (P. 140)

[set alternate to コマンド -- 別の出力ファイルを設定する \(Cmdlet\)](#) (P. 125)

[set console コマンド -- コンソールへの出力を非表示にする \(Cmdlet\)](#) (P. 126)

accept コマンド -- 標準入力から文字列を読み取り、変数に割り当てる (Cmdlet)

オプションで入力を要求して **Enter** キーが押されるまで文字を読み取り、改行なしで文字列を変数に割り当てます。

コマンドの構文は、以下のとおりです。

```
accept [prompt] to var [-hidden]
```

prompt

(オプション) 表示する文字列を指定します。

var

読み取った一連の文字列を割り当てる変数の名前。

-hidden

(オプション) パスワードを入力するために入力文字列をマスクします。

例

文字列を読み取り、変数 **name** に割り当てます。

```
accept "名前を入力してください: " to name
```

パスワードを入力するために、入力をマスクします。

```
accept "パスワードを入力してください: " to password -hidden
```

関連項目

[wait コマンド -- キーが押されるのを待機する \(Cmdlet\)](#) (P. 131)

[PW GET コマンド -- 非表示データの入力 \(Funclet\)](#) (P. 100)

arrdump コマンド -- 配列の表示 (Cmdlet)

このコマンドは、一次元配列の要素をすべてリスト表示します。

コマンドの構文は、以下のとおりです。

```
arrdump arr
```

arr

表示する配列を定義します。

例

配列を表示します。

```
a = [1,2,"a", true, new Date()];  
arrdump a
```

関連項目

[objdump コマンド -- オブジェクトの列挙可能なプロパティの表示 \(Cmdlet\)](#) (P. 93)

ASSERT コマンド -- 数式の比較(Funclet)

このコマンド (funclet) は、2 つの数式から成るステートメントまたはアサーションと操作が **true** と **false** のいずれになるかを確認します。数式は、以下の要素で構成されます。

- 関数
- オペレーション
- 小数
- 10 進数、8 進数、または 16 進数形式の数値

ステートメントまたはアサーションが **false** の場合、AutoShell はカスタマイズ可能なエラー メッセージを生成します。

8 進数形式で数値を書き込むには、**0123** のように値の前に **0** を付けます。16 進数形式で数値を書き込むには、**0x123** のように値の前に **0x** を付けます。

コマンドの構文は、以下のとおりです。

```
ASSERT x relation v [-msg msg]
```

x

式 **v** と比較する数式を定義します。

relation

比較に使用する関係を指定します。オプションは、以下のとおりです。

- eq (等しい)
- ne (等しくない)
- gt (より大きい)
- lt (より少ない)
- ge (以上)
- le (以下)

v

数式を定義します。

-msg *msg*

(オプション) 比較が **false** を返す場合に、デフォルトのエラー メッセージではなく代替エラー メッセージを生成します。

デフォルト: null

例

ステートメント `6>7` が誤っていることを示します。

```
ASSERT 6 gt 7 -msg "これは正しくありません。"  
これは正しくありません。
```

8 進数値 `0123` が `0x9e` と等しいかどうかを確認します。

```
ASSERT 0123 eq 0x9e  
Failed: 0123==158, actual=83
```

PI が `4.5*sin(PI/2)` より大きいかどうかを確認します。

```
ASSERT Math.PI gt 4.5*Math.sin(Math.PI/2)  
Failed: Math.PI>4.5, actual=3.141592653589793
```

cat、type コマンド -- テキスト ファイルの表示 (Cmdlet)

これらのコマンド (cmdlet) は、標準出力ストリームに 1 つ以上のテキスト ファイルのコンテンツを書き込みます。複数のファイルを連結するには、`set alternate` と共に使用します。

コマンドの構文は以下のとおりです。

```
{cat|type} fileList
```

fileList

表示するファイル名 (絶対パスまたは相対パス) のカンマ区切りのリストを定義します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

例

Windows ブート オプションを表示します。

```
cat c:\boot.ini
```

2 つのファイルを連結します。

```
set alternate to output.txt  
cat input1.txt, input2.txt  
set alternate to
```

関連項目:

[memoRead -- 文字列にテキスト ファイルを読み込む \(関数\)](#) (P. 136)

cd、chdir コマンド -- ディレクトリの変更 (Cmdlet、Funclet)

これらのコマンドは、ローカルシステム上の現在のディレクトリを変更します。**p**を指定すると、新規ディレクトリを定義します。**p**が相対パスの場合、コマンドはディレクトリのプレフィックスとして現在のパスを使用します。

コマンドの構文は、以下のとおりです。

```
cd [p] (Cmdlet)
chdir [p] (Funclet)
```

p

(オプション) 変更するディレクトリを定義します。

デフォルト : ""

例

ルートディレクトリに移動します。

```
cd /
```

ディレクトリ **c:/test** に移動します。

```
cd c:/test
```

親ディレクトリに移動します。

```
cd ..
```

関連項目

[dir、ls コマンド -- ファイルおよびディレクトリ情報の取得 \(Funclet\)](#) (P. 71)

[mkdir コマンド -- ディレクトリの作成 \(Cmdlet\)](#) (P. 81)

[pwd コマンド -- 作業ディレクトリ名の出力 \(Cmdlet\)](#) (P. 101)

[curDir -- 現在のディレクトリを取得する \(関数\)](#) (P. 134)

copy、cp コマンド -- ファイルとディレクトリのコピー (Funclet)

これらのコマンド (funclet) は、ファイルまたはディレクトリを別の場所にコピーします。コピー元またはコピー先の場所のどちらかはローカルサーバ上に存在する必要があります。2つのリモートシステム間での操作はサポートされていません。リモートホストを指定するには、パス名の前にホスト名とその後に2個のコロン (::) を指定します。操作はSFTPによって実行されます。したがって、リモートシステムにクライアントAutoShellがインストールされている必要はありません。SFTPアクセスが可能なSSHサーバがあれば十分です。set alternate コマンドを使用して、ファイルに出力を書き込むことができます。

コマンドの構文は、以下のとおりです。

```
{copy|cp} src trg [-silent] [-retval] [-R] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

src

コピーするディレクトリまたはファイルを指定する検索マスクを定義します。Windowsの区切り文字には、円記号 (¥) またはスラッシュ (/) を使用できます。UNIX系のシステムでは、スラッシュを使用する必要があります。プラットフォーム間のスクリプトの互換性のために、パスの指定にはスラッシュのみを使用してください。

検索マスクは、DOSのワイルドカード検索と互換性のある、簡略化された正規表現構文を使用します。

- * は、ゼロ個以上の任意の文字に一致します。
- ? は、任意の1文字に一致します。
- [abc] は、a、b または c のいずれか1文字に一致します。
- [a-f] は、a ~ f のいずれかに一致します。
- [^a-f] は、a ~ f 以外のいずれかに一致します。
- [-_] は、- または _ に一致します。[^_] はそのほかの任意の文字に一致します。開始角かっこ、または ^ の直後に指定されている場合、ダッシュ「-」は特殊文字ではありません。
- [a^] は、a または a^ に一致します。（開始角かっこの直後に指定されていない場合、キャレット「^」は特殊文字ではありません）。
- ¥*, ¥?, ¥[, ¥], ¥¥ は、それぞれ1個の *, ?, [,], ¥ に一致します。

他の文字はすべて特殊文字ではありません。それぞれの文字に一致します。

trg

コピー先のパスを定義します。

-silent

(オプション) アクティブな出力ストリームへの出力を抑制します。

-retval

(オプション) 戻り値を表す配列を作成します。配列には新規ファイル名が格納されます。要素はそれぞれ、1つの宛先ファイルの完全修飾名を表す文字列から構成されます。

-R

(オプション) ディレクトリを再帰的にコピーします。

-user *username*

(オプション) リモートノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$User

-pass *password*

(オプション) リモートノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$Pass

-key *key*

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-port *portnumber*

(オプション) ターゲット システムに接続するポートです。

デフォルト : SSH の標準ポート 22。

例

ファイル **c:/test/test1.log** をディレクトリ **c:/test1** にコピーします。

```
copy c:/test/test1.log c:/test1
```

ディレクトリ **c:/test** 全体をディレクトリ **c:/test1** にコピーします。

```
cp c:/test c:/test1 -R
```

ディレクトリ **c:/test** 全体をリモート ホスト **remotesys** のディレクトリ **c:/test1** にコピーします。

```
cp c:/test1/ remotesys::c:/test1/ -R
```

関連項目

[mv、ren コマンド -- ファイルとディレクトリの移動 \(Funclet\)](#) (P. 84)

DBG_PROMPT コマンド -- 確認のためにスクリプトの中断を実行 (Cmdlet)

DBG_PROMPT コマンド (cmdlet) は、スクリプト実行を中断し、現在のスクリプト コンテキストで式を評価する対話型プロンプトを表示します。通常は現在の変数の値を確認するために使用され、空の入力行が入力されるまで、デバッグ プロンプトが繰り返し表示されます。

コマンドの構文は、以下のとおりです。

DBG_PROMPT [*prompt*]

prompt

表示するオプションのプロンプトの文字列を指定します。複数のプロンプトを使用する場合、プロンプトを使用して、中断した場所を示すこともできます。

デフォルト : "Dbg> "

例

ループの最後の繰り返しで中断します。

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT
        ? i
}
```

カスタム プロンプトを使用して中断します。

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT "i==10> "
        ? i
}
```

関連項目

[DBG_PROMPT_LINE コマンド -- 確認のために行番号を表示してスクリプトの中断を実行 \(Cmdlet\)](#) (P. 67)

DBG_PROMPT_LINE コマンド -- 確認のために行番号を表示してスクリプトの中断を実行 (Cmdlet)

DBG_PROMPT_LINE コマンド (cmdlet) は、スクリプト実行を中断し、現在のスクリプトコンテキストで式を評価する対話型プロンプトを表示します。プロンプトには現在の行番号が自動的に含まれます。通常は現在の変数の値を確認するために使用され、空の入力行が入力されるまで、デバッグプロンプトが繰り返し表示されます。

コマンドの構文は、以下のとおりです。

DBG_PROMPT_LINE [*prompt*]

prompt

表示するオプションのプロンプトの文字列を指定します。複数のプロンプトを使用する場合、プロンプトを使用して、中断した場所を示すこともできます。プロンプトを指定すると、「prompt(n)>」と表示されます。

デフォルト: "Dbg(n)> "

n

入力プロンプトとして行番号を表示します。

例

ループの最後の繰り返しで中断します。

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT_LINE
        ? i
}
```

カスタム プロンプトを使用して中断します。

```
for(i=0; i<11; i++)
{
    if(i==10)
        DBG_PROMPT_LINE "i==10> "
    ? i
}
```

関連項目

[DBG_PROMPT コマンド -- 確認のためにスクリプトの中断を実行 \(Cmdlet\)](#)
(P. 66)

del、rm コマンド -- ファイルの削除(Funclet)

これらのコマンド (funclet) は、ファイルまたはディレクトリを削除します。リモートホストを指定するには、パス名の前にホスト名とその後に 2 個のコロン (::) を指定します。操作は **SFTP** によって実行されます。したがって、リモートシステムにクライアント **AutoShell** がインストールされている必要はありません。**SFTP** アクセスが可能な **SSH** サーバがあれば十分です。

コマンドの構文は、以下のとおりです。

```
{del|rm} src [-silent] [-retval] [-R] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

src

削除するファイルまたはディレクトリを定義します。

-silent

(オプション) アクティブな出力ストリームへの出力を抑制します。

-retval

(オプション) 戻り値を表す配列を作成します。配列には新規ファイル名が格納されます。要素はそれぞれ、1 つの宛先ファイルの完全修飾名を表す文字列から構成されます。

-R

(オプション) ディレクトリを再帰的にコピーします。

-user username

(オプション) リモート ノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$User

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$Pass

-key key

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとし、公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

`-port portnumber`

(オプション) ターゲットシステムに接続するポートです。

デフォルト : SSH の標準ポート 22。

例

ファイル `c:/test/test1.log` を削除します。

```
del c:/test/test1.log
```

ディレクトリ `c:/test` 全体を削除します。

```
rm c:/test -R
```

リモート ホスト `remotesys` のディレクトリ `c:/test` 全体を削除します。

```
del remotesys::c:/test1/ -R
```

dir、ls コマンド -- ファイルおよびディレクトリ情報の取得(Funclet)

これらのコマンド (funclet) は、ローカルシステムまたはリモートシステムの指定されたディレクトリ内のコンテンツ情報をリスト表示します。**set alternate to** コマンドを使用して、ファイルに出力を書き込むことができます。

コマンドの構文は以下のとおりです。

```
{dir|ls} mask [-silent] [-retval] [-l] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

mask

オプションで絶対パスまたは相対パスのディレクトリ指定を含む検索マスクで文字列を定義します。

検索マスクは、DOS のワイルドカード検索と互換性のある、簡略化された正規表現構文を使用します。

- * は、ゼロ個以上の任意の文字に一致します。
- ? は、任意の 1 文字に一致します。
- [abc] は、a、b または c のいずれか 1 文字に一致します。
- [a-f] は、a ～ f のいずれかに一致します。
- [^a-f] は、a ～ f 以外のいずれかに一致します。
- [-_] は、- または _ に一致します。[^_] はそのほかの任意の文字に一致します。開始角かっこ、または ^ の直後に指定されている場合、ダッシュ「-」は特殊文字ではありません。
- [a^] は、a または a^ に一致します。（開始角かっこの直後に指定されていない場合、キャレット「^」は特殊文字ではありません）。
- ¥*、¥?、¥[, ¥], ¥¥ は、それぞれ 1 個の *、?、[,], ¥ に一致します。

他の文字はすべて特殊文字ではありません。それぞれの文字に一致します。

デフォルト: ""

-silent

(オプション) アクティブな出力ストリームへの出力を抑制します。

-retval

(オプション) 戻り値を表す配列を作成します。配列にはファイルの情報が格納されます。各要素は、ファイル名を表す文字列で構成されます。引数 **-l** を指定すると、各要素は以下の項目を含む配列で構成されます。

- **item[0]** - ファイル名を表す文字列
- **item[1]** - ファイルサイズを表す数値
- **item[2]** - 最終変更日時を表す日付
- **item[3]** - ディレクトリ フラグを表すブール値
- **item[4]** - パスを含む完全修飾ファイル名を表す文字列

-l

(オプション) 出力に長いリスト形式を使用します。

-user *username*

(オプション) リモート ノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、**AutoShell** へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : **\$\$User**

-pass *password*

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、**AutoShell** へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : **\$\$Pass**

-key key

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-port portnumber

(オプション) ターゲット システムに接続するポートです。

デフォルト : SSH の標準ポート 22。

例

現在のディレクトリにあるファイル名を取得します。

```
dir
```

ディレクトリ **c:\test** にあるファイル名を取得します。

```
dir c:/test
```

ディレクトリ `c:¥test` と `c:¥test¥more` にあるファイル名を取得し、ファイル `test.list` のコンテンツとして出力します。

```
set alternate to test.list
dir c:¥test
dir c:¥test¥more
set alternate to
```

リモートサーバ `ascli1` の `c:¥temp` にあるディレクトリのコンテンツをリスト表示します。

```
dir ascli1::c:/temp
```

後で処理するためにディレクトリ `c:/test/more` にあるファイル名を配列で取得し、標準出力への出力を抑制します。

```
set result disp off
aRet = dir c:/test/more -silent -retval
```

後で処理するためにディレクトリ `c:/test/more` にあるファイルの詳細情報を配列で取得します。

```
aRet = dir c:/test/more -silent -retval -l
```

関連項目

[cd、chdir コマンド -- ディレクトリの変更 \(Cmdlet、Funclet\)](#) (P. 62)

[pwd コマンド -- 作業ディレクトリ名の出力 \(Cmdlet\)](#) (P. 101)

[set alternate to コマンド -- 別の出力ファイルを設定する \(Cmdlet\)](#) (P. 125)

exit、quit コマンド -- AutoShell セッションの終了 (Cmdlet)

これらのコマンド(cmdlet)は、現在の Autoshell セッションを終了します。
対話型コマンド プロンプトまたはスクリプト内でコマンドを発行します。

コマンドの構文は、以下のとおりです。

```
exit
quit
```

関連項目

[setProcExitCode -- AutoShell 終了コードを設定する \(関数\)](#) (P. 157)

external コマンド -- ネイティブ外部関数の宣言 (Cmdlet)

ネイティブ外部関数を宣言します。

コマンドの構文は、以下のとおりです。

```
external decl
```

```
decl
```

ネイティブ外部関数を指定します。

get-autoShellClassInfo コマンド -- Autoshell ネイティブ クラスのメソッドとプロパティの表示 (Cmdlet)

このコマンド (cmdlet) は、AutoShell ネイティブ クラスのメソッドとプロパティを表示します。AutoShell では、標準の Spidermonkey/JavaScript スクリプト環境を拡張する多数のクラスが定義されています。製品固有の AutoShell ロード モジュール (ALM) では、追加のクラスを登録できます。これらのクラスには、コンストラクタ、メソッド、変数、列挙型のタイプ情報が含まれます。この情報は、get-autoShellClassInfo cmdlet を使用して表示できます。

注: この情報はヘルプの情報とは異なります。AutoShell は、内部クラス定義のみに基づいて情報を生成します。

コマンドの構文は、以下のとおりです。

```
get-autoShellClassInfo class [-super]
```

class

情報を表示する AutoShell クラスを指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。クラス名の代わりにオブジェクト名を渡すことができます。そのような場合、get-autoShellClassInfo コマンドは、オブジェクトからクラス名を取得してクラス情報を問い合わせます。文字列に変換されないように、オブジェクトは丸かっこで囲んで渡す必要があります。

例

RemoteTarget クラスの詳細を表示します。

```
get-autoShellClassInfo RemoteTarget
```

オブジェクトから OSRedirect クラスの詳細を表示します。

```
os = new OSRedirect();  
get-autoShellClassInfo (os)
```

関連項目:

[help コマンド -- ヘルプ情報をリクエスト \(Cmdlet\)](#) (P. 80)

[get-help コマンド -- ヘルプ情報の表示 \(Funclet\)](#) (P. 77)

get-help コマンド -- ヘルプ情報の表示 (Funclet)

このコマンド (**funclet**) は、AutoShell のコマンド、関数、クラスのヘルプ情報を表示し、オプションでヘルプ情報をファイルにリダイレクトします。

コマンドの構文は、以下のとおりです。

```
get-help [mask] [-output outFile]
```

mask

(オプション) ヘルプ情報が必要なコマンドをフィルタするマスクを指定します。ワイルドカードとしてアスタリスク「*」と疑問符「?」を使用できます。

デフォルト: "help"

outFile

(オプション) ヘルプ情報を保存するファイルを絶対パスまたは相対パスで指定します。同じ名前とパスの既存ファイルは上書きされます。

デフォルト: ""

例

以下のコマンドは、**run-remote funclet** のヘルプを表示します。

```
help run-remote
```

以下のコマンドは、すべてのコンストラクタとメソッドを含めて **RemoteTarget** クラスのヘルプを表示します。

```
help RemoteTarget*
```

以下のコマンドは、ヘルプ トピックをすべて表示し、ファイルに書き込みます。

```
help * -output help.txt
```

関連項目

[help コマンド -- ヘルプ情報をリクエスト \(Cmdlet\)](#) (P. 80)

[get-autoShellClassInfo コマンド -- Autoshell ネイティブ クラスのメソッドとプロパティの表示 \(Cmdlet\)](#) (P. 76)

get-remoteResult コマンド -- リモート ターゲットから結果を取得 (Funclet)

この funclet は、RemoteTarget オブジェクトからリモート実行の結果を取得します。リモート実行の戻り値は、リクエストを発行したシステムにネットワーク経由で転送する必要があります。結果は、リモートシステム上で XML 表現でシリアル化されます。RemoteTarget.result() メソッドを使用すると、シリアル化された表現で結果の値が返されます。

シリアル化された表現は、RemoteTarget オブジェクト自体をパラメータとする get-remoteResult() funclet を使用して、標準の JavaScript 値に変換します。通常は、スクリプトで result() を直接呼び出す必要はありません。シリアル化ではタイプ情報が保持されます。そのため、リモート ノードが数値を返すと、get-remoteResult() は数値を返します。リモートシステムが Date オブジェクトを返すと、元のシステム上でも Date オブジェクトになります。

リモートスクリプトからの戻り値は、単純なデータ型に制限されません。リモート実行でサーバ間での大量のデータ転送を可能にし、複雑な配列やデータのみオブジェクトを返すこともできます。唯一の制限は、データ構造に循環参照を含むことができないということです。

コマンドの構文は、以下のとおりです。

```
get-remoteResult remoteTarget
```

この funclet は、リモート実行プロセスのポリモーフィック値を返します。実行が完了していない場合、戻り値は未定義のタイプとなります。別のエラーが発生した場合は、例外が作成されます。

remoteTarget

リモート ターゲット システムの名前を指定します。

例

リモート実行のシリアル化された表現と実際の値を取得します。

```
rt = new RemoteTarget("ascli1", "bob", "ca123456");
run-remote "new Date()" on rt -wait
? rt.result()           // XML 文字列
v = get-remoteResult(rt) // result() を内部で呼び出し
? v
? typeName(v)           // Date
```

関連項目

[RemoteTarget クラス](#) (P. 173)

[RemoteTarget.output メソッド](#) (P. 187)

[RemoteTarget.result メソッド](#) (P. 189)

get-webServiceInfo コマンド -- WSDL 情報の表示 (Cmdlet)

get-webServiceInfo は、WSDL 情報を表示します。

```
get-webServiceInfo w [-detail]
```

w

Web サービスの名前を指定します。

help コマンド -- ヘルプ情報をリクエスト(Cmdlet)

このコマンド (cmdlet) は、AutoShell のコマンド、関数、クラスのヘルプ情報を表示し、オプションでヘルプ情報をファイルにリダイレクトします。

コマンドの構文は、以下のとおりです。

```
help [mask] [-output outFile]
```

mask

(オプション) ヘルプ情報が必要なコマンドをフィルタするマスクを指定します。ワイルドカードとしてアスタリスク「*」と疑問符「?」を使用できます。

デフォルト : "help"

outFile

(オプション) ヘルプ情報を保存するファイルを絶対パスまたは相対パスで指定します。同じ名前とパスの既存ファイルは上書きされます。

デフォルト : ""

例

以下のコマンドは、run-remote funclet のヘルプを表示します。

```
help run-remote
```

以下のコマンドは、すべてのコンストラクタとメソッドを含めて RemoteTarget クラスのヘルプを表示します。

```
help RemoteTarget*
```

以下のコマンドは、ヘルプ トピックをすべて表示し、ファイルに書き込みます。

```
help * -output help.txt
```

関連項目

[get-autoShellClassInfo コマンド -- Autoshell ネイティブ クラスのメソッドとプロパティの表示 \(Cmdlet\)](#) (P. 76)

[get-help コマンド -- ヘルプ情報の表示 \(Funclet\)](#) (P. 77)

mkdir コマンド -- ディレクトリの作成 (Cmdlet)

このコマンドは、ディレクトリが存在しない場合、ローカル システムまたはリモート システム上にディレクトリを作成します。

このコマンドは、指定されたディレクトリの場所を指すパス要素を自動的に作成します。 リモート ホストを指定するには、パス名の前にホスト名とその後に 2 個のコロン (::) を指定します。操作は **SFTP** によって実行されます。したがって、リモート システムにクライアント **AutoShell** がインストールされている必要はありません。**SFTP** アクセスが可能な **SSH** サーバがあれば十分です。

コマンドの構文は、以下のとおりです。

```
mkdir d [-silent] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

d

新しいディレクトリの名前を定義します。

-silent

(オプション) コマンドは、標準出力への出力を抑制します。

-user username

(オプション) リモート ノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、**AutoShell** へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : `$$User`

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、**AutoShell** へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : `$$Pass`

-key *key*

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-port *portnumber*

(オプション) ターゲット システムに接続するポートです。

デフォルト : SSH の標準ポート 22。

例

ディレクトリ **c:¥temp¥xyz** を作成します。

```
mkdir c:¥temp¥xyz
```

リモート サーバ **ascl1** 上にディレクトリを作成します。

```
mkdir ascl1:./home/alice/data
```

現在の作業ディレクトリのサブディレクトリとしてディレクトリ **temp** を作成し、標準出力への出力を抑制します。

```
mkdir temp -silent
```

関連項目

[cd、chdir コマンド -- ディレクトリの変更 \(Cmdlet、Funclet\)](#) (P. 62)

[dir、ls コマンド -- ファイルおよびディレクトリ情報の取得 \(Funclet\)](#) (P. 71)

[rmdir コマンド -- ディレクトリの削除 \(Funclet\)](#) (P. 104)

[pwd コマンド -- 作業ディレクトリ名の出力 \(Cmdlet\)](#) (P. 101)

mv、ren コマンド -- ファイルとディレクトリの移動(Funclet)

これらのコマンドは、ファイルまたはディレクトリを別の場所に移します。移動元または移動先の場所のどちらかはローカル サーバ上に存在する必要があります。2つのリモートシステム間での操作はサポートされていません。リモートホストを指定するには、パス名の前にホスト名とその後に2個のコロン (::) を指定します。操作はSFTPによって実行されます。したがって、リモートシステムにクライアント AutoShell がインストールされている必要はありません。SFTP アクセスが可能なSSHサーバがあれば十分です。set alternate コマンドを使用して、ファイルに出力を書き込むことができます。

移動と名称変更は、この場合同じです。

コマンドの構文は以下のとおりです。

```
{mv|ren} src trg [-silent] [-retval] [-R] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

src

移すディレクトリまたはファイルを指定する検索マスクを定義します。Windows の区切り文字には、円記号 (¥) またはスラッシュ (/) を使用できます。UNIX 系のシステムでは、スラッシュを使用する必要があります。プラットフォーム間のスクリプトの互換性のために、パスの指定にはスラッシュのみを使用してください。

検索マスクは、DOS のワイルドカード検索と互換性のある、簡略化された正規表現構文を使用します。

- * は、ゼロ個以上の任意の文字に一致します。
- ? は、任意の 1 文字に一致します。
- [abc] は、a、b または c のいずれか 1 文字に一致します。
- [a-f] は、a ～ f のいずれかに一致します。
- [^a-f] は、a ～ f 以外のいずれかに一致します。
- [-_] は、- または _ に一致します。[^_] はそのほかの任意の文字に一致します。開始角かっこ、または ^ の直後に指定されている場合、ダッシュ「-」は特殊文字ではありません。
- [a^] は、a または a^ に一致します。（開始角かっこの直後に指定されていない場合、キャレット「^」は特殊文字ではありません）。
- ¥*, ¥?, ¥[, ¥], ¥¥ は、それぞれ 1 個の *, ?, [,], ¥ に一致します。

他の文字はすべて特殊文字ではありません。それぞれの文字に一致します。

trg

移動先のパスを定義します。

-silent

(オプション) アクティブな出力ストリームへの出力を抑制します。

-retval

(オプション) 戻り値を表す配列を作成します。配列には新規ファイル名が格納されます。要素はそれぞれ、1つの宛先ファイルの完全修飾名を表す文字列から構成されます。

-R

(オプション) ディレクトリを再帰的にコピーします。

-user *username*

(オプション) リモートノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$User

-pass *password*

(オプション) リモートノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$Pass

-key *key*

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-port *portnumber*

(オプション) ターゲット システムに接続するポートです。

デフォルト : SSH の標準ポート 22。

例

ファイル **c:/test/test1.log** をディレクトリ **c:/test1** に移します。

```
mv c:/test/test1.log c:/test1
```

ディレクトリ **c:/test** 全体をディレクトリ **c:/test1** に移します。

```
mv c:/test c:/test1 -R
```

ディレクトリ **c:/test** 全体をリモート ホスト **remotesys** のディレクトリ **c:/test1** に移します。

```
mv c:/test1/ remotesys::c:/test1/ -R
```

関連項目

[copy、cp コマンド -- ファイルとディレクトリのコピー \(Funclet\)](#) (P. 63)

new-comObject コマンド -- COM オブジェクトのインスタンス化 (Funclet)

COM は、COM クライアントが使用するサービスをアプリケーションで公開できるようにする、Windows 固有の標準 API です。COM クライアントをサポートすると、Internet Explorer、Microsoft Office、さまざまな管理関連 OS サービス（特に WMI）など、広範囲の COM サーバにアクセスできるようになります。new-comObject コマンドは、スクリプト作成に利用できるサービスを COM サーバで公開できるようにする、プロキシクライアント オブジェクトを作成します。

コマンドの構文は、以下のとおりです。

```
new-comObject progid [-locale lcid]
```

progid

インスタンス化する COM オブジェクトを識別する文字列を指定します。この文字列は、人が判読できるプログラム ID (PROGID) か内部 CLSID になります。CLSID を指定する場合は、中かっこで囲む必要があります。

lcid

複数のロケールをサポートする COM サーバに対し、希望するロケールを定義する整数を指定します。

デフォルト : 0

戻り値

new-comObject は、AutoShell と COM サーバの間のプロキシを表すオブジェクトを返します。このオブジェクトは、COM サーバで公開されているメソッドとプロパティをミラーリングします。COM サーバのパブリック インターフェースの詳細については、該当のサーバのプログラムドキュメントを参照してください。公開されているメソッドとプロパティには、返された AutoObject で、標準の JavaScript 構文を使用してアクセスします。

例

Internet Explorer で Web ページを開き、その URL を問い合わせます。

```
o = new-comObject InternetExplorer.Application
// Internet Explorer を表示
o.Visible = true;
o.navigate("www.ca.com");
// アクティブなドキュメント (別の COM オブジェクト) を取得
doc = o.document
? doc.Url      // http://www.ca.com/us など
```

スクリプト作成オブジェクトを使用して、コンピュータ名を取得します。

```
o = new-comObject WScript.Network
? o.ComputerName
```

WMI を使用して、オペレーティング システムと論理ドライブに関する情報を取得します。

```
loc=new-comObject WbemScripting.SWbemLocator
// 実際には、localhost ではセキュリティは不要
loc.Security_.AuthenticationLevel=0; // wbemAuthenticationLevelDefault
loc.Security_.ImpersonationLevel=3; // wbemImpersonationLevelImpersonate
wmisrv=loc.connectServer("localhost", "root¥cimv2");
items = wmisrv.ExecQuery("Select * from Win32_OperatingSystem");
// OS が 1 つしか稼働していないため、
// 返されるコレクションのアイテムは 1 つのみ
os = items[0];
? "OS の種類:", os.Caption;
? "サービス パック:", os.CSDVersion;
? "コンピュータ名:", os.CSName;
? "空き物理メモリ:", os.FreePhysicalMemory;
? "登録ユーザ:", os.RegisteredUser;
? "可視メモリの合計サイズ:", os.TotalVisibleMemorySize;
? "バージョン:", os.Version;
? "Windows ディレクトリ:", os.WindowsDirectory;
```



```
// ディスク情報
items = wmisrv.ExecQuery("Select * from Win32_LogicalDisk");
// 返されたディスクに対して処理をループ
for(i in items)
{
    disk = items[i];
    ? "説明: ", disk.Description;
    ? "デバイス ID: ", disk.DeviceID;
    ? "ファイル システム: ", disk.FileSystem;
    ? "空き容量: ", disk.FreeSpace;
    ? "名前: ", disk.Name;
    ? "サイズ: ", disk.Size;
}
```

new-SSHSession コマンド -- 安全な SSH セッションの新規作成(Funclet)

new-SSHSession コマンドは、SSH サーバ上でセッションを作成し、そのセッションを表す **CASHELL** オブジェクトを返します。このセッション上でアクションを行うには、このオブジェクトを保存する必要があります。このコマンドは通常、**run-SSHCommand** または **run-SSHShell** コマンドと共に使用します。

コマンドの構文は、以下のとおりです。

```
new-SSHSession -host hostname [-user username] [-pass password] [-key key]  
[-prompt prompt][-port portnumber]
```

-host *hostname*

SSH サーバの名前です。

-user *username*

(オプション) リモート ノードへのログインに使用するユーザ名を指定します。パスワードまたは公開鍵認証を使用する場合、ユーザ名は必須です。ユーザ名を指定しない場合、**AutoShell** へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : **\$\$User**

-pass *password*

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、**AutoShell** へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : **\$\$Pass**

-key *key*

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

-prompt *prompt*

(オプション) 入力できるようになったことを示すために SSH サーバが表示するプロンプト文字列の末尾の固定部分 (例 : AutoShell では「::->」) を指定します。この情報は、SSH セッションで実行されたコマンドの完了を判断するために必要です。できるだけ多くの固定文字を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : "::->"

-port *portnumber*

(オプション) ターゲットシステムに接続するポートです。

デフォルト : SSH の標準ポート 22。

戻り値

CASHELL オブジェクトを返します。このオブジェクトを run-SSHCommand funclet に渡してコマンドを実行したり、run-SSHSession に渡して SSH サーバ上で対話型セッションを実行したりすることができます。

例

SSH セッションを作成します、サーバでは、指定したアカウントに対し、SSH シェルとして AutoShell を設定しているものとします。コマンドを実行し、切断します。

```
ssh=new-SSHSession -host client8 -user bob -pass xyz42 -prompt ::->
? run-SSHCommand -session ssh -command "! ver"
ssh.disconnect();
```

関連項目:

[run-SSHCommand コマンド -- SSH \(Funclet\) を使用してリモート コマンド
を実行する \(P. 115\)](#)

[run-SSHShell コマンド -- 対話型の SSH コマンド ラインを実行する \(Funclet\)
\(P. 119\)](#)

new-webService コマンド -- Web サービス オブジェクトの作成 (Funclet)

このコマンドは、Web サービス オブジェクトを作成します。

コマンドの構文は、以下のとおりです。

```
new-webService WSDLURI [-endpoint endpoint] [-user username] [-pass password]  
[-proxy proxyname] [-proxyUser username] [-proxyPass password]
```

WSDLURI

WSDLURI を指定します。

-endpointt *endpoint*

(オプション) エンド ポイントを指定します。

デフォルト: ""

-user *username*

(オプション) リモート ノードへのログインに使用するユーザ名を指定します。パスワード認証を使用する場合、ユーザ名は必須です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$User

-pass *password*

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$Pass

-proxy proxyname

(オプション) プロキシの名前を指定します。

デフォルト: ""

-proxyUser username

(オプション) プロキシへのログインに使用するユーザ名を指定します。

デフォルト: \$\$User

-proxyPass password

(オプション) プロキシへのログインに使用するパスワードを指定します。

デフォルト: \$\$Pass

objdump コマンド -- オブジェクトの列挙可能なプロパティの表示 (Cmdlet)

このコマンド (cmdlet) は、指定されたオブジェクトとそのタイプ、および現在値 (オプション) の列挙可能なプロパティをリスト表示します。

コマンドの構文は、以下のとおりです。

```
objdump obj [-detail]
```

obj

表示するオブジェクトを指定します。

-detail

(オプション) このオプションを指定すると、リスト表示されるプロパティの現在値も表示されます。

例

オブジェクトのプロパティを表示します。

```
o = new Object();  
o.num = 42;  
o.str = "Hello";  
objdump o
```

現在値を表示します。

```
objdump o -detail
```

関連項目

[arrdump コマンド -- 配列の表示 \(Cmdlet\)](#) (P. 58)

push-client コマンド -- Autoshell クライアントをリモート Windows システムにインストール (Funclet)

このコマンドは、リモート Windows システムに AutoShell クライアントをインストールします。

このコマンドは、1 つ以上のリモート Windows システム上でコピーおよび実行を行います。利用可能なバージョンは以下のとおりです。

- 単一ターゲットバージョン (1)
- 複数ターゲットバージョン (2)

コマンドの構文は、以下のとおりです。

- (1) `push-client -host hostname [-localPath path] [-user username>] [-pass password]`
- (2) `push-client [-localPath path] [-user username>] [-pass password] on remoteTargets`

単一ターゲットのコマンドは、成功すると `true` を返します。

複数ターゲットのコマンドは、WinRemote オブジェクトの配列を返します。この配列の各要素は、*remoteTargets* リストにリストされたリモート Windows ノードの位置に対応します。WinRemote オブジェクトは、エラーの発生を確認し、出力を問い合わせ、実行が完了した場合に結果を返すことができます。WinRemote オブジェクトを使用してリモート ノードをすでに指定している場合、配列には元の RemoteTarget オブジェクトへの参照が格納されます。ターゲットシステムが文字列によって指定されている場合、新しい RemoteTarget オブジェクトが内部に作成され、配列に割り当てられます。

-host *hostname*

(オプション) リモート **Windows** ホストの名前を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

-localPath *path*

(オプション) コピーする **AutoShell** クライアントを示すパスを指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで **SSH** セッションを作成する場合にのみ必要です。

デフォルト : ""

-user *username*

(オプション) リモートの **Windows** サーバにログインするユーザ名を指定します。このユーザには、**\$ADMIN** 共有に接続する権限が必要です。ユーザ名を指定しない場合、**AutoShell** へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$User

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : `$$Pass`

on remoteTargets

指定したスクリプトを実行するリモートシステムのカンマ区切りリストを指定します。ターゲット ノードは、ホスト名/アドレスと WinRemote オブジェクトの 2 つの方法で指定できます。WinRemote オブジェクトを使用するときに名前またはアドレスを使用してターゲットシステムを指定すると、たとえば別のホストに別の認証情報を使用するなど、より柔軟に実行プロセスを制御できます。リモートターゲット リストには、ホスト名/アドレスを持つ WinRemote オブジェクトおよび文字列を混合させることができます。ターゲット リストの要素は自動的に文字列化されません。そのため、ホスト名にリテラル文字列を使用する場合は引用符で囲む必要があります。

例

host1 に AutoShell クライアントをインストールする方法

```
push-client -host host1
```

host1 と host2 に AutoShell クライアントをインストールする方法

```
x = push-client on "host1", "host2";  
? x[1].hasCompleted();  
? x[1].result();  
? x[1].output();
```


関連項目:

[rem-client コマンド -- リモート Windows システムから Autoshell クライアントを削除 \(Funclet\) \(P. 103\)](#)

[run-client コマンド -- リモート Windows システム上で Autoshell クライアントを実行 \(Funclet\) \(P. 106\)](#)

[push-winRemote コマンド -- 複数のリモート Windows システム上でのコピーと実行 \(Funclet\) \(P. 97\)](#)

[run-winRemote コマンド -- 複数のリモート Windows システムでコマンドを実行する \(Funclet\) \(P. 122\)](#)

push-winRemote コマンド -- 複数のリモート Windows システム上でのコピーと実行 (Funclet)

このコマンドは、1 つ以上のリモート Windows システム上でコピーおよび実行を行います。利用可能なバージョンは以下のとおりです。

- 単一ターゲットバージョン (1)
- 複数ターゲットバージョン (2)

コマンドの構文は、以下のとおりです。

(1) `push-winRemote -host hostname -localPath path [-cmd cmd] [-user username] [-pass password]`

(2) `push-winRemote -localPath path [-cmd cmd] [-user username] [-pass password] on remoteTargets`

単一ターゲットバージョンは、実行したコマンドの結果コードを返します。

複数ターゲットバージョンは、WinRemote オブジェクトの配列を返します。この配列の要素はそれぞれ、*remoteTargets* リストの対応する位置にあるリモート Windows ノードに対応します。WinRemote オブジェクトは、エラーの発生を確認し、出力を問い合わせ、実行が完了した場合に結果を返すことができます。WinRemote オブジェクトを使用してリモートノードが指定されている場合、配列には元の RemoteTarget オブジェクトへの参照が含まれます。ターゲットシステムが文字列によって指定されている場合、新しい RemoteTarget オブジェクトが内部に作成され、配列に割り当てられます。

-host *hostname*

リモート **Windows** ホストの名前を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

-localPath *path*

コピーする **AutoShell** クライアントを示すパスを指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで **SSH** セッションを作成する場合にのみ必要です。

デフォルト : ""

-cmd *cmd*

(オプション) リモートの **Windows** システムで実行するコマンドを指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで **SSH** セッションを作成する場合にのみ必要です。

デフォルト : ""

-user *username*

(オプション) リモートの **Windows** システムにログインするユーザ名を指定します。このユーザには、**\$ADMIN** 共有に接続する権限が必要です。ユーザ名を指定しない場合、**AutoShell** へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$User

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: `$$Pass`

on remoteTargets

指定したスクリプトを実行するリモートシステムのカンマ区切りリストを指定します。ターゲット ノードは、ホスト名/アドレスと WinRemote オブジェクトの 2 つの方法で指定できます。WinRemote オブジェクトを使用するときに名前またはアドレスを使用してターゲットシステムを指定すると、たとえば別のホストに別の認証情報を使用するなど、より柔軟に実行プロセスを制御できます。リモートターゲット リストには、ホスト名/アドレスを持つ WinRemote オブジェクトおよび文字列を混合させることができます。ターゲット リストの要素は自動的に文字列化されません。したがって、ホスト名用のリテラル文字列を使用する場合は、引用符で囲む必要があります。

例

host1 に myinstall.exe をインストールする方法

```
push-winRemote -host host1 -localpath c:%installations%myinstall.exe -cmd  
myinstall.exe
```

host1 と host 2 に myinstall.exe をインストールし、host1 の結果を確認する方法

```
push-winRemote -cmd myinstall.exe -localpath c:%installations%myinstall.exe on  
"host1", "host2";  
? x[0].hasCompleted();  
? x[0].result();  
? x[0].output();
```

関連項目:

[run-winRemote コマンド -- 複数のリモート Windows システムでコマンドを実行する \(Funclet\)](#) (P. 122)

[push-client コマンド -- Autoshell クライアントをリモート Windows システムにインストール \(Funclet\)](#) (P. 94)

[run-client コマンド -- リモート Windows システム上で Autoshell クライアントを実行 \(Funclet\)](#) (P. 106)

[rem-client コマンド -- リモート Windows システムから Autoshell クライアントを削除 \(Funclet\)](#) (P. 103)

PW_GET コマンド -- 非表示データの入力 (Funclet)

このコマンド (funclet) は、オプションで入力を促すメッセージを表示し、Enter キーが押されるまで文字を読み取ります。入力された各文字は、入力を非表示にするためにアスタリスク (*) で表示されます。

コマンドの構文は、以下のとおりです。

`PW_GET(prompt)`

このコマンドは、入力された文字を保持する文字列をクリア テキストで返します。

prompt

入力プロンプトとして表示される文字列です。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式を丸かっこで囲み、式が自動的に引用されないようにしてください。

デフォルト: ""

例

コマンドの一部としてクリア テキストでパスワードを入力できない場合は、このコマンドを使用してパスワードを非表示にします。

```
run-remote Math.PI on "ascl1" -user alice -pass PW_GET(Enter password:)
```

関連項目

[accept コマンド -- 標準入力から文字列を読み取り、変数に割り当てる \(Cmdlet\)](#) (P. 58)

pwd コマンド -- 作業ディレクトリ名の出力 (Cmdlet)

このコマンドは、現在の作業ディレクトリの名前を出力します。

コマンドの構文は、以下のとおりです。

```
pwd
```

例

AutoShell がデフォルトの場所内にインストールされている場合、`pwd` コマンドを実行すると `C:\Program Files\CA\SC\AutoShellManager` と表示されます。

query-service コマンド -- Windows サービスのステータスの問い合わせ (Funclet)

`query-service` コマンドは、ローカル システムまたはリモート システム上の Windows サービスのステータスを問い合わせます。`query-service` コマンドは、サービスのステータスを文字列として返します。可能な戻り値は以下のとおりです。

- 停止済み
- 起動
- 停止
- 実行
- 続行待ち
- 一時停止待ち
- エラーで停止
- クエリは失敗しました

コマンドの構文は、以下のとおりです。

```
query-service -svcName sName [-host hostname] [-user username] [-pass password]
```

-svcName *sName*

サービスの名前を指定します。

-host *hostname*

(オプション) リモート Windows システムの名前。

デフォルト: ""

-user *username*

(オプション) リモート **Windows** システムへのログオンに使用するユーザ名。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-pass *password*

リモート **Windows** システムへのログオンに使用するパスワード。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

例

ローカル **Apache** サービスを問い合わせる方法

```
? query-service -svcName Apache2.2
```

リモート サーバ **REMOTESYS** の **Microsoft SQL Server** サービスを問い合わせる方法

```
? query-service -svcName MSSQLSERVER -host REMOTESYS -user admin -pass notsecret
```

関連項目 :

[start-service コマンド -- リモートまたはローカル システムの **Windows** サービスを開始する \(Funclet\)](#) (P. 127)

[stop-service コマンド -- リモートまたはローカル システムの **Windows** サービスを停止する \(Funclet\)](#) (P. 129)

rem-client コマンド -- リモート Windows システムから Autoshell クライアントを削除 (Funclet)

rem-client コマンドは、リモート Windows システムから Autoshell クライアントを削除します。

コマンドの構文は、以下のとおりです。

```
rem-client -host hostname [-user username] [-pass password]
```

このコマンドは、成功すると **true** を返します。

-host *hostname*

リモート Windows ホストを指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

-user *username*

(オプション) リモートの Windows システムにログインするユーザ名を指定します。このユーザには、\$ADMIN 共有に接続する権限が必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$User

-pass *password*

(オプション) リモート ノードへのログイン時に使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後、非表示でパスワードを入力するには、引数としてオプションのプロンプトと PW_GET() を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$Pass

例

host1 の AutoShell クライアントをアンインストールします。

```
rem-client -host host1
```

関連項目:

[push-client コマンド -- Autoshell クライアントをリモート Windows システムにインストール \(Funclet\) \(P. 94\)](#)

[push-winRemote コマンド -- 複数のリモート Windows システム上でのコピーと実行 \(Funclet\) \(P. 97\)](#)

[run-client コマンド -- リモート Windows システム上で Autoshell クライアントを実行 \(Funclet\) \(P. 106\)](#)

[run-winRemote コマンド -- 複数のリモート Windows システムでコマンドを実行する \(Funclet\) \(P. 122\)](#)

rmkdir コマンド -- ディレクトリの削除(Funclet)

このコマンドはディレクトリを削除します。指定するディレクトリは空である必要があります。ファイルやサブディレクトリを含むことはできません。リモートホストを指定するには、パス名の前にホスト名とその後に 2 個のコロン (::) を指定します。操作は SFTP によって実行されます。したがって、リモートシステムにクライアント AutoShell がインストールされている必要はありません。SFTP アクセスが可能な SSH サーバがあれば十分です。

コマンドの構文は、以下のとおりです。

```
rmkdir d [-silent] [-user username] [-pass password] [-key key phrase] [-port portnumber]
```

d

削除するディレクトリの名前を指定します。リモートホストを指定するには、パス名の前にホスト名とその後に 2 個のコロン (::) を指定します。操作は SFTP によって実行されます。したがって、リモートシステムにクライアント AutoShell がインストールされている必要はありません。SFTP アクセスが可能な SSH サーバがあれば十分です。set alternate コマンドを使用して、ファイルに出力を書き込むことができます。

-silent

(オプション) コマンドは、標準出力への出力を抑制します。

-user username

(オプション) リモート ノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$User

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$Pass

-key key

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとし、公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-port *portnumber*

(オプション) ターゲットシステムに接続するポートです。

デフォルト: SSH の標準ポート 22。

例

ディレクトリ **c:\temp** を削除します。

```
rmdir c:/temp
```

現在の作業ディレクトリのサブディレクトリ **temp** を削除し、標準出力への出力を抑制します。

```
rmdir temp -silent
```

リモート システム **ascl1** からディレクトリを削除します。

```
rmdir ascl1:/home/alice/data
```

関連項目

[dir、ls コマンド -- ファイルおよびディレクトリ情報の取得 \(Funclet\)](#) (P. 71)

[cd、chdir コマンド -- ディレクトリの変更 \(Cmdlet、Funclet\)](#) (P. 62)

[mkdir コマンド -- ディレクトリの作成 \(Cmdlet\)](#) (P. 81)

[del、rm コマンド -- ファイルの削除 \(Funclet\)](#) (P. 68)

[pwd コマンド -- 作業ディレクトリ名の出力 \(Cmdlet\)](#) (P. 101)

run-client コマンド -- リモート Windows システム上で Autoshell クライアントを実行 (Funclet)

run-client コマンド (funclet) は、リモート Windows システム上で Autoshell クライアントを実行します。入出力は、呼び出し側の AutoShell にリダイレクトされます。

コマンドの構文は、以下のとおりです。

```
run-client -host hostname [-user username>] [-pass password]
```

このコマンドは **true** または **false** を返します。

-host *hostname*

リモート Windows ホストの名前を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

-user username

(オプション) リモートの Windows システムにログインするユーザ名を指定します。このユーザには、\$ADMIN 共有に接続する権限が必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$User

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと PW_GET() を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$Pass

例

host1 で AutoShell クライアントを実行します。

```
run-client -host host1
```

関連項目:

[push-client コマンド -- Autoshell クライアントをリモート Windows システムにインストール \(Funclet\)](#) (P. 94)

[push-winRemote コマンド -- 複数のリモート Windows システム上でのコピーと実行 \(Funclet\)](#) (P. 97)

[rem-client コマンド -- リモート Windows システムから Autoshell クライアントを削除 \(Funclet\)](#) (P. 103)

[run-winRemote コマンド -- 複数のリモート Windows システムでコマンドを実行する \(Funclet\)](#) (P. 122)

run-local コマンド -- ローカル システム上でスクリプトを実行(Funclet)

`run-local funclet` は、現在の `AutoShell` セッションで、文字列またはローカル マシン上のディスク ファイル内に含まれている `JavaScript` コードを実行し、結果を返します。関数宣言に含まれていないコードは直接実行されます。関数宣言は後で関数を実行できるように処理されますが、関数は呼び出しません。`JavaScript` ファイルは対話型コマンドプロンプトでパス名を入力するだけで実行できます。一方、スクリプト ファイルから外部スクリプトを実行したり、対話型コマンドプロンプトから呼び出す場合にスクリプト ファイルにパラメータを渡したりするには、`run-local` を使用することが唯一の方法です。

コマンドの構文は、以下のとおりです。

```
run-local [script] [-file file] [-with args,...]
```

戻り値は、指定したスクリプトによって異なります。

`script`

(オプション) 実行する `JavaScript` コードが含まれている文字列を指定します。この `funclet` は、文字列またはファイルからスクリプトを実行します。そのため、*script* 引数と *file* 引数はどちらかしか指定できません。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

`-file file`

(オプション) 実行するスクリプト ファイルを絶対パスまたは相対パスで指定します。この `funclet` は、文字列またはファイルからスクリプトを実行します。そのため、*script* 引数と *file* 引数はどちらかしか指定できません。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

`-with args,...`

スクリプトに渡す実際の引数をカンマ区切りのリストで指定します。スクリプト コードでは、標準的な `JavaScript` 引数配列を使用してこれらの引数にアクセスできます。

例

単純な式を評価します。

```
run-local 1+2
```

引数を渡し、メモリで式を評価します。

```
run-local "arguments[0]*arguments[0]" -with 2
```

スクリプト ファイルを書き込み、実行します。

```
s = "var i;"
s += "for(i=0;i<arguments.length;i++)"
s += "qout(arguments[i]);"
s += "true;"
memoWrit("script.js", s);
run-local -file script.js -with 1, "abc", new Date()
```

注: スクリプト コードが関数定義内に含まれていない場合、スクリプト コードは単に直接実行されます。 スクリプトは **return** ステートメントを使用して結果（この場合は **true**）を返していませんが、最後に実行される式で指定しています。

関連項目:

[run-remote コマンド -- リモートシステム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

run-remote コマンド -- リモート システム上でスクリプトを実行 (Funclet)

`run-remote` コマンドは、1 つ以上のリモート システム上で、文字列または ディスク ファイルに含まれている **JavaScript** コードを実行します。リモート 実行を行うには、リモート ターゲット ノードに **AutoShell** クライアントをインストールし、適切に設定する必要があります。スクリプト ファイルはマネージャ システム上に保存し、実行するターゲット システムに転送する必要があります。デフォルトでは、スクリプトは非同期で実行されます。つまり、リモート 実行が完了する前もローカル処理は継続されています。

コマンドの構文は、以下のとおりです。

```
run-remote [script] [-file file] [-wait] [-with args,...] [-user username]
[-pass password] [-key key phrase] [-port portnumber] on remoteTargets
```

`run-remote` コマンドは、指定したサーバでのスクリプトの実行状態を表す **RemoteTarget** オブジェクトの配列を返します。この配列の要素はそれぞれ、**remoteTargets** リストの対応する位置にあるリモート ノードに対応します。**RemoteTarget** オブジェクトはいくつかの結果を確認できます。たとえば、エラーの発生、実行の完了、および成功の可否などです。**RemoteTarget** オブジェクトを使用してリモート ノードをすでに指定している場合、配列には元の **RemoteTarget** オブジェクトへの参照が格納されます。ターゲット システムが文字列によって指定されている場合、新しい **RemoteTarget** オブジェクトが内部に作成され、配列に割り当てられます。

script

(オプション) 実行する **JavaScript** コードが含まれている文字列を指定します。この **funclet** は、文字列またはファイルからスクリプトを実行します。そのため、**script** 引数と **file** 引数はどちらかしか指定できません。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-file file

(オプション) 実行するスクリプト ファイルを絶対パスまたは相対パスで指定します。この **funclet** は、文字列またはファイルからスクリプトを実行します。そのため、**script** 引数と **file** 引数はどちらかしか指定できません。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-wait

(オプション) デフォルトでは、**run-remote** はリモート実行を開始すると戻ってきます。リモート処理は非同期で行われます。**-wait** オプションを指定すると、**run-remote** はすべてのリモート タスクが正常に完了するか、エラーで中止されるまで待機します。

-with args,...

(オプション) スクリプトに渡す実際の引数をカンマ区切りのリストで指定します。スクリプト コードでは、標準的な JavaScript 引数配列を使用してこれらの引数にアクセスできます。

-user username

(オプション) リモート ノードへのログインに使用するユーザ名を指定します。パスワードまたは公開鍵認証を使用する場合、ユーザ名は必須です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。指定したユーザ名は、**remoteTargets** リスト内のホスト名で識別されるリモートシステムにのみ使用されます。**RemoteTarget** オブジェクトを使用してホストを指定している場合、**RemoteTarget** オブジェクトによってホストごとにユーザ名が個別に設定されます。

デフォルト : \$\$User

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。指定したパスワードは、`remoteTargets` リスト内のホスト名で識別されるリモートシステムにのみ使用されます。`RemoteTarget` オブジェクトを使用してホストを指定している場合、`RemoteTarget` オブジェクトによってホストごとにユーザ名が個別に設定されます。

デフォルト : `$$Pass`

-key key

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとし、公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : `""`

phrase

(オプション) 秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : `""`

-port *portnumber*

(オプション) ターゲット システムに接続するポートを指定します。指定したポートは、**remoteTargets** リスト内のホスト名で識別されるリモート システムにのみ使用されます。**RemoteTarget** オブジェクトを使用してホストを指定している場合、**RemoteTarget** オブジェクトによってホストごとにポートが個別に設定されます。

デフォルト : SSH の標準ポート 22。

on *remoteTargets*

指定したスクリプトを実行するリモート システムのカンマ区切りリストを指定します。ターゲット ノードは、ホスト名/アドレスと **WinRemote** オブジェクトの 2 つの方法で指定できます。**WinRemote** オブジェクトを使用するときには名前またはアドレスを使用してターゲット システムを指定すると、たとえば別のホストに別の認証情報を使用するなど、より柔軟に実行プロセスを制御できます。リモート ターゲット リストには、ホスト名/アドレスを持つ **WinRemote** オブジェクトおよび文字列を混合させることができます。ターゲット リストの要素は自動的に文字列化されません。したがって、ホスト名用のリテラル文字列を使用する場合は、引用符で囲む必要があります。

例

単純な式を評価し、実行が完了するまで待機します。

```
aRT = run-remote "1+2" on "ascli1" -wait
? "Error occurred:", aRT[0].errorOccurred()
? "Result:", get-remoteResult(aRT[0])
```

2つのリモートシステムにあるファイルのスクリプトを評価して、実行が完了するまで待機し、出力（「Hello World!」）および結果（42）を表示します。

```
s="? 'Hello World';42;"
// 説明のみのためにスクリプト ファイルを作成する
// 次のように呼び出すこともできる
// aRT=run-remote (s) on "ascli1", "ascil2"
memoWrit("hello.js", s);
aRT=run-remote -file hello.js on "ascli1", "ascil2"
for(i=0; i<aRT.length; i++)
{
    ? "Target:", aRT[i].getHostName()
    ? "Error occurred:", aRT[i].errorOccurred()
    ? "Output:", aRT[i].errorOccurred()
    ? "Result:", get-remoteResult(aRT[i])
}
```

RemoteTarget オブジェクトを使用して 2 つのリモートシステムのスクリプトを評価し、ターゲットシステムに異なる認証情報を指定します。

```
rt1 = new RemoteTarget("ascli1", "alice", "casogood42");
rt2 = new RemoteTarget("ascli2", "bob", "!secret!7");
// リモート システムから OS バージョンの文字列を取得する
aRT = run-remote "! ver" on rt1, rt2
// 自分自身を待機する
while(true)
{
    // エラーが発生した場合、hasCompleted() は
    // 同様に true を返す
    if(rt1.hasCompleted() && rt2.hasCompleted())
        break;
    sleep(500);
}
// 元の RemoteTarget オブジェクト（上記参照）
// または返された配列（下記参照）内の参照の
// いずれかを使用できる。
for(i=0; i<aRT.length; i++)
{
    ? "Target:", aRT[i].getHostName()
    ? "Error occurred:", aRT[i].errorOccurred()
    ? "Output:", aRT[i].errorOccurred()
}
```

関連項目:

[get-remoteResult コマンド -- リモート ターゲットから結果を取得 \(Funclet\)](#) (P. 78)

[RemoteTarget クラス](#) (P. 173)

[run-local コマンド -- ローカル システム上でスクリプトを実行 \(Funclet\)](#) (P. 108)

run-SSHCommand コマンド -- SSH (Funclet) を使用してリモート コマンドを実行する

run-SSHCommand は、既存の SSH セッション内でコマンドを実行するか、コマンドを実行するためのセッションを作成します。指定されたコマンドが実行された後は、そのセッションは廃棄されます。AutoShell はプライマリ リモート通信ルートとして SSH を使用しますが、このコマンドのリモート ノードではクライアント AutoShell を実行している必要はありません。run-remote コマンドはクライアント AutoShell を実行しているリモート ノードでコマンドを実行する場合に適しています。これは、このコマンドがコマンドの実行に関する詳細情報を返すためです。

コマンドは SSH サーバを実行している任意のシステムで実行できます。利用可能なコマンドとそのオプションは、リモート ノードに設定されたシェルに依存します。

コマンドの構文は、以下のとおりです。

```
run-SSHCommand [-host hostname] [-user username] [-pass password] [-key key phrase]
[-prompt prompt] -command cmd [-port portnumber] [-session ssh]
```

-host *hostname*

SSH サーバの名前です。

-user username

(オプション) リモート ノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$User

-pass password

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : \$\$Pass

-key key

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-prompt *prompt*

(オプション) 入力できるようになったことを示すために SSH サーバが表示するプロンプト文字列の末尾の固定部分 (例: **AutoShell** では「**::->**」) を指定します。この情報は、SSH セッションで実行されたコマンドの完了を判断するために必要です。できるだけ多くの固定文字を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: "**::->**"

-port *portnumber*

(オプション) ターゲットシステムに接続するポートです。

デフォルト: SSH の標準ポート **22**。

-cmd *cmd*

(オプション) リモートの **Windows** システムで実行するコマンドを指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト: **cmd**

-session *ssh*

新しい **SSHSession** によって返される **CASHELL** オブジェクト。既存の SSH セッション オブジェクトを渡す場合、**run-SSHCommand** では、指定されたコマンドを実行するために SSH セッションを作成する必要はありません。複数のコマンドセッション ネゴシエーションを実行する場合にログオンを 1 回のみ実行すると、通常は複数のセッションを割り当てるよりもパフォーマンスが向上します。このパラメータを指定すると、**host**、**user**、**pass**、**key**、**phrase**、および **port** に渡された値は無視されます。この情報は、セッションの作成時にすでに指定されています。

デフォルト: **null**

例

明示的にユーザ名とパスワードを指定し、SSH セッションを使用して、リモート ノードのディレクトリのリスト表示を実行します。

```
s=run-SSHCommand -host srv14 -user bob -pass xyz42 -prompt :> -command ls
```

AutoShell のログインに使用したユーザ名とパスワードを使用し、SSH セッションを使用して、リモート ノードのディレクトリのリスト表示を実行します。

```
s = run-SSHCommand -host srv14 -prompt ":>" -command "ls"
```

明示的に割り当てられた SSH セッションで 2 つのコマンドを実行します。

```
ssh=new-SSHSession -host client8 -user bob -pass xyz42 -prompt ::->
? run-SSHCommand -session ssh -command "! ver"
? run-SSHCommand -session ssh -command "ls *.* -l"
ssh.disconnect();
```

関連項目:

[new-SSHSession コマンド -- 安全な SSH セッションの新規作成 \(Funclet\)](#) (P. 90)

[run-remote コマンド -- リモート システム上でスクリプトを実行 \(Funclet\)](#) (P. 110)

run-SSHShell コマンド -- 対話型の SSH コマンドラインを実行する(Funclet)

`run-SSHCommand` コマンドは、入力行を受け入れ、SSH を使用してリモートシステムにそれらを送信して、リモート ノードからの出力を表示します。コマンドラインは、既存の SSH セッションまたは新しい SSH セッションで実行できます。入力が終了すると、そのセッションは廃棄されます。AutoShell はプライマリ リモート通信ルートとして SSH を使用しますが、このコマンドのリモート ノードでクライアント AutoShell を実行する必要はありません。コマンドは SSH サーバを実行している任意のシステムで実行できます。利用可能なコマンドとそのオプション、および入力ループを終了する方法は、リモート ノードに設定されたシェルに依存します。通常、入力ループは「quit」または「exit」を入力することで終了できます。

コマンドの構文は、以下のとおりです。

```
run-SSHShell [-host hostname] [-user username] [-pass password] [-key key phrase]
[-prompt prompt] [-port portnumber] [-session ssh]
```

-host *hostname*

SSH サーバの名前です。

-user *username*

(オプション) リモート ノードにログインするユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$User

-pass *password*

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと `PW_GET()` を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト: \$\$Pass

-key *key*

(オプション) リモート ノードへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで SSH セッションを作成する場合にのみ必要です。

デフォルト : ""

phrase

秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-prompt *prompt*

(オプション) 入力できるようになったことを示すために SSH サーバが表示するプロンプト文字列の末尾の固定部分 (例 : AutoShell では「**::->**」) を指定します。この情報は、SSH セッションで実行されたコマンドの完了を判断するために必要です。できるだけ多くの固定文字を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : "::->"

-port *portnumber*

(オプション) ターゲットシステムに接続するポートです。

デフォルト : SSH の標準ポート 22。

-session ssh

新しい `SSHSession` によって返される `CASHELL` オブジェクトを指定します。既存の `SSH` セッション オブジェクトを渡す場合、`run-SSHShell` では、コマンドラインからコマンドを実行するために `SSH` セッションを作成する必要はありません。このパラメータを指定すると、*host*、*user*、*pass*、*key*、*phrase*、および *port* に渡された値は無視されます。この情報は、セッションの作成時に指定されます。

デフォルト : `null`

例

パスワード認証を使用する `SSH` セッションを使用して、ホストで対話型のコマンドラインシェルを実行します。

```
run-SSHShell -host svr14 -user bob -pass xyz42 -prompt :>
```

明示的に `SSH` セッションを割り当てて、最初にコマンドを実行してからコマンドラインの入力ループに入ります。

```
ssh=new-SSHSession -host client8 -user bob -pass xyz42 -prompt ::->
run-SSHCommand -session ssh -command "cd /"
run-SSHShell -session ssh
ssh.disconnect();
```

関連項目

[new-SSHSession コマンド -- 安全な SSH セッションの新規作成 \(Funclet\)](#) (P. 90)

[run-SSHCommand コマンド -- SSH \(Funclet\) を使用してリモート コマンドを実行する](#) (P. 115)

run-winRemote コマンド -- 複数のリモート Windows システムでコマンドを実行する(Funclet)

run-winRemote コマンドは、1 つまたは複数のリモート Windows システムでコマンドを実行します。コマンドには次の 2 つの形式があります。

- 単一ターゲットバージョン
- 複数ターゲットバージョン

単一ターゲットバージョンは、呼び出し元の AutoShell に入出力をリダイレクトします。

複数ターゲットバージョンの run-winRemote は非同期です。結果を取得する方法については、サンプルコードを参照してください。

コマンドの構文は、以下のとおりです。

(1) run-winRemote -host *hostname* [-cmd *cmd*] [-user *username*] [-pass *password*] [-workDir *workDir*]

(2) run-winRemote [-cmd *cmd*] [-user *username*] [-pass *password*] [-workDir *workDir*] on *remoteTargets*

単一ターゲットバージョンの run-winRemote は、実行されたコマンドの結果コードを返します。

複数ターゲットバージョンは、WinRemote オブジェクトの配列を返します。この配列の各要素は、*remoteTargets* リストにリストされたリモート Windows ノードの位置に対応します。WinRemote オブジェクトは、たとえば実行が完了しているか、エラーが発生しているか、または実行が正常に完了しているかなどのいくつかの結果を確認できます。WinRemote オブジェクトを使用してリモート ノードが指定されている場合、配列には元の RemoteTarget オブジェクトへの参照が含まれます。ターゲットシステムが文字列によって指定されている場合、新しい RemoteTarget オブジェクトが内部に作成され、配列に割り当てられます。

-host *hostname*

(オプション) リモート Windows ホストの名前を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

-cmd *cmd*

(オプション) リモートの **Windows** システムで実行するコマンドを指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。このパラメータは、公開鍵認証を使用して、オンザフライで **SSH** セッションを作成する場合にのみ必要です。

デフォルト : `cmd`

-user *username*

(オプション) リモートの **Windows** システムにログインするユーザ名を指定します。このユーザには、**\$ADMIN** 共有に接続する権限が必要です。ユーザ名を指定しない場合、**AutoShell** へのログイン時に入力されたユーザ名が使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : `$$User`

-pass *password*

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、**AutoShell** へのログイン時に入力されたパスワードが使用されます。このコマンドを発行した後に非表示でパスワードを入力するには、引数としてオプションのプロンプトと **PW_GET()** を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : `$$Pass`

-workDir workdir

実際の作業ディレクトリを指定します。

デフォルト: ""

on remoteTargets

指定したスクリプトを実行するリモートシステムのカンマ区切りリストを指定します。ターゲットノードは、ホスト名/アドレスと WinRemote オブジェクトの 2 つの方法で指定できます。WinRemote オブジェクトを使用するときに名前またはアドレスを使用してターゲットシステムを指定すると、たとえば別のホストに別の認証情報を使用するなど、より柔軟に実行プロセスを制御できます。リモートターゲットリストには、ホスト名/アドレスを持つ WinRemote オブジェクトおよび文字列を混合させることができます。ターゲットリストの要素は自動的に文字列化されません。したがって、ホスト名用のリテラル文字列を使用する場合は、引用符で囲む必要があります。

例

host1 で cmd.exe (コマンド シェル) を実行するには、次のコマンドを実行します。

```
run-winRemote -host host1 -cmd cmd.exe
```

host1 および host2 で mybatch.bat を実行し、host2 の結果を確認するには、次のコマンドを実行します。

```
x = run-winRemote -cmd mybatch.bat -localpath c:%installations%myinstall.exe on  
"host1", "host2";  
? x[1].hasCompleted();  
? x[1].result();  
? x[1].output();
```

関連項目:

[push-winRemote コマンド -- 複数のリモート Windows システム上でのコピーと実行 \(Funclet\)](#) (P. 97)

[push-client コマンド -- Autoshell クライアントをリモート Windows システムにインストール \(Funclet\)](#) (P. 94)

[run-client コマンド -- リモート Windows システム上で Autoshell クライアントを実行 \(Funclet\)](#) (P. 106)

[rem-client コマンド -- リモート Windows システムから Autoshell クライアントを削除 \(Funclet\)](#) (P. 103)

set alternate to コマンド -- 別の出力ファイルを設定する (Cmdlet)

このコマンド (cmdlet) は、別の出力ファイル (?、??、qout()、または qqout()) を設定またはリセットします。別の出力ファイルを設定すると、コマンドまたは関数からの任意の出力は **stdout** に書き込まれるだけでなく、指定されたファイルにも書き込まれます。別の出力ファイルのみに書き込むには、「**set console off**」コマンドを使用して **stdout** を非表示にします。

コマンドの構文は、以下のとおりです。

```
set alternate to outFile [-append]
```

outFile

別の出力ファイルの絶対パスまたは相対パスを持つ文字列を指定します。ファイル名が指定されていない場合、コマンドは別の出力ストリームを前の設定にリストアします。

-append

(オプション) 指定されたファイルに出力ストリームを追加します。

例

1 ～ 10 までの数字を **output.txt** とコンソールに書き込みます。

```
set alternate to output.txt
for(i=1; i<11; i++)
{
    ? i
}
set alternate to
```

11 ～ 20 までの数字を **output.txt** とコンソールに追加します。

```
set alternate to output.txt -append
for(i=11; i<21; i++)
{
    ? i
}
set alternate to
```

ファイルにディレクトリ リストを書き込みます。

```
set alternate to dir.txt
dir -l
set alternate to
```

関連項目

[? コマンド -- 出力を標準出力に書き込む \(Cmdlet\)](#) (P. 57)
[gout -- 出力に続いて改行を stdout に書き込む \(関数\)](#) (P. 140)
[?? コマンド -- リストを標準出力に書き込む \(Cmdlet\)](#) (P. 56)
[qqout -- stdout に出力を書き込む \(関数\)](#) (P. 141)
[set console コマンド -- コンソールへの出力を非表示にする \(Cmdlet\)](#) (P. 126)

set console コマンド -- コンソールへの出力を非表示にする (Cmdlet)

このコマンド (cmdlet) は、?、??、gout()、または qqout() によって生成された対話型モードのコンソールである stdout への出力を非表示にするために使用されます。通常、ファイルに書き込む場合に set alternate コマンドと一緒に使用されます。

コマンドの構文は、以下のとおりです。

```
set console {on|off}
```

on

出力を stdout に書き込みます。

off

コンソールへの出力を非表示にします。

例

コンソールへの出力を非表示にして、1 ～ 10 までの数字を output.txt に書き込みます。

```
set console off
set alternate to output.txt
for(i=1; i<11; i++)
{
    ? i
}
set alternate to
set console on
```

関連項目

[set alternate to コマンド -- 別の出力ファイルを設定する \(Cmdlet\)](#) (P. 125)

set result display on/off コマンド -- 自動結果表示をオンまたはオフにする (Cmdlet)

このコマンドは結果表示をオンまたはオフにします。

コマンドの構文は、以下のとおりです。

```
set result disp {on|off}
```

関連項目

[dir、ls コマンド -- ファイルおよびディレクトリ情報の取得 \(Funclet\)](#) (P. 71)

start-service コマンド -- リモートまたはローカル システムの Windows サービスを開始する (Funclet)

start-service コマンドは、ローカルまたはリモート システムの Windows サービスを開始します。

コマンドの構文は、以下のとおりです。

```
start-service -svcName sName [-host hostname] [-user username] [-pass password]
```

このコマンドは、成功した場合は 1、失敗した場合は 0 を返します。

-svcName *sName*

サービスの名前を指定します。

-host *hostname*

(オプション) リモート サーバの名前です。

デフォルト: ""

-user *username*

(オプション) リモート ノードへのログインに使用するユーザ名を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-pass *password*

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

例

ローカルの Apache サービスを開始するには、次のコマンドを実行します。

```
? start-service -svcName Apache2.2
```

リモート サーバ REMOTESRV の Microsoft SQL Server サービスを開始するには、次のコマンドを実行します。

```
? start-service -svcName MSSQLSERVER -host REMOTESRV -user admin -pass notsecret
```


stop-service コマンド -- リモートまたはローカル システムの Windows サービスを停止する (Funclet)

stop-service コマンドは、ローカルまたはリモート システムの Windows サービスを停止します。

コマンドの構文は、以下のとおりです。

```
stop-service -svcName sName [-host hostname] [-user username] [-pass password]
```

このコマンドは、成功した場合は **1**、失敗した場合は **0** を返します。

-svcName *sName*

サービスの名前を指定します。

-host *hostname*

(オプション) リモート サーバの名前です。

デフォルト : ""

-user *username*

(オプション) リモート ノードへのログインに使用するユーザ名を指定します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

-pass *password*

(オプション) リモート ノードへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

デフォルト : ""

例

ローカルの **Apache** サービスを停止するには、次のコマンドを実行します。

```
? stop-service -svcName Apache2.2
```

リモート サーバ **REMOTESRV** の **Microsoft SQL Server** サービスを停止するには、次のコマンドを実行します。

```
? stop-service -svcName MSSQLSERVER -host REMOTESRV -user admin -pass notsecret
```

start-java コマンド -- Java 仮想マシンを起動する (Cmdlet)

このコマンドは **Java** 仮想マシンを起動します。

コマンドの構文は、以下のとおりです。

```
start-java [-classpath cp]
```

cp

(オプション) クラス パスを指定します。

デフォルト : ""

wait コマンド -- キーが押されるのを待機する(Cmdlet)

このコマンドでは次のバージョンが提供されています。

- オプションでプロンプトを表示し、キーが押されるのを待機し、押されたキーと等しい文字を文字列変数に割り当てます。
- オプションでプロンプトを表示し、キーが押されるのを待機します。

コマンドの構文は、以下のとおりです。

```
wait [prompt] to var  
wait [prompt]
```

prompt

(オプション) 入力プロンプトとして表示される文字列です。プロンプトを指定しない場合、コマンドは自動的に「キーをどれか押してください...」を表示します。プロンプトを表示しない場合は、空の文字列("")を渡します。引用符で囲まれていない引数トークンは自動的に文字列に変換されます。式のコードは丸かっこ内に配置して、式が自動的に引用符で囲まれないようにします。

var

押されたキーの文字を割り当てる変数の名前です。

例

キーが押されるのを待機し、カスタム プロンプト メッセージを表示します。

```
do  
{  
    wait "Enter a number between 1 and 5:" to x  
    x = parseInt(x);  
}  
while(x<1 || x>5)
```

関連項目:

[accept コマンド -- 標準入力から文字列を読み取り、変数に割り当てる \(Cmdlet\)](#) (P. 58)

weak external コマンド -- ネイティブ外部関数を宣言する(Cmdlet)

このコマンドはネイティブ外部関数を宣言します。

コマンドの構文は、以下のとおりです。

```
weak external decl  
decl
```

宣言を指定します。

AutoShell 関数

このセクションでは、AutoShell の中心的な関数の詳細について説明します。関数には、0 個以上の引数を受け取り、値を返さない、または 1 つの値を返す、一連の処理手順が定義されています。関数の引数は丸かっこに囲んで渡され、JavaScript 言語の構文に準拠している必要があります。リテラル文字列は引用符内に配置して、文字列内部の特殊文字（円記号など）はエスケープする必要があります。

例

```
// s を n 回連結する  
function repeat(s,n)  
{  
    var i;  
    var ret="";  
    for(i=0; i < n; i++)  
        ret += s;  
    return(ret);  
}  
? repeat("*", 10);  
? repeat("¥¥", 25);
```

base64Decode -- base64 でエンコードされた文字列をデコードする(関数)

この関数は、**base64** でエンコードされた文字列のデコードを実行します。入力文字列に対して **base64Encode()** を実行する必要はありません。**base64** でエンコードされた有効な文字列であれば、デコードのために渡すことができます。

この関数の構文は以下のとおりです。

base64Decode(sEncoded)

この関数は、デコードされたデータを保持する文字列を返します。入力データの形式が不正である場合などのエラーが発生すると、関数は例外を発生させます。

sEncoded

デコードする文字列を指定します。

例

文字列をデコードします（エラー処理も行います）。

```
try
{
    ? base64Decode("SGVsbG8gV29ybGQh");
}
catch(ex)
{
    ? "Error decoding string"
}
```

関連項目

[base64Encode -- 文字列を base64 でエンコーディングする \(関数\)](#) (P. 134)

base64Encode -- 文字列を base64 でエンコーディングする(関数)

この関数は、入力データを **base64** でエンコーディングします。

この関数の構文は以下のとおりです。

```
base64Encode(sData)
```

この関数は、**sData** で渡されたデータを **base64** でエンコードし、それを保持する文字列を返します。結果の文字列は 1 行が 76 文字の長さにフォーマットされます。

sData

エンコードする文字列を定義します。

例

「Hello World」をエンコードします。

```
? base64Encode("Hello World!")
```

関連項目

[base64Decode -- base64 でエンコードされた文字列をデコードする \(関数\)](#)
(P. 133)

curDir -- 現在のディレクトリを取得する(関数)

この関数は現在のディレクトリを取得します。

この関数の構文は以下のとおりです。

```
curDir()
```

この関数は、現在の作業ディレクトリの絶対パスを保持する文字列を返します。

例

親ディレクトリに移動し、そのパスを取得します。

```
chdir ..  
? curDir()
```

関連項目

[cd、chdir コマンド -- ディレクトリの変更 \(Cmdlet、Funclet\)](#) (P. 62)

gete -- 環境変数を取得する (関数)

環境変数の値を取得します。これは、標準の JavaScript 環境変数以外の環境変数を取得する代替手段として機能します。

この関数の構文は以下のとおりです。

`gete(sVar)`

この関数は、`sVar` によって指定された環境変数の現在の値を保持する文字列を返します。変数が存在しない場合は、空の文字列が返されます。

`sVar`

環境変数を指定します。

例

現在のパス設定を取得します。

```
? gete("PATH");
```

関連項目

[pute -- 環境変数を設定する \(関数\)](#) (P. 138)

memoRead -- 文字列にテキストファイルを読み込む(関数)

memoRead() 関数は、テキスト ファイルを文字列に読み込みます。

注: バイナリ情報が含まれるファイルに対して memoRead() を使用すると、予測できない結果が発生します。

この関数の構文は次のとおりです。

```
memoRead(sFileName [,nOffset] [,nCount])
```

この関数は、テキスト ファイルの内容を JavaScript の文字列として返します。

sFilename

テキスト ファイル名を指定します。指定されたファイルが存在しない場合、この関数は例外を生成します。

nOffset

(オプション) テキスト ファイルの先頭から読み取りプロセスの開始点までの文字数を指定します。

nCount

(オプション) 文字列に読み込む文字数を指定します。

例

Windows の起動設定を読み取ります。

```
s=memoRead("c:\boot.ini");  
? s
```

オフセット 10 からファイルの終わりまでテキスト ファイルを読み取ります。

```
s = memoRead("file.txt", 10);
```

テキスト ファイルをオフセット 10 から 5 文字読み取ります。

```
s = memoRead("file.txt", 10, 5);
```

関連項目

[memoWrit -- 文字列をファイルに書き込む \(関数\)](#) (P. 137)

[cat、type コマンド -- テキスト ファイルの表示 \(Cmdlet\)](#) (P. 61)

memoWrit -- 文字列をファイルに書き込む (関数)

memoWrit() 関数は、JavaScript の文字列の内容をファイルに書き込みます。

注: 指定したファイルが存在する場合は、既存のファイルを置換するか、文字列を既存の文字列の内容に追加できます。

この関数の構文は次のとおりです。

```
memoWrit(sFileName, sString [, bAppend])
```

sString

文字列を定義します。

sFileName

テキスト ファイルの名前を定義します。指定されたファイルが存在しない場合、この関数は例外を生成します。

, bAppend

true に設定すると、この関数は指定されたファイルに文字列を追加します。

この関数は、指定されたファイルに書き込まれたバイト数を返します。

注: ファイルへのアクセスに問題がある場合、この関数は例外を生成します。

例

テキスト ファイルに 1 行を書き込みます。

```
memoWrit("file.txt", "First line¥");
```

既存のファイルに 2 行目を追加します。

```
memoWrit("file.txt", "Second line¥", true);
```

関連項目

[memoRead -- 文字列にテキスト ファイルを読み込む \(関数\)](#) (P. 136)

platform -- オペレーティング システム タイプを照会する(関数)

この関数は、現在スクリプトを実行しているオペレーティング システムのタイプを返します。

この関数の構文は以下のとおりです。

```
platform()
```

この関数は、システム タイプ名を文字列で返します。

例

スクリプトが **Windows** 上で実行されていることを確認します。

```
if(platform()!="Windows")
{
    ? "This script uses Windows specific features!"
    ? "Please run on Windows only."
    quit
}
start-service -svcName Apache2.2
```

関連項目

[shellType -- AutoShell タイプを照会する \(関数\)](#) (P. 157)

pute -- 環境変数を設定する(関数)

既存の環境変数の値を設定するか、変数が存在しない場合は指定された値で環境変数を作成します。これは、標準の **JavaScript** 環境変数以外の環境変数を設定する代替手段として機能します。

この関数の構文は以下のとおりです。

```
pute(sVar, sVal)
```

この関数は、成功または失敗を示す **true** または **false** を返します。

sVar

環境変数の名前を指定します。

sVal

環境変数の値を定義します。

例

パスに現在の作業ディレクトリを追加します。

```
pute(gete("PATH")+";"+curDir());
```

関連項目

[gete -- 環境変数を取得する \(関数\)](#) (P. 135)

qout -- 出力に続いて改行を stdout に書き込む (関数)

qout 関数は、出力に続いて改行を **stdout** に書き込みます。この関数は、リスト内の各引数の文字列表現を **stdout** に書き込みます。各引数の間に 1 つのスペースが自動的に配置されます。**set console** および **set alternate** コマンドを使用して、出力を非表示にしたり、出力をリダイレクトすることができます。出力は改行文字で終了します。

この関数の構文は次のとおりです。

```
qout([x,...])
```

x

(オプション) 表示する引数のリストを定義します。リストが空の場合は改行文字が出力されます。

この関数は値を返しません。

例

"Hello World!" を出力します。

```
? "Hello World!"
```

数字のリストを出力します。

```
? 1,2,3
```

空の行を出力します。

```
?
```

1 ~ 10 の数字を出力します。

注: ? コマンドは、行の初めのみで使用できます。式の中からテキストを出力するには、対応する **qout()** 関数を使用します。

```
for(i=1;i<11;i++) qout(i);
```

関連項目

[?? コマンド -- リストを標準出力に書き込む \(Cmdlet\)](#) (P. 56)

[qqout -- stdout に出力を書き込む \(関数\)](#) (P. 141)

qqout -- stdout に出力を書き込む (関数)

qqout 関数は、リスト内の各引数の文字列表現を **stdout** に書き込みます。各引数の間に 1 つのスペースが自動的に配置されます。**set console** および **set alternate** コマンドを使用して出力をリダイレクトしたり、出力を非表示にすることができます。**?**、**??**、**qout()**、または **qqout()** を使用して呼び出した後の出力は、元の出力の直後に出力されます。そのため、**??** または **qqout()** は、通常はいくつかの呼び出しを使用して出力行を作成するために使用されます。

この関数の構文は次のとおりです。

```
qqout([x,...])
```

x

(オプション) 表示する引数のリストを定義します。リストが空の場合は改行文字が出力されます。

この関数は値を返しません。

例

分離スペースを持つ 3 つの列ヘッダを出力します。

```
?? "Col1"  
?? " "  
?? "Col2"  
?? " "  
? "Col3"
```

3 番目のラベルは、**?** 関数を使用して出力されます。そのため、カーソルは次の行の先頭に移動します。

関連項目

[? コマンド -- 出力を標準出力に書き込む \(Cmdlet\)](#) (P. 57)

[qout -- 出力に続いて改行を stdout に書き込む \(関数\)](#) (P. 140)

[?? コマンド -- リストを標準出力に書き込む \(Cmdlet\)](#) (P. 56)

[set alternate to コマンド -- 別の出力ファイルを設定する \(Cmdlet\)](#) (P. 125)

[set console コマンド -- コンソールへの出力を非表示にする \(Cmdlet\)](#) (P. 126)

regCreateKey -- レジストリ キーを作成する(関数)

`regCreateKey` 関数は、ローカル システムまたはリモート システムの `sKeyParent` によって指定されたレジストリ キーの下に、キーまたはキー階層を作成します。

この関数の構文は以下のとおりです。

`regCreateKey(sKeyParent, sKey)`

`sKeyParent`

親キーを指定します。

`sKey`

新しいキーを定義します。

この関数が失敗すると、例外が発生します。それ以外の場合は `true` が返されます。指定されたキーまたはキー階層の一部が存在する場合、この関数は成功します。

例

ローカル マシンの **SOFTWARE** キーの下に **ACEM** というキーを作成します (エラー処理も行います)。

```
try {
    regCreateKey("HKLM\\SOFTWARE", "ACME");
    ? "Success"
}
catch(e) {
    ? "Failure", e
}
```

キー階層を作成します。

```
regCreateKey("HKLM\\SOFTWARE", "ACME\\HOME");
```

リモート マシン `client8` に **ACEM** というキーを作成します。

```
regCreateKey("client8::HKLM\\SOFTWARE", "ACME");
```

関連項目

[regSetVal -- レジストリの値を設定する \(関数\)](#) (P. 155)

[regDeleteKey -- レジストリ キーまたはキー階層を削除する \(関数\)](#) (P. 144)

regCreateSubkeys -- 配列からサブキーを作成する(関数)

regCreateSubKeys 関数は、ローカル システムまたはリモート システムの配列の指定されたキーの下に、キーのサブツリーを作成します。配列には、**sKey** で始まる相対パス文字列として作成されるキーの名前が含まれる必要があります。 **regGetSubKeys()** は、このタイプの配列を返します。

この関数の構文は以下のとおりです。

```
regCreateSubKeys(sKey, aSubKeys)
```

sKey

レジストリ キーを指定します。

aSubKeys

キーのサブツリーを指定します。

この関数が失敗すると、例外が発生します。それ以外の場合は **true** が返されます。

例

HKLM¥Software¥ACME の下に、'x'、'y'、および 'z' というキーを作成します。

```
regCreateSubKeys("HKLM¥Software¥ACME", ["x", "y", "z"]);
```

入れ子のサブキー ツリーを作成します。

```
regCreateSubKeys("HKLM¥Software¥ACME", ["a¥b¥c", "x¥y¥z"]);
```

リモート システム **client8** からサブキー ツリーを取得し、それをリモート システム **client9** に適用します。

```
arr = regGetSubKeys("client8::HKLM¥Software¥ACME", true);  
regSetSubKeys("client9::HKLM¥Software¥ACME", arr);
```

注: キーの値は個別に処理する必要があります。

関連項目

[regCreateKey -- レジストリ キーを作成する \(関数\)](#) (P. 142)

[regGetSubKey -- レジストリ キーのサブキーを取得する \(関数\)](#) (P. 148)

regDeleteKey -- レジストリ キーまたはキー階層を削除する(関数)

`regDeleteKey` 関数は、ローカル システムまたはリモート システムのレジストリから、キーおよび関連するすべての値を削除します。指定されたキーにサブキーが含まれる場合、デフォルトでは削除は失敗します。ただし、この関数は、*fRecurse* に **true** を渡すことによって、入れ子のすべてのサブキーを含む *sKey* を削除します。

この関数の構文は次のとおりです。

```
regDeleteKey(sKey [, fRecurse])
```

sKey

レジストリ キーを指定します。

, fRecurse

入れ子のサブキーを削除します。

この関数が失敗すると、例外が発生します。それ以外の場合は **true** が返されます。

例

HKLM¥Software¥ACME の「Settings」という名前のキーを削除します(エラー処理を行います)。

```
try {
    regDeleteKey("HKLM¥Software¥ACME¥Settings");
    ? "Success"
}
catch(e) {
    ? "Failure:", e
}
```

サブキーに対して強制的に再帰的削除を実行して「Settings」キーを削除します。

```
regDeleteKey("HKLM¥Software¥ACME¥Settings", true);
```

リモートマシン `client8` の「Settings」というキーを削除します。

```
regDeleteKey("client8::HKLM¥Software¥ACME¥Settings");
```


関連項目

[regCreateKey -- レジストリ キーを作成する \(関数\)](#) (P. 142)

[regDeleteVal -- レジストリの値を削除する \(関数\)](#) (P. 145)

regDeleteVal -- レジストリの値を削除する(関数)

regSetVal 関数は、ローカル システムまたはリモート システムの *sKey* によって指定されたレジストリ キーの値を削除します。

この関数の構文は次のとおりです。

regDeleteVal(sKey, sValName)

sKey

レジストリ キーを指定します。

sValName

値を指定します。

この関数が失敗すると、例外が発生します。それ以外の場合は **true** が返されます。

例

HKLM¥Software¥ACME の「Label」という名前の値を削除します（エラー処理を行います）。

```
try {
    regDeleteVal("HKLM¥Software¥ACME", "Label");
    ? "Success"
}
catch(e) {
    ? "Failure:", e
}
```

リモートマシン **client8** の値を削除します。

```
regDeleteVal("client8::HKLM¥Software¥ACME", "Name");
```

関連項目

[regDeleteKey -- レジストリ キーまたはキー階層を削除する \(関数\)](#) (P. 144)

[regSetVal -- レジストリの値を設定する \(関数\)](#) (P. 155)

regGetKeyValues -- レジストリ キーの値情報を取得する(関数)

regGetKeyValues 関数は、ローカルまたはリモート システムのキーの名前、実際の値、およびレジストリの値のタイプを持つ二次元配列を返します。返された配列を **regSetKeyValues()** に渡し、システム間で値の設定をコピーすることができます。

この関数の構文は次のとおりです。

regGetKeyValues(sKey)

sKey

レジストリ キーを指定します。

この関数は二次元配列を返します。 トップレベルの各要素は、値の名前を表す文字列、設定のポリモーフィック値、および値タイプの整数を含む 3 つの要素のサブ配列を保持します。レジストリの値のタイプによって戻り値のタイプが決まります。

- **REG_DWORD** 値は数値として返されます
- **REG_SZ** および **REG_EXPAND_SZ** は文字列として返されます
- **REG_MULTI_SZ** は文字列の配列として返されます
- **REG_EXPAND_SZ** の値は、現在の環境変数を使用してすでに展開されています
- **REG_BINARY** の値は AutoShell の CharArr オブジェクトとして返されます。これは、JavaScript 文字列変数がバイナリ データを正しく処理できないためです。

この関数が失敗すると、例外が発生します。

例

HKLM¥Software¥ACME¥Settings キーの値情報を取得して表示します。

```
arr = regGetKeyValues("HKLM¥Software¥ACME¥Settings");
l = arr.length;
for(i=0; i < l; i++)
{
  ? "Name: ", arr[i][0]
  ? "Value:", arr[i][1]
  ?? "Type: "
  switch(arr[i][2])
  {
    case REG_SZ:
      ? "REG_SZ"
      break;
    case REG_EXPAND_SZ:
      ? "REG_EXPAND_SZ"
      break;
    case REG_BINARY:
      ? "REG_BINARY"
      break;
    case REG_DWORD:
      ? "REG_DWORD"
      break;
    case REG_MULTI_SZ:
      ? "REG_MULTI_SZ"
      break;
    default:
      ? "Unknown"
  }
}
```

リモート システム client8 から値を取得し、リモート システム client9 にその値を適用します。

```
arr = regGetKeyValues("client8:HKLM¥Software¥ACME¥Settings");
regSetKeyValues("client9:HKLM¥Software¥ACME¥Settings", arr);
```

関連項目

[regGetVal -- レジストリの値を取得する（関数）](#) (P. 150)

[regSetKeyValues -- 配列のレジストリ キーの値を設定する（関数）](#) (P. 154)

regGetSubKey -- レジストリ キーのサブ キーを取得する(関数)

regGetSubKeys 関数は、ローカル システムまたはリモート システムの指定されたレジストリ キーのサブ キーを配列で返します。デフォルトでは、直接のサブキーのみが返されます。オプションで、すべてのサブキーを再帰的に取得できます。

この関数の構文は以下のとおりです。

```
regGetSubKeys(sKey, fRecurse)
```

sKey

レジストリ キーを指定します。

, fRecurse

入れ子のサブキーを取得します。

この関数は、**sKey** のサブキーの名前を保持する配列を返します。個々のサブキー名が **sKey** から開始される相対パスとして返されます。返された配列を **regCreateSubKeys()** に渡し、システム間でサブキー ツリーをコピーできます。

この関数が失敗すると、例外が発生します。

例

HKLM¥Software¥CA のサブキーを取得します（エラー処理を行います）。

```
try {
    var arr, i, l;
    arr = regGetSubKeys("HKLM¥Software¥CA");
    l = arr.length;
    for(i=0; i < l; i++)
    {
        ? arr[i]
    }
}
catch(e) {
    ? "Failure:", e
}
```

再帰的に HKLM¥Software¥CA のサブキーをすべて取得します。

```
arr = regGetSubKeys("HKLM¥Software¥CA", true);  
l = arr.length;  
for(i=0; i < l; i++)  
{  
    ? arr[i]  
}
```

リモート システム **client8** からサブキー ツリーを取得し、それをリモート システム **client9** に適用します。

```
arr = regGetSubKeys("client8:HKLM¥Software¥ACME", true);  
regSetSubKeys("client9:HKLM¥Software¥ACME", arr);
```

注: キーの値は個別に処理する必要があります。

関連項目

[regCreateKey -- レジストリ キーを作成する \(関数\)](#) (P. 142)

[regCreateSubkeys -- 配列からサブキーを作成する \(関数\)](#) (P. 143)

regGetVal -- レジストリの値を取得する(関数)

regGetVal 関数は、ローカル システムまたはリモート システムの *key* によって指定されたレジストリ キーの下に *valName* の値の設定を取得します。

この関数の構文は次のとおりです。

```
regGetVal(sKey, sValName)
```

sKey

レジストリ キーを指定します。

sValName

値を指定します。

指定された値が存在する場合、この関数は指定された値の設定を表すポリモーフニック値を返します。 レジストリの値のタイプによって戻り値のタイプが決まります。

- **REG_DWORD** 値は数値として返されます
- **REG_SZ** および **REG_EXPAND_SZ** は文字列として返されます
- **REG_MULTI_SZ** は文字列の配列として返されます。
- **REG_EXPAND_SZ** の値は、現在の環境変数を使用してすでに展開されています。
- **REG_BINARY** の値は AutoShell の CharArr オブジェクトとして返されます。これは、JavaScript 文字列変数がバイナリ データを正しく処理できないためです。

指定されたキーまたは値が存在しない場合、この関数は例外を生成します。

例

HKLM¥Software¥ACME キーから「Number」という名前の値を取得します。

```
regGetVal("HKLM¥¥Software¥¥ACME", "Number");
```

HKLM¥Software¥ACME キーの「Label」という名前の値を取得します (エラー処理を行います)。

```
try {  
  regGetVal("HKLM¥¥Software¥¥ACME", "Label");  
  ? "Success"  
}  
catch(e) {  
  ? "Failure:", e  
}
```

リモートマシン client8 から値を取得します。

```
regGetVal("client8::HKLM¥¥Software¥¥ACME", "Name");
```

関連項目

[regSetVal -- レジストリの値を設定する \(関数\)](#) (P. 155)

[regGetKeyValues -- レジストリ キーの値情報を取得する \(関数\)](#) (P. 146)

regIsKey -- レジストリ キーの存在を確認する (関数)

regIsKey 関数は、指定されたレジストリ キーがローカルまたはリモートシステムに存在するかどうかを確認します。

この関数の構文は次のとおりです。

```
regIsKey(sKey)
```

sKey

レジストリ キーを指定します。

この関数は、**sKey** の存在を示すブール値を返します。

true

キーが存在することを示します。

false

キーが存在しないことを示します。

例

ローカル システムの HKLM¥SOFTWARE の下に ACME キーを配置します。

```
if(regIsKey("HKLM¥SOFTWARE¥ACME"))  
? "Key was found!"  
else  
? "Key does not exist"
```

リモート システム ascl1 で同じ確認を実行します。

```
if(regIsKey("ascl1::HKLM¥SOFTWARE¥ACME"))  
? "Key was found"  
else  
? "Key does not exist"
```

関連項目

[regCreateKey -- レジストリ キーを作成する \(関数\)](#) (P. 142)

[regIsVal -- レジストリの値の存在を確認する \(関数\)](#) (P. 153)

regIsVal -- レジストリの値の存在を確認する(関数)

regIsVal 関数は、指定されたレジストリの値がローカルまたはリモート システムに存在するかどうかを確認します。

この関数の構文は次のとおりです。

regIsVal(sKey, sValName)

sKey

レジストリ キーを指定します。

sValName

値の名前を指定します。

この関数は、**sValName** の存在を示すブール値を返します。

true

値が存在することを示します。

false

値が存在しないことを示します。

例

HKLM¥SOFTWARE¥ACME の値 'home' が存在することを確認します。

```
if(regIsVal("HKLM¥SOFTWARE¥ACME", "home"))
{
    ? "Value was found"
}
else
{
    ? "Value does not exist"
}
```

関連項目

[regGetVal -- レジストリの値を取得する \(関数\)](#) (P. 150)

[regIsKey -- レジストリ キーの存在を確認する \(関数\)](#) (P. 151)

[regSetVal -- レジストリの値を設定する \(関数\)](#) (P. 155)

regSetKeyValues -- 配列のレジストリ キーの値を設定する(関数)

regSetKeyValues() 関数は、ローカルまたはリモート システムの二次元配列から、キーの値を作成または設定します。配列のトップレベルの各要素は、値の名前を表す文字列、設定する **JavaScript** のポリモーフィックの値、およびオプションで作成または設定する値のタイプを指定する整数を持つサブ配列を含む必要があります。**regGetKeyValues()** は、このタイプの配列を返します。

この関数の構文は次のとおりです。

```
regSetKeyValues(sKey, aVals)
```

sKey

レジストリ キーを指定します。

sVals

値を含む配列を定義します。

この関数が失敗すると、例外が発生します。それ以外の場合は **true** が返されます。

例

HKLM\Software\ACME に 'Number' および 'Label' という値を設定します。

```
vals = [{"Number", 42}, {"Label", "Hello"}];  
regSetKeyValues("HKLM\Software\ACME\Settings", vals);
```

明示的にタイプを指定して値を設定します。

```
vals = new Array(2);  
val[0] = ["MyPath", "%Path%;c:\MyDir", REG_EXPAND_SZ];  
val[1] = ["Number", 42, REG_DWORD];  
regSetKeyValues("HKLM\Software\ACME\Settings", vals);
```

リモート システム **client8** から値を取得し、リモート システム **client9** にその値を適用します。

```
arr = regGetKeyValues("client8:HKLM\Software\ACME\Settings");  
regSetKeyValues("client9:HKLM\Software\ACME\Settings", arr);
```

関連項目

[regGetKeyValues -- レジストリ キーの値情報を取得する \(関数\)](#) (P. 146)

[regSetVal -- レジストリの値を設定する \(関数\)](#) (P. 155)

regSetVal -- レジストリの値を設定する(関数)

`regSetVal` は、ローカル システムまたはリモート システムの *sKey* によって指定されたレジストリ キーの値を作成または設定します。 *sValName* によって識別される値が存在しない場合は、その値が作成されます。それ以外の場合は、既存の設定がその *値*によって上書きされます。

この関数の構文は次のとおりです。

```
regSetVal(sKey, sValName, value, [type])
```

sKey

レジストリ キーを指定します。

sValName

値の名前を指定します。

値

値を定義します。

type

値のタイプを指定します。

この関数が失敗すると、例外が発生します。 それ以外の場合は `true` が返されます。

例

HKLM¥Software¥ACME キーの REG_DWORD 値の「Number」という名前の値に 42 を設定します。

```
regSetVal("HKLM¥Software¥ACME", "Number", 42);
```

HKLM¥Software¥ACME キーの REG_SZ 値の「Label」という名前の値に「Hello」を設定します（エラー処理を行います）。

```
try {
    regSetVal("HKLM¥Software¥ACME", "Label", "Hello");
    ? "Success"
}
catch(e) {
    ? "Failure:", e
}
```

