



# HP Universal CMDB

ソフトウェア・バージョン : コンテンツ・パック 17.00 (CP17)

## ディスカバリ / インテグレーション・コンテンツ・ガイド - ディスカバリ・モジュール

ドキュメント・リリース日 : 2015 年 9 月  
ソフトウェア・リリース日 : 2015 年 9 月

## ご注意

## 保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載は、追加保証を提供するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

## 権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

## 著作権について

© 2002 - 2015 Hewlett-Packard Development Company, L.P.

## 商標について

Adobe™ は Adobe Systems Incorporated の商標です。

Microsoft® および Windows® は、Microsoft Corporation の米国における登録商標です。

UNIX® は、The Open Group の登録商標です。

## ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ・ ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ・ ドキュメントリリース日は、ドキュメントが更新されるたびに更新されます。
- ・ ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を示します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。<https://softwaresupport.hp.com/>

このサイトを表示するには、HP Passportへの登録とサインインが求められます。HP Passport IDに登録するには、HPサポートサイトで**【登録】**をクリックするか、HP Passportのログインページで**【アカウントの作成】**をクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、最新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

## サポート

次のHPソフトウェアサポートのサイトを参照してください。<https://softwaresupport.hp.com>

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- ・ 関心のあるナレッジドキュメントの検索
- ・ サポートケースの登録とエンハンスメント要求のトラッキング
- ・ ソフトウェアパッチのダウンロード
- ・ サポート契約の管理
- ・ HPサポート窓口の検索
- ・ 利用可能なサービスに関する情報の閲覧
- ・ 他のソフトウェアカスタマーとの意見交換
- ・ ソフトウェアトレーニングの検索と登録

一部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDに登録するには、HPサポートサイトで**【登録】**をクリックするか、HP Passportのログインページで**【アカウントの作成】**をクリックします。

アクセスレベルの詳細については、次のWebサイトをご覧ください。<https://softwaresupport.hp.com/web/softwaresupport/access-levels>

**HP Software Solutions Now**は、HPSWのソリューションと統合に関するポータルWebサイトです。このサイトでは、お客様のビジネスニーズを満たすHP製品ソリューションを検索したり、HP製品間の統合に関する詳細なリストやTILプロセスのリストを閲覧することができます。このWebサイトのURLは<http://h20230.www2.hp.com/sc/solutions/index.jsp>です。

# 目次

第1部: Cloud and Virtualization > Cloud .....	27
第1章: Amazon Web Services ディスカバリ .....	28
概要 .....	29
トポロジ .....	30
Amazon EC2 .....	30
Amazon RDS .....	31
EC2 サービスおよび RDS サービスの検出方法 .....	32
AWS_by_WebServices アダプタ .....	34
AWS by Web Services ジョブ .....	36
トラブルシューティングおよび制限事項 .....	37
第2章: VMware vCloud ディスカバリ .....	38
概要 .....	39
サポートされるバージョン .....	39
トポロジ .....	40
vCloud Director による vCloud の検出方法 .....	41
URL を使用した vCloud の検出方法 .....	42
プローブへの vCloud SDK の依存関係の追加方法 .....	42
vCloud_Director_by_vCloud_API アダプタ .....	43
vCloud_Director_URL_by_vCloud_API アダプタ .....	45
vCloud Director by vCloud API ジョブ .....	47
vCloud Director URL by vCloud API ジョブ .....	48
トラブルシューティングおよび制限事項 .....	49
第2部: Cloud and Virtualization > Virtualization .....	50
第3章: Citrix Xen ディスカバリ .....	51
概要 .....	52
サポートされるバージョン .....	52
トポロジ .....	53
Citrix Xen トポロジの検出方法 .....	53
Citrix Xen Connection ジョブ .....	54
はじめに .....	54
アダプタ情報 .....	55
Citrix Xen Topology ジョブ .....	56
はじめに .....	56
アダプタ情報 .....	57

第4章: HP IVM ディスカバリ .....	60
概要 .....	61
サポートされるバージョン .....	61
トポロジ .....	61
HP IVM トポロジの検出方法 .....	62
HP IVM by Shell ジョブ .....	62
HP Integrity Virtual Machine by Shell アダプタ .....	63
第5章: HP Partitioning Solution ディスカバリ .....	65
概要 .....	66
サポートされるバージョン .....	66
トポロジ .....	67
HP vPars および nPars の検出方法 .....	70
HP nPars and vPars by Shell ジョブ .....	71
ディスカバリ・メカニズム .....	75
トラブルシューティングおよび制限事項 .....	99
第6章: Hyper-V ディスカバリ .....	100
概要 .....	101
サポートされるバージョン .....	101
トポロジ .....	101
Hyper-V の検出方法 .....	102
ディスカバリ・メカニズム .....	103
\\root\virtualization 名前空間のディスカバリ・メカニズム .....	104
\\root\virtualization\v2 名前空間のディスカバリ・メカニズム .....	110
Hyper-V Topology by Shell ジョブ .....	117
Hyper-V Topology by WMI ジョブ .....	119
作成または変更されたエンティティ .....	121
トラブルシューティングおよび制限事項 .....	122
第7章: IBM Virtualization ディスカバリ .....	123
概要 .....	124
サポートされるバージョン .....	124
トポロジ .....	125
IBM 仮想化の検出方法 .....	131
IBM Virtualization by Shell ジョブ .....	133
アダプタ情報 .....	134
IBM LPAR and VIO Server Topology by Shell ジョブ .....	136
はじめに .....	136
アダプタ情報 .....	137
IBM PureFlex Topology by Shell ジョブ .....	140
IBM 仮想化コマンド .....	143



VIO サーバ側のコマンド .....	155
LPAR 側のコマンド .....	165
作成または変更されたエンティティ .....	166
トラブルシューティングおよび制限事項 .....	168
第8章: Oracle VM Server for SPARC Technology ディスカバリ .....	169
概要 .....	170
サポートされるバージョン .....	170
トポロジ .....	171
Oracle VM Server for SPARC テクノロジーの検出方法 .....	173
Oracle VM Server for SPARC Technology by Shell アダプタ .....	174
Oracle VM Server for SPARC Technology by Shell ジョブ .....	177
ディスカバリ・フロー .....	178
全般 .....	178
Oracle VM Server for SPARC Technology by Shell ジョブのフロー .....	178
コマンド .....	179
トラブルシューティングおよび制限事項 .....	183
第9章: Oracle VM for x86 ディスカバリ .....	184
概要 .....	185
サポートされるバージョン .....	185
トポロジ .....	186
Oracle VM for x86 トポロジの検出方法 .....	187
Oracle VM for x86 by Manager Main CLI ジョブ .....	187
Oracle VM Manager Discovery By MainCli アダプタ .....	189
第10章: Solaris Zones ディスカバリ .....	193
概要 .....	194
サポートされるバージョン .....	194
トポロジ .....	195
Solaris Zones の検出方法 .....	196
Solaris Zones by TTY ジョブ .....	197
トリガ・クエリ .....	197
アダプタ .....	197
パラメータ .....	197
作成または変更されたエンティティ .....	198
ディスカバリ・メカニズム .....	200
トラブルシューティングおよび制限事項 .....	211
第11章: VMware .....	212
VMware の前提条件 .....	212
VMware Infrastructure ディスカバリ .....	215
サポートされるバージョン .....	217

VIM プロトコルでサポートされるバージョン .....	217
CIM プロトコルでサポートされるバージョン .....	217
VIM プロトコルの SSL サポート .....	217
トポロジ .....	217
VMware VIM Topology の検出方法 .....	226
Manual VMware VIM Connection ジョブの実行方法 .....	229
CIM を介した VMware ESX サーバ・トポロジの検出方法 .....	229
Manual VMware VIM Connection ジョブ .....	231
VMware ESX Connection by CIM ジョブ .....	232
VMware ESX Connection by VIM ジョブ .....	235
VMware ESX Topology by CIM ジョブ .....	238
VMware ESX Topology by VIM ジョブ .....	241
VMware vCenter Connection by VIM ジョブ .....	245
VMware vCenter Topology by VIM ジョブ .....	247
VMware VMotion ディスカバリおよびイベント追跡 .....	253
概要 .....	254
サポートされるバージョン .....	254
VMware VMotion の検出方法とイベントの追跡方法 .....	254
VMware vMotion Monitor by VIM ジョブ .....	255
VMware ディスカバリのトラブルシューティングおよび制限事項 .....	258
トラブルシューティング .....	259
制限事項 .....	259
第 12 章: Xen and KVM ディスカバリ .....	261
概要 .....	262
サポートされるバージョン .....	262
トポロジ .....	262
ディスカバリ・メカニズム .....	263
KVM および Xen の検出方法 .....	264
Xen and KVM by Shell アダプタ .....	264
Xen and KVM by Shell ジョブ .....	267
第 3 部: Clustering and Load Balancing > Failover Clusters .....	269
第 13 章: EMC AutoStart ディスカバリ .....	270
概要 .....	271
サポートされるバージョン .....	271
トポロジ .....	271
EMC AutoStart の検出方法 .....	273
EMC AutoStart by Shell ジョブ .....	274
アダプタ .....	274