REG_EXPAND_SZ 値を設定します。

```
regSetVal("HKLM¥Software¥ACME", "MyPath", "%Path%;c:¥MyDir", REG_EXPAND_SZ);
```

REG_MULTI_SZ 値を設定します。

```
arr=["Alice", "Bob", "Carol"];
regSetVal("HKLM¥Software¥ACME", "People", arr);
```

リモートマシン client8 に文字列の値を設定します。

```
regSetVal("client8::HKLM¥Software¥ACME", "Name", "Bob");
```

関連項目

[regCreateKey -- レジストリ キーを作成する（関数）](#) (P. 142)

[regDeleteVal -- レジストリの値を削除する（関数）](#) (P. 145)

[regGetVal -- レジストリの値を取得する（関数）](#) (P. 150)

[regSetKeyValues -- 配列のレジストリ キーの値を設定する（関数）](#) (P. 154)

setProcExitCode -- AutoShell 終了コードを設定する(関数)

この関数は、ユーザ スクリプトによって **AutoShell** の終了を設定し、使用可能な定義済みの **AutoShell** 終了コードを上書きします。Windows システムでは、**AutoShell** が終了した後に、`%errorlevel%` 変数を使用して終了コードを取得できます。

この関数の構文は以下のとおりです。

```
setProcExitCode(iExitCode)
```

iExitCode

終了コードを指定します。

例

AutoShell 終了コードを **-42** に設定し、**AutoShell** を終了します。

```
setProcExitCode(-42);  
quit
```

関連項目

[exit、quit コマンド -- AutoShell セッションの終了 \(Cmdlet\)](#) (P. 74)

shellType -- AutoShell タイプを照会する(関数)

この関数は、スクリプトが現在マネージャ **AutoShell** の内部で実行されているのか、またはクライアント **AutoShell** の内部で実行されているのかを判断するために使用されます。クライアント **AutoShell** は、マネージャ **AutoShell** のすべての機能を提供してはおりません（たとえば、リモート実行機能はありません）。`shellType()` は、スクリプトが目的の **AutoShell** バージョンの内部で実行されていることを確認するために使用できます。

この関数の構文は以下のとおりです。

```
shellType()
```

この関数は、現在アクティブな **AutoShell** のタイプを示す文字列を返します。

- manager
- client

例

AutoShell のタイプを確認し、クライアント シェルの内部で実行されている場合は終了します。

```
if(shellType()!="manager")
{
    ? "This script requires AutoShell manager!"
    quit
}
```

関連項目

[platform -- オペレーティング システム タイプを照会する \(関数\)](#) (P. 138)

typeName -- 式のタイプ名を取得する(関数)

この関数は、任意の式のタイプ名を取得します。具体的には、オブジェクトの場合は定義するクラスの名前を返します。

この関数の構文は以下のとおりです。

```
typeName(val)
```

この関数は、渡された値のタイプ名を保持する文字列を返します。単純なタイプの場合は、以下のいずれかの文字列を返します。

- Boolean
- function
- null
- number
- string
- void
- xml

オブジェクトの場合は、定義するクラスの名前を返します。

```
val
```

式を定義します。

例

いくつかの式のタイプを取得します。

```
? typeName(19) // 数値
d = new Date;
? typeName(d) // 日付
a = ["a","b","c"];
? typeName(a) // 配列
? typeName(a[0]) // 文字列
```

関連項目

[gete -- 環境変数を取得する \(関数\)](#) (P. 135)

[platform -- オペレーティングシステムタイプを照会する \(関数\)](#) (P. 138)

AutoShell クラス

このセクションでは、AutoShell クラス定義の詳細について説明します。AutoShell は JavaScript を基にしています。そのため、JavaScript と同様に関数オブジェクトを使用してオブジェクト指向を実現しています。関数は、通常の関数、クラス、コンストラクタ、またはメソッドとして機能します。

オブジェクトは、new 演算子を使用して関数インスタンスとして作成します。以下に例を挙げます。

```
now = new Date();
```

また、AutoShell は、new 演算子を使用してインスタンス化する独自のクラスを実装しています。多くの場合、AutoShell クラスは AutoShell コマンド定義を使用してカプセル化されているため、AutoShell クラスを直接操作する必要はありません。

コンストラクタは、オブジェクトを作成するときに呼び出されるメソッドです。複数のコンストラクタが用意されているクラスもあります。この場合、呼び出されるコンストラクタは、オブジェクトのインスタンス化の実際の引数によって変わります。

メソッドは、「」演算子を使用してオブジェクト上で呼び出される、オブジェクトに関連付けられた一連の関数です。オブジェクト データへのアクセスや操作には this プロパティを使用します。

JavaScript 言語リファレンスも参照してください。

例

現在の時刻から 10 時間後の日付と時間を算出します。午前 0 時をまたいだ場合も、日付は適切に処理されます。

```
d = new Date();  
d.setHours(d.getHours()+10)
```

関連項目

[OSRedirect クラス](#) (P. 160)

[RemoteTarget クラス](#) (P. 173)

OSRedirect クラス

OSRedirect は、ローカルでスクリプトを実行する `run-local` コマンドと共に使用するユーティリティクラスです。主に、子プロセスの出力を JavaScript 文字列にリダイレクトするために使用します。!! コマンドで使用するシステム変数 `$$stdout` も OSRedirect クラスのインスタンスです。

コンストラクタ

OSRedirect()

メソッド

bool clear()

bool errorOccurred()

string errout()

bool hasCompleted()

string output()

int result()

イベントハンドラ

void onCompleted(fCompleted)

void onError(fError)

void receivedErrOutput(s)

void receivedOutput(s)

void receivedResult(rc)

関連項目

[OSRedirect.OSRedirect コンストラクタ](#) (P. 161)

[OSRedirect.clear メソッド](#) (P. 162)

[OSRedirect.errorOccurred メソッド](#) (P. 163)

[OSRedirect.errout メソッド](#) (P. 164)

[OSRedirect.hasCompleted メソッド](#) (P. 165)

[OSRedirect.onCompleted メソッド](#) (P. 166)

[OSRedirect.onError メソッド](#) (P. 167)

[OSRedirect.output メソッド](#) (P. 168)

[OSRedirect.receivedErrOutput メソッド](#) (P. 169)

[OSRedirect.receivedOutput メソッド](#) (P. 170)

[OSRedirect.receivedResult メソッド](#) (P. 171)

[OSRedirect.result メソッド](#) (P. 172)

[run-local コマンド -- ローカル システム上でスクリプトを実行 \(Funclet\)](#) (P. 108)

OSRedirect.OSRedirect コンストラクタ

コンストラクタは、オブジェクトを作成するときに呼び出します。

コンストラクタの構文は、以下のとおりです。

```
OSRedirect()
```

コンストラクタは値を返しません。エラーが発生した場合は例外が発生されます。

例

OSRedirect オブジェクトを作成し、呼び出し内で使用します。

```
out = new OSRedirect();
! ping asclil -output out
? out.output();
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

OSRedirect.clear メソッド

このメソッドは、**OSRedirect()** オブジェクトの出力バッファとエラー出力バッファをリセットします。このメソッドは **!** および **!!** コマンドで内部的に呼び出されるため、複数の **OS** 呼び出しで同じ **OSRedirect()** インスタンスを使用している場合は明示的に呼び出す必要はありません。

このメソッドの構文は以下のとおりです。

```
clear()
```

このメソッドは値を返しません。

例

```
out = new OSRedirect();  
! ping asclil -output out  
? out.output();  
out.clear();  
? out.output();    // 出力は空
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

OSRedirect.errorOccurred メソッド

このメソッドは、リモート実行中にエラーが発生したかどうかを確認します。エラーが発生している場合、`errout()` はエラーの説明を提供する場合があります。

このメソッドの構文は以下のとおりです。

```
errorOccurred()
```

このメソッドは以下の値を返します。

true

エラーが発生したことを示します。

false

エラーが発生しなかったことを示します。

例

無効なコマンドを実行してエラー チェックを行います。

```
!! asdfghjk
if($$stdout.errorOccurred())
    ? "エラー: ", $$stdout.errout()
else
    ? $$stdout.output()
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.errout メソッド](#) (P. 164)

OSRedirect.errout メソッド

子プロセスは、以下のタイプの結果情報を生成します。

- 画面に書き込まれる標準出力
- 画面に書き込まれるエラー出力
- 最終終了コード (errorlevel)

このメソッドの構文は以下のとおりです。

```
errout()
```

このメソッドは、子プロセスによって標準エラー出力に書き込まれた出力を文字列で返します。

例

nslookup を呼び出し、標準出力とエラー出力を表示します。

```
!! nslookup ascli1  
? $$stdout.output()  
? $$stdout.errout()
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.errorOccurred メソッド](#) (P. 163)

[OSRedirect.output メソッド](#) (P. 168)

OSRedirect.hasCompleted メソッド

このメソッドは、リモート コマンドの実行が完了したかどうかを確認します。オプションの `-wait` スイッチを指定せずに `run-remote` を呼び出すと、非同期で実行されます。コマンドが完了していて結果を利用できるかどうかを確認するには、このメソッドを呼び出します。コマンドが正常に完了した場合、メソッド `hasCompleted()` は `true` を返します。その後の処理の妨げとなるエラーが発生した場合は、`false` を返します。

このメソッドの構文は以下のとおりです。

`hasCompleted()`

このメソッドは以下の値を返します。

true

コマンドの処理が完了したことを示します。

false

ホストが見つからないなど、コマンドの処理が完了していないことを示します。

例

```
!! dir c:\¥*. * /s
? $$stdout.hasCompleted()    // true
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\) \(P. 55\)](#)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\) \(P. 54\)](#)

OSRedirect.onCompleted メソッド

イベントハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベントハンドラが存在する場合は、子プロセスが完了すると呼び出されます。

このメソッドの構文は以下のとおりです。

```
onCompleted(fCompleted)
```

注: AutoShell はイベントハンドラの戻り値を無視します。

例

完了イベントハンドラを指定します。

```
out = new OSRedirect();
out.onCompleted = function(f){if(f)qout("ret = ", this.result());};
! ver -output out
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.result メソッド](#) (P. 172)

OSRedirect.onError メソッド

イベント ハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベント ハンドラが存在する場合は、**OSRedirect** オブジェクトのエラー状態が変更されると呼び出されます。

このメソッドの構文は以下のとおりです。

```
onError(fError)
```

注: **AutoShell** はイベント ハンドラの戻り値を無視します。

例

完了イベント ハンドラを指定します。

```
out = new OSRedirect();
out.onError = function(f){if(f)qout("エラーが発生しました。");};
! ver -output out
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.errorOccurred メソッド](#) (P. 163)

OSRedirect.output メソッド

子プロセスは、次の 3 つのタイプの結果情報を生成します。通常は画面に書き込まれる標準出力、通常は画面に書き込まれるエラー出力、および最終終了コード (**errorlevel**)。このメソッドは、子プロセスによって標準出力に書き込まれた出力を文字列で返します。

このメソッドの構文は以下のとおりです。

```
output()
```

このメソッドは、コマンド出力が含まれる文字列を返します。

例

Windows の OS バージョン文字列を取得します。

```
!! ver  
? $$stdout.output()
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.errout メソッド](#) (P. 164)

[OSRedirect.result メソッド](#) (P. 172)

OSRedirect.receiveErrOutput メソッド

イベント ハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベント ハンドラが存在する場合は、子プロセスが標準エラー出力に書き込むと呼び出されます。このハンドラは複数回呼び出せます。

このメソッドの構文は以下のとおりです。

```
receiveErrOutput(s)
```

注: AutoShell はイベント ハンドラの戻り値を無視します。

例

受信したエラー出力を表示します。

```
out = new OSRedirect();
out.receiveErrOutput = function(s){qout(s)};
! ver -output out
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.receiveOutput メソッド](#) (P. 170)

OSRedirect.receivedOutput メソッド

イベント ハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベント ハンドラが存在する場合は、子プロセスが標準出力に書き込むと呼び出されます。このハンドラは複数回呼び出せます。

このメソッドの構文は以下のとおりです。

```
receivedOutput(s)
```

注: AutoShell はイベント ハンドラの戻り値を無視します。

例

受信した出力を表示します。

```
out = new OSRedirect();  
out.receivedOutput = function(s){qout(s)};  
! ver -output out
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.receivedErrOutput メソッド](#) (P. 169)

[OSRedirect.receivedResult メソッド](#) (P. 171)

OSRedirect.receiveResult メソッド

イベント ハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベント ハンドラが存在する場合は、子プロセスが終了コードを設定すると呼び出されます。

このメソッドの構文は以下のとおりです。

```
receiveResult(rc)
```

注: AutoShell はイベント ハンドラの戻り値を無視します。

例

イベント ハンドラからの終了コードを表示します。

```
out = new OSRedirect();
out.receiveResult = function(rc){qout("終了コード: ", rc);};
! ver -output out
```

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\)](#) (P. 55)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\)](#) (P. 54)

[OSRedirect.receiveOutput メソッド](#) (P. 170)

OSRedirect.result メソッド

子プロセスは、次の 3 つのタイプの結果情報を生成します。通常は画面に書き込まれる標準出力、通常は画面に書き込まれるエラー出力、および最終終了コード (errorlevel)。

このメソッドの構文は以下のとおりです。

```
result()
```

このメソッドは、子プロセスの終了コードを返します。

例

Windows の ping は終了コードを、指定したホストが応答した場合は 0 に、応答しなかった場合は 1 に設定します。結果を確認し表示します。

```
!! ping ascll1
if($?stdout.result()==0)
    ? "Ping 成功"
else
    ? "Ping 失敗"
```

注: アプリケーション固有のエラーの場合、OSRedirect() の errorOccurred() フラグは設定されません。

関連項目

[! コマンド -- コマンドまたは子プロセスの呼び出し \(Cmdlet\) \(P. 55\)](#)

[!! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 \(Cmdlet\) \(P. 54\)](#)

[OSRedirect.output メソッド \(P. 168\)](#)

RemoteTarget クラス

RemoteTarget は、リモートでスクリプトを実行する **run-remote** コマンドと共に使用するユーティリティ クラスです。**RemoteTarget** では、ホスト名、アドレス、認証に使用する認証情報などのターゲット システムの詳細を指定したり、エラー状態、出力、最終結果などのステータスや結果情報を取得したりします。

コンストラクタ

```
RemoteTarget(host [,user] [,pass] [,key] [,phrase] [, port])
```

メソッド

```
void abort()  
int createRemoteContext()  
int destroyRemoteContext()  
bool errorOccurred()  
bool execute(strScript)  
int getError()  
string getHostName()  
bool hasCompleted()  
string output()  
string result()  
bool wasAborted()
```

イベント ハンドラ

```
int onError(e)  
int receivedOutput(s)  
int receivedResult(s)
```

変数

```
static readonly REM_ERR_NONE  
static readonly REM_ERR_CON  
static readonly REM_ERR_AUTH  
static readonly REM_ERR_CHN  
static readonly REM_ERR_READ  
static readonly REM_ERR_EXEC
```

static readonly REM_ERR_ABORT

static readonly REM_ERR_DISC

関連項目

[RemoteTarget.RemoteTarget コンストラクタ](#) (P. 175)

[RemoteTarget.abort メソッド](#) (P. 177)

[RemoteTarget.createRemoteContext メソッド](#) (P. 178)

[RemoteTarget.destroyRemoteContext メソッド](#) (P. 179)

[RemoteTarget.errorOccurred メソッド](#) (P. 180)

[RemoteTarget.execute メソッド](#) (P. 181)

[RemoteTarget.getError メソッド](#) (P. 182)

[RemoteTarget.getHostName メソッド](#) (P. 184)

[RemoteTarget.hasCompleted メソッド](#) (P. 185)

[RemoteTarget.onError メソッド](#) (P. 186)

[RemoteTarget.output メソッド](#) (P. 187)

[RemoteTarget.receivedOutput メソッド](#) (P. 188)

[RemoteTarget.receivedResult メソッド](#) (P. 188)

[RemoteTarget.result メソッド](#) (P. 189)

[RemoteTarget.wasAborted メソッド](#) (P. 191)

[run-remote コマンド -- リモート システム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

RemoteTarget.RemoteTarget コンストラクタ

コンストラクタは、オブジェクトを作成するときに呼び出します。コンストラクタは値を返しません。エラーが発生した場合は例外が発生されます。

コンストラクタの構文は、以下のとおりです。

```
RemoteTarget(host [,user] [,pass] [,key] [,phrase] [, port])
```

host

リモート ホストの名前を指定します。

user

(オプション) リモート ノードへのログインに使用するユーザ名を指定します。ユーザ名はパスワードまたは公開鍵認証に必要です。ユーザ名を指定しない場合、AutoShell へのログイン時に入力されたユーザ名が使用されます。

デフォルト : \$\$User

pass

(オプション) リモート ホストへのログインに使用するパスワードを指定します。パスワードを指定しない場合、AutoShell へのログイン時に入力されたパスワードが使用されます。

デフォルト : \$\$Pass

key

(オプション) リモート ホストへのログインに使用する、RSA の秘密鍵が保存されているファイルへの絶対パスまたは相対パスを指定します。秘密鍵とパスワードが指定された場合、AutoShell はまず公開鍵でログオンしようとします。公開鍵でのログオンに失敗した場合は、パスワードでログオンします。

デフォルト : ""

phrase

(オプション) 秘密鍵のパスフレーズを指定します。キーが暗号化されていない場合、パスフレーズは不要です。

デフォルト : ""

port

(オプション) リモート ホストに接続するポートを指定します。

デフォルト : SSH の標準ポート 22。

例

パスワード認証を使用して、RemoteTarget オブジェクトを作成します。

```
rt = new RemoteTarget("ascli1", "bob", "ca123456");
```

公開鍵認証を使用して RemoteTarget オブジェクトの作成を試行し、公開鍵認証が失敗した場合はパスワードを使用します。

```
rt = new RemoteTarget("ascli1", "bob", "ca123456", "bob_rsa.key");
```

この場合、ファイル bob_rsa.key に RSA の秘密鍵が含まれています。

関連項目

[run-remote コマンド -- リモート システム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

RemoteTarget.abort メソッド

リモート スクリプトは、デフォルトでは非同期で実行されます。 未完了の操作を中止する必要がある場合は、リモート操作に関連付けられている **RemoteTarget** オブジェクトで **abort()** を呼び出します。

このメソッドの構文は以下のとおりです。

```
abort()
```

このメソッドは値を返しません。

例

リモート操作を中止します。

```
rt = new RemoteTarget("ascli1");
run-remote Math.sin(Math.PI/2) on rt
if(!rt.hasCompleted())
{
    rt.abort()
    ? "操作を中止しました: ", rt.wasAborted()
}
```

関連項目

[run-remote コマンド -- リモート システム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

[RemoteTarget.wasAborted メソッド](#) (P. 191)

RemoteTarget.createRemoteContext メソッド

デフォルトでは、**RemoteTarget** オブジェクトでコマンドが実行されるたびに新しい **Autoshell** セッションが新規作成されます。同じターゲット上で複数のコマンドを実行する場合、永続的な単一のセッションを使用して複数のスクリプトを実行することをお勧めします。また、永続的なセッションでは状態が保持されます。たとえば、ある実行で作成された変数をその後の評価でも使用できます。

RemoteTarget オブジェクト上で **createRemoteContext()** を呼び出すと、リモート システム上に **Autoshell** セッションが新規作成されます。その後、**destroyRemoteContext()** が呼び出されるか **RemoteTarget** オブジェクトが回収されるまで、オープンな状態が維持されます。コンテキストが存在するときに **createRemoteContext()** を呼び出すと、既存のコンテキストを破棄して新しいコンテキストが作成されます。

このメソッドの構文は以下のとおりです。

```
createRemoteContext()
```

このメソッドは以下のいずれかのエラー コードを返します。

RemoteTarget.REM_ERR_NONE

成功を示します。

RemoteTarget.REM_ERR_CON

ホストが不明であるなど、接続エラーを示します。

RemoteTarget.REM_ERR_AUTH

認証エラーを示します。

例

永続的なセッションを作成し、最初の評価で変数を定義して、2 回目の評価でその変数を取得します。

```
rt=new RemoteTarget("ascli1");
if(rt.createRemoteContext()==RemoteTarget.REM_ERR_NONE)
{
    run-remote x=42 on rt
    run-remote x on rt -wait
    ? get-remoteResult(rt) // 42
    rt.destroyRemoteContext();
    ? get-remoteResult(rt) // 空
}
else
{
    ? "エラーが発生しました。";
}
```

関連項目

[RemoteTarget.destroyRemoteContext メソッド](#) (P. 179)

RemoteTarget.destroyRemoteContext メソッド

このメソッドは、前に `createRemoteContext()` で作成したコンテキストを破棄します。永続的なコンテキストを破壊した後も、リモート実行で `RemoteTarget` オブジェクトを引き続き使用できます。別のコンテキストが作成されるまで、スクリプトで評価されるたびにコンテキストは内部的に作成および破棄されます。

このメソッドの構文は以下のとおりです。

```
destroyRemoteContext()
```

このメソッドは以下のいずれかのエラー コードを返します。

`RemoteTarget.REM_ERR_NONE`

成功を示します。

`RemoteTarget.REM_ERR_DISC`

リモート システムから切断していることを示します。

例

永続的なセッションを作成し、最初の評価で変数を定義して、2 回目の評価でその変数を取得します。

```
rt=new RemoteTarget("ascli1");
if(rt.createRemoteContext()==RemoteTarget.REM_ERR_NONE)
{
    run-remote x=42 on rt
    run-remote x on rt -wait
    ? get-remoteResult(rt) // 42
    rt.destroyRemoteContext();
    ? get-remoteResult(rt) // 空
}
else
{
    ? "エラーが発生しました。";
}
```

関連項目

[RemoteTarget.createRemoteContext メソッド](#) (P. 178)

RemoteTarget.errorOccurred メソッド

このメソッドは、リモート実行中にエラーが発生したかどうかを確認します。エラーが発生している場合、`getError()` はさらに詳しい情報を提供します。

このメソッドの構文は以下のとおりです。

```
errorOccurred()
```

このメソッドは以下の値を返します。

true

エラーが発生したことを示します。

false

エラーが発生しなかったことを示します。

例

リモート実行の完了後にエラーを確認します。

```
rt = new RemoteTarget("ascli1");
run-remote Math.cos(0) on rt -wait
if(rt.errorOccurred())
{
    ? "エラーが発生しました。"
}
else
{
    ? get-remoteResult(rt)
}
```

関連項目

[RemoteTarget.getError メソッド](#) (P. 182)

RemoteTarget.execute メソッド

このメソッドは、**run-remote** コマンドで呼び出されるコア メソッドです。**run-remote** は、指定されたスクリプトに特定の処理を実行してからこのメソッドに渡すため、直接このメソッドを呼び出さないでください。

このメソッドの構文は以下のとおりです。

```
execute(strScript)
```

このメソッドは以下の値を返します。

true

非同期実行が正常に開始されたことを示します。

false

リモート実行を開始するときにエラーが発生したことを示します。

例

式を評価します（推奨の方法）。

```
rt = new RemoteTarget("ascli1");  
run-remote Math.sqrt(1) on rt  
? get-remoteResult(rt);
```

式を評価します（非推奨の方法）。

```
rt = new RemoteTarget("ascli1");  
rt.execute("Math.sqrt(1)");  
? get-remoteResult(rt);
```

関連項目

[run-remote コマンド -- リモート システム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

RemoteTarget.getError メソッド

このメソッドは、リモート実行が失敗した後、詳細なエラー情報を返します。

このメソッドの構文は以下のとおりです。

```
getError()
```

このメソッドは以下のいずれかのエラー コードを返します。

RemoteTarget.REM_ERR_NONE

成功を示します。

RemoteTarget.REM_ERR_CON

ホストが不明であるなど、接続エラーを示します。

RemoteTarget.REM_ERR_AUTH

認証エラーを示します。

RemoteTarget.REM_ERR_CHN

チャネルの確立でエラーが発生したことを示します。

RemoteTarget.REM_ERR_READ

リモート システムからの読み取りでエラーが発生したことを示します。

RemoteTarget.REM_ERR_EXEC

式の評価でエラーが発生したことを示します。

RemoteTarget.REM_ERR_ABORT

ユーザが操作を中止したことを示します。

RemoteTarget.REM_ERR_DISC

リモート システムからの切断でエラーが発生したことを示します。

例

詳細なエラー情報を表示します。

```
rt = new RemoteTarget("ascli1");
run-remote Math.sqrt(1) on rt -wait
switch(rt.getError())
{
    case RemoteTarget.REM_ERR_NONE:
        ? "Success"
        break;
    case RemoteTarget.REM_ERR_CON:
        ? "Error connecting"
        break;
    case RemoteTarget.REM_ERR_AUTH:
        ? "Error authenticating"
        break;
    case RemoteTarget.REM_ERR_CHN:
        ? "Error establishing channel"
        break;
```

```
case RemoteTarget.REM_ERR_READ:
    ? "Error reading from remote system"
    break;
case RemoteTarget.REM_ERR_EXEC:
    ? "Error evaluating expression"
    break;
case RemoteTarget.REM_ERR_ABORT:
    ? "Operation aborted by user"
    break;
case RemoteTarget.REM_ERR_DISC:
    ? "Error disconnecting from remote system"
    break;
default:
    ? "Unknown error"
}
```

関連項目

[RemoteTarget.errorOccurred メソッド](#) (P. 180)

[RemoteTarget.onError メソッド](#) (P. 186)

RemoteTarget.getHostName メソッド

このメソッドは、**RemoteTarget** オブジェクトを作成するときに指定したホスト名またはアドレスを返します。

このメソッドの構文は以下のとおりです。

```
getHostName()
```

このメソッドは、ターゲット ノードのホスト名またはアドレスの文字列を返します。

例

暗黙的に **RemoteTarget** オブジェクトを作成し、ホスト名を取得します。

```
aRT = run-remote "! ver" on "ascl11"
? aRT[0].getHostName()
```


関連項目

[run-remote コマンド -- リモート システム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

[RemoteTarget.RemoteTarget コンストラクタ](#) (P. 175)

RemoteTarget.hasCompleted メソッド

このメソッドは、リモート スクリプトの評価の実行状態を確認します。
abort() 呼び出しで実行が中止されている場合、処理中にエラーが発生している場合、またはターゲット ノード上でスクリプトが正常に実行されている場合は、実行は完了していると見なされます。 実行が正常に完了したのか、エラーが発生したのかを判断するには、**errorOccurred()** と **getError()** メソッドを使用します。

このメソッドの構文は以下のとおりです。

hasCompleted()

このメソッドは以下の値を返します。

true

実行は完了しています。

false

実行は完了していません。

例

手動で完了の待機を実装します。

```
aRT = run-remote "! ver" on "ascii1"
while(!aRT[0].hasCompleted())
{
    sleep(500);
}
if(aRT[0].errorOccurred())
{
    ? "Error"
}
else
{
    ? "Result:", get-remoteResult(aRT[0])
}
```

通常は、`run-remote` コマンドの `-wait` オプションのみをスクリプトで使用します。

関連項目

[run-remote コマンド -- リモートシステム上でスクリプトを実行 \(Funclet\)](#)
(P. 110)

[RemoteTarget.abort メソッド](#) (P. 177)

[RemoteTarget.errorOccurred メソッド](#) (P. 180)

[RemoteTarget.getError メソッド](#) (P. 182)

RemoteTarget.onError メソッド

イベントハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベントハンドラが存在する場合は、リモート実行時にエラーが発生すると呼び出されます。

このメソッドの構文は以下のとおりです。

```
onError(e)
```

注: **AutoShell** はイベントハンドラの戻り値を無視します。

例

RemoteTarget のエラー ハンドラを指定します。

```
rt = new RemoteTarget("ascli1");  
// イベント ハンドラの割り当て  
rt.onError = function(e){qout("エラーが発生しました。コード = ", e);};  
run-remote "2*2" on rt
```

関連項目

[RemoteTarget.errorOccurred メソッド](#) (P. 180)

[RemoteTarget.getError メソッド](#) (P. 182)

RemoteTarget.output メソッド

スクリプトは、2 つのタイプの結果情報を生成します。通常は画面に書き込まれる出力と、最終戻り値です。このメソッドは、リモートシステム上で実行したスクリプトによって生成された出力を文字列で返します。スクリプトが実行されている間、出力は累積されます。そのため、スクリプトの実行が完了する前にこのメソッドを呼び出すと、空でない結果が返されることがあります。

このメソッドの構文は以下のとおりです。

```
output()
```

このメソッドは、スクリプトが標準出力に書き込んだデータの文字列を返します。

例

Windows ホスト上で実行中のサービスをリモートで問い合わせます。

```
rt = new RemoteTarget("ascli1", "bob", "casogood42");  
run-remote "! sc query" on rt -wait  
? rt.output()
```

関連項目

[RemoteTarget.result メソッド](#) (P. 189)

RemoteTarget.receivedOutput メソッド

イベント ハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベント ハンドラが存在する場合は、リモート ターゲットから出力を受信すると呼び出されます。このハンドラは複数回呼び出せます。

このメソッドの構文は以下のとおりです。

```
receivedOutput(s)
```

注: AutoShell はイベント ハンドラの戻り値を無視します。

例

RemoteTarget の出力ハンドラを指定し、関連付けられているオブジェクトにイベント ハンドラでアクセスします。

```
rt = new RemoteTarget("ascli1");  
// 出力ハンドラの割り当て  
rt.receivedOutput = function(s){  
    qout("受信した出力 = ", s);  
    qout("出力元のホスト = ", this.getHostName());  
};  
run-remote "! ver" on rt
```

関連項目

[RemoteTarget.output メソッド](#) (P. 187)

RemoteTarget.receivedResult メソッド

イベント ハンドラは、ユーザ スクリプトで実装できるオプションの関数で、特定のイベントが発生したときに **AutoShell** によって呼び出されます。このイベント ハンドラが存在する場合は、リモート ターゲットから最終結果の値を受信すると呼び出されます。

このメソッドの構文は以下のとおりです。

```
receivedResult(s)
```

注: AutoShell はイベント ハンドラの戻り値を無視します。

例

`RemoteTarget` の結果ハンドラを指定し、関連付けられているオブジェクトにイベント ハンドラでアクセスします。

```
rt = new RemoteTarget("ascli1");
// 出力ハンドラの割り当て
rt.receivedResult = function(s){
    qout("受信した結果 = ", s);
    qout("出力元のホスト = ", this.getHostName());
};
run-remote "Math.SQRT2" on rt
```

関連項目

[RemoteTarget.result メソッド](#) (P. 189)

RemoteTarget.result メソッド

スクリプトは、2 つのタイプの結果情報を生成します。通常は画面に書き込まれる出力と、最終戻り値です。このメソッドは、リモート実行プロセスの戻り値を表したものを取得します。結果を返すことがスクリプト評価における最後の手順であるため、スクリプトの実行が正常に完了した場合、このメソッドは有効な値のみを返します。`hasCompleted()` は `true` を返し、`errorOccurred()` は `false` を返します。リクエストを発行したシステムにネットワーク経由で転送する必要があるため、リモート実行の戻り値は XML 表現にシリアル化されます。`RemoteTarget.result()` メソッドは、シリアル化された表現で結果の値を返します。

シリアル化された表現は、`RemoteTarget` オブジェクト自体をパラメータとする `get-remoteResult()` funclet を使用して、標準の JavaScript 値に変換します。通常は、スクリプトで `result()` を直接呼び出す必要はありません。シリアル化によりタイプ情報が保持されます。たとえば、リモート ノードが数値を返すと、`get-remoteResult()` は数値を返します。リモート システムが `Date` オブジェクトを返すと、元のシステム上でも日付になります。

リモート スクリプトからの戻り値は、単純なデータ型に制限されません。リモート実行でサーバ間での大量のデータ転送を可能にし、複雑な配列やデータのみオブジェクトを返すこともできます。唯一の制限は、データ構造に循環参照を含むことができないということです。

このメソッドの構文は以下のとおりです。

```
result()
```

リモート スクリプトは、値を **XML** 表現にシリアル化した文字列を返します。

例

リモート実行のシリアル化された表現と実際の値を取得します。

```
rt = new RemoteTarget("ascli1", "bob", "ca123456");
run-remote "[1,2,3]" on rt -wait
? rt.result()
v = get-remoteResult(rt) // result() を内部で呼び出し
? typeof v             // 配列
? typeof v[0]          // 数値
arrdump v
```

関連項目

[get-remoteResult コマンド -- リモート ターゲットから結果を取得 \(Funclet\)](#)
(P. 78)

[RemoteTarget.output メソッド](#) (P. 187)

RemoteTarget.wasAborted メソッド

このメソッドは、プログラムの `abort()` が呼び出されてリモート スクリプトの実行が中止されているかどうかを確認します。

このメソッドの構文は以下のとおりです。

`wasAborted()`

このメソッドは以下の値を返します。

`true`

実行が中止されていることを示します。

`false`

実行が中止されていないことを示します。

例

リモート操作が中止されているかどうかを確認します。

```
rt = new RemoteTarget("ascli1");
run-remote Math.sin(3/2*Math.PI) on rt
if(!rt.hasCompleted())
{
    rt.abort()
    ? "操作を中止しました: ", rt.wasAborted()
}
```

関連項目

[RemoteTarget.abort メソッド](#) (P. 177)

AutoShell ロード可能モジュールのコマンドリファレンス

このセクションでは、オプションで使用する以下のプラットフォーム固有の AutoShell ロード可能モジュールに属するコマンドについて説明します。

- Microsoft Hyper-V
- Oracle Solaris ゾーン
- VMware vCenter Server

関連項目

[CA IBM LPAR AutoShell コマンド](#) (P. 352)

[CA Microsoft Hyper-V AutoShell コマンド](#) (P. 237)

[CA Oracle Solaris ゾーン AutoShell コマンド](#) (P. 383)

[CA VMware vCenter Server AutoShell コマンド](#) (P. 192)

CA VMware vCenter Server AutoShell コマンド

スクリプトに AutoShell を使用して CA VMware vCenter Server コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドを CLI でも利用できます。

dpmvc-addesxhost コマンド -- ESX ホスト(Funclet)の追加

dpmvc-addesxhost コマンドは、vCenter Server に ESX ホストを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-addesxhost
-esx_host_name esxhostname
-vc_server vcservername
-esx_host_user esxhostuser
-esx_host_password esxhostpassword
[-management_ip managementip]
[-host_connect_port hostconnectport]
[-folder_name foldername]
[-datacenter_name datacentername]
[-cluster_name clustername]
[-vim_account_user vimaccountuser]
[-vim_account_password vimaccountpassword]
```

-esx_host_name esxhostname

VM をホストする ESX サーバの名前を指定します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-esx_host_user esxhostuser

(オプション) ESX ホストにアクセスするユーザ名を指定します。

-esx_host_password *esxhostpassword*

ESX ホストへのアクセスに使用するパスワードを指定します。

-management_ip *managementip*

(オプション) 管理 IP を指定します。

-host_connect_port *hostconnectport*

(オプション) vCenter Server と通信するために ESX ホストで使用されるポートを指定します。

-folder_name *foldername*

(オプション) ESX ホストを追加できるフォルダの名前を指定します。

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-cluster_name *clustername*

(オプション) ESX ホストを追加できるクラスタの名前を指定します。

-vim_account_user *vimaccountuser*

(オプション) VIM アカウントのユーザ名を指定します。

-vim_account_password *vimaccountpassword*

(オプション) VIM ホストにアクセスするためのパスワードを指定します。

例: ESX ホストを追加する

この例では、「esx1」という ESX ホストを「myvcenter」という vCenter Server に追加します。

```
dpmvc-addesxhost -esx_host_name esx1 -vc_server myvcenter -esx_host_user admin  
-esx_host_password topsecret
```

dpmvc-addvmvdisk コマンド -- 仮想ディスクの追加 (Funclet)

dpmvc-addvmvdisk コマンドは、VM に仮想ディスクを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-addvmvdisk
-datastore_name datastorename
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
-capacity_in_mb disk_capacity
[-controller_key controllerkey]
[-disk_mode
{append|independent_nonpersistent|independent_persistent|nonpersistent|persistent
|undoable}]
-thin_provisioning {no|yes}
[-unit_number unitnumber]
```

-datastore_name *datastorename*

新しい VM が存在するデータ ストアの名前を指定します。

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-capacity_in_mb *disk_capacity*

仮想ディスクのサイズを MB で指定します。

-controller_key *controllerkey*

(オプション) ディスクのコントローラ キーを指定します。

-disk_mode {append | independent_nonpersistent | independent_persistent | nonpersistent | persistent | undoable}

(オプション) ディスクのモードを指定します。

-thin_provisioning {no | yes}

(オプション) ディスクがシンプロビジョニングである必要があるかどうかを示します。

-unit *disk_unitnumber*

(オプション) ディスクのユニット数を指定します。

例: 仮想ディスクを追加する

以下の例は、VM *myvm* に仮想ディスクを追加します。

```
dpmvc-addvmvdisk -datastore_name disk1 -vm_name myvm -vc_server myvcenterserver  
-capacity_in_mb 20000 -thin_provisioning no
```

dpmvc-addvmvnic コマンド -- 仮想 NIC の追加(Funclet)

dpmvc-addvmvnic コマンドは、VM に仮想 NIC を追加します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-addvmvnic  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-devicetype {e1000 | vmxnet}  
-network_name networkname  
[-mac_address macaddress]  
-wake_on_lan_enabled {no | yes}
```

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-devicetype {*e1000* | *vmxnet*}

ネットワーク デバイスのタイプを指定します。

-network_name *networkname*

NIC で接続する必要がある仮想ネットワークを指定します。以下の命名規則に基づいて、標準スイッチと分散仮想スイッチの名前を区別できます。

- 標準スイッチの名前はネットワーク名です。
- 分散仮想スイッチの名前は、**dvPort** グループ名の後に、丸かっこで囲まれた分散仮想スイッチ名を連結したもの (**dvPortGroupName** (**dvSwitchName**)) です。

-mac_address *macaddress*

(オプション) ネットワーク デバイスの **MAC** (Media Access Control) アドレスを指定します。

-wake_on_lan_enabled {no | yes}

LAN が有効な場合に VM を起動するかどうかを指定します。

例: vNICを追加する

以下の例は、VM に仮想 NIC を追加します。

```
dpmvc-addvmvnic -vm_name myvm -vc_server myvcenterserver -devicetype vmxnet  
-network_name net1 -wake_on_lan_enabled no
```

dpmvc-clone コマンド -- VM のクローン作成 (Funclet)

dpmvc-clone コマンドでは、VM のクローンを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc-clone
{-template_name templatename | -cloned_from vmname}
{-datacenter_name datacentername | -vc vcenterservername}
-vm_name vmname
-esx_host_name esxhostname
-datacenter_name datacentername
-data_store_name datastorename
-compute_resource_name name
-resource_pool_name resourcepool
-spec_name specificationname
[-auto_deploy value]
[-deploy_template templatename]
[-network_connection
"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_
gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask[,win_dns_server=wins_dns_
server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_sec
ondary"]
[-global_dns_search_suffix suffix1[,suffix2,suffix3,...]]
[-linux_domain_name domain_name]
[-linux_dns_servers primary_dns=value,secondary_dns=value,tertiary_dns=value]
[-create_disk MB,datastorename,controller]
[-modify_disk key, datastorename]
[-set_memory number]
[-set_cpu number]
-vm_os_username username
[-vm_os_password password]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-scalability_server scalabilityservername]

-datacenter_name datacentername
```

VM が存在するデータ センターを指定します。

-vc *vcenterservername*

VM が存在する vCenter Server を指定します。

-compute_resource_name *name*

VM を作成するクラスタまたは VMware ESX ホストの名前を指定しま
す。

-esx_host_name *esxhostname*

VM が存在する VMware ESX サーバを指定します。

-datastore_name *datastorename*

新しい VM が存在するデータストアの名前を指定します。

-vm_name *vmname*

VM を指定します。

-resource_pool_name *resourcepool*

クローンを作成する VM の選択元のリソースプールの名前を指定します。

-spec_name *specificationname*

(オプション) クローン作成する仮想マシンに使用する仕様の名前を指定します。

-auto_deploy {yes|no}

CA Virtual Assurance エージェントを自動的に展開するかどうかを指定します。オプションは、以下のとおりです。

yes

CA Virtual Assurance エージェントを自動的に展開します。

no

CA Virtual Assurance エージェントが自動的に展開されないようにします。

デフォルト : no

-deploy_template *templatename*

(オプション) CA Virtual Assurance で作成された汎用テンプレートの名前を指定します。

注: VMware vCenter で作成および管理されているテンプレートと、このテンプレートを混同しないでください。

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask,win_dns_server=wins_dns_server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_secondary"

Windows :

(オプション) VM のネットワーク接続を作成します。このパラメータは複数回指定できます。シーケンス番号は、ネットワーク接続を識別する 1 から始まる整数値です。スペースが含まれる場合は、パラメータ全体を丸かっこで囲む必要があります。nic、network_name、ip_address、def_gateway、subnet_mask 以外のすべての値は、値なしで名前を入力するとリセットできます (例: "nic=123,alt_gateway=")。

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask"

Linux :

(オプション) 1 つ以上のネットワーク接続を作成します。このパラメータは複数回指定できます。シーケンス番号は、ネットワーク接続を識別する 1 から始まる整数値です。スペースが含まれる場合は、パラメータ全体を丸かっこで囲む必要があります。値なしで名前を入力するとリセットできるのは、alt_gateway のみです (例: "nic=123,alt_gateway=")。

-global_dns_search_suffix suffix1{suffix2,suffix3,...}

Linux :

1 つ以上の DNS 検索サフィックスを指定します。

Windows :

(オプション) 1 つ以上の DNS 検索サフィックスを指定します。

-linux_domain_name domain_name

Linux :

(オプション) Linux VM のドメイン名を指定します。

`-linux_dns_servers`
`primary_dns=value{secondary_dns=value,tertiary_dns=value}`

Linux :

(オプション) Linux VM の DNS サーバの IP アドレスを指定します。値を省略すると、値の設定を解除できます (例 :
`primary_dns=123.123.123.13,secondary_dns=,tertiary_dns=123.123.123.15`
)。

`-create_disk MB,datastore_name,controller`

(オプション) 1 つ以上の追加のハードドライブを作成します。このパラメータは複数回指定できます。

MB

ハードドライブのサイズをメガバイトで定義します。

Datastore_name

このハードドライブのデータストアの名前を指定します。現在のところ、VM ごとに 1 つのデータストアのみがサポートされています。

Controller

このハードドライブ用のコントローラ キーを指定します。

`-modify_disk {key,datastore_name}`

(オプション) ハードドライブが存在するデータストアを指定します。複数指定できます。

key

既存のディスクを指定します。

datastore_name

このディスクを移すデータストアの名前を指定します。

`-set_memory number`

(オプション) クローン作成する VM のメモリの量をメガバイト (MB) で定義します。この値は、テンプレートで指定された値よりも優先されます。

`-set_cpu number`

(オプション) クローン作成する VM の CPU の数を定義します。この値は、テンプレートで指定された値よりも優先されます。

-vm_os_username *user name*

クローン作成する VM のユーザを指定します。イメージを自動展開する場合は、このユーザ名も認証に使用されます。

Windows : カスタマイズの仕様に定義したユーザ名である必要があります。

Linux : テンプレートで定義したユーザ名である必要があります。

-vm_os_password *password*

(オプション) クローン作成する VM のユーザのパスワードを指定します。イメージを自動展開する場合は、このパスワードも認証に使用されます。

Windows : カスタマイズの仕様に定義したパスワードと同じである必要があります。

Linux : テンプレートで定義したパスワードと同じである必要があります。

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、**dpmutil** 認証情報設定コマンドを使用して生成した **CA Virtual Assurance** 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループホストとユーザに対して使用できるコンポーネント ID を指定します。

-template_name *templatename*

クローン作成する仮想マシンに使用するテンプレートの名前を指定します。このパラメータは **-vm_name_cloned_from** を置換します。また、両方を指定することはできません。

-vm_name_cloned_from *vmname*

クローン作成する仮想マシンのテンプレートとして使用する仮想マシンの名前を指定します。このパラメータは **-template_name** を置換します。また、両方を指定することはできません。

-resource_pool_name *resourcepool*

クローンを作成する VM の選択元のリソースプールの名前を指定します。

-scalability_server *servername*

(オプション) ソフトウェア配信用の Unicenter DSM スケーラビリティ サーバを指定します。

-spec_name *specificationname*

(オプション) クローン作成する仮想マシンに使用する仕様の名前を指定します。

-compute_resource_name *name*

VM を作成するクラスタまたは VMware ESX ホストの名前を指定します。

例: グローバル認証情報を使用して、VM のクローンを作成する

以下の例は、テンプレート BaseW2k3 を使用して、データ センター VAS/MyCity に VM testvm01 を作成します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes  
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1  
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator  
-vm_os_password testpassword
```

例: 認証ファイルとコンポーネントを使用して VM のクローンを作成する

以下の例は、テンプレート BaseW2k3 を使用して、データ センター VAS/MyCity に VM testvm01 を作成します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes  
-spec_name w2k3_spec_01 -datacenter_name Specifies VAS/MyCity -datastore_name  
storage1  
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator  
-vm_os_password testpassword -auth_file c:\localauth.dat -auth_comp Imaging
```

例: グローバル認証情報を使用して VM のクローンを作成し、CPU を 2 つ作成する

以下の例は、テンプレート BaseW2k3 を使用してデータ センター VAS/MyCity に VM testvm01 を作成し、VM 用の CPU を 2 つ作成します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VPMcluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -set_cpu 2
```

例: デフォルトの認証ファイルを使用して VM のクローンを作成し、メモリを 4096 MB に設定する

以下の例は、テンプレート BaseW2k3 を使用してデータ センター VAS/MyCity に VM testvm01 を作成し、メモリを 4096 に設定します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VPMcluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -set_memory 4096
```

例: デフォルトの認証ファイルを使用して VM のクローンを作成し、ハード ディスクを 2 つ作成する

以下の例は、テンプレート BaseW2k3 を使用してデータ センター VAS/MyCity に VM testvm01 を作成し、データストア storage1 用のハードディスクを 2 つ作成します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VAScluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000
-create_disk 10000,storage1,2001
```

例: VM のクローンを作成し、ネットワーク接続を作成する

以下の例は、テンプレート BaseW2k3 を使用してデータ センター VAS/MyCity に VM testvm01 とネットワーク接続を作成し、初期値を設定します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VASCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -network_connection "nic=1,network_name=VM
Network,ip_address=123.321.5.22,def_gateway=123.321.1.108,subnet_mask=255.255.255
.0,win_dns_server=123.321.42.1,win_alt_dns=123.321.3.101,wins_primary=123.321.3.3
"
```

例: デフォルトの認証ファイルを使用して、電源がオフになっている VM のクローンを作成する

以下の例は、テンプレートの代わりに VM testvm02 を使用して、データ センター VAS/MyCity に VM testvm01 を作成します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -datacenter_name "VAS/MyCity" -vm_name testvm01 -datastore_name storage1
-esx_host_name vc1.ca.com -resource_pool_name Resources/DPMTest
-compute_resource_name DPMCluster/ussdCluster -vm_name_cloned_from testvm02
-auto_deploy yes -spec_name w2k3_spec_01 -vm_os_username administrator
-vm_os_password mypassword
```

例: デフォルトの認証ファイルを使用して VM のクローンを作成し、既存のハードディスクを変更する

以下の例は、テンプレート BaseW2k3 を使用してデータ センター VAS/MyCity に VM testvm01 を作成し、データストア storage1 用のハードディスクを 2 つ作成します。クローン操作が完了すると、CA Virtual Assurance エージェントは自動的に展開されます。

```
dpmvc-clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name VAS/MyCity -datastore_name storage1
-compute_resource_name VPMCluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000 -modify_disk
2000,storage2
```

dpmvc-createdatastore コマンド -- データストアの作成 (Funclet)

dpmvc-createdatastore コマンドは、データストアを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-createdatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
-datastore_type {local|nas|vmfs}
[-vmfs_device_path path]
[-vmfs_block_size_mb blocksize]
[-local_path lpath]
[-nas_user_name nasuser]
[-nas_password naspasswd]
[-nas_remote_hostname nasremhost]
[-nas_remote_path nsarempath]
[-nas_access_mode {read-only|read-write}]
[-sc sc_url]
```

-datastore_name *datastorename*

新しい VM が存在するデータ ストアの名前を指定します。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-datastore_type {vmfs|nas|local}

作成するデータストアのタイプ（VMFS、NAS、またはローカル）を指定します。

-vmfs_device_path *path*

（オプション）データストアを作成するときに使用する、VMFS ディスクのデバイスパスを指定します。 `getavailablescsidisks` コマンドを使用すると、このパスを取得できます。

-vmfs_block_size_mb *blocksize*

（オプション）データストアを作成するときに使用するブロック サイズを MB で指定します。利用可能なブロック サイズは 1、2、4、および 8 です。ブロック サイズに応じて、VM ディスク ファイルの最大サイズはそれぞれ 256 GB、512 GB、1024 GB、および 2048 GB となります。

-local_path *lpath*

（オプション）ローカル データストアを作成する場合、データストアの作成に使用した、ESX ホストのローカル ファイル システム パスをこのパスで指定します。

-nas_user_name *nasuser*

（オプション） NAS データストアの作成に使用するリモート ホスト ユーザ名を指定します。

-nas_password *naspasswd*

（オプション） NAS データストアの作成に使用するリモート ホスト パスワードを指定します。

-nas_remote_hostname *nasremhost*

（オプション） ネットワークベースのストレージをホストするサーバのホスト名を指定します。

-nas_remote_path *nasrempath*

（オプション） NAS データストアに使用する、リモート サーバのファイル システム パスを指定します。

-nas_access_mode *nasaccmode*

（オプション） NAS データストアに使用する、ネットワークベースのストレージへのアクセス権を指定します。利用可能なモードは `read-only` と `read-write` です。

デフォルト : "read-write"

dpmvc-cycle コマンド -- VM のサイクル(Funclet)

dpmvc-cycle コマンドは VM の電源オン、電源オフ、リセット、中断を行います。

このコマンドの形式は、以下のとおりです。

```
dpmvc-cycle
-powerop {poweron|poweroff|reset|suspend|shutdown guest}
{-datacenter_name datacentername | -vc vcenterservername}
-vm_name vmname

-powerop {poweron|poweroff|reset|suspend|shutdown guest}
```

VM 上で実行する電源操作を指定します。 オプションは、以下のとおりです。

poweron

VM の電源をオンにします。

poweroff

VM の電源をオフにします。

reset

VM をリセットします。

suspend

一時的に VM を停止します。

shutdown guest

ゲスト OS を正常にシャットダウンします。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vcenterservername*

VM が存在する vCenter Server を指定します。

-vm_name *vmname*

VM を指定します。

例: データセンターに複数の vCenter Server がある場合、VM の電源をオフにする

以下の例は、特定の vCenter Server から VM の電源をオフにします。

```
dpmvc-cycle -powerop poweroff -vc my_server1 -vm_name my_vm
```

例: 単一 vCenter Server 環境で VM の電源をオンにする

以下の例は、データセンター VAS/MyCity に属している VM testvm の電源をオフにします。

```
dpmvc-cycle -vm_name testvm -powerop poweron -datacenter_name VAS/MyCity
```

dpmvc-datastore コマンド -- データストア プロパティの取得 (Funclet)

dpmvc-datastore コマンドは、特定の VMware vCenter データストアの空き領域と容量設定を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-datastore  
{-datacenter_name datacentername | -vc vccenterservername}  
-datastore_name datastorename  
[-getproperty {all|capacity|freespace}]  
-getproperty {all|capacity|freespace}
```

(オプション) 取得するプロパティを指定します。オプションは、以下のとおりです。

all

データストアの容量と空き領域を取得します。

capacity

データストアの容量を取得します。

freespace

データストアの空き領域を取得します。

-datacenter_name *datacentername*

VM が存在するデータセンターを指定します。

-vc *vccenterservername*

VM が存在する vCenter Server を指定します。

`-datastore_name datastorename`

新しい VM が存在するデータ ストアの名前を指定します。

例: データストアのすべてのプロパティを取得する

以下の例は、データストアの空き領域と容量の両方を取得します。

```
dpmvc-datastore -getproperty all -datacenter_name VAS/MyCity -datastore_name storage1
```

dpmvc-delete コマンド -- VM の破棄(Funclet)

dpmvc-delete コマンドは、電源オフ状態の VM を削除します。このコマンドを使用して、未使用のリソースをクリーンアップし、解放します。

重要: このコマンドを発行する *前に*、重要なデータをバックアップしたことを確認してください。このコマンドは、ディスク イメージを含め、VM の VM データ ストアとデータ ファイルを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-delete  
{-datacenter_name datacentername | -vc vcenterservername}  
-vm_name vmname
```

`-datacenter_name datacentername`

VM が存在するデータ センターを指定します。

`-vc vcenterservername`

VM が存在する vCenter Server を指定します。

`-vm_name vmname`

VM を指定します。

例: 単一 vCenter Server 環境で VM を削除する

以下の例は、VM vm11 を削除します。

```
dpmvc-delete -vm_name vm11 -datacenter_name lab444
```

例: 複数 vCenter Server 環境で VM を削除する

以下の例は、VM vm11 を削除します。

```
dpmvc-delete -vm_name vm11 -datacenter_name lab444 -vc vc_server_1
```

dpmvc-distributedswitch コマンド -- 分散仮想スイッチの管理 (Funclet)

dpmvc-distributedswitch コマンドでは、分散仮想スイッチを管理できます。

- データ センターに新しい分散仮想スイッチを追加する
- 分散仮想スイッチのプロパティを更新する
- 分散仮想スイッチを削除する
- 分散仮想スイッチに分散ポート グループを新規作成する
- 分散仮想スイッチのポート グループ プロパティを更新する
- 分散仮想スイッチから分散ポート グループを削除する
- 分散仮想スイッチの分散ポート グループの名前を変更する

このコマンドの形式は、以下のとおりです。

```
dpmvc-distributedswitch {-vds_add | -vds_update}  
-vc_server vcservername  
-datacenter_name datacentername  
-vds_folder vdsfolder  
-switch_name switchname  
[-hostnics hostname1:nic1,nic2,...nicn [;hostname2:nic2,...nicn >]*]  
[-uplink_port_names uplink1[,uplink2,...,uplinkn]]  
[-maxports maxports]
```

```
dpmvc-distributedswitch -vds_remove  
-vc_server vcservername  
-switch_name switchname
```

```
dpmvc-distributedswitch -add_portgroup  
-vc_server vcservername  
-switch_name switchname  
-portgroup_name portgroupname  
[-bindtype earlyBinding | ephemeral | lateBinding]  
[-vlan vlanID]  
[-numports numberofports]
```

```
dpmvc-distributedswitch -update_portgroup  
-vc_server vcservername  
-portgroup_name portgroupname  
[-portgroup_newname portgroupnewname]  
[-bindtype earlyBinding | ephemeral | lateBinding]  
[-vlan vlanID]  
[-numports numberofports]
```

```
dpmvc-distributedswitch -remove_portgroup  
-vc_server vcservername  
-portgroup_name portgroupname
```

```
dpmvc-distributedswitch -rename_portgroup  
-vc_server vcservername  
-portgroup_name portgroupname  
-portgroup_newname portgroupnewname
```

```
-vc_server vcservername
```

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

```
-datacenter_name datacentername
```

分散仮想スイッチが属しているデータセンターを指定します。

```
-vds_folder vdsfolder
```

CA Virtual Assurance エクスプローラ ペイン内の分散仮想スイッチのフォルダを指定します。

```
-switch_name switchname
```

操作を実行するスイッチ名を指定します。

-hostnics *hostname1:nic1,nic2,...nicn* [*;hostname2:nic2,...nicn* >]

(オプション) ESX ホスト メンバに関連付けられている NIC のリストを指定します。

-uplink_port_names *uplink1[,uplink2,...,uplinkn]*

(オプション) 使用するアップリンク ポート名をカンマ区切りのリストで指定します。

-maxports *maxports*

(オプション) スイッチのポートの最大数を指定します。この数は、スイッチの全ポート グループのポート数 (numports) の合計が上限となります。

-bindtype *earlyBinding | ephemeral | lateBinding*

(オプション) ポート グループのバインド タイプを指定します。有効な値は以下のとおりです。

earlyBinding

VM がポートグループにバインドする際にポートを割り当てます。このバインディング タイプは、接続性の常時確保に有効ですが、ポートが永続的に予約されます。デフォルトはこのバインディング タイプです。

lateBinding

VM の電源がオンで NIC の状態が接続済みの場合、VM にポートを割り当てます。このバインディング タイプは、VM の電源がオフになるか、NIC が切断されると、ポートを割り当て直します。

LateBinding は vCenter で設定できます。

ephemeral

VM の電源がオンで NIC の状態が接続済みの場合、VM にポートを割り当てます。このバインディング タイプは、VM の電源がオフになるか、NIC が切断されると、ポートを割り当て直します。

ephemeral バインディングは、ESX ホストと vCenter で設定できます。

-numports *numberofports*

(オプション) ポート グループのポートの数を指定します。

-portgroup_name *portgroupname*

ポート グループ名を指定します。

-portgroup_newname *portgroupnewname*

新しいポートグループ名を指定します。

-vlan *vlanID*

(オプション) 仮想ポートグループの操作に使用する整数値 (VLAN ID) を指定します。

例: 分散仮想スイッチを新規作成する

以下の例は、分散仮想スイッチを新規作成します。

```
dpmvc-distributedswitch -vds_add -vc_server vc5master -datacenter_name dc3  
-vds_folder vds -switch_name vds1
```

例: 既存の分散仮想スイッチを更新する

以下の例は、分散仮想スイッチを更新します。 **MYSERVER1** の NIC を指定し、分散仮想スイッチから **MYSERVER2** を削除します。分散仮想スイッチからホストを削除する場合は、NIC を割り当てずにサーバ名指定します (**MYSERVER2:** のように、サーバ名の後にコロンを付けます)。

```
dpmvc-distributedswitch -vds_update -vc_server VAS-VC5 -datacenter_name VC5  
-switch_name vdistSwitch -hostnics MYSERVER1:nxmgt2,tmp2;MYSERVER2:
```

このコマンドは、以下のように **vdistSwitch** を更新します。

- ホスト **MYSERVER1** は NIC **nxmgt2** と **tmp2** を使用します。
- ホスト **MYSERVER2** はスイッチから削除されます。

dpmvc-entermaintenancemode コマンド -- 保守モードに設定する (Funclet)

dpmvc-entermaintenancemode コマンドは、ESX ホストを保守モードに設定します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-entermaintenancemode  
-esx_host_name esxhostname  
-vc_server vcservername  
-timeout timeout  
[-evacuate_powered_off_vm {no | yes}]  
  
-esx_host_name esxhostname
```

VM が存在する VMware ESX サーバを指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-timeout *timeout*

保守モードに入る、または保守モードを終了するまでのタイムアウト時間を指定します。

-evacuate_powered_off_vm {no | yes}

(オプション) VM 停止後に終了するかどうかを指定します。

例: 保守モードを開始する

以下の例は、ESX ホストを保守モードに設定します。

```
dpmvc-entermaintenancemode -esx_host_name esx1 -vc_server myvcserver -timeout 60
```

dpmvc-exitmaintenancemode コマンド -- 保守モードの終了 (Funclet)

dpmvc-exitmaintenancemode コマンドは、ESX ホストの保守モードを終了します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-exitmaintenancemode  
-esx_host_name esxhostname  
-vc_server vcservername  
-timeout timeout
```

-esx_host_name *esxhostname*

VM が存在する VMware ESX サーバを指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-timeout *timeout*

保守モードに入る、または保守モードを終了するまでのタイムアウト時間を指定します。

例: 保守モードを終了する

以下の例は、ESX ホスト `esx1` の保守モードを終了します。

```
dpmvc-exitmaintenancemode -esx_host_name esx1 -vc_server myvcserver  
-timeout 60
```

dpmvc-faulttolerance コマンド -- フォールトトレラント操作の指定 (Funclet)

`dpmvc-faulttolerance` コマンドは、フォールトトレランスのオン、オフ、有効化、無効化、テスト、またはセカンダリ VM のマイグレーションを行います。

このコマンドの形式は、以下のとおりです。

```
dpmvc-faulttolerance  
-turn_on -vm_name <value> [-esx_host_name <value>]|  
-turn_off -vm_name <value>|  
-enable -vm_name <value>|  
-disable -vm_name <value>|  
-test_ft -vm_name <value>|  
-migrate_secondary -vm_name <value> -esx_host_name <value>
```

-turn_on

指定した VM のフォールトトレランスをオンにします。

(オプション) セカンダリ VM の ESX ホスト名を指定します。

-turn_off

指定した VM のフォールトトレランスをオフにします。

注: この操作はセカンダリ VM を削除します。

-enable

指定した VM のフォールトトレランスを有効化します。

-disable

指定した VM のフォールトトレランスを無効化します。

-test_ft

指定した VM のフォールトトレランスをテストします。

-migrate_secondary

フォールトトレランス セカンダリ VM を別の ESX サーバにマイグレートします。 `-esx_host_name` オプションが必須です。

-vm_name

コンピュータ リソースの名前を定義します。

-esx_host_name

VM が存在する ESX サーバを指定します。

-vc_server

Virtual Center サーバ ホスト名を指定します。

dpmvc-getavailablescsidisks コマンド -- 利用可能な SCSI ディスクの取得 (Funclet)

dpmvc-getavailablescsidisks コマンドは、利用可能な SCSI ディスクをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-getavailablescsidisks  
[-datastore_name datastorename]  
-esx_host_name esxhostname  
-vc_server vcservername  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]
```

-datastore_name *datastorename*

新しい VM が存在するデータ ストアの名前を指定します。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

dpmvc-gethosthba コマンド -- ホストバスアダプタの取得(Funclet)

dpmvc-gethosthba コマンドは、指定された ESX ホスト上で設定されているホストバスアダプタのリストを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-gethosthba
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]

-esx_host_name esxhostname
```

VM が存在する VMware ESX サーバを指定します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user wsuser

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password wspassword

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

dpmvc-getresources コマンド -- VM リソースの取得(Funclet)

dpmvc-getresources コマンドは、CPU またはメモリの、共有制限または予約を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-getresources  
-vm_name vmname  
{-datacenter_name datacentername | -vc vcenterservername}  
[-resource {all|vm_cpulimit|vm_cpureserv|vm_memlimit|vm_memresrv}]
```

-vm_name *vmname*

VM を指定します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vcenterservername*

VM が存在する vCenter Server を指定します。

-resource {all|vm_cpulimit|vm_cpureserv|vm_memlimit|vm_memresrv}

特定のリソースを取得するのか、すべてのリソースを取得するのかを指定します。オプションは、以下のとおりです。

all

VM のすべての CPU、メモリ共有制限、予約を取得します。

vm_cpulimit

VM の CPU 共有の数の制限を取得します。

vm_cpureserv

VM の CPU 予約の共有の数を取得します。

vm_memlimit

VM のメモリ共有の数の制限を取得します。

vm_memresrv

VM のメモリ予約の共有の数を取得します。

例: 単一 vCenter Server 環境の VM のリソースをすべて取得する

以下の例は、usa-vm2 のリソースをすべて取得します。

```
dpmvc-getresources -resource all -datacenter_name MyCity -vm_name usa-vm2
```

dpmvc-getshares コマンド -- VM 共有の取得(Funclet)

dpmvc-getshares コマンドでは、VM に割り当てられている CPU またはメモリの共有の数を確認できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc-getshares  
{-datacenter_name datacentername | -vc vceterservername}  
-vm_name vmname  
[-share {cpu | memory}]  
-share {cpu | memory}
```

VM に割り当てられている CPU またはメモリの共有の数を表示します。共有を使用すると、利用できる物理リソースが、稼働中のほかの VM より比例的に多く割り当てられている VM を判断できます。VM により多くの共有を割り当てると、より多くの物理リソースがその VM に割り当てられます。VM に割り当てる共有を減らすと、その VM に割り当てられる物理リソースも少なくなります。

cpu

CPU 値を取得することを指定します。

memory

メモリ値を取得することを指定します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vceterservername*

VM が存在する vCenter Server を指定します。

-vm_name *vmname*

VM を指定します。

例: VM の CPU 値を取得する

以下の例は、usa-vm2 の CPU 値を取得します。

```
dpmvc-getshares -resource cpu -datacenter_name MyCity -vm_name usa-vm2
```

dpmvc-getsnapshots コマンド -- VM のスナップショットのリスト表示 (Funclet)

dpmvc-getsnapshots コマンドは、VM のスナップショットをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-getsnapshots  
{-datacenter_name datacentername | -vc vcenterservername}  
-vm_name vmname  
[-name_only]
```

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vcenterservername*

VM が存在する vCenter Server を指定します。

-vm_name *vmname*

VM を指定します。

-name_only

(オプション) スナップショット名のみを取得します。

例: スナップショットをリスト表示する

以下の例は、VM のスナップショットをリスト表示します。

```
dpmvc-getsnapshots -vc my_server1 -vm_name my_vm
```

dpmvc-imgjobcheck コマンド -- クローン作成ジョブ ステータスの取得 (Funclet)

dpmvc-imgjobcheck コマンドは、特定の CA Virtual Assurance ジョブ ID または特定の vCenter タスク ID のクローン作成ジョブ ステータスを取得します。ジョブ ID は、特定の dpmvc コマンドで使用します。このコマンドを使用して、vCenter のステータスを確認します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-imgjobcheck  
-status job ID  
[-vc vcenterservername]
```

-status *jobID*

ジョブステータスの取得に使用するジョブ ID を定義します。

-vc *vccenterservername*

VM が存在する vCenter Server を指定します。

例: 単一 vCenter Server 環境でジョブ ID を使用して、クローン作成ジョブステータスを取得する

以下の例は、CA Virtual Assurance ジョブ ID を使用してクローニングジョブステータスを取得します。

```
dpmvc-imgjobcheck -status 42
```

dpmvc-migrate コマンド -- VM のマイグレート(Funclet)

dpmvc-migrate コマンドは、あるホストサーバから別のホストサーバに VM をマイグレートします。

このコマンドの形式は、以下のとおりです。

```
dpmvc-migrate  
{-datacenter_name datacentername | -vc vccenterservername}  
-vm_name vm  
-migrate_to_target_host name  
-migrate_to_target_resource_pool_name name  
-migrate_to_compute_resource_name name
```

-datacenter_name *datacentername*

VM が存在するデータセンターを指定します。

-vc *vccenterservername*

VM が存在する vCenter Server を指定します。

-vm_name *vmname*

VM を指定します。

-migrate_to_target_host *name*

VM をマイグレートするホストの名前を指定します。

-migrate_to_target_resource_pool_name *name*

ターゲットリソースプールを指定します。

-migrate_to_compute_resource_name *name*

VM をマイグレートするクラスタまたは VMware ESX ホストの名前を指定します。

例: VM をマイグレートする

以下の例は、データ センター VPM/MyCity から ServerC-dcavc に VM MyVM1 をマイグレートします。

```
dpmvc-migrate -vm_name MyVM1 -datacenter_name VAS/MyCity
-migrate_to_compute_resource_name VASCluster/ClusterA
-migrate_to_target_resource_pool_name Resources/VASTest
-migrate_to_target_host ServerC-dcavc.MyCompany.com
```

dpmvc-removedatastore コマンド -- データストアの削除 (Funclet)

dpmvc-removedatastore コマンドは、データストアを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-removedatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
-datastore_name datastorename
```

新しい VM が存在するデータ ストアの名前を指定します。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

dpmvc-removevmvdisk コマンド -- 仮想ディスクの削除(Funclet)

dpmvc-removevmvdisk コマンドは、VM から仮想ディスクを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc-removevmvdisk  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-disk_device_key diskdevicekey  
-delete_data {yes | no}
```

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-disk_device_key *diskdevicekey*

ディスクのデバイス キーを整数で指定します (例 : 2000) 。

-delete_data {yes | no}

ディスクを削除する際にデータも削除するかどうかを指定します。

例: 仮想ディスクを削除する

以下の例は、VM myvm からデバイス キー 2000 の仮想ディスクを削除します。

```
dpmvc-removevmvdisk -vm_name myvm -vc_server myvcserver -disk_device_key 2000  
-delete_data yes
```

dpmvc-removevmvnic コマンド -- 仮想 NIC の削除 (Funclet)

dpmvc-removevmvnic コマンドは、VM から仮想 NIC を削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc removevmvnic  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-device_key devicekey
```

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-device_key *devicekey*

ネットワーク インターフェースのデバイス キーを整数で指定します
(例：3000)。

例: vNIC を削除する

以下の例は、VM から vNIC を削除します。

```
dpmvc-removevmvnic -vm_name myvm -vc_server myvcserver -device_key 3000
```

dpmvc-setresources コマンド -- VM リソースの設定 (Funclet)

dpmvc-setresources コマンドでは、CPU またはメモリの、共有制限または予約を調整できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc-setresources  
-resource {setcpulimit|setcpureserv|setmemlimit|setmemresrv}  
-value value  
{-datacenter_name datacentername | -vc vcservername}  
-vm_name vmname
```

setcpulimit

VM の CPU 共有の数の制限を定義します。

setcpureserv

VM の CPU 予約の共有の数を定義します。

setmemlimit

VM のメモリ共有の数の制限を定義します。

setmemresrv

VM のメモリ予約の共有の数を定義します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vccenterservername*

VM が存在する vCenter Server を指定します。

-vm_name *vmname*

VM を指定します。

-value *value*

リソース パラメータの値を指定します。

例: 単一 vCenter Server 環境でリソースの CPU 制限を 1000 共有に設定する

以下の例は、仮想マシン MyVM1 の CPU 共有制限を 1000 に調整します。

```
dpmvc-setresources -resource setcpulimit -value 1000 -vm_name MyVM1 -datacenter_name VAS/MyCity
```

dpmvc-setshares コマンド -- VM 共有の設定 (Funclet)

dpmvc-setshares コマンドでは、VM に割り当てられている CPU またはメモリの共有の数を設定できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc-setshares  
{-datacenter_name datacentername | -vc vccenterservername}  
-vm_name vmname  
-setshare {cpu_add_prop_value | cpu_subtract_prop_value |  
cpu_overwrite_prop_value | mem_add_prop_value | mem_subtract_prop_value |  
-mem_overwrite_prop_value}  
-value value
```

`-setshare {cpu_add_prop_value | cpu_subtract_prop_value |
cpu_overwrite_prop_value | mem_add_prop_value |
mem_subtract_prop_value | mem_overwrite_prop_value}`

追加、削除、変更する CPU またはメモリの共有の数を定義します。共有を使用すると、利用できる物理リソースが、稼働中のほかの VM より比例的に多く割り当てられている VM を判断できます。VM により多くの共有を割り当てると、より多くの物理リソースがその VM に割り当てられます。VM に割り当てる共有を減らすと、その VM に割り当てられる物理リソースも少なくなります。

`cpu_add_prop_value`

VM に追加する CPU 共有の数を定義します。

`-cpu_subtract_prop_value value`

VM から削除する CPU 共有の数を定義します。

`-cpu_overwrite_prop_value value`

VM で使用する CPU 共有の数を定義します。

`-mem_add_prop_value value`

VM に追加するメモリ共有の数を定義します。

`-mem_subtract_prop_value value`

VM から削除するメモリ共有の数を定義します。

`-mem_overwrite_prop_value value`

VM で使用する CPU 共有の数を定義します。

`-datacenter_name datacentername`

VM が存在するデータ センターを指定します。

`-vc vcenterservername`

VM が存在する vCenter Server を指定します。

`-vm_name vmname`

VM を指定します。

`-value value`

リソース パラメータの値を指定します。

例: 単一 vCenter Server 環境で VM の CPU 共有の数を増やす

以下の例は、データ センター VAS/MyCity 上の仮想マシン MyVM01 の CPU 共有を 1000 増やします。

```
dpmvc-setshares -vm_name MyVM01 -datacenter_name VAS/MyCity -setshare  
cpu_add_prop_value -value 1000
```

dpmvc-snapshot コマンド -- スナップショットの管理(Funclet)

dpmvc-snapshot コマンドでは、VM のスナップショットを管理できます。このコマンドを使用して、スナップショットの作成、1 つまたはすべてのスナップショットの削除、スナップショットへの復元を実行できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc-snapshot  
-create  
{-datacenter_name datacentername | -vc vccenterservername}  
-vm_name vmname  
-vm_snapshot_name vmsnapshotname  
[-vm_snapshot_desc description]  
[-withmemory {true|false}]  
[-poweron]
```

```
dpmvc-snapshot  
-remove  
{-datacenter_name datacentername | -vc vccenterservername}  
-vm_name vmname  
[-vm_snapshot_name vmsnapshotname [-withchildren]]
```

```
dpmvc-snapshot  
-revert  
{-datacenter_name datacentername | -vc vccenterservername}  
-vm_name vmname  
-vm_snapshot_name vmsnapshotname
```

-operation {create|remove|revert}

VM スナップショットを作成、削除、または復元するように指定します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vccenterservername*

VM が存在する vCenter Server を指定します。

-vm_name *vmname*

VM を指定します。

-vm_snapshot_name *vmsnapshotname*

VM スナップショットの名前を定義します。

-vm_snapshot_desc *description*

(オプション) VM スナップショットの説明を定義します。

-withmemory {true | false}

(オプション) スナップショットにメモリが含まれていることを指定します。このパラメータは、**-remove** または **-revert** には無効です。

-withchildren

(オプション) スナップショットのすべての子を削除するように指定します。このパラメータは、**-create** または **-revert** には無効です。

-poweron

(オプション) VM の電源をオンにするように指定します。このパラメータは、**-remove** または **-revert** には無効です。

例: 単一 vCenter Server 環境で VM スナップショットを作成する

以下の例は、VM システム MyVM01 で Hello world という名前のスナップショットを、「First snapshot」という説明で作成します。

```
dpmvc-snapshot -create -vm_snapshot_name "Hello world" -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity
```

例: 単一 vCenter Server 環境で VM スナップショットを復元する

以下の例は、MyVM01 を Hello world という名前のスナップショットに復元します。

```
dpmvc-snapshot -revert -vm_snapshot_name "Hello world" -vm_name MyVM01
-datacenter_name VAS/MyCity
```

例: 単一 vCenter Server 環境で VM スナップショットを削除する

以下の例は、VM システム MyVM01 から Hello world という名前のスナップショットを削除します。

```
dpmvc-snapshot -remove -vm_snapshot_name "Hello world" -vm_name MyVM01
-datacenter_name VAS/MyCity
```

dpmvc-templatetovm コマンド -- テンプレートを VM に変換する (Funclet)

dpmvc-templatetovm コマンドは、仮想マシンテンプレートを VM に変換します。この機能を使用すると、たとえば、変換済みの VM にパッチやソフトウェア更新プログラムを適用してから、テンプレートに変換できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc-templatetovm
-template_name templatename
{-datacenter_name datacentername | -vc vccenterservername}
-compute_resource_name hostname
-resource_pool_name resourcepool
-esx_host_name hostname
```

-template_name *templatename*

テンプレートの名前を定義します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vccenterservername*

VM が存在する vCenter Server を指定します。

-compute_resource_name *name*

VM を作成するクラスタまたは VMware ESX ホストの名前を指定します。

-resource_pool_name *resourcepool*

クローンを作成する VM の選択元のリソース プールの名前を指定します。

-esx_host_name *esxhostname*

VM が存在する VMware ESX サーバを指定します。

例: 単一 vCenter Server 環境でテンプレートを VM に変換する

以下の例は、convertTest という名前のテンプレートを VM に変換します。

```
dpmvc-templatetovm -datacenter_name VAS/MyCity -template_name convertTest
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com
-resource_pool_name Resources/QA
```

dpmvc vapp コマンド -- vApp の管理

dpmvc vapp コマンドは、vApps 上の以下の操作をサポートします。

- vApp の新規作成
- クローン
- 電源オン
- 電源オフ
- 中断
- VMware vCenter からの vApp の削除
- VMware vCenter からの登録解除
- vApp への VM の追加
- vApp へのリソースプールの追加
- vApp への vApp の追加
- vApp 設定の更新

このコマンドの形式は、以下のとおりです。

```
dpmvc-vapp -create
```

```
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-cpu_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]  
[-mem_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
```

```
dpmvc-vapp -clone
```

```
-vc_server vcservername  
-vapp_path vapppath  
[-target_vapp_path targetvapppath]  
[-vapp_name vappname]  
[-target_vapp_host targetvapphost]  
[-target_vapp_datastore targetvappdatastore]
```

```
dpmvc-vapp [-power_on | -power_off | -suspend | -delete | -unregister ]
```

```
-vc_server vcservername  
-vapp_path vapppath
```

```
dpmvc-vapp -add_vms_to_vapp
```

```
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-vms vms]
```

```
dpmvc-vapp -add_rps_to_vapp
```

```
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-rps resourcepool]
```

```
dpmvc-vapp -add_vapps_to_vapp
```

```
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-vapps vapps]  
[-locale iso639value]
```

```
dpmvc-vapp -update_vapp_config
```

```
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-config_settings  
vappName,objecType,startAction,startDelay,startOrder,stopAction,stopDelay,waitFor  
Guest]  
[-cpu_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]  
[-mem_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]  
[-locale iso639value]
```

```
-vc_server vcservername
```

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

```
-vapp_path vapppath
```

vApp パスを指定します。

```
-vapp_name vappname
```

(オプション) vApp 名を指定します。

-cpu_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

(オプション) vApp 設定を更新する際の CPU 割り当て設定を指定します。

isExpandableReservation

割り当てが拡張可能かどうか指定します。可能な値: **true** または **false**

limit

割り当ての制限を指定します。

reservation

予約の割り当てを指定します。

sharesLevel

共有レベルを指定します。0 は低、1 は通常、2 は高、3 はカスタムです。例: **true,-1,1000,1,4000**

shares

割り当ての共有を指定します。

-mem_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

vApp 設定を更新する際のメモリ割り当て設定を指定します。

isExpandableReservation

割り当てが拡張可能かどうか指定します。可能な値: **true** または **false**

limit

割り当ての制限を指定します。

reservation

予約の割り当てを指定します。

sharesLevel

共有レベルを指定します。0 は低、1 は通常、2 は高、3 はカスタムです。例: **true,-1,1000,1,4000**

shares

割り当ての共有を指定します。

-vms vms

(オプション) vApp に追加する VM を指定します。

-rps resourcepool

(オプション) vApp に追加するリソース プールの名前を指定します。

-vapps vapps

(オプション) vApp に追加する vApps を指定します。

-config_settings

vappName,objecType,startAction,startDelay,startOrder,stopAction,stopDelay,waitForGuest

(オプション) vApp の構成設定を指定します。適切なフォーマットは以下のとおりです。

startAction none/powerOn

開始のアクションを指定します。利用可能なオプションは none または powerOn です。

stopAction none/powerOff/guestShutdown/suspend

開始のアクションを指定します。利用可能なオプションは none、powerOff、guestShutdown、または guestShutdown です。

-target_vapp_path targetvapppath

作成する vApp のターゲット vApp パスを指定します。

-target_vapp_host targetvapphost

(オプション) 新しい vApp が存在すべきホストを指定します。

-target_vapp_datastore targetvappdatastore

(オプション) vApp が存在すべきデータ ストアの名前を指定します。

dpmvc-virtualswitch コマンド -- 仮想スイッチの管理(Funclet)

dpmvc-virtualswitch コマンドでは、仮想スイッチを管理できます。

- 仮想スイッチを作成する
- 仮想スイッチのプロパティを更新する
- 仮想スイッチを削除する
- 仮想スイッチにポート グループを作成する
- 仮想スイッチのポート グループ プロパティを更新する
- 仮想スイッチからポート グループを削除する
- 仮想スイッチのポート グループの名前を変更する

このコマンドの形式は、以下のとおりです。

```
dpmvc-virtualswitch {-vs_add | -vs_update}  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-nic_names nicname1 [,nicname2, ...]
```

```
dpmvc-virtualswitch -vs_remove  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname
```

```
dpmvc-virtualswitch -add_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-portgroup_name portgroupname  
[-vlan vlanID]
```

```
dpmvc-virtualswitch -update_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-portgroup_name portgroupname  
[-portgroup_newname portgroupnewname]  
[-vlan vlanID]
```

```
dpmvc-virtualswitch -remove_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-portgroup_name portgroupname  
[-vlan vlanID]
```

```
dpmvc-virtualswitch -rename_portgroup  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-portgroup_name portgroupname  
-portgroup_newname portgroupnewname
```

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-switch_name *switchname*

操作を実行するスイッチ名を指定します。

-nic_names *nicname1* [*nicname2*, ...]

物理 NIC をカンマ区切りのリストで指定します。複数の NIC を指定する場合、"sc1,nfs1,mnic1,mnic2" のように二重引用符を使用して引数をエスケープします。

-portgroup_name *portgroupname*

ポート グループ名を指定します。

-portgroup_newname *portgroupnewname*

新しいポート グループ名を指定します。

-vlan *vlanID*

(オプション) 仮想ポート グループの操作に使用する整数値 (VLAN ID) を指定します。

例

以下の例は、仮想スイッチを新規作成します。

```
dpmvc-virtualswitch -vs_add -vc_server vc5master -esx_host_name esx5  
-switch_name switch1 -nic_names "sc1,nfs1,mnic1,mnic2"
```

dpmvc-vmtotemplate コマンド -- VM をテンプレートに変換する(Funclet)

dpmvc-vmtotemplate コマンドは、仮想マシンをテンプレートに変換します。この機能により、データセンターの複数のサーバ間の整合性を強化できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc-vmtotemplate
-vm_name vmname
{-datacenter_name datacentername | -vc vccenterservername}
-vm_name vmname
```

VM を指定します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc *vccenterservername*

VM が存在する vCenter Server を指定します。

例: 単一 vCenter Server 環境で VM をテンプレートに変換する

以下の例は、VM を convertTest という名前のテンプレートに変換します。

```
dpmvc-vmtotemplate -datacenter_name VAS/MyCity -vm_name convertTest
```

dpmvc-wait コマンド -- ジョブの待機時間(Funclet)

dpmvc-wait コマンドは、ジョブの実行完了を待機する時間を指定します。このコマンドは、ジョブが完了するかタイムアウトが発生するまで、ステータスを返しません。

このコマンドの形式は、以下のとおりです。

```
dpmvc-wait
-job <ジョブ ID>
[-timeout timeout]
```

-job *jobid*

ジョブ ID を指定します。

-timeout *timeout*

ジョブ ステータスを指定するタイムアウト値を指定します。

例: ジョブのステータス

以下の例は、ジョブ 125635 の実行完了を 20 秒間待機するように指定します。

```
dpmvc-wait -job 125635 -timeout 20
```

CA Microsoft Hyper-V AutoShell コマンド

スクリプトで AutoShell を使用して CA Hyper-V AutoShell コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドは CLI 内でも利用できます。

注: Hyper-V Server 環境では *vmname* オプションは一意ではありません。便宜上、*vmname* によって VM を指定できます。ただし、*vmname* が一意でない場合、このコマンドは失敗します。その場合は、代わりに *vmid* オプションを使用する必要があります。AutoShell コマンドでは、*vmname* と *vmid*、および *template* と *templateid* のパラメータのペアを使用できません。たとえば、*vmname* を使用する場合は *vmid* を使用できません。

dpmhv-addVMNic (Cmdlet) コマンド -- VM にネットワーク インターフェース コントローラを追加する

dpmhv-addVMNic コマンドを使用すると VM にネットワーク アダプタを追加することができ、そのアダプタを仮想スイッチに接続するオプションが提供されます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-addVMNic
-host hostname
-vm vmname|-vmid vmguid
[-mac_address macaddress]
[-switch switchname]
[-legacy]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-mac_address *macaddress*

(オプション) 新しいネットワーク アダプタの MAC アドレスを定義します。有効なエントリ: 次の形式の 12 桁の 16 進数値 (例: A ~ F、0 ~ 9、a ~ f) : 0013724C2140 または 00:13:72:4C:21:40。

注: *macaddress* を指定しない場合、Hyper-V は MAC アドレスを自動的に生成します。

-switch *switchname*

VM のネットワーク アダプタを接続する仮想ネットワーク スイッチの名前を指定します。このオプションを指定しない場合、VM のネットワーク アダプタは仮想ネットワークに接続されません。

デフォルト: なし

-legacy

(オプション) Hyper-V 合成アダプタの代わりにレガシー ネットワーク アダプタを作成します。

注: Hyper-V 統合サービスがないオペレーティング システムをインストールする場合にこのオプションを使用します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: VM にネットワーク インターフェース コントローラを追加する

この例では、VM 「TestVM」に NIC を追加します。

```
dpmhv-addVMNic
-host hvserver
-vm TestVM
-switch "Local Area Connection
- Virtual Network"
```

dpmhv-addVMSCSIController (Cmdlet) コマンド -- VM に SCSI コントローラを追加する

dpmhv-addVMSCSIController コマンドを使用すると、VM に最大 4 つの SCSI コントローラを追加できます。SCSI コントローラには最大 64 個のディスクを追加できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-addVMSCSIController  
-host hostname  
-vm vmname|-vmid vmguid  
[-retval]  
[-silent]  
  
-host hostname
```

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: VM に SCSI コントローラを追加する

この例では、VM 「TestVM」に SCSI コントローラを追加します。

```
dpmhv-addVMSCSIController -host hvserver -vm TestVM
```

dpmhv-changeVMState (Cmdlet) コマンド -- VM の状態を変更する

dpmhv-changeVMState コマンドは、VM の状態を変更します。start、suspend、stop、pause、shutdown、およびrebootなどのさまざまな状態を使用できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-changeVMState  
-host hostname  
-vm vmname|-vmid vmguid  
-state {start,suspend,stop,pause,shutdown,reboot}  
[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

`-state {start,suspend,stop,pause,shutdown,reboot}`

VM の状態を指定します。状態のオプションを以下に示します。

`start`

VM を起動または再開します。

`suspend`

VM を一時的に中断します。

`stop`

VM を停止します。

`pauses`

VM を一時的に停止します。

シャットダウン

VM をシャットダウンします。

`reboot`

VM を再起動します。

注:

無効な状態変更を指定すると、システムは例外をレポートします。

`-retval`

後で処理できるように値を返します。

`-silent`

画面に出力しないように指定します。

例: VM の状態を変更する

この例では、VM 「TestVM」を起動します。

```
dpmhv-changeVMState -host hvserver -vm TestVM -state start
```

dpmhv-createTemplateFromVM (Funclet) コマンド -- VM からテンプレートを作成する

dpmhv-createTemplateFromVM コマンドは、既存の VM を使用してテンプレートを作成します。テンプレートを使用して複数の VM を作成できます。

テンプレートによって、VM の仕様および VM に接続された仮想ハードディスクがコピーされます。スナップショットを持つ VM を使用してテンプレートを作成する場合、スナップショットなしで現在の状態のみがエクスポートされます。このコマンドは、各 Hyper-V Server のテンプレートカタログを作成します。テンプレートカタログは以下の場所に保持され、以下の場所から照会できます

す： %ALLUSERSPROFILE%\ca\%vpm%\Hyper-V_Templates\%TemplateCatalog.xml

このコマンドの形式は、以下のとおりです。

```
dpmhv-createTemplateFromVM
-host hostname
-vm vmname|-vmid vmguid
-template templatename
[-async]
[-path pathname]
[-description description]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-template *templatename*

(オプション) テンプレート カタログで作成された汎用テンプレートの名前を指定します。

注: テンプレート名はローカルテンプレート カタログ内で一意である必要があります。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-dest *pathname*

(オプション) 作成する VM のパスを指定します (テンプレートが格納されます)。このオプションを指定しない場合、VM は Hyper-V Server のデフォルトの場所に作成されます。作成されたテンプレートの名前は次の場所に保存されます:

す: %ALLUSERSPROFILE%\ca¥vpm¥Hyper-V_Templates。

注: Hyper-V のデフォルトの場所に VM を作成しないことをお勧めします。テンプレートを使用して VM を作成すると、仮想ディスクイメージファイル名が競合します。

-description *description*

(オプション) テンプレートの説明を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

dpmhv-createTemplateFromVM コマンドが同期で実行される場合、このコマンドは空の文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が *dpmhv-getJobErrorInfo*、*dpmhv-getJobInfo*、または *dpmhv-getJobStatus* に渡されます。実行中にエラーが発生した場合、コマンドは例外をレポートします。

例: VM からテンプレートを作成する

この例では、VM 「TestVM」からテンプレート「MyTemplate」を作成します。

```
dpmhv-createTemplateFromVM -vm TestVM -template MyTemplate -host hvserver
```

dpmhv-createVirtDisk(Funclet)コマンド -- 仮想ディスクを作成する

dpmhv-createVirtDisk コマンドを使用すると、仮想ディスクを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-createVirtDisk
-host hostname
-path pathname
-type{fixed,dynamic}
-sizeGB size
[-async]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-path *pathname*

作成する仮想ディスクのパスを指定します。

-type {fixed, dynamic}

作成する仮想ディスクのタイプを指定します。オプションは、以下のとおりです。

fixed

仮想ディスク タイプを固定として指定します。

dynamic

仮想ディスク タイプを動的として指定します。

-sizeGB *size*

仮想ディスクのサイズをギガバイトで指定します。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: 仮想ディスクを作成する

この例では、サイズが 1 GB の仮想ディスクを作成します。

```
dpmhv-createVirtDisk -host hvserver -path C:¥data¥Disks¥01.vhd -type fixed -sizeGB 1
```

dpmhv-createVirtFloppy (Cmdlet) コマンド -- 仮想フロッピーを作成する

dpmhv-createVirtFloppy コマンドを使用すると、仮想フロッピー ディスク イメージファイルを作成できます。 **dpmhv-setVMVirtFloppy** コマンドを使用すると、仮想フロッピーを VM に接続できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-createVirtFloppy  
-host hostname  
-path pathname[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-path *pathname*

作成する仮想フロッピー ディスク イメージファイルのパスを指定します。指定しない場合、イメージファイルは Hyper-V 仮想ハードディスクのデフォルト ディレクトリに作成されます。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: 仮想フロッピーを作成する

この例では、仮想フロッピー ディスク イメージファイル「fdd.vfd」を作成します。

```
dpmhv-createVirtFloppy -host hvserver -path C:¥VMDisks¥fdd.vfd
```

dpmhv-createVM(Funclet)コマンド -- VM を作成する

dpmhv-createVM コマンドを使用すると、VM を作成できます。このコマンドは、ネットワーク アダプタまたはハードディスクなどのシステム コンポーネントを持たない VM を作成します。VM を作成する際にそれぞれのコマンドを使用して、システム コンポーネントを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-createVM
-host hostname
-name name
[-path pathname]
[-template templatename]
[-templateid vmguid]
[-mem mem]
[-cpus cpus]
[-cpuidlimit limit]
[-cpureserve reserve]
[-cpulimit limit]
[-cpuweight weight]
[-startAction {none|auto|always}]
[-startDelay delay]
[-stopAction {save,off,shutdown}]
[-recoveryAction {none,restart,revert}]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-name *name*

(オプション) VM の名前を指定します。

注: VM の名前は一意である必要があります。

-dest *pathname*

(オプション) 作成する VM のパスを指定します (テンプレートが格納されます)。このオプションを指定しない場合、VM は Hyper-V Server のデフォルトの場所に作成されます。作成されたテンプレートの名前は次の場所に保存されます:

す: %ALLUSERSPROFILE%\ca\vm\Hyper-V_Templates。

注: Hyper-V のデフォルトの場所に VM を作成しないことをお勧めします。テンプレートを使用して VM を作成すると、仮想ディスクイメージファイル名が競合します。

-template *templatename*

(オプション) 既存の VM の名前を指定します。このオプションは、新しい VM を作成する場合に既存の VM の設定をコピーするために使用されます。既存の VM から VM を作成しないことをお勧めします。2 つの VM で仮想ディスクなどのリソースが共有され、同時に稼働できなくなる可能性があります。

注: ローカルのテンプレート カタログ内ではテンプレート名が一意である必要があります。

-templateid *vmguid*

(オプション) 既存 VM の一意の ID (GUID) を指定します。このオプションは、既存 VM の設定をコピーして新しい VM を作成するために使用します。既存の VM から VM を作成しないことをお勧めします。2 つの VM で仮想ディスクなどのリソースが共有され、同時に稼働できなくなる可能性があります。

デフォルト: -1

注: このオプションは VM カタログのテンプレート ID ではなく、Hyper-V 環境の既存 VM を表します。

-mem *mem*

(オプション) 作成する VM の RAM メモリをメガバイト (MB) で指定します。

デフォルト: -1

-cpus *cpus*

(オプション) VM に割り当てる CPU コアの数指定します。

-cpuidlimit *limit*

(オプション) VM の CPU ID 機能の制限を指定します。このオプションを指定すると、Windows NT のようなレガシーオペレーティングシステムとの互換性が向上します。

デフォルト: -1

-cpureserve *reserve*

(オプション) VM 用に予約する CPU の割合を指定します。このオプションを指定しない場合、Hyper-V Server は全体的なシステムの使用状況に基づいて CPU サイクルを割り当てます。

デフォルト: -1

-cpulimit *limit*

(オプション) VM で使用する CPU リソース数の制限を指定します。このオプションを指定すると、CPU 能力が異なる物理ホスト間で VM を移行できるだけでなく、互換性が向上します。

デフォルト : -1

-cpuweight *weight*

(オプション) 仮想マシンの相対的な加重を指定します。複数の VM が稼働している場合、このオプションはリソースの割り当てを制御します。有効なエントリ : 整数、1 ~ 10000。

デフォルト : -1

-startAction {none,auto,always}

(オプション) Hyper-V を起動する際に VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

何も実行しません。

auto

VM を自動的に起動します。

注: Hyper-V ホストのシャットダウン前に VM が稼働中だった場合は、このオプションを使用します。

always

VM を常に起動します。

-startDelay *delay*

(オプション) Hyper-V ホストが起動し、稼働してから VM を起動するまでの遅延時間を秒数で指定します。

デフォルト : -1

-stopAction {save,off,shutdown}

(オプション) 物理 Hyper-V をシャットダウンする際に VM で実行するアクションを指定します。オプションは、以下のとおりです。

save

VM を中断します。

off

VM の電源をオフにします。

shutdown

システムをシャットダウンします。

-recoveryAction {none,restart,revert}

(オプション) 予期しないシャットダウンから Hyper-V ホストを再起動した後、VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

何も実行しません。

restart

VM を再起動します。

revert

VM の最後のスナップショットを復元します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

このコマンドは、新しく作成された VM の一意の識別子 (GUID) を持つ文字列を返します。

例: デフォルト設定を使用して VM を作成する

この例では、デフォルト設定を使用して VM 「TestVM」を作成します。

```
dpmhv-createVM -host hvserver -name TestVM
```

例: ユーザ定義のディレクトリにデフォルト設定を使用して VM を作成する

この例では、ユーザ定義のディレクトリにデフォルト設定を使用して VM 「TestVM」を作成します。

```
dpmhv-createVM -host hvserver -name TestVM -path c:¥VMs¥TestVM
```

例: メモリおよび CPU コアのカスタムの値を持つ VM を作成する

この例では、メモリおよび CPU コアのカスタムの値を持つ VM 「TestVM」を作成します。

```
dpmhv-createVM -host hvserver -name TestVM -mem 1024 -cpus 2
```

dpmhv-createVMFromTemplate(Funclet)コマンド -- テンプレートを使用して VM を作成する

dpmhv-createVMFromTemplate コマンドは、以前に作成した Microsoft System Center Virtual Machine Manager (SCVMM) テンプレートから VM を作成します。また、オプションで Windows イメージの設定をカスタマイズします。

1 つのテンプレートを使用して任意の数の仮想マシンを作成できるように、テンプレートは変更せずに保持されます。このコマンドは SCVMM がホストするテンプレートのみをサポートします。SCVMM を使用しないローカルテンプレートのサポートについては、dpmhv-createVMFromTemplateEx を参照してください。

このコマンドの形式は、以下のとおりです。

```
dpmhv-createVMFromTemplate
-host hostname
-vm vmname
-template templatename
-dest pathname
-scvmmHost scvmmHostname
-adminPass adminPW
[-computerName computerName]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-productKey key]
[-userName UserName]
[-organization orgname]
[-domain domainname]
[-domainAdmin domad]
[-domainAdminPass dompw]
[-workgroup wgname]
[-adminUser adminUser]
[-adminUserPass adminUserPW]
[-mem mem]
[-cpus cpus]
[-hardwareProf hwprof]
[-guestOSProf osprof]
[-startVM]

-host hostname
```

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-template *templatename*

VM の作成に使用するソース テンプレートの名前を指定します。

注: ローカルのテンプレート カatalog内ではテンプレート名が一意である必要があります。

-dest *pathname*

(オプション) 作成する VM のパスを指定します (テンプレートが格納されます)。このオプションを指定しない場合、VM は Hyper-V Server のデフォルトの場所に作成されます。作成されたテンプレートの名前は次の場所に保存されます:

す: %ALLUSERSPROFILE%\ca¥vpm¥Hyper-V_Templates。

注: Hyper-V のデフォルトの場所に VM を作成しないことをお勧めします。テンプレートを使用して VM を作成すると、仮想ディスクイメージファイル名が競合します。

-scvmmHost *scvmmHostname*

(オプション) Microsoft System Center Virtual Machine Manager (SCVMM) ライブラリ サーバのホスト名を指定します。VM のプロビジョニングで SCVMM の統合を使用する場合に、このパラメータは有効です。

-adminPass *adminPW*

(オプション) このオプションは、VM のデフォルト管理者アカウントのパスワードを設定するために使用されます。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。非同期実行の場合、このパラメータは無視されます。

注: このオプションを正常に設定するには、Sysprep ツールを使用して、Windows サーバ管理者のパスワードを空に設定します。

-computerName *computerName*

(オプション) VM のコンピュータ名を指定します。このパラメータをサポートするには、Sysprep ツールを使用するイメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-ip4addr *ip4addr*

(オプション) VM のインターフェースに割り当てる静的 IPv4 アドレスを指定します。特定のインターフェースの IP アドレスを設定するには、IP アドレスの前に、既知のインターフェース名と区切り記号として「#」が付けられます。たとえば、

-ip4addr "Local Area Connection#192.168.1.200" などです。テンプレートイメージに複数のネットワーク アダプタがある場合、IP アドレスは最初のインターフェースに割り当てられます。このオプションは、コマンドを非同期で実行する場合は無効です。

-ip4dhcp *ip4dhcpInt*

(オプション) VM の特定のインターフェースの DHCP をオンにするオプションを指定します。このオプションにはインターフェース名も指定できます。たとえば、

-ip4dhcp "Local Area Connection" などです。テンプレートイメージに複数のネットワーク アダプタがある場合、最初のインターフェースの DHCP がオンになります。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : local

-ip4mask *ip4mask*

(オプション) VM に割り当てるサブネットマスクを指定します。このオプションは、

-ip4addr オプションと共に使用されます。インターフェース名を **-ip4addr** オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-ip4gw *ip4gw*

(オプション) VM のゲートウェイを設定するオプションを指定します。このオプションは、**-ip4addr** オプションと共に使用します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

注: インターフェース名を

-ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。

-ip4metric *ip4metric*

(オプション) VM に設定するインターフェース メトリックを指定します。このオプションは、**-ip4addr** オプションと共に使用します。インターフェース名を

-ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : 1

-ip4dns *ip4dns*

(オプション) VM に設定する **DNS** サーバを指定します。このオプションは、**-ip4addr** オプションと共に使用します。インターフェース名を **-ip4addr** オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-adminPass *adminPW*

(オプション) このオプションは、VM のデフォルト管理者アカウントのパスワードを設定するために使用されます。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。非同期実行の場合、このパラメータは無視されます。

注: このオプションを正常に設定するには、**Sysprep** ツールを使用して、**Windows Server 2003** の管理者パスワードを空に設定します。

-productKey *key*

(オプション) VM の **Windows** 製品アクティベーション キーを指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-userName *UserName*

(オプション) 新しい VM にインストールされる、(**sysprep** ツールを使用して作成された) **Windows** イメージのユーザ名を指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-organization *orgname*

(オプション) 新しい VM にインストールされる、(sysprep ツールを使用して作成された) Windows イメージの組織名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-domain *domainname*

(オプション) VM のドメイン名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。ドメインとワークグループの仕様を同時に指定することはできません。

-domainAdmin *domad*

(オプション) ドメインに作成するドメイン管理者のユーザアカウントを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-domainAdminPass *dompw*

(オプション) ドメインに作成するドメイン管理者ユーザアカウントのパスワードを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-workgroup *wgname*

(オプション) VM 用に作成するワークグループを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。ドメインとワークグループの仕様を同時に指定することはできません。このオプションは、コマンドを非同期で実行する場合は無効です。

-adminUser *adminUser*

(オプション) デフォルトの Administrators グループの一部として作成する管理者ユーザ名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-adminUserPass *adminUserPW*

(オプション) デフォルトの **Administrators** グループの一部として作成する管理者ユーザ アカウントのパスワードを指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-mem *mem*

(オプション) 作成する **VM** の **RAM** メモリをメガバイト (**MB**) で指定します。

デフォルト : -1

-cpus *cpus*

(オプション) **VM** に割り当てる **CPU** コアの数指定します。

デフォルト : -1

-scvmmHost *scvmmHostname*

(オプション) **Microsoft System Center Virtual Machine Manager (SCVMM)** ライブラリ サーバのホスト名を指定します。**VM** のプロビジョニングで **SCVMM** の統合を使用する場合に、このパラメータは有効です。

-hardwareProf *hwprof*

(オプション) **Microsoft System Center Virtual Machine Manager (SCVMM)** ライブラリ サーバで定義されているハードウェア プロファイルの名前を指定します。このパラメータは、**SCVMM** テンプレートに保存されているハードウェア設定を上書きします。**VM** のプロビジョニングで **SCVMM** の統合を使用する場合に、このパラメータは有効です。

-guestOSProf *osprof*

(オプション) **Microsoft System Center Virtual Machine Manager (SCVMM)** ライブラリ サーバで定義されているゲストオペレーティング システム プロファイルの名前を指定します。このパラメータは、**SCVMM** テンプレートに保存されているオペレーティング システム設定を上書きします。**VM** のプロビジョニングで **SCVMM** の統合を使用する場合に、このパラメータは有効です。

-startVM

(オプション)VM の作成後に VM を自動的に起動するためのオプションを指定します。デフォルトでは、新しい VM は電源がオフの状態です。VM を起動するのに必要な IP 設定オプションをカスタマイズする必要があります。

戻り値

dpmhv-createVMFromTemplate が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が *dpmhv-getJobErrorInfo*、*dpmhv-getJobInfo*、または *dpmhv-getJobStatus* に渡されます。実行中にエラーが発生した場合、コマンドは例外をレポートします。

例: SCVMM テンプレートから VM を作成する

この例では、SCVMM テンプレートから VM を作成し、ハードウェアおよびゲスト OS プロファイルを指定するオプションを使用します。

```
dpmhv-createVMFromTemplate -host hvserver -vm NewVM -dest c:¥VMs¥NewBox -template Win2k3Clean  
-scvmmHost SCVMMLibSrv -hardwareProf hw_def -guestOSProf os_def -async -adminPass #admin#
```

関連項目

[dpmhv-createVMFromTemplateEx \(Funclet\) コマンド -- テンプレートを使用して VM を作成する \(P. 258\)](#)

dpmhv-createVMFromTemplateEx(Funclet) コマンド -- テンプレートを使用して VM を作成する

dpmhv-createVMFromTemplate コマンドを使用すると、既存のテンプレートを使用して VM を作成できます。元のテンプレートが **Sysprep** 状態で作成された場合は、このコマンドパラメータを使用して、イメージを作成した VM の **Windows** 設定をカスタマイズできます。VM を作成する際に、それぞれのコマンドを使用して、システム コンポーネントを作成できます。1 つのテンプレートを使用して複数の仮想マシンを作成できます。

注: コマンドを同期で実行する場合は、**Windows** 設定イメージをカスタマイズできます。非同期実行の場合は、実行が完了した後に **dpmhv-setSysprepProperties** コマンドを使用してイメージをカスタマイズできます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-createVMFromTemplateEx
-host hostname
-vm vmname
-template templatename
-dest pathname
[-computerName computerName]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-disableAdmin]
[-adminPass adminPW]
[-autoLogon alcnt]
[-duplicatorString dupstr]
[-timeZone timezone]
[-productKey key]
[-userName UserName]
[-organization orgname]
[-domain domainname]
[-domainAdmin domad]
[-domainAdminPass dompw]
[-workgroup wgname]
[-adminUser adminUser]
[-adminUserPass adminUserPW]
[-custom custom]
[-async]
[-mem mem]
[-cpus cpus]
[-cpuidlimit limit]
[-cpufeatlimit limit]
[-cpureserve reserve]
[-cpulimit limit]
[-cpuweight weight]
[-startAction {none,auto,always}]
[-startDelay delay]
[-stopAction {save,off,shutdown}]
[-recoveryAction {none,restart,revert}]
[-scvmmHost scvmmHostname]
[-hardwareProf hwprof]
[-guestOSProf osprof]
[-startVM]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-template *templatename*

VM の作成に使用するソース テンプレートの名前を指定します。

注: ローカルのテンプレート カタログ内ではテンプレート名が一意である必要があります。

-dest *pathname*

(オプション) 作成する VM のパスを指定します (テンプレートが格納されます)。このオプションを指定しない場合、VM は Hyper-V Server のデフォルトの場所に作成されます。作成されたテンプレートの名前は次の場所に保存されます:

す: %ALLUSERSPROFILE%\ca\vm\Hyper-V_Templates。

注: Hyper-V のデフォルトの場所に VM を作成しないことをお勧めします。テンプレートを使用して VM を作成すると、仮想ディスクイメージファイル名が競合します。

-computerName *computerName*

(オプション) VM のコンピュータ名を指定します。このパラメータをサポートするには、Sysprep ツールを使用するイメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-ip4addr *ip4addr*

(オプション) VM のインターフェースに割り当てる静的 IPv4 アドレスを指定します。特定のインターフェースの IP アドレスを設定するには、IP アドレスの前に、既知のインターフェース名と区切り記号として「#」が付けられます。たとえば、

-ip4addr "Local Area Connection#192.168.1.200" などです。テンプレートイメージに複数のネットワーク アダプタがある場合、IP アドレスは最初のインターフェースに割り当てられます。このオプションは、コマンドを非同期で実行する場合は無効です。

-ip4dhcp *ip4dhcplnt*

(オプション) VM の特定のインターフェースの DHCP をオンにするオプションを指定します。このオプションにはインターフェース名も指定できます。たとえば、

-ip4dhcp "Local Area Connection" などです。テンプレートイメージに複数のネットワーク アダプタがある場合、最初のインターフェースの DHCP がオンになります。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト: local

-ip4mask *ip4mask*

(オプション) VM に割り当てるサブネットマスクを指定します。このオプションは、

-ip4addr オプションと共に使用されます。インターフェース名を -ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-ip4gw *ip4gw*

(オプション) VM のゲートウェイを設定するオプションを指定します。このオプションは、-ip4addr オプションと共に使用します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

注: インターフェース名を

-ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。

-ip4metric *ip4metric*

(オプション) VM に設定するインターフェースメトリックを指定します。このオプションは、-ip4addr オプションと共に使用します。インターフェース名を

-ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト: 1

-ip4dns *ip4dns*

(オプション) VM に設定する DNS サーバを指定します。このオプションは、-ip4addr オプションと共に使用します。インターフェース名を -ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-disableAdmin

(オプション) VM のデフォルトの管理者アカウントを無効にするオプションを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-adminPass *adminPW*

(オプション) このオプションは、VM のデフォルト管理者アカウントのパスワードを設定するために使用されます。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。非同期実行の場合、このパラメータは無視されます。

注: このオプションを正常に設定するには、Sysprep ツールを使用して、Windows サーバ管理者のパスワードを空に設定します。

-autoLogon *alcnt*

(オプション) デフォルト管理者アカウントを使用して自動的にログインするために、作成するアカウント数を設定するオプションを指定します。このアカウントは Sysprep プロセスの完了後に作成されます。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト :

-1

-duplicatorString *dupstr*

(オプション) VM のレジストリに設定するシステム デュプリケータの名前を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-timeZone *timezone*

(オプション) テンプレートを使用して作成された VM が使用するタイムゾーンを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト :

-1

-productKey *key*

(オプション) VM の Windows 製品アクティベーション キーを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-userName *UserName*

(オプション) 新しい VM にインストールされる、(sysprep ツールを使用して作成された) Windows イメージのユーザ名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-organization *orgname*

(オプション) 新しい VM にインストールされる、(sysprep ツールを使用して作成された) Windows イメージの組織名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-domain *domainname*

(オプション) VM のドメイン名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。ドメインとワークグループの仕様を同時に指定することはできません。

-domainAdmin *domad*

(オプション) ドメインに作成するドメイン管理者のユーザ アカウントを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-domainAdminPass *dompw*

(オプション) ドメインに作成するドメイン管理者ユーザ アカウントのパスワードを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-workgroup *wgname*

(オプション) VM 用に作成するワークグループを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。ドメインとワークグループの仕様を同時に指定することはできません。このオプションは、コマンドを非同期で実行する場合は無効です。

-adminUser *adminUser*

(オプション) デフォルトの Administrators グループの一部として作成する管理者ユーザ名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-adminUserPass *adminUserPW*

(オプション) デフォルトの Administrators グループの一部として作成する管理者ユーザ アカウントのパスワードを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

-custom *custom*

(オプション) Sysprep プロセスの最後に実行する、カンマ区切りのカスタム コマンドのリストを指定します。このパラメータには、Sysprep の Windows イメージが必要です。非同期実行の場合、このパラメータは無視されます。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-mem *mem*

(オプション) 作成する VM の RAM メモリをメガバイト (MB) で指定します。

デフォルト: -1

-cpus *cpus*

(オプション) VM に割り当てる CPU コアの数指定します。

デフォルト: -1

-cpuidlimit *limit*

(オプション) VM の CPU ID 機能の制限を指定します。このオプションを指定すると、Windows NT のようなレガシーオペレーティングシステムとの互換性が向上します。

デフォルト : -1

-cpufeatlimit *limit*

(オプション) VM で使用する CPU リソースの機能の制限を指定します。このオプションを指定すると、CPU 能力が異なる物理ホスト間で VM を移行できるだけでなく、互換性が向上します。

デフォルト : -1

-cpureserve *reserve*

(オプション) VM 用に予約する CPU の割合を指定します。このオプションを指定しない場合、Hyper-V Server は全体的なシステムの使用状況に基づいて CPU サイクルを割り当てます。

デフォルト : -1

-cpulimit *limit*

(オプション) VM で使用する CPU リソース数の制限を指定します。このオプションを指定すると、CPU 能力が異なる物理ホスト間で VM を移行できるだけでなく、互換性が向上します。

デフォルト : -1

-cpuweight *weight*

(オプション) 仮想マシンの相対的な加重を指定します。複数の VM が稼働している場合、このオプションはリソースの割り当てを制御します。有効なエントリ : 整数、1 ~ 10000。

デフォルト : -1

-startAction {none,auto,always}

(オプション) Hyper-V を起動する際に VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

何も実行しません。

auto

VM を自動的に起動します。

注: Hyper-V ホストのシャットダウン前に VM が稼働中だった場合は、このオプションを使用します。

always

VM を常に起動します。

-startDelay *delay*

(オプション) Hyper-V ホストが起動し、稼働してから VM を起動するまでの遅延時間を秒数で指定します。

デフォルト: -1

-stopAction {save,off,shutdown}

(オプション) 物理 Hyper-V をシャットダウンする際に VM で実行するアクションを指定します。オプションは、以下のとおりです。

save

VM を中断します。

off

VM の電源をオフにします。

shutdown

システムをシャットダウンします。

-recoveryAction {none,restart,revert}

(オプション) 予期しないシャットダウンから Hyper-V ホストを再起動した後、VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

何も実行しません。

restart

VM を再起動します。

revert

VM の最後のスナップショットを復元します。

-scvmmHost *scvmmHostname*

(オプション) Microsoft System Center Virtual Machine Manager (SCVMM) ライブラリ サーバのホスト名を指定します。VM のプロビジョニングで SCVMM の統合を使用する場合に、このパラメータは有効です。

-hardwareProf *hwprof*

(オプション) Microsoft System Center Virtual Machine Manager (SCVMM) ライブラリ サーバで定義されているハードウェア プロファイルの名前を指定します。このパラメータは、SCVMM テンプレートに保存されているハードウェア設定を上書きします。VM のプロビジョニングで SCVMM の統合を使用する場合に、このパラメータは有効です。

-guestOSProf *osprof*

(オプション) Microsoft System Center Virtual Machine Manager (SCVMM) ライブラリ サーバで定義されているゲストオペレーティングシステム プロファイルの名前を指定します。このパラメータは、SCVMM テンプレートに保存されているオペレーティングシステム設定を上書きします。VM のプロビジョニングで SCVMM の統合を使用する場合に、このパラメータは有効です。

-startVM

(オプション) VM の作成後に VM を自動的に起動するためのオプションを指定します。デフォルトでは、新しい VM は電源がオフの状態です。VM を起動するのに必要な IP 設定オプションをカスタマイズする必要があります。

戻り値

`dpmhv-createVMFromTemplateEX` が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が `dpmhv-getJobErrorInfo`、`dpmhv-getJobInfo`、または `dpmhv-getJobStatus` に渡されます。実行中にエラーが発生した場合、コマンドは例外をレポートします。

例: テンプレートから VM を作成する

この例では、完全な静的 IPv4 設定としてテンプレート設定を使用して、VM「TestVM」を作成します。

```
dpmhv-createVMFromTemplateEx -host hvserver -vm NewVM -dest c:\VMs\NewBox -template Win2k3SysPrepped -ip4addr 192.168.1.25 -ip4mask 255.255.255.0 -ip4gw 192.168.1.1 -ip4dns 192.168.1.1 -computerName NewBox
```

関連項目

[dpmhv-createVMFromTemplate \(Funclet\) コマンド -- テンプレートを使用して VM を作成する \(P. 251\)](#)

`dpmhv-destroyTemplate (Cmdlet) コマンド -- テンプレートを削除する`

`dpmhv-destroyTemplate` コマンドを使用すると、テンプレートカタログから既存のテンプレートを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-destroyTemplate  
-template templatename  
-host hostname  
  
-template templatename
```

(オプション) テンプレート カタログで作成された汎用テンプレートの名前を指定します。

注: テンプレート名はローカル テンプレート カタログ内で一意である必要があります。

```
-host hostname
```

Hyper-V Server ホストの名前を指定します。

例: テンプレートを削除する

この例では、テンプレート カタログからテンプレートを削除します。

```
dpmhv-destroyTemplate -host hvserver -template MyTemplate
```

dpmhv-destroyVM(Cmdlet)コマンド -- VM を削除する

dpmhv-destroyVM コマンドを使用すると、既存の VM 仕様、仮想ディスク、フロッピー イメージ、および関連するスナップショットを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-destroyVM  
-host hostname  
-vm vmname|-vmid vmguid  
[-delvhd]  
[-delvfd]  
[-deliso]  
[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-delvhd

(オプション) VM に割り当てられた仮想ハード ディスクを削除します。

-delvfd

(オプション) VM に割り当てられた仮想フロッピー ディスクを削除します。

-deliso

(オプション) VM に割り当てられた ISO イメージを削除します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: VM を削除する

この例では、仮想ハードディスクおよびフロッピー ディスクが接続されている VM を削除します。

```
dpmhv-destroyVM -host hvserver -vm TestVM -delvhd -delvfd
```

dpmhv-expandVirtDisk(Funclet)コマンド -- 仮想ディスクを拡張する

dpmhv-expandVirtDisk コマンドを使用すると、仮想ディスク サイズを増やすことができます。拡張された仮想ディスク領域を使用するには、仮想マシンで実行しているゲスト オペレーティング システムの仮想ディスクをパーティション化する必要があります。

このコマンドの形式は、以下のとおりです。

```
dpmhv-expandVirtDisk  
-host hostname  
-path pathname  
-sizeGB size  
[-async]  
[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-path *pathname*

拡張する仮想ディスクのパス名を指定します。

-sizeGB *size*

仮想ディスクのサイズをギガバイトで指定します。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

`dpmhv-expandVirtDisk` が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が `dpmhv-getJobErrorInfo`、`dpmhv-getJobInfo`、または `dpmhv-getJobStatus` に渡されます。同期モードで実行中にエラーが発生した場合、このコマンドはエラー コードを返してエラーをレポートします。

例: 仮想ディスクを拡張する

この例では、仮想ディスクのサイズを 4 GB に拡張します。

```
dpmhv-expandVirtDisk -host hvserver -path C:\VMDisks\%hdd.vhd -sizeGB 4
```

`dpmhv-exportVM (Funclet) コマンド -- VM をエクスポートする`

`dpmhv-exportVM` コマンドを使用すると、関連する仮想ディスクおよびフロッピー ディスクを含む VM の仕様をエクスポートできます。このコマンドは廃止されるため、代わりに `dpmhv-exportVMEx` を使用してください。

このコマンドの形式は、以下のとおりです。

```
dpmhv-exportVM  
-host hostname  
-vm vmname|-vmid vmguid  
-dest dest  
[-nostate]  
[-async]  
[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-dest *dest*

エクスポートする VM のデスティネーションパスを指定します。

-nostate

(オプション) VM の規格のみをエクスポートすることを指定します。このオプションは、エクスポートされた VM が元の VM ディスクイメージを参照するのを防ぎます。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

`dpmhv-exportVM` が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が `dpmhv-getJobErrorInfo`、`dpmhv-getJobInfo`、または `dpmhv-getJobStatus` に渡されます。同期モードで実行中にエラーが発生した場合、このコマンドはエラーコードを返してエラーをレポートします。

例: VM をエクスポートする

この例では、関連する仮想ディスクおよびフロッピーディスクを含む VM の仕様をエクスポートします。

```
dpmhv-exportVM -host hvserver -vm VM_0000001 -dest c:¥DATA¥Export¥
```


dpmhv-exportVMEx(Funclet)コマンド -- VM をエクスポートする

dpmhv-exportVMEx コマンドを使用すると、関連する仮想ディスク、フロッピー ディスク、およびスナップショットを含む **VM** の仕様をエクスポートできます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-exportVMEx
-host hostname
-vm vmname|-vmid vmguid
-dest dest
[-noruntimeInfo]
[-nosnapshots]
[-nostorage]
[-nosubdir]
[-snapshot ss]
[-snapshotid ssguid]
[-async]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-dest *dest*

エクスポートする VM のデスティネーションパスを指定します。

-noruntimeInfo

(オプション) VM のランタイム情報をエクスポートしないように指定します。

-nosnapshots

(オプション) VM のスナップショットをエクスポートしないように指定します。このオプションは、VM の現在の状態のみをエクスポートします。

-nostorage

(オプション) VM の仮想ディスクをエクスポートしないように指定します。

注: VM のスナップショットをエクスポートする場合はこのオプションを使用しません。

-nosubdir

(オプション) エクスポートパスとして指定されたディレクトリの下に、VM 表示名を持つサブディレクトリを作成しないように指定します。

-snapshot ss

(オプション) VM のスナップショットの表示名を指定します。エクスポートされた VM は、ソース VM と同じ状態 (スナップショット) です。エクスポートされた VM にはスナップショットは含まれません。

注: Hyper-V 環境でスナップショットの表示名が一意でない場合は、**-ssid** オプションを使用します。

-snapshotid ssguid

(オプション) VM のスナップショットの一意的 ID (GUID) を指定します。エクスポートされた VM は、ソース VM と同じ状態 (スナップショット) です。エクスポートされた VM にはスナップショットは含まれません。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

`dpmhv-exportVMEx` が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が `dpmhv-getJobErrorInfo`、`dpmhv-getJobInfo`、または `dpmhv-getJobStatus` に渡されます。同期モードで実行中にエラーが発生した場合、このコマンドはエラーコードを返してエラーをレポートします。

例: スナップショットなしで VM をエクスポートする

この例では、スナップショットなしで VM をエクスポートします。

```
dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:¥Ex¥TestVM1 -nosubdir -nosnapshots
```

`dpmhv-getHostSwitches (Funclet)` コマンド -- Hyper-V ホストのホストスイッチを表示する

`dpmhv-getHostSwitches` コマンドは、Hyper-V ホストに定義されたホストスイッチ（仮想ネットワーク）を表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getHostSwitches
-host hostname
[-retval]
[-silent]
[-detail]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

（オプション）仮想ネットワーク（ホスト ネットワーク）の詳細を表示します。

戻り値

dpmhv-getHostSwitches コマンドは、指定されたホストの 1 つの仮想スイッチ（仮想ホスト）を表す各オブジェクトを持つ、オブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- BytesReceivedPersec
- BytesSentPersec
- Caption
- ElementName
- Name
- StatusDescriptions

詳細モードでは以下のプロパティが返されます。

- BroadcastPacketsReceivedPersec
- BroadcastPacketsSentPersec
- BytesPersec
- BytesReceivedPersec
- BytesSentPersec
- Caption
- CreationClassName
- Dedicated
- Description
- DirectedPacketsReceivedPersec
- DirectedPacketsSentPersec
- ElementName
- EnabledDefault
- EnabledState
- HealthState
- IdentifyingDescriptions
- InstallDate
- LearnedMacAddresses

- LearnedMacAddressesPersec
- MaxChimneyOffloads
- MaxVMQOffloads
- MulticastPacketsReceivedPersec
- MulticastPacketsSentPersec
- Name
- Name1
- NameFormat
- NumLearnableAddresses
- OperationalStatus
- OtherDedicatedDescriptions
- OtherEnabledState
- OtherIdentifyingInfo
- PacketsFlooded
- PacketsFloodedPersec
- PacketsPersec
- PacketsReceivedPersec
- PacketsSentPersec
- Path
- PowerManagementCapabilities
- PrimaryOwnerContact
- PrimaryOwnerName
- PurgedMacAddresses
- PurgedMacAddressesPersec
- RequestedState
- ResetCapability
- Roles
- ScopeOfResidence
- Status
- StatusDescriptions
- TimeOfLastStateChange

例: ホストスイッチを表示する

この例では、Hyper-V ホストのホスト スイッチ リストを表示します。

```
dpmhv-getHostSwitches -host hvserver
```

dpmhv-getJobInfo (Funclet) コマンド -- ジョブ情報を表示する

dpmhv-getJobInfo コマンドは、完了した割合、エラー情報、および開始時刻などの非同期ジョブに関連するすべての情報を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getJobInfo  
-jobref job  
[-silent]  
[-retval]  
[-detail]
```

-jobref *job*

非同期ジョブのジョブ ID を指定します。

注: 非同期ジョブの実行を完了する際、Hyper-V はジョブ情報を数分間（5 分間）保持します。この保持時間が経過するとこの情報は無効になります。

-silent

画面に出力しないように指定します。

-retval

後で処理できるように値を返します。

-detail

（オプション）非同期ジョブの詳細情報を表示します。

戻り値

`dpmhv-getJobInfo` コマンドは、現在のジョブの状態を表す各オブジェクトを返します。

Hyper-V のジョブの場合

標準モードでは、ローカルの Hyper-V の以下のプロパティが返されます。

- Caption
- ElementName
- ErrorCode
- ErrorDescription
- JobStatus
- PercentComplete
- ResultVMName
- Status
- StatusDescriptions

詳細モードでは、ローカルの Hyper-V の以下のプロパティが返されます。

- Cancellable
- Caption
- DeleteOnCompletion
- Description
- ElapsedTime
- ElementName
- ErrorCode
- ErrorDescription
- ErrorSummaryDescription
- HealthState
- InstallDate
- InstanceID
- JobRunTimes

- JobState
- JobStatus
- LocalOrUtcTime
- Name
- Notify
- Oll- OperationalStatus
- OtherRecoveryAction
- Owner
- Path
- PercentComplete
- Priority
- RecoveryAction
- ResultVMName
- RunDay
- RunDayOfWeek
- RunMonth
- RunStartInterval
- ScheduledStartTime
- StartTime
- Status
- StatusDescriptions
- TimeBeforeRemoval
- TimeOfLastStateChange
- TimeSubmitted
- UntilTime

Microsoft System Center Virtual Machine Manager (SCVMM) のジョブの場合

標準モードでは、SCVMM の以下のプロパティが返されます。

- CurrentStep
- Description

- ErrorInfo
- IsCompleted
- Oll
- ProgressValue
- ResultVMName
- Status
- StatusString

詳細モードでは、SCVMM の以下のプロパティが返されます。

- AdditionalMessages[]
- AreAuditRecordsAvailable
- AuditRecords[]
- CmdletName
- CurrentStep
- Description
- EndTime
- ErrorInfo
- ID
- IsCompleted
- IsFullyCached
- IsRestartable
- IsStoppable
- IsVisible
- MarkedForDeletion
- Name
- Oll
- Owner
- OwnerSID
- PROTipID
- Progress
- ProgressValue

- ResultName
- ResultObjectID
- ResultObjectType
- ResultObjectTypeName
- ResultVMName
- ServerConnection
- Source
- StartTime
- ステータス
- StatusString
- Steps[]
- Target
- TargetObjectID
- TargetObjectType
- WasNotifiedOfCancel

例: ジョブ情報を表示する

この例では、エクスポートジョブの完了のステータスを表示します。

```
jref = dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:¥Ex-async
do
{
    sleep(5000)
    job = dpmhv-getJobInfo -jobref (jref) -silent -retval
    ? "Completion:", job.PercentComplete+"%"
} while(job.ErrorCode==0 && job.PercentComplete<100)
? job.StatusDescriptions
if(job.ErrorCode!=0)
{
    ? job.ErrorDescription
}
```

例: SCVMM のジョブ情報を表示する

この例では、SCVMM テンプレートを使用して作成された新しい VM の完了のステータスを表示します。

```
jref=dpmhv-createVMFromTemplate -host hvserver -vm NewVM -dest c:\VMs\NewBox
-template Win2k3Clean -scvmmHost SCVMMLibSrv -async
do
{
    sleep(5000)
    job = dpmhv-getJobInfo -jobref (jref) -silent -retval
    ? "Completion:", job.ProgressValue+"%"
} while(!job.IsCompleted)
? job.StatusString
s = job.ErrorInfo;
errorCode = parseInt(s.substr(s.lastIndexOf('(')+1).left(-1));
if(errorCode!=0)
{
    ? job.ErrorInfo
}
```

dpmhv-getJobStatus(Funclet)コマンド -- ジョブ ステータスを表示する

dpmhv-getJobStatus コマンドは、非同期ジョブのステータスを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getJobStatus
-jobref job
-jobref job
```

非同期ジョブのジョブ ID を指定します。

注: 非同期ジョブの実行を完了する際、Hyper-V はジョブ情報を数分間（5 分間）保持します。この保持時間が経過するとこの情報は無効になります。

戻り値

`dpmhv-getJobStatus` コマンドは、指定されたジョブの以下のステータス情報を返します。

- キャンセル
- 作成
- 失敗
- 進行中
- 成功
- 不明
- 待機中

例: ジョブステータスを表示する

この例では、エクスポートジョブのステータスを表示します。

```
jref = dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:\¥Ex -async  
dpmhv-getJobState -jref (jref)
```

`dpmhv-getPhysDisks (Funclet) コマンド` -- 物理ディスクをすべて表示する

`dpmhv-getPhysDisks` コマンドは、Hyper-V Server の仮想マシンに接続された物理ディスクをリスト表示します。物理ディスクは、Windows のディスク マネージャではオフラインステータスとして表示されます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getPhysDisks  
-host hostname  
[-silent]  
[-retval]  
[-detail]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) 仮想マシンに接続された物理ディスクの詳細を表示します。

戻り値

`dpmhv-getPhysDisks` コマンドは、指定された仮想マシンの SCSI コントローラを表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- Caption
- DriveNumber
- ElementName
- Name
- Path
- StatusDescriptions

詳細モードでは以下のプロパティが返されます。

- AdditionalAvailability
- Availability
- Capabilities
- CapabilityDescriptions
- Caption
- CompressionMethod
- CreationClassName
- DefaultBlockSize
- Description
- DeviceID
- DriveNumber
- ElementName
- EnabledDefault
- EnabledState
- ErrorCleared
- ErrorDescription
- ErrorMethodology
- HealthState
- IdentifyingDescriptions

- InstallDate
- LastCleaned
- LastErrorCode
- LoadTime
- LocationIndicator
- MaxAccessTime
- MaxBlockSize
- MaxMediaSize
- MaxQuiesceTime
- MaxUnitsBeforeCleaning
- MedialsLocked
- MinBlockSize
- MountCount
- Name
- NeedsCleaning
- NumberOfMediaSupported
- OperationalStatus
- OtherEnabledState
- OtherIdentifyingInfo
- Path
- PowerManagementCapabilities
- PowerManagementSupported
- PowerOnHours
- RequestedState
- Security
- Status
- StatusDescriptions
- StatusInfo
- SystemCreationClassName
- SystemName
- TimeOfLastMount

- TimeOfLastStateChange
- TotalMountTime
- TotalPowerOnHours
- UncompressedDataRate
- UnitsDescription
- UnitsUsed
- UnloadTime

例: 物理ディスクをすべて表示する

この例では、仮想マシンに接続された物理ディスクに関する基本情報を表示します。

```
dpmhv-getPhysDisks -host hvserver
```

dpmhv-getVersion(Funclet)コマンド -- Web サービスのバージョンを表示する

dpmhv-getVersion コマンドは、CA Virtual Assurance の Web サービスのバージョンを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getVersion  
[-retval]  
[-silent]
```

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

dpmhv-getVersion コマンドは、CA Virtual Assurance の Web サービスのバージョンを示す文字列を返します。

例: Web サービスのバージョンを表示する

この例では、CA Virtual Assurance の Web サービスのバージョンを表示します。

```
ver = dpmhv-getVersion -retval -silent  
? ver
```


dpmhv-getVMInfo(Funclet)コマンド -- VM をすべて表示する

dpmhv-getVMInfo コマンドは、仮想マシンの情報および設定を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getVMInfo  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

dpmhv-getVMInfo コマンドは、仮想マシンの情報を持つオブジェクトを返します。

標準モードでは以下のプロパティが返されます。

- vm
- Caption
- ElementName
- Name
- OperationalStatus
- StatusDescriptions
- vminfo
- MemoryUsage
- ProcessorLoad
- ProcessorLoadHistory
- vmsettings
- AutoActivate
- AutomaticRecoveryAction
- AutomaticShutdownAction
- AutomaticStartupAction
- AutomaticStartupActionDelay
- Description
- ElementName
- ExternalDataRoot
- SnapshotDataRoot
- SystemName
- vmcursettings
- BaseBoardSerialNumber
- BIOSGUID
- BIOSSerialNumber
- ChassisAssetTag
- ChassisSerialNumber

- SystemName

詳細モードでは以下のプロパティが返されます。

- vm
- AssignedNumaNodeList
- Caption
- CreationClassName
- Dedicated
- Description
- ElementName
- HealthState
- IdentifyingDescriptions
- InstallDate
- Name
- NameFormat
- OnTimeInMilliseconds
- OperationalStatus
- OtherDedicatedDescriptions
- OtherEnabledState
- OtherIdentifyingInfo
- PowerManagementCapabilities
- PrimaryOwnerContact
- PrimaryOwnerName
- ProcessID
- RequestedState
- ResetCapability
- Roles
- Status
- StatusDescriptions
- TimeOfLastConfigurationChange
- TimeOfLastStateChange
- vminfo

- AsynchronousTasks
- CreationTime
- ElementName
- EnabledState
- GuestOperatingSystem
- HealthState
- Heartbeat
- MemoryUsage
- Name
- Notes
- NumberOfProcessors
- OperationalStatus
- Path
- ProcessorLoad
- ProcessorLoadHistory
- Snapshots
- StatusDescriptions
- ThumbnailImage
- UpTime
- vmsettings
- AdditionalRecoveryInformation
- AllowFullISCSICommandSet
- AutoActivate
- AutomaticRecoveryAction
- AutomaticShutdownAction
- AutomaticStartupAction
- AutomaticStartupActionDelay
- Caption
- CreationTime
- DebugChannelId
- Description

- ElementName
- ExternalDataRoot
- InstanceID
- OtherVirtualSystemType
- Path
- ScopeOfResidence
- SettingType
- SnapshotDataRoot
- SystemName
- Version
- VirtualSystemType
- vmcursettings
- BIOSGUID
- BIOSSerialNumber
- BaseBoardSerialNumber
- ChassisAssetTag
- ChassisSerialNumber
- SystemName

例: VM をすべて表示する

この例では、仮想マシンの情報を表示します。

```
dpmhv-getVMInfo -host hvserver -vm TestVM
```

dpmhv-getVMProperties(Funclet)コマンド -- VM のプロパティを表示する

dpmhv-getVMProperties コマンドは、CPU、メモリ、およびアクションの設定などの VM のプロパティを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getVMProperties  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]
```

`-vm vmname`

VM の名前を指定します。

`-vmid vmguid`

VM の一意の ID を指定します。

`-host hostname`

Hyper-V Server ホストの名前を指定します。

`-retval`

後で処理できるように値を返します。

`-silent`

画面に出力しないように指定します。

戻り値

`dpmhv-getVMProperties` コマンドは、次のような VM プロパティを表す各オブジェクトを返します。

- `memory`
- `cpu`
- `cpuSocketCount`
- `cpuidLimit`
- `cpuReserve`
- `cpuLimit`
- `cpuWeight`
- `actions`
- `startAction`
- `startDelay`
- `stopAction`
- `recoveryAction`

例: VM のプロパティを表示する

この例では、VM プロパティ (VM CPU およびメモリ割り当て) を表示します。

```
dpmhv-getVMProperties -host hvserver -vm TestVM
```

例: VM の CPU およびメモリ割り当てを表示する

この例では、VM「TestVM」の CPU およびメモリ割り当てを表示します。

```
props=dpmhv-getVMProperties -host hvserver -vm TestVM -retval -silent
? "Memory:", props.memory+"MB"
? "CPU cores:", props.cpu.cpuSocketCount
```

dpmhv-getVMState(Funclet)コマンド -- VM 状態を表示する

dpmhv-getVMState コマンドは、VM の電源状態を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-getVMState
-vm vmname|-vmid vmguid
-host hostname
[-retval]
[-silent]
```

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-silent

画面に出力しないように指定します。

-retval

後で処理できるように値を返します。

戻り値

dpmhv-getVMState コマンドは、VM の電源状態を示す文字列を返します。以下の値が返されます。

- disabled : VM はオフになっています
- enabled : VM は実行中です
- paused : VM は一時停止中です
- suspended : VM は中断しています

例: VM 状態を表示する

この例では、VM が実行中かどうかを確認します。

```
s = dpmhv-getVMState -host hvserver -vm TestVM -retval -silent
if(s=="enabled")
? "VM is running"
else
? "VM is not running"
```

dpmhv-importVM (Funclet) コマンド -- VM をインポートする

dpmhv-importVM コマンドは、*dpmhv-exportVM* コマンドを使用してエクスポートされた、エクスポート済みの VM をインポートします。エクスポートされた VM と同じ場所に VM をインポートする場合、VM をインポートできるのは 1 回限りです。

このコマンドの形式は、以下のとおりです。

```
dpmhv-importVM
-host hostname
-path pathname
[-vm vmname]
[-nonewid]
[-async]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-path *pathname*

インポートする VM のパスを指定します。

-vm *vmname*

VM の名前を指定します。

-nonewid

(オプション) インポートする VM に対して一意の GUID を生成しないように指定します。インポートされた VM には、ソース VM (エクスポートされた VM) と同じ GUID があります。同じ GUID を持つ VM が 2 つあると競合が発生するため、お勧めできません。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

dpmhv-importVM が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が dpmhv-getJobErrorInfo、dpmhv-getJobInfo、または dpmhv-getJobStatus に渡されます。同期モードで実行中にエラーが発生した場合、このコマンドはエラーコードを返してエラーをレポートします。

例: VM をインポートする

この例では、dpmhv-exportVM コマンドを使用してエクスポートされた VM をインポートします。

```
dpmhv-exportVM -host hvserver -vm TestVM -dest c:¥Ex¥MyVM dpmhv-importVM  
-host hvserver -path c:¥Ex¥MyVM -vm TestVM1
```

さらに、dpmhv-importVM コマンドを発行して、最初にエクスポートされた VM から複数の VM を作成できます。

dpmhv-importVMEx(Funclet)コマンド -- VM をコピーおよびインポートする

dpmhv-importVMEx コマンドを使用すると、*dpmhv-exportVMEx* コマンドを使用してエクスポートされた、エクスポート済みの VM をインポートしてコピーできます。エクスポートされた VM とは異なる場所に VM をインポートする場合は、何度も VM をインポートできます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-importVMEx
-host hostname
-path pathname
[-dest dest]
[-vm vmname]
[-nonewid]
[-async]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-path *pathname*

インポートしてコピーする VM のパスを指定します。

-dest *dest*

VM をインポートしてコピーするデスティネーションパスを指定します。

-vm *vmname*

VM の名前を指定します。

-nonewid

(オプション) インポートする VM に対して一意の GUID を生成しないように指定します。インポートされた VM には、ソース VM (エクスポートされた VM) と同じ GUID があります。同じ GUID を持つ VM が 2 つあると競合が発生するため、お勧めできません。

-async

(オプション) コマンドを非同期で実行するには、このオプションを指定します。デフォルトでは、このコマンドは同期して実行されます。

注: このオプションを実行する前にこのコマンドの実行が完了した場合、操作ステータスの確認に使用するジョブ ID が返されます。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

`dpmhv-importVMEx` が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が `dpmhv-getJobErrorInfo`、`dpmhv-getJobInfo`、または `dpmhv-getJobStatus` に渡されます。同期モードで実行中にエラーが発生した場合、このコマンドはエラー コードを返してエラーをレポートします。

例: VM をインポートする

この例では、`dpmhv-exportVMEx` コマンドを使用してエクスポートされた VM をインポートします。

```
dpmhv-exportVMEx -host hvserver -vm TestVM -dest c:¥Ex¥MyVM -nosubdir  
dpmhv-importVMEx -host hvserver -path c:¥Ex¥MyVM -dest C:¥VMs¥TestVM1 -vm TestVM1
```

さらに、`dpmhv-importVMEx` コマンドを発行して、最初にエクスポートされた VM から複数の VM を作成できます。

dpmhv-setSysprepProperties (Cmdlet) コマンド -- 仮想ディスクを作成する

dpmhv-setSysprepProperties コマンドを使用すると、仮想ディスク イメージを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-setSysprepProperties
-host hostname
-vm vmname|-vmid vmguid|-jobref job
[-computerName computerName]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-disableAdmin]
[-adminPass adminPW]
[-autoLogon alcnt]
[-duplicatorString dupstr]
[-timeZone timezone]
[-productKey key]
[-userName UserName]
[-organization orgname>]
[-domain domname]
[-domainAdmin domad]
[-domainAdminPass dompw]
[-workgroup wgname]
[-adminUser adminUser]
[-adminUserPass adminUserPW]
[-custom custom]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

デフォルト : `$$HVHost`

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-jobref *job*

非同期ジョブのジョブ ID を指定します。

注: 非同期ジョブの実行を完了する際、Hyper-V はジョブ情報を数分間（5 分間）保持します。この保持時間が経過するとこの情報は無効になります。

-computerName *computerName*

（オプション）コンピュータの名前を指定します。このパラメータをサポートするには、Sysprep ツールを使用するイメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト: Null

-ip4addr *ip4addr*

（オプション）VM のインターフェースに割り当てる静的 IPv4 アドレスを指定します。特定のインターフェースの IP アドレスを設定するには、IP アドレスの前に、既知のインターフェース名と区切り記号として「#」が付けられます。たとえば、

-ip4addr "Local Area Connection#192.168.1.200" などです。テンプレートイメージに複数のネットワーク アダプタがある場合、IP アドレスは最初のインターフェースに割り当てられます。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト: ""

-ip4dhcp *ip4dhcpInt*

（オプション）VM の特定のインターフェースの DHCP をオンにするオプションを指定します。このオプションにはインターフェース名も指定できます。たとえば、

-ip4dhcp "Local Area Connection" などです。テンプレートイメージに複数のネットワーク アダプタがある場合、最初のインターフェースの DHCP がオンになります。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト: local

-ip4mask *ip4mask*

(オプション) VM に割り当てるサブネットマスクを指定します。このオプションは、

-ip4addr オプションと共に使用されます。インターフェース名を -ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : ""

-ip4gw *ip4gw*

(オプション) VM のゲートウェイを設定するオプションを指定します。このオプションは、-ip4addr オプションと共に使用します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

注: インターフェース名を

-ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。

デフォルト : ""

-ip4metric *ip4metric*

(オプション) VM に設定するインターフェースメトリックを指定します。このオプションは、-ip4addr オプションと共に使用します。インターフェース名を

-ip4addr オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : 1

-ip4dns *ip4dns*

(オプション)VM に設定する DNS サーバを指定します。このオプションは、**-ip4addr** オプションと共に使用します。 インターフェース名を **-ip4addr** オプションに指定する場合、このオプションにも同じインターフェース名を使用する必要があります。 このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。 このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : ""

-disableAdmin

(オプション) VM のデフォルトの管理者アカウントを無効にするオプションを指定します。 このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。 このオプションは、コマンドを非同期で実行する場合は無効です。

-adminPass *adminPW*

(オプション) このオプションは、VM のデフォルト管理者アカウントのパスワードを設定するために使用されます。 このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。 非同期実行の場合、このパラメータは無視されます。

注: このオプションを正常に設定するには、**Sysprep** ツールを使用して、**Windows Server 2003** の管理者パスワードを空に設定します。

デフォルト : Null

-autoLogon *alcnt*

(オプション) デフォルト管理者アカウントを使用して自動的にログインするために、作成するアカウント数を設定するオプションを指定します。このアカウントは **Sysprep** プロセスの完了後に作成されます。
hyperv_sp_dupstr このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト :

-1

-duplicatorString *dupstr*

(オプション) VM のレジストリに設定するシステム デュプリケータの名前を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-timeZone *timezone*

(オプション) テンプレートを使用して作成された VM が使用するタイムゾーンを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト :

-1

-productKey *key*

(オプション) VM の Windows 製品アクティベーション キーを指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-userName *UserName*

(オプション) Windows 内部の VM のライセンスが付与されているユーザ名を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-organization *orgname*

(オプション) Windows コピー内部の VM のライセンスが付与されている組織を指定します。このパラメータをサポートするには、Sysprep ツールを使用して作成された Windows イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-domain *domainname*

(オプション) VM のドメイン名を指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。ドメインとワークグループの仕様を同時に指定することはできません。

デフォルト : Null

-domainAdmin *domad*

(オプション) ドメインに作成するドメイン管理者のユーザアカウントを指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-domainAdminPass *dompw*

(オプション) ドメインに作成するドメイン管理者ユーザアカウントのパスワードを指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-workgroup *wgname*

(オプション) VM 用に作成するワークグループを指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。ドメインとワークグループの仕様を同時に指定することはできません。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-adminUser *adminUser*

(オプション) デフォルトの **Administrators** グループの一部として作成する管理者ユーザ名を指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-adminUserPass *adminUserPW*

(オプション) デフォルトの **Administrators** グループの一部として作成する管理者ユーザ アカウントのパスワードを指定します。このパラメータをサポートするには、**Sysprep** ツールを使用して作成された **Windows** イメージが必要です。このオプションは、コマンドを非同期で実行する場合は無効です。

デフォルト : Null

-custom *custom*

(オプション) **Sysprep** プロセスの最後に実行する、カンマ区切りのカスタム コマンドのリストを指定します。このパラメータには、**Sysprep** の **Windows** イメージが必要です。非同期実行の場合、このパラメータは無視されます。

デフォルト : Null

戻り値

dpmhv-setSysprepProperties が同期で実行される場合、このコマンドは空文字列を返します。非同期で実行される場合、ジョブ ID を表す文字列が **dpmhv-getJobErrorInfo**、**dpmhv-getJobInfo**、または **dpmhv-getJobStatus** に渡されます。実行中にエラーが発生した場合は例外がレポートされます。

例: 仮想ディスクを作成する

この例では、2 GB の動的な仮想ディスクを作成します。

```
dpmhv-createVirtDisk -host hvserver -path C:\VMDisks\hdd.vhd -type dynamic -sizeGB 2
```

dpmhv-setVMClustered (Cmdlet) コマンド -- クラスタに VM を設定する

dpmhv-setVMClustered コマンドを使用すると、共有のクラスタ リソースに対して VM を追加または削除できます。Hyper-V Server 環境では、クラスタ内の VM を共有のクラスタ リソースにしたり、Hyper-V Server のプライベート リソースにすることができます。VM がクラスタ共有リソースである場合、クラスタ ノード間でクイック マイグレーションまたはライブ マイグレーションを実行するために VM を有効にすることはできません。また、クラスタ共有ストレージの代わりに、プライベート ストレージ (ローカルのハード ドライブにある仮想ディスクなど) にある VM リソースをマイグレートすることはできません。

このコマンドの形式は、以下のとおりです。

```
dpmhv-setVMClustered  
-host hostname  
-vm vmname|-vmid vmguid  
-clustered {on,off}[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-clustered {on, off}

クラスタの共有モードを指定します。

on

VM は共有のクラスタ リソースです。

off

VM は Hyper-V Server のプライベート リソースです。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: クラスタに VM を設定する

この例では、共有のクラスタ リソースに VM を追加します。

```
dpmhv-setVMClustered -host hvserver -vm TestVM -clustered on
```

dpmhv-setVMDisk (Funclet) コマンド -- ドライブ コントローラにドライブまたはディスクを作成する

dpmhv-setVMDisk コマンドを使用すると、ドライブ コントローラにドライブまたはディスクを作成できます。このオプションを使用して、イメージを接続または切断できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-setVMDisk  
-host hostname  
-vm vmname|-vmid vmguid  
[-path pathname]  
-drive{HDD,DVD}  
-type{scsi,ide}[-id id]  
[-lun lun]  
[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-path *pathname*

(オプション) ドライブ コントローラにドライブまたはディスクを作成する場合のパスを指定します。

-drive {HDD, DVD}

VM に割り当てるドライブ タイプを指定します。以下のドライブ タイプを割り当てることができます。

HDD

ハード ドライブを表します。

DVD

CD (Compact Disk、コンパクト ディスク) または DVD (Digital Video Disk、デジタル ビデオ ディスク) のいずれかのドライブを表します。

-type {scsi, ide}

VM の仮想ハード ディスクに接続するコントローラ タイプを指定します。Hyper-V では、以下の 2 つのタイプをサポートしています。

SCSI

コントローラ タイプとして SCSI を指定します。

IDE

コントローラ タイプとして IDE を指定します。

-id *id*

(オプション) コントローラ タイプで指定したコントローラの ID を指定します。コントローラ タイプが IDE の場合、値は 0 または 1 です。SCSI の場合、VM に追加された SCSI コントローラの数に基づいて、値は 0、1、2、または 3 になります。ID が指定されない (または -1 が指定された) 場合、コマンドは、チャンネルが利用可能である指定されたタイプの最初のコントローラを選択します。

デフォルト :

-1

-lun *lun*

(オプション) <type> および <id> の形式で、コントローラの論理装置を指定します。IDE の場合、値は 0 または 1 です。SCSI の場合は limit [0, 63] が含まれます。装置番号が指定されない (または -1 が指定された) 場合、コマンドは、指定されたコントローラで利用可能なチャンネルを選択します。

デフォルト :

-1

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: SCSI コントローラに仮想ドライブを作成する

この例では、最初の SCSI コントローラの最初のチャンネルに仮想ハードドライブイメージファイルを接続します。

```
dpmhv-setVMDisk -host hvserver -vm TestVM -type scsi -id 0 -lun 0 -drive HDD  
-path c:¥VHDs¥disk.vhd
```

例: VM に DVD を接続する

この例では、VM 「Test VM」 に物理 DVD-ROM ドライブを接続します。

```
dpmhv-setVMDisk -host hvserver -vm TestVM -type ide -id 0 -lun 1 -drive DVD  
-path D:
```

例: SCSI コントローラに物理ホストディスクを接続する

この例では、物理ホストディスクを取得し、最初の利用可能な SCSI チャンネルにそのディスクを接続します。

```
pd=dpmhv-getPhysDisks -host hvserver -retval -silent  
  
dpmhv-setVMDisk -host hvserver -vm TestVM -type scsi -drive HDD -path (pd[0].Path)
```

dpmhv-setVMProperties (Cmdlet) コマンド -- VM のプロパティを設定する

dpmhv-setVMProperties コマンドを使用すると、VM のプロパティを設定できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-setVMProperties
-host hostname
-vm vmname|-vmid vmguid
[-name name]
[-mem mem]
[-cpus cpus]
[-cpuidlimit limit]
[-cpureserve reserve]
[-cpulimit limit]
[-cpuweight weight]
[-startAction {none,auto,always}]
[-startDelay delay]
[-stopAction {save,off,shutdown}]
[-recoveryAction {none,restart,revert}]
[-retval]
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-name *name*

VM の名前を表示します。

-mem *mem*

(オプション) 作成する VM の RAM メモリをメガバイト (MB) で指定します。

デフォルト : -1

-cpus *cpus*

(オプション) VM に割り当てる CPU コアの数指定します。

-cpuidlimit *limit*

(オプション) VM の CPU ID 機能の制限を指定します。このオプションを指定すると、Windows NT のようなレガシーオペレーティングシステムとの互換性が向上します。

デフォルト : -1

-cpureserve *reserve*

(オプション) VM 用に予約する CPU の割合を指定します。このオプションを指定しない場合、Hyper-V Server は全体的なシステムの使用状況に基づいて CPU サイクルを割り当てます。

デフォルト : -1

-cpulimit *limit*

(オプション) VM で使用する CPU リソース数の制限を指定します。このオプションを指定すると、CPU 能力が異なる物理ホスト間で VM を移行できるだけでなく、互換性が向上します。

デフォルト : -1

-cpuweight *weight*

(オプション) 仮想マシンの相対的な加重を指定します。複数の VM が稼働している場合、このオプションはリソースの割り当てを制御します。有効なエントリ : 整数、1 ~ 10000。

デフォルト : -1

-startAction {none,auto,always}

(オプション) Hyper-V を起動する際に VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

何も実行しません。

auto

VM を自動的に起動します。

注: Hyper-V ホストのシャットダウン前に VM が稼働中だった場合は、このオプションを使用します。

always

VM を常に起動します。

-startDelay *delay*

(オプション) Hyper-V ホストが起動し、稼働してから VM を起動するまでの遅延時間を秒数で指定します。

デフォルト : -1

-stopAction {save,off,shutdown}

(オプション) 物理 Hyper-V をシャットダウンする際に VM で実行するアクションを指定します。オプションは、以下のとおりです。

save

VM を中断します。

off

VM の電源をオフにします。

shutdown

システムをシャットダウンします。

-startDelay *delay*

(オプション) Hyper-V ホストが起動し、稼働してから VM を起動するまでの遅延時間を秒数で指定します。

デフォルト : -1

-recoveryAction {none,restart,revert}

(オプション) 予期しないシャットダウンから Hyper-V ホストを再起動した後、VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

何も実行しません。

restart

VM を再起動します。

revert

VM の最後のスナップショットを復元します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: CPU コアの数を変更する

この例では、VM に割り当てる CPU コアの数を変更します。コマンドを正常に実行するには、VM の電源をオフにします。

```
dpmhv-setVMProperties -host hvserver -vm TestVM -cpus 2
```

dpmhv-SetVMVirtFloppy (Cmdlet) コマンド -- VM のフロッピードライブに仮想フロッピーイメージを接続する

dpmhv-SetVMVirtFloppy コマンドを使用すると、Hyper-V ホストの指定された VM フロッピードライブに仮想フロッピーイメージを接続できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-setVMVirtFloppy  
-host hostname  
-vm vmname|-vmid vmguid  
[-path pathname]  
[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-path *pathname*

(オプション) 作成する仮想フロッピーイメージ (*.vfd) ファイルのパスを指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

例: Hyper-V ホストの VM フロッピー ドライブにフロッピー イメージを接続する

この例では、Hyper-V ホスト「TestVM」の VM フロッピー ドライブにフロッピー イメージ「floppy.vfd」を接続します。

```
dpmhv-setVMVirtFloppy
-host hvserver
-vm TestVM
-path c:\img\floppy.vfd
```

例: Hyper-V ホストの VM フロッピー ドライブからフロッピー イメージの接続を解除する

この例では、Hyper-V ホストの VM フロッピー ドライブからフロッピー イメージの接続を解除します。

```
dpmhv-setVMVirtFloppy -host hvserver -vm TestVM
```

dpmhv-ShowClusterSharedVolumes (Funclet) コマンド -- クラスタ共有ボリュームをすべて表示する

dpmhv-ShowClusterSharedVolumes コマンドは、Hyper-V Server のすべてのクラスタ共有ボリュームを表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showClusterSharedVolumes
-host hostname
[-retval]
[-silent]

-host hostname
```

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

dpmhv-ShowClusterSharedValumes コマンドは、オブジェクトの配列から現在のキーと値ペアを返します。

例: クラスタ共有ボリュームをすべて表示する

この例では、ホスト「hvserver」の共有ボリュームのリストを返します。

```
dpmhv-showClusterSharedVolumes -host hvserver
```

dpmhv-ShowDirectories(Funclet)コマンド -- ディレクトリをすべて表示する

dpmhv-ShowDirectories コマンドは、システムの指定されたディレクトリの下にあるサブディレクトリをすべて表示します。指定するシステムには、Hyper-V Server またはリモート WMI を有効にした Windows システムを使用できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showDirectories  
-host hostname  
-dir dir  
[-retval]  
[-silent]  
[-detail]  
  
-host hostname
```

Hyper-V Server ホストの名前を指定します。

-dir *dir*

ディレクトリ名を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) ディレクトリ リストの詳細を表示します。

戻り値

dpmhv-ShowDirectories コマンドは、指定された親ディレクトリの下にあるサブディレクトリを表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- Caption
- Drive
- FileName
- Name
- Path
- Writeable \$<<prop_detail
- AccessMask
- Archive
- CSCreationClassName
- CSName
- Caption
- Compressed
- CompressionMethod
- CreationClassName
- CreationDate
- Description
- Drive
- EightDotThreeFileName
- Encrypted
- EncryptionMethod
- Extension
- FSCreationClassName
- FSName
- FileName
- FileSize
- FileType
- Hidden

- InUseCount
- InstallDate
- LastAccessed
- LastModified
- Name
- Path
- Readable
- Status
- System
- Writeable

例: ディレクトリをすべて表示する

この例では、ホスト hvserver のルートディレクトリ (c : ドライブ) 下にあるサブディレクトリを表示します。

```
dpmhv-showDirectories
-host hvserver
-dir c:¥
```

例: Program Files の下のディレクトリをすべて表示する

この例では、「C:¥Program Files」のサブディレクトリを表示します。

```
function walkDir(sPath)
{
  dirs = dpmhv-showDirectories -host hvserver -dir (sPath) -retval -silent
  for each (subdir in dirs)
  {
    ? subdir.Name
    walkDir(subdir.Name);
  }
}
walkDir("C:¥Program Files");
```

dpmhv-ShowHosts(Funclet) コマンド -- Hyper-V Server をすべて表示する

dpmhv-ShowHosts コマンドは、Hyper-V Server をすべて表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showHosts
[-detail]
[-retval]
[-silent]
```

-detail

(オプション) 仮想ネットワーク (ホスト ネットワーク) の詳細を表示します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

dpmhv-ShowHosts コマンドは、Hyper-V Server を表すオブジェクトの配列を返します。標準モードでは「Name」プロパティが返されます。

詳細モードでは以下のプロパティが返されます。

- AssignedNumaNodeList
- CIMV2_AdminPasswordStatus
- CIMV2_AutomaticManagedPagefile
- CIMV2_AutomaticResetBootOption
- CIMV2_AutomaticResetCapability
- CIMV2_BootOptionOnLimit
- CIMV2_BootOptionOnWatchDog
- CIMV2_BootROMSupported
- CIMV2_BootupState
- CIMV2_Caption
- CIMV2_ChassisBootupState
- CIMV2_CreationClassName
- CIMV2_CurrentTimeZone
- CIMV2_DNSHostName
- CIMV2_DaylightInEffect
- CIMV2_Description
- CIMV2_Domain
- CIMV2_DomainRole
- CIMV2_EnableDaylightSavingsTime
- CIMV2_FrontPanelResetStatus
- CIMV2_InfraredSupported
- CIMV2_InitialLoadInfo
- CIMV2_InstallDate
- CIMV2_KeyboardPasswordStatus
- CIMV2_LastLoadInfo
- CIMV2_Manufacturer
- CIMV2_Model

- CIMV2_Name
- CIMV2_NameFormat
- CIMV2_NetworkServerModeEnabled
- CIMV2_NumberOfLogicalProcessors
- CIMV2_NumberOfProcessors
- CIMV2_OEMLogoBitmap
- CIMV2_OEMStringArray
- CIMV2_PCSysType
- CIMV2_PartOfDomain
- CIMV2_PauseAfterReset
- CIMV2_PowerManagementCapabilities
- CIMV2_PowerManagementSupported
- CIMV2_PowerOnPasswordStatus
- CIMV2_PowerState
- CIMV2_PowerSupplyState
- CIMV2_PrimaryOwnerContact
- CIMV2_PrimaryOwnerName
- CIMV2_ResetCapability
- CIMV2_ResetCount
- CIMV2_ResetLimit
- CIMV2_Roles
- CIMV2_Status
- CIMV2_SupportContactDescription
- CIMV2_SystemStartupDelay
- CIMV2_SystemStartupOptions
- CIMV2_SystemStartupSetting
- CIMV2_SystemType
- CIMV2_ThermalState
- CIMV2_TotalPhysicalMemory
- CIMV2_UserName
- CIMV2_WakeUpType

- CIMV2_Workgroup
- Caption
- ClusterName
- CreationClassName
- Dedicated
- Description
- ElementName
- EnabledDefault
- EnabledState
- HealthCritical
- HealthOk
- HealthState
- HostCPUUtilisationPercent
- HostMemoryAvailableMB
- IdentifyingDescriptions
- InstallDate
- IsHyperVClusterNode
- NET_ArpAlwaysSourceRoute
- NET_ArpUseEtherSNAP
- NET_Caption
- NET_DHCPEnabled
- NET_DHCPLeaseExpires
- NET_DHCPLeaseObtained
- NET_DHCPServer
- NET_DNSDomain
- NET_DNSDomainSuffixSearchOrder
- NET_DNSEnabledForWINSResolution
- NET_DNSHostName
- NET_DNSServerSearchOrder
- NET_DatabasePath
- NET_DeadGWDetectEnabled

- NET_DefaultIPGateway
- NET_DefaultTOS
- NET_DefaultTTL
- NET_Description
- NET_DomainDNSRegistrationEnabled
- NET_ForwardBufferMemory
- NET_FullDNSRegistrationEnabled
- NET_GatewayCostMetric
- NET_IGMPLevel
- NET_IPAddress
- NET_IPConnectionMetric
- NET_IPEnabled
- NET_IPFilterSecurityEnabled
- NET_IPPortSecurityEnabled
- NET_IPSecPermitIPProtocols
- NET_IPSecPermitTCPPorts
- NET_IPSecPermitUDPPorts
- NET_IPSubnet
- NET_IPUseZeroBroadcast
- NET_IPXAddress
- NET_IPXEnabled
- NET_IPXFrameType
- NET_IPXMediaType
- NET_IPXNetworkNumber
- NET_IPXVirtualNetNumber
- NET_Index
- NET_InterfaceIndex
- NET_KeepAliveInterval
- NET_KeepAliveTime
- NET_MACAddress
- NET_MTU

- NET_NumForwardPackets
- NET_PMTUBHDetectEnabled
- NET_PMTUDiscoveryEnabled
- NET_ServiceName
- NET_SettingID
- NET_TcpMaxConnectRetransmissions
- NET_TcpMaxDataRetransmissions
- NET_TcpNumConnections
- NET_TcpUseRFC1122UrgentPointer
- NET_TcpWindowSize
- NET_TcpipNetbiosOptions
- NET_WINSEnableLMHostsLookup
- NET_WINSHostLookupFile
- NET_WINSPrimaryServer
- NET_WINSScopeID
- NET_WINSSecondaryServer
- Name
- NameFormat
- OnTimeInMilliseconds
- OperationalStatus
- OtherDedicatedDescriptions
- OtherEnabledState
- OtherIdentifyingInfo
- Path
- PowerManagementCapabilities
- PrimaryOwnerContact
- PrimaryOwnerName
- ProcessID
- RequestedState
- ResetCapability
- Roles

- Status
- StatusDescriptions
- TimeOfLastConfigurationChange
- TimeOfLastStateChange
- VSMS_BiosLockString
- VSMS_Caption
- VSMS_DefaultExternalDataRoot
- VSMS_DefaultVirtualHardDiskPath
- VSMS_Description
- VSMS_ElementName
- VSMS_InstanceID
- VSMS_MaximumMacAddress
- VSMS_MinimumMacAddress
- VSMS_NumaSpanningEnabled
- VSMS_PrimaryOwnerContact
- VSMS_PrimaryOwnerName
- VSMS_ScopeOfResidence

例: Hyper-V Server をすべて表示する

この例では、Hyper-V Server のリストを返します。

```
dpmhv-showHosts
```

例: ホスト上の Hyper-V Server および VM をすべて表示する

この例では、Hyper-V Server のリストと各ホストの VM を表示します。

```
a = dpmhv-showHosts -retval -silent
for each(h in a)
{
  ?
  ? "Host:", h.Name
  dpmhv-showVMs -host (h.Name)
}
```

dpmhv-ShowLogicalDisks (Funclet) コマンド -- 論理ディスクをすべて表示する

dpmhv-ShowLogicalDisks コマンドは、Windows システムに接続されている論理ディスクをすべて表示します。指定するシステムには、Hyper-V Server またはリモート WMI を有効にした Windows システムを使用できます。このコマンドを使用すると、リモートの Hyper-V Server の Windows ファイルシステムの VM の場所がわかります。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showLogicalDisks  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-host *hostname*

ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) 論理ディスク リストの詳細を表示します。

戻り値

dpmhv-ShowLogicalDisks コマンドは、指定されたホストに接続された論理ディスクを表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- Caption
- DeviceID
- FreeSpace
- Name
- Path
- Size

詳細モードでは以下のプロパティが返されます。

- Access
- Availability
- BlockSize
- Caption
- Compressed
- ConfigManagerErrorCode
- ConfigManagerUserConfig
- CreationClassName
- Description
- DeviceID
- DriveType
- ErrorCleared
- ErrorDescription
- ErrorMethodology
- FileSystem
- FreeSpace
- InstallDate
- LastErrorCode
- MaximumComponentLength
- MediaType

- Name
- NumberOfBlocks
- PNPDeviceID
- Path
- PowerManagementCapabilities
- PowerManagementSupported
- ProviderName
- Purpose
- QuotasDisabled
- QuotasIncomplete
- QuotasRebuilding
- Size
- Status
- StatusInfo
- SupportsDiskQuotas
- SupportsFileBasedCompression
- SystemCreationClassName
- SystemName
- VolumeDirty
- VolumeName
- VolumeSerialNumber

例: 論理ディスクをすべて表示する

この例では、ホストに接続された論理ディスクのリストを返します。

```
dpmhv-showLogicalDisks -host hvserver
```


dpmhv-ShowSCVMMHardwareProfiles (Funclet) コマンド -- SCVMM ハードウェア プロファイルをすべて表示する

dpmhv-ShowSCVMMHardwareProfiles コマンドは、Microsoft System Center Virtual Machine Manager (SCVMM) によって定義されたハードウェア プロファイルのリストを表示します。このリストは、SCVMM テンプレートをを使用して VM を作成するとき場合に VM で利用可能です。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showSCVMMHardwareProfiles
-scvmmHost scvmmHostname
[-retval]
[-silent]
[-detail]
-scvmmHost scvmmHostname
```

(オプション) Microsoft System Center Virtual Machine Manager (SCVMM) ライブラリ サーバのホスト名を指定します。VM のプロビジョニングで SCVMM の統合を使用する場合に、このパラメータは有効です。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) ディレクトリ リストの詳細を表示します。

戻り値

dpmhv-ShowSCVMMHardwareProfiles コマンドは、指定されたライブラリ サーバの下で定義されたハードウェア プロファイル表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- CPUCount
- Description
- Memory
- Name

詳細モードでは以下のプロパティが返されます。

- Accessibility
- AddedTime
- BootOrder[]
- Connection
- CPUCount
- CPUMax
- CPUReserve
- CPUType
- ID
- IsFullyCached
- MarkedForDeletion
- Name
- ServerConnection
- Description
- DiskIO
- Enabled
- ExpectedCPUUtilization
- HostDrive
- ID

- IsFullyCached
- IsHighlyAvailable
- LimitCPUForMigration
- LimitCPUFunctionality
- MarkedForDeletion
- Memory
- ModifiedTime
- MostRecentTask
- Name
- NetworkUtilization
- NumLockEnabled
- ObjectType
- Owner
- RelativeWeight
- ServerConnection
- CEIPOptIn
- CPUPriority
- Channel
- CompanyName
- DatabaseInstanceName
- DatabaseName
- DatabaseServerName
- DiskIOPriority
- EvaluationDaysLeft
- FQDN
- FullyQualifiedDomainName
- IsConnected
- IsEvaluationVersion
- IsWorkgroupEdition
- LibraryRefresherEnabled
- LibraryRefresherFrequency

- MOMReportingEnabled
- MOMReportingServerURL
- MemoryPriority
- MinimumSupportedAgentVersion
- NetworkPriority
- ObjectCache
- OpsMgrReportingEnabled
- OpsMgrReportingServerURL
- OpsMgrServer
- PROAutomationLevel
- PROMonitoringLevel
- PhysicalAddressRangeEnd
- PhysicalAddressRangeStart
- PlacementGoal
- ProductID
- ProductVersion
- Profile
- SelfServiceContactEmail
- ServerInterfaceVersion
- UserName
- VMConnectDefaultPort
- VMRCAccessAccount
- VMRCDefaultPort
- ShareSCSIBus
- UndoDisksEnabled
- VirtualCOMPorts[]
- VirtualDVDDrives[]
- VirtualFloppyDrive
- ServerConnection
- VirtualFloppyDisk
- VirtualNetworkAdapters[]

- VirtualSCSIAdapters[]

例: SCVMM のハードウェア プロファイルをすべて表示する

この例では、SCVMM のライブラリ サーバ SCVMMLibSrv のハードウェア プロファイルを表示します。

```
dpmhv-showSCVMMHardwareProfiles
-scvmmHost SCVMMLibSrv
```

例: 複数の CPU のハードウェア プロファイルを表示する

この例では、複数の CPU コアのために定義されたハードウェア プロファイルをすべて表示します。

```
profs = dpmhv-showSCVMMHardwareProfiles -scvmmHost SCVMM -retval -silent
for each(prof in profs)
{
  if(prof.CPUCount > 1)
    ? prof.Name, "has more than 1 CPU cores"
}
```

dpmhv-ShowSCVMMOSProfiles (Funclet) コマンド -- ゲスト OS プロファイルをすべて表示する

dpmhv-ShowSCVMMOSProfiles コマンドは、SCVMM テンプレートを使用して VM を作成する間に VM で利用可能な、ゲストオペレーティングシステム (OS) プロファイルをすべて表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showSCVMMOSProfiles
-scvmmHost scvmmHostname
[-retval]
[-silent]
[-detail]

-scvmmHost scvmmHostname
```

(オプション) Microsoft System Center Virtual Machine Manager (SCVMM) ライブラリ サーバのホスト名を指定します。VM のプロビジョニングで SCVMM の統合を使用する場合に、このパラメータは有効です。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) ディレクトリ リストの詳細を表示します。

戻り値

dpmhv-ShowSCVMMOSProfiles コマンドは、指定された SCVMM ライブラリ サーバのゲスト OS プロファイルを表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- 管理
- AdminPasswordHasValue
- ComputerName
- Description
- DomainAdmin
- DomainAdminPasswordHasValue
- JoinDomain
- JoinWorkgroup
- Name
- OperatingSystem
- OrgName
- ProductKeyHasValue

詳細モードでは以下のプロパティが返されます。

- Accessibility
- AddedTime
- 管理
- AdminPasswordHasValue
- ComputerName
- 説明
- DomainAdmin
- DomainAdminPasswordHasValue
- Enabled
- FullName
- GuiRunOnceCommands
- ID
- IsFullyCached

- JoinDomain
- JoinWorkgroup
- MarkedForDeletion
- MergeAnswerFile
- ModifiedTime
- MostRecentTask
- Name
- ObjectType
- OperatingSystem
- AllowsOrgNameInSysprep
- Architecture
- Description
- Edition
- ID
- IsCustomizationAllowed
- IsFullyCached
- MarkedForDeletion
- Name
- ProductType
- RequiresAdministratorAccountNameInSysprep
- RequiresPIDInSysprep
- RequiresXMLSysprepFormat
- ServerConnection
- Version
- OrgName
- Owner
- ProductKeyHasValue
- ServerConnection
- CEIPOptIn
- CPUPriority
- Channel

- CompanyName
- DatabaseInstanceName
- DatabaseName
- DatabaseServerName
- DiskIOPriority
- EvaluationDaysLeft
- FQDN
- FullyQualifiedDomainName
- IsConnected
- IsEvaluationVersion
- IsWorkgroupEdition
- LibraryRefresherEnabled
- LibraryRefresherFrequency
- MOMReportingEnabled
- MOMReportingServerURL
- MemoryPriority
- MinimumSupportedAgentVersion
- Name
- NetworkPriority
- ObjectCache
- OpsMgrReportingEnabled
- OpsMgrReportingServerURL
- OpsMgrServer
- PROAutomationLevel
- PROMonitoringLevel
- PhysicalAddressRangeEnd
- PhysicalAddressRangeStart
- PlacementGoal
- ProductID
- ProductVersion
- Profile

- SelfServiceContactEmail
- ServerInterfaceVersion
- UserName
- VMConnectDefaultPort
- VMRCAccessAccount
- VMRCDefaultPort
- SysprepScript
- TimeZone

例: ゲスト OS プロファイルをすべて表示する

この例では、SCVMM のライブラリ サーバ SCVMMLibSrv のゲスト OS プロファイルを表示します。

```
dpmhv-showSCVMMOSProfiles  
-scvmmHost SCVMMLibSrv
```

例: パスワード持たないゲスト OS プロファイルをすべて表示する

この例では、管理者パスワードを持たないゲスト OS プロファイルを表示します。

```
profs = dpmhv-showSCVMMOSProfiles -scvmmHost SCVMMLibSrv -retval -silent  
for each(prof in profs)  
{  
  if(prof.AdminPasswordHasValue=="false")  
    ? prof.Name,"does not have an Admin password defined"  
}
```

dpmhv-ShowTemplates (Funclet) コマンド --Hyper-V ホストのテンプレートを表示する

dpmhv-ShowTemplates コマンドは、Hyper-V ホストまたは Microsoft System Center Virtual Machine Manager (SCVMM) ライブラリ サーバのテンプレートのリストを返します。テンプレートとは、テンプレートカタログで管理され、新しい VM をプロビジョニングするために使用される、エクスポート済みの VM を表します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showTemplates  
[-host hostname]  
[-scvmmHost scvmmHost]  
[-detail]  
[-retval]  
[-silent]
```

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-scvmmHost *scvmmHost*

(オプション) SCVMM ライブラリ サーバの名前を指定します。このパラメータは、指定されたライブラリ サーバで、SCVMM を使用して外部的に作成および管理されているテンプレートを返します。このパラメータは、SCVMM 統合を使用して VM をプロビジョニングする場合に有効です。

-detail

(オプション) テンプレート リストの詳細を表示します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

dpmhv-ShowTemplates コマンドは、テンプレートを表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- ElementName
- Name

詳細モードでは以下のプロパティが返されます。

- DiskspaceRequired
- ElementName
- MemoryAllocation
- Name
- NetworkInterfaceAddresses
- NetworkInterfaces
- NumberOfProcessors
- TemplateCreationDate
- TemplateDescription
- TemplateFilePath
- TemplateSourceVM

例: VM テンプレートのリストを取得する

この例では、VM テンプレートのリストを返します。

```
dpmhv-showTemplates  
-host hvserver
```

例: テンプレートとプロセスの詳細情報を取得する

この例では、テンプレートとプロセスの詳細情報を持つテンプレートリストを返します。

```
tt = dpmhv-showTemplates -host hvserver -silent -retval  
for each(t in tt)  
? t.ElementName, t.Description
```

dpmhv-ShowVMDisks (Funclet) コマンド -- 仮想ディスクをすべて表示する

dpmhv-ShowVMDisks コマンドは、Hyper-V ホスト サーバの VM の仮想ディスクをすべて表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showVMDisks  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) 仮想ディスク リストの詳細を表示します。

戻り値

dpmhv-ShowVMDisks コマンドは、VM に接続されたディスクを表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- Caption
- Connection
- ElementName
- HostResource
- ResourceSubType

詳細モードでは以下のプロパティが返されます。

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- VirtualQuantity

- VirtualSystemIdentifiers
- Weight

例: VM の仮想ディスクをすべて表示する

この例では、VM に接続された仮想ディスクのリストを返します。

```
dpmhv-showVMDisks  
-host hvserver  
-vm TestVM
```

例: ディスクおよび仮想ハードドライブ イメージ ファイルをすべて表示する

この例では、VM に接続されたディスクおよび仮想ハード ドライブ イメージ ファイルのリストを返します。

```
dd = dpmhv-showVMDisks -host hvserver -vm TestVM -detail -retval -silent  
for each(d in dd)  
{  
  if(d.ResourceSubType=="Microsoft Virtual Hard Disk")  
  {  
    ? d.Connection  
  }  
}
```

dpmhv-showVMFloppy(Funclet) コマンド -- フロッピードライブのプロパティを表示する

dpmhv-showVMFloppy コマンドを使用すると、Hyper-V ホストのすべての VM に接続されたすべてのフロッピー ドライブに関する情報を取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showVMFloppy  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

`-host hostname`

Hyper-V Server ホストの名前を指定します。

`-retval`

後で処理できるように値を返します。

`-silent`

画面に出力しないように指定します。

`-detail`

(オプション) VM のフロッピー ドライブの詳細情報を表示します。

戻り値

dpmhv-showVMFloppy コマンドは、VM のフロッピー ドライブ プロパティの詳細情報を表すオブジェクトを返します。

標準モードでは以下のプロパティが返されます。

- Caption
- Connection
- ElementName

詳細モードでは以下のプロパティが返されます。

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- VirtualQuantity
- VirtualSystemIdentifiers
- Weight

例: VM のフロッピードライブをすべて表示する

この例では、VM に接続されたフロッピー ドライブのリストを返します。

```
dpmhv-showVMFloppy -host hvserver -vm TestVM
```

dpmhv-showVMNICs (Funclet) コマンド -- ネットワーク インターフェース カードをすべて表示する

dpmhv-showVMNICs コマンドは、Hyper-V ホストの VM にある NIC (Network Interface Cards、ネットワーク インターフェース カード) をすべて表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showVMNICs  
-vm vmname|-vmid vmguid  
-host hostname  
[-retval]  
[-silent]  
[-detail]
```

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) NIC リストの詳細を表示します。

戻り値

dpmhv-showVMNICs コマンドは、オブジェクトを返します。

標準モードでは以下のプロパティが返されます。

- Address
- ElementName
- StaticMacAddress

詳細モードでは以下のプロパティが返されます。

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- StaticMacAddress
- VirtualQuantity
- VirtualSystemIdentifiers

■ Weight

例: VM の NIC 情報を表示する

この例では、ホスト「hvserver」の VM「TestVM」にある NIC 情報を表示します。

```
dpmhv-showVMNICs -host hvserver -vm TestVM
// VM の最初のアダプタの接続先である仮想ネットワークを検索する
anic = dpmhv-showVMNICs -host hvserver -vm TestVM -detail -silent -retval
lines = anic[0].Connection.split(",");
for each(line in lines)
{
    if(line.startsWith("SystemName="))
    {
        nsw = line.substr(11);
        nsw = nsw.trim('');
        break;
    }
}
// スイッチを取得する
asw = dpmhv-getHostSwitches -host hvserver -silent -retval
// 一致するものを検索する
for each(sw in asw)
{
    if(sw.Name == nsw)
    {
        ?? "NIC",anic[0].ElementName
        ? " is connected to switch", sw.ElementName
    }
}
```

dpmhv-showVMSCSI (Funclet) コマンド -- SCSI コントローラをすべて表示する

dpmhv-showVMSCSI コマンドは、Hyper-V ホストの VM にある SCSI コントローラをすべて表示します。Hyper-V Server には最大 4 つの SCSI コントローラを割り当てることができます。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showVMSCSI
-vm vmname|-vmid vmguid
-host hostname
[-retval]
[-silent]
[-detail]
```

-vm *vmname*

VM の名前を指定します。

-vmid *vmguid*

VM の一意の ID を指定します。

-host *hostname*

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

-detail

(オプション) SCSI コントローラ リストの詳細を表示します。

戻り値

dpmhv-showVMSCSI コマンドは、指定された VM の SCSI コントローラを表すオブジェクトの配列を返します。

標準モードでは以下のプロパティが返されます。

- Caption
- ElementName

詳細モードでは以下のプロパティが返されます。

- Address
- AllocationUnits
- AutomaticAllocation
- AutomaticDeallocation
- Caption
- Connection
- ConsumerVisibility
- Description
- ElementName
- HostResource
- InstanceID
- Limit
- MappingBehavior
- OtherResourceType
- Parent
- Path
- PoolID
- Reservation
- ResourceSubType
- ResourceType
- VirtualQuantity
- VirtualSystemIdentifiers
- Weight

例: SCSI コントローラをすべて表示する

この例では、ホスト「hvserver」の VM「TestVM」にあるすべての SCSI コントローラの基本的な詳細を表示します。

```
dpmhv-showVMSCSI  
-host hvserver  
-vm TestVM
```

例: すべての SCSI コントローラの詳細を表示する

この例では、ホスト「hvserver」の VM「TestVM」にある SCSI コントローラの詳細を表示します。

```
dpmhv-showVMSCSI -host hvserver -vm TestVM -detail
```

dpmhyperv ShowVMs(Funclet)コマンド -- VM をすべて取得する

dpmhv-showVMs コマンドは、Hyper-V ホストサーバの VM のリストを返します。このコマンドは、指定された VM の名前と対応する GUID を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhv-showVMs  
-host hostname[-retval]  
[-silent]  
  
-host hostname
```

Hyper-V Server ホストの名前を指定します。

-retval

後で処理できるように値を返します。

-silent

画面に出力しないように指定します。

戻り値

`dpmhv-showVMs` コマンドは、指定されたホストの VM を表すオブジェクトの配列を返します。

オブジェクトには以下のプロパティが含まれます。

`guid`

VM の一意の ID を表します。

`name`

VM の表示名を表します。

例: VM をすべて取得する

この例では、ホスト「`hvserver`」の VM と、それに対応する GUID のリストを返します。

```
dpmhv-showVMs  
-host hvserver
```

例: すべての VM の VM プロパティを表示する

この例では、ホスト「`hvserver`」にあるすべての VM のプロパティを返します。

```
a = dpmhv-showVMs -host hvserver -retval -silent  
for each(vm in a)  
{  
  ?  
  ? "VM:", vm.ElementName  
  dpmhv-getVMProperties -host hvserver -vmid (vm.Name)  
}
```

CA IBM LPAR AutoShell コマンド

スクリプトに **AutoShell** を使用して **CA IBM LPAR** コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドを CLI でも利用できます。

dpmlpar-cycle コマンド -- 論理パーティションのサイクル(Funclet)

dpmlpar-cycle コマンドは、論理パーティションの電源オン、電源オフ、リセット、中断を行います。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-cycle
-powerop {activate|restart|shutdown}
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-type {delayed|immediate|os_shutdown|immediate_os_shutdown}]
[-profile_name profilename]
[-activate_bootmode {normal|open_firmware}]
[-activate_keylock {normal|manual}]
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-powerop {activate|restart|shutdown}

LPAR 上で実行する電源操作を指定します。オプションは、以下のとおりです。

activate

LPAR をオンにします。

restart

LPAR をオフにした後、必要に応じてオンにします。

シャットダウン

LPAR をオフにします。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*immediate*|*os_shutdown*|*immediate_os_shutdown*}

イメージング操作タイプ リソース グループ (*res_group*) と個別リソース (*individual_res*) のどちらを使用するかを指定します。オプションは、以下のとおりです。

immediate

パーティションをすぐにシャットダウンします。データが部分的にしか更新されていない場合、このオプションは望ましくない結果を引き起こす場合があります。

os_shutdown

標準のシャットダウン コマンドを発行してパーティションをシャットダウンします。このオプションを正常に実行するには、パーティションのイメージが作成されている必要があります。

immediate_os_shutdown

オペレーティング システムのシャットダウン コマンドを発行して、できるだけ短時間でパーティションをシャットダウンします。このコマンドは、他のユーザへのメッセージ送信など、標準のシャットダウン アクティビティを省略します。このオプションを正常に実行するには、パーティションのイメージが作成されている必要があります。

-profile_name *lparprofile* (HMC のみ)**[-profile_name** *lparname*] (IVM のみ)

論理パーティションをアクティブ化するときに使用するパーティション プロファイルを指定します。HMC には必須です。IVM については任意で、指定した場合には、その名前がパーティション名と一致する必要があります。

-activate_bootmode {normal|open_firmware}

(オプション) アクティブ化操作のキーロック モードを指定します。オプションは、以下のとおりです。

normal

標準の方法でパーティションを開始します。

open_firmware

パーティションを開始し、ファームウェア プロンプトを開きます。

-activate_keylock {normal|manual|bypass}

(オプション) アクティブ化操作のキーロック モードを指定します。オプションは、以下のとおりです。

normal

自動モードでパーティションを開始します。アクティブ化にはユーザ操作を必要としません。

手動

手動モードでパーティションを開始します。アクティブ化にはユーザ操作が必要となります。

bypass

キーロック モードをアクティブにしません。

-verbose *add_commandinfo*

コマンドの実行方法に関する追加情報を提供します。

-sc *URL*

(オプション) サービス コントローラの **URL** を指定します。

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: デフォルトプロファイルを使用して論理パーティションをアクティブ化する

以下の例は、デフォルトプロファイルを使用して論理パーティション **testlpar** をアクティブ化します。

```
dpmlpar-cycle -powerop activate -hmc uslihmc -managed_system testComputer  
-partition_name testlpar -profile_name testlparprofile
```

例: 指定したプロファイルを使用して論理パーティションをアクティブ化する

以下の例は、プロファイル **testlparprofile** を使用して論理パーティション **testlpar** をアクティブ化します。

```
dpmlpar-cycle -powerop activate -hmc uslihmc -managed_system testComputer  
-partition_name testlpar -profile_name testlparprofile
```

例: 論理パーティションをアクティブ化し、ファームウェア プロンプトを開く

以下の例は、プロファイル **testlparprofile** を使用して論理パーティション **testlpar** をアクティブ化し、ファームウェア プロンプトを開きます。

```
dpmlpar-cycle -powerop activate -hmc usliivm -managed_system testMS  
-partition_name testlpar -activate_bootmode open_firmware
```

例: 論理パーティションの遅延シャットダウン

以下の例は、論理パーティション **testlpar** の遅延シャットダウンを実行します。

```
dpmlpar-cycle -hmc testivm -powerop shutdown -managed_system managedsys1  
-partition testlpar -type delayed
```

dpmlpar-delete コマンド -- 論理パーティションの削除 (Funclet)

dpmlpar-delete コマンドは、非アクティブな論理パーティションを削除します。

重要: このコマンドを発行する *前に*、重要なデータをバックアップしたことを確認してください。このコマンドは、LPAR および LPAR のデータ ファイルを削除します。このコマンドを発行する際は、LPAR の電源をオフにする必要があります。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-delete
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-hmc name

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system managedsystemname

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name partitionname

削除する論理パーティションの名前を指定します。このパーティションは、管理対象システム上に存在する必要があります。

-verbose add_commandinfo

コマンドの実行方法に関する追加情報を提供します。

-sc URL

(オプション) サービス コントローラの URL を指定します。

-ws_remote_user username

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password password

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: 論理パーティションを削除する

以下の例は、論理パーティション `lpar01` を削除します。

```
dpmlpar-delete -hmc hmc02 -managed_system system05 -partition_name lpar01
```

dpmlpar-getresources コマンド -- LPAR リソースの取得(Funclet)

`dpmlpar-getresources` コマンドは、IBM AIX LPAR のメモリ ユニット、プロセッサ ユニット、プロセッサを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-getresources
-resource {all|memory|processors|processor_units}
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]

-resource {all|memory|processors|processor_units}
```

特定のリソースを取得するのか、すべてのリソースを取得するのかを指定します。オプションは、以下のとおりです。

all

すべてのパーティション リソースを取得、表示します。

memory

パーティションのメモリ リソースのみを取得、表示します。

processors

パーティションのプロセッサ リソースのみを取得、表示します。

processor_units

パーティションのプロセッサ ユニット リソースのみを取得、表示します。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースをリスト表示する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-verbose *add_commandinfo*

コマンドの実行方法に関する追加情報を提供します。

-sc *URL*

(オプション) サービス コントローラの URL を指定します。

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: LPAR のすべてのリソースを取得する

以下の例は、lpar06 のリソースをすべて取得します。

```
dpmlpar-getresources -resource all -partition_name lpar06  
-hmc hmc_02 -managed_system abcsystem4
```

例: LPAR のメモリリソースを取得する

以下の例は、lpartest01 のメモリ リソースを取得します。

```
dpmlpar-getresources -resource memory -partition_name lpartest01  
hmc ivm_02 -managed_system abcsystem2
```

dpmlpar-image コマンド -- IBM AIX LPAR の作成 (Funclet)

dpmlpar-image コマンドは、HMC とターゲット IBM AIX LPAR 管理対象システムを使用して IBM AIX LPAR を作成します。LPAR を作成した後、オプションで IBM AIX オペレーティングシステムを展開できます。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-image  
-hmc name -managed_system managedsystemname  
-partition_name partitionname  
-profile_name profilename  
-min_mem size  
-desired_mem size  
-max_mem size  
  
-proc_mode {shared|dedicated}  
-min_proc_units units  
-desired_proc_units units  
-max_proc_units units  
-min_procs number  
-desired_procs number  
-max_procs number  
  
-max_virtual_slots number  
-share_mode {capped|uncapped}  
[-uncap_weight weight]  
[-io_slot DRC index, {true|false}]  
[-lpar_io_pool_id id {,id, ...}]
```



```

[-virtual_serial_adapter slot_num, remote_lpar_name, remote_slot_num, is_required]
[-virtual_scsi_client_adapter slot_num, remote_lpar_name, remote_slot_num,
is_required] |
[-virtual_scsi_server_client_adapter virtual_io_server_name, backing_device_name,
client_slot_num, is_required]
[-virtual_eth_adapter slot_num, is_IEEE, port_vlan_id, (additional_vlan_id,
additional_vlan_id, ...), trunk_priority, is_required]

[-virtual_fc_server_client_adapter virtual_io_server_name, physical_fc_port_name,
client_slot_num, is_required]
[virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required; ...]
[-verbose add_commandinfo]

[-create_logical_volume <false|true>]
[-logical_volume_size <value>]
[-volume_group_names <value>]
[-logical_volume_name <value>]
[-use_logical_volume_name_as_prefix {no|yes}]
[-default_vio_server_name vioserver]

[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]

```

リソースグループを使用して AIX をプロビジョニングするための追加パラメータ

```

-provision_aix true
-type res_group
-res_group_name resourcegroupname
-machine_res_name machineresourcenname
-target_username targetusername
[-target_password targetpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
-nim_master_host_name nimmasterhostname
[-scalability_server servername]
[-deploy_template templatenname]
-auto_deploy {yes|no}
[-wait [timeout]]

```

個別リソースを使用して AIX をプロビジョニングするための追加パラメータ

```
-provision_aix true
-type individual_res
-machine_res_name machineresourcename
-lpp lppresource
-spot spotresource
-bosinst_data bosdata
-resolv_conf resolveconf
-fb_script fbscript
-post_inst_scripts script1,script2,script3
-target_username targetusername
[-target_password targetpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
-nim_master_host_name nimmasterhostname
[-scalability_server servername]
[-deploy_template templatename]
-auto_deploy {yes|no}
[-wait [timeout]]

-hmc name
```

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

```
-managed_system managedsystemname
```

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

```
-partition_name partitionname
```

作成する論理パーティションの名前を指定します。

```
-profile_name lparprofile (HMC のみ)
```

```
[-profile_name lparname] (IVM のみ)
```

論理パーティションをアクティブ化するときに使用するパーティションプロファイルを指定します。HMC には必須です。IVM については任意で、指定した場合には、その名前がパーティション名と一致する必要があります。

```
-min_mem size
```

パーティションの最小メモリ容量を定義します。

-desired_mem size

パーティションの推奨メモリ容量を定義します。

-max_mem size

パーティションの最大メモリ容量を定義します。

-proc_mode {shared|dedicated}

パーティションのプロセッサ モードのタイプを指定します。オプションは、以下のとおりです。

shared

他のパーティションとプロセッサ リソースを共有します。

dedicated

専用のプロセッサ リソースがパーティションにあることを指定します。

-min_proc_units units

このパーティションの最小プロセッサ ユニット数を定義します。

値の範囲： .01 ずつインクリメント

注： このオプションは共有プロセッサでのみ使用できます。

-desired_proc_units units

このパーティションに割り当てられているプロセッサ ユニット数を定義します。

値の範囲： .01 ずつインクリメント

注： このオプションは共有プロセッサでのみ使用できます。

-max_proc_units units

このパーティションの最大プロセッサ ユニット数を定義します。

値の範囲： .01 ずつインクリメント

注： このオプションは共有プロセッサでのみ使用できます。

-min_procs number

このパーティションの最小仮想プロセッサ数を定義します。

注： このオプションは共有プロセッサでのみ使用できます。

-desired_procs *number*

このパーティションに割り当てられている仮想プロセッサ数を定義します。

注: このオプションは共有プロセッサでのみ使用できます。

-max_procs *number*

このパーティションの最大仮想プロセッサ数を定義します。

注: このオプションは共有プロセッサでのみ使用できます。

-shared_mode {capped|uncapped}

管理対象システムで、別のパーティションにコミットしていない、共有プロセッサプールのアイドルプロセッサユニットを論理パーティションで使えるようにするかどうかを指定します。

注: このオプションは共有プロセッサでのみ使用できます。

オプションは、以下のとおりです。

capped

パーティションにコミットされているプロセッサユニットのみを使用できます。

uncapped

利用可能な場合は共有プロセッサプールのアイドルプロセッサユニットをパーティションで使用できます。

-uncap_weight *weight*

(オプション) 共有モードとして **uncapped** を選択した場合に、処理優先度の加重平均を定義します。

値の範囲: 0 ~ 255

-max_virtual_slots *number*

このパーティションの最大仮想アダプタ数を定義します。

デフォルト: 2

値の範囲: 2 ~ 65,536

-io_slot DRC-Index,{true|false}

(オプション) 物理コンポーネントの I/O スロットを指定します。異なる I/O デバイスにこの値を複数回指定できます。

DRC-Index

スロットの動的再構成コネクタ (DRC) インデックスを指定します。

true

I/O スロットに DRC インデックスが必須であることを指定します。

false

I/O スロットに DRC インデックスは必須でないことを指定します。

-io_pool_id id {,id ...}

(オプション) HMC のアクティブな介入なしで、指定された論理パーティショングループが引き継いで使用できる I/O アダプタグループを定義します。パーティショングループは、I/O プール ID をカンマで区切って指定します。

-virtual_serial_adapter slot_num, remote_lpar_name, remote_slot_num, is_required

(オプション) この論理パーティションの仮想シリアルアダプタを定義します。変数 **is_required** は、**true** または **false** に設定します。サポートされているのはクライアントシリアルアダプタのみですが、複数回指定できます。

デフォルト: スロット 1 と 2 で作成された 2 つのサーバシリアルアダプタ

注: IVM 上で LPAR を作成する場合にはサポートされていません。

-virtual_eth_adapter_client slot_num, is_IEEE, port_vlan_id1 (vlanid2,vlanid3,...), trunk_priority, is_required

(オプション) この論理パーティションの仮想イーサネットアダプタを定義します。複数回指定できます。オプションの値は **vlan_id** と **trunk_priority** です。変数 **is_IEEE** と **is_required** は、**true** または **false** に設定します。追加の **vlan_id** は、カンマ区切りかつこで囲む必要があります。

注: IVM 上で LPAR を作成するとき、**slot_num** は 4 以上である必要があります。

例: 4,false,1,,,true

-virtual_scsi_client_adapter slot_num, remote_lpar_name, remote_slot_num, is_required

(オプション) この論理パーティションの仮想クライアント SCSI アダプタを定義します。複数回指定できます。変数 *is_required* は、true または false に設定します。

-virtual_scsi_server_client_adapter virtual_io_server_name, backing_device_name, client_slot_num, is_required

(オプション) 仮想 SCSI サーバデバイスと仮想 SCSI クライアントデバイスを定義します。この定義は、サーバデバイス作成後に自動的にリンクされ、サーバスロット番号が特定されます。このパラメータは複数回指定できます。変数 *is_required* は、true または false に設定します。

例 : CUST-VIOSERVER,hdisk5,3,true

注: `dpmlpar image` コマンドを使用して論理ボリュームを作成し、`-create_logical_volume` オプションを「yes」に設定した場合、`-virtual_scsi_server_client_adapter` オプションは不要で省略されます。

-virtual_fc_server_client_adapter virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required [;virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required;...]

(オプション) 仮想ファイバチャネル (FC) クライアントアダプタ (VIO サーバ上の FC サーバアダプタ) を定義し、物理 FC ポートへ FC サーバアダプタを関連付けます。`-provision_aix` オプションが true に設定されている場合、このオプションは無効です。このパラメータは複数回指定できます。変数 *is_required* は、true または false に設定します。

-create_logical_volume {no|yes}

(オプション) 論理ボリュームを作成するかどうかを指定します。

デフォルト : no

注: `dpmlpar image` コマンドを使用して論理ボリュームを作成し、`-create_logical_volume` オプションを「yes」に設定した場合、`-virtual_scsi_server_client_adapter` オプションは不要で省略されます。

-logical_volume_size logical_volume_size

(オプション) 論理ボリュームのサイズを MB で定義します。

-volume_group_names volume_group_1, volume_group_2, volume_group_3

(オプション) 論理ボリュームグループ名をカンマ区切りのリストで指定します。

-logical_volume_name *logical_volume_name*

(オプション) 論理ボリュームの名前を定義します。

-use_logical_volume_name_as_prefix {no|yes}

(オプション) プレフィックスとして論理ボリューム名を使用するかどうかを指定します。

デフォルト: no

-default_vio_server_name *vioserver*

(オプション) 論理ボリュームの作成に使用する VIO サーバを指定します。

-itcm_server *itcm_servername*

(オプション) Unicenter DSM サーバの名前を指定します。

-sc *URL*

(オプション) サービスコントローラの URL を指定します。

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモートユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモートパスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

プロビジョニングに必須のパラメータ

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、**dpmutil** 認証情報設定コマンドを使用して生成した **CA Virtual Assurance** 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-auto_deploy {yes|no}

CA Virtual Assurance エージェントを自動的に展開するかどうかを指定します。 オプションは、以下のとおりです。

yes

CA Virtual Assurance エージェントを自動的に展開します。

no

CA Virtual Assurance エージェントが自動的に展開されないようにします。

デフォルト : no

-bosinst_data *bos_install_data_resource*

ベース オペレーティング システム (BOS) のインストール プログラムに関する情報を含むファイルを指定します。 -install_type が mksysb である場合はオプションです。 -install_type が rte である場合は必須です。

-deploy_template *templatename*

(オプション) CA Virtual Assurance で作成された汎用テンプレートの名前を指定します。

注: VMware vCenter で作成および管理されているテンプレートと、このテンプレートを混同しないでください。

-fb_script *fbscript*

(オプション) BOS のインストール プロセスが完了した後、NIM クライアントが初めて起動するときにデバイスの設定に使用するファイルの名前を定義します。

-lpp *lpp_resource*

イメージング要求に使用するライセンス プログラム製品ファイルを指定します。 -install_type が mksysb である場合はオプションです。 -install_type が rte である場合は必須です。

-machine_res_name *machineresourcename*

マシン リソースの名前を定義します。 この名前は NIM マスタで事前定義されている必要があります。

-nim_master_host_name *nimmasterhostname*

イメージ展開を実行する NIM マスタ ホスト名を定義します。

-post_inst_scripts *script1,script2,script3*

(オプション) インストール後に実行するスクリプトをカンマ区切りのリストで指定します。

-provision_aix {true|false}

パーティションを作成した後、NIM を使用してイメージを作成するかどうかを指定します。

true

作成したパーティションのイメージに NIM を使用します。true に設定すると、NIM リソース グループまたは NIM の個別リソース パラメータが使用されます。詳しくは、「dpmnim image コマンド - リソース グループを使用した IBM AIX イメージの展開」、「dpmnim image コマンド - 個別リソースを使用した IBM AIX イメージの展開」を参照してください。ジョブ ID は、NIM プロビジョニングが開始されると返されます。

false

作成したパーティションのイメージに NIM を使用しません。ジョブ ID は返されません。

-res_group_name *resource group name*

リソース グループの名前を定義します。

-resolve_conf *resolveconf*

(オプション) 有効な */etc/resolv.conf* エントリが含まれているファイルを定義します。このエントリで、ローカル リゾルバルーチンのドメイン名プロトコル ネーム サーバ情報を定義します。

-scalability_server *servername*

(オプション) ソフトウェア配信用の Unicenter DSM スケーラビリティ サーバを指定します。

-spot *spotResource [mksysbResource]*

イメージング要求に使用する共有製品オブジェクト ツリーを定義します。

-target_password *targetpassword*

(オプション) イメージを展開するターゲット ホスト サーバにエージェントを展開するパスワードを定義します。パスワードを指定しない場合は、認証ファイルから取得されます。

注: 認証ファイルの設定には **dpmutil CLI** を使用します。

-target_username *targetusername*

イメージを展開するターゲット ホスト サーバにエージェントを展開するユーザ名を定義します。

-type {*res_group*|*individual_res*}

イメージング操作タイプ リソース グループと個別リソースのどちらを使用するかを指定します。

res_group

イメージングでリソース グループを使用するように指定します。

individual_res

イメージングで個別リソースを使用するように指定します。

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (**timeout** の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの **timeout** の値 (分) がタイムアウトするまでの時間となります。 **timeout** の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほかに、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-verbose *add_commandinfo*

コマンドの実行方法に関する追加情報を提供します。

例: IBM AIX 論理パーティションを作成する

この例では、論理パーティションを作成しますが、プロビジョニングは行いません。

```
dpmlpar-image -hmc ivm01 -managed_system testMS -partition_name lpartest01
-profile_name Default -max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5 -proc_mode shared
-share_mode uncapped -uncap_weight 123 -min_proc_units .5 -max_proc_units 3.25
-desired_proc_units 1.75 -virtual_serial_adapter 2,serial_partition,1,true
-virtual_scsi_client_adapter 3,scsi_partition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true -io_slot 21020003,false
```

例: IBM AIX 論理パーティションを作成し、個別リソースを使用してイメージングする

この例では、論理パーティションを作成し、個別リソースを使用してプロビジョニングを行います。