トリガ・クエリ .....	274
パラメータ .....	275
EMC_AutoStart_by_Shell アダプタ .....	276
ディスカバリ・フロー .....	279
EMC AutoStart ディスカバリ・コマンド .....	281
第 14 章: IBM High Availability Cluster Multiprocessing(HACMP)ディスカバリ .....	283
概要 .....	284
サポートされるバージョン .....	284
トポロジ .....	285
IBM HACMP の検出方法 .....	286
ディスカバリ・メカニズム .....	288
HACMP Topology Discovery ジョブ .....	294
HACMP Application Discovery ジョブ .....	295
第 15 章: Microsoft Cluster ディスカバリ .....	299
Microsoft Cluster Server View トポロジ .....	300
サポートされるバージョン .....	301
Microsoft Cluster Server の検出方法 .....	302
MS Cluster by NTCMD or UDA ジョブ .....	302
第 16 章: Red Hat Cluster Suite ディスカバリ .....	306
概要 .....	307
サポートされるバージョン .....	307
トポロジ .....	307
Red Hat Cluster Suite トポロジの検出方法 .....	308
Red Hat Cluster by Shell ジョブ .....	308
Red Hat Cluster by Shell アダプタ .....	310
制限事項 .....	312
第 17 章: HP ServiceGuard Cluster ディスカバリ .....	313
概要 .....	314
サポートされるバージョン .....	314
トポロジ .....	314
HP ServiceGuard クラスター・トポロジの検出方法 .....	315
Service Guard Cluster Topology by TTY ジョブ .....	316
Service Guard Cluster Topology アダプタ .....	318
HP ServiceGuard クラスターのコマンド .....	319
第 18 章: Solaris Cluster ディスカバリ .....	326
概要 .....	327
サポートされるバージョン .....	327
トポロジ .....	327

Sun クラスタの検出方法 .....	329
Sun Cluster by Shell ジョブ .....	329
Sun Cluster by Shell アダプタ .....	331
Solaris Cluster ディスカバリ・コマンド .....	333
クラスタの名前の取得 .....	333
クラスタのノードの取得 .....	335
ノード名のIPへの解決 .....	335
ノードのステータスの取得 .....	336
リソース・グループおよびリソースの取得 .....	336
リソース・グループおよびリソースの詳細の取得 .....	337
クラスタ相互接続情報の取得 .....	348
定数構成の取得 .....	352
第19章: Veritas ディスカバリ .....	353
概要 .....	354
サポートされるバージョン .....	354
トポロジ .....	355
Veritas Cluster Server の検出方法 .....	356
Veritas Cluster by Shell ジョブ .....	356
第4部: Clustering and Load Balancing > Load Balancers .....	360
第20章: Load Balancer ディスカバリ .....	361
概要 .....	362
サポートされるバージョン .....	362
トポロジ .....	363
ロード・バランサの検出方法 .....	363
Alteon Application Switch by SNMP ジョブ .....	366
アダプタ情報 .....	367
F5 BIG-IP LTM by SNMP ジョブ .....	369
アダプタ情報 .....	370
F5 BIG-IP LTM by Shell ジョブ .....	372
バージョン .....	372
前提条件 .....	372
アダプタ情報 .....	372
Cisco CSS by SNMP ジョブ .....	376
アダプタ情報 .....	377
Citrix NetScaler by SNMP ジョブ .....	379
アダプタ情報 .....	380
検出された CIT .....	381
第21章: Microsoft Network Load Balancing(NLB)ディスカバリ .....	384

概要 .....	385
サポートされるバージョン .....	385
トポロジ .....	386
Microsoft NLB(ネットワーク負荷分散)システムの検出方法 .....	387
コマンド・ライン・ユーティリティを使用した NLB の検出方法 .....	388
MS NLB by NTCMD or UDA ジョブ .....	389
MS NLB by NTCMD or UDA アダプタ .....	391
NLB(ネットワーク負荷分散)アーキテクチャのコンポーネント .....	396
用語集 .....	397
<b>第5部: Database .....</b>	<b>399</b>
第22章: ホスト資格情報によるデータベース接続 ディスカバリ .....	400
概要 .....	401
サポートされるバージョン .....	401
トポロジ .....	402
Oracle .....	402
Microsoft SQL .....	402
ホスト資格情報によるデータベース接続の検出方法 .....	403
DB Connections by Shell ジョブ .....	403
DB Connections by WMI ジョブ .....	407
トラブルシューティングおよび制限事項 .....	408
第23章: IBM DB2 Database ディスカバリ .....	410
サポートされるバージョン .....	411
IBM DB2 トポロジ .....	411
DB2 フル・トポロジの検出方法 .....	412
Databases TCP Ports ジョブ .....	416
アダプタ情報 .....	419
DB2 Topology by SQL ジョブ .....	420
DB2 Universal Database Connection by SQL ジョブ .....	423
DB2 Topology by SQL アダプタ .....	426
DB2 Connection by SQL アダプタ .....	431
アプリケーション署名とプラグイン .....	434
アプリケーション署名 .....	434
プラグイン .....	434
PortNumberToPortName 構成 .....	435
トラブルシューティングおよび制限事項 .....	435
トラブルシューティング .....	435
制限事項 .....	435
第24章: HP NonStop ディスカバリ .....	436

概要 .....	437
サポートされるバージョン .....	437
トポロジ .....	438
HP NonStop の検出方法 .....	439
HP NonStop Topology by Shell ジョブ .....	441
HP NonStop ディスカバリ・コマンド .....	443
第 25 章: SAP HANA Database ディスカバリ .....	449
概要 .....	450
サポートされるバージョン .....	450
トポロジ .....	450
ディスカバリ・メカニズム .....	450
SAP HANA データベースの検出方法 .....	452
HanaDb by Shell ジョブ .....	453
HanaDb_by_Shell アダプタ .....	454
ディスカバリ・フロー .....	457
第 26 章: MS-SQL ディスカバリ .....	458
概要 .....	459
サポートされるバージョン .....	459
トポロジ .....	459
Microsoft SQL Server のデータベース・アプリケーションの検出方法 .....	461
OS 資格情報を使用した MS SQL Server コンポーネントの検出方法 .....	462
Microsoft SQL Server のデータベース・アプリケーションのディスカバリ .....	462
OS 資格情報による SQL サーバ・ディスカバリ .....	463
第 27 章: SAP MaxDB ディスカバリ .....	465
概要 .....	466
サポートされるバージョン .....	466
トポロジ .....	466
SAP MaxDB の検出方法 .....	467
MaxDb by Shell ジョブ .....	468
MaxDb by Shell アダプタ .....	468
第 28 章: データベース間の MySQL レプリケーション・ディスカバリ .....	472
概要 .....	473
サポートされるバージョン .....	473
トポロジ .....	473
MySQL の構成ジョブとレプリケーション・ジョブの検出方法 .....	474
MySQL by Shell ジョブ .....	475
トラブルシューティングおよび制限事項 .....	480
第 29 章: Oracle Database サーバ・ディスカバリ .....	481

概要 .....	482
サポートされるバージョン .....	482
トポロジ .....	483
Oracle Database サーバの検出方法 .....	483
Oracle TNS Names トポロジの検出方法 .....	484
Oracle Database サーバの検出方法 - ライトウェイト .....	485
Oracle Database Connection by SQL- Lightweight ジョブ .....	486
Oracle Connection by SQL- Lightweight アダプタ .....	487
Oracle Database Connection by SQL ジョブ .....	489
Oracle Connection by SQL アダプタ .....	490
Oracle TNS Names by LDAP ジョブ .....	492
Oracle TNS Names by LDAP アダプタ .....	493
Oracle Topology by SQL ジョブ .....	495
Oracle Database Topology by SQL アダプタ .....	496
トラブルシューティングおよび制限事項 .....	499
<b>第30章: Oracle Real Application Cluster(RAC)ディスカバリ .....</b>	<b>500</b>
概要 .....	501
サポートされるバージョン .....	501
トポロジ .....	501
Oracle Real Application Cluster(RAC)の検出方法 .....	502
Oracle Listeners by Shell ジョブ .....	503
Oracle RAC Topology by Shell ジョブ .....	507
構成アイテム .....	510
関係 .....	511
トラブルシューティングおよび制限事項 .....	512
<b>第31章: Sybase ディスカバリ .....</b>	<b>513</b>
概要 .....	514
サポートされるバージョン .....	514
トポロジ .....	515
Sybase の検出方法 .....	515
Sybase Database Connection by SQL ジョブ .....	516
Sybase Topology by SQL ジョブ .....	518
<b>第6部: Enterprise Applications .....</b>	<b>522</b>
<b>第32章: Active Directory ディスカバリ .....</b>	<b>523</b>
概要 .....	524
サポートされるバージョン .....	524
トポロジ .....	525
Active Directory のドメイン・コントローラとトポロジの検出方法 .....	527

Active Directory Connection by LDAP ジョブ .....	528
Active Directory Topology by LDAP ジョブ .....	531
第 33 章: Cisco UCS .....	534
概要 .....	535
Cisco UCS Connection ジョブ .....	535
バージョン .....	535
前提条件 .....	535
アダプタ情報 .....	535
Cisco UCS Topology ジョブ .....	536
バージョン .....	536
前提条件 .....	536
アダプタ情報 .....	537
Cisco UCS Manual ジョブ .....	538
バージョン .....	539
前提条件 .....	539
アダプタ情報 .....	539
第 34 章: Microsoft Exchange Server with Active Directory ディスカバリ .....	542
概要 .....	543
サポートされるバージョン .....	544
トポロジ .....	545
Active Directory を使用した Microsoft Exchange Server トポロジの検出方法 .....	549
Microsoft Exchange Topology by LDAP ジョブ .....	550
トラブルシューティングおよび制限事項 .....	553
第 35 章: NTCMD または UDA による Microsoft Exchange Server ディスカバリ .....	554
概要 .....	555
サポートされるバージョン .....	555
トポロジ .....	555
NTCMD または UDA による Microsoft Exchange Server の検出方法 .....	559
Microsoft Exchange Connection by NTCMD or UDA ジョブ .....	560
Microsoft Exchange Topology by NTCMD or UDA ジョブ .....	561
作成または変更された CIT .....	563
第 36 章: Microsoft Exchange Server by PowerShell ディスカバリ .....	565
概要 .....	566
サポートされるバージョン .....	566
トポロジ .....	566
PowerShell を使用した Microsoft Exchange の検出方法 .....	570
PowerShell のリモート処理の構成方法 .....	571
Active Directory 側の構成方法 .....	572
Microsoft Exchange Topology by PowerShell ジョブ .....	574