```
dpmlpar-image -hmc hmc01 -managed_system testMS -partition_name IRTEST
-profile_name Default -max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 1 -max_procs 1 -proc_mode dedicated
-virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),0,true -io_slot 21020003,false
-provision_aix true -type individual_res -lpp 530lpp_res -spot 530spot_res
-bosinst_data 530_bid_ow -resolv_conf master_net_conf -post_inst_scripts piScript
-machine_res_name MachineName -nim_master_host_name machine.mydomain.com
-auto_deploy no -target_username root -target_password password
```

例: IBM AIX 論理パーティションを作成し、リソースグループを使用してイメージングする

この例では、論理パーティションを作成し、リソースグループを使用してプロビジョニングを行います。

```
dpmlpar-image -hmc hmc01 -managed_system testMS -partition_name RGTEST
-profile_name Default -max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5 -proc_mode dedicated
-virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true -io_slot 21020003,false
-provision_aix true -type res_group -res_group_name Res_grp -machine_res_name
Machine
-nim_master_host_name machine.mydomain.com -auto_deploy no
-target_username root -target_password password
```

dpmlpar-imgjobcheck コマンド -- IBM AIX イメージング ジョブ ステータスの取得 (Funclet)

dpmlpar-imgjobcheck コマンドは、特定の CA Virtual Assurance ジョブ ID の IBM AIX 論理パーティション展開イメージング ジョブ ステータスを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-imgjobcheck -status jobID [-verbose add_commandinfo]
```

-status *jobID*

ジョブ ステータスの取得に使用する CA Virtual Assurance ジョブ ID を定義します。

-verbose *add_commandinfo*

コマンドの実行方法に関する追加情報を提供します。

例: ジョブ ID を使用して IBM AIX イメージング ジョブ ステータスを取得する

以下の例は、CA Virtual Assurance ジョブ ID 42 を使用して、IBM AIX 論理パーティション イメージング ジョブ ステータスを取得します。

```
dpmlpar-imgjobcheck -status 42
```

dpmlpar-list コマンド -- HMC リソースのリスト表示 (Funclet)

dpmlpar-list コマンドは、ハードウェア管理コンソールの情報をリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-list
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-display {managed_systems | managed_system_details | partitions | io_components |
backing_devices | scsi_adapters | partition_details | profiles | wwpns}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-display

{managed_systems|managed_system_details|partitions|io_components|backing_devices|scsi_adapters|partition_details|profiles|wwpns}

リスト表示するリソースを指定します。オプションは、以下のとおりです。

managed_systems

管理ハードウェア コンソールで制御する、管理対象システムをリスト表示します。

managed_system_details

管理対象システムに関する詳細情報を提供します。

partitions

管理対象システムのパーティションをリスト表示します。

io_components

管理対象システムの I/O コンポーネントをリスト表示します。

backing_devices

サーバ仮想 SCSI アダプタへのバックング デバイスとして接続できるデバイスを一覧表示します。

scsi_adapters

管理対象システムの SCSI アダプタをリスト表示します。

partition_details

パーティションの詳細をリスト表示します。

profiles

パーティションのプロファイルをリスト表示します。

wwpns

パーティションのワールドワイドポート名を表示します。

-hmc name

情報をリストする HMC/IVM を指定します。

-managed_system *managedsystemname*

(オプション) 情報をリスト表示する論理パーティションが存在する、管理対象システムを指定します。

注: managed_systems オプションと共に使用することはできません。

-partition_name *partitionname*

(オプション) 情報をリスト表示する論理パーティションを指定します。

注: managed_systems オプションと共に使用することはできません。

-verbose *add_commandinfo*

コマンドの実行方法に関する追加情報を提供します。

-sc *URL*

(オプション) サービスコントローラの URL を指定します。

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: 管理対象システムのリストを表示する

以下の例は、ハードウェア管理コンソール labhmc の管理対象システムを表示します。

```
dpmlpar-list -display managed_systems -hmc labhmc
```

例: SCSI アダプタを表示する

以下の例は、管理対象システム testMS の SCSI アダプタを表示します。

```
dpmlpar-list -display scsi_adapters -hmc labhmc  
-managed_system testMS
```

例: パーティションの詳細を表示する

以下の例は、論理パーティション `testlpar` の詳細を表示します。

```
dpmlpar-list -display partition_details -hmc uslihmc  
-managed_system testMS -partition_name testlpar
```

dpmlpar-setresources コマンド -- メモリ リソースの追加 (Funclet)

`dpmlpar-setresources` コマンドは、IBM AIX LPAR のメモリ ユニットを調整します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-setresources  
-add_memory value  
-hmc name  
-managed_system managedsystemname  
-partition_name partitionname  
-type {dynamic|all}  
[-verbose add_commandinfo]  
[-sc URL]  
[-ws_remote_user username]  
[-ws_remote_password password]  
[-pre]  
[-post]
```

-add_memory *value*

パーティションに追加するメモリの量を定義します。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {dynamic|all}

調整が一時的なものか永続的なものを指定します。 オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-verbose add_commandinfo

コマンドの実行方法に関する追加情報を提供します。

-sc URL

(オプション) サービス コントローラの URL を指定します。

-ws_remote_user username

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password password

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: 論理パーティションにメモリを追加する

以下の例は、論理パーティション `lptest01` に 128 MB のメモリを追加します。

```
dpm\par-setresources -add_memory 128 -partition_name lptest01  
-hmc uni02 -managed_system usil01system1 -type dynamic
```


例：論理パーティションにメモリを追加し、プロファイルを更新する

以下の例は、論理パーティション `lptest01` に 128 MB のメモリを追加し、パーティションプロファイルファイルを更新します。

```
dpmlpar-setresources -add_memory 128 -partition_name lptest01
-hmc uni02 -managed_system usil01system1 -type all -profile_name lptest01
```

dpmlpar-setresources コマンド -- プロセッサ リソースの追加 (Funclet)

`dpmlpar-setresources` コマンドは、IBM AIX LPAR のプロセッサおよびプロセッサユニットを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-setresources
{-add_processors|add_processor_units} value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-add_processors value

パーティションに追加するプロセッサの数を定義します。

-add_processor_units value

パーティションに追加するプロセッサユニットを定義します。

値の範囲： .01 ずつインクリメント

-hmc name

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system managedsystemname

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*dynamic*|*all*}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-verbose *add_commandinfo*

コマンドの実行方法に関する追加情報を提供します。

-sc *URL*

(オプション) サービス コントローラの **URL** を指定します。

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: パーティションにプロセッサを追加し、プロファイルを更新する

以下の例は、パーティション **lptest01** にプロセッサを追加し、パーティションプロファイルを更新します。

```
dpmlpar-setresources -add_processors 1 -partition_name lptest01  
-hmc uni02 -managed_system MSystem1 -type all -profile_name lptest01
```

例: パーティションにプロセッサとプロセッサ ユニットを追加する

以下の例は、パーティション `lptest01` にプロセッサとプロセッサ ユニット リソースを一時的に追加します。

```
dpmlpar-setresources -add_processors 1 -add_processor_units 4.5  
partition_name lptest01 -hmc uni02 -managed_system MSystem1 -type dynamic
```

dpmlpar-setresources コマンド -- メモリ リソースの削除 (Funclet)

`dpmlpar-setresources` コマンドは、IBM AIX LPAR のメモリ ユニットを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-setresources  
-subtract_memory value  
-hmc name  
-managed_system managedsystemname  
-partition_name partitionname  
-type {dynamic|all}  
[-verbose add_commandinfo]  
[-sc URL]  
[-ws_remote_user username]  
[-ws_remote_password password]  
[-pre]  
[-post]
```

-subtract_memory *value*

パーティションから減らすメモリの量を定義します。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {dynamic|all}

調整が一時的なものか永続的なものかを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-verbose add_commandinfo

コマンドの実行方法に関する追加情報を提供します。

-sc URL

(オプション) サービスコントローラの URL を指定します。

-ws_remote_user username

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password password

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: 論理パーティションからメモリを削除する

以下の例は、論理パーティション `lptest01` から 128 MB のメモリを削除します。

```
dpm\par-setresources -subtract_memory 128 -partition_name lptest01  
-hmc con02 -managed_system MSsystem1 -type dynamic
```

例：論理パーティションからメモリを削除し、プロファイルを更新する

以下の例は、論理パーティション `lptest01` から 128 MB のメモリを削除し、パーティションプロファイルファイルを更新します。

```
dpmlpar-setresources -subtract_memory 128 -partition_name lptest01
-hmc con02 -managed_system MSsystem1 -type all
```

dpmlpar-setresources コマンド -- プロセッサ リソースの削除 (Funclet)

`dpmlpar-setresources` コマンドは、IBM AIX LPAR からプロセッサおよびプロセッサユニットを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar-setresources
{-subtract_processors|subtract_processor_units} value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-verbose add_commandinfo]
[-sc URL]
[-ws_remote_user username]
[-ws_remote_password password]
[-pre]
[-post]
```

-subtract_processors value

パーティションから減らすプロセッサの数を定義します。

-subtract_processor_units value

パーティションから減らすプロセッサユニットを定義します。

値の範囲： .01 ずつインクリメント

-hmc name

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system managedsystemname

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*dynamic*|*all*}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-verbose *add_commandinfo*

コマンドの実行方法に関する追加情報を提供します。

-sc *URL*

(オプション) サービス コントローラの URL を指定します。

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: パーティションからプロセッサを削除し、プロファイルを更新する

以下の例は、パーティション `lpartest01` からプロセッサを削除し、パーティションプロファイルを更新します。

```
dpmlpar-setresources -subtract_processors 1 -partition_name lpartest01  
-hmc hmc02 -managed_system MSsystem1 -type all
```

例: パーティションからプロセッサとプロセッサ ユニートを削除する

以下の例は、パーティション `lptest01` のプロセッサとプロセッサ ユニトリソースを一時的に調整します。

```
dpmldpar-setresources -subtract_processors 1 -subtract_processor_units 4.5  
-partition_name lptest01 -hmc ivm02 -managed_system MSystem1 -type dynamic
```

CA Oracle Solaris ゾーン AutoShell コマンド

スクリプトに **AutoShell** を使用して **CA Oracle Solaris** ゾーン コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドを **CLI** でも利用できます。

dpmzone-associateproject コマンド -- タスクとプロジェクトの関連付け(Funclet)

dpmzone-associateproject コマンドは、タスクとプロジェクトを関連付けます。プロジェクトとタスクは、**Solaris 10** オペレーティング システムのワークロードの識別に使用される基本的なエンティティです。プロジェクトはタスクの集合であり、タスクはプロセスの集合です。プロセスはそれぞれ **1** つのタスクのみに属し、タスクはそれぞれ **1** つのプロジェクトのみに属します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-associateproject  
-host hostname -name zonename  
-proj_name pname -task_id tid  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

プロジェクトを提供するゾーンを指定します。

-proj_name *pname*

プロジェクトを指定します。

-task_id *tid*

タスクの ID 番号を指定します。

例

タスクとプロジェクトを関連付けます。

```
dpmzone-associateproject -host SolarisServer2 -name myzone1 -project_name myproject1  
-task_id 1954
```

dpmzone-clonezone コマンド -- ゾーンのクローン作成 (Funclet)

dpmzone-clonezone コマンドは、既存のゾーンを複製したゾーンを作成します。クローン作成プロセスを開始するには、ソースゾーンを停止する必要があります。クローン作成を Solaris 10 サーバ上で行う場合は、時間がかかる場合があります。

このコマンドの形式は、以下のとおりです。

```
dpmzone-clonezone  
-host hostname  
-name zonename  
-new_name nzonename  
-new_path npath
```

-host *hostname*

Solaris ゾーンサーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

クローンを作成するゾーンを指定します。

-new_name *nzonename*

新しいゾーン名を定義します。

-new_path *npath*

グローバルゾーンから現在のゾーンへのパス名を定義します。

例

ホスト SolarisServer2 にゾーンのクローンを作成します。

```
dpmzone-clonezone -host SolarisServer2 -name myzone1 -new_name myzone2  
-new_path /opt/zones/myzone2
```


dpmzone-createandinstallzone コマンド -- ゾーンの作成およびインストール

createandinstallzone コマンドは、Solaris ホスト上でカスタム パラメータを使用してゾーンを作成し、インストールします。

このコマンドの形式は、以下のとおりです。

```
dpmzone-createandinstallzone  
[-sc sc_host]  
-host hostname  
-name zonename  
-path zonepath  
-type type  
[-archive_path path]  
[-autoboot]  
[-if_type name]  
[-ip ip]  
[-pool_name pool]  
[-sched_type sched]  
[-phy_mem pmem]  
[-swap_mem smem]  
[-lock_mem lmem]  
[-desc desc]  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

新しいゾーンの名前を定義します。

-path *zonepath*

新しいゾーンのパスを定義します。

-type *type*

新しいゾーンのタイプを指定します。オプションは、以下のとおりです。

native

アプリケーションを実行する、Solaris 10 オペレーティング環境の非グローバルゾーンを作成します。

whole-root

パッケージを継承しない完全ルート ゾーンを作成します。

branded

アプリケーションを実行する、非ネイティブ オペレーティング環境を含む非グローバルゾーンを作成します。

デフォルト : native

-archive_path *path*

(オプション) ブランド化ゾーンのオペレーティング環境インストーラのパスを指定します。

制限 : この引数はブランド化ゾーンでは必須ですが、ネイティブゾーンでは任意です。

-autoboot

(オプション) システム起動時にゾーンを自動的に起動するように指定します。

注 : ゾーン サービスがサーバ上で無効になっている場合、このプロパティの設定に関係なく、ゾーンは自動起動されません。サービスを有効化するには、以下のコマンドを使用します。

```
svcadm enable svc:/system/zones:default
```

-if_type *name*

(オプション) ゾーンで使用するネットワーク インターフェースのタイプを指定します (例 : eri0)。

-ip *ip*

(オプション) ゾーンの IP アドレスを定義します。

-pool_name *pool*

(オプション) ゾーンを関連付けるプールの名前を指定します。

-sched_type *shed*

(オプション) 共有に基づいた CPU 時間の割り当てに使用する、スケジューラのタイプを指定します。共有は、プロジェクトに割り当てられたシステムの CPU リソース部分です。有効な値は以下のとおりです。

ts

タイム共有スケジューラを指定します。これにより、すべてのプロセスに平等に CPU リソースを分配し、特定のプロセスに CPU リソースを集中させません。ts は Solaris オペレーティング環境用のデフォルト スケジューラです。

fss

フェア シェア スケジューラを指定します。これにより、共有に基づいて CPU 時間を割り当てることができます。

-phy_mem *pmem*

(オプション) ゾーンに割り当てる物理メモリを定義します。この数値にはスケール (K、M、G、T) を適用できます。たとえば、1M は 1 メガバイトです。

-swap_mem *smem*

(オプション) ゾーンに割り当てるスワップメモリを定義します。

-lock_mem *lmem*

(オプション) ゾーンに割り当てるロック済みメモリを定義します。ロック済みメモリはページングできません。

-desc *description*

(オプション) ゾーンの説明を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmzone-createpool コマンド -- リソース プールの作成 (Funclet)

dpmzone-createpool コマンドは、Solaris 10 ホスト上でリソース プールを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-createpool
-host hostname
-pset_name pset
-cpu_min mincpu
-cpu_max maxcpu
-pool_name pool
-sched_type sched
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-pset_name *pset*

プロセッサセットを定義します。各プロセッサセット (CPU グループ) には、ゼロ個以上の CPU を含むことができます。

-cpu_min *mincpu*

プロセッサセットの CPU の最小数を定義します。

-cpu_max *maxcpu*

プロセッサセットの CPU の最大数を定義します。

-pool_name *pool*

ゾーンを関連付けるプールの名前を定義します。

-sched_type *shed*

共有に基づいた CPU 時間の割り当てに使用する、スケジューラのタイプを指定します。共有されるのは、プロジェクトに割り当てられたシステム CPU リソースの一部です。オプションは、以下のとおりです。

ts

すべてのプロセスに適正に CPU リソースを割り当て、特定のプロセスに CPU リソースを集中させないタイム共有スケジューラを指定します。ts は Solaris オペレーティング環境のデフォルトスケジューラです。

fss

共有に基づいて CPU 時間を割り当てることができる、フェアシェアスケジューラを指定します。

例

ホスト SolarisServer2 上でリソース プール POOL1 を作成します。

```
dpmzone-createpool -host SolarisServer2 -pset_name PSET1 -cpu_min 1 -cpu_max 20
-pool_name POOL1 -sched_type FSS
```

dpmzone-createproject コマンド -- プロジェクトの作成 (Funclet)

dpmzone-createproject コマンドは、ゾーンでプロジェクトを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-createproject
-host hostname -name zonename
-user_id uid
[-project_id pid]
-proj_name pname
-host hostname
```

Solaris ゾーンサーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

ゾーン名を指定します。

-user_id *uid*

ゾーンのユーザを 1 人指定します。

例：root

-project_id pid

(オプション) プロジェクト ID を定義します。ID を割り当てるか、またはシステムによって自動的に ID を生成させることができます。

-proj_name pname

新しいプロジェクトの名前を定義します。

例

ゾーンでプロジェクトを作成します。

```
dpmzone-createproject -host SolarisServer2 -name myzone1 -user_id root  
-proj_name myProject
```

dpmzone-createzone コマンド -- ゾーン作成(Funclet)

dpmzone-createzone コマンドは、Solaris 10 ホスト上でカスタム パラメータを使用してゾーンを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-createzone  
-host hostname  
-name zonename  
-path zonepath  
-type type  
[-autoboot]  
[-if_type name]  
[-ip ip]  
[-pool_name pool]  
[-sched_type sched]  
[-phy_mem pmem]  
[-swap_mem smem]  
[-lock_mem lmem]  
[-desc desc]
```

-host hostname

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name zonename

新しいゾーンの名前を定義します。

-path zonepath

グローバル ゾーンから現在のゾーンへのパス名を定義します。

-type *type*

新しいゾーンのタイプを指定します。オプションは、以下のとおりです。

native

アプリケーションを実行する、Solaris 10 オペレーティング環境の非グローバルゾーンを作成します。

whole-root

パッケージを継承しない完全ルート ゾーンを作成します。

branded

アプリケーションを実行する、非ネイティブ オペレーティング環境を含む非グローバルゾーンを作成します。

デフォルト : native

-autoboot

システム起動時にゾーンを自動的に起動するように指定します。

注: ゾーン サービスがサーバ上で無効になっている場合、このプロパティの設定に関係なく、ゾーンは自動起動されません。サービスを有効化するには、以下のコマンドを使用します。

```
svcadm enable svc:/system/zones:default
```

-if_type *name*

ゾーンで使用するネットワーク インターフェースを指定します（例 : eri0）。

-ip *ip*

（オプション）ゾーンの IP アドレスを定義します。

-pool_name *pool*

ゾーンに関連付けるプールを指定します。

-sched_type *shed*

共有に基づいて、CPU 時間を割り当てるために使用されるスケジューラのタイプを指定します。共有は、プロジェクトに割り当てられたシステムの CPU リソース部分です。オプションは、以下のとおりです。

ts

タイム共有スケジューラを指定します。これにより、すべてのプロセスに平等に CPU リソースを分配し、特定のプロセスに CPU リソースを集中させません。ts は Solaris オペレーティング環境用のデフォルトスケジューラです。

fss

フェアシェアスケジューラを指定します。これにより、共有に基づいて CPU 時間を割り当てることができます。

-phy_mem *pmem*

ゾーンに割り当てる物理メモリを定義します。この数値にはスケール (K、M、G、T) を適用できます。たとえば、1M は 1 メガバイトです。

-swap_mem *smem*

ゾーンに割り当てるスワップメモリを定義します。

-lock_mem *lmem*

ゾーンに割り当てるロック済みメモリを定義します。ロック済みメモリはページングできません。

-desc *description*

(オプション) ゾーンの説明を定義します。

例

WebServer4 という名前のゾーンを作成します。

```
dpmzone-createzone -host MySolarisServer -name WebServer4 -path /Zones/Apache
-type native -autoboot -if_type eri0 -ip 192.168.100.100 -pool_name WebServer
-sched_type fss -phy_mem 1024M -swap_mem 2048M
```


dpmzone-customcommand コマンド -- カスタム コマンドの実行(Funclet)

dpmzone-customcommand コマンドでは、Solaris サーバ上で外部のコマンドまたはスクリプトを実行できます。

注: 制限付きの外部コマンドの詳細については、このガイドの付録「Custom Commands」を参照してください。

このコマンドの形式は、以下のとおりです。

```
dpmzone-customcommand
-host hostname
-cmd command
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-cmd *command*

実行するプログラムまたはスクリプトを定義します。

例

ホスト SolarisServer2 上でコマンドを実行します。

```
dpmzone-customcommand -host SolarisServer2 -cmd
/usr/local/lab/scripts/test_connections.pl
```

dpmzone-deleteproject コマンド -- プロジェクトの削除(Funclet)

dpmzone-deleteproject コマンドは、ゾーンからプロジェクトを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-deleteproject
-host hostname
-name zonename
-proj_name pname
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

プロジェクトに関連付けられているゾーンを指定します。

-proj_name *pname*

削除するプロジェクトを指定します。

例

プロジェクトを削除します。

```
dpmzone-deleteproject -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone-deletezone コマンド -- ゾーンの削除(Funclet)

dpmzone-deletezone コマンドでは、Solaris サーバ上のゾーンを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone-deletezone
```

```
-host hostname
```

```
-name zonename
```

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

削除するゾーンを指定します。

例

ホスト SolarisServer2 上のゾーンを削除します。

```
dpmzone-deletezone -host SolarisServer2 -name myzone
```

dpmzone-installzone コマンド -- ゾーンのインストール (Funclet)

dpmzone-installzone コマンドは、Solaris 10 サーバ上でインストール コマンドを起動します。インストール処理は時間がかかる場合があります。インストール処理が実行されていることを確認するには、**showprocesses** コマンドでゾーンのステータスを確認します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-installzone
-host hostname
-name zonename
[-archive_path path]
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

インストールするゾーンを指定します。

-archive_path *path*

(オプション) ブランド化ゾーンのオペレーティング環境インストーラのパスを定義します。

制限：この引数はブランド化ゾーンでは必須ですが、ネイティブゾーンでは任意です。

例

ネイティブ ゾーンをインストールします。

```
dpmzone-installzone
-host SolarisServer
-name MyZone3
```

Solaris 8 を使用して **SolarisServer** にブランド化ゾーンをインストールします。

```
dpmzone-installzone -host SolarisServer -name MyZone4
-archive_path /opt/zoneos/branded/Solaris8Installer
```

dpmzone-movezone コマンド -- ゾーンの移動 (Funclet)

dpmzone-movezone コマンドでは、あるパスから別のパスにゾーンを移動できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone-movezone  
-host hostname  
-name zonename  
-new_path path
```

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

新しい場所に移動するゾーンを指定します。

-new_path *path*

ゾーンを移動する新しいパスを定義します。

例

ホスト SolarisServer2 上のゾーンを移動します。

```
dpmzone-movezone -host SolarisServer2 -name myzone1  
-new_path /opt/zones/myzone1
```

dpmzone-rebootzone コマンド -- ゾーンの再起動 (Funclet)

dpmzone-rebootzone コマンドでは、Oracle Solaris サーバ上のゾーンを再起動できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone-rebootzone  
-host hostname  
-name zonename
```

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

再起動するゾーンを指定します。

例

ホスト SolarisServer2 上のゾーンを再起動します。

```
dpmzone-rebootzone -host SolarisServer2 -name myzone
```

dpmzone-renamezone コマンド -- ゾーンの名称変更 (Funclet)

dpmzone-renamezone コマンドでは、Oracle Solaris サーバ上のゾーンの名前を変更できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone-renamezone  
-host hostname  
-name czonename  
-new_name nzonename  
-host hostname
```

Solaris ゾーンサーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *czonename*

現在のゾーン名を指定します。

-new_name *nzonename*

新しいゾーン名を定義します。

例

ホスト SolarisServer2 上のゾーンの名前を変更します。

```
dpmzone-renamezone -host SolarisServer2 -name myzone -new_name yourzone
```

dpmzone-setpoolscheduler コマンド -- プール スケジューラの設定 (Funclet)

dpmzone-setpoolscheduler コマンドは、リソース プールに使用するスケジューラのタイプを設定します。リソース プールには、フェアシェア スケジューラ (FSS) とタイム共有スケジューラ (TS) の 2 つのスケジューラ タイプがあります。

このコマンドの形式は、以下のとおりです。

```
dpmzone-setpoolscheduler
-host hostname
-pool_name poolname
-sched_type sched
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-pool_name *poolname*

パーティション分割可能な、関連付けられているリソース グループを指定します。

-sched_type *shed*

共有に基づいた CPU 時間の割り当てに使用する、スケジューラのタイプを指定します。共有されるのは、プロジェクトに割り当てられたシステム CPU リソースの一部です。オプションは、以下のとおりです。

ts

すべてのプロセスに適正に CPU リソースを割り当て、特定のプロセスに CPU リソースを集中させないタイム共有スケジューラを指定します。**ts** は **Solaris** オペレーティング環境のデフォルト スケジューラです。

fss

共有に基づいて CPU 時間を割り当てることができる、フェアシェア スケジューラを指定します。

例

リソース プールのスケジューリングにフェア シェア スケジューラを使用するように設定します。

```
dpmzone-setpoolscheduler  
-host SolarisServer  
-pool_name test_pool  
-sched_type FSS
```

リソース プールのスケジューリングにタイム共有スケジューラを使用するように設定します。

```
dpmzone-setpoolscheduler  
-host SolarisServer  
-pool_name test_pool  
-sched_type TS
```

dpmzone-setprocessorset コマンド -- プロセッサ セットの CPU 最大数の設定 (Funclet)

dpmzone-setprocessorset コマンドは、プロセッサ セットの CPU の最大数を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-setprocessorset  
-host hostname  
-processor_set_name psetname  
-cpu_max maxcpu  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-processor_set_name *psetname*

プロセッサ セットを指定します。各プロセッサ セット (CPU グループ) には、ゼロ個以上のプロセッサを含むことができます。

-cpu_max *maxcpu*

プロセッサ セットの CPU の最大数を定義します。

例

プロセッサセットの CPU の最大数を設定します。

```
dpmzone-setprocessorset -host SolarisServer -processor_set_name test_pset -cpu_max 64
```

dpmzone-showinterfaces コマンド -- ネットワーク インターフェースの表示 (Funclet)

dpmzone-showinterfaces コマンドは、Solaris 10 ホスト上に存在するネットワーク インターフェースをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showinterfaces  
-host hostname  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

例

ホスト SolarisServer2 上のネットワーク インターフェースをすべてリスト表示します。

```
dpmzone-showinterfaces -host SolarisServer2
```

dpmzone-showpools コマンド -- リソース プールの表示 (Funclet)

dpmzone-showpools コマンドは、Solaris 10 ホスト上に存在するリソース プールをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showpools  
-host hostname  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

例

ホスト SolarisServer2 上のリソース プールをすべてリスト表示します。

```
dpmzone-showpools -host SolarisServer2
```

dpmzone-showprojects コマンド -- プロジェクトのプロパティのリスト表示 (Funclet)

dpmzone-showprojects コマンドは、プロジェクトのプロパティをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showprojects  
-host hostname  
-name zonename  
-proj_name pname  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-name zonename
```

ゾーンを指定します。

```
-proj_name pname
```

プロジェクトを指定します。

例

プロジェクトのプロパティをリスト表示します。

```
dpmzone-showprojects -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone-showresourceset コマンド -- リソース セットの表示 (Funclet)

dpmzone-showresourceset コマンドは、リソース プールのリソースをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showresourceset
```

```
-host hostname
```

```
-pool_name poolname
```

```
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-pool_name poolname
```

パーティション分割可能な、関連付けられているリソース グループを指定します。

例

MyZonesResourcePool 内のリソース セットをリスト表示します。

```
dpmzone-showresource_set -host SolarisServer2 -pool_name MyZonesResourcePool
```

dpmzone-showtasks コマンド -- プロジェクトのタスクのリスト表示 (Funclet)

dpmzone-showtasks コマンドは、プロジェクトのタスクをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showtasks
```

```
-host hostname
```

```
-name zonename
```

```
-proj_name pname
```

```
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-name zonename
```

ゾーンを指定します。

-proj_name *pname*

プロジェクトを指定します。

例

プロジェクトのタスクをリスト表示します。

```
dpmzone-showtasks -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone-showzoneinterfaces コマンド -- ゾーンのネットワーク インターフェースの表示 (Funclet)

dpmzone-showzoneinterfaces コマンドは、ゾーンのネットワーク インターフェースをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showzoneinterfaces
```

```
-host hostname
```

```
-name zonename
```

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

ゾーンを指定します。

例

SolarisServer2 上のゾーン **MyZone3** のネットワーク インターフェースをすべてリスト表示します。

```
dpmzone-showzoneinterfaces -host SolarisServer2 -name MyZone3
```

dpmzone-showzoneprocess コマンド -- ゾーンのプロセスの表示 (Funclet)

dpmzone-showzoneprocess コマンドは、ゾーン上のプロセスをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showzoneprocess
```

```
-host hostname
```

```
-name zonename
```

```
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-name zonename
```

ゾーンを指定します。

例

SolarisServer2 上のゾーン MyZone3 のプロセスをリスト表示します。

```
dpmzone-showzoneprocess -host SolarisServer2 -name MyZone3
```

dpmzone-showzoneprojects コマンド -- ゾーンのプロジェクトの表示 (Funclet)

dpmzone-showzoneprojects コマンドは、ゾーンのプロジェクトをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showzoneprojects
```

```
-host hostname
```

```
-name zonename
```

```
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-name zonename
```

ゾーンを指定します。

例

SolarisServer2 上のゾーン MyZone3 のプロジェクトをすべてリスト表示します。

```
dpmzone-showzoneprojects -host SolarisServer2 -name MyZone3
```

dpmzone-showzones コマンド -- ゾーンの表示 (Funclet)

dpmzone-showzones コマンドは、Solaris ホスト上に存在するゾーンをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone-showzones  
-host hostname  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

例

ホスト SolarisServer2 上のゾーンをすべてリスト表示します。

```
dpmzone-showzones -host SolarisServer2
```

dpmzone-startzone コマンド -- ゾーンの起動 (Funclet)

dpmzone-startzone コマンドでは、Solaris サーバ上のゾーンを起動できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone-startzone  
-host hostname  
-name zonename  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-name zonename
```

起動するゾーンを指定します。

例

ホスト SolarisServer2 上のゾーンを起動します。

```
dpmzone-startzone -host SolarisServer2 -name myzone
```

dpmzone-stopzone コマンド -- ゾーンの停止 (Funclet)

dpmzone-stopzone コマンドでは、Solaris サーバ上のゾーンを停止できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone-stopzone  
-host hostname  
-name zonename  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-name zonename
```

停止するゾーンを指定します。

例: ホスト上のゾーンを停止する

ホスト SolarisServer2 上のゾーンを停止します。

```
dpmzone-stopzone -host SolarisServer2 -name myzone
```

dpmzone-uninstallzone コマンド -- ゾーンのアンインストール (Funclet)

dpmzone-uninstallzone コマンドでは、Solaris サーバ上のゾーンをアンインストールできます。

このコマンドの形式は、以下のとおりです。

```
dpmzone-uninstallzone  
-host hostname  
-name zonename  
-host hostname
```

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

```
-name zonename
```

アンインストールするゾーンを指定します。

例

ホスト SolarisServer2 上のゾーンをアンインストールします。

```
dpmzone-uninstallzone -host SolarisServer2 -name myzone
```

展開とポリシー設定の AutoShell コマンド

このセクションでは、展開とポリシー設定の AutoShell コマンドの詳細について、アルファベット順に説明します。AutoShell コマンドラインから ISM 関数にアクセスできます。AutoShell コマンドラインプロンプトから `help ism-*` コマンドを使用すると、関数の完全なリストを表示できます。

ism-applyPolicyForService コマンド -- すべてのシステムポリシーを適用する (Funclet)

`ism-applyPolicyForService` コマンドは、管理対象サービスのすべてのコンピュータ システムにポリシーを適用します。

このコマンドの形式は、以下のとおりです。

```
ism-applyPolicyForService
-policyName policy_name
-policyType policy_type
-serviceName service_name
-templateName templateName
-templateOs templateOs
```

```
-policyName policy_name
```

適用するポリシー名を指定します。

デフォルト : Null

```
-policyType policy_type
```

ポリシー タイプを指定します。

```
-serviceName service_name
```

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として 2 つの円記号を使用します。区切り文字として 1 つの円記号を使用すると、エスケープ シーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥servicename」の代わりに「Enterprise¥¥Data Center¥¥servicename」と指定します。

-templateName *templateName*

テンプレートの名前を指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

例: すべてのシステムにポリシーを適用する

この例では、管理対象のサービス CA Virtual Assurance にポリシー SystemEDGE を適用します。

```
ism-applyPolicyForService -policyName policy5 -policyType SystemEDGE -serviceName  
"Enterprise¥¥Data Center¥¥CA Virtual Assurance Services" -templateName template2  
-templateOs windows
```

ism-applyPolicies コマンド -- ホストのグループへのポリシーの適用 (Funclet)

ism-applyPolicies コマンドは、指定されたポリシーをホストのグループに適用します。

このコマンドの形式は、以下のとおりです。

```
ism-applyPolicies  
-policyName policy_name  
-policyType policy_type  
-hostName host_name  
[-timeout time]
```

-policyName *policy_name*

適用するポリシー名を指定します。

-policyType *policy_type*

ポリシー タイプを指定します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-timeout *time*

(オプション) このコマンドでホストが正常に設定されるまで待機する時間の制限を秒数で指定し、そのあとタイムアウトが発生するようにします。指定されない場合、コマンドはホストが正常に設定されるのを待機しません。

デフォルト: 30 秒

注: 0 を指定すると、デフォルトで、システム全体のタイムアウトが発生します。

例: ホストのグループへのポリシーの適用

この例では、指定したポリシーをホストのグループに適用します。

```
ism-applyPolicies -policyName policy5 -policyType SystemEDGE -hostName "s1"-timeout 30
```

ism-applyPolicyWait コマンド -- ポリシーを適用して待機する(Funclet)

ism-applyPolicyWait コマンドは、ホストにポリシーを適用し、ポリシーが適用されるホスト名を返します。ポリシーが適用されるかタイムアウトが発生するまで、このコマンドはホスト名を返しません。

このコマンドの形式は、以下のとおりです。

```
ism-applyPolicyWait  
-policyName policy_name  
-policyType policy_type  
-hostName host_name  
[-timeout time]
```

-policyName *policy_name*

適用するポリシー名を指定します。

-policyType *policy_type*

ポリシー タイプを指定します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-timeout *time*

(オプション) このコマンドでホストが正常に設定されるまで待機する時間の制限を秒数で指定し、そのあとタイムアウトが発生するようにします。指定されない場合、コマンドはホストが正常に設定されるのを待機しません。

デフォルト：30 秒

例: ポリシーを適用して待機する

ホスト `hvserver` にポリシー `policy5` を適用します。

```
ism-applyPolicy -policyName policy5 -policyType SystemEDGE.SRM.Policies  
-hostname hvserver -timeout 400
```

ism-applyPolicyTemplates コマンド -- ホストのグループへのポリシー テンプレートの適用 (Funclet)

`ism-applyPolicyTemplates` コマンドは、指定されたポリシーをホストのグループに適用します。

このコマンドの形式は、以下のとおりです。

```
ism-applyPolicyTemplates  
-templateName templateName  
-templateOs templateOs  
-hostName hostname  
[-timeout time]  
[-replaceExisting replaceExisting]
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-timeout *time*

(オプション) このコマンドでホストが正常に設定されるまで待機する時間の制限を秒数で指定し、そのあとタイムアウトが発生するようにします。指定されない場合、コマンドはホストが正常に設定されるのを待機しません。

デフォルト: 30 秒

注: 0 を指定すると、デフォルトで、システム全体のタイムアウトが発生します。

-replaceExisting *replaceExisting*

(オプション) ホストの既存の設定を新しい設定に置換します。

例: ホストのグループへのポリシー テンプレートの適用

ホストのグループへのポリシー テンプレートの適用

```
ism-applyPolicyTemplates -templateName template2 -templateOs windows -hostName  
"s1","s2","s3"-timeout 400 -replaceExisting yes
```

ism-applyPoliciesAndTemplates コマンド -- ホストのグループへのポリシーおよびテンプレートの適用 (Funclet)

ism-applyPoliciesAndTemplates コマンドは、ポリシーとテンプレートの両方をホストのグループに適用します。

このコマンドの形式は、以下のとおりです。

```
ism-applyPoliciesAndTemplates  
-policyName policyName  
-policyType policyType  
-hostName hostname  
-templateName templateName  
-templateOs templateOs  
[-timeout time]  
[-replaceExisting replaceExisting]
```

-policyName *policy_name*

適用するポリシー名を指定します。

-policyType *policy_type*

ポリシー タイプを指定します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-templateName *templateName*

テンプレートの名前を指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

-timeout *time*

(オプション) このコマンドでホストが正常に設定されるまで待機する時間の制限を秒数で指定し、そのあとタイムアウトが発生するようにします。指定されない場合、コマンドはホストが正常に設定されるのを待機しません。

デフォルト : 30 秒

注: 0 を指定すると、デフォルトで、システム全体のタイムアウトが発生します。

-replaceExisting *replaceExisting*

(オプション) ホストの既存の設定を新しい設定に置換します。

例: ホストのグループへのポリシーおよびテンプレートの適用

ホストのグループへのポリシーおよびテンプレートの適用

```
ism-applyPoliciesAndTemplates -policyName policy6 -policyType  
SystemEDGE.SRM.Policies -hostName "s4","s5","s6" -templateName template2  
-templateOs windows -timeout 400 -replaceExisting yes
```

ism-applyPoliciesForService コマンド -- すべてのシステムにポリシーを適用する(Funclet)

ism-applyPoliciesForService コマンドは、ポリシーを管理対象サービス内のすべてのコンピュータ システムに適用します。

このコマンドの形式は、以下のとおりです。

```
ism-applyPoliciesForService  
-policyName policy_name  
-policyType policy_type  
-serviceName service_name  
[-timeout time]
```

-policyName *policy_name*

適用するポリシー名を指定します。

デフォルト : Null

-policyType *policy_type*

ポリシー タイプを指定します。

-serviceName *service_name*

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として **2** つの円記号を使用します。区切り文字として **1** つの円記号を使用すると、エスケープシーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥servicename」の代わりに「Enterprise¥¥Data Center¥¥servicename」と指定します。

-timeout *time*

(オプション) このコマンドでホストが正常に設定されるまで待機する時間の制限を秒数で指定し、そのあとタイムアウトが発生するようにします。指定されない場合、コマンドはホストが正常に設定されるのを待機しません。

デフォルト: 30 秒

注: 0 を指定すると、デフォルトで、システム全体のタイムアウトが発生します。

例: すべてのシステムにポリシーを適用する

この例では、管理対象のサービス **CA Virtual Assurance** にポリシーを適用します。

```
ism-applyPoliciesForService -policyName policy7 -policyType SystemEDGE -serviceName "Enterprise¥¥Data Center¥¥CA Virtual Assurance Services" -timeout 400
```

ism-applyPolicyTemplatesForService コマンド -- すべてのシステムにポリシー テンプレートを適用する(Funclet)

ism-applyPolicyTemplatesForService コマンドは、管理対象サービスのすべてのコンピュータ システムにポリシー テンプレートを適用します。

このコマンドの形式は、以下のとおりです。

```
ism-applyPolicyTemplatesForService
-templateName templateName
-templateOs templateOs
-serviceName serviceName
[-timeout time]
[-replaceExisting replaceExisting]
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

-serviceName *service_name*

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として 2 つの円記号を使用します。区切り文字として 1 つの円記号を使用すると、エスケープ シーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥servicename」の代わりに「Enterprise¥¥Data Center¥¥servicename」と指定します。

-timeout *time*

(オプション) このコマンドでホストが正常に設定されるまで待機する時間の制限を秒数で指定し、そのあとタイムアウトが発生するようにします。指定されない場合、コマンドはホストが正常に設定されるのを待機しません。

デフォルト: 30 秒

注: 0 を指定すると、デフォルトで、システム全体のタイムアウトが発生します。

-replaceExisting *replaceExisting*

(オプション) ホストの既存の設定を新しい設定に置換します。

例: すべてのシステムにポリシー テンプレートを適用する

この例では、管理対象のサービス CA Virtual Assurance にポリシー テンプレート SystemEDGE を適用します。

```
ism-applyPolicyTemplatesForService -templateName template3 -templateOs windows  
-serviceName "Enterprise¥¥Data Center¥¥CA Virtual Assurance Services" -timeout 200  
-replaceExisting yes
```

ism-applyPoliciesAndTemplatesForService コマンド -- すべてのシステムにポリシーおよびテンプレートを適用する(Funclet)

ism-applyPoliciesAndTemplatesForService コマンドは、ポリシーとテンプレートの両方を、管理対象サービス内のすべてのコンピュータ システムに適用します。

このコマンドの形式は、以下のとおりです。

```
ism-applyPoliciesAndTemplatesForService
-policyName policy_name
-policyType policy_type
-serviceName service_name
-templateName templateName
-templateOs templateOs
[-timeout time]
[-replaceExisting replaceExisting]
```

-policyName *policy_name*

適用するポリシー名を指定します。

デフォルト : Null

-policyType *policy_type*

ポリシー タイプを指定します。

-serviceName *service_name*

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として 2 つの円記号を使用します。区切り文字として 1 つの円記号を使用すると、エスケープ シーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥servicename」の代わりに「Enterprise¥¥Data Center¥¥servicename」と指定します。

-templateName *templateName*

テンプレートの名前を指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

-timeout *time*

(オプション) このコマンドでホストが正常に設定されるまで待機する時間の制限を秒数で指定し、そのあとタイムアウトが発生するようにします。指定されない場合、コマンドはホストが正常に設定されるのを待機しません。

デフォルト: 30 秒

注: 0 を指定すると、デフォルトで、システム全体のタイムアウトが発生します。

-replaceExisting *replaceExisting*

(オプション) ホストの既存の設定を新しい設定に置換します。

例: すべてのシステムにポリシーおよびテンプレートを適用する

この例では、管理対象のサービス **CA Virtual Assurance** にポリシーとテンプレートの両方を適用します。

```
ism-applyPoliciesAndTemplatesForService -policyName policy5 -policyType SystemEDGE  
-serviceName "Enterprise¥¥Data Center¥¥CA Virtual Assurance Services" -templateName  
template2 -templateOs windows -timeout 400 -replaceExisting yes
```

ism-cancelJob コマンド -- ジョブをキャンセルする(Funclet)

ism-cancelJob コマンドは、ジョブの保留中のタスクをキャンセルします。ジョブの UUID またはフレンドリ ID のいずれかを指定できます。

注: このコマンドはアクティブなタスクをキャンセルすることはできません。

このコマンドの形式は、以下のとおりです。

```
ism-cancelJob -jobID job_ID
```

-jobID *job_ID*

ジョブの UUID またはフレンドリ ID を指定します。

例: ジョブをキャンセルする

この例では、ジョブ **J00004** で保留中のすべてのタスクをキャンセルします。

```
ism-cancelJob -jobID J00004
```


ism-copyPolicy コマンド -- ポリシーをコピーする(Funclet)

ism-copyPolicy コマンドは、既存のポリシーからコピーしてポリシーを作成します。

このコマンドの形式は、以下のとおりです。

```
ism-copyPolicy -policyName policy_name  
-policyType policy_type  
-policyVersion [policy_version]  
-newName new_policy_name
```

-policyName *policy_name*

適用するポリシー名を指定します。

-policyType *policy_type*

ポリシー タイプを指定します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-policyVersion *policy_version*

(オプション) 適用するポリシーのバージョンを指定します。

デフォルト : Null

-newName *new_policy_name*

コピーされたポリシーの名前を定義します。

例: ポリシーをコピーする

この例では、既存のデフォルト ポリシーを使用してポリシー **test1ver1** を作成します。

```
ism-copyPolicy -policyName default -policyType SystemEDGE -policyVersion 1 -newName  
test1ver1
```

ism-createJobTemplate コマンド -- ジョブ テンプレートを作成する(Funclet)

ism-createJobTemplate コマンドは、既存のジョブからジョブ テンプレートを作成します。このコマンドは、作成されたジョブのテンプレート ID を返します。

このコマンドの形式は、以下のとおりです。

```
ism-createJobTemplate -name new_template_name  
-jobID existing_job_ID
```

-name *new_template_name*

テンプレートの名前を定義します。

-jobID *job_ID* *existing_job_ID*

既存のジョブ UUID またはフレンドリ ID を指定します。

例: ジョブ テンプレートを作成する

以下の例では、既存のジョブ ID J00004 からジョブ テンプレート Job_template_08 を作成します。

```
ism-createJobTemplate -name Job_template_08 -jobID J00004
```

ism-deleteJobTemplate コマンド -- ジョブ テンプレートを削除する(Funclet)

ism-deleteJobTemplate コマンドは、既存のジョブ テンプレートを削除します。重複した名前が指定された場合、このコマンドはエラーを返します。

このコマンドの形式は、以下のとおりです。

```
ism-deleteJobTemplate job_template_name
```

job_template_name

削除するテンプレート名を指定します。テンプレートの UUID も指定できます。

例: ジョブ テンプレートを削除する

この例では、既存のジョブ テンプレート Job_template_08 を削除します。

```
ism-deleteJobTemplate Job_template_08
```

ism-deletePolicy コマンド -- ポリシーを削除する(Funclet)

ism-deletePolicy コマンドは、既存のポリシーを削除します。ポリシーのバージョンを指定しない場合、最新のバージョンが選択されます。

このコマンドの形式は、以下のとおりです。

```
ism-deletePolicy -policyName policy_name  
-policyType policy_type [-policyVersion policy_version]
```

-policyName *policy_name*

適用するポリシー名を指定します。

-policyType *policy_type*

ポリシー タイプを指定します。

-policyVersion *policy_version*

(オプション) 適用するポリシーのバージョンを指定します。

デフォルト : Null

例: ポリシーを削除する

この例では、既存のポリシー **policy5** を削除します。

```
ism-deletePolicy -policyName policy5 -policyType SystemEDGE.SRM.Policies  
-policyVersion 3.2
```

ism-deployPackageToHost コマンド -- ホストにパッケージを展開する(Funclet)

ism-deployPackageToHost コマンドは、指定されたパッケージ ラッパーをホストに展開します。このコマンドは、展開ジョブの **UUID** を返します。

このコマンドの形式は、以下のとおりです。

```
ism-deployPackageToHost  
[-jobName job_name]  
-package package_name  
[-packageVer package_version]  
[-wrapper wrapper_name]  
-hostName host_name  
-user user_name  
-password password  
-acceptEula  
[-encrypted]  
[-redeploy]  
[-ignoreDepends]
```

-jobName *job_name*

(オプション) 展開ジョブ リストと監査証跡で使用されているジョブ名を指定します。

-package *package_name*

展開パッケージの名前または ID を指定します。

-packageVer *package_version*

(オプション) 展開パッケージのバージョンを指定します。

-wrapper *wrapper_name*

(オプション) 使用するパッケージ ラッパーの名前を指定します。
ラッパーを指定しない場合、デフォルトのラッパーが使用されます。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-user *user_name*

ログインするユーザ名を指定します。

-password *user_passsword*

ログインするユーザのパスワードを指定します。

-acceptEula

展開するすべてのパッケージのエンド ユーザ使用許諾契約 (EULA) に同意するように指示します。

-encrypted

(オプション) ユーザ名とパスワードを暗号化するように指示します。

-redeploy

(オプション) 既存パッケージを強制的に再展開するように指示します。

-ignoreDepends

(オプション) パッケージの依存関係に失われている部分があっても、強制的に展開するように指示します。

例: ホストにパッケージを展開する

この例では、ホスト `hvserver` にパッケージ `SystemEDGE` を展開します。

```
ism-deployPackageHost
-package "CA SystemEDGE"
-hostName hvserver
-user admin
-password #test#
-acceptEula
-redeploy
-ignoreDepends
```

ism-deployTemplate コマンド -- テンプレートからパッケージを展開する (Funclet)

`ism-deployTemplate` コマンドは、テンプレートに定義されたすべてのパッケージをすべてのコンピュータに展開します。このコマンドは、展開ジョブの `UUID` を返します。

このコマンドの形式は、以下のとおりです。

```
ism-deployTemplate
[-jobName job_name]
-template template_name
-user user_name
-password password
-acceptEula
[-encrypted]
[-redploy]
[-ignoreDepends]
```

-jobName *job_name*

(オプション) 展開ジョブリストと監査証跡で使用されているジョブ名を指定します。

-template *template_name*

展開のソースとして使用するジョブテンプレートの名前または `UUID` を指定します。同じ名前のテンプレートが複数ある場合は、テンプレートの `UUID` を使用します。

-user *user_name*

ログインするユーザ名を指定します。

-password *user_password*

ログインするユーザのパスワードを指定します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-acceptEula

展開するすべてのパッケージのエンド ユーザ使用許諾契約 (EULA) に同意するように指示します。

-encrypted

(オプション) ユーザ名とパスワードを暗号化するように指示します。

-redeploy

(オプション) 既存パッケージを強制的に再展開するように指示します。

-ignoreDepends

(オプション) パッケージの依存関係に失われている部分があっても、強制的に展開するように指示します。

例: テンプレートからすべてのパッケージを展開する

この例では、テンプレート **SysTemplate** に定義されたすべてのパッケージをすべてのコンピュータに展開します。

```
ism-deployTemplate
-template SysTemplate
-user admin
-password #test#
-hostName hvserver
-acceptEula
-redeploy
-ignoreDepends
```

ism-deployTemplateForService コマンド -- テンプレートのパッケージをサービスに展開する (Funclet)

ism-deployTemplate コマンドは、ジョブテンプレートに定義されたすべてのパッケージを、サービスにリスト表示されたすべてのコンピュータに展開します。このコマンドは、展開ジョブの **UUID** を返します。

このコマンドの形式は、以下のとおりです。

```
ism-deployTemplateForService
[-jobName job_name]
-template template_name
-service service_name
-user user_name
-password password
-acceptEula
[-encrypted]
[-redeploy]
[-ignoreDepends]
```

-jobName *job_name*

(オプション) 展開ジョブリストと監査証跡で使用されているジョブ名を指定します。

-template *template_name*

展開のソースとして使用するジョブテンプレートの名前または **UUID** を指定します。同じ名前のテンプレートが複数ある場合は、テンプレートの **UUID** を使用します。

-serviceName *service_name*

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として **2** つの円記号を使用します。区切り文字として **1** つの円記号を使用すると、エスケープシーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥servicename」の代わりに「Enterprise¥¥Data Center¥¥servicename」と指定します。

-user *user_name*

ログインするユーザ名を指定します。

-password *user_password*

ログインするユーザのパスワードを指定します。

-acceptEula

展開するすべてのパッケージのエンド ユーザ使用許諾契約 (EULA) に同意するように指示します。

-encrypted

(オプション) ユーザ名とパスワードを暗号化するように指示します。

-redeploy

(オプション) 既存パッケージを強制的に再展開するように指示します。

-ignoreDepends

(オプション) パッケージの依存関係に失われている部分があっても、強制的に展開するように指示します。

例: テンプレート パッケージをサービスに展開する

この例では、テンプレート `SysTemplate` に定義されたすべてのパッケージを、サービス `Enterprise¥Data Center¥MyNewService` に展開します。

```
ism-deployTemplateForService -template SysTemplate
-service Enterprise¥Data Center¥MyNewService
-user admin -password #test# -hostName hvserver
-acceptEula -redeploy -ignoreDepends
```

ism-deployTemplateToHost コマンド -- テンプレートのパッケージをホストに展開する (Funclet)

`ism-deployTemplate` コマンドは、ジョブ テンプレートに定義されたすべてのパッケージを指定されたホストに展開します。このコマンドは、展開ジョブの `UUID` を返します。

このコマンドの形式は、以下のとおりです。

```
ism-deployTemplate
[-jobName job_name]
-template template_name
-hostname host_name
-user user_name
-password password
-acceptEula
-encrypted
[-redeploy]
[-ignoreDepends]
```


-jobName *job_name*

(オプション) 展開ジョブ リストと監査証跡で使用されているジョブ名を指定します。

-template *template_name*

展開のソースとして使用するジョブ テンプレートの名前または UUID を指定します。 同じ名前のテンプレートが複数ある場合は、テンプレートの UUID を使用します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-user *user_name*

ログインするユーザ名を指定します。

-password *user_password*

ログインするユーザのパスワードを指定します。

-hostName *host_name*

ホスト コンピュータの名前を指定します。

-acceptEula

展開するすべてのパッケージのエンド ユーザ使用許諾契約 (EULA) に同意するように指示します。

-encrypted

(オプション) ユーザ名とパスワードを暗号化するように指示します。

-redeploy

(オプション) 既存パッケージを強制的に再展開するように指示します。

-ignoreDepends

(オプション) パッケージの依存関係に失われている部分があっても、強制的に展開するように指示します。

例: テンプレート パッケージをホストに展開する

この例では、テンプレート `SysTemplate` に定義されたすべてのパッケージをホスト `hvserver` に展開します。

```
ism-deployTemplate -template SysTemplate
-hostname hvserver
-user admin
-password #test#
-hostName hvserver
-acceptEula
-redeploy
-ignoreDepends
```

ism-encryptString コマンド -- 文字列を暗号化する (Funclet)

`ism-encryptString` コマンドは、指定されたテキストを暗号化し、暗号化されたテキストをコンソールに返します。

このコマンドの形式は、以下のとおりです。

```
ism-ism-encryptString encrypted_text
encrypted_text
```

暗号化するテキストを指定します。

例: 文字列を暗号化する

この例では、テキスト文字列 `passwordText` を暗号化します。

```
ism-deleteJobTemplate passwordText
```

ism-listComputerSystems コマンド -- すべてのコンピュータシステムを表示する (Funclet)

`ism-listComputerSystems` コマンドは、ホスト名、プラットフォーム、管理対象サービス、および展開されたパッケージを含むすべてのコンピュータシステムをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listComputerSystems [hostFilter]
```

hostFilter

(オプション) ホスト名を部分的に指定します。アスタリスク (*) などのワイルドカードを使用します。

デフォルト : Null

例: すべてのコンピュータシステムを表示する

以下の例では、ホスト名の最初の文字が **h** であるすべてのコンピュータシステムを表示します。

```
ism-listComputerSystems h*
```

ism-listDeployedSystems コマンド -- パッケージを展開したすべてのホストをリスト表示する (Funclet)

ism-listDeployedSystems コマンドは、特定のパッケージを展開したすべてのホスト名をリスト表示します。

注: このビューは監査証跡ビューです。そのため、後でホストが削除されたり再度イメージ化されても、そのホストはそのままこのリストに表示されます。

このコマンドの形式は、以下のとおりです。

```
ism-listDeployedSystems -package package_name  
-platform platform_name [-version package_version]
```

-package *package_name*

展開パッケージの名前または ID を指定します。

-platform *platform_name*

パッケージのオペレーティングシステムを指定します。

-version *package_version*

(オプション) パッケージのバージョンを指定します。

注: 複数のパッケージバージョンがある場合にのみ、バージョンを指定します。

例: パッケージを展開したすべてのホストをリスト表示する

この例では、パッケージ `7e033e5d-test-404d-82in-a2f7bbe960a4` を展開したホスト名をすべてリスト表示します。

```
ism-listDeployedSystems -package 7e033e5d-test-404d-82in-a2f7bbe960a4 -platform Windows
```

ism-listJobs コマンド -- すべてのジョブをリスト表示する(Funclet)

`ism-listJob` コマンドは、展開ジョブをすべてリスト表示します。ワイルドカードを使用して、ジョブステータスおよびオペレーティング環境の部分的な名前ごとにフィルタできます。

このコマンドの形式は、以下のとおりです。

```
ism-listJobs [-status job_status] [-platformFilter platform_filter]
```

-status *job_status*

(オプション) 展開ジョブのステータスを示します。アクティブ、保留、完了、または失敗のステータス名をカンマで区切ったリストによって、フィルタを指定できます。

-platformFilter *platform_filter*

(オプション) オペレーティング環境の完全な名前、またはワイルドカードを使用して部分的な名前を示します。

例: ステータスが Active であるジョブをすべてリスト表示する

この例では、ステータスが **Active** であるジョブをすべてリスト表示します。

```
ism-listJobs -status Active
```

ism-listJobStatus コマンド -- ジョブステータスをリスト表示する(Funclet)

`ism-listJobStatus` コマンドは、展開ジョブのステータスを表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listJobStatus -jobID job_ID
```

-jobID *job_ID*

ジョブの UUID またはわかりやすい ID を指定します。

例: ジョブステータスをリスト表示する

この例では、ジョブ `aa-12348-test0-9883` のステータスを表示します。

```
ism-listJobStatus -jobId aa-12348-test0-9883
```

ism-listJobTemplates コマンド -- ジョブテンプレートをすべてリスト表示する(Funclet)

`ism-listJobTemplates` コマンドは、定義済みの展開ジョブテンプレートの UUID と名前をすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listJobTemplates [-detail]
```

-detail

各ジョブテンプレートのパッケージおよびターゲットの詳細を表示します。

例: ジョブステータスをリスト表示する

この例では、展開ジョブテンプレートを表示します。

```
ism-listJobTemplates
```

ism-listPackages コマンド -- すべてのパッケージをリスト表示する(Funclet)

`ism-listPackages` コマンドは、ラッパーおよびパッケージをすべてリスト表示します。

注: 既存のラッパーを使用して参照されるパッケージは表示されません。

このコマンドの形式は、以下のとおりです。

```
ism-listJob [-platformFilter platform_filter]
```

-platformFilter *platform_filter*

(オプション) オペレーティング環境の完全な名前または部分的な名を示します。アスタリスク (*) などのワイルドカードを使用します。

例: すべてのパッケージをリスト表示する

この例では、**w** という文字で始まるオペレーティング環境名を持つラッパーおよびパッケージをすべてリスト表示します。

```
ism-listPackages -platformFilter w*.
```

ism-listPolicies コマンド -- すべてのポリシーをリスト表示する(Funclet)

ism-listPolicies コマンドは、すべてのポリシーをリスト表示します。ポリシータイプごとにポリシーをフィルタできます。

このコマンドの形式は、以下のとおりです。

```
ism-listPolicies [-policyTypeFilter policytype_filter] [-allVersions]
```

-policyTypeFilter *policytype_filter*

(オプション) ポリシータイプのフィルタを示します。

-allVersions

(オプション) すべてのポリシーのすべてのバージョンをリスト表示します。

例: すべてのポリシーをリスト表示する

この例では、すべてのポリシーのすべてのバージョンをリスト表示します。

```
ism-listPolicies -allVersions
```

ism-listPolicyStatus コマンド -- ポリシーステータスをリスト表示する(Funclet)

ism-listPolicyStatus コマンドは、すべてのホストのポリシー配布ステータスをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listPolicyStatus -policyName policy_name
```

```
-policyType policy_type [-policyVersion policy_version]
```

-policyName *policy_name*

適用するポリシー名を指定します。

-policyType *policy_type*

ポリシータイプを指定します。

-policyVersion *policy_version*

(オプション) 適用するポリシーのバージョンを指定します。

デフォルト: Null

例: ポリシー ステータスをリスト表示する

この例では、ポリシー `policy5` のポリシー配布ステータスをリスト表示します。

```
ism-listPolicyStatus -policyName policy5  
-policyType SystemEDGE.SRM.Policies -policyVersion 3.2
```

ism-listServiceSystems コマンド -- サービスのすべてのコンピュータ システムをリスト表示する (Funclet)

`ism-listServiceSystems` コマンドは、サービス内でフィルタ文字列に一致するすべてのコンピュータ システムをリスト表示します。 リストには、ホスト名、プラットフォーム、管理対象サービス、および展開されたパッケージが含まれます。

このコマンドの形式は、以下のとおりです。

```
ism-listServiceSystems service_filter
```

service_filter

完全または部分的なサービス名を指定します。 アスタリスク (*) などのワイルドカードを使用します。

例: サービスのすべてのコンピュータ システムをリスト表示する

この例では、`b` という文字で始まるすべてのサービスのすべてのコンピュータ システムをリスト表示します。

```
ism-listServiceSystems b*
```

ism-listSysEdgeConfig コマンド -- SystemEDGE の設定をリスト表示する (Funclet)

`listSysEdgeConfig` コマンドは、指定されたホスト名およびポリシー タイプのマネージャ サーバにキャッシュされた、`SystemEDGE` の設定を表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listSysEdgeConfig -hostName host_name -policyType policy_type
```

-hostName host_name

ホスト コンピュータの名前を指定します。

`-policyType policy_type`

ポリシー タイプを指定します。

例: SystemEDGE の設定をリスト表示する

この例では、SystemEDGE の設定をリスト表示します。

```
ism-listSysEdgeConfig -hostname hvserver -policyType SystemEDGE.SRM.Policies
```

ism-listSystemPoliciesForService コマンド -- サービスのすべてのポリシーをリスト表示する (Funclet)

ism-listSystemPoliciesForService コマンドは、サービスの各ホストに展開されたすべてのポリシーをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listSystemPoliciesForService -serviceName service_name
```

`-serviceName service_name`

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として 2 つの円記号を使用します。区切り文字として 1 つの円記号を使用すると、エスケープシーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥servicename」の代わりに「Enterprise¥¥Data Center¥¥servicename」と指定します。

例: サービスのすべてのポリシーをリスト表示する

この例では、サービス Enterprise¥DataCenter¥MyNewService のポリシーをすべてリスト表示します。

```
ism-listServiceSystems Enterprise¥DataCenter¥MyNewService
```

ism-listTemplateEulas コマンド -- テンプレート内の EULA をリスト表示する (Funclet)

ism-listTemplateEulas コマンドは、テンプレート内のすべての EULA (End-User License Agreements、エンドユーザ使用許諾契約) のテキストをすべて表示します。テンプレート名が複数ある場合は、テンプレートの UUID を使用します。

このコマンドの形式は、以下のとおりです。

```
ism-listTemplateEulas -template template_name
```


-template *template_name*

テンプレート名を指定します。テンプレートの UUID も指定できます。

例: テンプレート内のすべての EULA を表示する

この例では、テンプレート `test_template` 内の EULA をすべて表示します。

```
ism-listTemplateEulas -template test_template
```

ism-renamePolicy コマンド -- ポリシーの名前を変更する(Funclet)

ism-renamePolicy コマンドは、最新バージョンのポリシーの名前を変更します。

注: 前のバージョンのポリシー名は変更できません。代わりにコピーを作成します。

このコマンドの形式は、以下のとおりです。

```
ism-renamePolicy -policyName policy_name  
-policyType policy_type  
-newName new_policy_name  
-policyName policy_name
```

適用するポリシー名を指定します。

-policyType *policy_type*

ポリシー タイプを指定します。

-newName *new_policy_name*

新しいポリシー名を定義します。

例: ポリシーの名前を変更する

この例では、ポリシー `policy5` の名前を新しい名前 `NewPolicy5` に変更します。

```
ism-applyPolicy -policyName policy5  
-policyType SystemEDGE.SRM.Policies -newName NewPolicy5
```

ism-setMaintenanceMode コマンド -- ホストの保守モードを設定する(Funclet)

ism-setMaintenanceMode コマンドは、ホストの保守モードを設定します。

このコマンドの形式は、以下のとおりです。

```
ism-setMaintenanceMode -hostname host_name  
[-on {true|false}]
```

-hostname *host_name*

ホスト コンピュータの名前を指定します。

-on {true|false}

(オプション) ホストを保守モードに設定するか定義します。システムを管理対象外モードに設定するには、**false** を指定します。

デフォルト : true

例: ホストを保守モードに設定する

この例では、ホスト **hvserver** を保守モードに設定します。

```
ism-setMaintenanceMode -hostname hvserver
```

ism-listSystemPolicies コマンド -- すべてのポリシーをリスト表示する(Funclet)

ism-listSystemPolicies コマンドは、各ホストに展開されたすべてのポリシーおよびポリシー テンプレートをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listSystemPolicies -hostName hostName
```

-hostName *hostName*

ホストの名前を指定します。

例: サービスのすべてのポリシーをリスト表示する

この例では、指定されたホスト **test2** に展開されたすべてのポリシーおよびポリシー テンプレートを配信ステータスと共にリスト表示します。ポリシーおよびポリシー テンプレートが最後に適用された時間と例外（ある場合）も表示されます。

```
ism-listSystemPolicies -hostName test2
```

出力

<i>Policy</i>	<i>Status</i>	<i>OS Type</i>	<i>Ver</i>	<i>Applied</i>	<i>Exc</i>

-- test2 -----					
-- Policies-----					
default SystemEDGE.SR Delivery requested	Windows 0	2011-08-02	18:35:36	No	
default SystemEDGE	Configured	Windows 0	2011-08-08	12:54:59	No
-- Templates -----					
test3 SystemEDGE	Configured	Windows 0	2011-08-08	12:54:59	No

ism-setMaintenanceModeForService コマンド -- サービスを保守モードに設定する(Funclet)

ism-setMaintenanceModeForService コマンドは、サービスのすべてのコンピュータ システムを保守モードに設定します。

このコマンドの形式は、以下のとおりです。

```
ism-setMaintenanceModeForService
-servicename service_name
[-on {true|false}]
-serviceName service_name
```

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として 2 つの円記号を使用します。区切り文字として 1 つの円記号を使用すると、エスケープ シーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥servicename」の代わりに「Enterprise¥¥Data Center¥¥servicename」と指定します。

-on {true|false}

サービス内のすべてのコンピュータ システムを保守モードに設定するか定義します。これらのシステムを管理対象外モードに設定するには、**false** を指定します。

デフォルト : **true**

例: サービスを保守モードに設定する

この例では、サービス Enterprise¥DataCenter¥MyNewService 内のすべてのホストを保守モードに設定します。

```
ism-setMaintenanceModeForService -servicename Enterprise¥DataCenter¥MyNewService  
-on true
```

ism-setSysEdgeLogLevel コマンド -- SystemEDGE のログ レベルを設定する (Funclet)

ism-setSysEdgeLogLevel コマンドは、ホストの SystemEDGE のログ レベルを設定します。

このコマンドの形式は、以下のとおりです。

```
ism-setSysEdgeLogLevel -hostname host_name -level level  
-hostName host_name
```

ホスト コンピュータの名前を指定します。

-level *level*

指定されたレベルまでメッセージをログに記録するようにエージェントに示します。レベルは次のとおりです: **fatal**、**critical**、**warning**、**info**、**debug**、**debug1**、**debug2**、**debug3**。値が高いほど、多くの情報がログに記録されます。一番低いレベルは **fatal** で、一番高いレベルは **debug3** です。

例: SystemEDGE のログ レベルを設定する

この例では、SystemEDGE のログ レベルを **fatal** に設定します。

```
ism-setSysEdgeLogLevel -hostname hvserver -level fatal
```

ism-setSysEdgeLogLevelForService コマンド -- サービスの SystemEDGE ログレベルを設定する (Funclet)

ism-setSysEdgeLogLevelForService コマンドは、サービス内のすべてのホストの SystemEDGE ログレベルを設定します。

このコマンドの形式は、以下のとおりです。

```
ism-setSysEdgeLogLevelForService -servicename service_name -level level  
-serviceName service_name
```

管理対象サービスの名前を指定します。

注: サービス名では、区切り文字として 2 つの円記号を使用します。区切り文字として 1 つの円記号を使用すると、エスケープシーケンスとして解釈されます。たとえば、サービス名は「Enterprise¥Data Center¥serviceName」の代わりに「Enterprise¥¥Data Center¥¥serviceName」と指定します。

-level level

指定されたレベルまでメッセージをログに記録するようにエージェントに示します。レベルは次のとおりです: fatal、critical、warning、info、debug、debug1、debug2、debug3。値が高いほど、多くの情報がログに記録されます。一番低いレベルは fatal で、一番高いレベルは debug3 です。

例: サービスの SystemEDGE のログレベルを設定する

この例では、サービス Enterprise¥DataCenter¥MyNewService の SystemEDGE のログレベルを設定します。

```
ism-setSysEdgeLogLevel -servicename Enterprise¥DataCenter¥MyNewService -level info
```

ism-updateAllExceptionSystems コマンド -- 最新のポリシーを適用する (Funclet)

ism-updateAllExceptionSystems コマンドは、ポリシー例外を持つホストをすべて検索し、最新のポリシーを適用します。このコマンドが成功した場合は OK を返します。成功しなかった場合はエラーメッセージを表示します。

このコマンドの形式は、以下のとおりです。

```
ism-updateAllExceptionSystems
```

例: 最新のポリシーを適用する

この例では、ポリシー例外を持つすべてのホストを検索し、最新のポリシーを適用します。

```
ism-updateAllExceptionSystems
```

ism-updateAllOldPolicies コマンド -- 古いポリシーをすべて更新する(Funclet)

ism-updateAllOldPolicies コマンドは、期限切れのポリシー設定を持つすべてのホストを検索し、最新のポリシーを適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateAllOldPolicies
```

例: 古いポリシーをすべて更新する

この例では、古いポリシーを持つすべてのホストを検索し、最新のポリシーを適用します。

```
ism-updateAllOldPolicies
```

ism-updateAllPolicies コマンド -- すべてのポリシーを更新する(Funclet)

ism-updateAllPolicies コマンドは、すべてのコンピュータに最新のポリシーを適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateAllPolicies
```

例: すべてのポリシーを更新する

この例では、すべてのコンピュータに最新のポリシーを適用します。

```
ism-updateAllPolicies
```

ism-updateAllUpToDateSystems コマンド -- 最新のシステムをすべて更新する(Funclet)

ism-updateAllUpToDateSystems コマンドは、最新の状態であるすべてのホストに最新のポリシーを再適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateAllUpToDateSystems
```

例: 最新のシステムをすべて更新する

この例では、最新の状態であるすべてのコンピュータに最新のポリシーを再適用します。

```
ism-updateAllUpToDateSystems
```

ism-updateExceptionSystems コマンド -- 最新のポリシーを適用する(Funclet)

ism-updateExceptionSystems コマンドは、ポリシーのポリシー例外を持つホストをすべて検索し、最新の設定を適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateExceptionSystems -policyName policy_name  
-policyType policy_type
```

```
-policyName policy_name
```

適用するポリシー名を指定します。

```
-policyType policy_type
```

ポリシー タイプを指定します。

例: 最新のポリシーに更新する

この例では、デフォルト ポリシーのポリシー例外をすべて更新し、最新の設定を適用します。

```
ism-updateExceptionSystems -policyName default -policyType SystemEDGE
```

ism-updateOldSystems コマンド -- 古いポリシーを更新する(Funclet)

ism-updateOldSystems コマンドは、指定されたポリシーの期限切れの設定を持つすべてのホストを検索し、最新のポリシー設定を適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateOldSystems -policyName policy_name  
-policyType policy_type
```

```
-policyName policy_name
```

適用するポリシー名を指定します。

```
-policyType policy_type
```

ポリシー タイプを指定します。

例: 古いポリシーを更新する

この例では、デフォルト ポリシーの期限切れの設定を持つすべてのホストを更新し、最新の設定を適用します。

```
ism-updateOldSystems -policyName default -policyType SystemEDGE
```

ism-updatePolicies コマンド -- ポリシーの設定を更新する(Funclet)

ism-updateSystems コマンドは、指定されたポリシーを実行するすべてのホストに最新の設定を適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateSystems -policyName policy_name  
-policyType policy_type  
-policyName policy_name
```

適用するポリシー名を指定します。

```
-policyType policy_type
```

ポリシー タイプを指定します。

例: ポリシーの設定を更新する

この例では、デフォルト ポリシーを実行するすべてのホストに最新の設定を適用します。

```
ism-updateSystems -policyName default -policyType SystemEDGE
```

ism-updateUpToDateSystems コマンド -- ポリシーを再適用する(Funclet)

ism-updateUpToDateSystems コマンドは、指定されたポリシーの最新の設定を、最新であると考えられるすべてのホストに再適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateUpToDateSystems -policyName policy_name  
-policyType policy_type  
-policyName policy_name
```

適用するポリシー名を指定します。

```
-policyType policy_type
```

ポリシー タイプを指定します。

例: ポリシーを再適用する

この例では、デフォルト ポリシーを実行するすべてのホストに最新の設定を再適用します。

```
ism-updateUpToDateSystems -policyName default -policyType SystemEDGE
```

ism-copyPolicyTemplate コマンド -- ポリシー テンプレートをコピーする(Funclet)

ism-copyPolicyTemplate コマンドは、既存のポリシー テンプレートからコピーしてポリシー テンプレートを作成します。

このコマンドの形式は、以下のとおりです。

```
ism-copyPolicyTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs  
-newName newName
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

-newName *newName*

コピーされたポリシー テンプレートの新しい名前を定義します。

例: ポリシー テンプレートをコピーする

この例では、選択されたポリシー テンプレート **test1** の新しいコピーを作成します。 コピーは **test2** という新しい名前で作成されます。

```
ism-copyPolicyTemplate -templateName test1 -templateType SystemEDGE -templateOs  
windows -newName test2
```

ism-deletePolicyTemplate コマンド -- ポリシー テンプレートを削除する(Funclet)

ism-deletePolicyTemplate コマンドは、既存のポリシー テンプレートを削除します。

このコマンドの形式は、以下のとおりです。

```
ism-copyPolicyTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

例: ポリシー テンプレートを削除する

この例では、既存のポリシー テンプレートを削除します。

```
ism-deletePolicyTemplate -templateName test3 -templateType SystemEDGE -templateOs  
windows
```

ism-listPolicyTemplateStatus コマンド -- ポリシー テンプレート ステータスをリスト表示する(Funclet)

ism-listPolicyTemplateStatus コマンドは、指定されたポリシー テンプレートを使用して、すべてのコンピュータ システムのポリシー テンプレート配信ステータスをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
ism-copyPolicyTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

例: ポリシー テンプレート ステータスをリスト表示する

この例では、すべてのコンピュータ システムのポリシー テンプレート 配信ステータスをリスト表示します。

```
ism-listPolicyTemplateStatus -templateName Test1 -templateType SystemEDGE  
-templateOs windows
```

ism-removePolicyTemplate コマンド -- ポリシー テンプレートを削除する (Funclet)

ism-removePolicyTemplate コマンドは、指定されたホストに関連付けられたポリシーのリストからポリシー テンプレートを削除します。

このコマンドの形式は、以下のとおりです。

```
ism-removePolicyTemplate -templateName templateName  
-templateType templateType  
-hostName hostName
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-hostName *hostName*

ホスト コンピュータの名前を指定します。

例: ポリシー テンプレートを削除する

この例では、指定されたホスト名に関連付けられたポリシーのリストからポリシー テンプレートを削除します。

```
ism-removePolicyTemplate -templateName templateName -templateType templateType  
-hostName hostName
```

ism-renamePolicyTemplate コマンド -- ポリシー テンプレートの名前を変更する(Funclet)

ism-renamePolicyTemplate コマンドは、ポリシー テンプレートの名前を変更します。

このコマンドの形式は、以下のとおりです。

```
ism-renamePolicyTemplate -templateName templateName
-templateType templateType
-templateOs templateOs
-newName newName
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

-newName *newName*

名前を変更するソフトウェア ポリシーの新しい名前を指定します。

例: ポリシー テンプレートの名前を変更する

この例では、選択したポリシー テンプレートの名前を変更します。

```
ism-renamePolicyTemplate -templateName test2 -templateType SystemEDGE -templateOs
windows -newName test3
```

ism-updateExceptionSystemsTemplate コマンド -- 最新のポリシー テンプレートを適用する(Funclet)

ism-updateExceptionSystemsTemplate コマンドは、ポリシー テンプレートのポリシー例外を持つホストをすべて検索し、最新の設定を適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateExceptionSystemsTemplate -templateName templateName
-templateType templateType
-templateOs templateOs
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

例: 最新のポリシー テンプレートに更新する

この例では、ポリシー テンプレートのポリシー例外をすべて更新し、最新の設定を適用します。

```
ism-updateExceptionSystemsTemplate -templateName tpl1 -templateType SystemEDGE  
-templateOs windows
```

ism-updateOldSystemsTemplate コマンド -- 古いシステム テンプレートをすべて更新する (Funclet)

ism-updateOldSystemsTemplate コマンドは、すべてのシステム テンプレートに関して期限切れの設定を持つホストをすべて検索し、最新の設定を適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateOldSystemsTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

例: 古いシステム テンプレートをすべて更新する

この例では、すべてのシステム テンプレートに関して古い設定を持つホストをすべて検索し、最新のポリシーを適用します。

```
ism-updateOldSystemsTemplate -templateName tpl1 -templateType SystemEDGE  
-templateOs windows
```

ism-updateSystemsTemplate コマンド -- システム テンプレートを適用する (Funclet)

ism-updateSystemsTemplate コマンドは、指定されたシステム テンプレートの最新の設定をすべてのホストに適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateSystemsTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

例: システム テンプレートを適用する

この例では、システム テンプレートを実行するすべてのホストに最新の設定を適用します。

```
ism-updateSystemsTemplate -templateName tpl1 -templateType SystemEDGE -templateOs  
windows
```

ism-updateUpToDateSystemsTemplate コマンド -- システム テンプレートを再適用する (Funclet)

ism-updateUpToDateSystemsTemplate コマンドは、指定されたシステム テンプレートが最新であると見なされるすべてのホストにそのシステム テンプレートの最新の設定を再適用します。

このコマンドの形式は、以下のとおりです。

```
ism-updateUpToDateSystemsTemplate -templateName templateName  
-templateType templateType  
-templateOs templateOs
```

-templateName *templateName*

テンプレートの名前を指定します。

-templateType *templateType*

ポリシー テンプレートのタイプを指定します。

-templateOs *templateOs*

ポリシー テンプレートのオペレーティング システムを指定します。

例: システム テンプレートを再適用する

この例では、システム テンプレートを実行するすべてのホストに最新の設定を再適用します。

```
ism-updateUpToDateSystemsTemplate -templateName tpl1 -templateType SystemEDGE  
-templateOs windows
```

ism-listPolicyTemplate コマンド -- すべてのポリシー テンプレートをリスト表示する (Funclet)

ism-listPolicyTemplate コマンドは、すべてのコンピュータ システムのポリシー テンプレートをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
ism-listPolicyTemplate -policyTypeFilter policyTypeFilter
```

-policyTypeFilter *policyTypeFilter*

すべてのポリシー テンプレートを並べ替えて表示するためのフィルタ タイプを指定します。

デフォルト : Null

例: すべてのポリシー テンプレートをリスト表示する

この例では、すべてのコンピュータ システムのポリシー テンプレートをすべてリスト表示します。

```
ism-listPolicyTemplate -policyTypeFilter policyTypeFilter
```


第 3 章: コマンドライン スクリプト

CA Virtual Assurance は、コマンドプロンプト ウィンドウから実行できるプラットフォーム コマンドを提供します。これらのコマンドは、VMware vCenter Server、Solaris ゾーン、Hyper-V、MSCS などの AutoShell コマンドに対応しています。

このセクションには、以下のトピックが含まれています。

- [コマンドライン命令 \(CLI\) \(P. 449\)](#)
- [CA Cisco UCS コマンド \(P. 450\)](#)
- [CA Hyper-V CLI コマンド \(P. 499\)](#)
- [CA IBM LPAR の CLI コマンド \(P. 586\)](#)
- [CA Microsoft Cluster Server CLI コマンド \(P. 632\)](#)
- [CA Solaris ゾーン CLI コマンド \(P. 656\)](#)
- [CA VMware vCenter Server の CLI コマンド \(P. 691\)](#)
- [管理コマンドラインユーティリティ \(P. 795\)](#)
- [Common Discovery コマンド \(P. 813\)](#)
- [収集エンジン コマンド \(P. 819\)](#)
- [イベント コマンド \(P. 843\)](#)
- [汎用シェル コマンド \(P. 848\)](#)
- [ヘルプデスク コマンド \(P. 855\)](#)
- [イメージング コマンド \(P. 864\)](#)
- [ポリシー コマンド \(P. 872\)](#)
- [ポリシー設定 \(P. 891\)](#)
- [リソースマネージャ コマンド \(P. 893\)](#)
- [リモートモニタリング コマンド \(P. 953\)](#)
- [ログ ファイル \(P. 964\)](#)

コマンドライン命令 (CLI)

コマンドライン命令 (CLI) は、スクリプトの作成やタスクの自動化に利用できます。CA Virtual Assurance ユーザ インターフェイスで実行できる処理であればすべて、コマンドライン命令を使用してスクリプトを記述できます。CLI コマンドは、独自のログ ファイルを生成します。

重要: コマンドプロンプトから CLI コマンドを実行するのに十分な権限があることを確認してください。

CA Cisco UCS コマンド

dpmucs CLI コマンドを使用すると、Cisco Unified Computing System (UCS) デバイス情報、統計、サービス プロファイル、およびプールを取得し、ブレードとシャーシの操作を実行できます。

関連項目

[dpmucs associateserviceprofile コマンド -- サービス プロファイルとブレードの関連付け \(P. 452\)](#)

[dpmucs cloneserviceprofile コマンド -- サービス プロファイルのクローン作成 \(P. 453\)](#)

[dpmucs createmacpool コマンド -- MAC アドレス プールの作成 \(P. 455\)](#)

[dpmucs createorg コマンド -- 組織の作成 \(P. 457\)](#)

[dpmucs createuuidpool コマンド -- UUID プールの作成 \(P. 463\)](#)

[dpmucs createserverpool コマンド -- サーバプールの作成 \(P. 458\)](#)

[dpmucs createserviceprofile コマンド -- サービス プロファイルの作成 \(P. 460\)](#)

[dpmucs createtemplatefromserviceprofile コマンド -- サービス プロファイルからのテンプレートの作成 \(P. 462\)](#)

[dpmucs createwwpool コマンド -- WWNN プールの作成 \(P. 465\)](#)

[dpmucs deletemacpool コマンド -- MAC アドレス プールの削除 \(P. 467\)](#)

[dpmucs deleteorg コマンド -- 組織の削除 \(P. 469\)](#)

[dpmucs deleteserverpool コマンド -- サーバプールの削除 \(P. 470\)](#)

[dpmucs deleteserviceprofile -- サービス プロファイルの削除 \(P. 472\)](#)

[dpmucs deleteuuidpool コマンド -- UUID プールの削除 \(P. 473\)](#)

[dpmucs deletewwpool コマンド -- WWNN プールの削除 \(P. 475\)](#)

[dpmucs disassociateserviceprofile コマンド -- サービス プロファイルの関連付け解除 \(P. 479\)](#)

[dpmucs failoverserviceprofile コマンド -- フェールオーバーのためのサービス プロファイルとブレードの関連付け \(P. 481\)](#)

[dpmucs getblades コマンド -- ブレード情報の取得 \(P. 483\)](#)

[dpmucs getchassis コマンド -- シャーシ情報の取得 \(P. 484\)](#)

[dpmucs getserviceprofilestatus コマンド -- サービス プロファイル ステータスの取得 \(P. 487\)](#)

[dpmucs getpolicy コマンド -- UCS Manager からのポリシーの取得 \(P. 486\)](#)

[dpmucs getswitches コマンド -- スイッチ情報の取得 \(P. 489\)](#)

[dpmucs gettopsystem コマンド -- システム情報の取得 \(P. 490\)](#)

[dpmucs listserviceprofiles コマンド -- サービス プロファイルの一覧表示 \(P. 492\)](#)

[dpmucs powercycle コマンド -- ブレードの電源サイクル操作 \(P. 493\)](#)

[dpmucs renamepool コマンド -- プール名の変更 \(P. 495\)](#)

[dpmucs serviceprofileupdate コマンド -- サービス プロファイル操作の実行 \(P. 497\)](#)

dpmucs associateserviceprofile コマンド -- サービス プロファイルとブレードの関連付け

このコマンドを使用すると、サービス プロファイルを特定のブレードに関連付けることができます。このコマンドでは、関連付け前のブレードのステータスが示されます。関連付けを確認するには、`getserviceprofilestatus` コマンドを使用します。

このコマンドの形式は、以下のとおりです。

```
dpmucs associateserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagername
-service_profile_dn serviceprofiledn
-blade_dn bladenam
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-service_profile_dn *serviceprofiledn*

ブレード サービス プロファイルの完全修飾された一意の名前。OS をプロビジョニングする前にブレード サービス プロファイルを作成する必要があります。サービス プロファイルには、起動の順序、ネットワーク設定、およびストレージ (ローカルまたはリモート) が含まれます。有効な形式 : `org-root/serviceprofile`。

-blade_dn *bladename*

ブレードの完全修飾された一意の名前。ブレードが入力されない場合、関連付けのない利用可能なブレードが選択されます。有効な形式：
systemid/chassisid/bladeid。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: プロファイルとブレードの関連付け

以下の例は、プロファイルを指定のブレードに関連付けます。

```
dpmucs associateserviceprofile -ucs_manager EngManager200
-service_profile_dn org-root/ls-dev-boot-from-san-esx-2
-blade_dn sys/chassis-1/blade-1
```

dpmucs cloneserviceprofile コマンド -- サービス プロファイルのクローン作成

このコマンドを使用すると、サービス プロファイルのクローンを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs cloneserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagename
-service_profile_dn serviceprofiledn
-cloneName clonename
[-org_path org_path]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-service_profile_dn *serviceprofiledn*

ブレード サービス プロファイルの完全修飾された一意の名前。 OS をプロビジョニングする前にブレード サービス プロファイルを作成する必要があります。サービス プロファイルには、起動の順序、ネットワーク設定、およびストレージ (ローカルまたはリモート) が含まれます。有効な形式 : `org-root/serviceprofile`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: サービス プロファイルのクローン作成

以下の例は、サービス プロファイルのクローンを作成します。

```
dpmucs cloneserviceprofile -ucs_manager EngManager200
-service_profile_dn org-root/ls-dev-boot-from-san-esx-2
-cloneName org-root/ls-dev-boot-from-san-esx-2
```

dpmucs createmacpool コマンド -- MAC アドレス プールの作成

このコマンドを使用すると、1 つ以上のスロット範囲を使って **MAC** アドレス プールを作成できます。プールはどの組織レベルでも作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs createmacpool
[-sc sc_url]
-ucs_manager ucsmanagename
-pool_name poolname
-description pooldescription
-org_path organizationpath
-ranges fromrange|torange fromrange|torange...
[-ws_user username]
[-ws_password password]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

```
-ucs_manager ucsmanagename
```

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_name poolname

プール名を指定します。有効なエントリ：1～16文字の英数字。

-description pooldescription

作成されるプールを定義します。

-org_path organizationpath

組織の完全修飾パス名を定義します。有効な形式：

Org-root/org-suborg1/org-suborg2/org-suborg3...

-ranges fromrange|torange fromrange|torange...

プールのスロット範囲を定義します。プールタイプごとの有効なエントリ：XX:XX:XX:XX:XX:XXの形式のMACアドレス、XXXX-XXXXXXXXXXXXの形式のUUID、およびXX:XX:XX:XX:XX:XX:XX:XXの形式のWWNN/WWPN。範囲のエントリは|で区切ります。また、スロットの範囲は1つのスペースで区切ります。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティチェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。caaipsecurityを使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166の組み合わせ（たとえばフランス語の場合はfr_FR）で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: MAC アドレス プールの作成

以下の例は、複数のプール スロットを使って MAC アドレス プールを作成します。

```
dpmucs createmacpool
-ucs_manager ucsmanager
-pool_name TEST_MAC_POOL
-description TEST_POOL -org_path ""
-ranges 00:25:B5:00:00:C0|00:25:B5:00:00:C3 00:25:B5:00:00:C4|00:25:B5:00:00:C9
00:25:B5:00:00:CA|00:25:B5:00:00:CF
-ws_user ba
-ws_password ba
```


dpmucs createorg コマンド -- 組織の作成

このコマンドを使用すると、組織を作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs createorg [-sc sc_url]  
-ucs_manager ucsmanagername  
-parentOrg_dn parentorganization  
-org_name organizationname  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-parentOrg_dn *parentorganization*

現在のコンポーネントの親組織の完全修飾名を定義します。有効な形式 : `parentorganization/org`。

-org_name *organizationname*

組織の完全修飾名を定義します。有効な形式 : `org-root/org-name`

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 組織の作成

以下の例は、親と下位の組織を作成します。

```
dpmucs deleteorg
-ucs_manager EngManager200
-parentOrg_dn org-root
-org_name TEST_ORG
```

dpmucs createserverpool コマンド -- サーバプールの作成

このコマンドを使用すると、サーバプールを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs createserverpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-description pooldescription
-org_path organizationpath
-chassisbladepairs pairs
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_name *poolname*

プール名を指定します。有効なエントリ : 1 ~ 16 文字の英数字。

-description *pooldescription*

作成されるプールを定義します。

-org_path *organizationpath*

組織の完全修飾パス名を定義します。有効な形式 :

`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-chassisbladepairs *pairs*

サーバ プールのシャーシとブレードのペアを定義します。有効なエントリ : シャーシ名の後ろにブレード名が続き (バー (|) によって区切ります) 、ペアの間を 1 つのスペースで区切ります。例 :

`chassis1|blade1 chassis2|blade2。`

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サーバプールの作成

以下の例は、複数のシャーシブレードペアを使ってサーバプールを作成します。

```
createserverpool
-ucs_manager ucsmanager
-pool_name TEST_SERVER_POOL
-description TEST POOL
-org_path ""
-chassisbladepairs Eng1|Blade2 Test1|Blade1 QA1|Blade3
```

dpmucs createserviceprofile コマンド -- サービス プロファイルの作成

このコマンドを使用すると、デフォルトの vNIC または vHBA を作成するオプションを使用して、ハードウェア デフォルトに基づく 1 つ以上の単純なサービス プロファイルを作成できます。サービス プロファイルは、テンプレートに基づいて作成することもできます。

このコマンドの形式は、以下のとおりです。

```
dpmucs createserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagername
-service_profile_name serviceprofilename
[-org_path org_path -description description]
-policyType profiletype
[-templatename templatename]
-number number
-createnvnic createnvnic
-createnvhba createnvhba]
[-ws_user username]
[-ws_password パスワード]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-service_profile_name *serviceprofilename*

新しいサービス プロファイルの名前を指定します。

-templatename *templatename*

テンプレート名を指定します。

-number *number*

作成されたオブジェクトの数を指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: サービス プロファイルの作成

以下の例は、指定した UCS Manager のサービス プロファイルを作成します。

```
dpmucs.exe createserviceprofile  
-ucs_manager Engmanager  
-service_profile_name sptest_1
```

dpmucs createtemplatefromserviceprofile コマンド -- サービス プロファイルからのテンプレートの作成

このコマンドを使用すると、既存のサービス プロファイルからテンプレートを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs createtemplatefromserviceprofile  
[-sc sc_url]  
-ucs_manager ucsmanagename  
-service_profile_dn serviceprofiledn  
-templatename templatename  
[-templatetype templatetype -org_path org_path]  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagename*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービス プロファイルからのテンプレートの作成

以下の例は、指定した UCS Manager のサービス プロファイルからテンプレートを作成します。

```
dpmucs.exe createtemplatefromserviceprofile
-ucs_manager Engmanager
-service_profile_dn sptest_1
-templatename template_sp_1
```

dpmucs createuuidpool コマンド -- UUID プールの作成

このコマンドを使用すると、1 つ以上のスロット範囲を使ってブレードのユニバーサル固有識別子 (UUID) のプールを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs createuuidpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-description pooldescription
-org_path organizationpath
-ranges fromrange|torange fromrange|torange...
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_name *poolname*

プール名を指定します。有効なエントリ : 1 ~ 16 文字の英数字。

-description *pooldescription*

作成されるプールを定義します。

-org_path *organizationpath*

組織の完全修飾パス名を定義します。有効な形式 :

`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-ranges *fromrange|torange fromrange|torange...*

プールのスロット範囲を定義します。プールタイプごとの有効なエントリ : `XX:XX:XX:XX:XX:XX` の形式の MAC アドレス、`XXXX-XXXXXXXXXXXX` の形式の UUID、および `XX:XX:XX:XX:XX:XX:XX:XX` の形式の `WWNN/WWPN`。範囲のエントリは | で区切ります。また、スロットの範囲は 1 つのスペースで区切ります。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: UUID プールの作成

以下の例は、複数のプール スロットを使って UUID プールを作成します。

```
dpmucs createuuidpool
-ucs_manager EngManager200
-pool_name TEST_POOL_UUID
-description TEST_POOL
-org_path ""
-ranges 5500-0000000000030|5500-0000000000032 5500-0000000000035|5500-0000000000040
5500-0000000000042|5500-0000000000045
-ws_user cba
-ws_password cba
```

dpmucs createwwpool コマンド -- WWNN プールの作成

このコマンドを使用すると、ブレードの World Wide Node Name (WWNN) ポートまたはノード プールを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs createwwpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-description pooldescription
-org_path organizationpath
-ranges fromrange|torange fromrange|torange...-wwpool_type type
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_name *poolname*

プール名を指定します。有効なエントリ : 1 ~ 16 文字の英数字。

-description *pooldescription*

作成されるプールを定義します。

-org_path *organizationpath*

組織の完全修飾パス名を定義します。有効な形式 :

`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-ranges *fromrange|torange fromrange|torange...*

プールのスロット範囲を定義します。プールタイプごとの有効なエントリ : `XX:XX:XX:XX:XX:XX` の形式の MAC アドレス、`XXXX-XXXXXXXXXXXX` の形式の UUID、および `XX:XX:XX:XX:XX:XX:XX:XX` の形式の WWNN/WWPN。範囲のエントリは | で区切ります。また、スロットの範囲は 1 つのスペースで区切ります。

-wwpool_type *type*

プールタイプをノードまたはポートとして定義します。有効なエントリ : `node` または `pool`。デフォルト = `node`。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: WWNN プールの作成

以下の例は、WWNN ノード プールを作成します。

```
dpmucs createwwpool
-ucs_manager EmgManager200
-pool_name TEST_POOL_WW
-description TEST_POOL
-org_path "" -ranges 20:00:00:25:B6:01:00:F0|20:00:00:25:B6:01:00:F3
20:00:00:25:B6:01:00:F5|20:00:00:25:B6:01:00:F7 -wwpool_type node -ws_user cba
-ws_password cba
```

dpmucs deletemacpool コマンド -- MAC アドレス プールの削除

このコマンドを使用すると、MAC アドレス プールを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs deletemacpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-org_path organizationpath
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_name *poolname*

プール名を指定します。有効なエントリ : 1 ~ 16 文字の英数字。

-org_path *organizationpath*

組織の完全修飾パス名を定義します。有効な形式 :
`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: MAC アドレス プールの削除

以下の例は、MAC アドレス プールを削除します。

```
dpmucs deletemacpool
-ucs_manager ucsmanager
-pool_name TEST_POOL
-org_path ""
-ws_user cba -ws_password cba
```

dpmucs deleteorg コマンド -- 組織の削除

このコマンドを使用すると、組織を削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs deleteorg
[-sc sc_url]
-ucs_manager ucsmanagername
-org_name organizationname
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-org_name *organizationname*

組織の完全修飾名を定義します。有効な形式: `org-root/org-name`

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 組織の削除

以下の例は、組織を削除します。

```
dpmucs deleteorg
-ucs_manager EngManager200
-org_dn org-root/org-TEST_ORG
```

dpmucs deleteserverpool コマンド -- サーバプールの削除

このコマンドを使用すると、サーバプールを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs deleteserverpool
[-sc sc_url]
-ucs_manager ucsmanagername
-org_path orgpath
-pool_name poolname
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-org_path *organizationpath*

組織の完全修飾パス名を定義します。有効な形式 :

`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-pool_name *poolname*

プール名を指定します。有効なエントリ : 1 ~ 16 文字の英数字。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: サーバプールの削除

以下の例は、サーバプールを削除します。

```
dpmucs.exe deleteserverpool  
-ucs_manager ucsmanager  
-org_path1/Eng/test  
-pool_name TEST_POOL
```

dpmucs deleteserviceprofile -- サービスプロファイルの削除

このコマンドを使用すると、サービスプロファイルを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs deleteserviceprofile  
[-sc sc_url]  
-ucs_manager ucsmanagername  
-service_profile_dn serviceprofiledn  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービスコントローラホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービスコントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービスコントローラホスト用のリスニングポートを定義します。

例: `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-service_profile_dn serviceprofiledn

ブレード サービス プロファイルの完全修飾された一意の名前。OS をプロビジョニングする前にブレード サービス プロファイルを作成する必要があります。サービス プロファイルには、起動の順序、ネットワーク設定、およびストレージ（ローカルまたはリモート）が含まれます。有効な形式： *org-root/serviceprofile*。

-ws_user username -ws_password password

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 *caaipsecurity* を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は *fr_FR*）で指定します。コマンドプロンプトのロケールを使用する場合は「*native*」を指定します。

dpmucs deleteuuidpool コマンド -- UUID プールの削除

このコマンドを使用すると、UUID プールを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs deleteuuidpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-org_path organizationpath
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_name *poolname*

プール名を指定します。有効なエントリ : 1 ~ 16 文字の英数字。

-org_path *organizationpath*

組織の完全修飾パス名を定義します。有効な形式 :
`Org-root/org-suborg1/org-suborg2/org-suborg3...`

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: UUID プールの削除

以下の例は、UUID プールを削除します。

```
dpmucs deleteuuidpool
-ucs_manager EngManager200
-pool_name "TEST_POOL"
-org_path ""
-ws_user cba
-ws_password cba
```

dpmucs deletewwpool コマンド -- WWNN プールの削除

このコマンドを使用すると、WWNN プールを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs deletewwpool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_name poolname
-org_path organizationpath
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_name poolname

プール名を指定します。有効なエントリ：1 ～ 16 文字の英数字。

-org_path organizationpath

組織の完全修飾パス名を定義します。有効な形式：

Org-root/org-suborg1/org-suborg2/org-suborg3...