トラブルシューティングおよび制限事項 .....	580
第 37 章: Microsoft Exchange Server by WMI ディスカバリ .....	582
概要 .....	583
サポートされるバージョン .....	583
トポロジ .....	584
WMI を使用した Microsoft Exchange Server 2003 の検出方法 .....	585
Microsoft Exchange Connection by WMI ジョブ .....	586
Microsoft Exchange Topology by WMI ジョブ .....	588
作成または変更された CIT .....	590
トラブルシューティングおよび制限事項 .....	591
第 38 章: Microsoft SharePoint ディスカバリ .....	592
概要 .....	593
サポートされるバージョン .....	593
トポロジ .....	594
Host Connection by Shell ジョブ .....	594
Host Applications by Shell ジョブ .....	594
Microsoft SharePoint Topology ジョブ .....	596
Microsoft SharePoint の検出方法 .....	596
Microsoft SharePoint Topology ジョブ .....	597
Microsoft SharePoint ディスカバリ・コマンド .....	602
トラブルシューティングおよび制限事項 .....	606
第 39 章: SAP ディスカバリの概要 .....	608
ディスカバリ・メカニズム .....	608
アプリケーション署名とプラグイン .....	609
第 40 章: SAP ABAP ディスカバリ .....	611
概要 .....	612
サポートされるバージョン .....	612
トポロジ .....	613
SAP ABAP の検出方法 .....	613
SAP Solution Manager Topology by SAP JCO ジョブ .....	618
SAP Solution Manager by SAP JCO ジョブ .....	620
SAP Applications by SAP JCO ジョブ .....	622
SAP ABAP Topology by SAP JCO ジョブ .....	624
SAP ABAP Connection by SAP JCO ジョブ .....	627
SAP ITS by NTCMD or UDA ジョブ .....	629
SAP System by Shell ジョブ .....	631
SAP TCP Ports ジョブ .....	632
アダプタ情報 .....	635
トラブルシューティングおよび制限事項 .....	636

第41章: SAP Java ディスカバリ .....	638
概要 .....	639
サポートされるバージョン .....	639
トポロジ .....	640
SAP Java 関連トポロジ全体の検出方法 .....	640
SAP Java Topology by HTTP ジョブ .....	642
SAP Java Topology by SAP JMX ジョブ .....	644
SAP Java Topology by WebServices ジョブ .....	647
SAP J2EE Topology by HTTP アダプタ .....	649
SAP J2EE Topology by JMX アダプタ .....	651
トラブルシューティングおよび制限事項 .....	654
第42章: SAP Solution Manager ディスカバリ .....	656
概要 .....	657
サポートされるバージョン .....	657
トポロジ .....	657
SAP Solution Manager の検出方法 .....	657
トラブルシューティングおよび制限事項 .....	659
第43章: Siebel ディスカバリ .....	660
概要 .....	661
サポートされるバージョン .....	661
トポロジ .....	662
Siebel トポロジ・ビュー .....	662
Siebel Web トポロジ・ビュー .....	663
Siebel トポロジの検出方法 .....	664
Siebel Application Server Configuration ジョブ .....	666
Siebel Application Servers ジョブ .....	668
Siebel Gateway Connection ジョブ .....	669
Siebel Web Applications by NTCMD or UDA ジョブ .....	671
Siebel Web Applications by TTY ジョブ .....	672
Siebel DB by NTCMD or UDA ジョブ .....	675
Siebel DB by TTY ジョブ .....	676
トラブルシューティングおよび制限事項 .....	678
第7部: Hosts and Resources .....	679
第44章: アプリケーション署名 .....	680
概要 .....	681
メソッドの機能 .....	681
署名リポジトリ .....	682
入力データ .....	682

トポロジ .....	683
アプリケーション署名のメカニズム .....	685
ジョブおよびアダプタ .....	686
ジョブおよび標準設定の動作 .....	686
Host Resources ジョブ・フロー .....	687
アプリケーション署名のフロー .....	688
アプリケーション署名の構成ファイル .....	692
構成ファイルの構造 .....	692
要素および属性 .....	693
プラグイン .....	698
第45章: Host Resources and Applications by PowerShell ディスカバリ .....	707
概要 .....	708
PowerShellによるホストのリソースおよびアプリケーションの検出方法 .....	708
Host Resources by PowerShell ジョブとHost Applications by PowerShell ジョブ .....	709
第46章: ホストのリソースおよびアプリケーションのディスカバリ .....	711
概要 .....	712
トポロジ .....	712
ホストのリソースおよびアプリケーションの検出方法 .....	713
インストール済みソフトウェアの検出方法を以前のメカニズムに戻す方法 .....	714
ホストのリソースおよびアプリケーションのディスカバリ .....	715
トラブルシューティングおよび制限事項 .....	721
第47章: IBM i by Eview ディスカバリ .....	723
概要 .....	724
ディスカバリの領域 .....	724
サポートされるバージョン .....	725
トポロジ .....	725
IBM i Resources .....	725
IBM i Objects .....	726
ディスカバリ・メカニズム .....	727
iSeries の検出方法 .....	727
IBM i Connection ジョブ .....	727
IBM i Objects ジョブ .....	728
IBM i Resources ジョブ .....	731
第48章: インベントリ・ディスカバリ .....	734
第49章: NetApp Filer ディスカバリ .....	735
概要 .....	736
サポートされるバージョン .....	736
トポロジ .....	736

NetApp Filer の検出方法 .....	738
NetApp Filer by WebServices ジョブ .....	741
第 50 章: SMI-S ディスカバリ .....	744
概要 .....	745
サポートされるバージョン .....	745
ディスカバリ・メカニズム .....	745
SMI-S の検出方法 .....	745
Storage Devices Connection by CIM ジョブ .....	746
Storage Devices Connection by CIM アダプタ .....	747
Storage Devices Topology by CIM ジョブ .....	749
Storage Devices Topology by CIM アダプタ .....	751
第 8 部: Mainframe .....	754
第 51 章: EView Agent ディスカバリ .....	755
概要 .....	756
サポートされるバージョン .....	756
トポロジ .....	757
EView Connection .....	757
LPAR Resources by EView .....	758
CICS by EView .....	758
DB2 by EView .....	759
IMS by EView .....	760
MQ by EView .....	761
EView によるメインフレームの検出方法 .....	762
ディスカバリ・メカニズム .....	763
LPAR Resources by EView ジョブ .....	764
EView Connection ジョブ .....	765
CICS by EView ジョブ .....	766
DB2 by EView ジョブ .....	767
IMS by EView ジョブ .....	768
MQ by EView ジョブ .....	769
トラブルシューティングおよび制限事項 .....	769
第 9 部: Middleware > Java EE Application Servers .....	772
第 52 章: Apache Tomcat ディスカバリ .....	773
概要 .....	774
サポートされるバージョン .....	775
トポロジ .....	776
Apache Tomcat の検出方法 .....	777

Bugzilla, Wordpress, MediaWiki の検出方法 .....	778
Apache Tomcat by Shell ジョブ .....	779
アダプタ情報 .....	779
第 53 章: GlassFish ディスカバリ .....	783
概要 .....	784
サポートされるバージョン .....	784
GlassFish Topology by Shell の検出方法 .....	784
JEE Glassfish by Shell ジョブ .....	785
Glassfish_By_Shell アダプタ .....	786
トラブルシューティングおよび制限事項 .....	789
第 54 章: JBoss ディスカバリ .....	790
概要 .....	791
サポートされるバージョン .....	791
JEE JBoss by JMX の検出方法 .....	791
JEE JBoss by Shell の検出方法 .....	796
JEE TCP Ports ジョブ .....	797
アダプタ情報 .....	799
JEE JBoss Connections by JMX ジョブ .....	802
アダプタ情報 .....	803
JEE JBoss by JMX ジョブ .....	806
JEE JBoss by Shell ジョブ .....	809
アダプタ情報 .....	810
トラブルシューティングおよび制限事項 .....	813
第 55 章: WebLogic ディスカバリ .....	814
概要 .....	815
サポートされるバージョン .....	815
WebLogic Topology by JMX の検出方法 .....	815
WebLogic Topology by Shell の検出方法 .....	817
JEE TCP Ports ジョブ .....	818
アダプタ情報 .....	820
JEE Weblogic Connections by JMX ジョブ .....	822
JEE Weblogic by JMX ジョブ .....	825
JEE Weblogic by Shell ジョブ .....	830
アダプタ情報 .....	830
トラブルシューティングおよび制限事項 .....	833
第 56 章: WebSphere ディスカバリ .....	835
概要 .....	836
サポートされるバージョン .....	836
WebSphere Topology by JMX の検出方法 .....	837