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: WWNN プールの削除

以下の例は、WWNN プールを削除します。

```
dpmucs deletewwpool
-ucs_manager EngManager200
-pool_name "TEST_POOL"
-org_path ""
-ws_user cba
-ws_password cba
```

dpmucs deployimage コマンド -- Cisco UCS イメージの展開

このコマンドを使用すると、RSI または ITCM を使って Cisco UCS ブレードにイメージを展開できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs deployimage
[-sc sc_url] -deploytype RSI|ITCM
ITCM のみ: -passwordEncrypted yes|no -osPassword password
-inputparam ucsManagerHost|UCS Manager
-inputparam bladeDn|bladeDN
-inputparam macAddress|macaddress
-inputparam osType|os_type
-inputparam profileDn|profileDN
-inputparam profileTemplateDn|profileTemplateDn
-inputparam selectFirstBlade|{false|true}
-inputparam serviceProfileName|serviceprofilename
-inputparam ... deploytype_options
[-ws_user username][-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-inputparam *argument*|*value*

イメージ展開の入力パラメータを指定します。RSI の引数と値のセットは引用符で囲む必要があります。

-bootServerName|*bootserver*

ITCM 展開用のブート サーバを指定します。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: RSI によるイメージの展開

以下の例は、RSI を使用して Cisco UCS ブレードにイメージを展開します。

```
dpmucs.exe deployimage -deploytype RSI
-inputparam ucsManagerHost|lodcisco1-mc1.ca.com
-inputparam ipAddress|blade-2
-inputparam macAddress|00:26:51:09:5D:E4
-inputparam osType|Microsoft_Windows-2008_R2_*-x86_64
-inputparam bladeDn|sys/chassis-1/blade-2
-inputparam serviceProfileName|blade2_netboot1
-inputparam profileDn|org-root/ls-blade2_netboot1
-inputparam selectFirstBlade|false
-inputparam profileTemplateDn|org-root/ls-template1
-inputparam RSIServer|ucs03-racemi
-inputparam RSIIImage|ATS_WIN2K8_IMG
-inputparam EnableScale|no
```

例: ITCM によるイメージの展開

以下の例は、ITCM を使用して Cisco UCS ブレードにイメージを展開します。

```
dpmucs.exe deployimage -deploytype ITCM
-passwordEncrypted no -osPassword secret123
-inputparam ucsManagerHost|lodcisco1-mc1.ca.com
-inputparam computerName|hostname1
-inputparam ipAddress|blade-2
-inputparam macAddress|00:26:51:09:5D:E4
-inputparam osType|Any
-inputparam osImage|Windows2003_Image -bootServerName|ITCM_BootServer1
-inputparam osUser|Administrator
-inputparam osPassword|password
-inputparam bladeDn|sys/chassis-1/blade-2
-inputparam serviceProfileName|blade2_netboot1
-inputparam profileDn|org-root/ls-blade2_netboot1
-inputparam selectFirstBlade|false
-inputparam templateName|Default_Software_List
-inputparam profileTemplateDn|org-root/ls-template1
```

dpmucs disassociateserviceprofile コマンド -- サービス プロファイルの関連付け解除

このコマンドを使用すると、サービス プロファイルをブレードから関連付け解除できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs diassociateserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagename
-service_profile_dn serviceprofiledn
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-service_profile_dn *serviceprofiledn*

ブレード サービス プロファイルの完全修飾された一意の名前。 OS をプロビジョニングする前にブレード サービス プロファイルを作成する必要があります。サービス プロファイルには、起動の順序、ネットワーク設定、およびストレージ (ローカルまたはリモート) が含まれます。有効な形式 : `org-root/serviceprofile`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: サービス プロファイルの関連付け解除

以下の例は、サービス プロファイルを関連付け解除します。

```
dpmucs diassociateserviceprofile
-ucs_manager EngManager200
-service_profile_dn org-root/ls-dev-boot-from-san-esx-10
```

dpmucs failoverserviceprofile コマンド -- フェールオーバーのためのサービス プロファイルとブレードの関連付け

このコマンドを使用すると、ブレード フェールオーバーの目的で、サービス プロファイルの既存の関連付けを別のプロファイルに転送できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs failoverserviceprofile
[-sc sc_url]
-ucs_manager ucsmanagername
-service_profile_dn serviceprofiledn
-blade_dn bladename
[-ws_user username]
[-ws_password password]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

```
-ucs_manager ucsmanagername
```

情報または操作を取得する場合の UCS Manager の名前を定義します。

-service_profile_dn serviceprofiledn

ブレード サービス プロファイルの完全修飾された一意の名前。OS をプロビジョニングする前にブレード サービス プロファイルを作成する必要があります。サービス プロファイルには、起動の順序、ネットワーク設定、およびストレージ（ローカルまたはリモート）が含まれます。有効な形式：*org-root/serviceprofile*。指定するプロファイルは関連付けのあるサービス プロファイルである必要があります。

-blade_dn bladename

ブレードの完全修飾された一意の名前。ブレードが入力されない場合、関連付けのない利用可能なブレードが選択されます。有効な形式：*systemid/chassisid/bladeid*。ブレードを指定しない場合（""）、利用可能なブレードがランダムに選択されます。

-ws_user username -ws_password password

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。*caaipsecurity* を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は *fr_FR*）で指定します。コマンドプロンプトのロケールを使用する場合は「*native*」を指定します。

例: フェールオーバー サービス プロファイルの作成

以下の例は、利用可能なブレードを、フェールオーバー用のサービス プロファイルにランダムに関連付けます。

```
dpmucs.exe failoverserviceprofile
-ucs_manager ucsmanager
-service_profile_dn org-root/ls-dev-boot-from-san-esx-2 -blade_dn "
```

dpmucs getblades コマンド -- ブレード情報の取得

このコマンドを使用すると、すべてのブレードまたは UCS Manager 用の特定のブレードに関する情報を取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs getblades  
[-sc sc_url]  
-ucs_manager ucsmanagename  
[-dn distinguishedname]  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagename*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-dn *distinguishedname*

Cisco UCS シャーシ、ブレード、またはスイッチの完全修飾の一意の名前を定義します。有効なエントリ : デバイス名、または空文字列 ("") (すべてのシャーシ、ブレード、またはスイッチをすべて表示する)。指定しない場合は、デバイスの詳細がリスト表示されます。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ブレード情報の取得

この例は、UCS Manager である EngManager200 用のブレード **-dn "sys/chassis-1/blade-1"** に関する情報を取得します。

```
dpmucs getblades
-ucs_manager EngManager200
-dn sys/chassis-1/blade-1
-ws_user cba
-ws_password cba
```

dpmucs getchassis コマンド -- シャーシ情報の取得

このコマンドを使用すると、すべてのシャーシまたは UCS Manager 内の特定のシャーシに関する情報を取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs getchassis
[-sc sc_url]
-ucs_manager ucsmanagername
[-dn distinguishedname]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-dn *distinguishedname*

Cisco UCS シャーシ、ブレード、またはスイッチの完全修飾の一意の名前を定義します。有効なエントリ: デバイス名、または空文字列 ("") (すべてのシャーシ、ブレード、またはスイッチをすべて表示する)。指定しない場合は、デバイスの詳細がリスト表示されます。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: シャーシ情報の取得

以下の例は、`sys/chassis-1` のシャーシ情報を取得します。

```
dpmucs getchassis
-ucs_manager EngManager200
-dn sys/chassis-1
-ws_user cba
-ws_password cba
```

dpmucs getpolicy コマンド -- UCS Manager からのポリシーの取得

このコマンドを使用すると、UCS Manager からポリシーを取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs getpolicy
[-sc sc_url]
-ucs_manager ucsmanagername
[-policyType policytype|All]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

`-sc sc_url`

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

`-ucs_manager ucsmanagername`

情報または操作を取得する場合の UCS Manager の名前を定義します。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ポリシーの取得

以下の例は、指定した UCS Manager のポリシーをすべて取得します。

```
dpmucs.exe getpolicy
-ucs_manager Engmanager
-policytype All
```

dpmucs getserviceprofilestatus コマンド -- サービス プロファイル ステータスの取得

このコマンドを使用すると、サービス プロファイルのステータスを取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs getserviceprofilestatus
[-sc sc_url]
-ucs_manager ucsmanagename
-service_profile_dn serviceprofiledn
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-service_profile_dn *serviceprofiledn*

ブレード サービス プロファイルの完全修飾された一意の名前。 OS をプロビジョニングする前にブレード サービス プロファイルを作成する必要があります。サービス プロファイルには、起動の順序、ネットワーク設定、およびストレージ (ローカルまたはリモート) が含まれます。有効な形式 : `org-root/serviceprofile`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: サービス プロファイル ステータスの取得

以下の例は、サービス プロファイル ステータスを取得します。

```
dpmucs getserviceprofilestatus
-ucs_manager ucsmanager
-service_profile_dn org-root/ls-dev-boot-from-san-esx-11
```

dpmucs getswitches コマンド -- スイッチ情報の取得

このコマンドを使用すると、すべてのスイッチまたは UCS Manager 用の特定のスイッチに関する情報を取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs getswitches
[-sc sc_url]
-ucs_manager ucsmanagename
[-dn distinguishedname]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ucs_manager *ucsmanagename*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-dn *distinguishedname*

Cisco UCS シャーシ、ブレード、またはスイッチの完全修飾の一意の名前を定義します。有効なエントリ：デバイス名、または空文字列 ("")（すべてのシャーシ、ブレード、またはスイッチをすべて表示する）。指定しない場合は、デバイスの詳細がリスト表示されます。

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: スイッチ情報の取得

以下の例は、UCS Manager である EngManager200 上のスイッチ **syst/switch-A** の情報を取得します。

```
dpmucs getswitches
-ucs_manager EngManager200
-dn syst/switch-A
-ws_user cba
-ws_password cba
```

dpmucs gettopsystem コマンド -- システム情報の取得

このコマンドを使用すると、UCS Manager 情報のトップ システムから情報を取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs gettopsystem
[-sc sc_url]
-ucs_manager ucsmanagername
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: システム情報の取得

以下の例は、UCS Manager である EngManager200 用のトップシステムを取得します。

```
dpmucs gettopsystem
-sc https://HOST/dpm/ucsws
-ucs_manager EngManager200
-ws_user cba
-ws_password cba
```

dpmucs listserviceprofiles コマンド -- サービス プロファイルの一覧表示

このコマンドを使用すると、指定した UCS Manager のサービス プロファイルを一覧表示できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs listserviceprofiles  
[-sc sc_url]  
-ucs_manager ucsmanagername  
[-ws_user username]  
[-ws_password password]  
[-locale iso639value]  
  
-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

```
-ucs_manager ucsmanagername
```

情報または操作を取得する場合の UCS Manager の名前を定義します。

```
-ws_user username -ws_password password
```

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

```
-locale iso639value
```

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: サービス プロファイルの一覧表示

以下の例は、UCS Manager である EngManager200 のサービス プロファイルを一覧表示します。

```
dpmucs listserviceprofiles
-ucs_manager EngManager200
-ws_user cba
-ws_password cba
```

dpmucs powercycle コマンド -- ブレードの電源サイクル操作

このコマンドを使用すると、ブレード上で電源サイクル操作を実行できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs powercycle
[-sc sc_url]
-ucs_manager ucsmanagername
[-dn distinguishedname]
-operation operation
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-dn *distinguishedname*

Cisco UCS シャーシ、ブレード、またはスイッチの完全修飾の一意の名前を定義します。有効なエントリ：デバイス名、または空文字列 ("")（すべてのシャーシ、ブレード、またはスイッチをすべて表示する）。指定しない場合は、デバイスの詳細がリスト表示されます。

-operation *operations*

ブレードの電源サイクル オプションを定義します。有効なエントリは次のとおりです。

- 2 = 電源サイクル。すぐにブレードの電源をオフにし、その後でブレードの電源をオンにします。
- 3 = 正常なシャットダウン。データを保存してからブレードの電力をオフにし、ブレードの電源をオンにすることができます。
- 4 = リセット即時。ブレードの電源のプラグを抜いて電源をオフにし、再びプラグを差すことができます。データは保存されません。
- 5 = リセット待機。電源のプラグを抜く前にデータを保存し、再びプラグを差すことができます。
- 6 = ソフトシャットダウン。データを保存できるように、実行中のすべてのアプリケーションにシャットダウンを通知します。
- 7 = シャットダウン。データを保存せずにブレードをシャットダウンします。
- 8 = 起動。ブレードを起動します。

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ブレードの電源サイクル操作

以下の例は、ブレード `sys/chassis-1/blade-3` に対してただちに電源サイクル操作を実行します。

```
dpmucs powercycle
-ucs_manager EngManager200
-dn "sys/chassis-1/blade-3"
-operation 2
-ws_user cba
-ws_password cba
```

dpmucs renamepool コマンド -- プール名の変更

このコマンドを使用すると、プール名を変更できます。

このコマンドの形式は、以下のとおりです。

```
dpmucs renamepool
[-sc sc_url]
-ucs_manager ucsmanagername
-pool_type pooltype
-pool_name poolname
-new_pool newpoolname
-org_path organizationpath
-description description
-ranges fromrange|torange fromrange|torange...
[-ws_user username]
[-ws_password password]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-pool_type *pooltype*

プール タイプを指定します。有効なプール タイプ（大文字と小文字を区別する）：MacPool、UuidPool、ComputePool、WWNodeNamePool、WWPortNamePool。

-pool_name *poolname*

プール名を指定します。有効なエントリ：1 ～ 16 文字の英数字。

-new_pool_name *newpoolname*

新しいプール名を定義します。有効なエントリ：1 ～ 16 文字、スペースおよび ^!@#\$\$%&*()+={}¥|;”<>,/?`~ を除く。

-org_path *organizationpath*

組織の完全修飾パス名を定義します。有効な形式：

Org-root/org-suborg1/org-suborg2/org-suborg3...

-ranges *fromrange|torange fromrange|torange...*

プールのスロット範囲を定義します。プール タイプごとの有効なエントリ：XX:XX:XX:XX:XX:XX の形式の MAC アドレス、XXXX-XXXXXXXXXXXX の形式の UUID、および XX:XX:XX:XX:XX:XX:XX:XX の形式の WWNN/WWPN。範囲のエントリは | で区切ります。また、スロットの範囲は 1 つのスペースで区切ります。

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は fr_FR）で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: プール名の変更

以下の例は、プール名を変更します。

```
dpmucs.exe renamepool
-ucs_manager ucsmanager
-pool_type UuidPool
-pool_name TEST_POOL
-new_pool_name NEW_TEST_POOL
-org_path ""
-description Windows pool
```

dpmucs serviceprofileupdate コマンド -- サービス プロファイル操作の実行

このコマンドを使用すると、次のいずれかのサービス プロファイル操作を実行できます： 関連付け、関連付け解除、フェールオーバー。

このコマンドの形式は、以下のとおりです。

```
dpmucs serviceprofileupdate
[-sc sc_url]
-ucs_manager ucsmanagername
-update_operation operation
-service_profile_dn serviceprofiledn
[-blade_dn bladedn]
[-ws_user username]
[-ws_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例： `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例： `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例： `80`

-ucs_manager *ucsmanagername*

情報または操作を取得する場合の UCS Manager の名前を定義します。

-update_operation operations

サービス プロファイルに対して実行する操作を定義します。有効なエントリ：

associate = サービス プロファイルと指定されたブレードを関連付けます。

disassociate = 指定されたブレードとサービス プロファイルの関連付けを解除します。

failover = あるサービス プロファイルから別のサービス プロファイルにフェールオーバーします。

-service_profile_dn serviceprofiledn

ブレード サービス プロファイルの完全修飾された一意の名前。OS をプロビジョニングする前にブレード サービス プロファイルを作成する必要があります。サービス プロファイルには、起動の順序、ネットワーク設定、およびストレージ（ローカルまたはリモート）が含まれます。有効な形式：*org-root/serviceprofile*。

-blade_dn bladename

ブレードの完全修飾された一意の名前。ブレードが入力されない場合、関連付けのない利用可能なブレードが選択されます。有効な形式：*systemid/chassisid/bladeid*。

-ws_user username -ws_password password

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービス プロファイルに対する操作の実行

以下の例は、指定したブレードのサービス プロファイルを関連付けます。

```
dpmucs.exe serviceprofileupdate
-ucs_manager Engmanager
-update_operation associate
-service_profile_dn org-root/ls-dev-boot-from-san-esx-1 -blade_dn
sys/chassis-1/blade-1
```

CA Hyper-V CLI コマンド

スクリプトで CLI を使用して CA Hyper-V コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。また、AutoShell では対応するコマンドも利用可能です。

dpmhyperv AddVMNic コマンド -- VM にネットワーク アダプタ コントローラを追加する

dpmhyperv AddVMNic コマンドは、VM にネットワーク アダプタを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv AddVMNic
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm VM_name|-vmguid vm_ID
[-mac mac_value]
[-switch switch_name]
[-legacy]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password password

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-mac *mac_value*

(オプション) ネットワーク アダプタの MAC (Media Access Control、メディア アクセス コントロール) アドレスを指定します。

-switch *switch_name*

(オプション) ネットワーク アダプタに接続する仮想ネットワーク スイッチの名前を指定します。このパラメータを指定しない場合、ネットワーク アダプタは接続しません。

-legacy

(オプション) Hyper-V 合成アダプタの代わりにレガシーのネットワーク アダプタを作成すべきかどうかを指定します。このオプションは、Hyper-V 統合サービスが利用できないオペレーティングシステムをインストールする場合のみに使用する必要があります。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM にネットワーク アダプタ コントローラを追加する

この例では、ネットワーク アダプタ コントローラを VM 「TestVM」に追加し、それをスイッチ「Local Area Connection - Virtual Network」に接続します。

```
dpmhyperv
-addVMNic
-ws_user dcaadmin
-ws_password #test#
-hypervHost hvserver
-vm TestVM
-switch "Local Area Connection - Virtual Network"
```

dpmhyperv AddVMSCSIController コマンド -- VM の SCSI コントローラを作成する

dpmhyperv AddVMSCSIController コマンドは、Hyper-V ホストの VM に SCSI コントローラを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv AddVMSCSIController
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password] -hypervHost hostname
-vm vm_name|-vguid vm_ID
[-pre]
[-post]
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password password

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost hostname

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストの VM に新しい SCSI コントローラを作成する

この例では、Hyper-V ホスト「**hserver**」の GUID

「**38F29B6A-8CE2-42D2-8269-BFED14644376**」を持つ VM に、新しい SCSI コントローラを作成します。

```
dpmhyperv AddVMSCSIController  
-ws_user dcaadmin  
-ws_password #test#  
-hypervHost hserver  
-vmguid 38F29B6A-8CE2-42D2-8269-BFED14644376
```

dpmhyperv AgentGet コマンド -- エージェントの値を返す

dpmhyperv AgentGet コマンドは、エージェントの値を返します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv AgentGet  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-modelPath modelpath_value attributeName attribute_name  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-jobRef *job_ref_identifier*

非同期ジョブの参照 ID を指定します。

-modelPath *modelpath_value*

値を取得するオブジェクトのモデルパスを指定します。

-attributeName *attribute_name*

取得する属性の名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: エージェントの値を返す

この例では、エージェントの値を返します。

```
dpmhyperv AgentGet
-ws_user dcaadmin
-ws_password #test#
-hypervHost hserver -modelPath
"https://localhost/aip/AOM/root/cimv2:CA_ComputerSystem.CreationClassName=¥"CA_Co
mputerSystem¥",Name=¥"8a2a30a0-a184-11df-84a8-00155d7c7405¥" -attributeName
DPM.HYPERV.SERVER.CRITICALVMS.MONITOR
```

dpmhyperv AgentSet コマンド -- エージェントの値を設定する

dpmhyperv AgentSet コマンドは、エージェントの値を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv AgentSet
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-modelPath model_path attributeName attribute_name
-snmppValue snmp_value
[-pre]
[-post]
[-locale iso639value]

-ws_user username
```

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-jobRef *job_ref_identifier*

非同期ジョブの参照 ID を指定します。

-modelPath *modelpath_value*

値を取得するオブジェクトのモデルパスを指定します。

-attributeName *attribute_name*

取得する属性の名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: エージェントの値を設定する

この例では、エージェントの値を設定します。

```
dpmhyperv AgentSet
-ws_user admin
-ws_password #test#
-hypervHost hserver
-modelPath
"https://localhost/aip/AOM/root/cimv2:CA_ComputerSystem.CreationClassName=%"CA_Co
mputerSystem%",Name=%"8a2a30a0-a184-11df-84a8-00155d7c7405%" -attributeName
DPM.HYPERV.SERVER.CRITICALVMS.MONITOR -snmpValue 1
```

dpmhyperv ChangeVMState コマンド -- VM の状態を変更する

dpmhyperv ChangeVMState コマンドは、Hyper-V ホストの VM の状態を変更します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ChangeVMState
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password] -hypervHost hostname
-vm vm_name|-vguid vm_ID
-state {Start|Suspend|Stop|Pause|Shutdown|Reboot}
-[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-state {Start|Suspend|Stop|Pause|Shutdown|Reboot}

VM の状態を指定します。 オプションは、以下のとおりです。

開始

VM をオンにします。

中断

VM を一時的に中断します。

停止

VM を停止します。

中断

VM を一時的に停止します。

シャットダウン

VM をシャットダウンします。

再起動

VM のハードリセットを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM の状態を変更する

この例では、VM「**vmtest**」の状態を有効に設定します。

```
dpmhyperv ChangeVMState -ws_user admin -ws_password #test# -hypervHost hvserver -vm  
vmtest -state start
```

dpmhyperv CreateTemplateFromVM コマンド -- VM からテンプレートを作成する

dpmhyperv CreateTemplateFromVM コマンドは、既存の VM からテンプレートを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv CreateTemplateFromVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vmname vm_name|-vmguid vm_ID
-template template_name
[-description template_description]
[-path template_location]
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-template *template_name*

新しいテンプレートの名前を定義します。

-description *template_description*

(オプション) テンプレートの説明を指定します。

-path *template_location*

(オプション) テンプレートを作成するディレクトリのフルパスを定義します。

-async

(オプション) コマンドを非同期モードで実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM からテンプレートを作成する

この例では、VM 「TestVM」 からテンプレート 「MyTemplate」 を作成します。

```
dpmhyperv -CreateTemplateFromVM -ws_user admin -ws_password #test#  
-hypervHost hvserver -vm TestVM -template MyTemplate
```

dpmhyperv CreateVirtDisk コマンド -- 仮想ディスク イメージを作成する

dpmhyperv CreateVirtDisk コマンドを使用すると、Hyper-V ホストに仮想ディスク イメージを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv CreateVirtDisk
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-path location
-diskType {dynamic|fixed}
-size drive_size
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-path *location*

仮想ディスク イメージのフルパスを定義します。

-diskType {fixed|dynamic}

仮想ディスク イメージのタイプを指定します。オプションは、以下のとおりです。

fixed

ディスクのサイズは固定されており、後から増やせないことを示します。

dynamic

ディスクのサイズは動的であり、後から必要に応じて増やせることを示します。

-size *drive_size*

ドライブのサイズを **GB** で指定します。

注: ディスクのサイズは後で変更できます。

-async

(オプション) 非同期モードでコマンドを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、**ISO 639_3166** の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 仮想ディスク イメージを作成する

この例では、**5 GB** の仮想ディスク「**01.vhd**」を作成します。

```
dpmhyperv CreateVirtDisk -hypervHost hvserver -path C:¥data¥Disks¥01.vhd -type  
dynamic -size 5 -async
```

dpmhyperv CreateVirtFloppy コマンド -- 仮想フロッピー ディスク イメージを作成する

dpmhyperv CreateVirtFloppy コマンドは、Hyper-V ホストに仮想フロッピー ディスク イメージを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv CreateVirtFloppy  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-path location  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-path *location*

仮想フロッピー イメージのフルパスを定義します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 仮想フロッピー イメージを作成する

この例では、Hyper-V ホスト「**hvserver**」に仮想フロッピー ディスク イメージを作成します。

```
dpmhyperv CreateVirtFloppy -ws_user dcaadmin -ws_password #test# -hypervHost
hvserver -path C:%data%Floppy%1.vfd
```

dpmhyperv CreateVM コマンド -- VM を作成する

dpmhyperv CreateVM コマンドは、Hyper-V ホストに VM を作成します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv CreateVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
[-templatevm vm_template]
[-templatevmguid template_ID]
[-path location]
[-memory memory_size]
[-cpuSocketCount total_CPU_socket_count]
[-cpuidLimit]
[-cpufeatlimit]
[-cpuReserve cpu_reserve]
[-cpuLimit max_CPU_resources]
[-cpuWeight cpu_weight]
[-startAction {none|auto|always}]
[-startDelay start_delay_secs]
[-stopAction {save|off|shutdown}]
[-recoveryAction {none|restart|revert}]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-templatevm *vm_template*

(オプション) 設定をコピーする VM の名前を指定します。このパラメータは、VM カタログのテンプレートではなく、Hyper-V 環境の既存の VM を識別します。

注: 共有リソースの競合が発生する場合があるため、既存の VM から VM を作成することはお勧めしません。

-templatevmguid *template_ID*

(オプション) 設定をコピーする VM の GUID を指定します。このパラメータは、VM カタログのテンプレートではなく、Hyper-V 環境の既存の VM を識別します。

注: 共有リソースの競合が発生する場合があるため、既存の VM から VM を作成することはお勧めしません。

-path *location*

(オプション) 新しい VM のフルパスを定義します。

-memory *memory_size*

(オプション) VM のメモリ サイズを定義します (MB)。

-cpuSocketCount *total_CPU_socket_count*

(オプション) VM の CPU コアの数进行を定義します。VM の CPU コアの数、Hyper-V ホストで利用可能な CPU コアの総数を超えることはできません。

-cpuidLimit

(オプション) プロセッサが CPUID の最大値を制限する必要があるかどうかを示します。

-cpufeatlimit

(オプション) VM がオペレーティング システムに対して公開された CPU 機能を制限する必要があるかどうかを示します。

-cpuReserve *cpu_reserve*

(オプション) VM が使用するために予約される CPU リソースの量を指定します。

-cpuLimit *max_CPU_resources*

(オプション) VM が消費できる最大 CPU リソース量を指定します。

-cpuWeight *cpu_weight*

(オプション) 仮想マシンの相対的な重みを 1 ～ 10000 の間で指定します。

デフォルト : 100

-startAction {none|auto|always}

(オプション) VM の起動後に実行するアクションを指定します。オプションは、以下のとおりです。

none

VM でアクションを実行しません。

auto

Hyper-V ホストがシャット ダウンされる前に VM が実行されていた場合、VM を自動的に起動します。

always

Hyper-V が起動するたびに、VM を起動します。

-startDelay *start_delay_secs*

(オプション) Hyper-V ホストが起動した後、VM を起動するための数秒の遅延を指定します。

-stopAction {save|off|shutdown}

(オプション) Hyper-V ホストが停止する前に VM で実行するアクションを指定します。オプションは、以下のとおりです。

save

VM を中断します。

off

VM をオフにします。

shutdown

VM をシャットダウンします。このオプションを使用するには、Hyper-V ツールをインストールする必要があります。

-recoveryAction {none|restart|revert}

(オプション) VM ワーク プロセスが異常終了して回復した後に、VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

アクションを実行しません。

restart

VM を再起動します。

revert

最後のスナップショットに戻します。このオプションを使用するには、VM に少なくとも 1 つのスナップショットがある必要があります。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM を作成する

この例では、「VM」フォルダに VM 「TestVM」を作成します。

```
dpmhyperv CreateVM -ws_user dcaadmin -ws_password #test# -hypervHost hvserver -name  
TestVM -path c:\VM\TestVM
```

dpmhyperv CreateVMFromTemplate コマンド -- SCVMM サーバテンプレートから VM を作成する

dpmhyperv CreateVMFromTemplate コマンドは、SCVMM サーバテンプレートから VM を作成します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv CreateTemplateFromVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
[-prompt {yes|no}] -hypervHost hostname
-vmname vm_name
-template template_name
[-dest destination]
[-computerName computer_Name]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcp]
[-ip4dhcpInt ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-adminPass admin_password]
[-productKey product_key]
[-userName user_name]
[-organization org_name]
[-domain domain_name]
[-domainAdmin domain_admin_login]
[-domainAdminPass domain_admin_password]
[-adminUser admin_username]
[-startVM start_vm]
{-memory memory_size}
[cpuSocketCount total_CPU_socket]
[-scvmmHost scvmm_host]
[-scvmmHardwareProf scvmm_hardware_profile]
[-scvmmGuestOSProf scvmm_guest_os_profile]
[-auto_deploy autodeploy]
[-async]
[-pre]
[-post]
[-locale iso639value]

-ws_user username
```

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-prompt {yes|no}

(オプション) ユーザ名とパスワードを求めるプロンプトを表示するかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-template *template_name*

新しいテンプレートの名前を定義します。

-dest *destination_path*

(オプション) テンプレートを作成するデスティネーションパスを指定します。

-computerName *computer_name*

(オプション) VM のコンピュータ名を指定します。

-ip4addr *ip4addr*

(オプション) VM インターフェースに割り当てる静的 IPv4 アドレスを指定します。

-ip4dhcp *ip4dhcp*

(オプション) DHCP を使用するかどうかを指定します。

デフォルト : local

-ip4dhcplnt *ip4dhcplnt*

(オプション) DHCP アドレスを指定します。

-ip4mask *network_mask*

(オプション) VM に割り当てるサブネットマスクを指定します。このオプションは **-ip4addr** オプションと共に使用します。

-ip4gw gateway_address

(オプション) VM のゲートウェイを設定するオプションを指定します。このオプションは **-ip4ddr** オプションと共に使用します。

-ip4metric ip_metric

(オプション) VM に対して設定するインターフェース メトリックを指定します。このオプションは **-ip4ddr** オプションと共に使用します。

-ip4dns dnsserver_ip

(オプション) VM の DNS サーバを指定します。このオプションは **-ip4ddr** オプションと共に使用します。

-adminPass admin_password

(オプション) VM のデフォルトの管理者パスワードを指定します。

-productKey product_key

(オプション) VM 用の Windows 製品アクティベーション キーを指定します。

-userName user_name

(オプション) VM の Windows ユーザ名を指定します。

-organization org_name

(オプション) VM の Windows OS の組織名を指定します。

-domain domain_name

(オプション) VM のドメイン名を指定します。

-domainAdmin domain_admin_login

(オプション) ドメイン管理者ログインを指定します。

-domainAdminPass domain_admin_password

(オプション) ドメイン管理者アカウントのパスワードを指定します。このオプションは非同期モードでは無効です。

-adminUser admin_username

(オプション) デフォルトの Administrators グループのメンバであるユーザ名を指定します。

-startVM start_vm

(オプション) VM を起動するかどうかを示します。

-memory *memory_size*

(オプション) メモリ サイズをバイト単位で定義します。

-cpuSocketCount *total_CPU_socket*

(オプション) VM 内のプロセッサ ソケットの数を指定します。

-scvmmHost *scvmm_hostname*

(オプション) SCVMM サーバホストの名前を指定します。

-scvmmHardwareProf *scvmm_hardware_profile*

(オプション) SCVMM に定義されたハードウェア プロファイルの名前を指定します。

-scvmmGuestOSProf *scvmm_guest_os_profile*

(オプション) SCVMM に定義されたゲスト オペレーティング システム プロファイルの名前を指定します。

-autodeploy *autodeploy*

(オプション) 自動的に展開することを示します。

-async

(オプション) 非同期モードでコマンドを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

注: ユーザが **-async** モードを使用していない場合、以下のパラメータは無視されます: **memory**、**cpuSocketCount**、**cpuidLimit**、**cpufeatlimit**、**cpuReserve**、**cpuLimit**、**cpuWeight**、**startAction**、**startDelay**、**stopAction**、および **recoveryAction**。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: スタンドアロン サーバのテンプレートから VM を作成する

この例では、スタンドアロンのテンプレート「Win2kSysPrepped」を使用して、VM「NewVM」を作成します。

```
dpmhyperv createVMFromTemplate -ws_user dcaadmin -ws_password #test# -hypervHost  
hvserver -vm NewVM -dest c:¥VMs¥NewBox -template Win2k3SysPrepped -ip4addr 127.0.0.1  
-ip4mask 255.255.255.0 -ip4gw 127.0.0.1 -ip4dns 127.0.0.1 -computerName NewBox
```

dpmhyperv CreateVMFromTemplate コマンド -- テンプレートから VM を作成する

dpmhyperv CreateVMFromTemplateEx コマンドは、スタンドアロンの Hyper-V Server テンプレートから VM を作成します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv CreateTemplateFromVMEx
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
[-prompt {yes|no}]
-hypervHost hostname
-vmname vm_name
-template template_name
[-dest destination]
[-computerName computer_Name]
[-ip4addr ip4addr]
[-ip4dhcp ip4dhcp]
[-ip4dhcpInt ip4dhcpInt]
[-ip4mask ip4mask]
[-ip4gw ip4gw]
[-ip4metric ip4metric]
[-ip4dns ip4dns]
[-disableAdmin]
[-adminPass admin_password]
[-autoLogon autologon]
[-duplicatorString duplicate_String]
[-timeZone timezone]
[-productKey product_key]
[-userName user_name]
[-organization org_name]
[-domain domain_name]
[-domainAdmin domain_admin_login]
[-domainAdminPass domain_admin_password]
[-workgroup workgroup_name]
[-adminUser admin_username]
[-adminUserPass admin_password] [adminGroup admin_group_name]
[-startVM start_vm]
[-custom custom]
[-memory memory_size] [cpuSocketCount total_CPU_socket]
[-cpuidLimit cpu_id_limit]
[-cpufeatlimit cpu_feat_limit]
[-cpuReserve cpu_reserve]
[-cpuLimit max_CPU_usage]
[-cpuWeight cpu_wieight]
[-startAction {none|auto|always}]
[-startDelay start_delay_secs]
[-stopAction {save|off|shutdown}]
[-recoveryAction {none,restart,revert}]
[-scvmmHost scvmm_host]
[-scvmmHardwareProf scvmm_hardware_profile]
[-scvmmGuestOSProf scvmm_guest_os_profile]
[-async]
[-pre]
```

`[-post]`

`[-locale iso639value]`

`-ws_user username`

(オプション) CA Virtual Assurance のユーザ名を指定します。

`-ws_password password`

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

`-ws_encrypted_password`

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

`-prompt {yes|no}`

(オプション) ユーザ名とパスワードを求めるプロンプトを表示するかどうかを指定します。

`-hypervHost hostname`

Hyper-V ホストの名前を指定します。

`-vm vm_name`

VM 名を指定します。

`-template template_name`

新しいテンプレートの名前を定義します。

`-dest destination_path`

(オプション) テンプレートを作成するデスティネーションパスを指定します。

`-computerName computer_name`

(オプション) コンピュータの名前を指定します。

`-ip4addr ip4addr`

(オプション) VM インターフェースに割り当てる静的 IPv4 アドレスを指定します。

`-ip4dhcp ip4dhcp`

(オプション) DHCP を使用するかどうかを指定します。

デフォルト : local

-ip4dhcplnt *ip4dhcplnt*

(オプション) DHCP アドレスを指定します。

-ip4mask *network_mask*

(オプション) VM に割り当てるサブネットマスクを指定します。このオプションは **-ip4ddr** オプションと共に使用します。

-ip4gw *gateway_address*

(オプション) VM のゲートウェイを設定するオプションを指定します。このオプションは **-ip4ddr** オプションと共に使用します。

-ip4metric *ip_metric*

(オプション) VM に対して設定するインターフェース メトリックを指定します。このオプションは **-ip4ddr** オプションと共に使用します。

-ip4dns *dnsserver_ip*

(オプション) VM の DNS サーバを指定します。このオプションは **-ip4ddr** オプションと共に使用します。

-disableAdmin

(オプション) VM のデフォルトの管理者アカウントを無効にするオプションを指定します。

-adminPass *admin_password*

(オプション) VM のデフォルトの管理者パスワードを指定します。

-autoLogon *autologon*

(オプション) Sysprep プロセスが完了した後に、デフォルトの管理者アカウントを使用して、自動的にログインするアカウント数を設定するオプションを指定します。

-duplicatorString *duplicate_string*

(オプション) VM レジストリに設定するシステム デュプリケータの名前を指定します。

-timeZone *timezone*

(オプション) テンプレートを使用して作成された VM が使用するタイムゾーンを指定します。

-productKey *product_key*

(オプション) VM 用の Windows 製品アクティベーション キーを指定します。

-userName *user_name*

(オプション) VM の Windows ユーザ名を指定します。

-organization *org_name*

(オプション) VM の Windows の組織名を指定します。

-domain *domain_name*

(オプション) VM のドメイン名を指定します。

-domainAdmin *domain_admin_login*

(オプション) ドメイン管理者ログインを指定します。

-domainAdminPass *domain_admin_password*

(オプション) ドメイン管理者アカウントのパスワードを指定します。
このオプションは非同期モードでは無効です。

-workgroup *workgroup_name*

(オプション) VM のワークグループを指定します。 このオプション
は非同期モードでは無効です。

-adminUser *admin_username*

(オプション) デフォルトの Administrators グループのメンバである
ユーザ名を指定します。

-adminUserPass *admin_password*

(オプション) デフォルトの Administrators グループのパスワードを指
定します。

-adminGroup *admin_group_name*

(オプション) 管理者のグループ名を指定します。

-startVM *start_vm*

(オプション) VM を起動するかどうかを示します。

-custom *custom*

(オプション) Sysprep プロセスの最後に実行される、カンマ区切りの
カスタム コマンドのリストを指定します。

-memory *memory_size*

(オプション) メモリ サイズをバイト単位で定義します。

-cpuSocketCount *total_CPU_socket*

(オプション) VM 内のプロセッサ ソケットの数を指定します。

-cpuidLimit *cpu_id_limit*

(オプション) プロセッサが CPUID の最大値を制限する必要があるかどうかを示します。

cpufeatlimit *cpu_feat_limit*

(オプション) VM がオペレーティングシステムに対して公開された CPU 機能を制限する必要があるかどうかを示します。

-cpuReserve *cpu_reserve*

(オプション) VM が使用するために予約される CPU リソースの量を指定します。

-cpuLimit *max_CPU_usage*

(オプション) VM が消費できる最大 CPU リソース量を指定します。

-cpuWeight *cpu_weight*

(オプション) 仮想マシンの相対的な重みを 1 ～ 10000 の間で指定します。

-startAction {none|auto|always}

(オプション) Hyper-V ホストが起動した後に VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

アクションを実行しません。

auto

Hyper-V ホストがシャットダウンされる前に VM が実行されていた場合、VM を自動的に起動します。

always

Hyper-V が起動するたびに、VM を起動します。

-startDelay *start_delay_secs*

(オプション) Hyper-V ホストが起動した後、VM を起動するための数秒の遅延を指定します。

-stopAction {save|off|shutdown}

(オプション) Hyper-V ホストがシャットダウンする前に VM で実行するアクションを指定します。オプションは、以下のとおりです。

save

VM を中断します。

off

VM をオフにします。

shutdown

VM をシャットダウンします。

-recoveryAction {none,restart,revert}

(オプション) 予期せずシャットダウンしてから Hyper-V ホストが再起動した後に VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

アクションを実行しません。

restart

VM を再起動します。

revert

最後のスナップショットに戻します。

-scvmmHost *scvmm_hostname*

(オプション) SCVMM サーバホストの名前を指定します。

-scvmmHardwareProf *scvmm_hardware_profile*

(オプション) SCVMM に定義されたハードウェア プロファイルの名前を指定します。

-scvmmGuestOSProf *scvmm_guest_os_profile*

(オプション) SCVMM に定義されたゲスト オペレーティング システム プロファイルの名前を指定します。

-async

(オプション) 非同期モードでコマンドを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

注: ユーザが **-async** モードを使用していない場合、以下のパラメータは無視されます: **memory**、**cpuSocketCount**、**cpuidLimit**、**cpufeatlimit**、**cpuReserve**、**cpuLimit**、**cpuWeight**、**startAction**、**startDelay**、**stopAction**、および **recoveryAction**。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: スタンドアロンの Hyper-V Server のテンプレートから VM を作成する

この例では、テンプレート「**Win2kSysPrepped**」を使用して、VM「**NewVM**」を作成します。

```
dpmhyperv -createVMFromTemplateEx -ws_user dcaadmin -ws_password #test# -hypervHost  
hvserver -vm NewVM -dest c:\VMs\NewBox -template Win2k3SysPrepped -ip4addr 127.0.0.1  
-ip4mask 255.255.255.0 -ip4gw 127.0.0.1 -ip4dns 127.0.0.1 -computerName NewBox
```

dpmhyperv DeleteVM コマンド -- VM を削除する

dpmhyperv DeleteVM コマンドは、Hyper-V ホストの VM を削除します。また、このコマンドは、VM に割り当てられた仮想ディスク、仮想フロッピードライブ、および ISO イメージを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv DeleteVM  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-vm vm_name|-vguid vm_ID  
[-delvhd]  
[-delvfd]  
[-deliso]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-delvhd

(オプション) 仮想ハード ドライブを削除するかどうかを示します。

-delvfd

(オプション) VM に関連付けられた仮想フロッピー ディスク イメージを削除するかどうかを示します。

-deliso

(オプション) VM に関連付けられた仮想 CD/DVD イメージを削除するかどうかを示します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストから VM を削除する

この例では、Hyper-V ホスト「hserver」から GUID 「346597794004894538F29B6A-8CE2-42」を持つ VM を削除します。また、このコマンドは、VM に割り当てられた仮想フロッピー ドライブおよび CD/DVD ISO イメージを削除します。

```
dpmhyperv DeleteVM -ws_user admin -ws_password #test# -hypervHost hserver -vmguid 38F29B6A-8CE2-42D2-8269-BFED14644376 -delvhd -delvfd -deliso
```

dpmhyperv ExpandVirtDisk コマンド -- 仮想ディスクのサイズを拡張する

dpmhyperv ExpandVirtDisk コマンドは、Hyper-V ホストの仮想ディスク イメージのサイズを拡張します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ExpandVirtDisk  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-path location  
-size disk_size  
[-async]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-path *location*

仮想ディスクの場所を定義します。

-size *drive_size*

ドライブのサイズを GB で指定します。

注: ディスクのサイズは後で変更できます。

-async

(オプション) 非同期モードでコマンドを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 仮想ディスクのサイズを拡張する

この例では、非同期モードの Hyper-V ホスト「**hserver**」の仮想ディスクイメージのサイズを 2 GB に拡張します。このコマンドは、作成されたタスクの Hyper-V ジョブ ID を返します。

```
dpmhyperv ExpandVirtDisk -ws_user admin -ws_password #test# -hypervHost hserver -path C:¥data¥Disks¥1.vhd -size 2 -async
```

dpmhyperv ExportVM コマンド -- VM をエクスポートする

dpmhyperv ExportVM コマンドは、VM の情報を持つファイルを Hyper-V ホストのディレクトリに保存します。このコマンドは廃止されるため、代わりに dpmhv ExportVMEx を使用してください。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ExportVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm vm_name|-vmguid vm_ID
-dest dest_location
[-nostate]
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-dest *dest_location*

(オプション) インポートするファイルのデスティネーション場所を定義します。

-nostate

(オプション) VM の規格のみをエクスポートすることを指定します。このオプションは、エクスポートされた VM が元の VM ディスク イメージを参照しようとするのを防ぎます。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: VM をエクスポートする

この例では、Hyper-V ホスト「`hserver`」の VM「`TestVM`」の情報を持つファイルを保存します。

```
dpmhv ExportVM -hypervHost hserver -vm TestVM -dest c:¥DATA¥Export
```

dpmhyperv ExportVMEx コマンド -- VM をエクスポートする

dpmhyperv ExportVM コマンドを使用すると、仮想マシンをエクスポートできます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ExportVM
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm vm_name|-vmguid vm_ID
-dest dest_folder
[-noruntimeInfo]
[-nosnapshots]
[-nostorage]
[-nosubdir]
[-snapshot snapshot_name][[-snapshotid snapshot_ID]
[-nostate]
[-async]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-dest *dest_location*

(オプション) インポートするファイルのデスティネーション場所を定義します。

-noruntimeInfo

(オプション) ランタイム関連情報をエクスポートしないように指定します。

-nosnapshots

(オプション) スナップショットをエクスポートしないように指定します。

-nostorage

(オプション) 仮想ディスク イメージファイルをエクスポートしないように指定します。

-nosubdir

(オプション) エクスポートパスとして指定されたディレクトリに、**VM** 表示名を持つサブ ディレクトリを作成しないように指定します。

-snapshot *snapshot_name*

(オプション) エクスポートするスナップショットの名前を指定します。

-snapshot *snapshot_ID*

(オプション) エクスポートするスナップショットの **GUID** を指定します。

-nostate

(オプション) **VM** の規格のみをエクスポートすることを指定します。このオプションは、エクスポートされた **VM** が元の **VM** ディスク イメージを参照しようとするのを防ぎます。

-async

(オプション) 非同期モードでコマンドを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM をエクスポートする

この例では、スナップショットなしで VM 「TestVM」 をエクスポートし、エクスポートされた VM 「textvmExp」 を作成します。

```
dpmhyperv ExportVMEx -ws_user admin -ws_password #test# -hypervHost hserver -vm TestVM -dest c:¥Ex¥TestVMExp -nosubdir -nosnapshots
```

dpmhyperv GetHostSwitches コマンド -- ホストのスイッチを取得する

dpmhyperv GetHostSwitches コマンドは、Hyper-V ホストのスイッチを返します。フィルタ条件を指定して特定のスイッチを取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetHostSwitches  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname [filter filter_condition]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-jobRef *job_ref_identifier*

非同期ジョブの参照 ID を指定します。

-filter *condition*

(オプション) 「*」などのフィルタ条件を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ホストのスイッチを取得する

この例では、ホスト「**hvserver**」のスイッチを返します。

```
dpmhyperv GetHostSwitches -ws_user admin -ws_password #test# -hypervHost hvserver
```

例: フィルタを使用してホストのスイッチを取得する

この例では、スイッチ名に「**n**」を含むスイッチをすべて返します。

```
dpmhyperv GetHostSwitches -ws_user admin -ws_password #test# -hypervHost hvserver  
-filter *n*
```

dpmhyperv GetJobErrorInfo コマンド -- ジョブのエラー情報を取得する

dpmhyperv GetJobErrorInfo コマンドは、Hyper-V ホストで実行されている非同期ジョブのエラーの詳細を返します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetJobErrorInfo  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-jobRef job_ref_identifier  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-jobRef *job_ref_identifier*

非同期ジョブの参照 ID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ジョブのエラー情報を取得する

この例では、ジョブ、

「¥¥hserver¥root¥virtualization:Msvm_ConcreteJob.InstanceID="86C9700348-4117-ABA8-445456"」のエラー情報を返します。

```
dpmhyperv GetJobErrorInfo -ws_user admin -ws_password admin -hypervHost hvserver
-jobRef
¥¥hserver¥root¥virtualization:Msvm_ConcreteJob.InstanceID="86C9700348-4117-ABA8-4
45456"
```

dpmhyperv GetJobInfo コマンド -- ジョブの情報を取得する

dpmhyperv GetJobInfo コマンドは、Hyper-V ホストの非同期ジョブの詳細を返します。詳細には、完了した割合、エラー情報、および開始時刻が含まれます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetJobInfo
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-jobRef job_ref_identifier
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-jobRef *job_ref_identifier*

非同期ジョブの参照 ID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ジョブの情報を取得する

この例では、Hyper-V ジョブ

「¥¥hserver¥root¥virtualization:Msvm_ConcreteJob.InstanceID="86TSTC4-ABA8-C82734036」に関する情報を返します。

```
dpmhyperv GetJobInfo -ws_user admin -ws_password #test# -hypervHost hvserver -jobRef
¥¥hserver¥root¥virtualization:Msvm_ConcreteJob.InstanceID="86TSTC4-ABA8-C82734036
"
```

dpmhyperv GetJobState コマンド -- ジョブの状態を取得する

dpmhyperv GetJobState コマンドは、Hyper-V ホストで実行されている非同期ジョブの状態を返します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetJobState  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-jobRef job_ref-identifier  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-jobRef *job_ref_identifier*

非同期ジョブの参照 ID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ジョブの状態を取得する

この例では、ジョブ

「¥¥hserver¥root¥virtualization:Msvm_ConcreteJob.InstanceID="86CD04TEST82795D6F036"」の状態を返します。

```
dpmhyperv GetJobState -ws_user admin -ws_password #test# -hypervHost hserver -jobRef ¥¥hserver¥root¥virtualization:Msvm_ConcreteJob.InstanceID="86CD04TEST82795D6F036"
```

dpmhyperv GetServerInfo コマンド -- すべての管理対象サーバを取得する

dpmhyperv GetServerInfo コマンドは、CA Virtual Assurance システムで、プロパティを持つ Hyper-V ホストをすべて返します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetServerInfo  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password password

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

この例では、CA Virtual Assurance システムに登録された各 Hyper-V ホストのプロパティを返します。

```
dpmhyperv GetServerInfo -ws_user admin -ws_password #test#
```

dpmhyperv GetVersion コマンド -- PMM のバージョンを取得する

dpmhyperv GetVersion コマンドは、PMM (Platform Management Module、プラットフォーム管理モジュール) のバージョンを返します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetVersion
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: PMM のバージョンを取得する

この例では、現在の Hyper-V PMM のバージョンを表示します。

```
dpmhyperv GetVersion -ws_user admin -ws_password #test#
```

dpmhyperv GetVMProperties コマンド -- VM のプロパティを取得する

dpmhyperv GetVMProperties コマンドを使用すると、VM のプロパティを表示できます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetVMProperties  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname|-vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password password

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM のプロパティを取得する

この例では、ホスト「hserver」の VM「vm-01」のプロパティを表示します。

```
dpmhyperv GetVMProperties -ws_user admin -ws_password #test# -hypervHost hserver -vm vm01
```

dpmhyperv GetVMState コマンド -- VM 状態を取得する

dpmhyperv GetVMState コマンドは、VM の現在の状態を返します。以下のリストでは 4 つの状態について説明します。

- disabled : VM はオフになっています
- enabled : VM は稼働中です
- paused : VM は一時停止中です
- suspended : VM は停止しています

このコマンドの形式は、以下のとおりです。

```
GetVMState  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname| -vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: VM 名を使用して VM 状態を取得する

この例では、ホスト「`hserver`」の VM「`vm01`」の状態を返します。

```
dpmhyperv GetVMState -ws_user admin -ws_password #test# -hypervHost hserver -vm test-vm01
```

例: VM の GUID を使用して VM 状態を取得する

この例では、ホスト「`hserver`」で GUID「`38FTESTED14644376`」を持つ仮想マシンの状態を返します。

```
dpmhyperv GetVMState -ws_user admin -ws_password #test# -hypervHost hserver -vmguid 38FTESTED14644376
```

dpmhyperv ImportVM コマンド -- VM をインポートする

`dpmhyperv ImportVM` コマンドを使用すると、Hyper-V `hypervHost` に VM をインポートできます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ImportVM  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-path location  
[-name vm_name]  
[-noNewID]  
[-async]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-path *location*

インポートされる VM の場所を定義します。

-vm *vm_name*

VM 名を指定します。

-noNewID

(オプション) 新しい VM GUID を生成しないように指示します。この場合、インポート元の VM とインポート先の VM は同じ GUID になります。他の VM との競合を避けるために、このオプションは使用しないことをお勧めします。

-async

(オプション) 非同期モードでコマンドを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストに VM をインポートする

この例では、Hyper-V ホスト「hserver」に VM 「test-vm01-imported」をインポートします。このコマンドでは新しい ID を生成しません。

```
dpmhyperv ImportVM -ws_user admin -ws_password #test# -hypervHost hserver -path C:¥data¥Export¥test-vm01 -name test-vm01-imported -noNewID.
```

dpmhyperv ImportVMEx コマンド -- VM をコピーおよびインポートする

dpmhyperv ImportVMEx コマンドは、すでに Hyper-V ホストにエクスポートされた VM をコピーおよびインポートします。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ImportVMEx  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-path location [dest dest_location]  
[-name vm_name]  
[-noNewID]  
[-async]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-path *location*

サーバの場所を定義します。

-dest *dest_location*

(オプション) インポートするファイルのデスティネーション場所を定義します。

-vm *vm_name*

VM 名を指定します。

-noNewID

(オプション) 新しい VM GUID を生成しないように指示します。この場合、インポート元の VM とインポート先の VM は同じ GUID になります。他の VM との競合を避けるために、このオプションは使用しないことをお勧めします。

-async

(オプション) 非同期モードでコマンドを実行します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: Hyper-V ホストに VM をコピーおよびインポートする

この例では、エクスポートされた VM を Hyper-V ホストにインポートします。その VM 用の新しい ID は生成しません。

```
dpmhyperv ImportVMEx -ws_user admin -ws_password #test# -hypervHost hserver -path C:%data%Export%test -name test-vm01-imported -noNewID
```


dpmhyperv SetClustered コマンド -- クラスタに対して VM を追加または削除する

dpmhyperv SetClustered コマンドは、Hyper-V クラスタに対して VM を追加または削除します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv SetClustered  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm VM_name|-vmguid vm_ID  
-clustered {on|off}  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-clustered {on|off}

VM をクラスタに追加するのか、クラスタから削除するのかを指定します。

on

VM をクラスタに追加することを指定します。

off

VM をクラスタから削除することを指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V クラスタに VM を追加する

この例では、GUID「38F29B6A-TEST-BFED14644376」を持つ VM を Hyper-V クラスタに追加します。

```
dpmhyperv SetClustered -ws_user admin -ws_password #test# -hypervHost hserver -vmguid 38F29B6A-TEST-BFED14644376 -clustered on
```

例: Hyper-V クラスタから VM を削除する

この例では、Hyper-V クラスタから VM「test-vm01」を削除します。

```
dpmhyperv SetClustered -ws_user admin -ws_password #test# -hypervHost hserver -vm test-vm01 -clustered off
```

dpmhyperv SetSysPrepProperties コマンド -- Sysprep プロパティを設定する

dpmhyperv SetSysPrepProperties コマンドを使用すると、Hyper-V Server で sysprep プロパティを設定できます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv SetSysPrepProperties
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm VM_name|-vmguid vm_ID
[-jobRef job_reference]
[-computerName computer_name]
[-ip4addr ip4_address]
[-ip4dhcp]
[-ip4dhcpint ip4dhcpint]
[-ip4mask network_mask]
[-ip4gw gateway_address]
[-ip4metric ip_metric]
[-ip4dns dnsserver_ip]
[-disableAdmin]
[-adminPass administrator_password]
[-autoLogon autologon]
[-duplicatorString duplicator_string]
[-timezone timezone]
[-productKey product_key]
[-userName user_name]
[-organization org_name]
[-domain domain_name]
[-domainAdmin domain_admin_login]
[-domainAdminPass domain_admin_password]
[-workgroup workgroup_name]
[-adminUser admin_username]
[-adminUserPass admin_password]
[-adminGroup admin_groupname]
[-custom custom]
[-pre]
[-post]
[-locale iso639value]

-ws_user username
```

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-jobRef *job_reference*

ジョブ ID を指定します。

-computerName *computer_name*

(オプション) コンピュータの名前を指定します。

-ip4addr *ip4_address*

(オプション) VM インターフェースに割り当てる静的 IPv4 アドレスを指定します。

-ip4dhcp *ip4dhcplnt*

(オプション) DHCP アドレスを指定します。

-ip4mask *network_mask*

(オプション) VM に割り当てるサブネットマスクを指定します。このオプションは **-ip4addr** オプションと共に使用します。

-ip4gw *gateway_address*

(オプション) VM のゲートウェイを設定するオプションを指定します。このオプションは **-ip4addr** オプションと共に使用します。

-ip4metric *ip_metric*

(オプション) VM に対して設定するインターフェース メトリックを指定します。このオプションは **-ip4addr** オプションと共に使用します。

-ip4dns dnsserver_ip

(オプション) VM の DNS サーバを指定します。このオプションは **-ip4addr** オプションと共に使用します。

-disableAdmin

(オプション) VM のデフォルトの管理者アカウントを無効にするオプションを指定します。このパラメータをサポートするには、Sysprep ツールを使って作成した Windows イメージが必要です。このオプションは非同期モードでは無効です。

-adminPass administrator_password

(オプション) VM のデフォルトの管理者パスワードを指定します。

-autoLogon autologon

(オプション) Sysprep プロセスが完了した後に、デフォルトの管理者アカウントを使用して、自動的にログインするアカウント数を設定するオプションを指定します。

-duplicatorString duplicate_string

(オプション) VM レジストリに設定するシステム デュプリケータの名前を指定します。

-timeZone timezone

(オプション) テンプレートを使用して作成された VM が使用するタイムゾーンを指定します。

デフォルト :

-1

-productKey product_key

(オプション) VM 用の Windows 製品アクティベーション キーを指定します。

-userName User_name

(オプション) VM の Windows ユーザ名を指定します。

-organization org_name

(オプション) VM の Windows ユーザ名を指定します。このオプションは非同期モードでは無効です。

-domain domain_name

(オプション) VM のドメイン名を指定します。

-domainAdmin *domain_admin_login*

(オプション) ドメイン管理者ログインを指定します。

-domainAdminPass *domain_admin_password*

(オプション) ドメイン管理者アカウントのパスワードを指定します。
このオプションは非同期モードでは無効です。

-workgroup *workgroup_name*

(オプション) VM のワークグループを指定します。このオプションは非同期モードでは無効です。

-adminUser *admin_username*

(オプション) デフォルトの Administrators グループのメンバであるユーザ名を指定します。

-adminUserPass *admin_password*

(オプション) デフォルトの Administrators グループのパスワードを指定します。

-custom *custom*

(オプション) Sysprep プロセスの最後に実行される、カンマ区切りのカスタム コマンドのリストを指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: Hyper-V Server で Sysprep プロパティを設定する

この例では、ホスト「hserver」で sysprep プロパティを設定します。

```
dpmhyperv SetSysPrepProperties -hypervHost hserver -vm VM_05 -ip4addr 130.119.0.40  
-ip4mask 255.255.0.0 -ip4gw 130.119.0.100
```

dpmhyperv SetVMDisk コマンド -- ドライブ コントローラにドライブまたはディスクを割り当てる

SetVMDisk コマンドを使用すると、ドライブ コントローラに仮想ドライブまたはディスクを割り当てることができます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv SetVMDisk
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password] -hypervHost hostname
-vm vm_name|-vmguid vm_ID
-drive {HDD|DVD} -driveType {scsi|ide}
[-path location]
-id drive_ID_number
[-lun logical_unit_num]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-drive {HDD|DVD}

イメージを割り当てるドライブのタイプを指定します。オプションは、以下のとおりです。

HDD

ドライブタイプがハードドライブであることを示します。

DVD

ドライブタイプが **CD/DVD** ドライブであることを示します。

-driveType {scsi|ide}

イメージを割り当てるドライブコントローラのタイプを指定します。オプションは、以下のとおりです。

scsi

ドライブコントローラが **SCSI** であることを示します。

ide

ドライブコントローラが **IDE** であることを示します。**CD/DVD** イメージの場合は、コントローラタイプが **IDE** である必要があります。

-path *location*

(オプション) 仮想ディスクまたはドライブの場所を定義します。

-id *drive_ID_number*

ドライブの ID 番号を指定します。IDE には **0**、SCSI ドライブには **0**、**1**、または **2** を指定します。

-lun *logical_unit_num*

(オプション) **SCSI** ドライブの **LUN** (Logical Unit Number、論理ユニット番号) を指定します。IDE コントローラの場合、LUN は **0** または **1** である必要があります。また、**SCSI** コントローラの場合、LUN は **0** ～ **63** である必要があります。番号が指定されない場合、利用可能なチャンネルは自動的に割り当てられます。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ドライブ コントローラにドライブを割り当てる

この例では、1 番目の SCSI コントローラの 1 番目のチャンネルに仮想ハードドライブ イメージ ファイル「**disk.vhd**」を割り当てます。

```
dpmhv-setVMDisk -hypervHost hvserver -vm TestVM -type scsi -id 0 -lun 0 -drive HDD
-path c:¥VHDs¥disk.vhd
```

dpmhyperv SetVMProperties コマンド -- VM のプロパティを設定する

dpmhyperv SetVMProperties コマンドを使用すると、VM のプロパティを変更できます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv SetVMProperties
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm vm_name|-vguid vm_ID
[-memory memory_size]
[-cpuSocketCount total_CPU_socket_count]
[-cpuidLimit CPU_ID_Limit]
[-cpufeatlimit CPU_Feature_Limit]
[-cpuReserve cpu_reserver]
[-cpuReserve cpu_reserve]
[-cpuLimit max_CPU_usage]
[-cpuWeight cpu_weight]
[-startAction {none|auto|always}]
[-startDelay start_delay_secs]
[-stopAction {save|off|shutdown}]
[-recoveryAction {none|restart|revert}]
[-pre]
[-post]
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-memory *memory_size*

(オプション) VM のメモリ サイズを定義します (MB)。

-cpuSocketCount *total_CPU_socket_count*

(オプション) VM の CPU コアの数进行定義します。CPU コアは、Hyper-V ホストで利用可能な CPU コアの総数を超えることはできません。

-cpuidLimit *CPU_ID_Limit*

(オプション) Windows NT などのレガシー オペレーティングシステムとの互換性を確保するために、VM の CPU ID を小さくする必要があるかどうかを示します。

cpufeatlimit *CPU_Feature_Limit*

(オプション) VM がオペレーティングシステムに対して公開された CPU 機能を制限する必要があるかどうかを示します。

-cpuReserve *cpu_reserve*

(オプション) VM が使用するために予約される CPU リソースの量を指定します。

-cpuLimit *max_CPU_usage*

(オプション) VM が消費できる最大 CPU リソース量を指定します。

-cpuWeight *cpu_weight*

(オプション) 仮想マシンの相対的な重みを 1 ～ 10000 の間で指定します。

デフォルト : 100

-startAction {none|auto|always}

(オプション) 起動時のアクションを実行する前に待機する時間間隔を指定します。オプションは、以下のとおりです。

none

アクションを実行しません。

auto

Hyper-V ホストがシャット ダウンされる前に VM が実行されていた場合、VM を自動的に起動します。

always

Hyper-V が起動するたびに、VM を起動します。

-startDelay *start_delay_secs*

(オプション) Hyper-V ホストが起動した後、VM を起動するための数秒の遅延を指定します。

-stopAction {save|off|shutdown}

(オプション) Hyper-V ホストがシャット ダウンする前に VM で実行するアクションを指定します。オプションは、以下のとおりです。

save

VM を中断します。

off

VM をオフにします。

shutdown

VM をシャットダウンします。

-recoveryAction {none|restart|revert}

(オプション) 予期せずシャットダウンしてから Hyper-V ホストが再起動した後に VM で実行するアクションを指定します。オプションは、以下のとおりです。

none

アクションを実行しません。

restart

VM を再起動します。

revert

最後のスナップショットに戻します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM のプロパティを設定する

この例では、割り当てる CPU コアの数を変更します。

```
dpmhyperv -setVMProperties -ws_user dcaadmin -ws_password #test# -hypervHost hserver  
-vm TestVM -cpus 2
```

dpmhyperv SetVMVirtFloppy コマンド -- 仮想フロッピー イメージを VM フロッピードライブに割り当てる

dpmhyperv SetVMVirtFloppy コマンドは、Hyper-V ホストの VM フロッピードライブに仮想フロッピー イメージを割り当てます。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv SetVMVirtFloppy  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -hypervHost hostname  
-vm vm_name|-vguid vm_ID  
[-path location]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-path *location*

仮想フロッピー イメージの場所を定義します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストの VM フロッピー ドライブに仮想フロッピー イメージを割り当てる

この例では、Hyper-V ホスト「**hserver**」の VM フロッピー ドライブに仮想フロッピー イメージ「**1.vfd**」を接続します。

```
dpmhyperv SetVMVirtFloppy -ws_user admin -ws_password test -hypervHost hserver  
-vm test-vm01 -path C:¥data¥Floppy¥1.vfd
```

dpmhyperv ShowClusterSharedVolumes コマンド -- クラスタ共有ボリュームをすべて表示する

dpmhyperv ShowClusterSharedVolumes コマンドは、Hyper-V Server のすべてのクラスタ共有ボリュームをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowClusterSharedVolumes  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password password

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストのクラスタ共有ボリュームをすべて表示する

この例では、ホスト「**hserver**」の共有ボリュームをすべてリスト表示します。

```
dpmhyperv ShowClusterSharedVolumes -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowDirectories コマンド -- Hyper-V Server のディレクトリのプロパティを表示する

dpmhyperv ShowDirectories コマンドは、Hyper-V ホストのディレクトリのすべてのプロパティをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowDirectories  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-directory directory_path  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-directory *location*

Hyper-V ホストのディレクトリの場所を定義します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストのディレクトリのプロパティを表示する

この例では、ホスト「**hserver**」の「**c:%test**」ディレクトリのすべてのプロパティをリスト表示します。

```
dpmhyperv ShowDirectories -ws_user admin -ws_password #test# -hypervHost hserver
-directory c:%test
```

dpmhyperv ShowLogicalDisks コマンド -- 論理ディスクをすべて表示する

dpmhyperv ShowLogicalDisks コマンドは、Hyper-V ホストのすべての論理ディスクとそのプロパティをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowLogicalDisks
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストの論理ディスクをすべて表示する

この例では、Hyper-V ホスト「**hserver**」の論理ディスクをすべてリスト表示します。

```
dpmhyperv ShowLogicalDisks -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowPhysDisks コマンド -- ホストの物理ディスクを表示する

dpmhyperv ShowPhysDisks コマンドは、Hyper-V ホストのすべての物理ディスクを表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowPhysDisks  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
[-prompt]  
-hypervHost hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password password

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-prompt {yes|no}

(オプション) ユーザ ID とパスワードを求めるプロンプトを表示するかどうかを指定します。

デフォルト : Yes

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ホストの物理ディスクを表示する

この例では、Hyper-V ホスト「**hserver**」の物理ディスクをすべて表示します。

```
dpmhyperv ShowPhysDisks -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowSCVMMHardwareProfiles コマンド -- SCVMM ハードウェア プロファイルをすべて表示する

dpmhyperv ShowSCVMMHardwareProfiles コマンドは、SCVMM ホストのハードウェア プロファイルをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowSCVMMHardwareProfiles  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password] -scvmmHost scvmm_hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-scvmmHost *scvmm_hostname*

(オプション) SCVMM サーバホストの名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: SCVMM ホストのハードウェア プロファイルを取得する

この例では、SCVMM ホスト「ms01」のハードウェア プロファイルをすべて表示します。

```
dpmhyperv ShowSCVMMHardwareProfiles -ws_user admin -ws_password #test# -scvmmHost ms01
```

dpmhyperv ShowSCVMMGuestOSProfiles コマンド -- ゲスト オペレーティング システムのプロファイルを表示する

dpmhyperv ShowSCVMMGuestOSProfiles コマンドは、SCVMM ホストのゲスト オペレーティング システムのプロファイルをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowSCVMMGuestOSProfiles  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-scvmmHost scvmm_hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-scvmmHost *scvmm_hostname*

(オプション) SCVMM サーバホストの名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ゲストオペレーティングシステムのプロファイルを表示する

この例では、SCVMM ホスト「**ms01**」のゲスト オペレーティング システム プロファイルをすべてリスト表示します。

```
dpmhyperv ShowSCVMMGuestOSProfiles -ws_user admin -ws_password #test# -scvmmHost ms01
```

dpmhyperv ShowTemplates コマンド -- VM テンプレートをすべて表示する

dpmhyperv ShowTemplates コマンドは、Hyper-V ホストの VM テンプレートをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowTemplates  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-filter condition]  
[-scvmmHost scvmm_hostname]  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-filter *condition*

(オプション) 「*」などのフィルタ条件を指定します。

-scvmmHost *scvmm_hostname*

(オプション) SCVMM サーバホストの名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM テンプレートをすべて表示する

この例では、ホスト「**hserver**」の VM テンプレートをすべてリスト表示します。

```
dpmhyperv ShowTemplates -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowVirtFloppy コマンド -- 仮想フロッピードライブのリストを表示する

dpmhyperv ShowVirtFloppy コマンドは、VM に割り当てられた仮想フロッピー ディスクをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv GetVMState
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
-vm vmname|-vmguid vm_ID
[-pre]
[-post]
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: フロッピー ディスクをすべて表示する

この例では、GUID「38900009000000yy」を持つ VM に割り当てられた仮想フロッピー ディスクをすべてリスト表示します。

```
dpmhyperv ShowVirtFloppy -ws_user admin -ws_password #test# -hypervHost hserver  
-vmguid 38900009000000yy
```


dpmhyperv ShowVMDisks コマンド -- 仮想ディスクをすべて表示する

dpmhyperv ShowVMDisks コマンドは、Hyper-V ホストの VM の仮想ディスクをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowVMDisks  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname|-vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Hyper-V ホストの仮想ディスクをすべて表示する

この例では、ホスト「**hserver**」の VM「**vm01**」でそのプロパティを持つ仮想ディスクをすべてリスト表示します。

```
dpmhyperv ShowVMDisks -ws_user admin -ws_password #test# -hypervHost hserver -vm test-vm01
```

dpmhyperv ShowVMNics コマンド -- ネットワーク インターフェース カードをすべて表示する

dpmhyperv ShowVMNics コマンドは、VM のネットワーク インターフェース カードをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowVMNics  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname|-vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user username

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password password

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネットワーク インターフェース カードをすべて表示する

この例では、ホスト「**hserver**」の VM「**vm01**」でそのプロパティを持つネットワーク インターフェース カードをすべてリスト表示します。

```
dpmhyperv ShowVMDisks -ws_user admin -ws_password #test# -hypervHost hserver -vm test-vm01
```

dpmhyperv ShowVMs コマンド -- VM をすべて表示する

dpmhyperv ShowVMs コマンドは、Hyper-V ホストの VM をすべてリスト表示します。このコマンドは、VM 名および対応する GUID を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowVMs  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM をすべて表示する

この例では、ホスト「**hserver**」のすべての VM と対応する GUID をリスト表示します。

```
dpmhyperv ShowVMs -ws_user admin -ws_password #test# -hypervHost hserver
```

dpmhyperv ShowVMSCSIControllers コマンド -- SCSI コントローラをすべて表示する

dpmhyperv ShowVMSCSIControllers コマンドは、VM の SCSI コントローラをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ShowVMSCSIControllers  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-vm vmname|-vmguid vm_ID  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-vm *vm_name*

VM 名を指定します。

-vmguid *vm_ID*

VM の一意の GUID を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: SCVMM ホストの SCSI コントローラをすべて表示する

この例では、ホスト「hserver」の仮想マシン「vm01」でそのプロパティを持つ SCSI コントローラをすべてリスト表示します。

```
dpmhyperv ShowVMSCSIControllers -ws_user admin -ws_password #test# -hypervHost hserver -vm vm01
```

dpmhyperv ValidateAgentSNMPAccess コマンド -- エージェントへの SNMP アクセスを検証する

dpmhyperv ValidateAgentSNMPAccess コマンドは、SNMP コミュニティ文字列を使用してエージェントにアクセスできるかどうかを検証します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ValidateAgentSNMPAccess  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
-hypervHost hostname  
-snmpAccessName snmp_name -checkWriteAccess write_access  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-host *hostname*

Hyper-V サーバのホスト名を指定します。

-snmpAccessName *snmp_name*

SNMP コミュニティ文字列を指定します。

-checkWriteAccess *write_access*

書き込みアクセスを選択するかどうかを指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: エージェントへの SNMP アクセスを検証する

この例では、ホスト「**hserver**」への SNMP アクセスを検証します。

```
dpmhyperv ValidateAgentsSNMPAccess -hypervHost hserver -snmpAccessName public
-checkWriteAccess -ws_user admin -ws_password #test#
```

dpmhyperv ValidateServerAccess コマンド -- サーバアクセスを確認する

dpmhyperv ValidateServerAccess コマンドは、ユーザがホスト サーバにアクセスできるかどうか確認します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv ValidateServerAccess
[-ws_user username]
[-ws_password password]
[-ws_encrypted_password]
-hypervHost hostname
[-user username]
[-pass password]
[-pre]
[-post]
[-locale iso639value]

-ws_user username
```

(オプション) **CA Virtual Assurance** のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-user *username*

(オプション) ユーザの名前を指定します。

-pass *password*

(オプション) ユーザのパスワードを指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後でイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サーバアクセスを確認する

この例では、ユーザが Hyper-V ホスト「**hserver**」にアクセスできるかどうか確認します。

```
dpmhyperv ValidateServerAccess -ws_user admin -ws_password #test# -hypervHost hserver
```


dpmhyperv DeleteTemplate コマンド -- テンプレートを削除する

dpmhyperv DeleteTemplate コマンドは、Hyper-V ホストのテンプレートを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmhyperv Deletetemplate  
[-ws_user username]  
[-ws_password password]  
[-ws_encrypted_password]  
[-prompt]  
-hypervHost hostname  
-template template_name  
[-pre]  
[-post]  
[-locale iso639value]
```

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

-ws_encrypted_password

(オプション) CA Virtual Assurance パスワードが暗号化されているかどうかを指定します。

-hypervHost *hostname*

Hyper-V ホストの名前を指定します。

-prompt {yes|no}

(オプション) ユーザ ID とパスワードを求めるプロンプトを表示するかどうかを指定します。

デフォルト : Yes

-template *template_name*

削除するテンプレート名を指定します。

-pre

(オプション) 操作が完了する前にイベントを生成します。

-post

(オプション) 操作が完了した後にイベントを生成します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: テンプレートを削除する

この例では、Hyper-V ホスト「**hserver**」にあるテンプレート「**template_5**」を削除します。

```
dpmhyperv ShowPhysDisks -ws_user admin -ws_password #test# -hypervHost hserver  
-template template_5
```

CA IBM LPAR の CLI コマンド

CLI を使用すると、CA IBM LPAR コマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドは AutoShell 内でも利用できます。

dpmlpar cycle Command -- 論理パーティションのサイクル

dpmlpar cycle コマンドは、論理パーティションの電源オン、電源オフ、リセット、中断を行います。

このコマンドの形式は、以下のとおりです。

```
dpmlpar cycle
[-sc sc_url]
-powerop {activate|restart|shutdown}
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-type {delayed|immediate|os_shutdown|immediate_os_shutdown}]
[-profile_name profilename]
[-activate_bootmode {normal|open_firmware}]
[-activate_keylock {normal|manual}]
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例：https://*hostname:port*/...

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例：localhost.ca.com

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例：80

-powerop {activate|restart|shutdown}

LPAR 上で実行する電源操作を指定します。オプションは、以下のとおりです。

activate

LPAR をオンにします。

restart

LPAR をオフにした後、必要に応じてオンにします。

シャットダウン

LPAR をオフにします。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {immediate|os_shutdown|immediate_os_shutdown}

イメージング操作タイプ リソース グループ (**res_group**) と個別リソース (**individual_res**) のどちらを使用するかを指定します。オプションは、以下のとおりです。

immediate

パーティションをすぐにシャットダウンします。データが部分的にしか更新されていない場合、このオプションは望ましくない結果を引き起こす場合があります。

os_shutdown

標準のシャットダウン コマンドを発行してパーティションをシャットダウンします。このオプションを正常に実行するには、パーティションのイメージが作成されている必要があります。

immediate_os_shutdown

オペレーティング システムのシャットダウン コマンドを発行して、できるだけ短時間でパーティションをシャットダウンします。このコマンドは、他のユーザへのメッセージ送信など、標準のシャットダウン アクティビティを省略します。このオプションを正常に実行するには、パーティションのイメージが作成されている必要があります。

-profile_name *lparprofile* (HMC のみ)

[-profile_name *lparname*] (IVM のみ)

論理パーティションをアクティブ化するとき使用するパーティション プロファイルを指定します。HMC には必須です。IVM については任意で、指定した場合には、その名前がパーティション名と一致する必要があります。

-activate_bootmode {normal|open_firmware}

(オプション) アクティブ化操作のキーロック モードを指定します。オプションは、以下のとおりです。

normal

標準の方法でパーティションを開始します。

open_firmware

パーティションを開始し、ファームウェア プロンプトを開きます。

-activate_keylock {normal|manual|bypass}

(オプション) アクティブ化操作のキーロック モードを指定します。オプションは、以下のとおりです。

normal

自動モードでパーティションを開始します。アクティブ化にはユーザ操作を必要としません。

手動

手動モードでパーティションを開始します。アクティブ化にはユーザ操作が必要となります。

bypass

キーロック モードをアクティブにしません。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale *iso639value*

例: デフォルトプロファイルを使用して論理パーティションをアクティブ化する

以下の例は、デフォルトプロファイルを使用して論理パーティション `testlpar` をアクティブ化します。

```
dpmlpar cycle -powerop activate -hmc uslihmc  
-managed_system testComputer -partition_name testlpar
```

例: 指定したプロファイルを使用して論理パーティションをアクティブ化する

以下の例は、プロファイル `testlparprofile` を使用して論理パーティション `testlpar` をアクティブ化します。

```
dpmlpar cycle -powerop activate -hmc usliivm  
-managed_system testComputer -partition_name testlpar  
-profile_name testlparprofile
```

例: 論理パーティションをアクティブ化し、ファームウェア プロンプトを開く

以下の例は、プロファイル `testlparprofile` を使用して論理パーティション `testlpar` をアクティブ化し、ファームウェア プロンプトを開きます。

```
dpmlpar cycle -powerop activate -hmc uslihmc  
-managed_system testMS -partition_name testlpar  
-activate_bootmode open_firmware
```

例: 論理パーティションの遅延シャットダウン

以下の例は、論理パーティション `testlpar` の遅延シャットダウンを実行します。

```
dpmlpar cycle -hmc testivm -powerop shutdown  
-managed_system managedsys1 -partition testlpar -type delayed
```

dpmlpar delete コマンド -- 論理パーティションの削除

dpmlpar delete コマンドは、非アクティブな論理パーティションを削除します。

重要: このコマンドを発行する *前に*、重要なデータをバックアップしたことを確認してください。このコマンドは、LPAR および LPAR のデータ ファイルを削除します。このコマンドを発行する際は、LPAR の電源をオフにする必要があります。

このコマンドの形式は、以下のとおりです。

```
dpmlpar delete
[-sc sc_url]
-hmc name
-managed_system managedsystemname
-partition_name partitionname
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <false|true>]
[-ws_remote_user username]
[-ws_remote_password password]
[-delete_logical_value <false|true>]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

削除する論理パーティションの名前を指定します。このパーティションは、管理対象システム上に存在する必要があります。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value***-prompt {*yes|no*}**

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale *iso639value*

例: 論理パーティションを削除する

以下の例は、論理パーティション `lpar01` を削除します。

```
dpmlpar delete -hmc hmc02 -managed_system system05  
-partition_name lpar01
```

dpmlpar getresources コマンド -- LPAR リソースの取得

`dpmlpar getresources` コマンドは、IBM AIX LPAR のメモリ ユニット、プロセッサユニット、プロセッサを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar getresources  
[-sc sc_url]  
-resource {all|memory|processors|processor_units}  
-hmc name  
-managed_system managedsystemname  
-partition_name partitionname  
[-pre] [-post]  
[-ws_user username]  
[-ws_password password]  
[-prompt <false|true>]  
[-ws_remote_user username]  
[-ws_remote_password password]  
[-locale iso639value]  
  
-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-resource {all|memory|processors|processor_units}

特定のリソースを取得するのか、すべてのリソースを取得するのかを指定します。オプションは、以下のとおりです。

all

すべてのパーティション リソースを取得、表示します。

memory

パーティションのメモリ リソースのみを取得、表示します。

processors

パーティションのプロセッサ リソースのみを取得、表示します。

processor_units

パーティションのプロセッサ ユニット リソースのみを取得、表示します。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースをリスト表示する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_remote_user username

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password password

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale iso639value

例: LPAR のすべてのリソースを取得する

以下の例は、lpar06 のリソースをすべて取得します。

```
dpmllpar getresources -resource all -partition_name lpar06  
-hmc ivm_02 -managed_system abcsystem4
```

例: LPAR のメモリリソースを取得する

以下の例は、lpartest01 のメモリ リソースを取得します。

```
dpmllpar getresources -resource memory -partition_name lpartest01  
-hmc hmc_02 -managed_system abcsystem2
```

dpmlpar image コマンド -- IBM AIX LPAR を作成する

dpmlpar image コマンドは、HMC/IVM とターゲット IBM AIX LPAR 管理対象システムを使用して IBM AIX LPAR を作成します。（オプション）LPAR を作成した後で IBM AIX オペレーティング システムを展開できます。

このコマンドの形式は、以下のとおりです。

```
dpmlpar image [-sc sc_url]
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-profile_name lparprofile (HMC のみ)
[-profile_name lparname] (IVM のみ)

-min_mem size
-desired_mem size
-max_mem size

-proc_mode {shared|dedicated}
-min_proc_units units
-desired_proc_units units
-max_proc_units units

-min_procs number
-desired_procs number
-max_procs number

-max_virtual_slots number
-share_mode {capped|uncapped}
[-uncap_weight weight]
[-io_slot DRC index,{true|false}]
[-lpar_io_pool_id id {,id, ...}]

[-virtual_serial_adapter slot_num, remote_lpar_name, remote_slot_num, is_required]
[-virtual_scsi_client_adapter slot_num, remote_lpar_name, remote_slot_num,
is_required] |
[-virtual_scsi_server_client_adapter virtual_io_server_name, backing_device_name,
client_slot_num, is_required]
[-virtual_eth_adapter slot_num, is_IEEE, port_vlan_id, (additional_vlan_id,
additional_vlan_id, ...), trunk_priority, is_required]
[-virtual_fc_server_client_adapter virtual_io_server_name, physical_fc_port_name,
client_slot_num, is_required]
[virtual_io_server_name,physical_fc_port_name,client_slot_num,is_required;...]

[-pre] [-post]
[-ws_user username]
[-ws_password password]
```

```
[-encrypted_password {no|yes}]
[-prompt <no|yes>]

[-ws_remote_user username]
[-ws_remote_password password]

[-create_logical_volume <false|true>]
[-logical_volume_size <value>]
[-volume_group_names <value>]
[-logical_volume_name <value>]
[-use_logical_volume_name_as_prefix {no|yes}]
[-default_vio_server_name vioserver]
[-itcm_server itcmservername]
[-locale iso639value]
```

NIM MKSYSB ユーティリティを使用して AIX をイメージングするためのパラメータ

```
[-install_type {rte|mksysb}]

-mksysb mksysb_resource

-lpp lpp_resource

-bosinst_data bos_install_data_resource

-image_data image_data_resource
```

リソース グループを使用して AIX をプロビジョニングするためのパラメータ

```
-provision_aix true
-type res_group
-res_group_name resourcegroupname
-machine_res_name machineresourcenname
-target_username targetusername
[-target_password targetpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
-nim_master_host_name nimmasterhostname
[-scalability_server servername]
[-deploy_template templatename]
-auto_deploy {yes|no}
[-wait [timeout]]
```

個別リソースを使用して AIX をプロビジョニングするためのパラメータ

```
-provision_aix true
-type individual_res
-machine_res_name machineresourcename
-lpp lppresource
-spot spotresource
-bosinst_data bosdata
-resolv_conf resolveconf
-fb_script fbscript
-post_inst_scripts script1,script2,script3
-target_username targetusername
[-target_password targetpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
-nim_master_host_name nimmasterhostname
[-scalability_server servername]
[-deploy_template templatename]
-auto_deploy {yes|no}
[-wait [timeout]]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

作成する論理パーティションの名前を指定します。

-profile_name *lparprofile* (HMC のみ)

[-profile_name *lparname*] (IVM のみ)

論理パーティションをアクティブ化するときに使用するパーティションプロファイルを指定します。HMC には必須です。IVM については任意で、指定した場合には、その名前がパーティション名と一致する必要があります。

-min_mem size

パーティションの最小メモリ容量を定義します。

-desired_mem size

パーティションの推奨メモリ容量を定義します。

-max_mem size

パーティションの最大メモリ容量を定義します。

-proc_mode {shared|dedicated}

パーティションのプロセッサ モードのタイプを指定します。オプションは、以下のとおりです。

shared

他のパーティションとプロセッサ リソースを共有します。

dedicated

専用のプロセッサ リソースがパーティションにあることを指定します。

-min_proc_units *units*

このパーティションの最小プロセッサ ユニット数を定義します。

値の範囲 : .01 ずつインクリメント

注: このオプションは共有プロセッサでのみ使用できます。

-desired_proc_units *units*

このパーティションに割り当てられているプロセッサ ユニット数を定義します。

値の範囲 : .01 ずつインクリメント

注: このオプションは共有プロセッサでのみ使用できます。

-max_proc_units *units*

このパーティションの最大プロセッサ ユニット数を定義します。

値の範囲：.01 ずつインクリメント

注：このオプションは共有プロセッサでのみ使用できます。

-min_procs *number*

このパーティションの最小仮想プロセッサ数を定義します。

注：このオプションは共有プロセッサでのみ使用できます。

-desired_procs *number*

このパーティションに割り当てられている仮想プロセッサ数を定義します。

注：このオプションは共有プロセッサでのみ使用できます。

-max_procs *number*

このパーティションの最大仮想プロセッサ数を定義します。

注：このオプションは共有プロセッサでのみ使用できます。

-shared_mode {*capped|uncapped*}

管理対象システムで、別のパーティションにコミットしていない、共有プロセッサ プールのアイドル プロセッサ ユニットの論理パーティションで使えるようにするかどうかを指定します。

注：このオプションは共有プロセッサでのみ使用できます。

オプションは、以下のとおりです。

capped

パーティションにコミットされているプロセッサ ユニットののみを使用できます。

uncapped

利用可能な場合は共有プロセッサ プールのアイドル プロセッサ ユニットのパーティションで使用できます。

-uncap_weight *weight*

(オプション) 共有モードとして **uncapped** を選択した場合に、処理優先度の加重平均を定義します。

値の範囲：0 ～ 255

-max_virtual_slots *number*

このパーティションの最大仮想アダプタ数を定義します。

デフォルト : 2

値の範囲 : 2 ~ 65,536

-io_slot *DRC-Index*,{true|false}

(オプション) 物理コンポーネントの I/O スロットを指定します。異なる I/O デバイスにこの値を複数回指定できます。

DRC-Index

スロットの動的再構成コネクタ (DRC) インデックスを指定します。

true

I/O スロットに DRC インデックスが必須であることを指定します。

false

I/O スロットに DRC インデックスは必須でないことを指定します。

-io_pool_id *id* {*id* ...}

(オプション) HMC のアクティブな介入なしで、指定された論理パーティショングループが引き継いで使用できる I/O アダプタグループを定義します。パーティショングループは、I/O プール ID をカンマで区切って指定します。

-virtual_serial_adapter *slot_num*, *remote_lpar_name*, *remote_slot_num*, *is_required*

(オプション) この論理パーティションの仮想シリアルアダプタを定義します。変数 *is_required* は、true または false に設定します。サポートされているのはクライアントシリアルアダプタのみですが、複数回指定できます。

デフォルト: スロット 1 と 2 で作成された 2 つのサーバシリアルアダプタ

注: IVM 上で LPAR を作成する場合にはサポートされていません。

-virtual_eth_adapter_client slot_num, is_IEEE, port_vlan_id1 (vlanid2,vlanid3,...), trunk_priority, is_required

(オプション) この論理パーティションの仮想イーサネットアダプタを定義します。複数回指定できます。オプションの値は **vlan_id** と **trunk_priority** です。変数 **is_IEEE** と **is_required** は、true または false に設定します。追加の **vlan_id** は、カンマ区切りかつこで囲む必要があります。

注: IVM 上で LPAR を作成するとき、**slot_num** は 4 以上である必要があります。

例: 4,false,1,,,true

-virtual_fc_server_client_adapter virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required [;virtual_io_server_name, physical_fc_port_name, client_slot_num, is_required;...]

(オプション) 仮想ファイバチャネル (FC) クライアントアダプタ (VIO サーバ上の FC サーバアダプタ) を定義し、物理 FC ポートへ FC サーバアダプタを関連付けます。**-provision_aix** オプションが true に設定されている場合、このオプションは無効です。このパラメータは複数回指定できます。変数 **is_required** は、true または false に設定します。

-virtual_scsi_client_adapter slot_num, remote_lpar_name, remote_slot_num, is_required

(オプション) この論理パーティションの仮想クライアント SCSI アダプタを定義します。複数回指定できます。変数 **is_required** は、true または false に設定します。

-virtual_scsi_server_client_adapter virtual_io_server_name, backing_device_name, client_slot_num, is_required

(オプション) 仮想 SCSI サーバデバイスと仮想 SCSI クライアントデバイスを定義します。この定義は、サーバデバイス作成後に自動的にリンクされ、サーバスロット番号が特定されます。このパラメータは複数回指定できます。変数 **is_required** は、true または false に設定します。

例: CUST-VIOSERVER,hdisk5,3,true

注: **dpmlpar image** コマンドを使用して論理ボリュームを作成し、**-create_logical_volume** オプションを「yes」に設定した場合、**-virtual_scsi_server_client_adapter** オプションは不要で省略されます。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-encrypted_password {no|yes}

(オプション) ユーザ パスワードが暗号化されているかどうかを指定します。

デフォルト: 「いいえ」

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト: Yes

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

`-create_logical_volume {no|yes}`

(オプション) 論理ボリュームを作成するかどうかを指定します。

デフォルト : no

注: `dpmlpar image` コマンドを使用して論理ボリュームを作成し、
`-create_logical_volume` オプションを「yes」に設定した場合、
`-virtual_scsi_server_client_adapter` オプションは不要で省略されます。

`-logical_volume_size logical_volume_size`

(オプション) 論理ボリュームのサイズを MB で定義します。

`-volume_group_names volume_group_1, volume_group_2, volume_group_3`

(オプション) 論理ボリューム グループ名をカンマ区切りのリストで指定します。

`-logical_volume_name logical_volume_name`

(オプション) 論理ボリュームの名前を定義します。

`-use_logical_volume_name_as_prefix {no|yes}`

(オプション) プレフィックスとして論理ボリューム名を使用するかどうかを指定します。

デフォルト : no

`-default_vio_server_name vioserver`

(オプション) 論理ボリュームの作成に使用する VIO サーバを指定します。

`-itcm_server itcm_servername`

(オプション) Unicenter DSM サーバの名前を指定します。

NIM mksysb によるイメージングに使用するパラメータ

-install_type {rte|mksysb}

(オプション) NIM のインストール タイプ。許可される値は **rte** または **mksysb** です。-install_type を指定しない場合は、**rte** がデフォルトです。

-mksysb mksysbResource

mksysb リソース。-install_type が mksysb である場合にのみ有効です。

-lpp lpp_resource

イメージング要求に使用するライセンス プログラム製品ファイルを指定します。-install_type が mksysb である場合はオプションです。

-install_type が **rte** である場合は必須です。

-bosinst_data bos_install_data_resource

ベース オペレーティング システム (BOS) のインストール プログラムに関する情報を含むファイルを指定します。-install_type が mksysb である場合はオプションです。-install_type が **rte** である場合は必須です。

-image_data imageDataResource

(オプション) ルート ボリューム上に物理ディスク、ボリューム グループ、論理ボリューム、ファイル システム、およびページング領域を設定する方法が記述されたイメージデータ リソース ファイルを指定します。

プロビジョニングに使用するパラメータ

-auth_file authorizationfilename

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、**dpmutil** 認証情報設定コマンドを使用して生成した **CA Virtual Assurance** 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp componentID

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-auto_deploy {yes|no}

CA Virtual Assurance エージェントを自動的に展開するかどうかを指定します。 オプションは、以下のとおりです。

yes

CA Virtual Assurance エージェントを自動的に展開します。

no

CA Virtual Assurance エージェントが自動的に展開されないようにします。

デフォルト : no

-bosinst_data *bos_install_data_resource*

ベース オペレーティング システム (BOS) のインストール プログラムに関する情報を含むファイルを指定します。 **-install_type** が **mksysb** である場合はオプションです。 **-install_type** が **rte** である場合は必須です。

-deploy_template *templatename*

(オプション) CA Virtual Assurance で作成された汎用テンプレートの名前を指定します。

注: VMware vCenter で作成および管理されているテンプレートと、このテンプレートを混同しないでください。

-fb_script *fbscript*

(オプション) BOS のインストール プロセスが完了した後、NIM クライアントが初めて起動するときにデバイスの設定に使用するファイルの名前を定義します。

-lpp *lpp_resource*

イメージング要求に使用するライセンス プログラム製品ファイルを指定します。 **-install_type** が **mksysb** である場合はオプションです。 **-install_type** が **rte** である場合は必須です。

-machine_res_name *machineresourcenam*

マシン リソースの名前を定義します。 この名前は NIM マスタで事前定義されている必要があります。

-nim_master_host_name *nimmasterhostname*

イメージ展開を実行する NIM マスタ ホスト名を定義します。

-post_inst_scripts *script1,script2,script3*

(オプション) インストール後に実行するスクリプトをカンマ区切りのリストで指定します。

-provision_aix {true|false}

パーティションを作成した後、NIM を使用してイメージを作成するかどうかを指定します。

true

作成したパーティションのイメージに NIM を使用します。true に設定すると、NIM リソース グループまたは NIM の個別リソース パラメータが使用されます。詳しくは、「dpmnim image コマンド - リソース グループを使用した IBM AIX イメージの展開」、「dpmnim image コマンド - 個別リソースを使用した IBM AIX イメージの展開」を参照してください。ジョブ ID は、NIM プロビジョニングが開始されると返されます。

false

作成したパーティションのイメージに NIM を使用しません。ジョブ ID は返されません。

-res_group_name *resource group name*

リソース グループの名前を定義します。

-resolve_conf *resolveconf*

(オプション) 有効な */etc/resolv.conf* エントリが含まれているファイルを定義します。このエントリで、ローカル リゾルバルーチンのドメイン名プロトコル ネーム サーバ情報を定義します。

-scalability_server *servername*

(オプション) ソフトウェア配信用の Unicenter DSM スケーラビリティ サーバを指定します。

-spot *spotResource [mkysysbResource]*

イメージング要求に使用する共有製品オブジェクト ツリーを定義します。

-target_password *targetpassword*

(オプション) イメージを展開するターゲット ホスト サーバにエージェントを展開するパスワードを定義します。パスワードを指定しない場合は、認証ファイルから取得されます。

注: 認証ファイルの設定には dpmutil CLI を使用します。

-target_username targetusername

イメージを展開するターゲット ホスト サーバにエージェントを展開するユーザ名を定義します。

-type {res_group|individual_res}

イメージング操作タイプ リソース グループと個別リソースのどちらを使用するかを指定します。

res_group

イメージングでリソース グループを使用するように指定します。

individual_res

イメージングで個別リソースを使用するように指定します。

-wait timeout

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (timeout の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの timeout の値 (分) がタイムアウトするまでの時間となります。timeout の値を指定せずに wait オプションを入力すると、CLI は caimgconf.cfg ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、timeout には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

例: IBM AIX 論理パーティションを作成する

この例では、論理パーティションを作成しますが、プロビジョニングは行いません。

```
dpmlpar image -hmc ivm01 -managed_system testMS
-partition_name lptest01 -profile_name lptest01
-max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5
-proc_mode shared -share_mode uncapped -uncap_weight 123
-min_proc_units .5 -max_proc_units 3.25 -desired_proc_units 1.75
-virtual_serial_adapter 2,serial_partition,1,true
-virtual_scsi_client_adapter 3,scsi_partition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true
-io_slot 21020003,false
```

例: IBM AIX 論理パーティションを作成し、個別リソースを使用してイメージングする

この例では、論理パーティションを作成し、個別リソースを使用してプロビジョニングを行います。

```
dpmlpar image -hmc hmc01 -managed_system testMS
-partition_name IRTEST -profile_name Default
-max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 1
-max_procs 1 -proc_mode dedicated
-virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),0,true
-io_slot 21020003,false -provision_aix true
-type individual_res -lpp 530lpp_res -spot 530spot_res
-bosinst_data 530_bid_ow -resolv_conf master_net_conf
-post_inst_scripts piScript -machine_res_name MachineName
-nim_master_host_name machine.mydomain.com -auto_deploy no
-target_username root -target_password password
```

例: IBM AIX 論理パーティションを作成し、リソース グループを使用してイメージングする

この例では、論理パーティションを作成し、リソース グループを使用してプロビジョニングを行います。

```
dpmlpar image -hmc hmc02 -managed_system testMS
-partition_name RGTEST -profile_name Default
-max_virtual_slots 10 -min_mem 128 -desired_mem 256
-max_mem 2048 -min_procs 1 -desired_procs 2 -max_procs 5
-proc_mode dedicated -virtual_serial_adapter 2,SerialPartition,1,true
-virtual_scsi_client_adapter 3,SCSIPartition,7,true
-virtual_eth_adapter 4,true,1,(22,35,54),,true
-io_slot 21020003,false -provision_aix true -type res_group
-res_group_name Res_grp -machine_res_name Machine
-nim_master_host_name machine.mydomain.com -auto_deploy no
-target_username root -target_password password
```

dpmlpar imgjobcheck コマンド -- IBM AIX イメージング ジョブ ステータスの取得

dpmlpar imgjobcheck コマンドは、特定の CA Virtual Assurance ジョブ ID の IBM AIX 論理パーティション展開イメージング ジョブ ステータスを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar imgjobcheck
[-sc sc_url]
-status jobID
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-status *jobID*

ジョブ ステータスの取得に使用する CA Virtual Assurance ジョブ ID を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

-prompt {*yes|no*}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : `Yes`

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

```
-ws_remote_password password
```

(オプション) CA Virtual Assurance リモート パスワードを指定します。

```
-locale iso639value
```

例: ジョブ ID を使用して IBM AIX イメージング ジョブ ステータスを取得する

以下の例は、CA Virtual Assurance ジョブ ID 42 を使用して、IBM AIX 論理パーティション イメージング ジョブ ステータスを取得します。

```
dpmlpar imgjobcheck -status 42
```

dpmlpar list コマンド -- HMC リソースのリスト表示

dpmlpar リスト コマンドは、ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) からの情報をリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar list [-sc sc_url]
-hmc name
[-managed_system managedsystemname]
[-partition_name partitionname]
-display {managed_systems | managed_system_details | partitions | io_components |
backing_devices | scsi_adapters | partition_details | profiles | wwpns}
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-display

`{managed_systems|managed_system_details|partitions|io_components|backing_devices|scsi_adapters|partition_details|profiles|wwpns}`

リスト表示するリソースを指定します。オプションは、以下のとおりです。

managed_systems

管理ハードウェア コンソールで制御する、管理対象システムをリスト表示します。

managed_system_details

管理対象システムに関する詳細情報を提供します。

partitions

管理対象システムのパーティションをリスト表示します。

io_components

管理対象システムの I/O コンポーネントをリスト表示します。

backing_devices

サーバ仮想 SCSI アダプタへのバックング デバイスとして接続できるデバイスを一覧表示します。

scsi_adapters

管理対象システムの SCSI アダプタをリスト表示します。

partition_details

パーティションの詳細をリスト表示します。

profiles

パーティションのプロファイルをリスト表示します。

wwpns

パーティションのワールドワイドポート名を表示します。

-hmc name

情報をリストする HMC/IVM を指定します。

-managed_system managedsystemname

(オプション) 情報をリスト表示する論理パーティションが存在する、管理対象システムを指定します。

注: managed_systems オプションと共に使用することはできません。

-partition_name partitionname

(オプション) 情報をリスト表示する論理パーティションを指定します。

注: managed_systems オプションと共に使用することはできません。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト: Yes

-ws_remote_user username

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password password

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale iso639value

例: 管理対象システムのリストを表示する

以下の例は、ハードウェア管理コンソール **labhmc** の管理対象システムを表示します。

```
dpmlpar list -display managed_systems -hmc labhmc
```

例: SCSI アダプタを表示する

以下の例は、管理対象システム **testMS** の SCSI アダプタを表示します。

```
dpmlpar list -display scsi_adapters -hmc uslihmc  
-managed_system testMS
```

例: パーティションの詳細を表示する

以下の例は、論理パーティション **testlpar** の詳細を表示します。

```
dpmlpar list -display partition_details -hmc uslihmc  
-managed_system testMS -partition_name testlpar
```

dpmlpar setresources コマンド -- メモリリソースの追加

dpmlpar setresources コマンドは、IBM AIX LPAR のメモリ ユニットを調整します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar setresources  
[-sc sc_url]  
-add_memory value  
-hmc name  
-managed_system managedsystemname  
-partition_name partitionname  
-type {dynamic|all}  
[-pre] [-post]  
[-ws_user username]  
[-ws_password password]  
[-prompt <no|yes>]  
[-ws_remote_user username]  
[-ws_remote_password password]  
[-locale iso639value]
```


-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-add_memory *value*

パーティションに追加するメモリの量を定義します。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*dynamic*|*all*}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password *password*

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale *iso639value*

(オプション)デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ(たとえばフランス語の場合は **fr_FR)で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。例: 論理パーティションにメモリを追加する**

以下の例は、論理パーティション **lptest01** に 128 MB のメモリを追加します。

```
dpmlpar setresources -add_memory 128 -partition_name lptest01  
-hmc uni02 -managed_system usil01system1 -type dynamic
```

例：論理パーティションにメモリを追加し、プロファイルを更新する

以下の例は、論理パーティション `lptest01` に 128 MB のメモリを追加し、パーティションプロファイルファイルを更新します。

```
dpmlpar setresources -add_memory 128 -partition_name lptest01
-hmc uni02 -managed_system usil01system1 -type all
-profile_name lptest01
```

dpmlpar setresources コマンド -- プロセッサ リソースの追加

`dpmlpar setresources` コマンドは、IBM AIX LPAR のプロセッサおよびプロセッサユニットを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar setresources
[-sc sc_url]
{-add_processors|add_processor_units} value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例： `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例： `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例： 80

-add_processors *value*

パーティションに追加するプロセッサの数を定義します。

-add_processor_units *value*

パーティションに追加するプロセッサ ユニットを定義します。

値の範囲：.01 ずつインクリメント

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*dynamic*|*all*}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_remote_user username

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password password

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale iso639value

例: パーティションにプロセッサを追加し、プロファイルを更新する

以下の例は、パーティション **lptest01** にプロセッサを追加し、パーティションプロファイルを更新します。

```
dpmlpar setresources -add_processors 1 -partition_name lptest01
-hmc uni02 -managed_system MSystem1 -type all -profile_name lptest01
```

例: パーティションにプロセッサとプロセッサ ユニットを追加する

以下の例は、パーティション **lptest01** にプロセッサとプロセッサ ユニット リソースを一時的に追加します。

```
dpmlpar setresources -add_processors 1 -add_processor_units 4.5
-partition_name lptest01 -hmc uni02 -managed_system MSystem1
-type dynamic
```

dpmlpar setresources コマンド -- メモリリソースの削除

dpmlpar setresources コマンドは、IBM AIX LPAR のメモリ ユニットを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar setresources
[-sc sc_url]
-subtract_memory value
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all}
[-pre] [-post]
[-ws_user username]
[-ws_password password]
[-prompt <no|yes>]
[-ws_remote_user username]
[-ws_remote_password password]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例： `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例： `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例： `80`

-subtract_memory *value*

パーティションから減らすメモリの量を定義します。

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*dynamic*|*all*}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

-prompt {*yes*|*no*}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_remote_user *username*

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

```
-ws_remote_password password
```

(オプション) CA Virtual Assurance リモート パスワードを指定します。

```
-locale iso639value
```

例: 論理パーティションからメモリを削除する

以下の例は、論理パーティション `lptest01` から 128 MB のメモリを削除します。

```
dpmlpar setresources -subtract_memory 128 -partition_name lptest01  
-hmc con02 -managed_system MSystem1 -type dynamic
```

例: 論理パーティションからメモリを削除し、プロファイルを更新する

以下の例は、論理パーティション `lptest01` から 128 MB のメモリを削除し、パーティションプロファイルファイルを更新します。

```
dpmlpar setresources -subtract_memory 128 -partition_name lptest01  
-hmc con02 -managed_system MSystem1 -type all
```

dpmlpar setresources コマンド -- プロセッサ リソースの削除

`dpmlpar setresources` コマンドは、IBM AIX LPAR からプロセッサおよびプロセッサユニットを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar setresources  
[-sc sc_url]  
{-subtract_processors|subtract_processor_units} value  
-hmc name  
-managed_system managedsystemname  
-partition_name partitionname  
-type {dynamic|all}  
[-pre] [-post]  
[-ws_user username]  
[-ws_password password]  
[-prompt <no|yes>]  
[-ws_remote_user username]  
[-ws_remote_password password]  
[-locale iso639value]
```


-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : 80

-subtract_processors *value*

パーティションから減らすプロセッサの数を定義します。

-subtract_processor_units *value*

パーティションから減らすプロセッサ ユニットを定義します。

値の範囲 : .01 ずつインクリメント

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {dynamic|all}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースと現在のプロファイルの値を調整します。調整された値はパーティションについて保存されます。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティチェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value**-prompt {yes|no}**

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_remote_user username

(オプション) CA Virtual Assurance リモート ユーザ名を指定します。

-ws_remote_password password

(オプション) CA Virtual Assurance リモート パスワードを指定します。

-locale iso639value

例: パーティションからプロセッサを削除し、プロファイルを更新する

以下の例は、パーティション `lptest01` からプロセッサを削除し、パーティションプロファイルを更新します。

```
dpmlpar setresources -subtract_processors 1 -partition_name lptest01  
-hmc hmc02 -managed_system MSystem1 -type all
```

例: パーティションからプロセッサとプロセッサ ユニットを削除する

以下の例は、パーティション `lptest01` のプロセッサとプロセッサ ユニットリソースを一時的に調整します。

```
dpmlpar setresources -subtract_processors 1 -subtract_processor_units 4.5  
-partition_name lptest01 -hmc hmc02 -managed_system MSystem1 -type dynamic
```

dpmlpar setresources コマンド -- メモリリソースの更新

`dpmlpar setresources` コマンドは、IBM AIX LPAR のメモリ ユニットを調整します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar setresources  
[-sc sc_url]  
-hmc name  
-managed_system managedsystemname  
-partition_name partitionname  
-type {dynamic|all|profile}  
-min_mem size  
-desired_mem size  
-max_mem size
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*dynamic*|*all*|*profile*}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースとプロファイルの値を調整します。調整された値はパーティションについて保存されます。

profile

プロファイル値を調整します。

-min_mem size

パーティションの最小メモリ容量を定義します。

-desired_mem size

パーティションの推奨メモリ容量を定義します。

-max_mem size

パーティションの最大メモリ容量を定義します。

dpmlpar setresources コマンド -- プロセッサ リソースの更新

dpmlpar setresources コマンドは、IBM AIX LPAR のプロセッサを更新します。

このコマンドの形式は、以下のとおりです。

```
dpmlpar setresources
[-sc sc_url]
-hmc name
-managed_system managedsystemname
-partition_name partitionname
-type {dynamic|all|profile}
[-proc_mode {shared|dedicated}]
[-min_proc_units units]
-desired_proc_units units
[-max_proc_units units]
[-min_procs number]
-desired_procs number
[-max_procs number]
[-share_mode {capped|uncapped} [-uncap_weight weight]]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-hmc *name*

論理パーティションが存在する管理対象システムを制御する ハードウェア管理コンソール (HMC) または Integrated Virtualization Manager (IVM) を指定します。

-managed_system *managedsystemname*

論理パーティションが存在する管理対象システムを指定します。管理対象システムは、HMC/IVM 上に存在する必要があります。

-partition_name *partitionname*

リソースを適用する論理パーティションを指定します。パーティションは、管理対象システム上に存在する必要があります。

-type {*dynamic*|*all*|*profile*}

調整が一時的なものか永続的なものを指定します。オプションは、以下のとおりです。

dynamic

現在のリソースのみを調整します。調整された値は、パーティションをシャットダウンすると失われます。

all

現在のリソースとプロファイルの値を調整します。調整された値はパーティションについて保存されます。

profile

プロファイル値を調整します。

-proc_mode {shared|dedicated}

(オプション) パーティションのプロセッサ モードのタイプを指定します。 オプションは、以下のとおりです。

shared

他のパーティションとプロセッサ リソースを共有します。

dedicated

専用のプロセッサ リソースがパーティションにあることを指定します。

-min_proc_units *units*

(オプション) パーティションの最小プロセッサ ユニット数を定義します。

値の範囲： .01 ずつインクリメント

注： このオプションは共有プロセッサでのみ使用できます。

-desired_proc_units *units*

このパーティションに割り当てられているプロセッサ ユニット数を定義します。

値の範囲： .01 ずつインクリメント

注： このオプションは共有プロセッサでのみ使用できます。

-max_proc_units *units*

(オプション) パーティションの最大プロセッサ ユニット数を定義します。

値の範囲： .01 ずつインクリメント

注： このオプションは共有プロセッサでのみ使用できます。

-min_procs *number*

(オプション) パーティションの最小仮想プロセッサ数を定義します。

注： このオプションは共有プロセッサでのみ使用できます。

-desired_procs *number*

このパーティションに割り当てられている仮想プロセッサ数を定義します。

注： このオプションは共有プロセッサでのみ使用できます。

-max_procs *number*

(オプション) パーティションの最大仮想プロセッサ数を定義します。

注: このオプションは共有プロセッサでのみ使用できます。

-shared_mode {capped|uncapped}

管理対象システムで、別のパーティションにコミットしていない、共有プロセッサ プールのアイドル プロセッサ ユニットの論理パーティションで使えるようにするかどうかを指定します。

注: このオプションは共有プロセッサでのみ使用できます。

オプションは、以下のとおりです。

capped

パーティションにコミットされているプロセッサ ユニットののみを使用できます。

uncapped

利用可能な場合は共有プロセッサ プールのアイドル プロセッサ ユニットをパーティションで使用できます。

-uncap_weight *weight*

(オプション) 共有モードとして **uncapped** を選択した場合に、処理優先度の加重平均を定義します。

値の範囲: 0 ~ 255

CA Microsoft Cluster Server CLI コマンド

スクリプトに CLI を使用して **CA MSCS** コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドは **AutoShell** 内でも利用できます。

dpmmscs getclusterinfo コマンド -- クラスタ情報の取得

getclusterinfo コマンドは、クラスタの一般情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmscs getclusterinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user ユーザ名]  
[-ws_password パスワード]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-cluster

クラスタ名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタ情報の取得

```
dpmmscs.exe getclusterinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs getgroupresourcesinfo コマンド -- クラスタ グループのリソース情報の取得

getgroupresourcesinfo コマンドは、クラスタ グループのリソース情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmscs getgroupresourcesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
-resgroup <resource group>  
[-locale iso639value]  
[-ws_user <ユーザ名>]  
[-ws_password <パスワード>]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-resgroup

リソース グループを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタグループのリソース情報の取得

```
dpmmscs.exe getgroupresourcesinfo -cluster MYCLUSTER -resgroup "Group 0" -ws_user  
admin -ws_password admin
```

dpmmcs getnetworkinterfacesinfo コマンド -- ネットワーク インターフェース情報の取得

getnetworkinterfacessinfo コマンドは、クラスタ ネットワーク インターフェース情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmcs getnetworkinterfacesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
-net <network name>  
[-locale iso639value]  
[-ws_user ユーザ名]  
[-ws_password パスワード]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-net

ネットワーク名を指定します。

-cluster

クラスタ名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタ ネットワーク インターフェース情報の取得

```
dpmmscs.exe getnetworkinterfacesinfo -cluster MYCLUSTER -net "Local Area Connection 2" -ws_user admin -ws_password admin
```

dpmmscs getnetworksinfo コマンド -- ネットワーク情報の取得

getnetworksinfo コマンドは、クラスタ ネットワーク情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmscs getnetworksinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user username]  
[-ws_password パスワード]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-cluster

クラスタ名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタ ネットワーク情報の取得

```
dpmmscs.exe getnetworksinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs getnodesinfo コマンド -- クラスタ ノード情報の取得

getnodesinfo コマンドは、クラスタ ノード情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmscs getnodesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user ユーザ名]  
[-ws_password パスワード]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタノード情報の取得

```
dpmmscs.exe getnodesinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs getresourcegroupsinfo コマンド -- クラスタリソースグループ情報の取得

getresourcegroupsinfo コマンドは、クラスタリソースグループ情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmscs getresourcegroupsinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user ユーザ名]  
[-ws_password パスワード]
```

-sc *sc_url*

(オプション) サービスコントローラホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービスコントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービスコントローラホスト用のリスニングポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタ リソース グループ情報の取得

```
dpmmscs.exe getresourcegroupsinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs getresourcesdependenciesinfo コマンド -- リソースの依存関係情報の取得

getresourcesdependenciesinfo コマンドは、クラスタ リソースの依存関係情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmscs getresourcesdependenciesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
-res <resource name>  
-restype <resource type>  
[-locale iso639value]  
[-ws_user ユーザ名]  
[-ws_password パスワード]
```


-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-res

リソース名を指定します。

-restype

リソース タイプを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタリソースの依存関係情報の取得

```
dpmmscs.exe getresourcedependenciesinfo -cluster MYCLUSTER -res "SQL Server Fulltext" -restype "Generic Service" -ws_user admin -ws_password admin
```

dpmmcs getresourcetypesinfo コマンド -- クラスタ リソース タイプ情報の取得

getresourcetypesinfo コマンドは、クラスタ リソース タイプ情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmcs getresourcetypesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user ユーザ名]  
[-ws_password パスワード]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-cluster

クラスタ名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスタリソース タイプ情報の取得

```
dpmmscs.exe getresourcetypesinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs getservicesinfo コマンド -- クラスタ サービス情報の取得

getservicesinfo コマンドは、クラスタ サービス情報を返します。

このコマンドの形式は、以下のとおりです。

```
dpmmscs getservicesinfo  
[-sc <scUrl>]  
-cluster <cluster>  
[-locale iso639value]  
[-ws_user ユーザ名]  
[-ws_password パスワード]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: クラスター サービス情報の取得

```
dpmmscs.exe getservicesinfo -cluster MYCLUSTER -ws_user admin -ws_password admin
```

dpmmscs movegroup コマンド -- 別のノードへのリソース グループの移動

movegroup コマンドでは、リソース グループを別のノードに移動させることができます。

このコマンドの形式は、以下のとおりです。

```
dpmmscs movegroup  
[ -sc <scUrl> ]  
-cluster <cluster>  
-resgroup <resource group>  
-node <node name>  
[ -locale <ロケール> ]  
[ -ws_user <ユーザ名> ]  
[ -ws_password <パスワード> ]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-resgroup

リソース グループを指定します。

-node

ノード名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: 別のノードへのリソース グループの移動

```
dpmmscs.exe movegroup -cluster MYCLUSTER -resgroup "Group 0" -node ATS-281-W2k3CL  
-ws_user admin -ws_password admin
```

dpmmscs pauseservice コマンド -- クラスタ サービスの一時停止

`pauseservice` コマンドでは、ノード上のクラスタ サービスを一時停止できます。

このコマンドの形式は、以下のとおりです。

```
dpmmscs pauseservice
[ -sc <scUrl> ]
-cluster <cluster>
-node <node name>
[ -locale <ロケール> ]
[ -ws_user <ユーザ名> ]
[ -ws_password <パスワード> ]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-cluster

クラスタ名を指定します。

-node

ノード名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: ノード上のクラスタサービスの一時停止

```
dpmmscs.exe pauseservice -cluster MYCLUSTER -node ATS-281-W2k3CL -ws_user admin  
-ws_password admin
```

dpmmscs bringgrouponline コマンド -- リソース グループのオンライン化

bringgrouponline コマンドでは、リソース グループをオンラインにすることができます。

このコマンドの形式は、以下のとおりです。

```
dpmmscs bringgrouponline  
[ -sc <scUrl> ]  
-cluster <cluster>  
-resgroup <resource group>  
[ -locale <ロケール> ]  
[ -ws_user <ユーザ名> ]  
[ -ws_password <パスワード> ]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-resgroup

リソース グループを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: リソース グループのオンライン化

```
dpmmscs.exe bringgrouponline -cluster MYCLUSTER -resgroup "Group 0" -ws_user admin  
-ws_password admin
```

dpmmscs bringresourceonline コマンド -- リソースのオンライン化

bringresourceonline コマンドでは、リソースをオンラインにすることができます。

このコマンドの形式は、以下のとおりです。

dpmmscs bringresourceonline

```
[ -sc <scUrl> ]  
-cluster <cluster>  
-res <resource name>  
[ -locale <ロケール> ]  
[ -ws_user <ユーザ名> ]  
[ -ws_password <パスワード> ]
```


-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-res

リソース名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: リソースのオンライン化

```
dpmmscs.exe bringresourceonline -cluster MYCLUSTER -res "SQL Server Fulltext"
-ws_user admin -ws_password admin
```

dpmmcs resumeservice コマンド -- クラスタ サービスの再開

resumeservice コマンドでは、ノード上のクラスタ サービスを再開できます。

このコマンドの形式は、以下のとおりです。

dpmmcs resumeservice

[-sc <scUrl>]
-cluster <cluster>
-node <node name>
[-locale <ロケール>]
[-ws_user <ユーザ名>]
[-ws_password <パスワード>]

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-cluster

クラスタ名を指定します。

-node

ノード名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: ノード上のクラスタ サービスの再開

```
dpmmscs.exe resumeservice -cluster MYCLUSTER -node ATS-281-W2k3CL -ws_user admin
-ws_password admin
```

dpmmscs startservice コマンド -- クラスタ サービスの開始

startservice コマンドでは、ノード上のクラスタ サービスを開始できます。

このコマンドの形式は、以下のとおりです。

```
dpmmscs startservice
```

```
[ -sc <scUrl> ]
```

```
-cluster <cluster>
```

```
-node <node name>
```

```
[ -locale <ローケル> ]
```

```
[ -ws_user <ユーザ名> ]
```

```
[ -ws_password <パスワード> ]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-node

ノード名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: ノード上のクラスタ サービスの開始

```
dpmmscs.exe startservice -cluster MYCLUSTER -node ATS-281-W2k3CL -ws_user admin  
-ws_password admin
```

dpmmscs stopservice コマンド -- クラスタ サービスの停止

stopservice コマンドでは、ノード上のクラスタ サービスを停止できます。

このコマンドの形式は、以下のとおりです。

dpmmscs stopservice

```
[ -sc <scUrl> ]  
-cluster <cluster>  
-node <node name>  
[ -locale <ロケール> ]  
[ -ws_user <ユーザ名> ]  
[ -ws_password <パスワード> ]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-node

ノード名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: ノード上のクラスタ サービスの停止

```
dpmmscs.exe stopservice -cluster MYCLUSTER -node ATS-281-W2k3CL -ws_user admin  
-ws_password admin
```

dpmmcs takegroupoffline コマンド -- リソース グループのオフライン化

takegroupoffline コマンドでは、リソース グループをオフラインにすることができます。

このコマンドの形式は、以下のとおりです。

```
dpmmcs takegroupoffline
[ -sc <scUrl> ]
-cluster <cluster>
-resgroup <resource group>
[ -locale <ロケール> ]
[ -ws_user <ユーザ名> ]
[ -ws_password <パスワード> ]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-cluster

クラスタ名を指定します。

-resgroup

リソース グループを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: リソース グループのオフライン化

```
dpmmscs.exe takegroupoffline -cluster MYCLUSTER -resgroup "Group 0" -ws_user admin  
-ws_password admin
```

dpmmscs takeresourceoffline コマンド -- リソースのオフライン化

takeresourceoffline コマンドでは、リソースをオフラインにすることができます。

このコマンドの形式は、以下のとおりです。

```
dpmmscs takeresourceoffline  
[ -sc <scUrl> ]  
-cluster <cluster>  
-res <resource name>  
[ -locale <ロケール> ]  
[ -ws_user <ユーザ名> ]  
[ -ws_password <パスワード> ]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-cluster

クラスタ名を指定します。

-res

リソース名を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-ws_user *username*

(オプション) CA Virtual Assurance のユーザ名を指定します。

-ws_password *password*

(オプション) CA Virtual Assurance のユーザのパスワードを指定します。

例: リソースのオフライン化

```
dpmmscs.exe takeresourceoffline -cluster MYCLUSTER -res "SQL Server Fulltext"
-ws_user admin -ws_password admin
```

CA Solaris ゾーン CLI コマンド

スクリプトに CLI を使用して CA Solaris ゾーン コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドは AutoShell でも利用できます。

dpmzone associateproject コマンド -- タスクとプロジェクトの関連付け

associateproject コマンドは、タスクとプロジェクトを関連付けます。タスクは、時間経過に沿った一連の作業を表すプロセスのコレクションです。タスクはそれぞれ、1 つのプロジェクトに関連付けられます。

このコマンドの形式は、以下のとおりです。

```
dpmzone associateproject
[-sc sc_host]
-host hostname
-name zonename
-proj_name pname
-task_id tid
[-pre]
[-post]
[-locale iso639value]
```


-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

プロジェクトを提供するゾーンを指定します。

-proj_name *pname*

プロジェクトを指定します。

-task_id *tid*

タスクの ID 番号を指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

タスクとプロジェクトを関連付けます。

```
dpmzone associateproject -host SolarisServer2 -name myzone1 -project_name myproject1  
-task_id 1954
```

dpmzone clonezone コマンド -- ゾーンのクローン作成

clonezone コマンドは、既存のゾーンを複製したゾーンを作成します。クローン作成プロセスを開始するには、ソース ゾーンを停止する必要があります。クローン作成を Solaris 10 サーバ上で行う場合は、時間がかかる場合があります。

このコマンドの形式は、以下のとおりです。

```
dpmzone clonezone [-sc sc_host]  
-host hostname  
-name zonename  
-new_name nzonename  
-new_path npath  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

クローンを作成するゾーンを指定します。

-new_name *nzonename*

新しいゾーン名を定義します。

-new_path *npath*

新しいゾーンのパスを定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例

ホスト `SolarisServer2` 上でゾーンのクローンを作成します。

```
dpmzone clonezone -host SolarisServer2 -name myzone1 -new_name myzone2
-new_path /opt/zones/myzone2
```

dpmzone createandinstallzone コマンド -- ゾーンの作成およびインストール

`createandinstallzone` コマンドは、Solaris ホスト上でカスタム パラメータを使用してゾーンを作成し、インストールします。

このコマンドの形式は、以下のとおりです。

```
dpmzone createandinstallzone
[-sc sc_host]
-host hostname
-name zonename
-path zonepath
-type type
[-archive_path path]
[-autoboot]
[-if_type name]
[-ip ip]
[-pool_name pool]
[-sched_type sched]
[-phy_mem pmem]
[-swap_mem smem]
[-lock_mem lmem]
[-desc desc]
[-pre]
[-post]
[-locale iso639value]

-sc sc_host
```

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

新しいゾーンの名前を定義します。

-path *zonepath*

新しいゾーンのパスを定義します。

-type *type*

新しいゾーンのタイプを指定します。オプションは、以下のとおりです。

native

アプリケーションを実行する、Solaris 10 オペレーティング環境の非グローバルゾーンを作成します。

whole-root

パッケージを継承しない完全ルート ゾーンを作成します。

branded

アプリケーションを実行する、非ネイティブ オペレーティング環境を含む非グローバルゾーンを作成します。

デフォルト： native

-archive_path *path*

(オプション) ブランド化ゾーンのオペレーティング環境インストーラのパスを指定します。

制限： この引数はブランド化ゾーンでは必須ですが、ネイティブゾーンでは任意です。

-autoboot

(オプション) システム起動時にゾーンを自動的に起動するように指定します。

注： ゾーン サービスがサーバ上で無効になっている場合、このプロパティの設定に関係なく、ゾーンは自動起動されません。サービスを有効化するには、以下のコマンドを使用します。

```
svcadm enable svc:/system/zones:default
```

-if_type name

(オプション) ゾーンで使用するネットワーク インターフェースのタイプを指定します (例: eri0)。

-ip ip

(オプション) ゾーンの IP アドレスを定義します。

-pool_name pool

(オプション) ゾーンを関連付けるプールの名前を指定します。

-sched_type shed

(オプション) 共有に基づいた CPU 時間の割り当てに使用する、スケジューラのタイプを指定します。共有は、プロジェクトに割り当てられたシステムの CPU リソース部分です。有効な値は以下のとおりです。

ts

タイム共有スケジューラを指定します。これにより、すべてのプロセスに平等に CPU リソースを分配し、特定のプロセスに CPU リソースを集中させません。ts は Solaris オペレーティング環境用のデフォルト スケジューラです。

fss

フェア シェア スケジューラを指定します。これにより、共有に基づいて CPU 時間を割り当てることができます。

-phy_mem pmem

(オプション) ゾーンに割り当てる物理メモリを定義します。この数値にはスケール (K、M、G、T) を適用できます。たとえば、1M は 1 メガバイトです。

-swap_mem smem

(オプション) ゾーンに割り当てるスワップメモリを定義します。

-lock_mem lmem

(オプション) ゾーンに割り当てるロック済みメモリを定義します。ロック済みメモリはページングできません。

-desc description

(オプション) ゾーンの説明を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmzone createpool コマンド -- リソース プールの作成

createpool コマンドは、Solaris 10 ホスト上でリソース プールを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmzone createpool  
[-sc sc_host]  
-host hostname  
-pset_name pset  
-cpu_min mincpu  
-cpu_max maxcpu  
-pool_name pool  
-sched_type sched  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-pset_name *pset*

プロセッサセットの名前を定義します。各プロセッサセット (CPU グループ) には、ゼロ個以上の CPU を含むことができます。

-cpu_min *mincpu*

プロセッサセットの CPU の最小数を定義します。

-cpu_max *maxcpu*

プロセッサ セットの CPU の最大数を定義します。

-pool_name *pool*

ゾーンを関連付けるプールの名前を指定します。

-sched_type *shed*

共有に基づいた CPU 時間の割り当てに使用する、スケジューラのタイプを指定します。共有されるのは、プロジェクトに割り当てられたシステム CPU リソースの一部です。オプションは、以下のとおりです。

ts

すべてのプロセスに適正に CPU リソースを割り当て、特定のプロセスに CPU リソースを集中させないタイム共有スケジューラを指定します。ts は Solaris オペレーティング環境のデフォルト スケジューラです。

fss

共有に基づいて CPU 時間を割り当てることができる、フェア シェア スケジューラを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上でリソース プール **POOL1** を作成します。

```
dpmzone createpool -host SolarisServer2 -pset_name PSET1 -cpu_min 1 -cpu_max 20  
-pool_name POOL1 -sched_type FSS
```

dpmzone createproject コマンド -- プロジェクトの作成

createproject コマンドは、ゾーンでプロジェクトを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmzone createproject  
[-sc sc_host]  
-host hostname  
-name zonename  
-user_id uid  
[-project_id pid]  
-proj_name pname  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

ゾーンを指定します。

-user_id *uid*

ユーザを指定します。

例： root

-project_id *pid*

(オプション) プロジェクト ID を定義します。ID を割り当てるか、またはシステムによって自動的に ID を生成させることができます。

-proj_name *pname*

新しいプロジェクトの名前を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ゾーンでプロジェクトを作成します。

```
dpmzone createproject -host SolarisServer2 -name myzone1 -user_id root
-proj_name myProject
```

dpmzone createzone コマンド -- ゾーンの作成

createzone コマンドは、Solaris ホスト上でカスタム パラメータを使用してゾーンを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmzone createzone
[-sc sc_host]
-host hostname
-name zonename
-path zonepath
-type type
[-autoboot]
[-if_type name]
[-ip ip]
[-pool_name pool]
[-sched_type sched]
[-phy_mem pmem]
[-swap_mem smem]
[-lock_mem lmem]
[-desc desc]
[-pre]
[-post]
[-locale iso639value]

-sc sc_host
```

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

新しいゾーンの名前を定義します。

-path *zonepath*

新しいゾーンのパスを定義します。

-type *type*

新しいゾーンのタイプを指定します。オプションは、以下のとおりです。

native

アプリケーションを実行する、Solaris 10 オペレーティング環境の非グローバルゾーンを作成します。

whole-root

パッケージを継承しない完全ルート ゾーンを作成します。

branded

アプリケーションを実行する、非ネイティブ オペレーティング環境を含む非グローバルゾーンを作成します。

デフォルト： native

-autoboot

(オプション) システム起動時にゾーンを自動的に起動するように指定します。

注： ゾーン サービスがサーバ上で無効になっている場合、このプロパティの設定に関係なく、ゾーンは自動起動されません。サービスを有効化するには、以下のコマンドを使用します。

```
svcadm enable svc:/system/zones:default
```

-if_type *name*

(オプション) ゾーンで使用するネットワーク インターフェースのタイプを指定します (例： **eri0**)。

-ip *ip*

(オプション) ゾーンの IP アドレスを定義します。

-pool_name pool

(オプション) ゾーンを関連付けるプールの名前を指定します。

-sched_type shed

(オプション) 共有に基づいた CPU 時間の割り当てに使用する、スケジューラのタイプを指定します。共有は、プロジェクトに割り当てられたシステムの CPU リソース部分です。有効な値は以下のとおりです。

ts

タイム共有スケジューラを指定します。これにより、すべてのプロセスに平等に CPU リソースを分配し、特定のプロセスに CPU リソースを集中させません。ts は Solaris オペレーティング環境用のデフォルト スケジューラです。

fss

フェア シェア スケジューラを指定します。これにより、共有に基づいて CPU 時間を割り当てることができます。

-phy_mem pmem

(オプション) ゾーンに割り当てる物理メモリを定義します。この数値にはスケール (K、M、G、T) を適用できます。たとえば、1M は 1 メガバイトです。

-swap_mem smem

(オプション) ゾーンに割り当てるスワップメモリを定義します。

-lock_mem lmem

(オプション) ゾーンに割り当てるロック済みメモリを定義します。ロック済みメモリはページングできません。

-desc description

(オプション) ゾーンの説明を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

WebServer4 という名前のゾーンを作成します。

```
dpmzone createzone -host MySolarisServer -name WebServer4 -path /Zones/Apache -type native
-autoboot -if_type eri0 -ip 192.168.100.100 -pool_name WebServer -sched_type fss
-phy_mem 1024M -swap_mem 2048M
```

dpmzone customcommand コマンド -- カスタム コマンドの実行

customcommand コマンドでは、Solaris サーバ上で外部のコマンドまたはスクリプトを実行できます。

注: 制限付きの外部コマンドの詳細については、このガイドの付録「Custom Commands」を参照してください。

このコマンドの形式は、以下のとおりです。

```
dpmzone customcommand
[-sc sc_host]
-host hostname
-cmd command
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-cmd *command*

実行するプログラムまたはスクリプトの名前を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上でコマンドを実行します。

```
dpmzone customcommand -host SolarisServer2  
-cmd /usr/local/private/scripts/test_connections.pl
```

dpmzone deleteproject コマンド -- プロジェクトの削除

deleteproject コマンドは、ゾーンからプロジェクトを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmzone deleteproject  
[-sc sc_host]  
-host hostname  
-name zonename  
-proj_name pname  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

プロジェクトに関連付けられているゾーンを指定します。

-proj_name *pname*

削除するプロジェクトを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

プロジェクトを削除します。

```
dpmzone deleteproject -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone deletezone コマンド -- ゾーンの削除

deletezone コマンドでは、Solaris サーバ上のゾーンを削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone deletezone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name zonename

削除するゾーンを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上のゾーンを削除します。

```
dpmzone deletezone -host SolarisServer2 -name myzone
```

dpmzone installzone コマンド -- ゾーンのインストール

installzone コマンドは、**Solaris** サーバ上でインストール コマンドを起動します。インストールプロセスには時間がかかる場合があります。インストール処理が実行されていることを確認するには、**showprocesses** コマンドでゾーンのステータスを確認します。

このコマンドの形式は、以下のとおりです。

```
dpmzone installzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-archive_path path]  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc sc_host

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

インストールするゾーンを指定します。

-archive_path *path*

(オプション) ブランド化ゾーンのオペレーティング環境インストーラのパスを指定します。

制限: この引数はブランド化ゾーンでは必須ですが、ネイティブゾーンでは任意です。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネイティブゾーンをインストールする

以下の例は、SolarisServer に MyZone3 をインストールします。

```
dpmzone installzone  
-host SolarisServer  
-name MyZone3
```

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例

Solaris 8 を使用して SolarisServer に MyZone4 をインストールします。

```
dpmzone installzone -host SolarisServer -name MyZone4  
-archive_path /opt/zoneos/branded/Solaris8Installer
```


dpmzone movezone コマンド -- ゾーンの移動

movezone コマンドでは、あるパスから別のパスにゾーンを移動できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone movezone  
[-sc sc_host]  
-host hostname  
-name zonename  
-new_path path  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

新しい場所に移動するゾーンを指定します。

-new_path *path*

ゾーンの新しいパスを定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト SolarisServer2 上のゾーンを移動します。

```
dpmzone movezone -host SolarisServer2 -name myzone1  
-new_path /opt/zones/myzone1
```

dpmzone rebootzone コマンド -- ゾーンの再起動

rebootzone コマンドでは、Solaris サーバ上のゾーンを再起動できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone rebootzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

再起動するゾーンを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上のゾーンを再起動します。

```
dpmzone rebootzone -host SolarisServer2 -name myzone
```

dpmzone renamezone コマンド -- ゾーンの名称変更

renamezone コマンドでは、Solaris 10 サーバ上のゾーンの名前を変更できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone renamezone  
[-sc sc_host]  
-host hostname  
-name czonename  
-new_name nzonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-old_name *czonename*

現在のゾーン名を指定します。

-new_name *nzonename*

新しいゾーン名を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上のゾーンの名前を変更します。

```
dpmzone renamezone -host SolarisServer2 -name myzone -new_name yourzone
```

dpmzone setpoolscheduler コマンド -- プール スケジューラの設定

setpoolscheduler コマンドは、リソース プールに使用するスケジューラのタイプを設定します。リソース プールには、フェア シェア スケジューラ (FSS) とタイム共有スケジューラ (TS) の 2 つのスケジューラ タイプがあります。

このコマンドの形式は、以下のとおりです。

```
dpmzone setpoolscheduler
```

```
[-sc sc_host]
```

```
-host hostname
```

```
-pool_name poolname
```

```
-sched_type sched
```

```
[-pre]
```

```
[-post]
```

```
[-locale iso639value]
```

```
-sc sc_host
```

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-pool_name *poolname*

パーティション分割可能な、関連付けられているリソース グループを指定します。

-sched_type *shed*

共有に基づいた CPU 時間の割り当てに使用する、スケジューラのタイプを指定します。共有されるのは、プロジェクトに割り当てられたシステム CPU リソースの一部です。オプションは、以下のとおりです。

ts

すべてのプロセスに適正に CPU リソースを割り当て、特定のプロセスに CPU リソースを集中させないタイム共有スケジューラを指定します。ts は Solaris オペレーティング環境のデフォルト スケジューラです。

fss

共有に基づいて CPU 時間を割り当てることができる、フェア シェア スケジューラを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: プール スケジューラを FSS に設定する

以下の例は、リソース プールのスケジューリングにフェア シェア スケジューラを使用するように設定します。

```
dpmzone setpoolscheduler -host SolarisServer -pool_name test_pool  
-sched_type FSS
```

例: プール スケジューラを TS に設定する

以下の例は、リソース プールのスケジューリングにタイム共有スケジューラを使用するように設定します。

```
dpmzone setpoolscheduler -host SolarisServer -pool_name test_pool  
-sched_type TS
```

dpmzone setprocessorset コマンド -- プロセッサ セットの CPU 最大数の設定

`setprocessorset` コマンドは、プロセッサ セットの CPU の最大数を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmzone setprocessorset  
[-sc sc_host]  
-host hostname  
-processor_set_name psetname  
-cpu_max maxcpu  
[-pre]  
[-post]  
[-locale iso639value]
```

`-sc sc_host`

(オプション) サービス コントローラのホスト名を指定します。

`-host hostname`

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

`-processor_set_name psetname`

プロセッサ セットを指定します。各プロセッサ セット (CPU グループ) には、ゼロ個以上のプロセッサを含むことができます。

-cpu_max *maxcpu*

プロセッサ セットの CPU の最大数を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

プロセッサ セットの CPU の最大数を設定します。

```
dpmzone setprocessorset -host SolarisServer -processor_set_name test_pset -cpu_max 64
```

dpmzone showinterfaces コマンド -- ネットワーク インターフェースの表示

showinterfaces コマンドは、Solaris 10 ホスト上に存在するネットワーク インターフェースをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

dpmzone showinterfaces

[-sc *sc_host*]

-host *hostname*

[-locale *iso639value*]

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上のネットワーク インターフェースをすべてリスト表示します。

```
dpmzone showinterfaces -host SolarisServer2
```

dpmzone showpools コマンド -- リソース プールの表示

showpools コマンドは、Solaris ホスト上に存在するリソース プールをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

dpmzone showpools

[-sc *sc_host*]

-host *hostname*

[-locale *iso639value*]

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト SolarisServer2 上のリソース プールをすべてリスト表示します。

```
dpmzone showpools -host SolarisServer2
```

dpmzone showprojects コマンド -- プロジェクトのプロパティの表示

showprojects コマンドは、プロジェクトのプロパティをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone showprojects  
[-sc sc_host]  
-host hostname  
-name zonename  
-proj_name pname  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

ゾーンを指定します。

-proj_name *pname*

ゾーンに関連付けられているプロジェクトを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例

プロジェクトのプロパティをリスト表示します。

```
dpmzone showprojects -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone showresourceset コマンド -- リソース セットの表示

`showresourceset` コマンドは、リソース プールのリソースをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone showresourceset  
[-sc sc_host]  
-host hostname  
-pool_name poolname  
[-locale iso639value]
```

`-sc sc_host`

(オプション) サービス コントローラのホスト名を指定します。

`-host hostname`

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

`-pool_name poolname`

パーティション分割可能な、関連付けられているリソース グループを指定します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例

`MyZonesResourcePool` 内のリソース セットをリスト表示します。

```
dpmzone showresourceset -host SolarisServer2 -pool_name MyZonesResourcePool
```

dpmzone showtasks コマンド -- プロジェクトのタスクのリスト表示

`showtasks` コマンドは、プロジェクトのタスクをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone showtasks  
[-sc sc_host]  
-host hostname  
-name zonename  
-proj_name pname  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

ゾーンを指定します。

-proj_name *pname*

ゾーンに関連付けられているプロジェクトを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例

プロジェクトのタスクをリスト表示します。

```
dpmzone showtasks -host SolarisServer2 -name myzone1 -project_name myproject1
```

dpmzone showzoneinterfaces コマンド -- ゾーンのネットワーク インターフェースの表示

`showzoneinterfaces` コマンドは、ゾーンのネットワーク インターフェースをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

`dpmzone showzoneinterfaces`

`[-sc sc_host]`

`-host hostname`

`-name zonename`

`[-locale iso639value]`

`-sc sc_host`

(オプション) サービス コントローラのホスト名を指定します。

`-host hostname`

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

`-name zonename`

ゾーンを指定します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例

SolarisServer2 上のゾーン `MyZone3` のネットワーク インターフェースをすべてリスト表示します。

```
dpmzone showzoneinterfaces -host SolarisServer2 -name MyZone3
```

dpmzone showzoneprocess コマンド -- ゾーンのプロセスの表示

`showzoneprocess` コマンドは、ゾーン上のプロセスをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

`dpmzone showzoneprocess`

`[-sc sc_host]`

`-host hostname`

`-name zonename`

`[-locale iso639value]`

`-sc sc_host`

(オプション) サービス コントローラのホスト名を指定します。

`-host hostname`

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

`-name zonename`

ゾーンを指定します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例

SolarisServer2 上のゾーン `MyZone3` のプロセスをリスト表示します。

```
dpmzone showzoneprocess -host SolarisServer2 -name MyZone3
```

dpmzone showzoneprojects コマンド -- ゾーンのプロジェクトの表示

`showzoneprojects` コマンドは、ゾーン上のプロジェクトをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone showzoneprojects  
[-sc sc_host]  
-host hostname  
-name zonename  
[-locale iso639value]
```

`-sc sc_host`

(オプション) サービス コントローラのホスト名を指定します。

`-host hostname`

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

`-name zonename`

ゾーンを指定します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例

SolarisServer2 上のゾーン `MyZone3` のプロジェクトをすべてリスト表示します。

```
dpmzone showzoneprojects -host SolarisServer2 -name MyZone3
```

dpmzone showzones コマンド -- ゾーンの表示

showzones コマンドは、Solaris ホスト上に存在するゾーンをすべてリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmzone showzones  
[-sc sc_host]  
-host hostname  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上のゾーンをすべてリスト表示します。

```
dpmzone showzones -host SolarisServer2
```

dpmzone startzone コマンド -- ゾーンの起動

startzone コマンドでは、Solaris 10 サーバ上のゾーンを起動できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone startzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

起動するゾーンを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上のゾーンを起動します。

```
dpmzone startzone -host SolarisServer2 -name myzone
```

dpmzone stopzone コマンド -- ゾーンの停止

stopzone コマンドでは、Solaris 10 サーバ上のゾーンを停止できます。

このコマンドの形式は、以下のとおりです。

```
dpmzone stopzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```


-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

停止するゾーンを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンド プロンプトのロケールを使用する場合は「**native**」を指定します。

例

ホスト **SolarisServer2** 上のゾーンを停止します。

```
dpmzone stopzone -host SolarisServer2 -name myzone
```

dpmzone uninstallzone コマンド -- ゾーンのアインストール

`uninstallzone` コマンドでは、Solaris 10 サーバ上のゾーンをアンインストールできます。

このコマンドの形式は、以下のとおりです。

```
dpmzone uninstallzone  
[-sc sc_host]  
-host hostname  
-name zonename  
[-pre]  
[-post]  
[-locale iso639value]
```

-sc *sc_host*

(オプション) サービス コントローラのホスト名を指定します。

-host *hostname*

Solaris ゾーン サーバを指定します。このパラメータは、大文字と小文字を区別します。

-name *zonename*

アンインストールするゾーンを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例

ホスト `SolarisServer2` 上のゾーンをアンインストールします。

```
dpmzone uninstallzone -host SolarisServer2 -name myzone
```

CA VMware vCenter Server の CLI コマンド

CLI を使用して、CA VMware vCenter Server のコマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。対応するコマンドは AutoShell でも利用可能です。

dpmvc addesxhost コマンド -- ESX ホストを追加する

dpmvc addesxhost コマンドは、vCenter Server に ESX ホストを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmvc addesxhost
-esx_host_name esxhostname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-esx_host_user esxhostuser
-esx_host_password esxhostpassword
[-management_ip managementip]
[-host_connect_port hostconnectport]
[-folder_name foldername]
[-datacenter_name datacentername]
[-cluster_name clustername]
[-vim_account_user vimaccountuser]
[-vim_account_password vimaccountpassword]
[-locale iso639value]

-esx_host_name esxhostname
```

VM をホストする ESX サーバの名前を指定します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。 *timeout* の値を指定せずに **wait** オプションを入力すると、CLI は `caimgconf.cfg` ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-esx_host_user *esxhostuser*

(オプション) ESX ホストにアクセスするユーザ名を指定します。

-esx_host_password *esxhostpassword*

ESX ホストへのアクセスに使用するパスワードを指定します。

-management_ip *managementip*

(オプション) 管理 IP を指定します。

-host_connect_port *hostconnectport*

(オプション) vCenter Server と通信するために ESX ホストで使用されるポートを指定します。

-folder_name *foldername*

(オプション) ESX ホストを追加できるフォルダの名前を指定します。

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-cluster_name *clustername*

(オプション) ESX ホストを追加できるクラスタの名前を指定します。

-vim_account_user *vimaccountuser*

(オプション) VIM アカウントのユーザ名を指定します。

-vim_account_password *vimaccountpassword*

(オプション) VIM ホストにアクセスするためのパスワードを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ESX ホストを追加する

この例では、「**esx1**」という ESX ホストを「**myvcenter**」という vCenter Server に追加します。

```
dpmvc addesx -esx_host_name esx1 -vc_server myvcenter -esx_host_user admin  
-esx_host_password topsecret
```

dpmvc addvmvdisk コマンド -- 仮想ディスクの追加

dpmvc addvmvdisk コマンドは、VM に仮想ディスクを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmvc addvmvdisk
-datastore_name datastorename
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-capacity_in_mb disk_capacity
[-controller_key controllerkey]
[-disk_mode
{append|independent_nonpersistent|independent_persistent|nonpersistent|persistent
|undoable}]
-thin_provisioning {no|yes}
[-unit_number unitnumber]
[-locale iso639value]
```

-datastore_name *datastorename*

新しい VM が存在するデータ ストアの名前を指定します。

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は `caimgconf.cfg` ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-capacity_in_mb *disk_capacity*

仮想ディスクのサイズを MB で指定します。

-controller_key *controllerkey*

(オプション) ディスクのコントローラ キーを指定します。

-disk_mode {append | independent_nonpersistent | independent_persistent | nonpersistent | persistent | undoable}

(オプション) ディスクのモードを指定します。

-thin_provisioning {no | yes}

(オプション) ディスクがシンプロビジョニングである必要があるかどうかを示します。

-unit *disk_unitnumber*

(オプション) ディスクのユニット数を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 仮想ディスクを追加する

以下の例は、VM **myvm** に仮想ディスクを追加します。

```
dpmvc addvmvdisk -datastore_name disk1 -vm_name myvm -vc_server myvcenterserver  
-capacity_in_mb 20000 -thin_provisioning no
```


dpmvc addvmvnic コマンド -- 仮想 NIC の追加

dpmvc addvmvnic コマンドは、VM に仮想 NIC を追加します。

このコマンドの形式は、以下のとおりです。

```
dpmvc addvmvnic
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-devicetype {e1000 | vmxnet}
-network_name networkname
[-mac_address macaddress]
-wake_on_lan_enabled {no | yes}
[-locale iso639value]
```

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (**timeout** の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの **timeout** の値 (分) がタイムアウトするまでの時間となります。**timeout** の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-devicetype {e1000 | vmxnet}

ネットワーク デバイスのタイプを指定します。

-network_name *networkname*

NIC で接続する必要がある仮想ネットワークを指定します。以下の命名規則に基づいて、標準スイッチと分散仮想スイッチの名前を区別できます。

- 標準スイッチの名前はネットワーク名です。
- 分散仮想スイッチの名前は、**dvPort** グループ名の後に、丸かっこで囲まれた分散仮想スイッチ名を連結したもの (**dvPortGroupName (dvSwitchName)**) です。

-mac_address *macaddress*

(オプション) ネットワーク デバイスの **MAC** (Media Access Control) アドレスを指定します。

`-wake_on_lan_enabled {no | yes}`

LAN が有効な場合に VM を起動するかどうかを指定します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: vNIC を追加する

以下の例は、VM に仮想 NIC を追加します。

```
dpmvc addvmvnic -vm_name myvm -vc_server myvcenterserver -devicetype vmxnet  
-network_name net1 -wake_on_lan_enabled no
```

dpmvc clone コマンド -- VM のクローン作成

dpmvc clone コマンドを使用して、VM のクローンを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc clone
[-sc sc_url]
-datacenter_name datacentername
-compute_resource_name name
-esx_host_name esxhostname
-datastore_name datastorename
-vm_name vmname
-resource_pool_name resourcepool
-spec_name specificationname
[-auto_deploy value]
[-deploy_template templatename]
[-network_connection
"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_
gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask[,win_dns_server=wins_dns_
server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_sec
ondary"]]
```

```
[-global_dns_search_suffix suffix1[,suffix2,suffix3,...]]
[-linux_domain_name domain_name]
[-linux_dns_servers primary_dns=value,secondary_dns=value,tertiary_dns=value]
[-create_disk MB,datastorename,controller]
[-modify_disk key, datastorename]
[-set_memory number]
[-set_cpu number]
-vm_os_username username
[-vm_os_password password]
[-vc_server vcservername]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID] {-template_name templatename |
-vm_name_cloned_from -vmname}
[-scalability_server scalabilityservername]
[-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-compute_resource_name *name*

VM を作成するクラスタまたは VMware ESX ホストの名前を指定します。

-esx_host_name *esxhostname*

VM が存在する VMware ESX サーバを指定します。

-datastore_name *datastorename*

新しい VM が存在するデータ ストアの名前を指定します。

-vm_name *vmname*

VM を指定します。

-resource_pool_name *resourcepool*

クローンを作成する VM の選択元のリソース プールの名前を指定します。

-spec_name *specificationname*

(オプション) クローン作成する仮想マシンに使用する仕様の名前を指定します。

-auto_deploy {yes|no}

CA Virtual Assurance エージェントを自動的に展開するかどうかを指定します。 オプションは、以下のとおりです。

yes

CA Virtual Assurance エージェントを自動的に展開します。

no

CA Virtual Assurance エージェントが自動的に展開されないようにします。

デフォルト : no

-deploy_template *templatename*

(オプション) CA Virtual Assurance で作成された汎用テンプレートの名前を指定します。

注: VMware vCenter で作成および管理されているテンプレートと、このテンプレートを混同しないでください。

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask,win_dns_server=wins_dns_server,win_alt_dns=wins_alt_dns,wins_primary=wins_primary,wins_secondary=wins_secondary"

Windows :

(オプション) VM のネットワーク接続を作成します。 このパラメータは複数回指定できます。 シーケンス番号は、ネットワーク接続を識別する 1 から始まる整数値です。 スペースが含まれる場合は、パラメータ全体を丸かっこで囲む必要があります。 nic、network_name、ip_address、def_gateway、subnet_mask 以外のすべての値は、値なしで名前を入力するとリセットできます (例 : "nic=123,alt_gateway=") 。

-network_connection

"nic=sequence_number,network_name=name,ip_address=ip_address,def_gateway=default_gateway,alt_gateway=alt_gateway,subnet_mask=subnet_mask"

Linux :

(オプション) 1 つ以上のネットワーク接続を作成します。 このパラメータは複数回指定できます。 シーケンス番号は、ネットワーク接続を識別する 1 から始まる整数値です。 スペースが含まれる場合は、パラメータ全体を丸かっこで囲む必要があります。 値なしで名前を入力するとリセットできるのは、alt_gateway のみです (例 : "nic=123,alt_gateway=") 。

`-global_dns_search_suffix suffix1{,suffix2,suffix3,...}`

Linux :

1 つ以上の DNS 検索サフィックスを指定します。

Windows :

(オプション) 1 つ以上の DNS 検索サフィックスを指定します。

`-linux_domain_name domain_name`

Linux :

(オプション) Linux VM のドメイン名を指定します。

`-linux_dns_servers`

`primary_dns=value{,secondary_dns=value,tertiary_dns=value}`

Linux :

(オプション) Linux VM の DNS サーバの IP アドレスを指定します。値を省略すると、値の設定を解除できます (例 : `primary_dns=123.123.123.13,secondary_dns=,tertiary_dns=123.123.123.15`) 。

`-create_disk MB,datastore_name,controller`

(オプション) 1 つ以上の追加のハード ドライブを作成します。このパラメータは複数回指定できます。

MB

ハード ドライブのサイズをメガバイトで定義します。

Datastore_name

このハード ドライブのデータ ストアの名前を指定します。現在のところ、VM ごとに 1 つのデータ ストアのみがサポートされています。

Controller

このハード ドライブ用のコントローラ キーを指定します。

-modify_disk {key,datastore_name}

(オプション) ハードドライブが存在するデータストアを指定します。複数指定できます。

key

既存のディスクを指定します。

datastore_name

このディスクを移すデータストアの名前を指定します。

-set_memory number

(オプション) クローン作成する VM のメモリの量をメガバイト (MB) で定義します。この値は、テンプレートで指定された値よりも優先されます。

-set_cpu number

(オプション) クローン作成する VM の CPU の数を定義します。この値は、テンプレートで指定された値よりも優先されます。

-vm_os_username user name

クローン作成する VM のユーザを指定します。イメージを自動展開する場合は、このユーザ名も認証に使用されます。

Windows : カスタマイズの仕様に定義したユーザ名である必要があります。

Linux : テンプレートで定義したユーザ名である必要があります。

-vm_os_password password

(オプション) クローン作成する VM のユーザのパスワードを指定します。イメージを自動展開する場合は、このパスワードも認証に使用されます。

Windows : カスタマイズの仕様に定義したパスワードと同じである必要があります。

Linux : テンプレートで定義したパスワードと同じである必要があります。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user *vcuser*

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password *vcpassword*

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmtutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-template_name *templatename*

クローン作成する仮想マシンに使用するテンプレートの名前を指定します。このパラメータは -vm_name_cloned_from を置換します。また、両方を指定することはできません。

-vm_name_cloned_from *vmname*

クローン作成する仮想マシンのテンプレートとして使用する仮想マシンの名前を指定します。このパラメータは -template_name を置換します。また、両方を指定することはできません。

-resource_pool_name *resourcepool*

クローンを作成する VM の選択元のリソース プールの名前を指定します。

-scalability_server *servername*

(オプション) ソフトウェア配信用の Unicenter DSM スケーラビリティ サーバを指定します。

-spec_name *specificationname*

(オプション) クローン作成する仮想マシンに使用する仕様の名前を指定します。

-compute_resource_name *name*

VM を作成するクラスタまたは VMware ESX ホストの名前を指定します。

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は `caimgconf.cfg` ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: グローバル認証情報を使用して VM のクローンを作成する

この例では、データ センター DCA/MyCity 上でテンプレート「BaseW2k3」を使用して新しい VM「testvm01」を作成します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -vc_server vc_server_1
```

例: 許可ファイルおよびコンポーネントを使用して VM のクローンを作成する

この例では、データ センター DCA/MyCity 上でテンプレート「BaseW2k3」を使用して新しい VM「testvm01」を作成します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -vc_server vc_server_1 -vc_user VCAdmin1
-auth_file c:\localauth.dat -auth_comp Imaging
```

例: グローバル認証情報を使用して VM のクローンを作成し、2 つの CPU を作成する

この例では、データ センター DCA/MyCity 上でテンプレート「BaseW2k3」を使用して新しい VM「testvm01」を作成し、この VM 用に 2 つの CPU を作成します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -set_cpu 2 -vc_server vc_server_1
```

例: デフォルトの許可ファイルを使用して VM のクローンを作成し、メモリを 4096 MB に設定する

この例では、データ センター DCA/MyCity 上でテンプレート「BaseW2k3」を使用して新しい VM「testvm01」を作成し、メモリを 4096 に設定します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -set_memory 4096 -vm_server vm_server_1
-vm_user VCAdmin1
```

例: デフォルトの許可ファイルを使用して VM のクローンを作成し、2 つのハードディスクを作成する

この例では、データ センター DCA/MyCity 上でテンプレート「BaseW2k3」を使用して新しい VM「testvm01」を作成し、データ ストア storage1 用に 2 つのハードディスクを作成します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000
-create_disk 10000,storage1,2001 -vm_server_1 -vm_user VCAdmin1
```

例: VM のクローンを作成し、ネットワーク接続を作成する

この例では、データ センター DCA/MyCity 上でテンプレート「BaseW2k3」を使用して新しい VM「testvm01」とネットワーク接続を作成し、ネットワーク接続を作成して初期値を設定します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -network_connection "nic=1,network_name=VM
Network,ip_address=123.321.5.22,def_gateway=123.321.1.108,subnet_mask=255.255.255
.0,win_dns_server=123.321.42.1,win_alt_dns=123.321.3.101,wins_primary=123.321.3.3
"
```

例: デフォルトの許可ファイルを使用して電源オフの VM から VM のクローンを作成する

この例では、データ センター DCA/MyCity 上でテンプレートの代わりに VM「testvm02」を使用して新しい VM「testvm01」を作成します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -datacenter_name "DCA/MyCity" -vm_name testvm01 -datastore_name storage1
-esx_host_name ussd-dpmvc.ca.com -resource_pool_name Resources/DPMTTest
-compute_resource_name DPMCluster/ussdCluster -vm_name_cloned_from testvm02
-auto_deploy yes -spec_name w2k3_spec_01 -vm_os_username administrator
-vm_os_password mypassword -vc_server vm_server_1.myco.com -vc_user VCAdmin1
```

例: デフォルトの許可ファイルを使用して VM のクローンを作成し、既存のハードディスクを変更する

この例では、データ センター DCA/MyCity 上でテンプレート「BaseW2k3」を使用して新しい VM「testvm01」を作成し、データ ストア storage1 用に 2 つのハードディスクを作成します。クローン操作が完了すると、CA Virtual Assurance エージェントが自動的に展開されます。

```
dpmvc clone -vm_name testvm01 -template_name Templates/BaseW2k3 -auto_deploy yes
-spec_name w2k3_spec_01 -datacenter_name DCA/MyCity -datastore_name storage1
-compute_resource_name DCACluster/ClusterServerA -vm_os_username administrator
-vm_os_password testpassword -create_disk 10000,storage1,1000 -modify_disk
2000,storage2
-vm_server_1 -vm_user VCAdmin1
```

dpmvc createdatastore コマンド -- データストアの作成

dpmvc createdatastore コマンドは、新しいデータストアを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmvc createdatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
-datastore_type {local|nas|vmfs}
[-vmfs_device_path path]
[-vmfs_block_size_mb blocksize]
[-local_path lpath]
[-nas_user_name nasuser]
[-nas_password naspasswd]
[-nas_remote_hostname nasremhost]
[-nas_remote_path nsarempath]
[-nas_access_mode {read-only|read-write}]
[-sc sc_url]
[-locale iso639value]

-datastore_name datastorename
```

新しい VM が存在するデータ ストアの名前を指定します。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-datastore_type {vmfs|nas|local}

作成するデータストアの種類 (VMFS、NAS、またはローカル) を指定します。

-vmfs_device_path *path*

(オプション) 新しいデータストアを作成するときに使用する VMFS ディスクのデバイスパスを指定します。このパスは、「getavailblescsidisks」コマンドを使用して取得できます。

-vmfs_block_size_mb *blocksize*

(オプション) 新しいデータストアを作成するときに使用するブロックサイズを指定します (MB 単位)。利用可能なブロックサイズは 1、2、4、および 8 です。これらのブロックサイズによって、VM ディスク ファイルをそれぞれ最大 256 GB、512 GB、1024 GB、および 2048 GB まで使用できます。

-local_path *lpath*

(オプション) このパスを使用して、ローカルデータストアの作成時に使用される ESX ホストのローカルファイルシステムパスを指定します。

-nas_user_name *nasuser*

(オプション) NAS データストア作成用のリモート ホスト ユーザ名を指定します。

-nas_password *naspasswd*

(オプション) NAS データストア作成用のリモート ホスト パスワードを指定します。

-nas_remote_hostname *nasremhost*

(オプション) ネットワークベースのストレージをホストするサーバのホスト名を指定します。

-nas_remote_path *nasrempath*

(オプション) NAS データストアに使用されるリモート サーバ上のファイルシステムパスを指定します。

-nas_access_mode nasaccmode

(オプション) NAS データストアに使用されるネットワークベースのストレージに対して希望するアクセスの種類を指定します。利用可能なモードは「読み取り専用」および「読み書き」です。指定しない場合のデフォルトのモードは「読み書き」です。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : **80**

dpmvc cycle コマンド -- VM のサイクル

dpmvc cycle コマンドは VM の電源オン、電源オフ、リセット、中断を行います。

このコマンドの形式は、以下のとおりです。

```
dpmvc cycle
[-sc sc_url]
-powerop {poweron|poweroff|reset|suspend|shutdown guest}
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcserver]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-powerop {poweron|poweroff|reset|suspend|shutdown guest}

VM 上で実行する電源操作を指定します。 オプションは、以下のとおりです。

poweron

VM の電源をオンにします。

poweroff

VM の電源をオフにします。

reset

VM をリセットします。

suspend

一時的に VM を停止します。

shutdown guest

ゲスト OS を正常にシャットダウンします。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。 単一の vCenter Server 環境の場合はオプションです。

-vc_user *vcuser*

(オプション) vCenter Server のユーザ ID を指定します。 グローバル認証情報が有効な場合はオプションです。

-vc_password *vcpassword*

(オプション) vCenter Server のユーザ ID のパスワードを指定します。 このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、**dpmutil** 認証情報設定コマンドを使用して生成した **CA Virtual Assurance** 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループホストとユーザに対して使用できるコンポーネント ID を指定します。

-wait *timeout*

(オプション) ジョブステータスを表示し、操作が完了、失敗、またはタイムアウト (**timeout** の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの **timeout** の値 (分) がタイムアウトするまでの時間となります。**timeout** の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: データセンターに複数の vCenter Server がある場合、VM の電源をオフにする

以下の例は、特定の vCenter Server から VM の電源をオフにします。

```
dpmvc cycle -powerop poweroff -vc_server my_server1 -datacenter_name my_dc -vm_name my_vm
```

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、VM の電源をオフにする

以下の例は、データセンター DCA/MyCitySan Diego に属している VM **testvm** の電源をオフにします。

```
dpmvc cycle -vm_name testvm -powerop poweroff -datacenter_name DCA/MyCity
```

例: デフォルトの認証ファイルを使用して、VM の電源をオフにする

以下の例は、データセンター DCA/MyCitySan Diego に属している VM **testvm** の電源をオフにします。

```
dpmvc cycle -vm_name testvm -powerop poweroff -datacenter_name DCA/MyCity -vc_server vc_server_1
```

例: 指定した認証ファイルとコンポーネントを使用して、VM の電源をオンにする

以下の例は、データ センター DCA/MyCitySan Diego に属している VM `testvm` の電源をオンにします。

```
dpmvc cycle -vm_name testvm -powerop poweron -datacenter_name DPM/San Diego  
-vc_server vc_server_1 -vc_user VCAdmin1 -auth_file c:\¥localauth.dat  
-auth_component Imaging
```

例: デフォルトの認証ファイルと指定したコンポーネントを使用して、VM の電源をオンにする

以下の例は、データ センター DCA/MyCitySan Diego に属している VM `testvm` の電源をオンにします。

```
dpmvc cycle -vm_name testvm -powerop poweron -datacenter_name DPM/San Diego  
-vc_server vc_server_1 -vc_user VCAdmin1 -auth_component Imaging
```

dpmvc datastore コマンド -- データ ストア プロパティの取得

`dpmvc datastore` コマンドは、特定の VMware vCenter データ ストアの空き領域と容量設定を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmvc datastore  
[-sc sc_url]  
-getproperty {all|capacity|freespace}  
-datacenter_name datacentername  
-datastore_name datastorename  
[-vc_server vcserver]  
[-pre]  
[-post]  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]  
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-getproperty {*all|capacity|freespace*}

(オプション) 取得するプロパティを指定します。オプションは、以下のとおりです。

all

データ ストアの容量と空き領域を取得します。

capacity

データ ストアの容量を取得します。

freespace

データ ストアの空き領域を取得します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-datastore_name *datastorename*

新しい VM が存在するデータ ストアの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: グローバル認証情報を使用して、データストアのすべてのプロパティを取得する

以下の例は、データ ストアの空き領域と容量の両方を取得します。

```
dpmvc datastore -getproperty all datacenter_name DCA/MyCity  
-datastore_name storage1 -vc_server vc_server_1
```

例: データストアのすべてのプロパティを取得する

以下の例は、データ ストアの空き領域と容量の両方を取得します。

```
dpmvc datastore -getproperty all datacenter_name DCA/MyCity  
-datastore_name storage1 -vc_server vc_server_1
```

例: データストアのすべてのプロパティを取得する

以下の例は、データ ストアの空き領域と容量の両方を取得します。

```
dpmvc datastore -getproperty all datacenter_name DCA/MyCity  
-datastore_name storage1 -vc_server vc_server_1
```

dpmvc delete コマンド -- VM の破棄

`dpmvc delete` コマンドは、電源オフ状態の VM を削除します。このコマンドを使用して、未使用のリソースをクリーンアップし、解放します。

重要: このコマンドを発行する *前に*、重要なデータをバックアップしたことを確認してください。このコマンドは、ディスク イメージを含め、VM の VM データ ストアとデータ ファイルを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc delete
[-sc sc_url]
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcservername]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user *vcuser*

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password *vcpassword*

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmtutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループホストとユーザに対して使用できるコンポーネント ID を指定します。

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、VM を破棄する

以下の例は、VM `vm11` を破棄します。

```
dpmvc delete -vm_name vm11 -datacenter_name "lab 444"
```

例: 複数 vCenter Server 環境でグローバル認証情報を使用して、VM を破棄する

以下の例は、VM `vm11` を破棄します。

```
dpmvc delete -vm_name vm11 -datacenter_name "lab 444" -vc_server vc_server_1
```

dpmvc distributedswitch コマンド -- 分散仮想スイッチの管理

dpmvc distributedswitch コマンドを使用して、分散仮想スイッチを管理できます。

- データセンターへの新しい分散仮想スイッチの追加
- 分散仮想スイッチのプロパティの更新
- 分散仮想スイッチの削除
- 分散仮想スイッチへの新しい分散ポート グループの作成
- 分散仮想スイッチのポート グループ プロパティの更新
- 分散仮想スイッチからの分散ポート グループの削除
- 分散仮想スイッチの分散ポート グループ名の変更

コマンドの形式は、以下のとおりです。

```
dpmvc distributedswitch {-vds_add | -vds_update}  
-vc_server vcservername  
[-datacenter_name datacentername]  
-vds_folder vdsfolder  
-switch_name switchname  
[-hostnics hostname1:nic1,nic2,...,nicn [;hostname1:nic2,...,nicn ]]  
[-uplink_port_names uplink1[,uplink2,...,uplinkn]]  
[-maxports maxports]  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
[-locale iso639value]
```

```
dpmvc distributedswitch -vds_remove  
-vc_server vcservername  
-switch_name switchname  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```

```
dpmvc distributedswitch -add_portgroup
-vc_server vcservername
-switch_name switchname
-portgroup_name portgroupname
[-bindtype earlyBinding | ephemeral | lateBinding]
[-vlan vlanID]
[-numports numberofports]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc distributedswitch -update_portgroup
-vc_server vcservername
-portgroup_name portgroupname
[-portgroup_newname portgroupnewname]
[-bindtype earlyBinding | ephemeral | lateBinding]
[-vlan vlanID]
[-numports numberofports]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc distributedswitch -remove_portgroup
-vc_server vcservername
-portgroup_name portgroupname
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
dpmvc distributedswitch -rename_portgroup
-vc_server vcservername
-portgroup_name portgroupname
-portgroup_newname portgroupnewname
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

```
-vc_server vcservername
```

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vds_folder *vdsfolder*

CA Virtual Assurance のエクスプローラ ペインで分散仮想スイッチのフォルダを指定します。

-switch_name *switchname*

操作を実行するスイッチの名前を指定します。

-hostnics *hostname1:nic1,nic2,...nicn* [*hostname1:nic2,...nicn*]

(オプション) ESX ホスト メンバに関連付けられた NIC のリストを指定します。

-uplink_port_names *uplink1[,uplink2,...,uplinkn]*

(オプション) 使用するアップリンク ポート名のカンマ区切りリストを指定します。

-maxports *maxports*

(オプション) ポートの最大数を指定します。

-bindtype *earlyBinding | ephemeral | lateBinding*

(オプション) ポート グループのバインド タイプを指定します。有効な値は以下のとおりです。

earlyBinding

VM がポートグループにバインドする際にポートを割り当てます。このバインディング タイプは、接続性の常時確保に有効ですが、ポートが永続的に予約されます。デフォルトはこのバインディング タイプです。

lateBinding

VM の電源がオンで NIC の状態が接続済みの場合、VM にポートを割り当てます。このバインディング タイプは、VM の電源がオフになるか、NIC が切断されると、ポートを割り当て直します。**LateBinding** は vCenter で設定できます。

ephemeral

VM の電源がオンで NIC の状態が接続済みの場合、VM にポートを割り当てます。このバインディング タイプは、VM の電源がオフになるか、NIC が切断されると、ポートを割り当て直します。**ephemeral** バインディングは、ESX ホストと vCenter で設定できます。

-numports *numberofports*

(オプション) ポート グループのポートの数を指定します。

-portgroup_name *portgroupname*

ポート グループ名を指定します。

-portgroup_newname *portgroupnewname*

新しいポート グループ名を指定します。

-vlan *vlanid*

(オプション) 仮想ポート グループの操作に使用する整数値 (*vlanid*) を指定します。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (**timeout** の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの **timeout** の値 (分) がタイムアウトするまでの時間となります。**timeout** の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: 新しい仮想分散スイッチの作成

この例では、新しい分散仮想スイッチを作成します。

```
dpmvc distributedswitch -vds_add -vc_server vc5master -datacenter_name dc3  
-vds_folder vds -switch_name vds1
```

例: 既存の仮想分散スイッチの更新

この例では、分散仮想スイッチを更新します。MYSERVER1 用の NIC を指定し、分散仮想スイッチから MYSERVER2 を削除します。分散仮想スイッチからホストを削除するときは、NIC を割り当てずにサーバ名を指定します (MYSERVER2 : のように、サーバ名の後ろにコロンを付けます)。

```
dpmvc distributedswitch -vds_update -vc_server VAS-VC5 -datacenter_name VC5  
-switch_name vdistSwitch -hostnics MYSERVER1:nxmgt2,tmp2;MYSERVER2:  
-ws_user admin -ws_password ca_admin -post
```

このコマンドによって `vdistSwitch` が以下のように更新されます。

- ホスト MYSERVER1 は NIC `nxmgt2` および `tmp2` を使用します。
- ホスト MYSERVER2 はスイッチから削除されます。

dpmvc entermaintenancemode コマンド -- 保守モードに設定する

dpmvc entermaintenancemode コマンドは、ESX ホストを保守モードに設定します。

このコマンドの形式は、以下のとおりです。

```
dpmvc entermaintenancemode
-esx_host_name esxhostname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-timeout timeout
[-evacuate_powered_off_vm {no | yes}]
[-locale iso639value]
```

-esx_host_name esxhostname

VM が存在する VMware ESX サーバを指定します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-timeout *timeout*

保守モードに入る、または保守モードを終了するまでのタイムアウト時間を指定します。

-evacuate_powered_off_vm {no | yes}

(オプション) VM 停止後に終了するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 保守モードを開始する

以下の例は、ESX ホストを保守モードに設定します。

```
dpmvc entermaintenancemode -esx_host_name esx1 -vc_server myvcserver -timeout 60
```

dpmvc exitmaintenancemode コマンド -- 保守モードの終了

dpmvc exitmaintenancemode コマンドは、ESX ホストの保守モードを終了します。

このコマンドの形式は、以下のとおりです。

```
dpmvc exitmaintenancemode
-esx_host_name esxhostname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-timeout timeout
[-locale iso639value]
```

-esx_host_name esxhostname

VM が存在する VMware ESX サーバを指定します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例：https://hostname:port/...

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例：localhost.ca.com

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例：80

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (**timeout** の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの **timeout** の値 (分) がタイムアウトするまでの時間となります。**timeout** の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-timeout *timeout*

保守モードに入る、または保守モードを終了するまでのタイムアウト時間を指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 保守モードを終了する

以下の例は、ESX ホスト **esx1** の保守モードを終了します。

```
dpmvc exitmaintenancemode -esx_host_name esx1 -vc_server myvcserver
-timout 60
```

dpmvc faulttolerance コマンド -- フォールトトレラント操作の指定

dpmvc faulttolerance コマンドは、フォールトトレランスのオン、オフ、有効化、無効化、テスト、またはセカンダリ VM のマイグレーションを行います。

このコマンドの形式は、以下のとおりです。

```
dpmvc faulttolerance
{-turn_on|-turn_off|-enable|-disable|-test_ft|-migrate_secondary }
[-datacenter_name datacentername]
-vm_name vmname
[-esx_host_name esxhostname]
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
[-locale iso639value]
```

-turn_on

指定された VM のフォールトトレランスをオンにします。

(オプション) セカンダリ VM の ESX ホスト名を指定します。

-turn_off

指定された VM のフォールトトレランスをオフにします。

注: この操作によってセカンダリ VM が削除されます。

-enable

指定された VM のフォールトトレランスを有効にします。

-disable

指定された VM のフォールトトレランスを無効にします。

-test_ft

指定された VM のフォールトトレランスをテストします。

-migrate_secondary

フォールトトレランスのセカンダリ VM を別の ESX サーバにマイグレートします。 **-esx_host_name** オプションは必須です。

-vm_name vmname

VM を指定します。

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-esx_host_name *esxhostname*

(オプション) VM が存在する VMware ESX サーバを指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は `caimgconf.cfg` ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmvc getavailablescsidisks コマンド -- 利用可能な SCSI ディスクの取得

dpmvc getavailablescsidisks コマンドは、利用可能な SCSI ディスクを一覧表示します。

このコマンドの形式は、以下のとおりです。

```
dpmvc getavailablescsidisks  
[-datastore_name datastorename]  
-esx_host_name esxhostname  
-vc_server vcservername  
[-ws_user wsuser]  
[-ws_password wspassword]  
[-prompt {yes|no}]  
[-sc sc_url]  
[-locale iso639value]
```

-datastore_name *datastorename*

新しい VM が存在するデータ ストアの名前を指定します。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : 80

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmvc gethosthba コマンド -- ホスト バス アダプタの取得

dpmvc gethosthba コマンドは、指定された **ESX** ホストに設定されたホストバス アダプタのリストを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmvc gethosthba
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-sc sc_url]
[-locale iso639value]

-esx_host_name esxhostname
```

VM が存在する VMware ESX サーバを指定します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user wsuser

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password wspassword

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

`-sc sc_url`

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmvc getresources コマンド -- VM リソースの取得

`dpmvc getresources` コマンドは、CPU またはメモリの、共有制限または予約を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmvc getresources
[-sc sc_url]
-vm_name vmname
-datacenter_name datacentername
-resource {all|cpulimit|cpureserv|memlimit|memresrv}
[-vc_server vcserver]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-vm_name *vmname*

VM を指定します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-resource {*all*|*vm_cpulimit*|*vm_cpureserv*|*vm_memlimit*|*vm_memresrv*}

特定のリソースを取得するのか、すべてのリソースを取得するのかを指定します。オプションは、以下のとおりです。

all

VM のすべての CPU、メモリ共有制限、予約を取得します。

vm_cpulimit

VM の CPU 共有の数の制限を取得します。

vm_cpureserv

VM の CPU 予約の共有の数を取得します。

vm_memlimit

VM のメモリ共有の数の制限を取得します。

vm_memresrv

VM のメモリ予約の共有の数を取得します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM のすべてのリソースを取得する

以下の例は、**usa-vm2** のリソースをすべて取得します。

```
dpmvc getresources -resource all -datacenter_name MyCity -vm_name usa-vm2
```

例: VM の CPU 制限を取得する

以下の例は、VM **uss-vm3** の個別リソース **cpulimit** を取得します。

```
dpmvc getresources -resource cpulimit -datacenter_name MyCity  
-vm_name uss-vm3 -vc_server vc_server_1
```

dpmvc getshares コマンド -- VM 共有の取得

dpmvc getshares コマンドでは、VM に割り当てられている CPU またはメモリの共有の数を確認できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc getshares
[-sc sc_url] {-cpu|-memory}
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcserver]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-cpu|-memory

VM に割り当てられている CPU またはメモリの共有の数を表示します。共有を使用すると、利用できる物理リソースが、稼働中のほかの VM より比例的に多く割り当てられている VM を判断できます。VM により多くの共有を割り当てると、より多くの物理リソースがその VM に割り当てられます。VM に割り当てる共有を減らすと、その VM に割り当てられる物理リソースも減ります。

-cpu

CPU 値を取得することを指定します。次のパラメータと共に使用しないでください。

-memory**-memory**

メモリ値を取得することを指定します。次のパラメータと共に使用しないでください。

-cpu**-datacenter_name *datacentername***

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM の CPU 値を取得する

以下の例は、**usa-vm2** の CPU 値を取得します。

```
dpmvc getshares -resource all -datacenter_name MyCity -vm_name usa-vm2  
-vc_server vc_server_1
```

例: 単一 vCenter Server 環境の VM の CPU 値を取得する

以下の例は、**usa-vm2** の CPU 値を取得します。

```
dpmvc getshares -resource all -datacenter_name MyCity -vm_name usa-vm2
```

例: 複数 vCenter Server 環境でグローバル認証情報を使用して、VM のメモリ値を取得する

以下の例は、VM **uss-vm3** の個別リソース **cpulimit** を取得して表示します。

```
dpmvc getshares -resource cpulimit -datacenter_name MyCity -vm_name uss-vm3  
-vc_server vc_server_1
```


dpmvc getsnapshots コマンド -- VM スナップショットの取得

dpmvc getsnapshots コマンドは、VM のスナップショットをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmvc getsnapshots
[-sc sc_url]
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcserver]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: スナップショットをリスト表示する

以下の例は、VM のスナップショットをリスト表示します。

```
dpmvc getsnapshots -vc_server my_server1 -datacenter_name my_dc  
-vm_name my_vm
```

dpmvc imgjobcheck コマンド -- クローン作成ジョブ ステータスの取得

dpmvc imgjobcheck コマンドは、特定の CA Virtual Assurance ジョブ ID または特定の VC タスク ID のクローン作成ジョブ ステータスを取得します。ジョブ ID は、特定の dpmvc コマンドで使用します。このコマンドを使用して、VC のステータスを確認します。

このコマンドの形式は、以下のとおりです。

```
dpmvc imgjobcheck
[-sc sc_url]
-status {job ID|vc task ID}
[-vc_server vcserver]
[-vc_user vcuser]
[-vc_password vcpass]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspass]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-status *jobID*

ジョブ ステータスの取得に使用するジョブ ID を定義します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user vcuser

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password vcpassword

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file authorizationfilename

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp componentID

(オプション) グループホストとユーザに対して使用できるコンポーネント ID を指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user wsuser

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password wspassword

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 単一 vCenter Server 環境でジョブ ID を使用して、クローン作成ジョブ ステータスを取得する

以下の例は、CA Virtual Assurance ジョブ ID を使用してクローニング ジョブ ステータスを取得します。

```
dpmvc imgjobcheck -status 42
```

例: 単一 vCenter Server 環境で VC タスク ID を使用して、クローン作成ジョブ ステータスを取得する

以下の例は、VC タスク ID を使用してクローニング ジョブ ステータスを取得します。

```
dpmvc imgjobcheck -status task -21099
```

dpmvc migrate コマンド -- VM のマイグレート

dpmvc migrate コマンドは、あるホスト サーバから別のホスト サーバに VM をマイグレートします。

このコマンドの形式は、以下のとおりです。

```
dpmvc migrate
[-sc sc_url]
-datacenter_name datacentername
-vm_name vm -migrate_to_target_host name
-migrate_to_target_resource_pool_name name
-migrate_to_compute_resource_name name
[-vc_server vcserver]
[-vc_user vcuser]
[-vc_password vcpasssword]
[-auth_file authorizationfilename]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspasssword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例： `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例： `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例： `80`

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-migrate_to_target_host *name*

VM をマイグレートするホストの名前を指定します。

-migrate_to_target_resource_pool_name *name*

ターゲット リソース プールを指定します。

-migrate_to_compute_resource_name *name*

VM をマイグレートするクラスタまたは VMware ESX ホストの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user *vcuser*

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password *vcpassword*

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmtutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: デフォルトの認証ファイルを使用して、VM をマイグレートする

以下の例は、データ センター DCA/MyCity から ServerC-dcavc に VM MyVM1 をマイグレートします。

```
dpmvc migrate -vm_name MyVM1 -datacenter_name DCA/MyCity
-migrate_to_compute_resource_name DCACluster/ClusterA
-migrate_to_target_resource_pool_name Resources/DCTest
-migrate_to_target_host ServerC-dcavc.MyCompany.com
-vm_server VM_Server1 -vc_user VCAdmin1
```

dpmvc removedatastore コマンド -- データストアの削除

dpmvc removedatastore コマンドは、データストアを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc removedatastore
-datastore_name datastorename
-esx_host_name esxhostname
-vc_server vcservername
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-sc sc_url]
[-locale iso639value]

-datastore_name datastorename
```

新しい VM が存在するデータ ストアの名前を指定します。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : 80

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmvc removevmvdisk コマンド -- 仮想ディスクの削除

dpmvc removevmvdisk コマンドは、VM から仮想ディスクを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc removevmvdisk
[-datacenter_name datacentername]
-vm_name vmname
-vc_server vcservername
[-sc sc_url]
[-wait timeout]
[-pre]
[-post]
-disk_device_key diskdevicekey
-delete_data {yes | no}
[-locale iso639value]

-datacenter_name datacentername
```

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-disk_device_key *diskdevicekey*

ディスクのデバイス キーを整数で指定します (例: 2000)。

-delete_data {yes | no}

ディスクを削除する際にデータも削除するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 仮想ディスクを削除する

以下の例は、VM **myvm** からデバイス キー 2000 の仮想ディスクを削除します。

```
dpmvc removevmdisk -vm_name myvm -vc_server myvcserver -disk_device_key 2000  
-delete_data yes
```

dpmvc removevmvnic コマンド -- 仮想 NIC の削除

dpmvc removevmvnic コマンドは、VM から仮想 NIC を削除します。

このコマンドの形式は、以下のとおりです。

```
dpmvc removevmvnic  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
-device_key devicekey  
[-locale iso639value]
```

-datacenter_name *datacentername*

(オプション) VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (**timeout** の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの **timeout** の値 (分) がタイムアウトするまでの時間となります。**timeout** の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-device_key *devicekey*

ネットワーク インターフェースのデバイス キーを整数で指定します (例: 3000)。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: vNIC を削除する

以下の例は、VM から vNIC を削除します。

```
dpmvc removevmvnic -vm_name myvm -vc_server myvcserver -device_key 3000
```

dpmvc setcpu コマンド -- VM の CPU 設定

dpmvc setcpu コマンドは、VM の CPU の数を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmvc setcpu  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-set_cpu number  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
[-locale iso639value]
```

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-set_cpu *number*

(オプション) クローン作成する VM の CPU の数を定義します。この値は、テンプレートで指定された値よりも優先されます。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例：https://hostname:port/...

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例：localhost.ca.com

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例：80

-wait timeout

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (timeout の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの timeout の値 (分) がタイムアウトするまでの時間となります。timeout の値を指定せずに wait オプションを入力すると、CLI は caimgconf.cfg ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほかに、timeout には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: VM の CPU を設定する

以下の例は、VM myvm の CPU 数 4 を設定します。

```
dpmvc setcpu -vm_name myvm -vc_server myvcserver -set_cpu 4
```


dpmvc setmemory コマンド -- VM のメモリ サイズの設定

dpmvc setmemory コマンドは、VM のメモリ サイズを設定します。

このコマンドの形式は、以下のとおりです。

```
dpmvc setmemory  
[-datacenter_name datacentername]  
-vm_name vmname  
-vc_server vcservername  
-set_memory number  
[-sc sc_url]  
[-wait timeout]  
[-pre]  
[-post]  
[-locale iso639value]
```

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-set_memory *number*

(オプション) クローン作成する VM のメモリの量をメガバイト (MB) で定義します。この値は、テンプレートで指定された値よりも優先されます。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。 *timeout* の値を指定せずに *wait* オプションを入力すると、CLI は `caimgconf.cfg` ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: VM のメモリ サイズを設定する

以下の例は、VM **myvm** のメモリ サイズを **4 GB** に設定します。

```
dpmvc setmemory -vm_name myvm -vc_server myvcserver -set_memory 4000
```

dpmvc setresources コマンド -- VM リソースの設定

dpmvc setresources コマンドでは、CPU またはメモリの共有制限または予約を調整できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc setresources
[-sc sc_url] {-setcpulimit|-setcpureserve|-setmemlimit|-setmemresrv} value
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcservername]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : 80

setcpulimit

VM の CPU 共有の数の制限を定義します。

setcpureserv

VM の CPU 予約の共有の数を定義します。

setmemlimit

VM のメモリ共有の数の制限を定義します。

setmemresrv

VM のメモリ予約の共有の数を定義します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user *vcuser*

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password vcpasssword

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file authorizationfilename

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp componentID

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-wait timeout

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (timeout の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの timeout の値 (分) がタイムアウトするまでの時間となります。timeout の値を指定せずに wait オプションを入力すると、CLI は caimgconf.cfg ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、timeout には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、リソースの CPU 制限を 1000 共有に設定する

以下の例は、仮想マシン **MyVM1** の CPU 共有制限を 1000 に調整します。

```
dpmvc setresources -setcpulimit 1000 -vm_name MyVM1  
-datacenter_name DCA/MyCity
```

例: 特定のコンポーネント ID を使用して、リソースの CPU 制限を 1000 に設定する

以下の例は、仮想マシン **MyVM1** の CPU 共有制限を 1000 に調整します。

```
dpmvc setresources -setcpulimit 1000 -vm_name MyVM1 -datacenter_name DCA/MyCity  
-vc_server vc_server_1 -vc_user VCAAdmin1 -auth_comp Imaging
```

dpmvc setshares コマンド -- VM 共有の設定

dpmvc setshares コマンドでは、VM に割り当てられている CPU またはメモリの共有の数を設定できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc setshares
[-sc sc_url] {-cpu_add_prop_value value|-cpu subtract_prop_value
value|-cpu_overwrite_prop_value value|-mem_add_prop_value
value|-mem_subtract_prop_value value|-mem_overwrite_prop_value value}
-datacenter_name datacentername
-vm_name vmname
[-vc_server vcservername]
[-vc_user vcuser]
[-vc_password vcpasssword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-wait [timeout]][-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspasssword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]

-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例：https://*hostname*:*port*/...

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例：localhost.ca.com

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例：80

`-cpu_add_prop_value value|-cpu subtract prop value
value|-cpu_overwrite_prop_value value|-mem_add_prop_value
value|-mem_subtract_prop_value value|-mem_overwrite_prop_value value`

追加、削除、変更する CPU またはメモリの共有の数を指定します。共有を使用すると、利用できる物理リソースが、稼働中のほかの VM より比例的に多く割り当てられている VM を判断できます。VM により多くの共有を割り当てると、より多くの物理リソースがその VM に割り当てられます。VM に割り当てる共有を減らすと、その VM に割り当てられる物理リソースも減ります。

`-cpu_add_prop_value value`

VM に追加する CPU 共有の数を指定します。

`-cpu_subtract_prop_value value`

VM から削除する CPU 共有の数を指定します。

`-cpu_overwrite_prop_value value`

VM で使用する CPU 共有の数を指定します。

`-mem_add_prop_value value`

VM に追加するメモリ共有の数を指定します。

`-mem_subtract_prop_value value`

VM から削除するメモリ共有の数を指定します。

`-mem_overwrite_prop_value value`

VM で使用する CPU 共有の数を指定します。

`-datacenter_name datacentername`

VM が存在するデータ センターを指定します。

`-vm_name vmname`

VM を指定します。

`-vc_server vcservername`

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

`-vc_user vcuser`

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password vcpasssword

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file authorizationfilename

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp componentID

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-wait timeout

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (timeout の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの timeout の値 (分) がタイムアウトするまでの時間となります。timeout の値を指定せずに wait オプションを入力すると、CLI は caimgconf.cfg ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、timeout には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、VM の CPU 共有の数を増やす

以下の例は、データ センター DCA/MyCity 上の仮想マシン MyVM01 の CPU 共有を 1000 増やします。

```
dpmvc setshares -vm_name MyVM01 -datacenter_name DCA/MyCity  
-cpu_add_prop_value 1000
```

例: 複数 vCenter Server 環境でグローバル認証情報を使用して、VM の CPU 共有の数を増やす

以下の例は、データ センター DCA/MyCity 上の仮想マシン MyVM01 の CPU 共有を 1000 増やします。

```
dpmvc setshares -vm_name MyVM01 -datacenter_name DCA/MyCity  
-cpu_add_prop_value 1000 -vc_server vc_server_1
```

例: 指定した認証ファイルとコンポーネント ID を使用して、VM の CPU 共有を上書きする

以下の例は、データ センター DCA/MyCity 上の仮想マシン MyVM01 の CPU 共有を 1200 に設定します。

```
dpmvc setshares -vm_name MyVM01 -datacenter_name DCA/MyCity  
-cpu_overwrite_prop_value 1200 -vc_server vc_server_1  
-vc_user VCAdmin1 -auth_file c:\localauth.dat -auth_comp Imaging
```

dpmvc snapshot コマンド -- スナップショットを管理する

dpmvc snapshot コマンドでは、VM のスナップショットを管理できます。このコマンドを使用して、スナップショットの作成、1 つまたはすべてのスナップショットの削除、スナップショットへの復元を実行できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc snapshot
[-sc sc_url]
-operation create
-datacenter_name datacentername -vm_name vmname
-vm_snapshot_name vmsnapshotname
[-vm_snapshot_desc description]
[-withmemory {yes|no}]
[-quiesce {yes|no}][-poweron]
[-vc_server vcservername]
[-vc_user vcusername]
[-vc_password vcpassword]
[-auth_file authorizationfile]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

```
dpmvc snapshot
[-sc sc_url]
-operation remove
-datacenter_name datacentername -vm_name vmname
[-vm_snapshot_name vmsnapshotname]
[-vm_snapshot_id snapid]
[-withchildren {yes|no}]
[-vc_server vcservername]
[-vc_user vcusername]
[-vc_password vcpassword]
[-auth_file authorizationfile]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

```
dpmvc snapshot
[-sc sc_url]
-operation revert
-datacenter_name datacentername -vm_name vmname
-vm_snapshot_name vmsnapshotname
[-vm_snapshot_id snapid]
[-vc_server vcservername]
[-vc_user vcusername]
[-vc_password vcpasssword]
[-auth_file authorizationfile]
[-auth_comp componentID] [-wait [timeout]]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspasssword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-operation {create|remove|revert}

VM スナップショットを作成、削除、または復元するように指定します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vm_name *vmname*

VM を指定します。

-vm_snapshot_name *vmsnapshotname*

VM スナップショットの名前を定義します。

-vm_snapshot_desc *description*

(オプション) VM スナップショットの説明を定義します。

-withmemory {true | false}

(オプション) スナップショットにメモリが含まれていることを指定します。このパラメータは、**-remove** または **-revert** には無効です。

-quiesce {yes|no}

(オプション) **yes** に設定すると、スナップショットを作成する際に仮想マシンの電源がオンの場合、VMware ツールは仮想マシン内のファイルシステムを休止します。これにより、ディスク スナップショットに一貫した状態のゲスト ファイルシステムが記録されます。仮想マシンの電源がオフになっているか、VMware ツールが利用できない場合、コマンドは **quiesce** フラグを無視します。

-vm_snapshot_id *snapid*

(オプション) スナップショット ID を定義します。

-withchildren

(オプション) スナップショットのすべての子を削除するように指定します。このパラメータは、**-create** または **-revert** には無効です。

-poweron

(オプション) VM の電源をオンにするように指定します。このパラメータは、以下のパラメータに対しては無効です。

-remove または

-revert

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user *vcuser*

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password vcpasssword

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file authorizationfilename

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp componentID

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-wait timeout

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (timeout の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの timeout の値 (分) がタイムアウトするまでの時間となります。timeout の値を指定せずに wait オプションを入力すると、CLI は caimgconf.cfg ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、timeout には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、VM スナップショットを作成する

この例では、VM システム **MyVM01** で **Hello world** という名前のスナップショットを、「**First snapshot**」という説明で作成します。

```
dpmvc snapshot -create -vm_snapshot_name "Hello world" -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity
```

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、VM スナップショットを復元する

この例では、**MyVM01** を **Hello world** という名前のスナップショットに復元します。

```
dpmvc snapshot -revert -vm_snapshot_name "Hello world" -vm_name MyVM01
-datacenter_name VAS/MyCity
```


例: 単一 vCenter Server 環境でグローバル認証情報を使用して、VM スナップショットを削除する

この例では、VM システム MyVM01 から Hello world という名前のスナップショットを削除します。

```
dpmvc snapshot -remove -vm_snapshot_name "Hello world" -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity
```

例: 複数 vCenter Server 環境でグローバル認証情報を使用して、VM のスナップショットを作成する

この例では、VM システム PayrollVM 上で PayrollSnapshot という名前のスナップショットを作成します。

```
dpmvc snapshot -create -vm_snapshot_name PayrollSnapshot -vm_snapshot_desc "First snapshot"
-vm_name MyVM01 -datacenter_name VAS/MyCity -vc_server vc_server_1
```

例: 指定した認証ファイルとコンポーネント ID を使用して、VM スナップショットを削除する

この例では、データ センター DCA/MyCity 上の仮想マシン MyVM01 の CPU 共有を 1200 に設定します。

```
dpmvc snapshot -remove -vm_snapshot_name "Snapshot 1" -vm_name MyVM01
-datacenter_name VAS/MyCity -vc_server vc_server_1 -vc_user VCAdmin1
-auth_file c:\localauth.dat -auth_comp Imaging
```

dpmvc templatetovm コマンド -- テンプレートを VM に変換する

dpmvc templatetovm コマンドは、仮想マシン テンプレートを VM に変換します。この機能を使用すると、たとえば、変換済みの VM にパッチやソフトウェア更新プログラムを適用してから、テンプレートに変換できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc templatetovm
[-sc sc_url]
-template_name templatename
-datacenter_name datacentername
-compute_resource_name hostname
-resource_pool_name resourcepool
-esx_host_name hostname
[-vc_server vmserver]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例： `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例： `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例： `80`

-template_name *templatename*

テンプレートの名前を定義します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-compute_resource_name *name*

VM を作成するクラスタまたは VMware ESX ホストの名前を指定します。

-resource_pool_name *resourcepool*

クローンを作成する VM の選択元のリソース プールの名前を指定します。

-esx_host_name *esxhostname*

VM が存在する VMware ESX サーバを指定します。

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user *vcuser*

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password *vcpassword*

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password *wspassword*

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザパスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、テンプレートを VM に変換する

以下の例は、**convertTest** という名前のテンプレートを VM に変換します。

```
dpmvc templatetovm -datacenter_name VAS/MyCity -template_name convertTest  
-compute_resource_name onDemand/myhost.myco.com  
-esx_host_name myhost.myco.com -resource_pool_name Resources/QA
```

例: 複数 vCenter Server 環境でグローバル認証情報を使用して、テンプレートを VM に変換する

以下の例は、`payrollSystem1` という名前のテンプレートを VM に変換します。

```
dpmvc templatetovm -datacenter_name VAS/MyCity -template_name payrollSystem1  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA -vc_server vc_server_1
```

例: 指定した認証ファイルとコンポーネント ID を使用して、テンプレートを VM に変換する

以下の例は、`Win2K3` という名前のテンプレートを VM に変換します。

```
dpmvc templatetovm -datacenter_name VAS/MyCity -template_name Win2K3  
-compute_resource_name onDemand/myhost.myco.com  
-esx_host_name myhost.myco.com -resource_pool_name Resources/QA  
-vc_server vc_server_1 -vc_user VCAdmin1 -auth_file c:\¥localauth.dat  
-auth_comp Imaging
```

dpmvc vapp コマンド -- vApp の管理

dpmvc vapp コマンドは、vApps 上の以下の操作をサポートします。

- vApp の新規作成
- クローン
- 電源オン
- 電源オフ
- 中断
- VMware vCenter からの v App の削除
- VMware vCenter からの登録解除
- vApp への VM の追加
- vApp へのリソースプールの追加
- vApp への vApp の追加
- vApp 設定の更新

コマンドの形式は、以下のとおりです。

```
dpmvc vapp -create
-vc_server vcservername
-vapp_path vapppath
[-vapp_name vappname]
[-cpu_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
[-mem_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]
[-locale iso639value]
```

```
dpmvc vapp -clone
-vc_server vcservername
-vapp_path vapppath
[-target_vapp_path targetvapppath]
[-vapp_name vappname]
[-target_vapp_host targetvapphost]
```

```
dpmvc vapp [-power_on | -power_off | -suspend | -delete | -unregister ]
-vc_server vcservername
-vapp_path vapppath
```

```
dpmvc vapp -add_vms_to_vapp
-vc_server vcservername
-vapp_path vapppath
[-vapp_name vappname]
[-vms vms]
```

```
dpmvc vapp -add_rps_to_vapp  
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-rps resourcepool]
```

```
dpmvc vapp -add_vapps_to_vapp  
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-vapps vapps]  
[-locale iso639value]
```

```
dpmvc vapp -update_vapp_config  
-vc_server vcservername  
-vapp_path vapppath  
[-vapp_name vappname]  
[-config_settings  
vappName,objecType,startAction,startDelay,startOrder,stopAction,stopDelay,waitFor  
Guest]  
[-cpu_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]  
[-mem_allocation isExpandableReservation, limit, reservation, sharesLevel, shares]  
[-locale iso639value]
```

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vapp_path vapppath

vApp パスを指定します。

-vapp_name vappname

(オプション) vApp 名を指定します。

-cpu_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

(オプション) vApp 設定を更新する際の CPU 割り当て設定を指定します。

isExpandableReservation

割り当てが拡張可能かどうか指定します。可能な値: **true** または **false**

limit

割り当ての制限を指定します。

reservation

予約の割り当てを指定します。

sharesLevel

共有レベルを指定します。0 は低、1 は通常、2 は高、3 はカスタムです。例: **true,-1,1000,1,4000**

shares

割り当ての共有を指定します。

-mem_allocation *isExpandableReservation, limit, reservation, sharesLevel, shares*

vApp 設定を更新する際のメモリ割り当て設定を指定します。

isExpandableReservation

割り当てが拡張可能かどうか指定します。可能な値: **true** または **false**

limit

割り当ての制限を指定します。

reservation

予約の割り当てを指定します。

sharesLevel

共有レベルを指定します。0 は低、1 は通常、2 は高、3 はカスタムです。例: **true,-1,1000,1,4000**

shares

割り当ての共有を指定します。

-vms vms

(オプション) vApp に追加する VM を指定します。

-rps resourcepool

(オプション) vApp に追加するリソース プールの名前を指定します。

-vapps vapps

(オプション) vApp に追加する vApps を指定します。

-config_settings

vappName,objecType,startAction,startDelay,startOrder,stopAction,stopDelay,waitForGuest

(オプション) vApp の構成設定を指定します。適切なフォーマットは以下のとおりです。

startAction none/powerOn

開始のアクションを指定します。利用可能なオプションは none または powerOn です。

stopAction none/powerOff/guestShutdown/suspend

開始のアクションを指定します。利用可能なオプションは none、powerOff、guestShutdown、または guestShutdown です。

-target_vapp_path targetvapppath

作成する vApp のターゲット vApp パスを指定します。

-target_vapp_host targetvapphost

(オプション) 新しい vApp が存在すべきホストを指定します。

-target_vapp_datastore targetvappdatastore

(オプション) vApp が存在すべきデータ ストアの名前を指定します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

dpmvc virtualswitch コマンド -- 仮想スイッチの管理

dpmvc virtualswitch コマンドを使用して、仮想スイッチを管理できます。

- 新しい仮想スイッチの作成
- 仮想スイッチのプロパティの更新
- 仮想スイッチの削除
- 仮想スイッチの新しいポート グループの作成
- 仮想スイッチのポート グループ プロパティの更新
- 仮想スイッチからのポート グループの削除
- 仮想スイッチのポート グループの名前変更

注: この操作は非同期で実行されず、結果はただちに返されます。ただし、PMM はこの操作をタスク化された操作として扱います。そのため、応答にはタスク ID が含まれますが、それは常に空の文字列 ("") です。

コマンドの形式は、以下のとおりです。

```
dpmvc virtualswitch {-vs_add | -vs_update}  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
-nic_names nicname1[,nicname2, ...]  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```

```
dpmvc virtualswitch -vs_remove  
-vc_server vcservername  
-esx_host_name esxhostname  
-switch_name switchname  
[-sc sc_url]  
[-wait [timeout]]  
[-pre]  
[-post]  
[-locale iso639value]
```

```
dpmvc virtualswitch -add_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-portgroup_name portgroupname
[-vlan vlanID]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]

dpmvc virtualswitch -update_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-portgroup_name portgroupname
[-portgroup_newname portgroupnewname]
[-vlan vlanID]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]

dpmvc virtualswitch -remove_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-portgroup_name portgroupname
[-vlan vlanID]
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]

dpmvc virtualswitch -rename_portgroup
-vc_server vcservername
-esx_host_name esxhostname
-switch_name switchname
-portgroup_name portgroupname
-portgroup_newname portgroupnewname
[-sc sc_url]
[-wait [timeout]]
[-pre]
[-post]
[-locale iso639value]
```

-vc_server *vcservername*

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-esx_host_name *esxhostname*

VM をホストする ESX サーバの名前を指定します。

-switch_name *switchname*

操作を実行するスイッチの名前を指定します。

-nic_names *nicname1 [nicname2, ...]*

物理 NIC のリストをカンマで区切って指定します。複数の NIC を指定する場合は、二重引用符を使用して引数をエスケープします（例："sc1,nfs1,mnic1,mnic2"）。

-portgroup_name *portgroupname*

ポート グループ名を指定します。

-portgroup_newname *portgroupnewname*

新しいポート グループ名を指定します。

-vlan *vlanID*

(オプション) 仮想ポート グループの操作に使用する整数値 (VLAN ID) を指定します。

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (**timeout** の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの **timeout** の値 (分) がタイムアウトするまでの時間となります。**timeout** の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほか、**timeout** には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例