WebSphere Topology by Shell の検出方法 .....	840
JEE TCP Ports ジョブ .....	842
アダプタ情報 .....	844
JEE WebSphere Connections by JMX ジョブ .....	846
JEE WebSphere by Shell or JMX ジョブ .....	850
アダプタ情報 .....	850
JEE WebSphere by Shell ジョブ .....	855
アダプタ情報 .....	855
トラブルシューティングおよび制限事項 .....	861
 第 10部: Middleware > Messaging Servers .....	 862
第 57章: Microsoft MQ (メッセージ・キュー) ディスカバリ .....	863
サポートされるバージョン .....	864
Microsoft MQ の検出方法 .....	864
Microsoft Message Queue Topology by NTCMD or UDA ジョブ .....	866
Microsoft Message Queue Topology by LDAP ジョブ .....	869
Microsoft MQ ディスカバリ・スクリプト .....	870
Microsoft MQ トポロジのディスカバリ方法 .....	871
Host Applications by Shell ジョブ .....	871
Microsoft Message Queue Topology by NTCMD or UDA ジョブ .....	874
Microsoft Message Queue Topology by LDAP ジョブ .....	880
第 58章: TIBCO BusinessWorks および EMS のディスカバリ .....	881
概要 .....	882
ディスカバリ・メカニズム .....	882
サポートされるバージョン .....	882
トポロジ .....	883
TIBCO BusinessWorks および EMS の検出方法 .....	884
TIBCO BusinessWorks by Shell ジョブ .....	886
TIBCO EMS by Shell ジョブ .....	888
第 59章: WebSphere MQ ディスカバリ .....	893
概要 .....	894
サポートされるバージョン .....	894
トポロジ .....	894
MQ Queue Dependency .....	895
MQ Q Manager Resources on Non-Local Cluster .....	896
MQ Namelist Membership .....	897
MQ Cluster Membership .....	897
MQ Channel Communication .....	898
MQ Alias Queue Managers .....	898

MQ Topology .....	899
WebSphere MQ の検出方法 .....	901
ディスカバリ・メカニズム .....	902
アダプタ .....	903
アダプタ・パラメータ .....	903
検出された CIT .....	904
関係 .....	906
トラブルシューティングおよび制限事項 .....	910
<b>第 11 部: Middleware &gt; Web Servers .....</b>	<b>911</b>
第 60 章: 基本 Web サーバ・ディスカバリ .....	912
概要 .....	913
サポートされるバージョン .....	913
基本 Web サーバの検出方法 .....	913
Web Server by Shell ジョブ .....	914
はじめに .....	914
トポロジ・マップ .....	915
サポートされているポリシー .....	915
アダプタ情報 .....	915
Web Server Detection using TCP Ports ジョブ .....	918
はじめに .....	918
トポロジ・マップ .....	919
サポートされているポリシー .....	919
アダプタ情報 .....	920
WebSphere to Web Server Dependency ジョブ .....	921
はじめに .....	921
トポロジ・マップ .....	923
サポートされているポリシー .....	923
アダプタ情報 .....	923
第 61 章: Microsoft Internet Information Services (IIS) ディスカバリ .....	926
サポートされるバージョン .....	927
Microsoft Internet Information Services (IIS) ディスカバリのトポロジ .....	927
Microsoft Internet Information Services (IIS) のトポロジの検出方法 .....	928
IIS Applications by NTCMD or UDA ジョブ .....	930
Bugzilla, Wordpress, MediaWiki のディスカバリ .....	933
トラブルシューティングおよび制限事項 .....	934
<b>第 12 部: Middleware &gt; Web Services .....</b>	<b>935</b>
第 62 章: UDDI レジストリ・ディスカバリ .....	936

概要 .....	937
サポートされるバージョン .....	937
トポロジ .....	937
UDDI プロセスの検出方法 .....	938
<b>第 13部: Middleware &gt; Proxy Servers .....</b>	<b>939</b>
第 63章: IBM Security Access Manager ディスカバリ .....	940
概要 .....	941
サポートされるバージョン .....	941
WebSeal Connection by Web Services ジョブ .....	941
WebSeal Connection By Shell ジョブ .....	944
WebSeal Topology by Web Services ジョブ .....	946
WebSeal Topology by Shell ジョブ .....	949
HTTP を使用して IBM Security Access Manager を検出する方法 .....	951
シェルを使用して IBM Security Access Manager を検出する方法 .....	952
<b>第 14部: Middleware &gt; Security Servers .....</b>	<b>954</b>
第 64章: Oracle Access Management ディスカバリ .....	955
Oracle Access Management Connection by Web Services ジョブ .....	956
はじめに .....	956
トポロジ・マップ .....	956
サポート・ポリシー .....	956
アダプタ情報 .....	956
Oracle Access Management Policies by Web Services ジョブ .....	958
はじめに .....	958
トポロジ・マップ .....	958
サポート・ポリシー .....	959
アダプタ情報 .....	959
Oracle Access Management Dependencies via URL ジョブ .....	960
はじめに .....	960
トポロジ・マップ .....	961
サポート・ポリシー .....	962
アダプタ情報 .....	962
<b>第 15部: Network Infrastructure .....</b>	<b>964</b>
第 65章: ネットワーク - 基本 ディスカバリ .....	965
概要 .....	967
Host Connection by Shell の検出方法 .....	969
Host Connection by SNMP の検出方法 .....	970



Host Connection by WMI の検出方法 .....	972
Client Connection by SNMP の検出方法 .....	972
Host Connection by Shell ジョブ .....	973
ディスカバリ・メカニズム .....	973
Windows のプロセス .....	974
UNIX ベースのプロセス .....	976
Nexus のプロセス .....	981
トリガ・クエリ .....	981
ジョブ・パラメータ .....	982
アダプタ .....	983
検出された CIT .....	983
トラブルシューティングおよび制限事項 .....	985
Host Connection by SNMP ジョブ .....	987
ディスカバリ・メカニズム .....	987
トリガ・クエリ .....	988
ジョブ・パラメータ .....	989
アダプタ .....	990
検出された CIT .....	990
トラブルシューティングおよび制限事項 .....	991
Host Connection by WMI ジョブ .....	993
ディスカバリ・メカニズム .....	993
トリガ・クエリ .....	995
ジョブ・パラメータ .....	996
アダプタ .....	996
検出された CIT .....	997
トラブルシューティングおよび制限事項 .....	997
Client Connection by SNMP ジョブ .....	999
ディスカバリ・メカニズム .....	999
トリガ CI .....	1000
ジョブ・パラメータ .....	1000
トリガ CI データ .....	1001
検出された CIT .....	1001
第 66 章: DNS ゾーン・ディスカバリ .....	1003
概要 .....	1004
サポートされるバージョン .....	1005
nslookup による DNS ゾーン の検出方法 .....	1006
DNS による DNS ゾーン の検出方法 .....	1007
DNS サーバで nslookup を使用してシェルでホストを検出する方法 .....	1007
DNS Zone by nslookup ジョブ .....	1008

アダプタ情報 .....	1009
DNS Zone by DNS ジョブ .....	1011
アダプタ情報 .....	1011
Hosts by Shell using nslookup on DNS Server ジョブ .....	1014
NSLOOKUP on DNS Server アダプタ .....	1015
ディスカバリ・メカニズム - Windows .....	1018
ディスカバリ・メカニズム - UNIX 系 .....	1020
用語集 .....	1021
第 67 章: ファイアウォール .....	1022
Firewall Topology by SNMP ジョブ .....	1022
はじめに .....	1022
トポロジ・マップ .....	1023
アダプタ情報 .....	1023
第 68 章: AS400 Host ディスカバリ .....	1028
概要 .....	1029
サポートされるバージョン .....	1029
トポロジ .....	1030
AS400 ホストの検出方法 .....	1031
Host Connection to AS400 ジョブ .....	1032
第 69 章: Host Connection by PowerShell ディスカバリ .....	1034
概要 .....	1035
サポートされるバージョン .....	1035
PowerShell によるホスト接続の検出方法 .....	1036
Host Connection by PowerShell ジョブ .....	1037
トラブルシューティングおよび制限事項 .....	1047
第 70 章: Layer2 ディスカバリ .....	1048
概要 .....	1049
サポートされるデバイス .....	1050
Layer2 オブジェクトの検出方法 .....	1051
保存されたファイルを使用して Layer2 接続を検出する方法 .....	1054
CDP または LLDP MIB を使用して Layer2 接続を検出する方法 .....	1056
シェルによる Layer2 トポロジの検出方法 .....	1056
Layer2 Topology Bridge-based by SNMP ジョブ .....	1057
Layer2 Topology by Shell ジョブ .....	1058
Layer2 Topology CDP-LLDP based by SNMP ジョブ .....	1060
Layer2 Topology VLAN-based by SNMP ジョブ .....	1060
Process Layer2 Saved Files ジョブ .....	1061
Merge VLANs by Ports ジョブ .....	1063
VLAN ports by SNMP ジョブ .....	1065

VLANs by SNMP ジョブ .....	1066
L2 Bridge by SNMP アダプタ .....	1068
Layer2 Topology by Shell アダプタ .....	1069
CDP/LLDP Neighbors Layer 2 Devices by SNMP アダプタ .....	1072
L2 Vlan by SNMP アダプタ .....	1073
Merge VLANs アダプタ .....	1075
Process Layer2 Collected Files アダプタ .....	1078
VMS Catalyst by SNMP アダプタ .....	1079
Catalyst Vlans by SNMP アダプタ .....	1080
関係 .....	1082
トラブルシューティングおよび制限事項 .....	1083
第71章: No-Credentials ディスカバリ .....	1084
概要 .....	1085
Data Flow Probe マシンの設定方法 .....	1086
Host Fingerprint with nmap の検出方法 .....	1087
Range IPs by nmap の検出方法 .....	1087
Host Fingerprint using nmap ジョブ .....	1088
Range IPs by nmap ジョブ .....	1090
トラブルシューティングおよび制限事項 .....	1092
第72章: ネットワーク接続のアクティブおよびパッシブ・ディスカバリ .....	1093
概要 .....	1094
サポートされるバージョン .....	1094
トポロジ .....	1095
プロセスの検出方法 .....	1096
TCPトラフィック・ジョブ .....	1097
Network Connectivity Data Analyzer ジョブ .....	1098
TcpDiscoveryDescriptor.xml ファイル .....	1101
サーバ検出方法 .....	1101
フィルタリング .....	1102
レポート .....	1103
第16部: Tools and Samples > Discovery Samples .....	1112
第73章: Dynamic Credential Sample ジョブ .....	1113
はじめに .....	1113
アダプタ情報 .....	1113
第74章: Config Parser アダプタ .....	1114
第75章: Execute Command アダプタ .....	1116
第17部: Tools and Samples > Discovery Tools .....	1118

第 76 章: File Monitor by Shell ジョブ .....	1119
アダプタ情報 .....	1119
トラブルシューティングおよび制限事項 .....	1121
第 77 章: HP ServiceGuard and Oracle RAC ディスカバリ .....	1122
概要 .....	1123
サポートされるバージョン .....	1123
Link DB DataFiles and Clustered FS ジョブの実行方法 .....	1123
Link DB Datafiles With Clustered FileSystem アダプタ .....	1124
Link DB DataFiles and Clustered FS ジョブ .....	1126
第 78 章: クラスタ化ソフトウェアの統合 .....	1127
概要 .....	1128
サポートされるソフトウェア .....	1128
クラスタ化ソフトウェアの統合方法 .....	1128
Merge Clustered Software ジョブ .....	1129
第 79 章: TCP/UDP Ports Discovery by Nmap .....	1132
概要 .....	1133
サポートされるバージョン .....	1133
TCP/UDP Ports by Nmap の検出方法 .....	1133
TCP Ports ジョブ .....	1134
アダプタ情報 .....	1137
第 18 部: Tools and Samples > SSL Certificate Discovery .....	1139
第 80 章: SSL Certificate ディスカバリ .....	1140
概要 .....	1141
サポートされるバージョン .....	1141
トポロジ .....	1141
ディスカバリ・メカニズム .....	1141
SSL Certificates の検出方法 .....	1142
SSL Certificate Discovery by HTTPS ジョブ .....	1143
SSL Certificate Discovery アダプタ .....	1145
追加情報 .....	1146
トラブルシューティングおよび制限事項 .....	1146
第 19 部: Tools and Samples > UD Agent Management .....	1147
第 81 章: Install UD Agent ジョブ .....	1148
アダプタ情報 .....	1148
第 82 章: Migrate DDMI Agent ジョブ .....	1151
アダプタ情報 .....	1151

第 83 章: UDA Status Collector ジョブ .....	1154
アダプタ情報 .....	1154
第 84 章: Uninstall UD Agent ジョブ .....	1156
アダプタ情報 .....	1156
第 85 章: Update UD Agent ジョブ .....	1159
アダプタ情報 .....	1159
ドキュメント・フィードバックの送信 .....	1163



## 第1部: Cloud and Virtualization > Cloud

# 第1章: Amazon Web Services ディスカバリ

## 本章の内容

- 概要 ..... 29
- トポロジ ..... 30
  - Amazon EC2 ..... 30
  - Amazon RDS ..... 31
- EC2 サービスおよび RDS サービスの検出方法 ..... 32
- AWS\_by\_WebServices アダプタ ..... 34
- AWS by Web Services ジョブ ..... 36
- トラブルシューティングおよび制限事項 ..... 37



## 概要

Amazon Web Services(AWS)は、リモート・コンピューティング・サービス(Web サービスとも呼ばれる)のコレクションであり、これらが連動してクラウド・コンピューティング・プラットフォームを形成し、Amazon.com によるインターネット全体をサポートします。

Amazon Web Services が提供するサービスは、REST(Representational State Transfer)プロトコルおよび SOAP プロトコルを介して HTTP によってアクセスされます。

AWS デイスカバリは、Amazon テクノロジーに基づいてクラウドの状態と構成を提示します。次に示す、下位レベルのインフラストラクチャ・サービスのデイスカバリがサポートされます。

### • EC2(コンピューティング)

Amazon EC2(Amazon Elastic Compute Cloud)は、クラウド内でサイズ変更が可能なコンピューティング・キャパシティを提供する。ホスト対象アプリケーションに必要なオペレーティング・システム、サービス、データベース、アプリケーション・プラットフォーム・スタックを使用して、仮想 Amazon EC2 環境を定義します。Amazon EC2 には、コンピューティング・リソースを管理するための機能完備の管理コンソールとAPI が用意されています。

### • RDS(リレーショナル・データベース)

Amazon RDS(Amazon Relational Database Service)は、クラウド内で MySQL や Oracle の配備のための機能を提供する Web サービスであり、バックアップ、スケーリング、パッチの適用などの時間がかかるタスクの管理も請け負います。

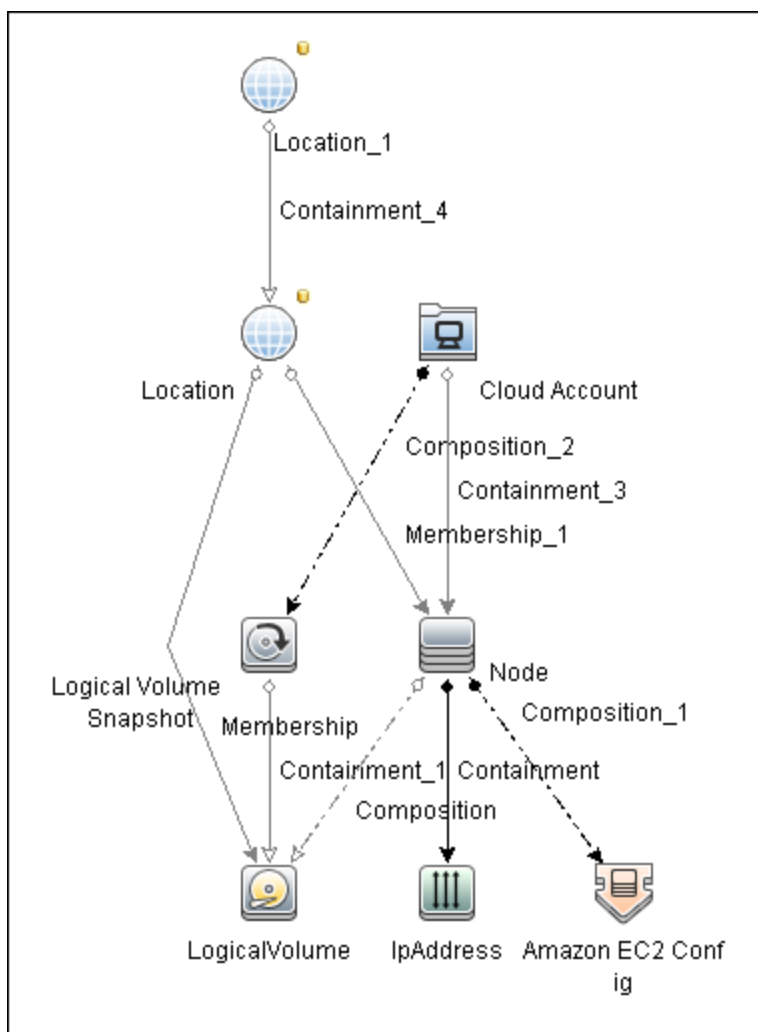
AWS との通信では、デイスカバリは Amazon SDK および IAM サービスを認証に使用します。

## トポロジ

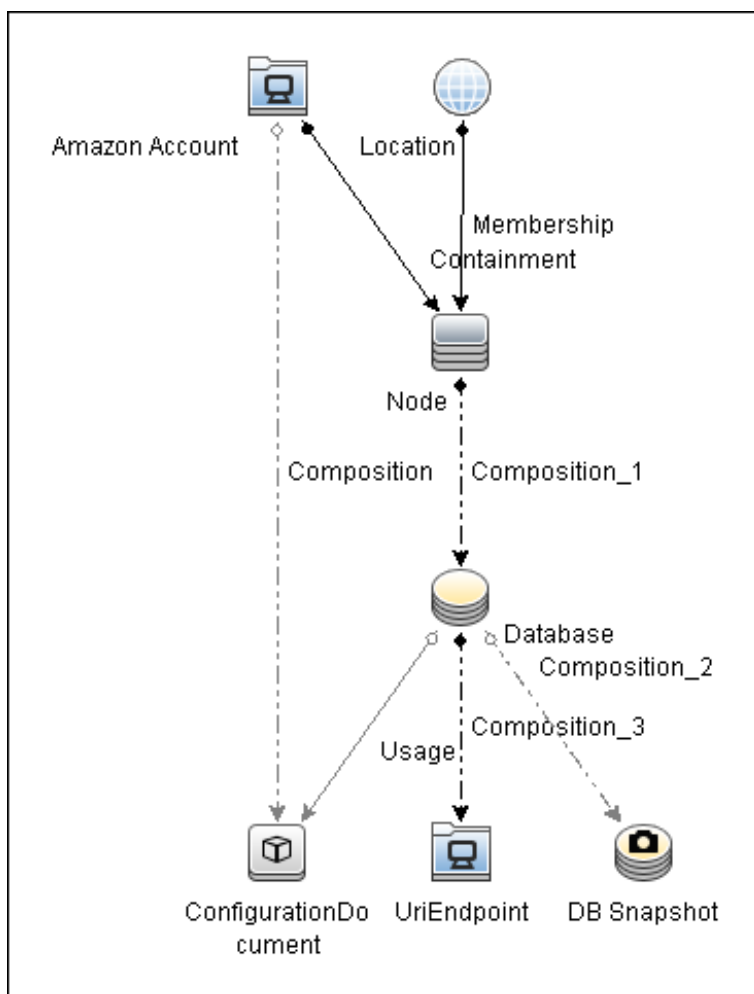
次の図に、AWS ディスカバリのトポロジを示します。

注: 検出された CIT のリストについては、[「検出された CIT」\(35ページ\)](#)を参照してください。

### Amazon EC2



## Amazon RDS



## EC2 サービスおよび RDS サービスの検出方法

このタスクでは、これら2つの下位レベルのAWSサービスを**AWS プロトコル**というデイスカバリ・プロトコルを使用して検出する方法を説明します。このデイスカバリ・プロセスによって、実行中のノード・インスタンスとそれらの構成(AMIに関する情報を含む)、対応するブロック・ストレージ、領域やゾーンに関する情報を持つスナップショットに関する情報を検出できます。デイスカバリ・ユーザが登録されているAmazon アカウントのスコープ内のすべてのトポロジがレポートされます。