この例では、新しい仮想スイッチを作成します。

```
dpmvc virtualswitch -vs_add -vc_server vc5master -esx_host_name esx5
-switch_name switch1 -nic_names "sc1,nfs1,mnic1,mnic2"
```

dpmvc vmtotemplate コマンド -- VM をテンプレートに変換する

dpmvc vmtotemplate コマンドは、仮想マシンをテンプレートに変換します。この機能により、データ センターの複数のサーバ間の整合性を強化できます。

このコマンドの形式は、以下のとおりです。

```
dpmvc vmtotemplate
[-sc sc_url]
-vm_name vmname
-datacenter_name datacentername
[-vc_server vmserver]
[-vc_user vcuser]
[-vc_password vcpassword]
[-auth_file authorizationfilename]
[-auth_comp componentID]
[-pre]
[-post]
[-ws_user wsuser]
[-ws_password wspassword]
[-prompt {yes|no}]
[-encrypted_password {yes|no}]
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-vm_name *vmname*

VM を指定します。

-datacenter_name *datacentername*

VM が存在するデータ センターを指定します。

-vc_server vcservername

複数の vCenter Server がある場合にアクセスする vCenter Server を指定します。単一の vCenter Server 環境の場合はオプションです。

-vc_user vcuser

(オプション) vCenter Server のユーザ ID を指定します。グローバル認証情報が有効な場合はオプションです。

-vc_password vcpassword

(オプション) vCenter Server のユーザ ID のパスワードを指定します。このオプションを省略すると、デフォルトの認証ファイルまたは特定の認証ファイル (-auth_file オプション) が指定されている場合は、ユーザ (-vc_user オプション) の暗号化されたパスワードとコンポーネント (-auth_comp オプション) が取得されます。

-auth_file authorizationfilename

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、dpmutil 認証情報設定コマンドを使用して生成した CA Virtual Assurance 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp componentID

(オプション) グループホストとユーザに対して使用できるコンポーネント ID を指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user wsuser

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスのユーザ名を指定します。

-ws_password wspassword

(オプション) VC プラットフォーム管理モジュール (PMM) に接続する Web サービスユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-encrypted_password {yes|no}

(オプション) VC ユーザ パスワードを暗号化するかどうかを指定します。

デフォルト : Yes

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 単一 vCenter Server 環境でグローバル認証情報を使用して、VM をテンプレートに変換する

以下の例は、VM を **convertTest** という名前のテンプレートに変換します。

```
dpmvc vmtotemplate -datacenter_name VAS/MyCity -template_name convertTest  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA
```

例: 複数 vCenter Server 環境でグローバル認証情報を使用して、VM をテンプレートに変換する

以下の例は、**payrollSystem1** という名前のテンプレートを VM に変換します。

```
dpmvc vmtotemplate -datacenter_name VAS/MyCity -template_name payrollSystem1  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA -vc_server vc_server_1
```

例: 指定した認証ファイルとコンポーネント ID を使用して、VM をテンプレートに変換する

以下の例は、**Win2K3** という名前のテンプレートを VM に変換します。

```
dpmvc vmtotemplate -datacenter_name VAS/MyCity -template_name Win2K3  
-compute_resource_name onDemand/myhost.myco.com -esx_host_name myhost.myco.com  
-resource_pool_name Resources/QA -vc_server vc_server_1 -vc_user VCAdmin1  
-auth_file c:\localauth.dat -auth_comp Imaging
```


dpmovf import コマンド -- OVF パッケージのインポート

dpmovf import コマンドは OVF パッケージをインポートし、VM または vApps を作成します。-properties 属性を使用すると、カスタム プロパティ ファイルを提供できます。カスタム プロパティ ファイルを使用すると、OVF パッケージで定義されているカスタム プロパティを指定できます。カスタム プロパティ ファイルには、プロパティ キーおよび対応するプロパティ値のリストが含まれます。

注: カスタム プロパティ ファイルがない場合は、properties.txt ファイルが作業ディレクトリに作成されます。デフォルトディレクトリは CA¥ProductName¥bin です。

このコマンドの形式は、以下のとおりです。

```
dpmovf import
-host vCenter_server
-user user_name
-password user_password
-name VM_VApp_name
-path OVF_file_path
-datacenter data_center
-datastore data_store
-resourcepool resource_pool
[-locale iso639value]
[-properties properties_file]
-host vCenter_server
```

vCenter Server ホストの名前を指定します。

-user user_name

ログインするユーザ名を指定します。

-password user_passsword

ログインするユーザのパスワードを指定します。

-name VM_VApp_name

VM または vApp の名前を指定します。

-path OVF_file_path

OVF ファイルパスを指定します。

-datacenter data_center

データ センター名を指定します。

-datastore *data_store*

データ ストアを指定します。

-resourcepool *resource_pool*

リソース プールを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-properties *properties_file*

(オプション) カスタム プロパティ ファイルパスを指定します。

例: CA Virtual Assurance を使用した、CA プラットフォーム用の OVF ファイルのインポート

この例では、CA プラットフォーム OVF パッケージをインポートし、vApp および VM を作成します。CA プラットフォーム OVF ファイルは **CA Platform_v1_0_0_92c.ovf**、ファイルの場所は **D:¥OVF¥CA_Platform** です。ユーザ名は **user123** です。vApp の以下の属性が指定されています: **my_datastore**、**my_datacenter**、および **my_resourcepool**。カスタム プロパティは **custom_properties.txt** ファイルで提供されます。

```
dpmovf import -path "D:¥OVF¥CA_Platform¥CA Platform_v1_0_0_92c.ovf" -name
"My_CA_Platform" -host my_host.company.com -user user123 -locale en-US -datastore
"my_datastore" -datacenter "my_datacenter" -resourcepool "my_resourcepool"
-properties "custom_properties.txt"
```

管理コマンド ライン ユーティリティ

dpmkpdb Command-- パフォーマンス データのインポートまたはエクスポート

dpmkpdb.exe は、パフォーマンス データをインポートおよびエクスポートするためのコマンドライン ユーティリティです。このユーティリティでは、CA Virtual Assurance リリース 12.8 にアップグレードする際、パフォーマンス データを保持できます。dpmkpdb.exe にアクセスするには、管理者権限が必要です。dpmkpdb.exe コマンドライン ユーティリティ構文では最初の引数として "import"、"export"、または "export_ce" のキーワードを使用します。

このコマンドの形式は、以下のとおりです。

```
dpmkpdb.exe {import | export | export_ce}
-ws_user username -ws_password password
-endpoint endpoint_URI
[-locale iso639value]
{-input input_file_name | -output output_file_name}
```

インポート

入力ファイルからパフォーマンス データをインポートします。

エクスポート

出力ファイルにパフォーマンス データをエクスポートします。

export_ce

収集エンジンから出力ファイルにパフォーマンス データをエクスポートします。

重要: キー パフォーマンス データベースはパフォーマンス データを収集します。前のリリースでパフォーマンス データを収集した収集エンジンは、互換性の理由により、まだ利用することができます。

-ws_user username -ws_password password

システムへのログインに使用する認証情報を指定します。

-endpoint endpoint_URI

エンドポイント URI を上書きします。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-input *input_file_name*

パフォーマンス データをインポートするための入力用ファイルを指定します。

-output *output_file_name*

パフォーマンス データをエクスポートするための出力用ファイルを指定します。

dpmutil の構文

dpmutil は、CA Virtual Assurance コンポーネントの設定を変更することができるコマンドライン ユーティリティです。 **dpmutil** を使用するには、管理 DB 情報がレジストリで利用可能である必要があります、Microsoft SQL クライアントがインストールされている必要があります。 **dpmutil** にアクセスするには、管理者権限を持つユーザ名が必要です。 **dpmutil** コマンドライン ユーティリティの構文では、**set**、**get**、または **delete** の各要求のいずれかを第 1 引数として指定します。指定した要求に応じて、コンポーネントに関する情報を設定、取得、または削除できます。無効な引数を入力したか引数を指定しなかった場合、構文情報および設定可能なコンポーネントのリストを示すエラー メッセージが表示されます。 **dpmutil** の構文は、以下のとおりです。

```
dpmutil {-set|-get|-delete}  
-category|component  
[-userusername] [-passwordpassword]  
[-help]  
[-validateonly]  
[-locale iso639value]
```

-set

コンポーネントの設定を定義または更新します。

-get

コンポーネントの設定を表示します。

-delete

コンポーネントの設定を削除します。

-category/component

データを取得または変更するカテゴリまたはコンポーネントを定義します。

-userusername

(オプション) EEM ユーザを指定します。このパラメータを指定しないと、設定の更新および削除、または既存の設定情報の表示を行う場合に、**dpmutil** によって有効な EEM 認証情報の入力が必要です。

-passwordpassword

(オプション) EEM パスワードを指定します。このパラメータを指定しないと、設定の更新および削除、または既存の設定情報の表示を行う場合に、**dpmutil** によって有効な EEM 認証情報の入力が必要です。

注: **user** パラメータおよび **password** パラメータの間とそれらの値にスペースが入らないようにしてください。

-help

dpmutil のヘルプを表示します。

-validateonly

(オプション) **-set** を使用したときに入力された情報を検証します。有効なコマンドは以下のとおりです。

```
dpmutil -set コマンド -validateonly
```

このコマンドは、情報の作成または更新を行いません。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

注: 変更を有効にするには、Apache HTTP サーバを再起動します。

dpmutil -amq コマンド--ActiveMQ Message Broker の設定

dpmutil set|get|delete amq コマンドを使用すると、CA Virtual Assurance の ActiveMQ Message Broker を設定できます。いくつかの CA Virtual Assurance コンポーネントは、コンポーネント間でデータを渡すために Message Broker を使用します。このコマンドは Message Broker URL を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -amq [-locale iso639value]
```

-set

ActiveMQ Message Broker の URL を定義します。

例 : *tcp://servername:port*

servername

ActiveMQ Message Broker がインストールされているサーバの名前を指定します。

port

ActiveMQ Message Broker のリスニング ポートを指定します。

デフォルト : 61616

-get

ActiveMQ Message Broker に設定された URL を表示します。

-delete

ActiveMQ Message Broker に設定された URL を削除します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は *fr_FR*) で指定します。コマンドプロンプトのロケールを使用する場合は「*native*」を指定します。

dpmutil ecs コマンド--Windows スケジューラの設定

dpmutil set|get ecs コマンドは、スケジューラ サーバを設定します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get} -ecs [-locale iso639value]
```

-set

スケジューラ サーバの設定を定義します。

-get

スケジューラ サーバの設定を表示します。

-ecs

設定対象のスケジューラ サーバを定義します。ECS がアクティブなスケジューラであるかどうかを指定するよう求められます（アクティブなスケジューラの場合は「Y」を入力します。設定をスキップするには、Enter キーを押します）。

-locale iso639value

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は fr_FR）で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

dpmutil -hd コマンド -- ヘルプ デスクの設定

dpmutil set|get hd コマンドは、CA Virtual Assurance のヘルプ デスク情報を設定します。このコマンドを使用して、CA Virtual Assurance の既存のインストール環境にあるヘルプ デスク システムを設定できます。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -hd [locale iso639value]
```

-set

ヘルプ デスク システムの設定を定義します。

-get

ヘルプ デスク システムの設定を表示します。

-delete

CA Virtual Assurance からヘルプ デスク サーバを削除しますが、削除したサーバは、履歴管理のために管理 DB およびパフォーマンス DB に残されます。各パラメータの入力を要求するメッセージが表示されます。

-hd

ヘルプ デスク サーバに接続するのに必要な情報を定義します。ヘルプ デスク サーバ名、管理者のユーザ名とパスワード、およびポートの入力が求められます。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmutil -hmc コマンド -- HMC サーバまたは IVM サーバの設定

`dpmutil hmc` コマンドは、設定された HMC サーバまたは IVM サーバを追加、変更、一覧表示、または削除します。このコマンドを使用して、CA Virtual Assurance の既存のインストール環境に HMC サーバまたは IVM サーバを設定できます。情報を設定または取得するには、CA Virtual Assurance の管理者ユーザ名とパスワードおよびポートが必要です。

注: 変更を有効にするには、Apache HTTP サーバを再起動します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -hmc [-locale iso639value]
```

-set

HMC サーバまたは IVM サーバの構成設定を追加または変更します (サーバ名、ユーザ名、パスワードなど)。

-get

HMC サーバまたは IVM サーバの構成設定を表示します (サーバ名、ユーザ名など)。

-delete

HMC サーバまたは IVM サーバを削除しますが、削除したサーバは、履歴管理のために管理 DB およびパフォーマンス DB に残されます。各パラメータの入力を要求するメッセージが表示されます。

-hmc

HMC サーバまたは IVM サーバを定義します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmutil -hyperv コマンド -- Hyper-V Server の設定

dpmutil set|get -hvserver コマンドは、仮想マシン (VM) のプロビジョニングで構成された **Hyper-V Server** を追加、変更、一覧表示、または削除します。このコマンドを使用すると、**CA Virtual Assurance** の既存のインストールに **Hyper-V** コンポーネントをインストールできます。情報を設定または取得するには、**CA Virtual Assurance** の管理者ユーザ名とパスワードおよびポートが必要です。この設定を変更した後で **Apache** を再起動する必要はありません。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -hyperv [locale iso639value]
```

-set

Hyper-V サーバの構成設定を追加または変更します (サーバ名、ユーザ名、ポート、プロトコルなど)。

-get

Hyper-V サーバの構成設定を表示します (サーバ名、ユーザ名、ポート、プロトコルなど)。

-delete

Hyper-V サーバを削除します。ただし、履歴上の目的で管理 DB とパフォーマンス DB 内には維持されます。各パラメータの入力を求められます。

-hyperv

Hyper-V サーバを定義します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmutil mgmtadb コマンド -- 管理データベースの設定

`dpmutil set|get -mgmtadb` コマンドは、製品のほとんどすべてのコンポーネントがメインデータベースとして使用する管理データベースを設定します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get} -mgmtadb [-locale iso639value]
```

-set

管理データベースの設定を定義します。

-get

管理データベースの設定を表示します。

-mgmtadb

設定対象の管理データベースを定義します。主な製品テーブルにアクセスするために使用するデータベース サーバ、バージョン、ポート、および認証情報 (SQL 認証を使用する場合) の入力が必要です。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

注: これらの情報を有効にするには、依存性のある製品サービス (CAAIPOracle および CAAIPTomcat) をリサイクルする必要があります。

dpmutil -perfdb コマンド -- パフォーマンス データベースの設定

dpmutil set|get -perfdb コマンドは、CA Virtual Assurance のパフォーマンス DB を設定します。このデータベースには、CA Virtual Assurance によって収集されたパフォーマンス情報がすべて格納されます。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -perfdb [-locale iso639value]
```

-set

パフォーマンス DB の設定を定義します。

-get

パフォーマンス DB の設定を表示します。

-delete

パフォーマンス DB の設定を削除します。

-perfdb

構成するコンポーネントを定義します。サーバ名、管理者のユーザ名とパスワード、データベース タイプ、データベース インスタンス、およびデータベース ポートの入力が必要です。パフォーマンス DB の認証情報は通常、管理 DB の認証情報と同じです。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

注: これらの情報を有効にするには、依存性のある製品サービス (CAAIPOracle および CAAIPTomcat) をリサイクルする必要があります。

dpmutil remotevcaim コマンド -- リモート VC AIM の設定

dpmutil set|get remotevcaim コマンドは、リモート VC AIM を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -remotevcaim [-locale iso639value]
```

-set

リモート VC AIM の設定を定義します。

-get

リモート VC AIM の設定を表示します。

-delete

現在設定されているリモート VC AIM を削除します。

-remotevcaim

設定対象のリモート VC AIM を定義します。VC AIM がインストールされているサーバ、VC 自体がインストールされているサーバ、および VC サーバへの接続に使用する SNMP 書き込みアクセス文字列の入力が求められます。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

dpmutil -sc コマンド -- サービスコントローラの設定

dpmutil set|get -sc コマンドを使用すると、CA Virtual Assurance のサービスコントローラを設定できます。サービスコントローラとは、異なる CA Virtual Assurance コンポーネント間での相互通信を可能にする通信トラフィックコントローラです。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -sc [-locale iso639value]
```

-set

サービスコントローラの設定を定義します。

-get

サービス コントローラの設定を表示します。

-delete

サービス コントローラの設定を削除します。

-sc

ローカル サーバのサービス コントローラを定義します。

モード

サービス コントローラを見つけるために使用される方法の設定を定義します。以下のモードがサポートされています。

SRV の自動ルックアップ

DNS SRV レコードを使用して、サービス コントローラを見つけます。

手動優先

指定された URL に基づいて、インストール済みで実行中のサービス コントローラを識別します。

例： `https://hostname:port/dpm/sc`

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は `fr_FR`）で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmutil -scvmmserver コマンド -- SCVMM Server の設定

dpmutil set|get -scvmmserver コマンドは、仮想マシン (VM) のプロビジョニングで構成された SCVMM サーバを追加、変更、一覧表示、または削除します。このコマンドを使用すると、CA Virtual Assurance の既存のインストールに SCVMM コンポーネントをインストールできます。情報を設定または取得するには、CA Virtual Assurance の管理者ユーザ名とパスワードおよびポートが必要です。この設定を変更した後で Apache を再起動する必要があります。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -scvmmserver [-locale iso639value]
```

-set

SCVMM サーバの構成設定を追加または変更します (サーバ名、ユーザ名、ポート、プロトコルなど)。

-get

SCVMM サーバの構成設定を表示します (サーバ名、ユーザ名、ポート、プロトコルなど)。

-delete

SCVMM サーバを削除します。ただし、履歴上の目的で管理 DB とパフォーマンス DB 内には維持されます。各パラメータの入力を求められます。

-scvmmserver

SCVMM サーバを定義します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmutil -snmp コマンド -- SNMP の設定

dpmutil set|get|delete -snmp コマンドは、SNMP 情報の受信用に、データ センターまたはサーバ レベルの **SNMP** 設定を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -snmp [-locale iso639value]
```

-set

データ センターまたはサーバの設定を追加または変更します。たとえば以下の項目があります。

- データ センターまたはサーバ レベル
- サーバ設定の場合は、サーバ名
- SNMP 名
- タイムアウト
- 再試行回数
- 書き込みアクセス
- バージョン
- セキュリティ ユーザ
- 認証タイプ
- 認証パスワード
- プライバシー タイプ
- プライバシー パスワード
- コミュニティ文字列

各パラメータの入力を要求するメッセージが表示されます。サーバとデータ センターの両方の **SNMP** 設定を使用するか、またはサーバの **SNMP** 設定のみを使用するように、サーバを設定できます。

-get

以下の項目など、データ センターまたはサーバの **SNMP** 設定を一覧表示します。

- **SNMP** 名
- バージョン
- ポート
- タイムアウト
- 再試行回数
- 書き込みアクセス
- 認証タイプ
- プライバシー タイプ
- セキュリティ ユーザ
- コミュニティ文字列

サーバでデータ センターのデフォルトの **SNMP** 設定が使用される場合でも、サーバの **SNMP** 設定のみを一覧表示できます。

-delete

データ センターまたはサーバの **SNMP** 設定を削除します。データ センターまたはサーバの **SNMP** 設定を削除するよう求められます。サーバの **SNMP** 設定の場合、サーバ名および **SNMP** ポートの入力が必要になります。データ センターの **SNMP** 設定の場合、**SNMP** 名およびポートの入力が求められます。

-snmp

SNMP 設定を定義します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、**ISO 639_3166** の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmutil -systemcredential コマンド -- 外部システムの認証情報の設定

`dpmutil set|get -systemcredential` コマンドは、外部システムの、グローバルなアクセス用とシステム固有のアクセス用の両方の認証情報を定義します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -systemcredential [-locale iso639value]
```

-set

グローバルなアクセス用とシステム固有のアクセス用の両方の外部システムの認証情報設定を定義します。

-get

外部システムの、グローバルなアクセス用とシステム固有のアクセス用の両方の認証情報設定を表示します。

-delete

削除するように選択された認証情報に応じて、グローバルなアクセス用とシステム固有のアクセス用の両方の外部システムの認証情報を削除します。

-systemcredential

設定対象の外部システムの、グローバルなアクセス用とシステム固有のアクセス用の両方の認証情報を定義します。（データセンター全体によって、またはサーバレベルでのみ継承された）認証情報レベルの入力が求められます。認証情報レベルに応じて、認証情報を適用する特定のサーバ名を入力するよう求められる場合があります。さらに、認証情報名、認証情報自体（ユーザ/パスワード）、認証情報が属するグループ（オプション）の入力が求められます。

-locale iso639value

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は `fr_FR`）で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmutil -sysuser コマンド -- サービス起動時に使用されるローカル認証情報の設定

dpmutil set|get -sysuser コマンドは、メインの製品サービス CAAIPApache のサービス起動時に使用されるローカル認証情報を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get} -sysuser [-locale iso639value]
```

-set

サービス起動時に使用されるローカル認証情報の設定を定義します。

-get

サービス起動時に使用されるローカル認証情報の設定を表示します。

-sysuser

設定対象のサービスの起動時に使用されるローカル認証情報を定義します。ユーザ名およびパスワードの入力が求められます。このユーザとパスワードは、設定された EEM サーバへのログインで有効な必要があります。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

注: これらの情報を有効にするには、依存性のある製品サービス (CAAIPApache および CAAIPTomcat) をリサイクルする必要があります。

dpmutil vcdefault コマンド -- デフォルトの VMware vCenter 管理者ユーザの設定

dpmutil set|get|delete vcdefault コマンドを使用すると、CA Virtual Assurance のデフォルトの VMware vCenter 管理者ユーザを設定できます。VMware vCenter サーバが検出されると、CA Virtual Assurance は、デフォルトの VMware vCenter 管理者ユーザ (vcdefault) を使用します。このコマンドは vcdefault を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -vcdefault [-locale iso639value]
```

-set

VMware vCenter の vcdefault ユーザを定義します。

-get

VMware vCenter ディスカバリのために設定されたデフォルト ユーザを表示します。

-delete

VMware vCenter ディスカバリのために設定されたデフォルト ユーザを削除します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmutil vcserver コマンド -- VMware vCenter の設定

dpmutil set|get vcserver コマンドは、仮想マシン (VM) のプロビジョニング用に設定された VMware vCenter サーバを追加、変更、一覧表示、または削除します。このコマンドを使用して、CA Virtual Assurance の既存のインストール環境に VMware VirtualCenter コンポーネントをインストールできます。情報を設定または取得するには、CA Virtual Assurance の管理者ユーザ名とパスワードおよびポートが必要です。この設定を変更した後で Apache を再起動する必要はありません。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -vcserver [-locale iso639value]
```

-set

サーバ名、ユーザ名、ポート、およびプロトコルを含めて VMware vCenter サーバの設定を追加または変更します。

-get

サーバ名、ユーザ名、ポート、およびプロトコルを含めて VMware vCenter サーバの設定を表示します。

-delete

VMware vCenter サーバを削除しますが、削除したサーバは、履歴管理のために管理 DB およびパフォーマンス DB に残されます。各パラメータの入力を要求するメッセージが表示されます。

-vcserver

VMware vCenter サーバを定義します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

dpmutil -vioserver コマンド -- VIO サーバの設定

`dpmutil -vioserver` コマンドは、設定された VIO サーバを追加、変更、一覧表示、または削除します。このコマンドを使用すると、CA Virtual Assurance の既存のインストールに VIO サーバを設定できます。情報を設定または取得するには、CA Virtual Assurance の管理者ユーザ名とパスワードおよびポートが必要です。

注: 変更を有効にするには、Apache HTTP サーバを再起動します。

このコマンドの形式は、以下のとおりです。

```
dpmutil {-set|-get|-delete} -vioserver [-locale iso639value]
```

-set

VIO サーバの構成設定を追加または変更します（サーバ名、ユーザ名、パスワードなど）。

-get

VIO サーバの構成設定を表示します（サーバ名、ユーザ名など）。

-delete

VIO サーバを削除します。ただし、履歴上の目的で管理 DB とパフォーマンス DB 内には維持されます。各パラメータの入力を要求するメッセージが表示されます。

-vioserver

VIO サーバを定義します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は `fr_FR`）で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

Common Discovery コマンド

検出コマンドは、IP アドレスまたはサブネット IP アドレスを使ってコンピュータを検出するのに使用されます。CLI を使用すると、ディスカバリコマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。

dpmcda cancelnetworkscan コマンド -- ネットワーク ディスカバリの取り消し

dpmcda cancelnetworkscan コマンドは、進行中のネットワーク ディスカバリを取り消します。

このコマンドの形式は、以下のとおりです。

```
dpmcda cancelnetworkscan [-sc sc_url] -network_name networkname [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-network_name *networkname*

サブネット名。重複するネットワーク名は認められません。固有のネットワーク名を持つ 1 つのサブネットのみを検出できます。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ネットワーク ディスカバリを取り消す

以下のコマンドは、進行中のネットワーク ディスカバリを取り消します。

```
dpmcda cancelnetworkscan -network_name Network1
```

dpmcda deletenetwork コマンド -- ネットワークの削除

dpmcda deletenetwork コマンドは、特定のネットワークを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmcda deletenetwork [-sc sc_url] -network_name networkname [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-network_name *networkname*

サブネット名。重複するネットワーク名は認められません。固有のネットワーク名を持つ 1 つのサブネットのみを検出できます。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ネットワークを削除する

以下のコマンドは、ネットワーク **Network1** を削除します。

```
dpmcda deletenetwork -network_name Network1
```

dpmcda discoverhost コマンド -- ホストを検出する

dpmcda discoverhost コマンドはホストを検出します。

このコマンドの形式は、以下のとおりです。

```
dpmcda discoverhost [sc sc_url] -host_name hostname [-system_user username  
-system_password password][-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-host_name *hostnames*

1 つ以上のホストを定義します。有効なエントリ : カンマ区切りのホスト名または IP アドレス。

-system_user *username* -system_password *password*

拡張ディスカバリ (ハードウェア特性の取得) 用のターゲットシステム認証情報を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ホストを検出する

この例では、ホストを検出します。

```
dpmcda discoverhost -host_name 172.24.255.255 -ws_user wsuser -ws_password
wsuserpassword
```

dpmcda discovernetwork コマンド -- ネットワークを検出する

dpmcda discovernetwork コマンドはネットワークを検出します。

このコマンドの形式は、以下のとおりです。

```
dpmcda discovernetwork [-sc sc_url] -ipv4_addr ipv4address -network_name networkname
[-dnsserver dnsserver] [-domain domain] [-system_user username -system_password
password] [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-ipv4_addr *ipv4address*

検出するサブネットの IPv4 アドレスを定義します。サブネットの有効な IPv4 アドレス形式: **xxx.xxx.x.***、**xxx.xxx.x.{1-10}**、**xxx.xxx.x.0/24**。

-network_name *networkname*

サブネット名。重複するネットワーク名は認められません。固有のネットワーク名を持つ 1 つのサブネットのみを検出できます。

-dns_server *dnsserver*

DNS サーバの IP アドレスを定義します。

-domain *domain*

DNS ディスカバリ用のドメイン名を定義します。

-system_user *username* -system_password *password*

拡張ディスカバリ（ハードウェア特性の取得）用のターゲットシステム認証情報を定義します。

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティチェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネットワークを検出する

この例では、**NetworkA** という名前の IPv4 アドレスのネットワークを検出します。

```
dpmcda discovernetwork -ipv4_addr 172.24.255.255 -network_name NetworkA -ws_user wsuser -ws_password wsuserpassword
```

収集エンジン コマンド

スクリプトに CLI を使用して収集エンジン コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。これらのコマンドを使用するには、SNMP の専門知識が必要です。

注: これらのコマンドを使用するには SNMP の専門知識が必要です。また、サポートされている MIB の列ベースのメトリックを設定する専門知識も必要です。

dpmce ceconfig -configuremetric コマンド -- メトリックを設定する

dpmce ceconfig -configuremetric コマンドでは、ホスト上のメトリックを追加、変更、削除できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -configuremetric [-sc sc_url] -host_name hostname -operation
add|delete -sourceType 3 -metricname metricname -metric_type metrictype
-metric_subtype metricsubtype -metric_instance metricinstance [-metric_method
exact|complement] [-metric_datatype metricdatatype] [-lower_threshold
lowerthreshold] [-upper_threshold upperthreshold] [-metric_enabled metricenabled]
[metric_update yes|no][-ws_user username -ws_password password][-locale
iso629value]
```

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-host_name hostname

ホスト コンピュータの名前を定義します。

-operation add|delete

エイリアス（-setalias）またはメトリック（-configuremetric）を追加または削除します。

-sourceType

メトリックを収集するソースを定義します。有効なエントリ：1 = システムメトリック（パフォーマンス ライトメトリック）、3 = SNMP インターフェースからの収集。

-metricname *metricname*

メトリックの名前を定義します。

-metric_type *metrictype*

メトリックのシンボル名が後に続く MIB 名。

-metric_subtype *MIBOID*

管理対象オブジェクトを一意に識別する MIB オブジェクト識別子（OID）を定義します。

-metric_instance *metricinstance*

収集するデータの列番号。

-metric_method exact|complement|exact_delta|exact_complement

計算と収集のメソッドを定義します。有効なエントリは以下のとおりです。

- **exact** = 厳密なメトリック値を使用します。メトリック値が大きくなると、全体的な使用率も高くなります。例：メモリ。
- **complement** = 補完的にメトリックを使用します。値が大きくなると、全体的な使用率は低くなります。例：CPU のアイドル率。
- **exact_delta** = 連続する値の差を使用します。例：
MIB2.Instance.ifInOctets（受信バイト数）。

complement_delta = 連続する値の差を補完的に使用します。例：CPU アイドル合計。

-metric_datatype *metricdatatype*

メトリックのデータ タイプを定義します。有効なエン트리：整数、浮動小数点数、倍精度浮動小数点数。

-lower_threshold *lowerthreshold*

データ センターの全体的な使用率の下限しきい値を定義します。下限しきい値は常に上限しきい値未満である必要があります。

値の範囲：0 ～ 100%

-upper_threshold *upperthreshold*

データ センターの全体的な使用率の上限しきい値を定義します。上限しきい値は常に下限しきい値を超えている必要があります。

値の範囲：1 ～ 100%

-metric_enabled *yes|no*

メトリックが収集で有効かどうかを定義します。有効なエン트리：yes または no。

-metric_update *yes|no*

メトリックをデータベースで更新するかどうかを定義します。有効なエン트리：yes または no。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: データ収集用のメトリックを設定する

この例では、ホストのメトリック設定データを追加します。

```
dpmce ceconfig -configuremetric -host_name S0L1054 -operation add -sourceType 3
-metricname SysEDGE.loadAverage15Min -metric_type SysEDGE.loadAverage15Min
-metric_subtype 1.3.6.1.4.1.546.1.1.7.8.28.0 -metric_method exact -metric_update
yes -ws_user admin -ws_password admin
```

dpmce ceconfig -configuremetricfilter コマンド -- メトリック フィルタを設定する

dpmce ceconfig -configuremetricfilter コマンドは、メトリック フィルタを追加、変更、削除します。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -configuremetricfilter [-sc sc_url] -host_name hostname -operation
add|delete -sourceType 3 -metric_os_type metricOSType -metric_type metrictype
-metric_subtype metricsubtype -metric_instance metricinstance [-metric_method
exact|complement] [-metric_datatype metricdatatype] [-metric_filter_default 0|1]
[-include_for_overall yes|no] [-lower_threshold lowerthreshold] [-upper_threshold
upperthreshold] [-metric_enabled yes|no] [-ws_user username -ws_password
password][-locale iso629value]
```

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-host_name hostname

ホスト コンピュータの名前を定義します。

-operation add|delete

エイリアス (-setalias) またはメトリック (-configuremetric) を追加または削除します。

-sourceType

メトリックを収集するソースを定義します。有効なエントリ : **1** = システム メトリック (パフォーマンス ライト メトリック)、**3** = SNMP インターフェースからの収集。

-metric_os_type metricOSType

メトリックのオペレーティング システム タイプを定義します。有効なエントリ：WINDOWS、AIX、HPUX、LINUX、SOLARIS。

-metricname metricname

メトリックの名前を定義します。

-metric_subtype MIBOID

管理対象オブジェクトを一意に識別する MIB オブジェクト識別子 (OID) を定義します。

-metric_instance metricinstance

収集するデータの列番号。

-metric_method exact|complement|exact_delta|exact_complement

計算と収集のメソッドを定義します。有効なエントリは以下のとおりです。

- **exact** = 厳密なメトリック値を使用します。メトリック値が大きくなると、全体的な使用率も高くなります。例：メモリ。
- **complement** = 補完的にメトリックを使用します。値が大きくなると、全体的な使用率は低くなります。例：CPU のアイドル率。
- **exact_delta** = 連続する値の差を使用します。例：MIB2.Instance.ifInOctets（受信バイト数）。
- **complement_delta** = 連続する値の差を補完的に使用します。例：CPU アイドル合計。

-metric_datatype metricdatatype

メトリックのデータ タイプを定義します。有効なエントリ：整数、浮動小数点数、倍精度浮動小数点数。

-metric_filter_default yes|no

メトリックをデフォルトで収集するかどうかを定義します。有効なエントリ：yes または no。

-include_for_overall Yes|No

メトリックが全体的な計算に含まれるかどうかを定義します。

Yes

全体的な計算にメトリックを含めます。

No

全体的な計算からメトリックを除外します。

-lower_threshold *lowerthreshold*

データ センターの全体的な使用率の下限しきい値を定義します。下限しきい値は常に上限しきい値未満である必要があります。

値の範囲： 0 ～ 100%

-upper_threshold *upperthreshold*

データ センターの全体的な使用率の上限しきい値を定義します。上限しきい値は常に下限しきい値を超えている必要があります。

値の範囲： 1 ～ 100%

-metric_enabled yes|no

メトリックが収集で有効かどうかを定義します。有効なエントリ：yes
または no。

-ws_user *username* **-ws_password** *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: メトリックフィルタを設定する

この例では、メトリック フィルタを追加します。

```
dpmce ceconfig -configuremetricfilter -operation add -sourceType 3 -metric_os_type  
WINDOWS -metricname "Fake.cpuTotalUserPercent" -metric_subtype "xxxxxxxxxxxxxxxxx"  
-metric_method complement_delta -ws_user dcauser -ws_password dcapassword
```


dpmce ceconfig -disablemetric コマンド -- メトリックを無効にする

dpmce ceconfig -disablemetric コマンドでは、収集されるホストのメトリックを無効にできます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -disablemetric [-sc sc_url] -host_name hostname -metricname metricname [-ws_user username -ws_password password] [-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-metricname *metricname*

メトリックの名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ホストのメモリ メトリックを無効にする

この例では、ホスト **ServerZ** のメトリック **Memory: AvailableMbytes** を無効にします。

```
dpmce ceconfig -disablemetric -host_name ServerZ -metricname Memory: AvailableMbytes
```

dpmce ceconfig -disablemetricforoverall コマンド -- メトリックを無効にし、全体的な計算から除外する

dpmce ceconfig -disablemetricforoverall コマンドでは、現在収集されているホスト メトリックを無効にし、全体的な計算から除外できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -disablemetricforoverall [-sc sc_url] -host_name hostname -metricname metricname [-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-metricname *metricname*

メトリックの名前を定義します。

`-ws_user username -ws_password password`

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ホストのメモリ メトリックを無効にする

この例では、メトリック `Memory: AvailableMbytes` を無効にし、ホスト `ServerZ` の全体的な計算から除外します。

```
dpmce ceconfig -disablemetricforoverall -host_name ServerZ -metricname Memory:
AvailableMbytes
```

dpmce ceconfig -enablemetric コマンド -- メトリックを有効にする

`dpmce ceconfig -enablemetric` コマンドでは、ホスト メトリックを有効にすることができます。メトリックは収集で利用できる必要があります。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -enablemetric [-sc sc_url] -host_name hostname -metricname metricname
[-include_for_overall Yes|No][-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-metricname *metricname*

メトリックの名前を定義します。

-include_for_overall Yes|No

メトリックが全体的な計算に含まれるかどうかを定義します。

Yes

全体的な計算にメトリックを含めます。

No

全体的な計算からメトリックを除外します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ホストのメモリ メトリックを有効にする

この例では、メトリック **Memory: AvailableMbytes** を有効にし、ホスト **ServerZ** の全体的な計算に組み込みます。

```
dpmce ceconfig -enablemetric -host_name ServerZ -metricname Memory: AvailableMbytes  
-include_for_overall Yes
```

dpmce ceconfig -enablemetricforoverall コマンド -- メトリックを有効にし、全体的な計算に組み込む

dpmce ceconfig -enablemetricforoverall コマンドでは、現在収集されているホスト メトリックを有効にし、全体的な計算に組み込むことができます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -enablemetricforoverall [-sc sc_url] -host_name hostname -metricname  
metricname [-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-metricname *metricname*

メトリックの名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ホストのメモリ メトリックを有効にし、全体的な計算に組み込む

この例では、メトリック **Memory: AvailableMbytes** を有効にし、ホスト **ServerZ** の全体的な計算に組み込みます。

```
dpmce ceconfig -enablemetricforoverall -host_name ServerZ -metricname Memory:
AvailableMbytes
```

dpmce ceconfig -getglobal コマンド -- データ センターの間隔を取得する

dpmce ceconfig -getglobal コマンドでは、データ センター レベルの収集、記録、保持の間隔を取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -getglobal [-sc sc_url][-ws_user username -ws_password
password][-locale iso629value]
```

`-sc sc_url`

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

`-ws_user username -ws_password password`

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: 収集、記録、保持の間隔を取得する

この例では、データ センター レベルの収集、記録、保持の間隔を取得します。

```
dpmce ceconfig -getglobal
```

dpmce ceconfig -getnode コマンド -- ホストの収集間隔を取得する

dpmce ceconfig -getnode コマンドでは、ホストの収集、記録、保持の間隔を取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -getnode [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ホスト ServerZ の収集、記録、保持の間隔を取得する

この例では、ホスト ServerZ の間隔を取得します。

```
dpmce ceconfig -getnode -host_name ServerZ
```

dpmce ceconfig -getmetriclist コマンド -- メトリックリストを取得する

dpmce -getmetriclist コマンドでは、特定のホストのメトリックを取得できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -getmetriclist [-sc sc_url] -host_system hostname [-ws_user username  
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: メトリックリストを取得する

この例では、ホスト **SOL1054** のメトリック リストを取得します。

```
dpmce ceconfig -getmetriclist -host_name SOL1054
```

dpmce ceconfig -refreshmetriclist コマンド -- メトリック収集を再試行する

dpmce ceconfig -refreshmetriclist コマンドでは、メトリック収集を再試行できます。たとえば、もし検出時にエージェントがインストールされていなければ、収集してもメトリックは検出されません。後でシステムにエージェントをインストールする場合、このコマンドを使用すると、システムを再検出しなくてもメトリック収集を再試行できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -refreshmetriclist [-sc sc_url] -host_name hostname [-ws_user username  
-ws_password password][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-host_name *hostname*

ホスト コンピュータの名前を定義します。

`-ws_user username -ws_password password`

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: メトリック収集を再試行する

この例では、いくつかのホスト上のメトリック収集を再試行します。

```
dpmce ceconfig -refreshmetriclist -host_name SOL1054:engManager200:QALAB01
```

dpmce ceconfig -setalias コマンド -- エイリアスを設定する

`dpmce ceconfig -setalias` コマンドは、ESX サーバの全体的な計算にメトリックを組み込めるように、ESX ホストのエイリアスを作成します。

このコマンドの形式は、以下のとおりです。

```
ceconfig -setalias [-sc sc_url] -operation add|delete -caller_id PMM -metricname
metricname -alias_host_name aliashostname -parent_host_name parenthostname
-include_for_overall yes|no [-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-operation *add|delete*

エイリアス (`-setalias`) またはメトリック (`-configuremetric`) を追加または削除します。

-caller_id *PMM*

エイリアスのインターフェース (製品管理モジュール) を定義します。
有効なエントリ : `PMM`。

-metricname *metricname*

メトリックの名前を定義します。有効なエントリ : メトリック名または `ALL` (エイリアスのホスト名に関連付けられているすべてのメトリックを削除する場合) 。

-alias_host_name *aliashostname*

`parent_host_name` (AIM) の管理対象の ESX サーバまたは VM を定義します。

-parent_host_name *parenthostname*

AIM と一緒にインストールされたシステムを定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: エイリアスを設定する

この例では、ESX ホストのエイリアスを設定します。

```
ceconfig -setalias -caller_id PMM -metricname SysEDGE.memInUseCapacity
-alias_host_name hostDA12 -parent_host_name engServer200 -include_for_overall yes
-ws_user admin -ws_password admin
```

dpmce ceconfig -setglobal コマンド -- データ センターの間隔を設定する

dpmce ceconfig -setglobal コマンドでは、データ センター レベルの収集、記録、保持の間隔としきい値を設定できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -setglobal [-sc sc_url] -recording_interval value
-retention_sec value -retention_day value
-lower_threshold lowerthreshold -upper_threshold upperthreshold
[-ws_user username -ws_password password][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-recording_interval *value*

記録間隔を定義します。**retention_day** または **retention_sec** を定義していない場合、この値は必須です。

単位： 秒

-retention_day *value*

パフォーマンス DB に日単位データの平均を保存する期間を定義します（日単位ロールアップデータの保持）。**recording_interval** または **retention_sec** を定義していない場合、この値は必須です。

単位： 日数

-retention_sec *value*

パフォーマンス DB にポーリングされたデータを保存する期間を定義します（ポーリングされたデータの保持）。この数を定義する場合は、収集された管理対象システム、サービスおよびメトリックの数を考慮します。保存されたポーリング済みデータ オブジェクトが徐々に蓄積されて、パフォーマンスに影響を与える場合があります。パフォーマンスの問題が発生する場合は、保存日数の数を減らします。

recording_interval または **retention_day** を定義していない場合、この値は必須です。

単位： 日数

-lower_threshold *lowerthreshold*

データ センターの全体的な使用率の下限しきい値を定義します。下限しきい値は常に上限しきい値未満である必要があります。

値の範囲： 0 ～ 100%

-upper_threshold *upperthreshold*

データ センターの全体的な使用率の上限しきい値を定義します。上限しきい値は常に下限しきい値を超えている必要があります。

値の範囲： 1 ～ 100%

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。**caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: データセンターの収集間隔と記録間隔を設定する

この例では、データセンターの収集間隔と記録間隔を設定します。

```
dpmce ceconfig -setglobal -recording_interval 120
```

例: データセンターのしきい値を設定する

この例では、データセンターの下限しきい値と上限しきい値を設定します。

```
dpmce ceconfig -setglobal -lower_threshold 10 -upper_threshold 90
```

dpmce ceconfig -setnode コマンド -- ホストの間隔を設定する

dpmce ceconfig -setnode コマンドでは、ホストの収集、記録、保持の間隔を設定できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -setnode [-sc sc_url] -host_name hostname -recording_interval value  
-retention_sec value -retention_day value [-ws_user username -ws_password  
password][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-recording_interval *value*

記録間隔を定義します。**retention_day** または **retention_sec** を定義していない場合、この値は必須です。

単位 : 秒

-retention_sec *value*

パフォーマンス DB にポーリングされたデータを保存する期間を定義します (ポーリングされたデータの保持)。この数を定義する場合は、収集された管理対象システム、サービスおよびメトリックの数を考慮します。保存されたポーリング済みデータ オブジェクトが徐々に蓄積されて、パフォーマンスに影響を与える場合があります。パフォーマンスの問題が発生する場合は、保存日数の数を減らします。

recording_interval または **retention_day** を定義していない場合、この値は必須です。

単位 : 日数

-retention_day value

パフォーマンス DB に日単位データの平均を保存する期間を定義します（日単位ロールアップデータの保持）。**recording_interval** または **retention_sec** を定義していない場合、この値は必須です。

単位：日数

-ws_user username -ws_password password

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。**caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ホスト ServerZ の保持間隔を設定する

この例では、ホスト **ServerZ** の保持間隔を設定します。

```
dpmce ceconfig -setnode -host_name ServerZ -retention_day 300
```

dpmce ceconfig -updatemetric コマンド -- メトリックを更新する

dpmce ceconfig -updatemetric コマンドでは、現在収集されているホスト メトリック、しきい値、全体的な計算の属性を更新できます。

このコマンドの形式は、以下のとおりです。

```
dpmce ceconfig -updatemetric [-sc sc_url] -host_name hostname -metricname metricname
[-lower_threshold lowerthreshold][-upper_threshold upperthreshold]
include_for_overall Yes|No [-ws_user username -ws_password password][-locale
iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-host_name *hostname*

ホスト コンピュータの名前を定義します。

-metricname *metricname*

メトリックの名前を定義します。

-lower_threshold *lowerthreshold*

データ センターの全体的な使用率の下限しきい値を定義します。下限しきい値は常に上限しきい値未満である必要があります。

値の範囲 : `0 ~ 100%`

-upper_threshold *upperthreshold*

データ センターの全体的な使用率の上限しきい値を定義します。上限しきい値は常に下限しきい値を超えている必要があります。

値の範囲 : `1 ~ 100%`

-include_for_overall Yes|No

メトリックが全体的な計算に含まれるかどうかを定義します。

Yes

全体的な計算にメトリックを含めます。

No

全体的な計算からメトリックを除外します。

`-ws_user username -ws_password password`

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ホスト ServerZ のメモリ メトリックを更新する

この例では、メトリック `Memory: AvailableMbytes` の上限しきい値と下限しきい値を更新し、ホスト `ServerZ` の全体的な計算に組み込みます。

```
dpnce ceconfig -updatemetric -host_name ServerZ -metricname Memory: AvailableMbytes
-lower_threshold 10 -upper_threshold 90 -include_for_overall Yes
```

イベントコマンド

CLI を使用すると、イベント コマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。

dpmeventmanager create_event コマンド -- イベントの作成

`dpmeventmanager create_event` コマンドは、イベントを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmeventmanager create_event [-sc sc_url] [-status eventstatus] [-component
eventcomponent] [-message eventmessage] [-source eventsource] [-target eventtarget]
[-action eventaction] [-audit {0|1}][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-status *eventstatus*

(オプション) イベントのステータスを指定します。

-component *eventcomponent*

(オプション) イベントが対応する製品コンポーネントを定義します。

例 : ポリシー、イメージングなど。

-message *eventmessagesubstring*

(オプション) 一致するイベント メッセージ サブ文字列を定義します。

-source *eventsourcesource*

(オプション) イベントを生成するコンピュータを定義します。

-target *eventtargettarget*

(オプション) イベントを発生させるターゲット コンピュータを定義します。

-action *eventactionaction*

(オプション) イベントのカテゴリを定義します。

-audit={0|1}

(オプション) イベントが監査イベントかどうかを指定します。1 (監査イベント) または 0 (監査イベントでない) を使用します。

デフォルト : 0

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サーバを検出するイベントを作成する

以下の例は、**ServerXYZ** のサーバ ディスカバリ用のイベントを作成します。

```
dpmeventmanager create_event -status 1 -component ccm -message "1 New Managed System  
Added - xyz" -source xyz -action "Inventory Event"
```

dpmeventmanager get_events コマンド -- イベントの取得

dpmeventmanager get_events コマンドは、イベントを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmeventmanager get_events [-sc sc_url] [-status event_status] [-component  
eventcomponent] [-message eventmessagesubstring] [-source eventsources] [-target  
eventtarget] [-action eventaction] [-from_start_date_time] [-to_end_date_time]  
[-audit {0|1}] [-user_name username][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-status *eventstatus*

(オプション) イベントのステータスを指定します。

-component *eventcomponent*

(オプション) イベントが対応する製品コンポーネントを定義します。

例：ポリシー、イメージングなど。

-message *eventmessagesubstring*

(オプション) 一致するイベント メッセージ サブ文字列を定義します。

-source *eventsourcesource*

(オプション) イベントを生成するコンピュータを定義します。

-target *eventtargettarget*

(オプション) イベントを発生させるターゲット コンピュータを定義します。

-action *eventactionaction*

(オプション) イベントのカテゴリを定義します。

-from *start_date_time*

(オプション) 取得するイベントの開始日時を、月 (*mm*)、日 (*dd*)、年 (*yyyy*) を使用して指定します。デフォルトは、日付に関係なく、すべてのイベントが取得されます。このパラメータでは、以下を指定できます。

mm

この月に開始される、作成済みのイベントが対象となります。

値の範囲： 1 ～ 12

dd

この日に開始される、作成済みのイベントが対象となります。

値の範囲： 1 ～ 31

yyyy

この年に開始される、作成済みのイベントが対象となります。

値の範囲： 1970 以降

hh

この時間に開始される、作成済みのイベントが対象となります。

値の範囲： 0 ～ 23

mm

この分に開始される、作成済みのイベントが対象となります。

値の範囲： 0 ～ 59

ss

この秒に開始される、作成済みのイベントが対象となります。

値の範囲： 0 ～ 59

-to end_date_time

(オプション) 取得するイベントの終了日時を、月 (*mm*)、日 (*dd*)、年 (*yyyy*) を使用して指定します。デフォルトは、日付に関係なく、すべてのイベントが取得されます。このパラメータでは、以下を指定できます。

mm

この月に終了する、作成済みのイベントが対象となります。

値の範囲： 1 ～ 12

dd

この日に終了する、作成済みのイベントが対象となります。

値の範囲： 1 ～ 31

yy

この年に終了する、作成済みのイベントが対象となります。

値の範囲： 01 ～ 09

hh

この時間に終了する、作成済みのイベントが対象となります。

値の範囲： 1 ～ 24

mm

この分に終了する、作成済みのイベントが対象となります。

値の範囲： 1 ～ 60

ss

この秒に終了する、作成済みのイベントが対象となります。

値の範囲： 1 ～ 60

-audit={0|1}

(オプション) イベントが監査イベントかどうかを指定します。1 (監査イベント) または 0 (監査イベントでない) を使用します。

デフォルト : 0

-user_name *username*

イベントをトリガしたポリシー変更を行ったユーザ名を定義します。CLI は、オペレーティング システムからユーザ名を自動的に判断します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 特定の期間のポリシー イベントを取得する

以下の例は、2009 年 3 月 8 日に午前 10 時 10 分 10 秒から現在までのすべてのポリシー イベントを取得します。

```
dpmeventmanager get_events -component policy -from "3/08/2009 10:10:10"
```

汎用シェル コマンド

CLI を使用すると、さまざまなコマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。

caaipsecurity コマンド -- Web サービスのセキュリティ チェック用の認証情報の指定

caaipsecurity コマンドでは、独自の Web サービス セッション用の認証情報を指定できます。Web サービス セッションで caaipsecurity を使用する場合、その他の CLI コマンドでは認証情報のプロンプトは表示されません。

このコマンドの形式は、以下のとおりです。

```
caaipsecurity setcurrentsession  
[-prompt {yes|no}]  
[-ws_user username -ws_password password -locale iso639value]
```

```
caaipsecurity deletecurrentsession  
[-prompt {yes|no}]  
[-ws_user username -ws_password password -locale iso639value]
```

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

createdpmdefaultgroup コマンド -- サービスの作成または更新

`createdpmdefaultgroup` コマンドは、製品展開によって使用されるサーバでサービスを作成または入力します。サービスが存在しない場合は作成され、サーバが追加されます。管理対象外のサーバまたは **CA Virtual Assurance** によって検出されていないサーバを追加する場合、サーバは自動的に検出および管理されます。サービスがすでに存在する場合、サーバは既存のサービスに単に追加されます。

このコマンドの形式は、以下のとおりです。

```
createdpmdefaultgroup
```

例: サービスの入力

以下のコマンドには引数がありません。

```
createdpmdefaultgroup
```

dpmcmd run コマンド -- スクリプトの実行

`dpmcmd run` コマンドは、スクリプトを実行するためのコマンドライン インターフェースです。

このコマンドの形式は、以下のとおりです。

```
dpmcmd run -cmdline commandline [-pre] [-post][-locale iso639value]
```

-cmdline *commandline*

実行するスクリプト ファイルがあるパス、またはシェル コマンドを定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: スクリプトを実行し、イベントを作成する

以下の例は、**MyScript** という名前のスクリプトを実行し、スクリプトの実行前と実行後にイベントを作成します。

```
dpmcmd run -cmdline C:%scripts%\MyScript.bat -pre -post
```

dpmrpt schedulereport コマンド -- レポートのスケジューリング

dpmrpt schedulereport コマンドを使用すると、レポートをスケジュールできます。

このコマンドの形式は、以下のとおりです。

```
dpmrpt deliver [-sc SC URL] -host_name ホスト名 -port_number ポート番号 -cms_name CMS
名 -cms_port_number CMS ポート番号 -auth auth -bo_username BO ユーザ名 [-bo_password BO
パスワード] -report_name レポート名 -encrypted {yes|no} [-start_time 開始時刻]
[-end_time 終了時刻] [-server_group レポート サービス] [-system_name システム名]
[-metric_name メトリック名] [-no_server_group サービスの数] [-no_system システムの数]
[-pre] [-post]
```

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-host_name レポート サーバ名

ホスト レポート サーバの名前を定義します。

-port number ポート番号

レポート サーバの通信ポートを定義します。

-report_name レポート名

BusinessObjects レポート名を定義します。

-cms_port number CMS ポート番号

BusinessObjects CMS の通信ポートを定義します。

-auth {secEnterprise|secLDAP|secWinAD}

使用するレポート サーバ認証プロトコルを定義します。デフォルトはないため、プロトコルを指定する必要があります。 オプションは、以下のとおりです。

secEnterprise

セキュリティで保護されたエンタープライズ認証プロトコルを指定します。

secLDAP

セキュリティで保護された LDAP 認証プロトコルを指定します。

secWinAD

セキュリティで保護された Windows AD 認証プロトコルを指定します。

-bo_username ユーザ名

BusinessObjects のユーザ名を定義します。

-bo_password パスワード

(オプション) BusinessObjects のユーザのパスワードを定義します。

-report_name レポート名

BusinessObjects レポート名を定義します。

-encrypted={Yes|No}

ユーザ名とパスワードを暗号化するかどうかを指定します。

Yes

ユーザ名とパスワードを暗号化することを指定します。

No

ユーザ名とパスワードを暗号化しないことを指定します。

-start_time 開始時刻

(オプション) レポートの開始時刻を指定します。

制限: ミリ秒

-end_time 終了時刻

(オプション) レポートの終了時刻を指定します。

制限: ミリ秒

-server_group レポート サービス

(オプション) レポートのサービスを定義します。

-system_name システム名

(オプション) レポート用のシステム (サーバ) の名前を定義します。

-metric_name メトリック名

(オプション) レポート メトリックの名前を定義します。

-no_server_group サービス数

(オプション) レポート内のサービスの数を定義します。

-no_system システム数

(オプション) レポート内のシステム (サーバ) の数を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: レポートのスケジューリング

以下の例は、BusinessObjects レポートが指定の時刻に実行されるようにスケジュールします。

```
dpmrpt schedulereport -host_hostname dcamreporting -port_number 8081 -cms_name  
dcamreporting -cms_port_number 8081 -auth secEnterprise -bo_username Administrator  
-encrypted no -report_name -start_time 86400000 -end_time 86400000
```

dpmssh run コマンド -- リモート シェル コマンドまたはスクリプトの実行

dpmssh run コマンドを使用すると、Linux/UNIX ターゲット システムでリモートからのログインおよびシェル コマンドまたはスクリプトの実行が可能になります。この場合、セキュリティで保護された通信のために Secure Socket Shell (SSH) を使用します。

このコマンドの形式は、以下のとおりです。

```
dpmssh run -cmdline commandline -target_host ターゲット ホスト名 -target_user ターゲッ  
ト ユーザ名 -target_password ターゲット パスワード [-pre] [-post]
```

-cmdline *commandline*

実行するスクリプト ファイルがあるパス、またはシェル コマンドを定義します。

-target_host *ターゲット ホスト名*

Linux または UNIX ターゲット ホスト サーバの名前を定義します。

-target_user *ターゲット ユーザ名*

Linux または UNIX ターゲット ホスト サーバのユーザ名を定義します。

-target_password *ターゲット パスワード*

Linux または UNIX ターゲット ホスト サーバのユーザ パスワードを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

例: シェル コマンドの実行およびイベントの作成

以下の例は、`/tmp` ディレクトリに対して `ls` コマンドを実行します。

```
dpmssh run -cmdline "ls /tmp" -target_host linuxserver -target_user myusername  
-target_password pswd -pre -post
```

例: リモート スクリプトの実行

以下の例は、`MyScript` という名前のスクリプトを実行します。

```
dpmssh run -cmdline "/tmp/MyScript.sh" -target_host unixbox -target_user root  
-target_password pass1
```

ヘルプ デスク コマンド

CLI を使用すると、ヘルプ デスク コマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。

dpmhd addcomment コマンド -- チケットをコメントで更新する

`dpmhd addcomment` コマンドは、既存のヘルプ デスク チケットにコメントを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmhd addcomment [-sc sc_url] -id ticketid -value comment [-locale iso629value]  
-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-id *ticketid*

チケットをオープンしたときに、ヘルプ デスク システムによって割り当てられた一意の ID を定義します。

例 : cr:100400

-value *comment*

チケットを更新するコメント文字列を定義します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: チケットにコメントを追加する

以下の例は、SERVICEA のすべての仮想マシンで電源オフ操作が開始されたことがわかるように、ヘルプデスク チケット **cr:400335** を更新します。

```
dpmhd addcomment -id cr:400335 -value "Powering off SERVICEA"
```

dpmhd close コマンド -- ヘルプ デスク チケットのクローズ

dpmhd close コマンドは、ヘルプ デスク チケットをクローズします。

このコマンドの形式は、以下のとおりです。

```
dpmhd close [-sc sc_url] -id ticketid [-locale iso629value]
```


-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-id *ticketid*

チケットをオープンしたときに、ヘルプ デスク システムによって割り当てられた一意の ID を定義します。

例 : `cr:100400`

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ヘルプ デスク チケットをクローズする

以下の例は、チケット ID `cr:400335` で識別されるヘルプ デスク チケットをクローズします。

```
dpmhd close -id cr:400335
```

dpmhd getattribute コマンド -- チケットの属性値の取得

`dpmhd getattribute` コマンドは、既存のヘルプ デスク チケット属性の値を取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhd getattribute [-sc sc_url] -id ticketid -name attr_name [-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-id *ticketid*

チケットをオープンしたときに、ヘルプ デスク システムによって割り当てられた一意の ID を定義します。

例 : `cr:100400`

-name *attr_name*

チケット属性の名前を定義します。サポートされている属性名は以下のとおりです。

- `description`
- `status`
- `summary`

注: これらの属性名は大文字と小文字を区別します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: チケット属性の値を取得する

以下の例は、チケット ID `cr:400335` で識別されるヘルプ デスク チケットの `description` 属性の値を取得します。

```
dpmhd getattribute -id cr:400335 -name description
```

dpmhd getstatus コマンド -- チケットのステータスの取得

`dpmhd getstatus` コマンドは、既存のヘルプ デスク チケットのステータスを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmhd getstatus [-sc sc_url] -id ticketid [-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-id *ticketid*

チケットをオープンしたときに、ヘルプ デスク システムによって割り当てられた一意の ID を定義します。

例 : `cr:100400`

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: チケットのステータスを取得する

以下の例は、チケット ID `cr:400335` で識別されるヘルプ デスク チケットのステータスを取得します。

```
dpmhd getstatus -id cr:400335
```

dpmhd open コマンド -- ヘルプ デスク チケットのオープン

dpmhd open コマンドは、新しいヘルプ デスク チケットをオープンします。

このコマンドの形式は、以下のとおりです。

```
dpmhd open [-sc sc_url] -summary summarytext [-description descriptiontext]  
[-affected_user username] [-entity_name entityname] [-type tickettype] [-template  
templatename] [-pre] [-post][-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-summary *summarytext*

チケットをオープンする理由について、簡単な説明を定義します。

-description *descriptiontext*

(オプション) チケットをオープンする理由を指定します。

-entity_name *entityname*

(オプション) ヘルプ デスク システムで既知の設定項目とチケットの照合に使用するサーバまたはサービスの名前を定義します。設定項目のホスト名がエンティティ名と同じ場合、チケットはその設定項目に関連付けられます。

-affected_user *username*

(オプション) このチケットに関連付けられているエンド ユーザを定義します。

-type *tickettype*

(オプション) オープンするチケットのタイプを定義します。このパラメータは、**-template** パラメータと共に指定する必要があります。タイプを指定しない場合は、ヘルプ デスク システムで設定されているデフォルトのチケットが使用されます。サポート対象のチケットタイプは、ヘルプ デスク システムの設定によって決まります。たとえば、ITIL サポートの **Unicenter Service Desk** を設定する場合、変更要求、インシデント、問題、リクエストのチケットタイプがサポート対象となります。

-template *templatename*

(オプション) チケットをオープンするときに使用するテンプレートの名前を定義します。テンプレートにはデフォルト チケット設定が含まれます。テンプレート オプションを使用する場合、チケットタイプでテンプレートを特定できるようにする必要があります。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、**ISO 639_3166** の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmhd setattribute コマンド -- チケットの属性の設定

dpmhd setattribute コマンドは、既存のヘルプ デスク チケットの属性を更新します。

このコマンドの形式は、以下のとおりです。

```
dpmhd setattribute [-sc sc_url] -id ticketid -name attr_name -value attr_value
[-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : 80

-id *ticketid*

チケットをオープンしたときに、ヘルプ デスク システムによって割り当てられた一意の ID を定義します。

例 : `cr:100400`

-name *attr_name*

チケット属性の名前を定義します。サポートされている属性名は以下のとおりです。

- `description`
- `status`
- `summary`

注: これらの属性名は大文字と小文字を区別します。

-value *attr_value*

属性を更新する値を定義します。

注: チケット ステータスを更新するために `setattribute` を使用する場合は、ステータスに関連付けられている内部コードを入力します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: チケットの属性を更新する

以下の例は、チケット ID `cr:400335` で識別されるヘルプ デスク チケットの `description` 属性の値を設定します。

```
dpmhd setattribute -id cr:400335 -name description -value "Poweroff server1"
```

dpmhd setstatus コマンド -- チケットのステータスの設定

`dpmhd setstatus` コマンドは、既存のヘルプ デスク チケットのステータスを更新します。

このコマンドの形式は、以下のとおりです。

```
dpmhd setstatus [-sc sc_url] -id ticketid -value statuscode [-locale iso629value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-id *ticketid*

チケットをオープンしたときに、ヘルプ デスク システムによって割り当てられた一意の ID を定義します。

例: `cr:100400`

-value *statuscode*

チケットを更新するステータス コードを指定します。たとえば **Unicenter Service Desk** の場合、ステータスに関連付けられている内部コードを入力します。 **Unicenter Service Desk** チケットのステータス設定に使用する有効なコードには、以下のようなコードがあります。

APP

ステータスを承認済みに設定します。

CL

ステータスをクローズに設定します。

OP

ステータスをオープンに設定します。

REJ

ステータスを拒否に設定します。

RSCH

ステータスを調査中に設定します。

WIP

ステータスを作業中に設定します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: チケットのステータスを更新する

以下の例は、チケット ID **cr:400335** で識別されるヘルプ デスク チケットのステータスを承認済みに更新します。

```
dpmhd setstatus -id cr:400335 -value APP
```

イメージング コマンド

CLI を使用すると、イメージング コマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。

dpmimg image コマンド -- Solaris イメージを展開する

JumpStart サーバから Solaris クライアントに Solaris イメージを展開する場合、JumpStart インストール サーバとは異なる JumpStart ブート サーバを使用して Solaris クライアントのイメージする場合、または Solaris 10 x86 クライアントのイメージする場合には、`dpmimg image` コマンドを使用します。

このコマンドの形式は、以下のとおりです。

```
dpmimg image [-sc sc_url] -target_host targethostname -target_mac macaddress
[-itcm_server itcm domain manager] -auto_deploy {Yes|No} [-deploy_template
templatename] -target_username username [-target_password password] [-auth_file
authorizationfile] [-auth_comp componentID] -img_host imagehostname -img_name
imagename -img_location location [-boot_img_name bootimagename] [-boot_img_location
bootimagelocation] [-boot_img_host bootimagehost] [-servproc_hostname
serviceprochostname] [-servproc_user serviceprocusername] [-servproc_pw
serviceprocpassword] -dhcp_hostname dhcphostname [-itcm_server SD_adapter_server]
[-scalability_server scalabilityservername] [-wait [timeout]] [-pre] [-post]
[[-ws_user <value> -ws_password <value>]][-prompt <yes|no>]][-locale iso639value]
-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例：https://*hostname*:*port*/...

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例：localhost.ca.com

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例：80

-target_host *targethostname*

イメージを展開するターゲット ホスト サーバの名前を定義します。

-target_mac *macaddress*

イメージを展開するコンピュータのハードウェア アドレスを定義します。

`-itcm_server itcmdomainmanager`

(オプション) **Software Delivery** アダプタがある場合は、Unicenter DSM ドメインマネージャの名前を定義します。1 つの **Software Delivery** アダプタまたは Unicenter DSM ドメインマネージャのみを設定する場合はオプションです。

注: CA Server Automation でのみ有効です。

`-auto_deploy {yes|no}`

CA Virtual Assurance エージェントを自動的に展開するかどうかを指定します。オプションは、以下のとおりです。

yes

CA Virtual Assurance エージェントを自動的に展開します。

no

CA Virtual Assurance エージェントが自動的に展開されないようにします。

デフォルト: no

`-deploy_template templatenam`

(オプション) CA Virtual Assurance で作成された汎用テンプレートの名前を指定します。

注: VMware vCenter で作成および管理されているテンプレートと、このテンプレートを混同しないでください。

`-target_username targetusername`

イメージを展開するターゲット ホスト サーバにエージェントを展開するユーザ名を定義します。

`-target_password targetpassword`

(オプション) イメージを展開するターゲット ホスト サーバにエージェントを展開するパスワードを定義します。パスワードを指定しない場合は、認証ファイルから取得されます。

注: 認証ファイルの設定には dpmutil CLI を使用します。

-auth_file *authorizationfilename*

(オプション) 暗号化されたパスワードの取得に使用する認証ファイルの名前をフルパスで指定します。このファイルには、**dpmutil** 認証情報設定コマンドを使用して生成した **CA Virtual Assurance** 管理者の認証情報が含まれます。このオプションを指定しないと、デフォルトの認証ファイルが使用されます。

-auth_comp *componentID*

(オプション) グループ ホストとユーザに対して使用できるコンポーネント ID を指定します。

-img_host *RSIServer*

RSI サーバの名前を定義します。

-img_name *imagename*

展開するイメージの名前を定義します。

-img_location *imagelocation*

展開するイメージの場所を定義します。

-boot_img_name *bootimagename*

(オプション) 展開するブート イメージの名前を定義します。

-boot_img_location *bootimagelocation*

(オプション) 展開するイメージの場所を定義します。

-boot_img_host *bootimagehost*

(オプション) ブート イメージが存在するサーバの名前を定義します。

-servproc_hostname *serviceprochostname*

(オプション) サービス プロセッサのホスト名 (Solaris 10 x86 タイプ) を定義します。

-servproc_user *serviceprocusername*

(オプション) サービス プロセッサに接続するユーザのユーザ名 (Solaris 10 x86 タイプ) を定義します。

-servproc_pw *serviceprocpassword*

(オプション) サービス プロセッサに接続するユーザのパスワード (Solaris 10 x86 タイプ) を定義します。

-dhcp_hostname *dhcphostname*

DHCP サーバへの接続に使用されるホスト名 (Solaris 10 x86 タイプ) を定義します。

-scalability_server *servername*

(オプション) ソフトウェア配信用の Unicenter DSM スケーラビリティ サーバを指定します。

-wait *timeout*

(オプション) ジョブ ステータスを表示し、操作が完了、失敗、またはタイムアウト (*timeout* の値を指定している場合) するまで待機するかどうかを指定します。このオプションを指定しない場合、CLI は完了を待たずに戻ります。応答がない場合、オプションの *timeout* の値 (分) がタイムアウトするまでの時間となります。*timeout* の値を指定せずに **wait** オプションを入力すると、CLI は **caimgconf.cfg** ファイルのデフォルト待機時間の 120 分を使用します。任意の正の整数のほかに、*timeout* には以下の値も使用できます。

""

タイムアウト値はありません。

0

完了するまで待機します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) イメージング サービス モジュールに接続するための Web サービス ユーザ名を指定します。

-ws_password *wspassword*

(オプション) イメージング サービス モジュールに接続するための Web サービス ユーザのパスワードを指定します。

-prompt {yes|no}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : Yes

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: Solaris クライアント コンピュータにイメージを展開する

この例では、JumpStart サーバから Solaris クライアント コンピュータに Solaris イメージを展開しますが、イメージの展開が完了するのを待機しません。

```
dpmimg image -imghost sunserver001 -img_name jump_5.10 -auto_deploy Yes -target_host
sunserver002 -target_username rootuser -target_password rootuserpassword
-target_mac 00:00:00:00:00:00 -img_location /export/jump_5.10
```

例: Solaris 10 x86 クライアント コンピュータにイメージを展開する

この例では、JumpStart サーバから Solaris 10 x86 クライアント コンピュータに Solaris イメージを展開します。

```
dpmimg image -target_host sunserver001 -target_mac 00:00:00:00:00:00 -auto_deploy no
-target_username root -target_password rootuserpassword -img_host sunimageserver001
-img_name sol_10_508_x86 -img_location /jsimages/SunImageServer/sol_10_508_x86
-servproc_hostname sunserver002 -servproc_user root -servproc_pw rootuserpassword
-dhcp_hostname sunimageserver002 -boot_img_host sunimageserver001 -boot_img_name
sol_10_508_x86_bt -boot_img_location /jsboot/SunImageServer/sol_10_508_x86_bt
-wait 120 -pre -post
```

例: 複数の JumpStart サーバを使用して Solaris サーバに Solaris イメージを展開する

この例では、複数の JumpStart サーバを使用して、Solaris サーバ sunserver002 に Solaris イメージを展開します。インストール OS イメージは sunserver003 にあります。また、ブート イメージはブート サーバ sunserver004 にあります。コマンドは、イメージが展開されるのを 2 時間を待機してから実行されます。

```
dpmimg image -target_host sunserver002 -target_mac 00:00:00:00:00:00 -auto_deploy yes -target_username root -target_password rootuserpassword -img_host sunserver003 -img_name sol_10_sparc -img_location /jsimages/sol_10_sparc -boot_img_host sunserver004 -boot_img_name sol_10_sparc_boot -boot_img_location /jsboot/sol_10_sparc_boot -wait 120 -pre -post
```

例: 複数の JumpStart サーバを使用して Solaris 10 x86 サーバに Solaris 10 x86 イメージを展開する

この例では、複数の JumpStart サーバを使用して、Solaris 10 x86 サーバ sunserver002 に Solaris 10 x86 イメージを展開します。インストール OS イメージは sunserver002 にあります。また、ブート イメージはブート サーバ sunserver003 にあります。このイメージは、サービス プロセッサ サーバ sunserver004 および DHCP サーバ dhcpserver001 に接続します。

```
dpmimg image -target_host sunserver002m -target_mac 00:00:00:00:00:00 -auto_deploy no -target_username root -target_password rootuserpassword -img_host sunserver002 -img_name sol_10_x86 -img_location /jsimages/sol_10_x86 -servproc_hostname sunserver004 -servproc_user root -servproc_pw rootuserpassword -dhcp_hostname dhcpserver001 -boot_img_host sunserver003 -boot_img_name sol_10_x86_boot -boot_img_location /jsboot/sol_10_x86_boot -wait 120 -pre -post
```

dpmimg imgjobcheck コマンド -- Solaris イメージング ジョブのステータスを取得する

dpmimg imgjobcheck コマンドは、特定の CA Virtual Assurance ジョブ ID を持つ Solaris イメージ ジョブのステータスを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmimg imgjobcheck [-sc sc_url] -status jobID [-pre] [-post] [[-ws_user <value> -ws_password <value>]][-prompt <yes|no>]][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-status={*job ID*}

ジョブ ステータスの取得に使用するジョブ ID を定義します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *wsuser*

(オプション) イメージング サービス モジュールに接続するための Web サービス ユーザ名を指定します。

-ws_password *wspassword*

(オプション) イメージング サービス モジュールに接続するための Web サービス ユーザのパスワードを指定します。

-prompt {*yes|no*}

(オプション) Web サービスのユーザ ID とパスワードの入力を要求するかどうかを指定します。

デフォルト : `Yes`

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: Solaris プロビジョニング ジョブのジョブ ステータスを取得する

この例では、特定の Solaris プロビジョニング ジョブのジョブ ステータスを取得します。

```
dpmimg imgjobcheck -status 45
```

ポリシー コマンド

スクリプトに `dmpolicy CLI` を使用してポリシー コマンドを自動化し、コマンドの結果に基づいてアクションを実行できます。

`dmpolicy createdefaultactions` コマンド -- デフォルト アクションを作成する

`dmpolicy createdefaultactions` コマンドは、4 つのデフォルトの「ヘルス状態の設定」アクションを作成します。これらのアクションはインストール中に作成されます。このコマンドは、これらのアクションを再作成する場合 (たとえば、これらのアクションを誤って削除した後など) に使用します。

このコマンドの形式は、以下のとおりです。

```
dmpolicy createdefaultactions [-sc sc_url] [-ws_user username -ws_password password] [-locale iso639value]
```


-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

dpmpolicy exportaction コマンド -- アクションをファイルにエクスポートする

dpmpolicy exportaction コマンドは、アクションをファイルにエクスポートします。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy exportaction [-sc sc_url] [-action_name actionname] -file filename  
[-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-action_name *actionname*

アクションの名前を定義します。このパラメータを指定しない場合、すべてのアクションがエクスポートされます。

-file *filename*

ファイル名を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: アクションをファイルにエクスポートする

この例では、単一のアクションをファイルにエクスポートします。

```
dpmpolicy exportaction -action_name ACTION_A -file C:¥ACTION_A.txt
```

dpmpolicy exportrule コマンド -- ルールをファイルにエクスポートする

dpmpolicy exportrule コマンドは、ルールをファイルにエクスポートします。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy exportrule [-sc sc_url] [-rule_name rulename] -file filename [-ws_user  
username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-rule_name *rulename*

作成したルールに割り当てた名前を定義します。

-file *filename*

ファイル名を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ルールをファイルにエクスポートする

この例では、単一のルールをファイルにエクスポートします。

```
dpmpolicy exportrule -rule_name SERVER_RULE_A -file C:%SERVER_RULE_A.txt
```

例: すべてのルールをファイルにエクスポートする

この例では、すべてのルールをファイルにエクスポートします。

```
dpmpolicy exportrule -file C:%ALL_RULES.txt
```

dpmpolicy findbestmachine コマンド -- ベスト サーバを検索する

dpmpolicy findbestmachine コマンドは、指定されたサーバのリストから全体的な使用率が最も低いサーバを検索します。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy findbestmachine [-sc sc_url] -machine_list machinelist [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ：カンマ区切りのマシン名。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。**caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ベストサーバを検索する

この例では、指定されたサーバ リストから全体的な使用率が最も低いサーバを検索します。

```
dpmpolicy findbestmachine -machine_list "machine1.ca.com, machine2.ca.com, 172.24.36.107"
```

dpmpolicy findcandidatemachine コマンド -- 候補サーバを検索する

dpmpolicy findcandidatemachine コマンドは、ユーザ定義の仕様に該当するサーバを検索します。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy findcandidatemachine [-sc sc_url] [-machine_list machinelist] -constraint Source:OP:Value [-constraint Source:OP:Value] [-ws_user username -ws_password password] [-locale iso639value]
```

`-sc sc_url`

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

`-machine_list machinelist`

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ : カンマ区切りのマシン名。

-constraint={Source:OP:Value},...

候補サーバが満たす必要のある仕様を定義します。オプションは、以下のとおりです。

Source={MEMORY|CPUSPEED|STORAGE|ARCH}

制約および関連付けられた値のタイプを指定します。

MEMORY

メモリ制約を指定します。

値の範囲： MB

CPUSPEED

CPU 処理速度制約を指定します。

値の範囲： MHz

STORAGE

ストレージ制約を指定します。

値の範囲： MB

ARCH

ハードウェア制約を指定します。

値の範囲： X86、SPARC、PA-RISC、POWERPC、X64

OP={EQ|NEQ|GT|GTE|LT|LTE}

オペランドを指定します。以下のようなオペランドがあります。

EQ = 等しい

NEQ = 等しくない

GT = より大きい

GTE = 以上

LT = 未満

LTE = 以下

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サーバを検索する

この例では、システム メモリが 2000 MB を超えているサーバを検索します。

```
dpmpolicy findcandidatemachine -constraint MEMORY:GT:2000
```

例: リスト内のサーバを検索する

この例では、指定されたサーバリストの中から、システム メモリが 2000 MB を超え、CPU が 2400 MHz 以上のサーバを検索します。

```
dpmpolicy findcandidatemachine -constraint MEMORY:GT:2000 -constraint  
CPUSPEED:GTE:2400 -machine_list alert.ca.com, 141.202.224.59
```

dpmpolicy getruleexeccount コマンド -- ルール実行回数を取得する

dpmpolicy getruleexeccount コマンドは、ルールを実行しようとした回数を返します。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy getruleexeccount [-sc sc_url] -rule_name rulename [-ws_user username  
-ws_password password][-locale iso639value]
```


-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-rule_name *rulename*

作成したルールに割り当てた名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ルール実行回数を取得する

この例では、ルールが実行された回数を取得します。

```
dppolicy getruleexccount -rule_name GENERIC_RULE
```

dpmpolicy importaction コマンド -- アクションをファイルからインポートする

dpmpolicy importaction コマンドは、アクションをファイルからインポートします。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy importaction [-sc sc_url] -file filename [-action_name actionname]  
[-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : 80

-file *filename*

ファイル名を定義します。

-action_name *actionname*

アクションの名前を定義します。 ファイルに含まれるアクションが 1 つのみの場合、このオプションを使用してインポート時のアクション名を変更できます。 ファイルに複数のアクションが含まれている場合、この値は無視されます。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。 認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: アクションをファイルからインポートする

この例では、アクションをインポートし、その名前を変更します。

```
dpmpolicy importaction -file C:\ACTION_A.txt -action_name ACTION_B
```

dpmpolicy importrule コマンド -- ルールをファイルからインポートする

dpmpolicy importrule コマンドは、ルールをファイルからインポートします。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy importrule [-sc sc_url] [-entity_type entitytype -entity_name entityname]  
-file filename [-rule_name rulename] [-ws_user username -ws_password password]  
[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-entity_type={*server*|*service*}

このルールがサーバまたはサービスに関連付けられているかどうかを定義します。

-entity_name *entityname*

サーバまたはサービスの名前を定義します。

-file filename

ファイル名を定義します。

-rule_name rulename

作成したルールに割り当てた名前を定義します。ファイルに含まれるルールが 1 つのみの場合、このオプションを使用してインポート時のルール名を変更できます。ファイルに複数のルールが含まれている場合、この値は無視されます。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ルールをファイルからインポートする

この例では、ルールをインポートし、その名前を変更します。

```
dpmpolicy importrule -file C:\RULE_A.txt -rule_name RULE_B
```

dpmpolicy resetruleexeccount コマンド -- ルールの試行回数をリセットする

dpmpolicy resetruleexeccount コマンドは、ルールの試行回数をゼロにリセットします。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy resetruleexeccount [-sc sc_url] -entity_type entitytype -entity_name entityname -rule_name rulename [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-entity_type={*server*|*service*}

このルールがサーバまたはサービスに関連付けられているかどうかを定義します。

-entity_name *entityname*

サーバまたはサービスの名前を定義します。

-rule_name *rulename*

作成したルールに割り当てた名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サーバのしきい値をリセットする

この例では、サーバ上で実行するルールのしきい値をリセットします。

```
dmpolicy resetruleexccount -entity_type 0 -entity_name machine1.ca.com -rule_name SERVER1_RULE
```

例: サービス ルールの試行回数をリセットする

この例では、サービス上で実行するルールの再試行回数をゼロにリセットします。

```
dpmpolicy resetruleexceccount -entity_type 1 -entity_name SERVICE1 -rule_name  
SERVICE1_RULE
```

dpmpolicy runaction コマンド -- アクションを実行する

dpmpolicy runaction コマンドは、アクションおよびアクション シーケンスを実行します。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy runaction [-sc sc_url] -action_name action name [-event_source event  
source] [-event_message event_message] [-rule_name rule name] [-server_name server  
name] [-service_name service name] [-propagate] [-vm_name vm name] [-datacenter  
datacenter name] [-host_system host system] [-wait_rc] [-pre] [-post] [-ws_user  
username -ws_password password] [-locale iso639value] [-D-overwrite Parameter=value]  
-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-action_name *actionname*

アクションの名前を定義します。

-event_source *eventsources*

(オプション) %EVENTSOURCE% 代替変数に入力する値を定義します。

-event_message *eventmessage*

(オプション) %EVENTMESSAGE% 代替変数に入力する値を定義します。

-rule_name *rulename*

(オプション) %RULENAME% 代替変数に入力する値を定義します。

-server_name *servername*

(オプション) %SERVER% 代替変数に入力する値を定義します。

-service_name *servicename*

(オプション) %SERVICE% 代替変数に入力する値を定義します。

-propagate

(オプション) すべてのサーバで実行する、-service_name オプションで指定したサービスのアクションを指定します。

-vm_name *vmname*

(オプション) %VMNAME% 代替変数に入力する値を定義します。

-datacenter_name *datacentername*

(オプション) %DATACENTER% 代替変数に入力する値を定義します。

-host_system *hostsystem*

(オプション) %HOSTSYSTEM% 代替変数に入力する値 (VMware ESX ホスト コンピュータの名前) を定義します。

-wait_rc

(オプション) コマンドスクリプトの実行アクションが完了するのを待機するかどうかを指定します。

-pre

(オプション) 操作を実行する前にイベントを生成するかどうかを指定します。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

-D-overwrite *Parameter=value*

(オプション) 指定したパラメータの新しい値を定義します。

例: サーバをオンラインにするアクションを実行する

この例では、サーバをオンラインにするアクションを実行します。

```
dmpolicy runaction -action_name BringServerOnline -server_name Server001  
-service_name Production -pre -post
```

例: アクションを実行する

この例では、アクションを実行します。

```
dmpolicy runaction -action_name Action1
```

例: サービス内のすべてのサーバに対してアクションを実行する

この例では、**SERVICE1** サービス内のすべてのサーバに対してアクションを実行します。

```
dmpolicy runaction -action_name ACTION1 -service_name SERVICE1 -propagate
```

例: VM を必要とするアクションを実行する

この例では、**VM** を使用してアクションを実行します。

```
dmpolicy runaction -action_name vc_action -vm_name Finance¥Group¥FinVMServer1  
-datacenter FINANCE¥NYC
```


例: 異なるパラメータ値でアクションを実行する

この例では、CA Virtual Assurance ユーザ インターフェイスで設定された値とは異なる `vm_name` パラメータの値でアクションを実行します。

```
dpmpolicy runaction -action_name Create_snapshot -D-vm_name=VM02
```

dpmpolicy setruleexeclimit コマンド -- ルールの制限を設定する

`dpmpolicy setruleexeclimit` コマンドは、ルールの試行回数を決める制限を設定します。

このコマンドの形式は、以下のとおりです。

```
dpmpolicy setruleexeclimit [-sc sc_url] -entity_type entitytype -entity_name  
entityname -rule_name rulename -limit_option option [-limit_value value] [-ws_user  
username -ws_password password] [-locale iso639value]
```

`-sc sc_url`

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

`-entity_type={server|service}`

このルールがサーバまたはサービスに関連付けられているかどうかを定義します。

`-entity_name entityname`

サーバまたはサービスの名前を定義します。

`-rule_name rulename`

作成したルールに割り当てた名前を定義します。

-limit_option=[limited|unlimited|disabled]

ルールを実行する回数の制限オプションを指定します。オプションは以下のとおりです。

limited

ルールを実行する回数を指定します。

unlimited

ルールを無制限に実行できる場合に指定します。

disabled

ルール制限オプションを無効にする場合に指定します。

-limit_value value

(オプション) 制限値を定義します。このオプションは、制限オプションが **limited** に設定されている場合にのみ、有効となります。

値の範囲：0 以上

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ルールの試行回数を設定する

この例では、サービスで実行するルールの試行回数を 40 回に制限します。

```
dmpolicy setruleexecclimit -entity_type service -entity_name SERVICE1 -rule_name SERVICE1_RULE -limit_option limited -limit_value 40
```

例: ルールの試行回数を無制限に設定する

この例では、サーバ上でルールを実行できる回数を無制限に設定します。

```
dmpolicy setruleexecclimit -entity_type server -entity_name machine1.ca.com -rule_name MACHINE1_RULE -limit_option unlimited
```

例: 制限オプションを無効にする

この例では、このルールの制限オプションを無効にします。

```
dmpolicy setruleexeclimit -entity_type server -entity_name machine2.ca.com  
-rule_name MACHINE2_RULE -limit_option disabled
```

ポリシー設定

caismutlity CLI を使用すると、[ポリシー設定] ユーザ インターフェースに新しい機能を追加することができます。

caismutlity コマンド -- [ポリシー設定] UI への機能の追加

caismutlity コマンドを使用すると、ユーザ インターフェースを設定する機能ポリシーが提供されます。機能ポリシーには以下のものが含まれます。

- [ポリシー設定] テンプレートとして、CA Virtual Assurance に既存の eHealth テンプレートをインポートします。[ポリシー設定] テンプレートは CA Virtual Assurance UI 上で更新され、SystemEDGE エージェントに適用されます。
- 以下の条件の場合に、新しい配布サーバにレポートするよう SystemEDGE に指示します。
 - 古い配布サーバが廃棄されている。
 - 配布サーバが追加されている。
 - 別の配布サーバがエージェントを管理する別の場所に移動されている。
- 以下の条件の場合に、新しいドメイン サーバにレポートするよう SystemEDGE に指示します。
 - 古いドメイン サーバが廃棄されている。
 - CA Virtual Assurance で並列アップグレードが実行されている。
- 以下の条件の場合に、SystemEDGE ポリシーとテンプレート、SRM ポリシーをドメイン サーバにインポートおよびエクスポートします。
 - 新しいポリシーとテンプレートが、実稼働コンピュータに移行されるテストコンピュータに作成されている。
 - CA Virtual Assurance で並列アップグレードが実行されている。

このコマンドの形式は、以下のとおりです。

```
caismutility
[-f server]
[-o sysEDGE|sysEDGESRM]
[-l]
[-d -p policy]
[-importEH dir|file]
[-useDirName]
[-overwrite]
[-c manager agent1]
[-user user]
[-password password]
[-locale locale]
[-import dir|file]
[-type SEPolicy|SETemplate|SRMPolicy|SRMThreshold|SRMTest|auto]
[-export dir]
```

注: オプションはすべて大文字と小文字を区別します。

-f *server*

ドメイン サーバに接続するように指定します。

-o {*sysEDGE|sysEDGESRM*}

使用する設定のタイプを指定します。

-l

利用可能なオブジェクトを一覧表示します。

-d {-p *policy*}

エージェントにポリシーを配信します。

-importEH {*dir|file*}

eHealth テンプレートをインポートします。

-useDirName

親ディレクトリの名前を指定します。

-overwrite

既存のテンプレートに上書きするように指定します。

-c *manager {agent1 ...}*

新しいマネージャにレポートするようにエージェントに指示します。

-user *user*

ユーザ名を指定します。

-password *password*

パスワードを指定します。

-locale *iso639value*

デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用するには、**"native"** を指定します。

-import {*dir*|*file*}

ファイルストアに追加するポリシーまたはテンプレートを指定します。

-type {*SEPolicy*|*SETemplate*|*SRMPolicy*|*SRMThreshold*|*SRMTest*|*auto*}

インポートするファイルのタイプを指定します。

-export *dir*

指定したディレクトリにポリシーおよびテンプレートをすべてコピーします。

例: 1 セットの eHealth テンプレートをインポートする方法

この例では 1 セットの eHealth テンプレートをインポートします。

```
caismutility -user xxx -password yyy -importEH c:%templates%
```

例: ドメイン サーバに SystemEDGE ポリシーをインポートする方法

この例では、ドメイン サーバに一部の SystemEDGE ポリシーをインポートします。

```
caismutility -user xxx -password yyy -type SEPolicy -import c:%policies
```

例: 新しいマネージャにレポートするようにエージェントに指示する

この例では、新しいマネージャ **manager.com** にホスト **agent.com** のエージェントを移動させます。

```
caismutility -user xxx -password yyy -o sysEDGE -c manager.com agent.com
```

リソース マネージャ コマンド

dpmresourcemgr CLI を使用すると、リソース マネージャ コマンドをスクリプト化および自動化し、コマンドの結果に基づいてアクションを実行できます。

dpmresourcemgr addbaremetal コマンド -- 物理コンピュータを追加する

addbaremetal コマンドは、データ センターに物理コンピュータを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr addbaremetal [-sc sc_url] -server_name hostname -MAC_address  
macaddress [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-host_name *source host name*

操作するソース サーバの名前を定義します。

-MAC_address *macaddress*

ターゲット コンピュータの **MAC** アドレスを定義します。有効なエントリ : 次の形式の 12 桁の 16 進数値 (たとえば A ~ F、0 ~ 9、a ~ f) : `0013724C2140` または `00:13:72:4C:21:40`。

-ws_user *username* -ws_password *password*

(オプション) **Web** サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: 物理コンピュータを追加する

この例では、MAC アドレス **00:13:72:4C:21:40** を持つコンピュータ **baremetal2** を追加します。

```
dpmresourcemgr addbaremetal -sc https://servicecontroller/dpm/sc -server_name  
baremetal2 -MAC_address 00:13:72:4C:21:40 -ws_user wsuser -ws_password  
wsuserpassword
```

dpmresourcemgr addcomputertonetwork コマンド -- ネットワークにサーバを追加する

dpmresourcemgr addcomputertonetwork コマンドは、既存のネットワークにサーバを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr addcomputertonetwork [-sc sc_url] -vlan_id vlanid -mac_list  
macaddresses [-vlan_tagged] [-ws_user username -ws_password password] [-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-vlan_id *vlanid*

仮想ネットワーク (VLAN) を示す整数値を定義します。有効なエントリ：1 ～ 4094 の整数。1 はデフォルトのネットワーク インターフェイス カード (NIC) を示します。

-mac_list *macaddresses*

サーバの NIC MAC アドレスのリストを定義します。有効なエントリ：次の形式の NIC MAC アドレスのカンマ区切りのリスト：

XX:XX:XX:XX:XX:XX。

-vlan_tagged

VLAN をタグ付けするように定義します。指定しない場合、VLAN はタグ付けされません。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンド プロンプトのロケールを使用する場合は「native」を指定します。

例: ネットワークにサーバを追加する

この例では、VLAN 4556 にサーバを追加します。

```
dpmresourcemgr addcomputertonetwork -vlan 4556 -mac_list 11:22:33:44:55:66 -ws_user  
wsuser -ws_password wsuserpassword
```

dpmresourcemgr addipaddresspool コマンド -- IP アドレス プールを追加する

dpmresourcemgr addipaddresspool コマンドは、静的 IP アドレスまたは DHCP を使用して IP アドレス プールを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr addipaddresspool [-sc sc_url] -subnet_address address -starting_ip  
ipaddress -ending_ip ipaddress -ip_assignment_type static|dhcp [-name name][-owner  
ownername][-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-subnet_address *address*

ネットワークの IP アドレスを定義します。有効なエントリ：
`xxx.xxx.xxx.xxx` または `xxx.xxx.xxx.xxx/xx`。

-starting_ip *ipaddress*

静的 IP アドレス プールで使用する IP 範囲の有効な開始 IP アドレスを定義します。有効なエントリ：`xxx.xxx.xxx.xxx`。

-ending_ip *ipaddress*

IP 範囲の終了 IP アドレスを定義します。有効なエン트리：
xxx.xxx.xxx.xxx。

-ip_assignment_type static | dhcp

サーバで IP アドレス（静的 IP アドレスまたは DHCP）の取得に使用するメソッドを定義します。

-name *name*

アイテム名を定義します。有効なエン트리：1 ～ 128 文字の英数字。

-owner *ownername*

IP アドレス プールの所有者を定義します。有効なエン트리：1 ～ 128 文字の英数字。

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は fr_FR）で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: IP アドレス プールを追加する

この例では、10.10.21.130 ～ 10.10.21.140 の IP アドレス範囲を使用して IP アドレス プール engpool を追加し、IP 割り当てメソッドを DHCP に設定します。

```
dpmresourcemgr addipaddresspool -subnet_address 10.10.21.0 -starting_ip  
10.10.21.130 -ending_ip 10.10.21.140 -ip_assignment_type DHCP -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr addmachinestosome コマンド -- サービスにサーバを追加する

addmachinestosome コマンドは、サービスにサーバを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr addmachinestosome [-sc sc_url] -service_name servicename  
-machine_list machinelist [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-service_name *servicename*

サービスの名前を定義します。有効なエントリ : サービス名へのフルパス。親サービスとサブサービスを区切るには、円記号¥を使用します。

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ : カンマ区切りのマシン名。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービスにサーバを追加する

この例では、**SERVICEA** という名前のサービスにサーバのリストを追加します。

```
dpmresourcemgr addmachinestoservice -service_name SERVICEA -machine_list  
"machine4.MyCompany.com, machine5.MyCompany.com, 192.168.255.255" -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr addtocontainer コマンド -- コンテナへの内容物の追加

dpmresroucemgr addtocontainer コマンドは、コンテナにサーバまたはサブコンテナを追加します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr addtocontainer [-sc sc_url] -container_name containername  
[-machine_list machinelist|-subcontainer_name subcontainername] [-ws_user username  
-ws_password password][-locale iso639value]
```

-container_name コンテナ名

オブジェクトのコレクションを定義します。有効なエントリ: 親コンテナと子コンテナの区切りに円記号 (¥) を使用した、コンテナ名のフルパス。

-machine_list *machinelist* | -subcontainer *subcontainername*

マシン リストまたはサブコンテナとそのオブジェクトを定義します。

-ws_user *username* -ws_password *password*

(オプション) **Web** サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: コンテナへの内容物の追加

この例では、**Managed** という名前のコンテナに、**ManagedVMs** という名前のサブコンテナを追加します。

```
dpmresourcemgr addtocontainer -container_name Managed -subcontainer_name ManagedVMs
```

dpmresourcemgr createcontainer コマンド -- コンテナの作成

createcontainer コマンドは、他のオブジェクトの集合を保持するためのオブジェクトを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr createcontainer [-sc sc_url] -parent_container parentcontainer  
-container_name containername -machine_list machinelist [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-parent_container *parentcontainer*

親サービスのフルパス サービス名を定義します。

-container_name コンテナ名

オブジェクトのコレクションを定義します。有効なエントリ：親コンテナと子コンテナの区切りに円記号 (¥) を使用した、コンテナ名のフルパス。

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ：カンマ区切りのマシン名。

-ws_user *username* **-ws_password** *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: コンテナの作成

この例では、**ManagedVMs** という名前のコンテナを削除します。

```
dpmresourcemgr createcontainer -parent_container Managed -container_name ManagedVMs
```

dpmresourcemgr createdefaultservice コマンド -- デフォルト サービスを作成する

createdefaultservice コマンドは、ローカル システム、およびインストール中に設定された任意の外部システムにデフォルト サービスを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr createdefaultservice [-sc sc_url] [-retry retrynumber]  
[-discover_local_subnet] [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-retry *retrynumber*

サービスを作成する場合の再試行回数を定義します。

-discover_local_subnet

指定された場合、デフォルト サービスを作成した後にローカル サブネットを検出します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: デフォルト サービスを作成する

この例では、10 回接続を試行し、自動的にローカル サブネットを検出するデフォルト サービスを作成します。

```
dpmresourcemgr createdefaultservice -retry 10 -discover_local_subnet -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr createipnetwork コマンド -- ネットワークを作成する

dpmresourcemgr createipnetwork コマンドは、予約マネージャ で使用するデータセンターの既存のネットワークを指定します。その後で、予約マネージャ のエンドユーザは、特定のネットワークの仮想マシンを予約できます。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr createipnetwork [-sc sc_url] -subnet_address subnetaddress -vlan_id vlanid -ip_netmask ipnetmask -name networkname -ip_gateway_list ipaddresses [-dns_domainname domainname [-description description] [-dns_list ipaddresses][-dns_suffix_list suffixes][-netbios_server_list ipaddresses][-ntp_server_list ipaddresses][-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-subnet_address *address*

ネットワークの IP アドレスを定義します。有効なエン트리 : `xxx.xxx.xxx.xxx` または `xxx.xxx.xxx.xxx/xx`。

-vlan_id *vlanid*

仮想ネットワーク (VLAN) を示す整数値を定義します。有効なエン트리 : 1 ~ 4094 の整数。1 はデフォルトのネットワーク インターフェース カード (NIC) を示します。

-ip_mask *ipnetmask*

サブネット マスクを定義します。有効なエン트리 : `xxx.xxx.xxx.x` 形式 または CIDR 形式 (`xxx.xxx.xxx.x/8` ~ `28`) の標準の IP ネットマスク。プレフィックスの長さは 8 ~ 28 です。

-name *name*

アイテム名を定義します。有効なエントリ：1 ～ 128 文字の英数字。

-ip_gateway_list *ipaddresses*

カンマ区切りのリストを使用して、デフォルト ゲートウェイおよび代替ゲートウェイの IP アドレスを定義します。デフォルト ゲートウェイおよび代替ゲートウェイは同じネットワーク上にある必要があります。有効なエントリ：xxx.xxx.xxx.xxx。

-dns_domainname *domainname*

ネットワークの DNS ドメイン名を定義します。有効なエントリ：ドット区切りのアドレス（例：xxx.yyy.com）ごとに 1 ～ 63 文字の英数字。アドレスを数字で始めることはできません。

-description *description*

ユーザがわかるようにネットワークの説明を定義します。有効なエントリ：1 ～ 128 文字の英数字。

-dns_list *ipaddresses*

優先 DNS サーバおよび代替 DNS サーバの DNS リストと、プライマリおよびセカンダリ Windows インターネット ネーム サービス (WINS) サーバを定義します。有効なエントリ：次の形式の IP アドレスのカンマ区切りのリスト：xxx.xxx.xxx.xxx。

-dns_suffix_list *suffixes*

ネットワークの DNS サフィックスのリストを定義します。有効なエントリ：次の形式のサフィックスのカンマ区切りのリスト：
name.name.com。

-netbios_server_list *ipaddresses*

NetBIOS ネーム サーバの IP アドレスのリストを定義します。有効なエントリ：次の形式の IP アドレスのカンマ区切りのリスト：
xxx.xxx.xxx.xxx。

-ntp_server_list *ipaddresses*

ネットワーク タイム プロトコル (NTP) サーバを定義します。有効なエントリ：次の形式の NTP サーバのカンマ区切りのリスト：
xxx.xxx.xxx.xxx。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネットワークを作成する

この例では、仮想マシンを予約するときに 予約マネージャ エンド ユーザ ポータルが使用する既存のデータセンター ネットワーク **san diego 209** サブネットを指定します。

```
dpmresourcemgr createipnetwork -subnet_address 10.10.21.0 -vlan_id 4001 -ip_mask 255.255.255.0 -name "san diego 209 subnet" -ip_gateway_list 10.10.21.124,10.10.21.125 -dns_domainname turbo.com -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr createresourcegroup コマンド -- リソース グループの作成

createresourcegroup コマンドは、パッケージ グループを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr createresourcegroup [-sc sc_url] -group_name groupname -group_description description -group_type type -group_attribute attribute -package_list packagelist [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-group_name *グループ名*

グループの名前を定義します。

-group_description *説明*

グループの説明を定義します。

-group_type *タイプ*

リソース グループのタイプを定義します。 パッケージ グループの場合は `1` に設定します。 すべてのグループの場合は `0` に設定します。

-group_attribute *属性*

グループの属性を定義します。

-package_list *packagelist*

個々のパッケージを定義します。

Example: `-package_list "<package name A>|<package procedure A>|<procedure type A>|<package name B>|..."`.

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。 認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: リソース グループの作成

この例では、リソース グループを作成します。