このタスクには次の手順が含まれます。

- [「前提条件 - プローブ IP アドレス」\(32ページ\)](#)
- [「前提条件 - 資格情報」\(32ページ\)](#)
- [「前提条件 - ドライバの設定」\(32ページ\)](#)
- [「デイスカバリの実行」\(33ページ\)](#)

### 1. 前提条件 - プローブ IP アドレス

デイスカバリには、トリガ対象範囲内の1つ以上のIPアドレスを持つプローブが必要です。

### 2. 前提条件 - 資格情報

AWS デイスカバリでは、AWS サービスへの認証要求に使用される3種類のアクセス資格情報の1つである**アクセス・キー**が使用されます。

UCMDBにおいてAWS 資格情報を指定するには、次を定義する必要があります：**AWS プロトコル**。

資格情報の値	AWS プロトコル名
Access Key ID	ユーザ名
Secret Access Key	パスワード

アクセス・キーの詳細については、[ここ](#)を参照してください。

### 3. 前提条件 - ドライバの設定

注: この手順は、AWS デイスカバリを実行する各プローブに対して実行する必要があります。

- a. Amazon SDK for Java を <http://aws.amazon.com/sdkforjava/> からダウンロードします。  
必要なバージョンは 1.9.32(以下で \${VERSION} と記載)以降です。
- b. この ZIP ファイルを一時フォルダに展開します(例 : \${AWS\_TEMP\_DIR})。
- c. **\${PROBE\_ROOT\_DIR}/content/lib/aws/** フォルダ(**\${AWS\_PROBE\_DIR}** として参照)を作成します。
- d. 次のサードパーティ製ライブラリの JAR ファイルと SDK を **\${AWS\_PROBE\_DIR}** にコピーします。

**\${AWS\_TEMP\_DIR}/lib/aws-java-sdk-\${VERSION}.jar**

**\${AWS\_TEMP\_DIR}/third-party/jackson-\*/.jar**

**\${AWS\_TEMP\_DIR}/third-party/httpcomponents-client-\*/.jar**

**\${AWS\_TEMP\_DIR}/third-party/joda-time-\*/.jar**

ファイル構造の例は次のようになります。

**C:\hp\UCMDB\DataFlowProbe\content\lib\aws :**

- aws-java-sdk-1.9.32.jar
- httpclient-4.3.jar
- httpcore-4.3.jar
- jackson-annotations-2.3.0.jar
- jackson-core-2.3.2.jar
- jackson-databind-2.3.2.jar
- joda-time-2.2.jar

#### 4. ディスカバリの実行

**AWS by Web Services** ジョブを実行します。

## AWS\_by\_WebServices アダプタ

本項の内容

- [「入力 CIT」\(34ページ\)](#)
- [「トリガ CI データ」\(34ページ\)](#)
- [「使用スクリプト」\(34ページ\)](#)
- [「検出された CIT」\(35ページ\)](#)

### 入力 CIT

Discovery Probe Gateway

### トリガ CI データ

名前	値
probeName	\${SOURCE.name}

### 使用スクリプト

- AWS\_by\_WebServices.py
- aws.py
- aws\_rds.py
- aws\_store.py
- entity.py
- db\_platform.py
- db\_builder.py

- db.py
- ec2.py
- iteratortools.py

## 検出された CIT

- Amazon Account
- Amazon EC2 Config
- Composition
- ConfigurationDocument
- Containment
- Database
- DB Snapshot
- IpAddress
- Location
- LogicalVolume
- Logical Volume Snapshot
- Membership
- Node
- UriEndpoint
- Usage

注: トポロジを表示するには, [「トポロジ」\(30ページ\)](#)を参照してください。

## AWS by Web Services ジョブ

本項の内容

- [「アダプタ」\(36ページ\)](#)
- [「トリガ・クエリ」\(36ページ\)](#)
- [「ディスカバリ・フロー」\(36ページ\)](#)

### アダプタ

このジョブでは, **AWS\_by\_WebServices** アダプタを使用します。

### トリガ・クエリ



### ディスカバリ・フロー

AWS の検出では, トリガの対象となる IP アドレスはないため, このジョブは, 該当範囲内の1つ以上の IP アドレスがあるプローブに対して開始されます(これは UCMDDB ワークフロー要件です)。

UCMDDB は, サービスを検索する前に, ディスカバリ・ユーザが所属するアカウントに関する情報を取得する必要があります。これは, IAM サービスを使用することで実行されます。ユーザは ARN(Amazon Resource Name)を持っており, ここにアカウント ID が格納されています。

### EC2 Service ディスカバリ



- リージョンおよびアベイラビリティゾーンの取得
- **実行中のインスタンスの取得**(この情報がないと、すべてのEBS ディスカバリが失敗)
- EBSに関する詳細情報の取得(実行中の各インスタンスに対するマッピングされたデバイスとして使用)
- マッピングされたEBSのみに関するEBS スナップショット情報の取得
- 実行中の各インスタンスのAMIの取得(AMIが見つからないと、対応するインスタンスはUCMDBにレポートされない)
- 各インスタンスに関するElastic IP情報の取得
- 各サービスの検出後、データは即時にUCMDBにレポート

### RDS Service ディスカバリ

- データベース・インスタンスの取得(この情報がないと、すべてのRDS ディスカバリが失敗)
- 各データベース・インスタンス・サーバの情報を補足するための、使用可能なすべてのエンジンの取得
- データベース・インスタンス内で入手できる情報を補足するための、セキュリティ・グループおよびパラメータ・グループの取得
- データベース・スナップショットの取得
- 各サービスの検出後、データは即時にUCMDBにレポート

## トラブルシューティングおよび制限事項

本項では、Amazon Web Services ディスカバリのトラブルシューティングと制限事項について説明します。

**問題:** 次のエラー・メッセージがディスカバリ中に発生します。

- `at com.sun.proxy.$Proxy49.getUser(Unknown Source)`
- `Request timestamp is too skewed.Timestamps must be within 900 seconds of server time.Timestamp date:2015-06-30T04:17:39.196Z`

**解決策:** プローブの時間とタイム・ゾーンを標準ローカル時間に変更します。

## 第2章: VMware vCloud ディスカバリ

### 本章の内容

概要 .....	39
サポートされるバージョン .....	39
トポロジ .....	40
vCloud Director による vCloud の検出方法 .....	41
URL を使用した vCloud の検出方法 .....	42
プローブへの vCloud SDK の依存関係の追加方法 .....	42
vCloud_Director_by_vCloud_API アダプタ .....	43
vCloud_Director_URL_by_vCloud_API アダプタ .....	45
vCloud Director by vCloud API ジョブ .....	47
vCloud Director URL by vCloud API ジョブ .....	48
トラブルシューティングおよび制限事項 .....	49

## 概要

VMware vCloud Director は、複数のクラスタからの IT リソースをグループ化することで、ポリシーに基づいた仮想データセンタを作成します。

vCloud ディスカバリのプロセスにより、組織、カタログ、仮想データセンタ、vApp(仮想マシン、vApp テンプレート、およびメディアを含む)が含まれる vCloud トポロジを検出できます。