```
dpmresourcemgr createresourcegroup -group_name "AGENT PACKAGE GROUP"
-group_description "Agent package group" -group_type 1 -group_attribute 0
-package_list "CA Product - Win r11.2|Install|INSTALL|Generic Agent Win32
r5.0|Install|INSTALL"
```

dpmresourcemgr createresourcetemplate コマンド -- サーバグループ テンプレートの作成

createresourcetemplate コマンドは、パッケージ テンプレートを作成します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr createresourcetemplate [-sc sc_url] -template_name templatename
-template_description description -template_os_type OSType [-group_list grouplist]
[-package_list packagelist][-ws_user username -ws_password password][-locale
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-template_name *templatename*

テンプレートの名前を定義します。

-template_description *説明*

テンプレートの説明を定義します。

-template_os_type *OS タイプ*

展開用のリソース テンプレートの OS タイプを定義します。

-group_list *グループリスト*

リソース テンプレートに追加するグループのリストを定義します。有効なエントリ：
-group_list "<グループ名 A>|<グループ タイプ 1= ソフトウェア パッケージ グループ>|<グループ名 B>|..."

-package_list *パッケージリスト*

個々のパッケージを定義します。グループ タイプを 1（パッケージ）に設定する場合、このオプションを指定します。有効なエントリ：
-package_list "<パッケージ名 A>|<パッケージ プロシージャ A>|<プロシージャ タイプ A>|<パッケージ名 B>|..."

-ws_user *username* **-ws_password** *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は fr_FR）で指定します。コマンド プロンプトのロケールを使用する場合は「native」を指定します。

例: リソース テンプレートの作成

この例では、パッケージ テンプレートを作成します。

```
dpmresourcemgr createresourcetemplate -template_name "Generic template"
-template_description "Generic template description" -template_os_type 0 -group_list
"Package group A|1|Package group B|1" -package_list "CAProduct - Win
r11.2|Install|INSTALL|Generic Agent Win32 r5.0|Install|INSTALL"
```

dpmresourcemgr createservice コマンド -- サービスを作成する

`createservice` コマンドは、サービスを作成します。1 つのサービスは複数のサブ サービスを持つことができます。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr createservice [-sc sc_url] -service_name servicename [-machine_list machinelist] [-lower_threshold lowerthreshold] [-upper_threshold upperthreshold] [-lag lag] [-priority priority] [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-service_name *servicename*

サービスの名前を定義します。有効なエントリ : サービス名へのフルパス。親サービスとサブサービスを区切るには、円記号 `¥` を使用します。

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ : カンマ区切りのマシン名。

-lower_threshold *lowerthreshold*

(オプション) サービスの全体的な使用率の下限しきい値を指定します。下限しきい値は上限しきい値未満である必要があります。有効なエントリ : `1 ~ 100 (%)`

-upper_threshold upperthreshold

(オプション) サービスの全体的な使用率の上限しきい値を指定します。上限しきい値は常に下限しきい値を超えている必要があります。
有効なエントリ：1～100 (%)

-lag lagvalue

(オプション) 遅延値を指定します。有効なエントリ：0 以上 (秒)。

-priority priorityvalue

(オプション) サービスの優先度を指定します。有効なエントリ：1 以上。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。**caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービスを作成する

この例では、**SERVICE1** という名前のサービスを作成します。

```
dpmresourcemgr createservice -service_name SERVICE1 -machine_list
"ComputerList.MyCompany.com, 172.31.255.255, Computer1.MyCompany.com"
-lower_threshold 20 -upper_threshold 80 -lag 1 -priority 1 -ws_user wsuser
-ws_password wsuserpassword
```

dpmresourcemgr deletebaremetal コマンド -- 物理コンピュータを削除する

deletebaremetal コマンドは、データ センターから物理コンピュータを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr deletebaremetal [-sc sc_url] -server_name hostname -MAC_address
macaddress [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-host_name *source host name*

操作するソース サーバの名前を定義します。

-MAC_address *macaddress*

ターゲット コンピュータの MAC アドレスを定義します。有効なエントリ : 次の形式の 12 桁の 16 進数値 (たとえば A ~ F、0 ~ 9、a ~ f) : `0013724C2140` または `00:13:72:4C:21:40`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: 物理コンピュータを削除する

この例では、データ センターから `EngServer1` サーバを削除します。

```
dpmresourcemgr deletebaremetal -server EngServer1 -MAC_address 11:22:33:44:55:66  
-ws_user wsuser -ws_password wsuserpassword
```


dpmresourcemgr deletecontainer コマンド -- コンテナの削除

deletecontainer コマンドは、コンテナとその内容物を削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr deletecontainer [-sc sc_url] -container_name containername  
[-recursive] [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例：https://*hostname*:*port*/...

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例：localhost.ca.com

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例：80

-container_name *コンテナ名*

オブジェクトのコレクションを定義します。有効なエントリ：親コンテナと子コンテナの区切りに円記号 (¥) を使用した、コンテナ名のフルパス。

-recursive

設定した場合、コマンドはサブ サービスまたはコンテナへと再帰的に実行されます。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: コンテナの削除

この例では、**ManagedVMs** という名前のコンテナを削除します。

```
dpmresourcemgr deletecontainer -container_name ManagedVMs.
```

dpmresourcemgr deleteipnetworks コマンド -- ネットワークを削除する

dpmresourcemgr deleteipnetworks コマンドは、CA Virtual Assurance からネットワークを削除します。ただし、そのネットワークを使用する現在または既存の予約がない場合にのみ削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr deleteipnetworks [-sc sc_url] -subnet_address_list subnetaddress [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-subnet_address_list *addresses*

IP アドレスを使用してネットワークのリストを定義します。有効なエントリは、`xxx.xxx.xxx.xxx` または `xxx.xxx.xxx.xxx/xx` の形式のカンマ区切りの IP アドレスです。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネットワークを削除する

この例では、CA Virtual Assurance によって使用される **san diego 209** サブネットを削除します。

```
dpmresourcemgr deleteipnetworks -subnet_address_list 10.10.21.0,10.10.21.130  
-ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr deletesystem コマンド -- システムを削除する

deletesystem コマンドは、データ センターからサーバを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr deletesystem [-sc sc_url] -machine_list machinelist [-post] [-ws_user  
username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ: カンマ区切りのマシン名。

-post

(オプション) 操作を実行した後でイベントを生成するかどうかを指定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: システムを削除する

この例では、**ServerB** という名前のサーバを削除します。

```
dpmresourcemgr deletesystem -machine_list "machine4.MyCompany.com,  
machine5.MyCompany.com, 192.168.255.255" -ws_user wsuser -ws_password  
wsuserpassword
```

dpmresourcemgr getdiscoverednetworks コマンド -- 検出されたネットワークを取得する

getdiscoverednetworks コマンドは、検出されたすべてのネットワークのリストを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr getdiscoverednetworks [-sc sc_url][-ws_user username -ws_password  
password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: すべてのネットワークを取得する

この例では、検出されたすべてのネットワークのリストを取得します。

```
dpmresourcemgr getdiscoverednetworks -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getipaddresspools コマンド -- IP アドレス プールをリスト表示する

`dpmresourcemgr getipaddresspools` コマンドは、所有者または割り当てタイプごとに既存の IP アドレス プールをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr getipaddresspools [-sc sc_url] [-subnet_address subnetaddress] [-owner ownername]  
[-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-subnet_address *address*

ネットワークの IP アドレスを定義します。有効なエントリ :
`xxx.xxx.xxx.xxx` または `xxx.xxx.xxx.xxx/xx`。

-owner *ownername*

IP アドレス プールの所有者を定義します。有効なエントリ : 1 ~ 128 文字の英数字。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: IP アドレス プールを更新する

この例では、IP アドレス 10.10.21.30 のネットワークの既存の IP アドレス プールをリスト表示します。

```
dpmresourcemgr getipaddresspools -subnet_address 10.10.21.130 -owner Reservation Manager -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getipnetworks コマンド -- ネットワークをリスト表示する

dpmresourcemgr getipnetworks コマンドは、createipnetwork コマンドを使用して、CA Virtual Assurance に指定されたネットワークをリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr getipnetworks [-sc sc_url][-ws_user username -ws_password password][-locale iso639value]
```

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネットワークをリスト表示する

この例では、CA Virtual Assurance によって使用されるネットワークをリスト表示します。

```
dpmresourcemgr getipnetworks -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getserverlist コマンド -- サーバのリストを取得する

getserverlist コマンドは、データ センターからサーバのリストを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr getserverlist [-sc sc_url] -service_name servicename [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-service_name *servicename*

サービスの名前を定義します。有効なエントリ: サービス名へのフルパス。親サービスとサブサービスを区切るには、円記号 **¥** を使用します。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サーバのリストを取得する

この例では、サービス **SERVICE2** に属するサーバのリストを取得します。

```
dpmresourcemgr getserverlist -service_name SERVICE2 -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getservicelist コマンド -- サービスのリストを取得する

getserverlist コマンドは、サーバからサービスのリストを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr getservicelist [-sc sc_url][-parent_service_name  
parentservice][-server_name servername][-recursive][-ws_user username -ws_password  
password][-locale iso639value]
```

-sc sc_url

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-parent_service_name *parentservicename*

親サービスを定義します。有効なエントリ：親サービスへのフルパス サービス。

-server_name *servername*

サーバの名前を定義します。指定しない場合、このコマンドは、定義されているすべてのサービスのリストを返します。

-recursive

設定された場合、この関数はサブサービスに対して再帰的に実行されます。

-ws_user *username* **-ws_password** *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービスのリストを取得する

この例では、親サービス **SERVICE2**、およびサーバ **EngServer1** に属するサービスのリストを取得します。

```
dpmresourcemgr getservicelist -parent service_name SERVICE2 -server_name EngServer1  
-ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr getsubnetdefaults コマンド -- サブネットのデフォルト値をリスト表示する

dpmresourcemgr getsubnetdefaults コマンドは、現在のネットワークのデフォルト値をリスト表示します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr getsubnetdefaults [-sc sc_url] -subnet_address address [-ws_user  
username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-subnet_address *address*

ネットワークの IP アドレスを定義します。有効なエントリ:
`xxx.xxx.xxx.xxx` または `xxx.xxx.xxx.xxx/xx`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ネットワークのデフォルト値をリスト表示する

この例では、IP アドレスが `10.10.21.130` のネットワークのデフォルト値を更新します。

```
dpmresourcemgr getsubnetdefaults -subnet_address 10.10.21.130 -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr removecomputerfromipnetwork コマンド -- ネットワークからサーバを削除する

`dpmresourcemgr removecomputerfromipnetwork` は、ネットワークからサーバを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr removecomputerfromipnetwork [-sc sc_url] -vlan_id vlanid -mac_list macaddresses [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-vlan_id *vlanid*

仮想ネットワーク (VLAN) を示す整数値を定義します。有効なエントリ : 1 ~ 4094 の整数。1 はデフォルトのネットワーク インターフェース カード (NIC) を示します。

-mac_list *macaddresses*

サーバの NIC MAC アドレスのリストを定義します。有効なエントリ : 次の形式の NIC MAC アドレスのカンマ区切りのリスト :

`xx:xx:xx:xx:xx:xx`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネットワークからサーバを削除する

この例では、VLAN 4556 からサーバ 11:22:33:44:55:66 を削除します。

```
dpmresourcemgr removecomputerfromipnetwork -vlan 4556 -mac_list 11:22:33:44:55:66  
-ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr removefromcontainer コマンド -- コンテナの内容物の削除

removefromcontainer コマンドは、マシンのリストまたはサブコンテナをコンテナから削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr removefromcontainer [-sc sc_url] -container_name containername  
[-machine_list machinelist|-subcontainer_name subcontainername] [-ws_user username  
-ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-container_name コンテナ名

オブジェクトのコレクションを定義します。有効なエントリ: 親コンテナと子コンテナの区切りに円記号 (¥) を使用した、コンテナ名のフルパス。

-machine_list machinelist | -subcontainer subcontainername

マシンのリストまたはサブコンテナとそのオブジェクトを定義します。
有効なエントリ： サブコンテナのフルパス コンテナ。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: コンテナの内容物の削除

この例では、**Managed** という名前のコンテナから、**ManagedVMs** という名前のサブコンテナを削除します。

```
dpmresourcemgr removefromcontainer -container_name Managed -subcontainer_name ManagedVMs.
```

dpmresourcemgr removeipaddresspool コマンド -- IP アドレス プールを削除する

dpmresourcemgr removeipaddresspool コマンドは、IP アドレス プールを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr removeipaddresspool [-sc sc_url] -name name|-starting_ip ipaddress [-ws_user username -ws_password password] [-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: 80

-name *name* | -starting_ip *ipaddress*

名前または開始 IP アドレス範囲を定義します。有効なエントリ: 1 ~ 128 の英数字、または `xxx.xxx.xxx.xxx`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: IP アドレス プールを削除する

この例では、アドレスが 10.10.21.130 で始まる IP アドレス プールを削除します。

```
dpmresourcemgr removeipaddresspool -starting_ip 10.10.21.130 -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr removemachinesfromservice コマンド -- サービスからサーバを削除する

removemachinesfromservice コマンドは、サービスからサーバを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr removemachinesfromservice [-sc sc_url] -service_name servicename  
-machine_list machinelist [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-service_name *servicename*

サービスの名前を定義します。有効なエントリ : サービス名へのフルパス。親サービスとサブサービスを区切るには、円記号 ¥ を使用します。

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ : カンマ区切りのマシン名。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービスからサーバを削除する

この例では、サービス **SERVICE2** から 2 つのサーバを削除します。

```
dpmresourcemgr removemachinesfromservice -service_name SERVICE2 -machine_list  
"machine1.MyCompany.com, machine2.MyCompany.com" -ws_user wsuser -ws_password  
wsuserpassword
```

dpmresourcemgr removeresourcegroup コマンド -- リソース グループの削除

removeresourcegroup コマンドは、リソース グループを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr removeresourcegroup [-sc sc_url] -group_name groupname -group_type  
grouptype [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-group_name *グループ名*

グループの名前を定義します。

-group_type *タイプ*

リソース グループのタイプを定義します。 パッケージ グループの場合は **1** に設定します。 すべてのグループの場合は **0** に設定します。

-ws_user *username* **-ws_password** *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。 認証情報が含まれていない場合、入力が必要になります。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。 コマンド プロンプトのロケールを使用する場合は「**native**」を指定します。

例: リソース グループの削除

この例では、**Agent Package Group** という名前のリソース グループを削除します。

```
dpmresourcemgr removeresourcegroup -group_name "AGENT PACKAGE GROUP" -group_type 1
```

dpmresourcemgr removeservice コマンド -- サービスを削除する

removeservice コマンドは、データ センターからサービスを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr removeservice [-sc sc_url] -service_name servicename  
[-recursive][-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-service_name *servicename*

サービスの名前を定義します。有効なエントリ: サービス名へのフルパス。親サービスとサブサービスを区切るには、円記号 `¥` を使用します。

-recursive

設定された場合、このコマンドはサブサービスに対して再帰的に実行されます。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: サービスを削除する

この例では、`TESTGROUP` という名前のサービスを削除します。

```
dpmresourcemgr removeservice -service_name TESTGROUP -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr removeresourcetemplate コマンド -- リソース テンプレートの削除

`removeresourcetemplate` コマンドは、パッケージ テンプレートを削除します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr removeresourcetemplate [-sc sc_url] -template_name templatename  
-template_os_type OSType [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-template_name *templatename*

テンプレートの名前を定義します。

-template_os_type *OS タイプ*

展開用のリソース テンプレートの OS タイプを定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。`caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: リソース テンプレートの削除

この例では、**Generic template** という名前のパッケージ テンプレートを削除します。

```
dpmresourcemgr removeresourcetemplate -template_name "Generic template"
-template_os_type 0
```

dpmresourcemgr retrieveresourcegroup コマンド -- リソース グループの取得

retrieveresourcegroup コマンドは、パッケージ グループを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr retrieveresourcegroup [-sc sc_url] [-group_name groupname]
-group_type type [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-group_name *グループ名*

グループの名前を定義します。

-group_type *タイプ*

リソース グループのタイプを定義します。 パッケージ グループの場合は **1** に設定します。 すべてのグループの場合は **0** に設定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: リソース グループの取得

この例では、任意のリソース タイプのパッケージ グループをすべて取得します。

```
dpmresourcemgr retrieveresourcegroup -group_type 0
```

例: 全パッケージ グループの取得

この例では、すべてのパッケージ グループ情報を取得します。

```
dpmresourcemgr retrieveresourcegroup -group_type 1
```

例: 特定のパッケージ グループの取得

この例では、特定のパッケージ グループ情報を取得します。

```
dpmresourcemgr retrieveresourcegroup -group_name "AGENT PACKAGE GROUP" -group_type 1
```

dpmresourcemgr retrieveresourcetemplate コマンド -- リソース テンプレートの取得

retrieveresourcetemplate コマンドは、パッケージ テンプレートを取得します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr retrieveresourcetemplate [-sc sc_url] [-template_name templatename]  
-template_os_type OSType [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-template_name *templatename*

テンプレートの名前を定義します。

-template_os_type *OS タイプ*

展開用のリソース テンプレートの OS タイプを定義します。すべての OS タイプについて -1 に設定します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: リソース テンプレートの取得

この例では、任意の OS タイプのパッケージ テンプレート情報を取得します。

```
dpmresourcemgr retrieveresourcetemplate -template_name "Generic template"
-template_os_type -1
```

dpmresourcemgr setcontainername コマンド -- コンテナ名の設定

setcontainername コマンドは、既存コンテナの名前を変更します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr setcontainername [-sc sc_url] -container_name containername  
-container_new_name containernewname [-ws_user username -ws_password  
password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-container_name *コンテナ名*

オブジェクトのコレクションを定義します。有効なエントリ: 親コンテナと子コンテナの区切りに円記号 (¥) を使用した、コンテナ名のフルパス。

-container_new_name *containernewname*

コンテナの新しい名前を定義します。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: コンテナ名の設定

この例では、コンテナの名前を **ManagedVM1** から **ManagedVM3** に変更します。

```
dpmresourcemgr setcontainername -container_name ManagedVM -container_new_name ManagedVM3.
```

dpmresourcemgr setcontainerservice コマンド -- サービスへのコンテナの設定

setcontainerservice コマンドはコンテナをサービスに変換します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr setcontainerservice [-sc sc_url] -container_name containername  
-machine_list machinelist -lower_threshold lowerthreshold -upper_threshold  
upperthreshold -lag lag -priority priority [-ws_user username -ws_password  
password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ: カンマ区切りのマシン名。

-lower_threshold *lowerthreshold*

(オプション) サービスの全体的な使用率の下限しきい値を指定します。下限しきい値は上限しきい値未満である必要があります。有効なエントリ: `1 ~ 100 (%)`

-upper_threshold *upperthreshold*

(オプション) サービスの全体的な使用率の上限しきい値を指定します。上限しきい値は常に下限しきい値を超えている必要があります。
有効なエントリ：1 ～ 100 (%)

-lag *lagvalue*

(オプション) 遅延値を指定します。有効なエントリ：0 以上 (秒)。

-priority *priorityvalue*

(オプション) サービスの優先度を指定します。有効なエントリ：1 以上。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。**caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービスへのコンテナの変換

この例では、**Managed1** という名前のコンテナをサービスに変換します。

```
dpmresourcemgr setcontainterservice -container_name Managed1 -machine_list  
"ComputerList.MyCompany.com, 172.31.255.255, Computer1.MyCompany.com"  
-lower_threshold 20 -upper_threshold 80 -lag 1 -priority 1
```

dpmresourcemgr setmanagementip コマンド -- ホストの管理 IP を設定する

dpmresourcemgr setmanagementip コマンドは、ホストの管理 IP アドレスを設定します。

注: 重複したホスト名が `server-name` パラメータに存在する場合、代わりに `system_uuid` パラメータを使用するよう要求するコンソールメッセージが表示されます。この状況が発生した場合、データベース テーブル `[aom2].[dbo].[_CA_ComputerSystem]` の `[_Name]` 列の対応するレコードから UUID を取得できます。または、「`...¥productname¥bin¥caaipaomwsclient.exe /enum=CA_ComputerSystem`」の出力から取得することもできます。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr setmanagementip [-sc sc_url] -server_name hostname | -system_uuid uuid  
-ip_address ipaddress [-ws_user username -ws_password password] [-locale iso639value]  
-sc sc_url
```

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

```
-ws_user username -ws_password password
```

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。`caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

dpmresourcemgr setmanagementstatus コマンド -- ホストの管理ステータスを設定する

setmanagementstatus コマンドは、管理対象または管理対象外の状態をホストに設定します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr setmanagementstatus [-sc sc_url] -server_name hostname  
-management_status 0|1 [-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : **80**

-host_name *source host name*

操作するソース サーバの名前を定義します。

-management_status 0 | 1

サーバの管理ステータスを定義します。有効なエントリ : **0** = 管理対象外、**1** = 管理対象。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力求められます。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は fr_FR) で指定します。コマンドプロンプトのロケールを使用する場合は「native」を指定します。

例: ホストに管理対象の状態を設定する

この例では、ServerA という名前のホストの管理ステータスを管理対象状態に設定します。

```
dpmcda setmanagementstatus -sc https://localhost/dpm/sc -host_name ServerA  
-management_status 1
```

例: ホストに管理対象外の状態を設定する

このコマンドを使用すると、ServerB という名前のホストの管理ステータスを管理対象外状態に設定できます。

```
dpmcda setmanagementstatus -host_name ServerB -management_status 0 -ws_user wsuser  
-ws_password wsuserpassword
```

dpmresourcemgr setserviceproperties コマンド -- サービス プロパティを設定する

setserviceproperties コマンドを使用すると、サービス プロパティを作成できます。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr setserviceproperties [-sc sc_url] -servicename servicename  
-lower_threshold lowerthreshold -upper_threshold upperthreshold -lag lag -priority  
priority [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-service_name *servicename*

サービスの名前を定義します。有効なエントリ : サービス名へのフルパス。親サービスとサブサービスを区切るには、円記号 `¥` を使用します。

-lower_threshold *lowerthreshold*

(オプション) サービスの全体的な使用率の下限しきい値を指定します。下限しきい値は上限しきい値未満である必要があります。有効なエントリ : `1 ~ 100 (%)`

-upper_threshold *upperthreshold*

(オプション) サービスの全体的な使用率の上限しきい値を指定します。上限しきい値は常に下限しきい値を超えている必要があります。有効なエントリ : `1 ~ 100 (%)`

-lag *lagvalue*

(オプション) 遅延値を指定します。有効なエントリ : `0 以上 (秒)`。

-priority *priorityvalue*

(オプション) サービスの優先度を指定します。有効なエントリ : `1 以上`。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が求められます。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービス プロパティを設定する

この例では、**Service12** というサービスにサービス プロパティを設定します。

```
dpmresourcemgr setserviceproperties -servicename Service12 -lower_threshold 2  
-upper_threshold 4 -log 2 -priority 2 -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr updateresourcegroup コマンド -- リソース グループの更新

updateresourcegroup コマンドは、パッケージ グループを更新します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr updateresourcegroup [-sc sc_url] -group_name groupname  
-group_description description -group_type grouptype -group_attribute attribute  
[-package_list packagelist][-ws_user username -ws_password password][-locale  
iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: **https://hostname:port/...**

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: **localhost.ca.com**

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: **80**

-group_name *グループ名*

グループの名前を定義します。

`-group_description` *説明*

グループの説明を定義します。

`-group_type` *タイプ*

リソース グループのタイプを定義します。 パッケージ グループの場合は **1** に設定します。 すべてのグループの場合は **0** に設定します。

`-group_attribute` *属性*

グループの属性を定義します。

`-package_list` *パッケージ リスト*

個々のパッケージを定義します。 グループ タイプを **1** (パッケージ) に設定する場合、このオプションを指定します。 有効なエントリ：

`-package_list "<パッケージ名 A>|<パッケージ プロシージャ A>|<プロシージャ タイプ A>|<パッケージ名 B>|..."`

`-ws_user username -ws_password password`

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。 認証情報が含まれていない場合、入力が必要です。 `caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

`-locale iso639value`

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。 コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: リソース グループの更新

この例では、リソース グループを更新します。

```
dpmresourcemgr updateresourcegroup -group_name "AGENT PACKAGE GROUP"
-group_description "Agent package group" -group_type 1 -group_attribute 0
-package_list "Generic Agent Win32 r5.0|install|INSTALL"
```


dpmresourcemgr updateresourcetemplate コマンド -- リソース テンプレートの更新

updateresourcetemplate コマンドは、パッケージテンプレートを更新します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr updateresourcetemplate [-sc sc_url] -template_name templatename  
-template_description templatedescription -template_os_type OSType [-group_list  
grouplist][-package_list package_list][-ws_user username -ws_password  
password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-template_name *templatename*

テンプレートの名前を定義します。

-template_description *説明*

テンプレートの説明を定義します。

-template_os_type *OS タイプ*

展開用のリソース テンプレートの OS タイプを定義します。

-group_list *グループリスト*

リソース テンプレートに追加するグループのリストを定義します。有効なエントリ : `-group_list "<グループ名 A>|<グループ タイプ 1= ソフトウェア パッケージ グループ>|<グループ名 B>|..."`

-package_list パッケージ リスト

個々のパッケージを定義します。グループ タイプを **1** (パッケージ) に設定する場合、このオプションを指定します。有効なエントリ:

-package_list "<パッケージ名 A>|<パッケージ プロシージャ A>|<プロシージャ タイプ A>|<パッケージ名 B>|...".

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、**ISO 639_3166** の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンド プロンプトのロケールを使用する場合は「**native**」を指定します。

例: パッケージ テンプレートの更新

この例では、パッケージ テンプレートを更新します。

```
dpmresourcemgr updateresourcetemplate -template_name "Generic template"
-template_description "Generic template description" -template_os_type 0 -group_list
"Package group A|1|Package group B|1" -package_list "Generic Agent Win32
r5.0|Install|INSTALL
```

dpmresourcemgr updateservice コマンド -- サービスを更新する

updateservice コマンドは、サービスの特性を変更します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr updateservice [-sc sc_url] -service_name servicename -machine_list
machinelist -lower_threshold lowerthreshold -upper_threshold upperthreshold -lag lag
-priority priority [-ws_user username -ws_password password][[-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-service_name *servicename*

サービスの名前を定義します。

-machine_list *machinelist*

(オプション) 利用可能なサーバのリストを指定します。有効なエントリ : カンマ区切りのマシン名。

-lower_threshold *lowerthreshold*

(オプション) サービスの全体的な使用率の下限しきい値を指定します。下限しきい値は上限しきい値未満である必要があります。有効なエントリ : `1 ~ 100 (%)`

-upper_threshold *upperthreshold*

(オプション) サービスの全体的な使用率の上限しきい値を指定します。上限しきい値は常に下限しきい値を超えている必要があります。有効なエントリ : `1 ~ 100 (%)`

-lag *lagvalue*

(オプション) 遅延値を指定します。有効なエントリ : `0` 以上 (秒)。

-priority *priorityvalue*

(オプション) サービスの優先度を指定します。有効なエントリ : `1` 以上。

-ws_user username -ws_password password

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale iso639value

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: サービスを更新する

この例では、サービス **TESTGROUP1** のしきい値、遅延、および優先度を更新します。

```
dpmresourcemgr updateservice -service_name TESTGROUP1 -machine_list  
"machine4.MyCompany.com, machine5.MyCompany.com, 192.168.255.255" -lower_threshold  
10 -upper_threshold 85 -lag 0 -priority 2 -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr updateipnetwork コマンド -- ネットワークを更新する

dpmresourcemgr updateipnetwork コマンドは、既存のネットワークのパラメータを更新します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr updateipnetwork [-sc sc_url] -subnet_address subnetaddress [-vlan_id  
vlanid] [-name name] [-description description] [-ws_user username -ws_password  
password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例 : `https://hostname:port/...`

hostname

サービス コントローラ がインストールされている完全修飾ホスト名を定義します。

例 : `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例 : `80`

-subnet_address *address*

ネットワークの IP アドレスを定義します。有効なエン트리 : `xxx.xxx.xxx.xxx` または `xxx.xxx.xxx.xxx/xx`。

-vlan_id *vlanid*

仮想ネットワーク (VLAN) を示す整数値を定義します。有効なエン트리 : `1 ~ 4094` の整数。`1` はデフォルトのネットワーク インターフェース カード (NIC) を示します。

-name *name*

アイテム名を定義します。有効なエン트리 : `1 ~ 128` 文字の英数字。

-description *description*

ユーザがわかるようにネットワークの説明を定義します。有効なエン트리 : `1 ~ 128` 文字の英数字。

-ws_user *username* -ws_password *password*

(オプション) Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要になります。`caaipsecurity` を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、`ISO 639_3166` の組み合わせ (たとえばフランス語の場合は `fr_FR`) で指定します。コマンドプロンプトのロケールを使用する場合は「`native`」を指定します。

例: ネットワークを更新する

この例では、ネットワーク `san diego 209` サブネットに説明を追加します。

```
dpmresourcemgr updateipnetwork -subnet_name 10.10.21.130 -vlan_id 4001 -name "san
diego 209 subnet" -description "5th floor lab" -ws_user wsuser -ws_password
wsuserpassword
```

dpmresourcemgr updatesubnetdefaults コマンド -- ネットワークのデフォルト値を更新する

`dpmresourcemgr updatesubnetdefaults` コマンドは、既存のネットワークのデフォルト値を更新します。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr updatesubnetdefaults [-sc sc_url] -subnet_address subnetaddress
-ip_gateway_list ipaddresses -dns_domainname domainname -dns_list ipaddresses
-dns_suffix_list suffixes -netbios_server_list ipaddresses -ntp_server_list
ipaddresses [-ws_user username -ws_password password][-locale iso639value]
```

`-sc sc_url`

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

`-subnet_address address`

ネットワークの IP アドレスを定義します。有効なエントリ:

`xxx.xxx.xxx.xxx` または `xxx.xxx.xxx.xxx/xx`。

-ip_gateway_list *ipaddresses*

カンマ区切りのリストを使用して、デフォルト ゲートウェイおよび代替ゲートウェイの IP アドレスを定義します。デフォルト ゲートウェイおよび代替ゲートウェイは同じネットワーク上にある必要があります。有効なエントリ：xxx.xxx.xxx.xxx。

-dns_domainname *domainname*

ネットワークの DNS ドメイン名を定義します。有効なエントリ：ドット区切りのアドレス（例：xxx.yyy.com）ごとに 1 ～ 63 文字の英数字。アドレスを数字で始めることはできません。

-dns_list *ipaddresses*

優先 DNS サーバおよび代替 DNS サーバの DNS リストと、プライマリおよびセカンダリ Windows インターネット ネーム サービス (WINS) サーバを定義します。有効なエントリ：次の形式の IP アドレスのカンマ区切りのリスト：xxx.xxx.xxx.xxx。

-dns_suffix_list *suffixes*

ネットワークの DNS サフィックスのリストを定義します。有効なエントリ：次の形式のサフィックスのカンマ区切りのリスト：
name.name.com。

-netbios_server_list *ipaddresses*

NetBIOS ネーム サーバの IP アドレスのリストを定義します。有効なエントリ：次の形式の IP アドレスのカンマ区切りのリスト：
xxx.xxx.xxx.xxx。

-ntp_server_list *ipaddresses*

ネットワーク タイム プロトコル (NTP) サーバを定義します。有効なエントリ：次の形式の NTP サーバのカンマ区切りのリスト：
xxx.xxx.xxx.xxx。

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。caaipsecurity を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

(オプション) デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ (たとえばフランス語の場合は **fr_FR**) で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: ネットワークのデフォルト値を更新する

この例では、ネットワーク **san diego 209** サブネットのデフォルト値を更新します。

```
dpmresourcemgr updatesubnetdefaults -subnet_address 10.10.21.130 -ip_gateway_list 10.10.33.0,10.10.34.0 -dns_domainname turbo.com -dns_list 10.10.55.21 -netbios_server_list 10.10.21.130 -ntp_server_list 10.10.33.135 -ws_user wsuser -ws_password wsuserpassword
```

dpmresourcemgr updateipaddresspool コマンド -- IP アドレス プールを更新する

dpmresourcemgr updateipaddresspool コマンドを使用すると、既存のプールの IP 割り当ての方法を編集できます。

このコマンドの形式は、以下のとおりです。

```
dpmresourcemgr updateipaddresspool [-sc sc_url] -name name|-starting_ip ipaddress -ip_assignment_type static|dhcp [-ws_user username -ws_password password][-locale iso639value]
```

-sc *sc_url*

(オプション) サービス コントローラ ホスト用の URL を定義します。

例: `https://hostname:port/...`

hostname

サービス コントローラがインストールされている完全修飾ホスト名を定義します。

例: `localhost.ca.com`

port

サービス コントローラ ホスト用のリスニング ポートを定義します。

例: `80`

-name *name* | -starting_ip *ipaddress*

名前または開始 IP アドレス範囲を定義します。有効なエントリ：1 ～ 128 の英数字、または xxx.xxx.xxx.xxx。

-ip_assignment_type *static* | *dhcp*

サーバで IP アドレス（静的 IP アドレスまたは DHCP）の取得に使用するメソッドを定義します。

-ws_user *username* -ws_password *password*

（オプション）Web サービスのセキュリティ チェックで使用する認証情報を指定します。認証情報が含まれていない場合、入力が必要です。 **caaipsecurity** を使用して独自のセッションをセットアップすることにより、認証情報のプロンプトを回避します。

-locale *iso639value*

（オプション）デフォルトの英語出力より優先させるロケールを、ISO 639_3166 の組み合わせ（たとえばフランス語の場合は **fr_FR**）で指定します。コマンドプロンプトのロケールを使用する場合は「**native**」を指定します。

例: IP アドレス プールを更新する

この例では、10.10.21.130 で始まる IP アドレス プールの IP 割り当て方法を DHCP に変更します。

```
dpmresourcemgr updateipaddresspool -starting_ip 10.10.21.130 -ip_assignment_type DHCP -ws_user wsuser -ws_password wsuserpassword
```

リモート モニタリング コマンド

リモート モニタリングでは、タスクの自動化に使用できるコマンドライン インターフェース (rmonwatch) が RM AIM に提供されます。たとえば、これを使用してリモート モニタリング システムをリモート モニタリング 設定セットに割り当てることができます。

このコマンドの形式は、以下のとおりです。

```
rmonwatch [options] getVersion | {add|del} configfile | getTable tablename
```

rmonwatch add コマンド -- MIB テーブルに行を追加する

`rmonWbemSysTable` および `rmonWbemCredTable` MIB テーブルに行を追加するには、`add` コマンドを使用します。

`add` コマンドは、システムおよび認証情報エントリに関する詳細を含む設定ファイルを読み取ります。このファイルは `.ini` スタイルの形式を使用し、その各セクションにはシステムまたは認証情報エントリが指定されています。セクション名は一意である必要があるため、`_<n>` という用語には 1 から始まる `<n>` が追加されます。各セクションには、それぞれの MIB テーブルの設定可能な属性用の変数と値の割り当てが含まれます。

例: サンプル設定ファイル(rmonwatch.cf)

```
[System_1]
SystemName = sys01-01
RowStatus = active
ConfigSet = default
Credentials = test
MaxInstances = 100
[Credentials_1]
Credentials = test
RowStatus = active
UserName = Administrator
Password = IE7vse8zkBzurAwz5880vyTpc9v
```

認証情報エントリの **Password** 属性は、暗号化されたフォームで指定する必要があります。パスワードを暗号化するには、以下のように **CA SystemEDGE se_enc** ユーティリティを使用します。

```
se_enc -s -i in.txt -o out.txt
```

暗号化するパスワードを **in.txt** に追加します（改行を追加しません）。このコマンドを呼び出したら、**out.txt** から暗号化されたパスワードをコピーし、それを **Password** 属性の値として使用します。

add コマンドは次の構文を使用します。

```
rmonwatch [options] add configfile
```

rmonwatch は次のオプションを使用します。

--version

プログラムのバージョン番号を表示し、終了します。

--help

rmonwatch のヘルプを表示します

`-h hostname | -h ipAddr`

(オプション) RM AIM を実行する CA SystemEDGE ホストを指定します。

デフォルト : localhost

`-p port`

(オプション) CA SystemEDGE SNMP ポートを指定します。

デフォルト : 161

`-c community`

(オプション) SNMP バージョン 1 および 2c の SNMP 読み取り/書き込みコミュニティ文字列を指定します。

デフォルト : public

例: add コマンドを使用して rmonwatch.cf に定義されたテーブル行を追加する

```
rmonwatch -c admin add rmonwatch.cf
RMONWBEM AIM Watch Program, version 1.0.0
### 認証情報エントリ「test」を処理する
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.12.1.3.4.116.101.115.116 -s Administrator
1.3.6.1.4.1.546.16.22.12.1.3.4.116.101.115.116 Administrator
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.12.1.4.4.116.101.115.116 -s IE7vse8zkBzurAwz5880vyTpc9v=
1.3.6.1.4.1.546.16.22.12.1.4.4.116.101.115.116 IE7vse8zkBzurAwz5880vyTpc9v=
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 -i 1
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 1
### システム エントリ「sys01-01」を処理する
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.3.8.115.121.115.48.49.45.48.49 -s "default"
1.3.6.1.4.1.546.16.22.11.1.3.8.115.121.115.48.49.45.48.49 default
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.4.8.115.121.115.48.49.45.48.49 -s "test"
1.3.6.1.4.1.546.16.22.11.1.4.8.115.121.115.48.49.45.48.49 test
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.5.8.115.121.115.48.49.45.48.49 -i 1
1.3.6.1.4.1.546.16.22.11.1.5.8.115.121.115.48.49.45.48.49 1
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.6.8.115.121.115.48.49.45.48.49 -i 100
1.3.6.1.4.1.546.16.22.11.1.6.8.115.121.115.48.49.45.48.49 100
E:¥sysedge¥bin¥snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 -i 1
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 1
```

rmonwatch del コマンド -- MIB テーブルから行を削除する

rmonWbemSysTable および rmonWbemCredTable MIB テーブルから行を削除するには、del コマンドを使用します。

del コマンドは、システムおよび認証情報エントリに関する詳細を含む設定ファイルを読み取ります。このファイルは .ini スタイルの形式を使用し、その各セクションにはシステムまたは認証情報エントリが指定されています。セクション名は一意である必要があるため、_*<n>* という用語には 1 から始まる *<n>* が追加されます。各セクションには、それぞれの MIB テーブルの設定可能な属性用の変数と値の割り当てが含まれます。

例: サンプル設定ファイル(rmonwatch.cf)

```
[System_1]
SystemName = sys01-01
RowStatus = active
ConfigSet = default
Credentials = test
MaxInstances = 100
[Credentials_1]
Credentials = test
RowStatus = active
UserName = Administrator
Password = password01
```

delete コマンドは次の構文を使用します。

```
rmonwatch [options] del configfile
```

rmonwatch は次のオプションを使用します。

--version

プログラムのバージョン番号を表示し、終了します。

--help

rmonwatch のヘルプを表示します

-h *hostname* | -h *ipAddr*

(オプション) RM AIM を実行する CA SystemEDGE ホストを指定します。

デフォルト : localhost

-p *port*

(オプション) CA SystemEDGE SNMP ポートを指定します。

デフォルト : 161

-c *community*

(オプション) SNMP バージョン 1 および 2c の SNMP 読み取り/書き込みコミュニティ文字列を指定します。

デフォルト : public

例: del コマンドを使用して rmonwatch.cf に定義されたテーブル行を削除する

```
rmonwatch -c admin del rmonwatch.cf
```

注: 行の削除は、設定ファイルのインデックス属性を指定する必要があります。

RMONWBEM AIM Watch Program, version 1.0.0

認証情報エントリ「test」を処理する

```
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o  
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 -i 6  
1.3.6.1.4.1.546.16.22.12.1.2.4.116.101.115.116 6
```

システム エントリ「sys01-01」を処理する

```
E:\sysedge\bin\snmpset -h localhost -p 161 -v 1 -c admin -m 0 -t 5 -r 3 -o  
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 -i 6  
1.3.6.1.4.1.546.16.22.11.1.2.8.115.121.115.48.49.45.48.49 6
```

rmonwatch getTable コマンド -- rmonWbem MIB テーブルを表示する

次の MIB テーブルの内容を表示するには、`getTable` コマンドを使用します。テーブルの名前には、太字で示された用語を含む必要があります。大文字または小文字を使用できます。

- `rmonWbemCSetTable`
- `rmonWbemSysTable`
- `rmonWbemCredTable`
- `rmonWbemConfTable`
- `rmonWbemQueryTable`
- `rmonWbemInstTable`

`getTable` コマンドは次の構文を使用します。

```
rmonwatch [options] getTable {cset | sys | cred | conf | query | inst}
```

`rmonwatch` は次のオプションを使用します。

`--version`

プログラムのバージョン番号を表示し、終了します。

`--help`

`rmonwatch` のヘルプを表示します

`-h hostname | -h ipAddr`

(オプション) RM AIM を実行する CA SystemEDGE ホストを指定します。

デフォルト : localhost

`-p port`

(オプション) CA SystemEDGE SNMP ポートを指定します。

デフォルト : 161

`-c community`

(オプション) SNMP バージョン 1 および 2c の SNMP 読み取り/書き込みコミュニティ文字列を指定します。

デフォルト : public

例: システム テーブルに対して getTable コマンドを実行する

```
rmonwatch getTable sys
RMONWBEM AIM Watch Program, version 1.0.0
E:¥sysedge¥bin¥walktree -h localhost -p 161 -v 1 -c public -m 0 -t 30 -r 3 -o
1.3.6.1.4.1.546.16.22.11
rmonWbemSysSystemName <sys01-01> .sys01-01
rmonWbemSysSystemName <sys02-01-8> .sys02-01-8
rmonWbemSysSystemName <sys03-01-7> .sys03-01-7
rmonWbemSysRowStatus active .sys01-01
rmonWbemSysRowStatus active .sys02-01-8
rmonWbemSysRowStatus active .sys03-01-7
rmonWbemSysDescr <nul> .sys01-01
rmonWbemSysDescr <nul> .sys02-01-8
rmonWbemSysDescr <nul> .sys03-01-7
rmonWbemSysContact <nul> .sys01-01
rmonWbemSysContact <nul> .sys02-01-8
rmonWbemSysContact <nul> .sys03-01-7
rmonWbemSysQualifiers .sys01-01
rmonWbemSysQualifiers .sys02-01-8
rmonWbemSysQualifiers .sys03-01-7
rmonWbemSysConfigSet <default> .sys01-01
rmonWbemSysConfigSet <metricNet,metricFS,metricDisk> .sys02-01-8
rmonWbemSysConfigSet <default> .sys03-01-7
rmonWbemSysCredentials <test> .sys01-01
rmonWbemSysCredentials <win2008> .sys02-01-8
rmonWbemSysCredentials <win7> .sys03-01-7
rmonWbemSysProtocol dcom .sys01-01
rmonWbemSysProtocol dcom .sys02-01-8
rmonWbemSysProtocol dcom .sys03-01-7
rmonWbemSysMaxInstances <100> .sys01-01
rmonWbemSysMaxInstances <100> .sys02-01-8
rmonWbemSysMaxInstances <100> .sys03-01-7
rmonWbemSysIPAddress <192.168.0.10> .sys01-01
rmonWbemSysIPAddress <192.168.0.11> .sys02-01-8
rmonWbemSysIPAddress <192.168.0.12> .sys03-01-7
rmonWbemSysMACAddress <00:12:3F:64:1D:AE> .sys01-01
rmonWbemSysMACAddress <00:50:56:97:7D:BA> .sys02-01-8
rmonWbemSysMACAddress <00:25:64:A9:98:86> .sys03-01-7
rmonWbemSysAdded <2010-5-17,17:26:35.0,+2:0> .sys01-01
rmonWbemSysAdded <2010-5-17,17:26:37.0,+2:0> .sys02-01-8
rmonWbemSysAdded <2010-5-17,17:26:40.0,+2:0> .sys03-01-7
rmonWbemSysLastBoot <2010-5-17,7:52:25.0,+2:0> .sys01-01
rmonWbemSysLastBoot <2010-5-1,20:53:22.0,+2:0> .sys02-01-8
rmonWbemSysLastBoot <2010-5-1,20:53:22.0,+2:0> .sys03-01-7
rmonWbemSysLastPoll <0d 00:10:41> .sys01-01
rmonWbemSysLastPoll <0d 00:11:11> .sys02-01-8
rmonWbemSysLastPoll <0d 00:11:11> .sys03-01-7
```



```

rmonWbemSysLastSuccess <0d 00:10:41> .sys01-01
rmonWbemSysLastSuccess <0d 00:11:11> .sys02-01-8
rmonWbemSysLastSuccess <0d 00:11:11> .sys03-01-7
rmonWbemSysLastError success .sys01-01
rmonWbemSysLastError success .sys02-01-8
rmonWbemSysLastError success .sys03-01-7
rmonWbemSysQueryNumber <14> .sys01-01
rmonWbemSysQueryNumber <17> .sys02-01-8
rmonWbemSysQueryNumber <14> .sys03-01-7
rmonWbemSysQueryFailed <0> .sys01-01
rmonWbemSysQueryFailed <0> .sys02-01-8
rmonWbemSysQueryFailed <0> .sys03-01-7

```

rmonwatch getVersion コマンド -- バージョン情報を表示する

rmonWbemVersion MIB 変数の値を表示するには、getVersion コマンドを使用します。

getVersion コマンドは次の構文を使用します。

```
rmonwatch [options] getVersion
```

rmonwatch は次のオプションを使用します。

--version

プログラムのバージョン番号を表示し、終了します。

--help

rmonwatch のヘルプを表示します

-h hostname | -h ipAddr

(オプション) RM AIM を実行する CA SystemEDGE ホストを指定します。

デフォルト : localhost

-p port

(オプション) CA SystemEDGE SNMP ポートを指定します。

デフォルト : 161

-c community

(オプション) SNMP バージョン 1 および 2c の SNMP 読み取り/書き込みコミュニティ文字列を指定します。

デフォルト : public

例: getVersion コマンドで返された結果

```
RMONWBEM AIM Watch Program, version 1.0.0
E:¥sysedge¥bin¥snmpget -h localhost -p 161 -v 1 -c public -m 0 -t 5 -r 3 -o
1.3.6.1.4.1.546.16.22.1.0
1.3.6.1.4.1.546.16.22.1.0 SysEDGE RMONWBEM AIM, VERSION:1.0.0 BUILD:10137
REQUIRES:5.0.0
```

SNMP の詳細パラメータ

次のパラメータは、`rmonwatch add`、`del`、`getTable`、および `getVersion` コマンドと同じです。これらのパラメータは、**SNMP バージョン 2c** または **3** を使用している場合、またはデフォルト以外のタイムアウトおよび再試行を使用している場合のみに必要です。

`-v snmpVersion`

(オプション) **SNMP** バージョンを指定します。設定可能な値は以下のとおりです。

1

2c

3

デフォルト : 1

`-u secName`

(オプション) **SNMPv3** セキュア ユーザの名前を指定します。

`-s secLevel`

(オプション) **SNMPv3** のセキュリティ レベルを指定します。可能な値は次のとおりです (デフォルト値はありません)。

1 = noAuthNoPriv

2 = AuthNoPriv

3 = AuthPriv

`-n contextName`

(オプション) **MIBMuxed** エージェントのインスタンス名を指定します (デフォルト値はありません)。

-a authPassword

(オプション) SNMPv3 のセキュリティに AuthNoPriv または AuthPriv が選択された場合に必要な認証パスワードを識別します (デフォルト値はありません)。

-A authProtocol

(オプション) 認証プロトコルを指定します。設定可能な値は以下のとおりです。

MD5 = 認証プロトコル HMAC-MD5

SHA = 認証プロトコル HMAC-SHA

デフォルト : MD5

-x privPassword

(オプション) SNMPv3 のセキュリティ レベルが 3 (AuthPriv) である場合のプライバシー (暗号化) パスワードを指定します。

-X encryptProtocol

(オプション) プライバシーに使用する暗号化プロトコルを指定します。設定可能な値は以下のとおりです。

DES - データ暗号化標準

AES - 128 ビットの暗号化キーを使用した高度暗号化標準 (AES128)

-m FIPSmode

(オプション) FIPS モードを指定します。設定可能な値は以下のとおりです。

0 = 非 FIPS

1 = FIPS 共存

2 = FIPS のみ

デフォルト : 0

-t timeout

(オプション) SNMP の set および get コマンドのタイムアウトを指定します。

デフォルト : 5

`--timeout=walktreetimeout`

(オプション) `walktree` コマンドの `SNMP` タイムアウトを指定します。

デフォルト : 30

`-r snmpretries`

(オプション) `SNMP` コマンドを再試行する回数を指定します。

デフォルト : 3

ログ ファイル

このセクションでは、利用可能な `CA Virtual Assurance` ログ ファイルの一部と、その構成設定を紹介します。

利用可能なログ ファイル

CA Virtual Assurance 内のさまざまなコンポーネントでは、ログ ファイルが生成されます。製品で問題が発生した場合には、これらのログ ファイルを検査できます。また、テクニカルサポートとスムーズにやり取りするためにも、これらのログ ファイルは必要です。ここに示すログ ファイルは以下のディレクトリに配置されます。

インストールパス¥CA¥VPM¥log

利用可能なログ ファイルと、各ログ ファイルに含まれる情報を以下に示します。

CA Virtual Assurance 外部コンポーネント マネージャ

DPMExtCompMgr.log

外部コンポーネント マネージャのログ情報が含まれます。

CLI ログ ファイル

dpmimgcli.log (dpmimg.exe)

発行されたイメージング コマンドのログ情報が含まれます。

dpmvccli.log (dpmvc.exe)

発行された vCenter コマンドのログ情報が含まれます。

dpmlparcli.log (dpmlpar.exe)

発行された LPAR コマンドのログ情報が含まれます。

dpmzonecli.log (dpmzone.exe)

発行された Solaris ゾーン コマンドのログ情報が含まれます。

dpmcmdcli.log (dpmcmd.exe)

発行された汎用シェル コマンドのログ情報が含まれます。

dpmresourcemgrcli.log (dpmresourcemgr.exe)

発行されたリソース マネージャ コマンドのログ情報が含まれます。

イベント マネージャ

DPMEventMgr.log

イベント マネージャのログ情報が含まれます。

イメージング

dpm-imaging.log

イメージング サービスのログ情報が含まれます。

開始

dpm-schedadapter.log

開始サービスのランタイム ログ情報が含まれます。

SchedInitialize.log

開始サービスの初期化ログ情報が含まれます。

パフォーマンス モニタ

coll_eng.log

パフォーマンス モニタのログ情報が含まれます。

ポリシー エンジン

capolicyeng.log

ポリシー エンジンのログ情報が含まれます。

リソース マネージャ

ResourceMgr.log

リソース マネージャのログ情報が含まれます。

サービス コントローラ

servcon.log

サービス コントローラのログ情報が含まれます。

VirtualCenter アダプタ

dpm-vcadapter.log

VirtualCenter アダプタのネイティブ ランタイム コンポーネントのログ情報が含まれます。

vcadapter_java.log

VirtualCenter アダプタの Java ランタイム コンポーネントのログ情報が含まれます。

ログ ファイル設定

各ログ ファイルを設定するには、メモ帳や同様のテキスト エディタを使って *dpmlogging.cfg* ファイルを編集します。ファイルは以下のディレクトリにあります。

インストールパス¥conf

以下のプロパティの設定を変更できます。

最大ログ サイズ(バイト)

ログ ファイルのサイズを定義します。

最大ロールオーバー

最大サイズに到達した後に保存するログ ファイルの数を定義します。

ログ レベル

ログでキャプチャするイベントのレベルを定義します。

利用できる構文とログ レベルは *dpmlogging.cfg* ファイルに示されています。

索引

!

- !! コマンド -- コマンドまたは子プロセスの呼び出しと自動キャプチャ出力 (Cmdlet) - 54
- ! コマンド -- コマンドまたは子プロセスの呼び出し (Cmdlet) - 55

?

- ?? コマンド -- リストを標準出力に書き込む (Cmdlet) - 56
- ? コマンド -- 出力を標準出力に書き込む (Cmdlet) - 57

A

- accept コマンド -- 標準入力から文字列を読み取り、変数に割り当てる (Cmdlet) - 58
- arrdump コマンド -- 配列の表示 (Cmdlet) - 58
- ASSERT コマンド -- 数式の比較 (Funclet) - 59
- AutoShell - 19
- AutoShell 関数 - 132
- AutoShell クラス - 159
- AutoShell コア リファレンス - 49
- AutoShell について - 19
- AutoShell の一般コマンド - 53
- AutoShell の呼び出し - 27
- AutoShell へのアクセス - 22
- AutoShell ロード可能モジュールのコマンド リファレンス - 191

B

- base64Decode -- base64 でエンコードされた文字列をデコードする (関数) - 133
- base64Encode -- 文字列を base64 でエンコーディングする (関数) - 134

C

- CA Cisco UCS コマンド - 450
- CA Hyper-V CLI コマンド - 499

- CA IBM LPAR AutoShell コマンド - 352
- CA IBM LPAR の CLI コマンド - 586
- CA Microsoft Cluster Server CLI コマンド - 632
- CA Microsoft Hyper-V AutoShell コマンド - 237
- CA Oracle Solaris ゾーン AutoShell コマンド - 383
- CA Solaris ゾーン CLI コマンド - 656
- CA Technologies 製品リファレンス - 3
- CA VMware vCenter Server AutoShell コマンド - 192
- CA VMware vCenter Server の CLI コマンド - 691
- caaipsecurity コマンド -- Web サービスのセキュリティ チェック用の認証情報の指定 - 849
- caismutility コマンド -- [ポリシー設定] UI への機能の追加 - 891
- cat、type コマンド -- テキスト ファイルの表示 (Cmdlet) - 61
- CA への連絡先 - 3
- cd、chdir コマンド -- ディレクトリの変更 (Cmdlet、Funclet) - 62
- CIM オブジェクトの取得 - 25
- CIM オブジェクトの前提と依存の取得 - 24
- CIM クラスの取得 - 23
- Common Discovery コマンド - 813
- Common Information Model (CIM) オブジェクト - 23
- copy、cp コマンド -- ファイルとディレクトリのコピー (Funclet) - 63
- createdpmdefaultgroup コマンド -- サービスの作成または更新 - 850
- curDir -- 現在のディレクトリを取得する (関数) - 134

D

- DBG_PROMPT_LINE コマンド -- 確認のために行番号を表示してスクリプトの中断を実行 (Cmdlet) - 67

DBG_PROMPT コマンド -- 確認のためにスクリプトの中断を実行 (Cmdlet) - 66

del、rm コマンド -- ファイルの削除 (Funclet) - 68

dir、ls コマンド -- ファイルおよびディレクトリ情報の取得 (Funclet) - 71

dpmcda cancelnetworkscan コマンド -- ネットワーク ディスカバリの取り消し - 814

dpmcda deletenetwork コマンド -- ネットワークの削除 - 815

dpmcda discoverhost コマンド -- ホストを検出する - 816

dpmcda discovernetwork コマンド -- ネットワークを検出する - 817

dpmce ceconfig -configuremetricfilter コマンド -- メトリックフィルタを設定する - 822

dpmce ceconfig -configuremetric コマンド -- メトリックを設定する - 819

dpmce ceconfig -disablemetricforoverall コマンド -- メトリックを無効にし、全体的な計算から除外する - 826

dpmce ceconfig -disablemetric コマンド -- メトリックを無効にする - 825

dpmce ceconfig -enablemetricforoverall コマンド -- メトリックを有効にし、全体的な計算に組み込む - 829

dpmce ceconfig -enablemetric コマンド -- メトリックを有効にする - 827

dpmce ceconfig -getglobal コマンド -- データセンターの間隔を取得する - 830

dpmce ceconfig -getmetriclist コマンド -- メトリックリストを取得する - 833

dpmce ceconfig -getnode コマンド -- ホストの収集間隔を取得する - 832

dpmce ceconfig -refreshmetriclist コマンド -- メトリック収集を再試行する - 834

dpmce ceconfig -setalias コマンド -- エイリアスを設定する - 835

dpmce ceconfig -setglobal コマンド -- データセンターの間隔を設定する - 837

dpmce ceconfig -setnode コマンド -- ホストの間隔を設定する - 839

dpmce ceconfig -updatemetric コマンド -- メトリックを更新する - 841

dpmcmd run コマンド -- スクリプトの実行 - 850

dpmeventmanager create_event コマンド -- イベントの作成 - 843

dpmeventmanager get_events コマンド -- イベントの取得 - 845

dpmhd addcomment コマンド -- チケットをコメントで更新する - 855

dpmhd close コマンド -- ヘルプ デスク チケットのクローズ - 856

dpmhd getattribute コマンド -- チケットの属性値の取得 - 857

dpmhd getstatus コマンド -- チケットのステータスの取得 - 859

dpmhd open コマンド -- ヘルプ デスク チケットのオープン - 860

dpmhd setattribute コマンド -- チケットの属性の設定 - 861

dpmhd setstatus コマンド -- チケットのステータスの設定 - 863

dpmhv-addVMNic (Cmdlet) コマンド -- VM にネットワーク インターフェース コントローラを追加する - 237

dpmhv-addVMSCSIController (Cmdlet) コマンド -- VM に SCSI コントローラを追加する - 239

dpmhv-changeVMState (Cmdlet) コマンド -- VM の状態を変更する - 240

dpmhv-createTemplateFromVM (Funclet) コマンド -- VM からテンプレートを作成する - 242

dpmhv-createVirtDisk (Funclet) コマンド -- 仮想ディスクを作成する - 244

dpmhv-createVirtFloppy (Cmdlet) コマンド -- 仮想フロッピーを作成する - 245

dpmhv-createVM (Funclet) コマンド -- VM を作成する - 246

dpmhv-createVMFromTemplate (Funclet) コマンド -- テンプレートを使用して VM を作成する - 251

dpmhv-createVMFromTemplateEx (Funclet) コマンド -- テンプレートを使用して VM を作成する - 258

dpmhv-destroyTemplate (Cmdlet) コマンド -- テンプレートを削除する - 268

dpmhv-destroyVM (Cmdlet) コマンド -- VM を削除する - 269

dpmhv-expandVirtDisk (Funclet) コマンド -- 仮想ディスクを拡張する - 270

dpmhv-exportVM (Funclet) コマンド -- VM をエクスポートする - 271

dpmhv-exportVMEx (Funclet) コマンド -- VM をエクスポートする - 273

dpmhv-getHostSwitches (Funclet) コマンド -- Hyper-V ホストのホスト スイッチを表示する - 275

dpmhv-getJobInfo (Funclet) コマンド -- ジョブ情報を表示する - 278

dpmhv-getJobStatus (Funclet) コマンド -- ジョブステータスを表示する - 283

dpmhv-getPhysDisks (Funclet) コマンド -- 物理ディスクをすべて表示する - 284

dpmhv-getVersion (Funclet) コマンド -- Web サービスのバージョンを表示する - 288

dpmhv-getVMInfo (Funclet) コマンド -- VM をすべて表示する - 289

dpmhv-getVMProperties (Funclet) コマンド -- VM のプロパティを表示する - 293

dpmhv-getVMState (Funclet) コマンド -- VM 状態を表示する - 295

dpmhv-importVM (Funclet) コマンド -- VM をインポートする - 296

dpmhv-importVMEx (Funclet) コマンド -- VM をコピーおよびインポートする - 298

dpmhv-setSysprepProperties (Cmdlet) コマンド -- 仮想ディスクを作成する - 300

dpmhv-setVMClustered (Cmdlet) コマンド -- クラスタに VM を設定する - 307

dpmhv-setVMDisk (Funclet) コマンド -- ドライブコントローラにドライブまたはディスクを作成する - 308

dpmhv-setVMProperties (Cmdlet) コマンド -- VM のプロパティを設定する - 311

dpmhv-SetVMVirtFloppy (Cmdlet) コマンド -- VM のフロッピードライブに仮想フロッピーイメージを接続する - 314

dpmhv-ShowClusterSharedVolumes (Funclet) コマンド -- クラスタ共有ボリュームをすべて表示する - 315

dpmhv-ShowDirectories (Funclet) コマンド -- ディレクトリをすべて表示する - 316

dpmhv-ShowHosts (Funclet) コマンド -- Hyper-V Server をすべて表示する - 318

dpmhv-ShowLogicalDisks (Funclet) コマンド -- 論理ディスクをすべて表示する - 326

dpmhv-ShowSCVMMHardwareProfiles (Funclet) コマンド -- SCVMM ハードウェアプロファイルをすべて表示する - 329

dpmhv-ShowSCVMMOSProfiles (Funclet) コマンド -- ゲスト OS プロファイルをすべて表示する - 333

dpmhv-ShowTemplates (Funclet) コマンド -- Hyper-V ホストのテンプレートを表示する - 339

dpmhv-ShowVMDisks (Funclet) コマンド -- 仮想ディスクをすべて表示する - 341

dpmhv-showVMFloppy (Funclet) コマンド -- フロッピードライブのプロパティを表示する - 343

dpmhv-showVMNICs (Funclet) コマンド -- ネットワーク インターフェース カードをすべて表示する - 346

dpmhv-showVMSCSI (Funclet) コマンド -- SCSI コントローラをすべて表示する - 348

dpmhyperv AddVMNic コマンド -- VM にネットワーク アダプタコントローラを追加する - 499

dpmhyperv AddVMSCSIController コマンド -- VM の SCSI コントローラを作成する - 501

dpmhyperv AgentGet コマンド -- エージェントの値を返す - 503

dpmhyperv AgentSet コマンド -- エージェントの値を設定する - 504

dpmhyperv ChangeVMState コマンド -- VM の状態を変更する - 506

dpmhyperv CreateTemplateFromVM コマンド -- VM からテンプレートを作成する - 508

dpmhyperv CreateVirtDisk コマンド -- 仮想ディスク イメージを作成する - 510

dpmhyperv CreateVirtFloppy コマンド -- 仮想フロッピー ディスク イメージを作成する - 512

dpmhyperv CreateVMFromTemplate コマンド -- SCVMM サーバ テンプレートから VM を作成する - 518

dpmhyperv CreateVMFromTemplate コマンド -- テンプレートから VM を作成する - 523

dpmhyperv CreateVM コマンド -- VM を作成する - 513

dpmhyperv DeleteTemplate コマンド -- テンプレートを削除する - 585

dpmhyperv DeleteVM コマンド -- VM を削除する - 530

dpmhyperv ExpandVirtDisk コマンド -- 仮想ディスクのサイズを拡張する - 532

dpmhyperv ExportVM コマンド -- VM をエクスポートする - 534

dpmhyperv ExportVMEx コマンド -- VM をエクスポートする - 536

dpmhyperv GetHostSwitches コマンド -- ホストのスイッチを取得する - 538

dpmhyperv GetJobErrorInfo コマンド -- ジョブのエラー情報を取得する - 540

dpmhyperv GetJobInfo コマンド -- ジョブの情報を取得する - 541

dpmhyperv GetJobState コマンド -- ジョブの状態を取得する - 543

dpmhyperv GetServerInfo コマンド -- すべての管理対象サーバを取得する - 544

dpmhyperv GetVersion コマンド -- PMM のバージョンを取得する - 545

dpmhyperv GetVMProperties コマンド -- VM のプロパティを取得する - 546

dpmhyperv GetVMState コマンド -- VM 状態を取得する - 548

dpmhyperv ImportVMEx コマンド -- VM をコピーおよびインポートする - 551

dpmhyperv ImportVM コマンド -- VM をインポートする - 549

dpmhyperv SetClustered コマンド -- クラスタに対して VM を追加または削除する - 553

dpmhyperv SetSysPrepProperties コマンド -- Sysprep プロパティを設定する - 555

dpmhyperv SetVMDisk コマンド -- ドライブコントローラにドライブまたはディスクを割り当てる - 559

dpmhyperv SetVMProperties コマンド -- VM のプロパティを設定する - 561

dpmhyperv SetVMVirtFloppy コマンド -- 仮想フロッピー イメージを VM フロッピードライブに割り当てる - 565

dpmhyperv ShowClusterSharedVolumes コマンド -- クラスタ共有ボリュームをすべて表示する - 566

dpmhyperv ShowDirectories コマンド -- Hyper-V Server のディレクトリのプロパティを表示する - 568

dpmhyperv ShowLogicalDisks コマンド -- 論理ディスクをすべて表示する - 569

dpmhyperv ShowPhysDisks コマンド -- ホストの物理ディスクを表示する - 570

dpmhyperv ShowSCVMMGuestOSProfiles コマンド -- ゲスト オペレーティング システムのプロファイルを表示する - 573

dpmhyperv ShowSCVMMHardwareProfiles コマンド -- SCVMM ハードウェア プロファイルをすべて表示する - 572

dpmhyperv ShowTemplates コマンド -- VM テンプレートをすべて表示する - 574

dpmhyperv ShowVirtFloppy コマンド -- 仮想フロッピードライブのリストを表示する - 575

dpmhyperv ShowVMDisks コマンド -- 仮想ディスクをすべて表示する - 577

dpmhyperv ShowVMNics コマンド -- ネットワーク インターフェース カードをすべて表示する - 578

dpmhyperv ShowVMs コマンド -- VM をすべて表示する - 579

dpmhyperv ShowVMs (Funclet) コマンド -- VM
をすべて取得する - 351

dpmhyperv ShowVMSCSIControllers コマンド --
SCSI コントローラをすべて表示する - 581

dpmhyperv ValidateAgentSNMPAccess コマンド
-- エージェントへの SNMP アクセスを検証す
る - 582

dpmhyperv ValidateServerAccess コマンド -- サ
ーバアクセスを確認する - 583

dpmimg image コマンド -- Solaris イメージを展開
する - 865

dpmimg imgjobcheck コマンド -- Solaris イメージ
ングジョブのステータスを取得する - 870

dpmkpdb Command-- パフォーマンス データの
インポートまたはエクスポート - 795

dpmlpar cycle Command -- 論理パーティション
のサイクル - 587

dpmlpar delete コマンド -- 論理パーティション
の削除 - 592

dpmlpar getresources コマンド -- LPAR リソース
の取得 - 594

dpmlpar image コマンド -- IBM AIX LPAR を作成
する - 597

dpmlpar imgjobcheck コマンド -- IBM AIX イメー
ジングジョブステータスの取得 - 611

dpmlpar list コマンド -- HMC リソースのリスト表
示 - 613

dpmlpar setresources コマンド -- プロセッサ リ
ソースの更新 - 629

dpmlpar setresources コマンド -- プロセッサ リ
ソースの削除 - 624

dpmlpar setresources コマンド -- プロセッサ リ
ソースの追加 - 619

dpmlpar setresources コマンド -- メモリ リソース
の更新 - 627

dpmlpar setresources コマンド -- メモリ リソース
の削除 - 622

dpmlpar setresources コマンド -- メモリ リソース
の追加 - 616

dpmlpar-cycle コマンド -- 論理パーティションの
サイクル (Funclet) - 353

dpmlpar-delete コマンド -- 論理パーティション
の削除 (Funclet) - 357

dpmlpar-getresources コマンド -- LPAR リソース
の取得 (Funclet) - 358

dpmlpar-image コマンド -- IBM AIX LPAR の作成
(Funclet) - 360

dpmlpar-imgjobcheck コマンド -- IBM AIX イメー
ジングジョブステータスの取得 (Funclet) -
372

dpmlpar-list コマンド -- HMC リソースのリスト表
示 (Funclet) - 372

dpmlpar-setresources コマンド -- プロセッサ リ
ソースの削除 (Funclet) - 381

dpmlpar-setresources コマンド -- プロセッサ リ
ソースの追加 (Funclet) - 377

dpmlpar-setresources コマンド -- メモリ リソース
の削除 (Funclet) - 379

dpmlpar-setresources コマンド -- メモリ リソース
の追加 (Funclet) - 375

dpmmcs bringgrouponline コマンド -- リソース
グループのオンライン化 - 647

dpmmcs bringresourceonline コマンド -- リソー
スのオンライン化 - 648

dpmmcs getclusterinfo コマンド -- クラスタ情報
の取得 - 633

dpmmcs getgroupresourcesinfo コマンド -- クラ
スタグループのリソース情報の取得 - 634

dpmmcs getnetworkinterfacesinfo コマンド --
ネットワーク インターフェース情報の取得 -
635

dpmmcs getnetworksinfo コマンド -- ネットワー
ク情報の取得 - 637

dpmmcs getnodesinfo コマンド -- クラスタノー
ド情報の取得 - 638

dpmmcs getresourcegroupsinfo コマンド -- クラ
スタリソースグループ情報の取得 - 639

dpmmcs getresourcesdependenciesinfo コマン
ド -- リソースの依存関係情報の取得 - 640

dpmmcs getresourcetypesinfo コマンド -- クラ
スタリソースタイプ情報の取得 - 642

dpmmcs getservicesinfo コマンド -- クラスタサ
ービス情報の取得 - 643

dpmmscs movegroup コマンド -- 別のノードへのリソースグループの移動 - 644

dpmmscs pauseservice コマンド -- クラスタサービスの一時停止 - 646

dpmmscs resumeservice コマンド -- クラスタサービスの再開 - 650

dpmmscs startservice コマンド -- クラスタサービスの開始 - 651

dpmmscs stopservice コマンド -- クラスタサービスの停止 - 652

dpmmscs takegroupoffline コマンド -- リソースグループのオフライン化 - 654

dpmmscs takeresourceoffline コマンド -- リソースのオフライン化 - 655

dpmovf import コマンド -- OVF パッケージのインポート - 793

dmpolicy createdefaultactions コマンド -- デフォルトアクションを作成する - 872

dmpolicy exportaction コマンド -- アクションをファイルにエクスポートする - 874

dmpolicy exportrule コマンド -- ルールをファイルにエクスポートする - 875

dmpolicy findbestmachine コマンド -- ベストサーバを検索する - 876

dmpolicy findcandidatemachine コマンド -- 候補サーバを検索する - 877

dmpolicy getrulerecount コマンド -- ルール実行回数を取得する - 880

dmpolicy importaction コマンド -- アクションをファイルからインポートする - 882

dmpolicy importrule コマンド -- ルールをファイルからインポートする - 883

dmpolicy resetrulerecount コマンド -- ルールの試行回数をリセットする - 884

dmpolicy runaction コマンド -- アクションを実行する - 886

dmpolicy setrulereclimit コマンド -- ルールの制限を設定する - 889

dpmresourcemgr addbaremetal コマンド -- 物理コンピュータを追加する - 894

dpmresourcemgr addcomputertonetwork コマンド -- ネットワークにサーバを追加する - 895

dpmresourcemgr addipaddresspool コマンド -- IP アドレスプールを追加する - 897

dpmresourcemgr addmachinestostervice コマンド -- サービスにサーバを追加する - 899

dpmresourcemgr addtocontainer コマンド -- コンテナへの内容物の追加 - 900

dpmresourcemgr createcontainer コマンド -- コンテナの作成 - 901

dpmresourcemgr createdefaultservice コマンド -- デフォルトサービスを作成する - 902

dpmresourcemgr createipnetwork コマンド -- ネットワークを作成する - 904

dpmresourcemgr createresourcegroup コマンド -- リソースグループの作成 - 906

dpmresourcemgr createresourcetemplate コマンド -- サーバグループテンプレートの作成 - 908

dpmresourcemgr createservice コマンド -- サービスを作成する - 910

dpmresourcemgr deletebaremetal コマンド -- 物理コンピュータを削除する - 911

dpmresourcemgr deletecontainer コマンド -- コンテナの削除 - 913

dpmresourcemgr deleteipnetworks コマンド -- ネットワークを削除する - 914

dpmresourcemgr deletesystem コマンド -- システムを削除する - 915

dpmresourcemgr getdiscoverednetworks コマンド -- 検出されたネットワークを取得する - 916

dpmresourcemgr getipaddresspools コマンド -- IP アドレスプールをリスト表示する - 917

dpmresourcemgr getipnetworks コマンド -- ネットワークをリスト表示する - 919

dpmresourcemgr getserverlist コマンド -- サーバのリストを取得する - 920

dpmresourcemgr getservicelist コマンド -- サービスのリストを取得する - 921

dpmresourcemgr getsubnetdefaults コマンド -- サブネットのデフォルト値をリスト表示する - 922

dpmresourcemgr
removecomputerfromipnetwork コマンド -- ネットワークからサーバを削除する - 924

dpmresourcemgr removefromcontainer コマンド -- コンテナの内容物の削除 - 925

dpmresourcemgr removeipaddresspool コマンド -- IP アドレス プールを削除する - 926

dpmresourcemgr removemachinesfromservice コマンド -- サービスからサーバを削除する - 928

dpmresourcemgr removeresourcegroup コマンド -- リソース グループの削除 - 929

dpmresourcemgr removeservice コマンド -- サービスを削除する - 930

dpmresourcemgr removesourcetemplate コマンド -- リソース テンプレートの削除 - 932

dpmresourcemgr retrieveresourcegroup コマンド -- リソース グループの取得 - 933

dpmresourcemgr retrieveresourcetemplate コマンド -- リソース テンプレートの取得 - 934

dpmresourcemgr setcontainername コマンド -- コンテナ名の設定 - 936

dpmresourcemgr setcontainerservice コマンド -- サービスへのコンテナの設定 - 937

dpmresourcemgr setmanagementip コマンド -- ホストの管理 IP を設定する - 939

dpmresourcemgr setmanagementstatus コマンド -- ホストの管理ステータスを設定する - 940

dpmresourcemgr setserviceproperties コマンド -- サービスプロパティを設定する - 941

dpmresourcemgr updateipaddresspool コマンド -- IP アドレス プールを更新する - 952

dpmresourcemgr updateipnetwork コマンド -- ネットワークを更新する - 948

dpmresourcemgr updateresourcegroup コマンド -- リソース グループの更新 - 943

dpmresourcemgr updateresourcetemplate コマンド -- リソース テンプレートの更新 - 945

dpmresourcemgr updateservice コマンド -- サービスを更新する - 946

dpmresourcemgr updatesubnetdefaults コマンド -- ネットワークのデフォルト値を更新する - 950

dpmrpt schedulereport コマンド -- レポートのスケジューリング - 851

dpmssh run コマンド -- リモートシェル コマンド またはスクリプトの実行 - 854

dpmucs associateserviceprofile コマンド -- サービスプロファイルとブレードの関連付け - 452

dpmucs cloneserviceprofile コマンド -- サービスプロファイルのクローン作成 - 453

dpmucs createmacpool コマンド -- MAC アドレスプールの作成 - 455

dpmucs createorg コマンド -- 組織の作成 - 457

dpmucs createserverpool コマンド -- サーバプールの作成 - 458

dpmucs createserviceprofile コマンド -- サービスプロファイルの作成 - 460

dpmucs createtemplatefromserviceprofile コマンド -- サービスプロファイルからのテンプレートの作成 - 462

dpmucs createuuidpool コマンド -- UUID プールの作成 - 463

dpmucs createwwpool コマンド -- WWNN プールの作成 - 465

dpmucs deletemacpool コマンド -- MAC アドレスプールの削除 - 467

dpmucs deleteorg コマンド -- 組織の削除 - 469

dpmucs deleteserverpool コマンド -- サーバプールの削除 - 470

dpmucs deleteserviceprofile -- サービスプロファイルの削除 - 472

dpmucs deleteuuidpool コマンド -- UUID プールの削除 - 473

dpmucs deletewwpool コマンド -- WWNN プールの削除 - 475

dpmucs deployimage コマンド -- Cisco UCS イメージの展開 - 477

dpmucs disassociateserviceprofile コマンド -- サービスプロファイルの関連付け解除 - 479

dpmucs failoverserviceprofile コマンド -- フェールオーバーのためのサービス プロファイルとブレードの関連付け - 481

dpmucs getblades コマンド -- ブレード情報の取得 - 483

dpmucs getchassis コマンド -- シャーシ情報の取得 - 484

dpmucs getpolicy コマンド -- UCS Manager からのポリシーの取得 - 486

dpmucs getserviceprofilestatus コマンド -- サービス プロファイル ステータスの取得 - 487

dpmucs getswitches コマンド -- スイッチ情報の取得 - 489

dpmucs gettopsystem コマンド -- システム情報の取得 - 490

dpmucs listserviceprofiles コマンド -- サービス プロファイルの一覧表示 - 492

dpmucs powercycle コマンド -- ブレードの電源サイクル操作 - 493

dpmucs renamepool コマンド -- プール名の変更 - 495

dpmucs serviceprofileupdate コマンド -- サービス プロファイル操作の実行 - 497

dpmutil ecs コマンド -- Windows スケジューラの設定 - 799

dpmutil -hmc コマンド -- HMC サーバまたは IVM サーバの設定 - 800

dpmutil mgmtldb コマンド -- 管理データベースの設定 - 802

dpmutil remotevcaim コマンド -- リモート VC AIM の設定 - 804

dpmutil vcdefault コマンド -- デフォルトの VMware vCenter 管理者ユーザの設定 - 811

dpmutil vcserver コマンド -- VMware vCenter の設定 - 812

dpmutil -vioserver コマンド -- VIO サーバの設定 - 813

dpmutil -amq コマンド -- ActiveMQ Message Broker の設定 - 798

dpmutil -hd コマンド -- ヘルプ デスクの設定 - 799

dpmutil -hyperv コマンド -- Hyper-V Server の設定 - 801

dpmutil -perfddb コマンド -- パフォーマンス データベースの設定 - 803

dpmutil -scvmmserver コマンド -- SCVMM Server の設定 - 806

dpmutil -sc コマンド -- サービス コントローラの設定 - 804

dpmutil -snmp コマンド -- SNMP の設定 - 807

dpmutil -systemcredential コマンド -- 外部システムの認証情報の設定 - 809

dpmutil -sysuser コマンド -- サービス起動時に使用されるローカル認証情報の設定 - 810

dpmutil の構文 - 796

dpmvc addesxhost コマンド -- ESX ホストを追加する - 691

dpmvc addvmvdisk コマンド -- 仮想ディスクの追加 - 694

dpmvc addvmvnic コマンド -- 仮想 NIC の追加 - 697

dpmvc clone コマンド -- VM のクローン作成 - 700

dpmvc createdatastore コマンド -- データストアの作成 - 710

dpmvc cycle コマンド -- VM のサイクル - 713

dpmvc datastore コマンド -- データストア プロパティの取得 - 717

dpmvc delete コマンド -- VM の破棄 - 720

dpmvc distributedswitch コマンド -- 分散仮想スイッチの管理 - 724

dpmvc entermaintenancemode コマンド -- 保守モードに設定する - 730

dpmvc exitmaintenancemode コマンド -- 保守モードの終了 - 732

dpmvc faulttolerance コマンド -- フォールトトレラント操作の指定 - 734

dpmvc getavailablescsidisks コマンド -- 利用可能な SCSI ディスクの取得 - 736

dpmvc gethosthba コマンド -- ホスト バス アダプタの取得 - 738

dpmvc getresources コマンド -- VM リソースの取得 - 739

dpmvc getshares コマンド -- VM 共有の取得 - 742

dpmvc getsnapshots コマンド -- VM スナップショットの取得 - 745

dpmvc imgjobcheck コマンド -- クローン作成ジョブステータスの取得 - 747

dpmvc migrate コマンド -- VM のマイグレート - 750

dpmvc removedatastore コマンド -- データストアの削除 - 753

dpmvc removevmvmdisk コマンド -- 仮想ディスクの削除 - 755

dpmvc removevmvnic コマンド -- 仮想 NIC の削除 - 757

dpmvc setcpu コマンド -- VM の CPU 設定 - 759

dpmvc setmemory コマンド -- VM のメモリサイズの設定 - 761

dpmvc setresources コマンド -- VM リソースの設定 - 763

dpmvc setshares コマンド -- VM 共有の設定 - 767

dpmvc snapshot コマンド -- スナップショットを管理する - 772

dpmvc templatetovm コマンド -- テンプレートを VM に変換する - 778

dpmvc vapp コマンド -- vApp の管理 - 230, 782

dpmvc virtualswitch コマンド -- 仮想スイッチの管理 - 786

dpmvc vmtotemplate コマンド -- VM をテンプレートに変換する - 790

dpmvc addesxhost コマンド -- ESX ホスト (Funclet) の追加 - 192

dpmvc addvmvmdisk コマンド -- 仮想ディスクの追加 (Funclet) - 194

dpmvc addvmvnic コマンド -- 仮想 NIC の追加 (Funclet) - 195

dpmvc clone コマンド -- VM のクローン作成 (Funclet) - 197

dpmvc createdatastore コマンド -- データストアの作成 (Funclet) - 205

dpmvc cycle コマンド -- VM のサイクル (Funclet) - 207

dpmvc-datastore コマンド -- データストア プロパティの取得 (Funclet) - 208

dpmvc-delete コマンド -- VM の破棄 (Funclet) - 209

dpmvc-distributedswitch コマンド -- 分散仮想スイッチの管理 (Funclet) - 210

dpmvc-entermaintenancemode コマンド -- 保守モードに設定する (Funclet) - 213

dpmvc-exitmaintenancemode コマンド -- 保守モードの終了 (Funclet) - 214

dpmvc-faulttolerance コマンド -- フォールトトレラント操作の指定 (Funclet) - 215

dpmvc-getavailablescsidisks コマンド -- 利用可能な SCSI ディスクの取得 (Funclet) - 216

dpmvc-gethosthba コマンド -- ホストバスアダプタの取得 (Funclet) - 217

dpmvc-getresources コマンド -- VM リソースの取得 (Funclet) - 218

dpmvc-getshares コマンド -- VM 共有の取得 (Funclet) - 219

dpmvc-getsnapshots コマンド -- VM のスナップショットのリスト表示 (Funclet) - 220

dpmvc-imgjobcheck コマンド -- クローン作成ジョブステータスの取得 (Funclet) - 220

dpmvc-migrate コマンド -- VM のマイグレート (Funclet) - 221

dpmvc-removedatastore コマンド -- データストアの削除 (Funclet) - 222

dpmvc-removevmvmdisk コマンド -- 仮想ディスクの削除 (Funclet) - 223

dpmvc-removevmvnic コマンド -- 仮想 NIC の削除 (Funclet) - 224

dpmvc-setresources コマンド -- VM リソースの設定 (Funclet) - 224

dpmvc-setshares コマンド -- VM 共有の設定 (Funclet) - 225

dpmvc-snapshot コマンド -- スナップショットの管理 (Funclet) - 227

dpmvc-templatetovm コマンド -- テンプレートを VM に変換する (Funclet) - 229

dpmvc-virtualswitch コマンド -- 仮想スイッチの管理 (Funclet) - 234

dpmvc-vmtotemplate コマンド -- VM をテンプレートに変換する (Funclet) - 236

dpmvc-wait コマンド -- ジョブの待機時間 (Funclet) - 236

dpmzone associateproject コマンド -- タスクとプロジェクトの関連付け - 656

dpmzone clonezone コマンド -- ゾーンのクローン作成 - 658

dpmzone createandinstallzone コマンド -- ゾーンの作成およびインストール - 659

dpmzone createpool コマンド -- リソース プールの作成 - 662

dpmzone createproject コマンド -- プロジェクトの作成 - 664

dpmzone createzone コマンド -- ゾーンの作成 - 665

dpmzone customcommand コマンド -- カスタム コマンドの実行 - 668

dpmzone deleteproject コマンド -- プロジェクトの削除 - 669

dpmzone deletezone コマンド -- ゾーンの削除 - 670

dpmzone installzone コマンド -- ゾーンのインストール - 671

dpmzone movezone コマンド -- ゾーンの移動 - 673

dpmzone rebootzone コマンド -- ゾーンの再起動 - 674

dpmzone renamezone コマンド -- ゾーンの名称変更 - 675

dpmzone setpoolscheduler コマンド -- プール スケジューラの設定 - 676

dpmzone setprocessorset コマンド -- プロセッサ セットの CPU 最大数の設定 - 678

dpmzone showinterfaces コマンド -- ネットワーク インターフェースの表示 - 679

dpmzone showpools コマンド -- リソース プールの表示 - 680

dpmzone showprojects コマンド -- プロジェクトのプロパティの表示 - 681

dpmzone showresourceset コマンド -- リソース セットの表示 - 682

dpmzone showtasks コマンド -- プロジェクトのタスクのリスト表示 - 683

dpmzone showzoneinterfaces コマンド -- ゾーンのネットワーク インターフェースの表示 - 684

dpmzone showzoneprocess コマンド -- ゾーンのプロセスの表示 - 685

dpmzone showzoneprojects コマンド -- ゾーンのプロジェクトの表示 - 686

dpmzone showzones コマンド -- ゾーンの表示 - 687

dpmzone startzone コマンド -- ゾーン の起動 - 687

dpmzone stopzone コマンド -- ゾーン の停止 - 688

dpmzone uninstallzone コマンド -- ゾーン のアンインストール - 690

dpmzone-associateproject コマンド -- タスクとプロジェクトの関連付け (Funclet) - 383

dpmzone-clonezone コマンド -- ゾーン のクローン作成 (Funclet) - 384

dpmzone-createandinstallzone コマンド -- ゾーン の作成およびインストール - 385

dpmzone-createpool コマンド -- リソース プールの作成 (Funclet) - 388

dpmzone-createproject コマンド -- プロジェクトの作成 (Funclet) - 389

dpmzone-createzone コマンド -- ゾーン の作成 (Funclet) - 390

dpmzone-customcommand コマンド -- カスタム コマンドの実行 (Funclet) - 393

dpmzone-deleteproject コマンド -- プロジェクトの削除 (Funclet) - 393

dpmzone-deletezone コマンド -- ゾーン の削除 (Funclet) - 394

dpmzone-installzone コマンド -- ゾーン のインストール (Funclet) - 395

dpmzone-movezone コマンド -- ゾーン の移動 (Funclet) - 396

dpmzone-rebootzone コマンド -- ゾーン の再起動 (Funclet) - 396

dpmzone-renamezone コマンド -- ゾーン の名称変更 (Funclet) - 397

dpmzone-setpoolscheduler コマンド -- プール スケジューラの設定 (Funclet) - 398
dpmzone-setprocessorset コマンド -- プロセッサセットの CPU 最大数の設定 (Funclet) - 399
dpmzone-showinterfaces コマンド -- ネットワーク インターフェースの表示 (Funclet) - 400
dpmzone-showpools コマンド -- リソース プールの表示 (Funclet) - 400
dpmzone-showprojects コマンド -- プロジェクトのプロパティのリスト表示 (Funclet) - 401
dpmzone-showresourceset コマンド -- リソースセットの表示 (Funclet) - 402
dpmzone-showtasks コマンド -- プロジェクトのタスクのリスト表示 (Funclet) - 402
dpmzone-showzoneinterfaces コマンド -- ゾーンのネットワーク インターフェースの表示 (Funclet) - 403
dpmzone-showzoneprocess コマンド -- ゾーンのプロセスの表示 (Funclet) - 404
dpmzone-showzoneprojects コマンド -- ゾーンのプロジェクトの表示 (Funclet) - 404
dpmzone-showzones コマンド -- ゾーンを表示 (Funclet) - 405
dpmzone-startzone コマンド -- ゾーンの起動 (Funclet) - 405
dpmzone-stopzone コマンド -- ゾーンの停止 (Funclet) - 406
dpmzone-uninstallzone コマンド -- ゾーンのアンインストール (Funclet) - 406

E

exit、quit コマンド -- AutoShell セッションの終了 (Cmdlet) - 74
external コマンド -- ネイティブ外部関数の宣言 (Cmdlet) - 75

G

get-autoShellClassInfo コマンド -- Autoshell ネイティブ クラスのメソッドとプロパティの表示 (Cmdlet) - 76
gete -- 環境変数を取得する (関数) - 135

get-help コマンド -- ヘルプ情報の表示 (Funclet) - 77
get-remoteResult コマンド -- リモート ターゲットから結果を取得 (Funclet) - 78
get-webServiceInfo コマンド -- WSDL 情報の表示 (Cmdlet) - 79

H

help コマンド -- ヘルプ情報をリクエスト (Cmdlet) - 80

I

ism-applyPoliciesAndTemplatesForService コマンド -- すべてのシステムにポリシーおよびテンプレートを適用する (Funclet) - 415
ism-applyPoliciesAndTemplates コマンド -- ホストのグループへのポリシーおよびテンプレートの適用 (Funclet) - 411
ism-applyPoliciesForService コマンド -- すべてのシステムにポリシーを適用する (Funclet) - 412
ism-applyPolicies コマンド -- ホストのグループへのポリシーの適用 (Funclet) - 408
ism-applyPolicyForService コマンド -- すべてのシステムポリシーを適用する (Funclet) - 407
ism-applyPolicyTemplatesForService コマンド -- すべてのシステムにポリシー テンプレートを適用する (Funclet) - 413
ism-applyPolicyTemplates コマンド -- ホストのグループへのポリシー テンプレートの適用 (Funclet) - 410
ism-applyPolicyWait コマンド -- ポリシーを適用して待機する (Funclet) - 409
ism-cancelJob コマンド -- ジョブをキャンセルする (Funclet) - 416
ism-copyPolicyTemplate コマンド -- ポリシー テンプレートをコピーする (Funclet) - 441
ism-copyPolicy コマンド -- ポリシーをコピーする (Funclet) - 417
ism-createJobTemplate コマンド -- ジョブ テンプレートを作成する (Funclet) - 418

ism-deleteJobTemplate コマンド -- ジョブ テンプレートを削除する(Funclet) - 418

ism-deletePolicyTemplate コマンド -- ポリシーテンプレートを削除する(Funclet) - 442

ism-deletePolicy コマンド -- ポリシーを削除する(Funclet) - 419

ism-deployPackageToHost コマンド -- ホストにパッケージを展開する(Funclet) - 419

ism-deployTemplateForService コマンド -- テンプレートのパッケージをサービスに展開する(Funclet) - 423

ism-deployTemplateToHost コマンド -- テンプレートのパッケージをホストに展開する(Funclet) - 424

ism-deployTemplate コマンド -- テンプレートからパッケージを展開する(Funclet) - 421

ism-encryptString コマンド -- 文字列を暗号化する(Funclet) - 426

ism-listComputerSystems コマンド -- すべてのコンピュータシステムを表示する(Funclet) - 426

ism-listDeployedSystems コマンド -- パッケージを展開したすべてのホストをリスト表示する(Funclet) - 427

ism-listJobStatus コマンド -- ジョブ ステータスをリスト表示する(Funclet) - 428

ism-listJobs コマンド -- すべてのジョブをリスト表示する(Funclet) - 428

ism-listJobTemplates コマンド -- ジョブ テンプレートをすべてリスト表示する(Funclet) - 429

ism-listPackages コマンド -- すべてのパッケージをリスト表示する(Funclet) - 429

ism-listPolicies コマンド -- すべてのポリシーをリスト表示する(Funclet) - 430

ism-listPolicyStatus コマンド -- ポリシー ステータスをリスト表示する(Funclet) - 430

ism-listPolicyTemplateStatus コマンド -- ポリシー テンプレート ステータスをリスト表示する(Funclet) - 442

ism-listPolicyTemplate コマンド -- すべてのポリシー テンプレートをリスト表示する(Funclet) - 447

ism-listServiceSystems コマンド -- サービスのすべてのコンピュータシステムをリスト表示する(Funclet) - 431

ism-listSysEdgeConfig コマンド -- SystemEDGE の設定をリスト表示する(Funclet) - 431

ism-listSystemPoliciesForService コマンド -- サービスのすべてのポリシーをリスト表示する(Funclet) - 432

ism-listSystemPolicies コマンド -- すべてのポリシーをリスト表示する(Funclet) - 434

ism-listTemplateEulas コマンド -- テンプレート内の EULA をリスト表示する(Funclet) - 432

ism-removePolicyTemplate コマンド -- ポリシーテンプレートを削除する(Funclet) - 443

ism-renamePolicyTemplate コマンド -- ポリシーテンプレートの名前を変更する(Funclet) - 444

ism-renamePolicy コマンド -- ポリシーの名前を変更する(Funclet) - 433

ism-setMaintenanceModeForService コマンド -- サービスを保守モードに設定する(Funclet) - 435

ism-setMaintenanceMode コマンド -- ホストの保守モードを設定する(Funclet) - 434

ism-setSysEdgeLogLevelForService コマンド -- サービスの SystemEDGE ログ レベルを設定する(Funclet) - 437

ism-setSysEdgeLogLevel コマンド -- SystemEDGE のログ レベルを設定する(Funclet) - 436

ism-updateAllExceptionSystems コマンド -- 最新のポリシーを適用する(Funclet) - 437

ism-updateAllOldPolicies コマンド -- 古いポリシーをすべて更新する(Funclet) - 438

ism-updateAllPolicies コマンド -- すべてのポリシーを更新する(Funclet) - 438

ism-updateAllUpToDateSystems コマンド -- 最新のシステムをすべて更新する(Funclet) - 438

ism-updateExceptionSystemsTemplate コマンド -- 最新のポリシー テンプレートを適用する(Funclet) - 444

ism-updateExceptionSystems コマンド -- 最新のポリシーを適用する(Funclet) - 439
ism-updateOldSystemsTemplate コマンド -- 古いシステム テンプレートをすべて更新する(Funclet) - 445
ism-updateOldSystems コマンド -- 古いポリシーを更新する(Funclet) - 439
ism-updatePolicies コマンド -- ポリシーの設定を更新する(Funclet) - 440
ism-updateSystemsTemplate コマンド -- システム テンプレートを適用する(Funclet) - 446
ism-updateUpToDateSystemsTemplate コマンド -- システム テンプレートを再適用する(Funclet) - 446
ism-updateUpToDateSystems コマンド -- ポリシーを再適用する(Funclet) - 440

M

memoRead -- 文字列にテキスト ファイルを読み込む(関数) - 136
memoWrit -- 文字列をファイルに書き込む(関数) - 137
mkdir コマンド -- ディレクトリの作成(Cmdlet) - 81
mv、ren コマンド -- ファイルとディレクトリの移動(Funclet) - 84

N

new-comObject コマンド -- COM オブジェクトのインスタンス化(Funclet) - 87
new-SSHSession コマンド -- 安全な SSH セッションの新規作成(Funclet) - 90
new-webService コマンド -- Web サービス オブジェクトの作成(Funclet) - 92

O

objdump コマンド -- オブジェクトの列挙可能なプロパティの表示(Cmdlet) - 93
OSRedirect.clear メソッド - 162
OSRedirect.errorOccurred メソッド - 163
OSRedirect.errout メソッド - 164

OSRedirect.hasCompleted メソッド - 165
OSRedirect.onCompleted メソッド - 166
OSRedirect.onError メソッド - 167
OSRedirect.OSRedirect コンストラクタ - 161
OSRedirect.output メソッド - 168
OSRedirect.receivedErrOutput メソッド - 169
OSRedirect.receivedOutput メソッド - 170
OSRedirect.receivedResult メソッド - 171
OSRedirect.result メソッド - 172
OSRedirect クラス - 160
OSRedirect クラスの使用 - 37

P

platform -- オペレーティング システム タイプを照会する(関数) - 138
push-client コマンド -- Autoshell クライアントをリモート Windows システムにインストール(Funclet) - 94
push-winRemote コマンド -- 複数のリモート Windows システム上でのコピーと実行(Funclet) - 97
pute -- 環境変数を設定する(関数) - 138
PW_GET コマンド -- 非表示データの入力(Funclet) - 100
pwd コマンド -- 作業ディレクトリ名の出力(Cmdlet) - 101

Q

qout -- 出力に続いて改行を stdout に書き込む(関数) - 140
qqout -- stdout に出力を書き込む(関数) - 141
query-service コマンド -- Windows サービスのステータスの問い合わせ(Funclet) - 101

R

regCreateKey -- レジストリ キーを作成する(関数) - 142
regCreateSubkeys -- 配列からサブキーを作成する(関数) - 143
regDeleteKey -- レジストリ キーまたはキー階層を削除する(関数) - 144

regDeleteVal -- レジストリの値を削除する(関数) - 145
regGetKeyValues -- レジストリ キーの値情報を取得する(関数) - 146
regGetSubKey -- レジストリ キーのサブ キーを取得する(関数) - 148
regGetVal -- レジストリの値を取得する(関数) - 150
regIsKey -- レジストリ キーの存在を確認する(関数) - 151
regIsVal -- レジストリの値の存在を確認する(関数) - 153
regSetKeyValues -- 配列のレジストリ キーの値を設定する(関数) - 154
regSetVal -- レジストリの値を設定する(関数) - 155
rem-client コマンド -- リモート Windows システムから Autoshell クライアントを削除(Funclet) - 103
RemoteTarget.abort メソッド - 177
RemoteTarget.createRemoteContext メソッド - 178
RemoteTarget.destroyRemoteContext メソッド - 179
RemoteTarget.errorOccurred メソッド - 180
RemoteTarget.execute メソッド - 181
RemoteTarget.getError メソッド - 182
RemoteTarget.getHostNames メソッド - 184
RemoteTarget.hasCompleted メソッド - 185
RemoteTarget.onError メソッド - 186
RemoteTarget.output メソッド - 187
RemoteTarget.receivedOutput メソッド - 188
RemoteTarget.receivedResult メソッド - 188
RemoteTarget.RemoteTarget コンストラクタ - 175
RemoteTarget.result メソッド - 189
RemoteTarget.wasAborted メソッド - 191
RemoteTarget クラス - 173
RemoteTarget クラスの使用 - 42
rmdir コマンド -- ディレクトリの削除(Funclet) - 104

rmonwatch add コマンド -- MIB テーブルに行を追加する - 954
rmonwatch del コマンド -- MIB テーブルから行を削除する - 957
rmonwatch getTable コマンド -- rmonWbem MIB テーブルを表示する - 959
rmonwatch getVersion コマンド -- バージョン情報を表示する - 961
run-client コマンド -- リモート Windows システム上で Autoshell クライアントを実行(Funclet) - 106
run-local Command の使用 - 39
run-local コマンド -- ローカル システム上でスクリプトを実行(Funclet) - 108
run-remote コマンドの使用 - 41
run-remote コマンド -- リモート システム上でスクリプトを実行(Funclet) - 110
run-SSHCommand コマンド -- SSH(Funclet)を使用してリモート コマンドを実行する - 115
run-SSHShell コマンド -- 対話型の SSH コマンドラインを実行する(Funclet) - 119
run-winRemote コマンド -- 複数のリモート Windows システムでコマンドを実行する(Funclet) - 122

S

set alternate to コマンド -- 別の出力ファイルを設定する(Cmdlet) - 125
set console コマンド -- コンソールへの出力を非表示にする(Cmdlet) - 126
set result display on/off コマンド -- 自動結果表示をオンまたはオフにする(Cmdlet) - 127
setProcExitCode -- AutoShell 終了コードを設定する(関数) - 157
shellType -- AutoShell タイプを照会する(関数) - 157
SNMP の詳細パラメータ - 962
start-java コマンド -- Java 仮想マシンを起動する(Cmdlet) - 130

start-service コマンド -- リモートまたはローカルシステムの Windows サービスを開始する (Funclet) - 127

stop-service コマンド -- リモートまたはローカルシステムの Windows サービスを停止する (Funclet) - 129

T

typeName -- 式のタイプ名を取得する(関数) - 158

W

wait コマンド -- キーが押されるのを待機する (Cmdlet) - 131

weak external コマンド -- ネイティブ外部関数を宣言する(Cmdlet) - 132

あ

イベント コマンド - 843

イメージング コマンド - 864

オペレーティング システムへのアクセス - 36

か

カテゴリ別コマンド、関数、クラス - 50

管理コマンドライン ユーティリティ - 795

関連ドキュメント - 16

キーボード(標準入力)からのデータの読み取り - 36

規則 - 17

コマンドライン スクリプト - 449

コマンドライン命令(CLI) - 449

コンソール(標準出力)へのデータの書き込み - 35

さ

収集エンジン コマンド - 819

スクリプトの実行 - 38

た

対象読者 - 15

対話型モードの使用 - 29

展開とポリシー設定の AutoShell コマンド - 407

は

はじめに - 15

バッチ モード - 31

ヘルプ デスク コマンド - 855

ポリシー コマンド - 872

ポリシー設定 - 891

本書の目的 - 15

ま

マネージャ シェルとクライアント シェル - 21

文字列化 - 33

や

有効な AutoShell ユーザ - 22

ら

リソース マネージャ コマンド - 893

リモート対話型モードの使用 - 30

リモートバッチ モード - 32

リモート モニタリング コマンド - 953

利用可能なログ ファイル - 965

ログ ファイル - 964

ログ ファイル設定 - 967