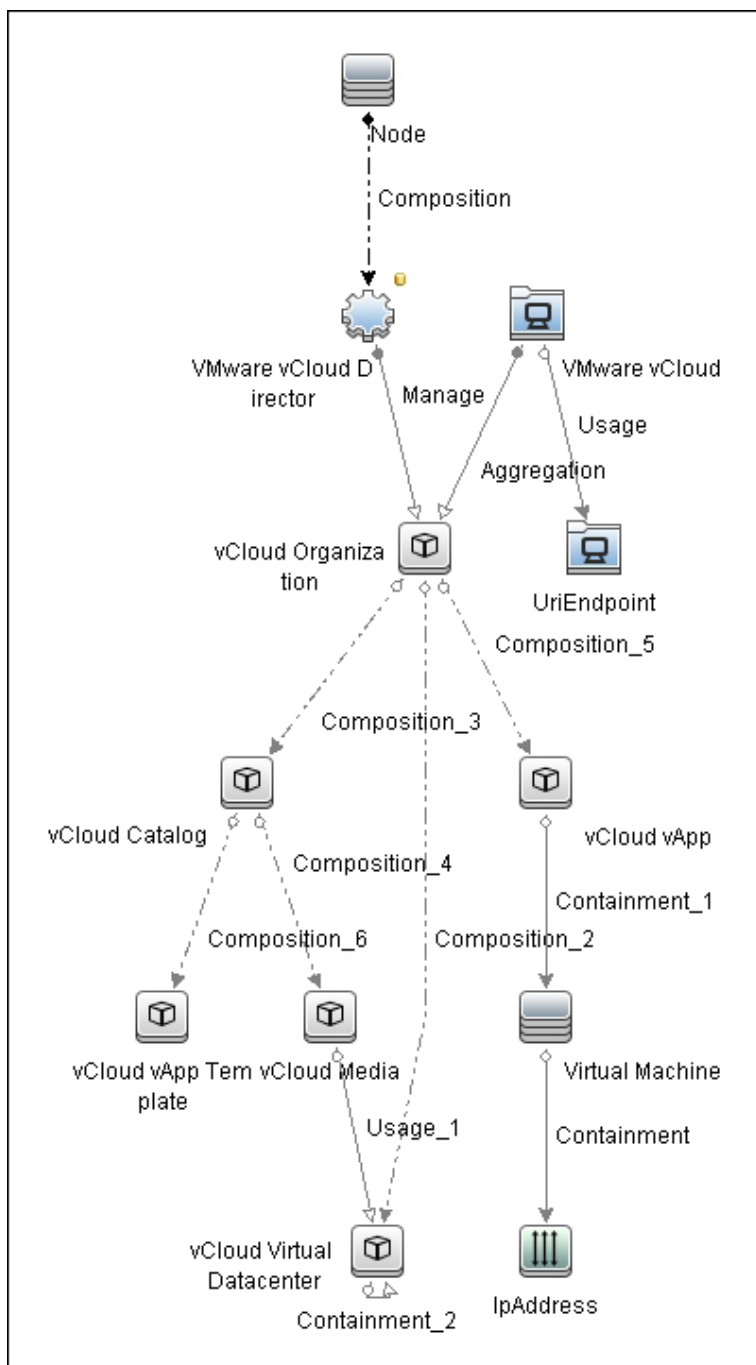
## サポートされるバージョン

VMware vCloud ディスカバリは、VMware vCloud Director バージョン 1.5～5.1.2 をサポートしています。

## トポロジ

次の図に、vCloud ディスカバリのトポロジを示します。

注: 検出された CIT のリストについては、「[検出された CIT](#)」(44ページ)を参照してください。



## vCloud Director による vCloud の検出方法

本項では、vCloud Director アプリケーションを検出することで vCloud トポロジを検出する方法について説明します。

本項の内容

- [「前提条件」\(41ページ\)](#)
- [「ジョブの実行」\(41ページ\)](#)

### 1. 前提条件

- a. vCloud Director アプリケーションが動作しているホストへのシェル接続が必要です。
- b. プローブに vCloud SDK jar ファイルがあることが必要です。[「プローブへの vCloud SDK の依存関係の追加方法」\(42ページ\)](#)を参照してください。
- c. 次の資格情報を定義します。

- **SSH または Telnet**
- **vCloud**

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ジョブの実行

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by Shell** ジョブを実行し、ターゲット・ホストおよびそのホストへのシェル接続を検出します。
- c. **Host Applications by Shell** ジョブを実行してターゲット・ホストのアプリケーション(VMware vCloud Director アプリケーションを含む)を検出します。
- d. **vCloud Director by vCloud API** ジョブを実行して vCloud トポロジを検出します。

## URL を使用した vCloud の検出方法

本項では、vCloud Director の URL を使用した vCloud トポロジの検出方法について説明します。

本項の内容

- 「前提条件」(42ページ)
- 「ジョブの実行」(42ページ)

### 1. 前提条件

- a. プローブに vCloud SDK jar ファイルがあることが必要です。「[プローブへの vCloud SDK の依存関係の追加方法](#)」(42ページ)を参照してください。
- b. **vCloud** 資格情報を定義します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ジョブの実行

- a. **vCloud Director URL by vCloud API** ジョブを実行して vCloud トポロジを検出します。
  - i. **baseUrl** パラメータに、ターゲット vCloud Director の接続 URL を設定します。
  - ii. ジョブのアクティブ化後、ディスカバリを実行するプローブを入力 CI として手動で追加します。

## プローブへの vCloud SDK の依存関係の追加方法

vCloud SDK の依存関係をプローブに追加するには、次の手順を実行します。

1. VMware コミュニティ・サイトから VMware vCloud SDK アーカイブをダウンロードします。

<http://communities.vmware.com/community/vmttn/developer/forums/vcloudsdkjava>

推奨されるバージョンは 5.1 です。

2. 次の jar ファイルを **%PROBE\_ROOT%\content\lib** フォルダにコピーします。

- SDK-<バージョン>\rest-api-schemas-<バージョン>.jar
- SDK-<バージョン>\vcloud-java-sdk-<バージョン>.jar

3. プローブを再起動します。

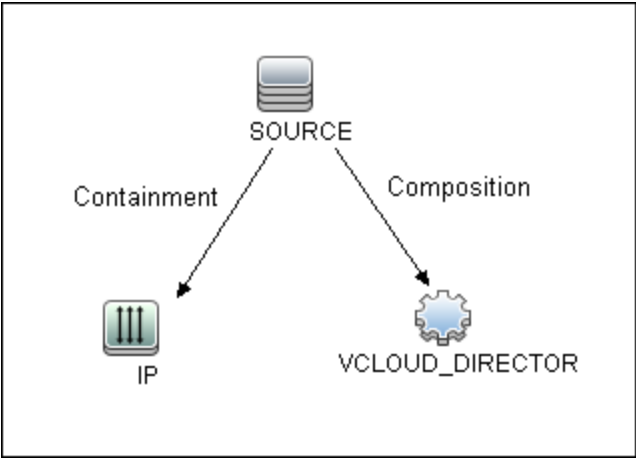
## vCloud\_Director\_by\_vCloud\_API アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Node

### 入力クエリ



ノード名	条件
SOURCE	なし
IP	NOT IP Probe Name null
VCLLOUD_DIRECTOR	DiscoveredProductName 等しい VMware vCloud Director

トリガ CI データ

名前	値
ip_addresses	\${IP.name}
vCloudDirectorId	S{VCLOUD_DIRECTOR.root_id}

使用スクリプト

- vcloud.py
- vcloud\_director\_by\_vcloud\_api.py
- vcloud\_discover.py
- vcloud\_report.py

検出された CIT

- Aggregation
- Composition
- Containment
- Interface
- IpAddress
- Manage
- Node
- UriEndpoint
- Usage
- vCloud Catalog



- vCloud Media
- vCloud Organization
- vCloud vApp
- vCloud vApp Template
- vCloud Virtual Datacenter
- VMware vCloud
- VMware vCloud Director

注: トポロジを表示するには, 「[トポロジ](#)」(40ページ)を参照してください。

## パラメータ

名前	説明
reportPoweredOffVms	<p><b>True</b> に設定すると, パワー・オフ状態の仮想マシンがレポートされます。</p> <p><b>False</b> に設定すると, パワー・オフ状態の仮想マシンはレポートされません。</p> <p>標準設定 :False</p>

## vCloud\_Director\_URL\_by\_vCloud\_API アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Discovery Probe Gateway

### 使用スクリプト

- vcloud.py

- vcloud\_director\_url\_by\_vcloud\_api.py
- vcloud\_discover.py
- vcloud\_report.py

## 検出された CIT

- Aggregation
- Composition
- Containment
- Interface
- IPAddress
- Manage
- Node
- UriEndpoint
- Usage
- vCloud Catalog
- vCloud Media
- vCloud Organization
- vCloud vApp
- vCloud vApp Template
- vCloud Virtual Datacenter
- VMware vCloud
- VMware vCloud Director

注: トポロジを表示するには、[「トポロジ」\(40ページ\)](#)を参照してください。

パラメータ

名前	説明
baseUrl	ターゲット vCloud Director の接続 URL
reportPoweredOffVms	<p><b>True</b> に設定すると、パワー・オフ状態の仮想マシンがレポートされます。</p> <p><b>False</b> に設定すると、パワー・オフ状態の仮想マシンはレポートされません。</p> <p>標準設定 :False</p>

vCloud Director by vCloud API ジョブ

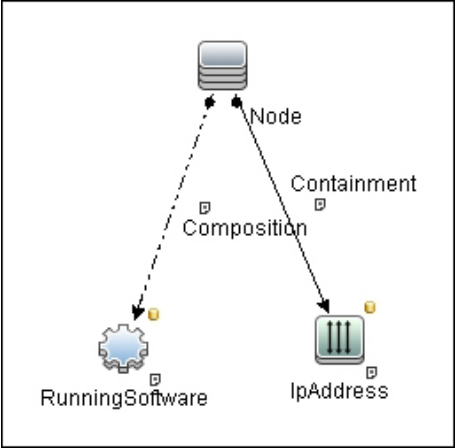
本項ではこのジョブに関する詳細を説明します。

アダプタ

このジョブでは、vCloud\_Director\_by\_vCloud\_API アダプタを使用します。

トリガ・クエリ

vcloud\_director\_on\_host\_with\_ip



ノード名	条件
Node	なし
IpAddress	NOT IP Probe Name null
RunningSoftware	DiscoveredProductName 等しい VMware vCloud Director

パラメータ

パラメータは、標準設定では上書きされず、アダプタからの値を使用します。

vCloud Director URL by vCloud API ジョブ

本項ではこのジョブに関する詳細を説明します。

• アダプタ

このジョブでは、vCloud\_Director\_URL\_by\_vCloud\_API アダプタを使用します。

• トリガ・クエリ

なし

• パラメータ

パラメータは、標準設定では上書きされず、アダプタからの値を使用します。

## トラブルシューティングおよび制限事項

仮想マシンがvAppの一部であり、利用可能なMACアドレスと接続済みのネットワークアダプタがない場合、その仮想マシンはレポートされません。

## 第2部: Cloud and Virtualization > Virtualization

## 第3章: Citrix Xen ディスカバリ

### 本章の内容

概要 .....	52
サポートされるバージョン .....	52
トポロジ .....	53
Citrix Xen トポロジの検出方法 .....	53
Citrix Xen Connection ジョブ .....	54
はじめに .....	54
アダプタ情報 .....	55
Citrix Xen Topology ジョブ .....	56
はじめに .....	56
アダプタ情報 .....	57

## 概要

Citrix XenServer は、仮想サーバ・インフラストラクチャの構築と管理を可能にするハイパーバイザ・プラットフォームです。Citrix Systems が開発したもので、Xen 仮想マシン・ハイパーバイザに基づきます。

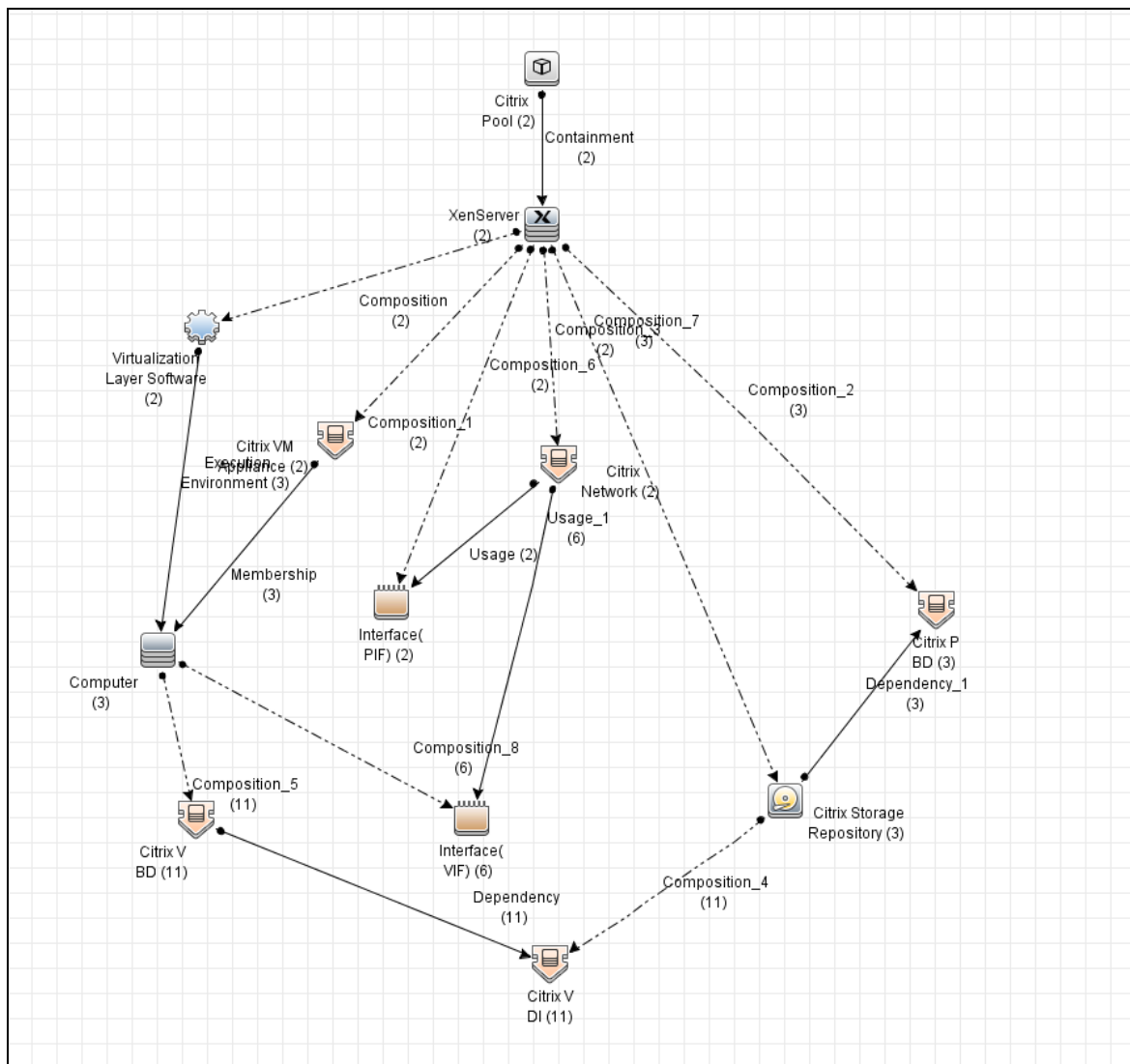
## サポートされるバージョン

Citrix Xen ディスカバリは次のバージョンの Citrix XenServer をサポートします: 6.2~6.5。



## トポロジ

次の図は、Citrix Xen ディスカバリのトポロジを示します。



## Citrix Xen トポロジの検出方法

本項では、Citrix Xen が管理するトポロジの検出方法について説明します。

前提条件

- 対象のXenServerがCitrix XenServer管理APIを開けていることを確認します。
- HTTPプロトコルを設定します。この情報の詳細については、『HP UCMDB Discovery and Integrations Content Guide - Supported Content』でHTTPプロトコル資格情報について説明する項を参照してください。

ディスカバリの実行

次のジョブを実行することによって、Citrix Xen Topology ディスカバリを実行します。

- Range IPs by ICMP(ターゲット IP を検出)
- Citrix Xen Connection(Citrix XenServerを検出)
- Citrix Xen Topology(Citrix XenServer のフル・トポロジを検出)

Citrix Xen Connection ジョブ

本項ではこのジョブに関する詳細を説明します。

はじめに

このジョブでは、HTTP XML-RPC プロトコルを介してトリガ IP アドレスに接続することによって、XenServerを検出します。

トリガ TQL



ノード名	条件
IpAddress	NOT IP Probe Name null

## アダプタ情報

このジョブでは, **Citrix Xen Connection** アダプタを使用します。

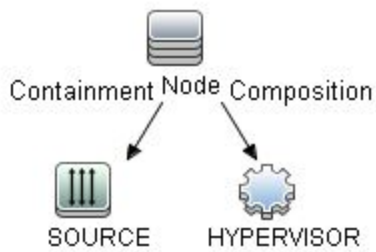
### アダプタ・タイプ

Jython

### 入力 CIT

IpAddress

### 入力 TQL



### トリガ CI データ

名前	値
credentialId	\${HYPERVISOR.credentials_id:}
ip_address	\${SOURCE.name}

### 使用スクリプト

- XenAPI.py
- citrix\_xen\_protocol.py
- citrix\_xen\_connection.py

## 検出された CIT

- Composition(unix, virtualization\_layer)
- Containment(unix, ip\_address)
- IpAddress
- Unix
- Virtualization Layer Software

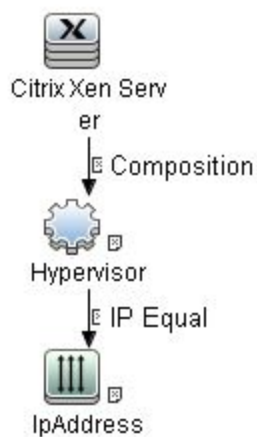
## Citrix Xen Topology ジョブ

本項ではこのジョブに関する詳細を説明します。

### はじめに

このジョブは、XenServer トポロジを検出します。

### トリガ TQL



ノード名	条件
Citrix XenServer	Node Operating System Installation type 等しい XenServer
Hypervisor	NOT Application IP null AND NOT Reference to the credentials dictionary entry null
IpAddress	NOT IP Probe Name null

## アダプタ情報

このジョブでは、**Citrix Xen Topology** アダプタを使用します。

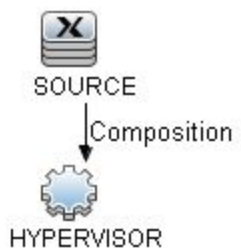
### アダプタ・タイプ

Jython

### 入力 CIT

Unix

### 入力 TQL



### トリガ CI データ

名前	値
credentialId	\${HYPERVISOR.credentials_id}
hypervisor_id	\${HYPERVISOR.root_id}
ip_address	\${HYPERVISOR.application_ip}

## 使用スクリプト

- XenAPI.py
- citrix\_xen\_models.py
- citrix\_xen\_protocol.py
- citrix\_xen\_topology.py

## 検出された CIT

- Citrix Network
- Citrix PBD
- Citrix Pool
- Citrix Storage Repository
- Citrix VBD
- Citrix VDI
- Citrix VM Application
- Composition(citrix\_storage\_repository, citrix\_vdi)
- Composition(host\_node, citrix\_vbd)
- Composition(host\_node, interface)
- Composition(unix, citrix\_network)
- Composition(unix, citrix\_pbd)
- Composition(unix, citrix\_pool)
- Composition(unix, citrix\_storage\_repository)
- Composition(unix, citrix\_vm\_appliance)
- Composition(unix, cpu)

- Composition(unix, interface)
- Composition(unix, virtualization\_layer)
- Computer
- Containment(citrix\_pool, unix)
- Containment(host\_node, ip\_address)
- Containment(interface, ip\_address)
- Cpu
- Dependency(citrix\_storage\_repository, citrix\_pbd)
- Dependency(citrix\_vbd, citrix\_vdi)
- ExecutionEnvironment(virtualization\_layer, host\_node)
- IpAddress
- Membership(citrix\_vm\_appliance, host\_node)
- Unix
- Usage(citrix\_network, interface)
- Virtualization Layer Software

# 第4章: HP IVM ディスカバリ

## 本章の内容

- 概要 ..... 61
- サポートされるバージョン ..... 61
- トポロジ ..... 61
- HP IVMトポロジの検出方法 ..... 62
- HP IVM by Shell ジョブ ..... 62
- HP Integrity Virtual Machine by Shell アダプタ ..... 63



IVM(Integrity Virtual Machines)は、HP-UX が稼働する任意の Itanium サーバ上で複数の仮想マシンを同時に実行できるようにする HP のソフトウェアです。HP IVM は、HP の Virtual Server Environment Suite に含まれます。

HP IVM ディスカバリは、HP IVM バージョン B.06.10.05 をサポートしています。

HP IVM トポロジは次に示すとおりです。



## HP IVM トポロジの検出方法

本項では、HP IVM が管理するトポロジの検出方法について説明します。

### 前提条件

- 実行中のIVM マシンにシェル接続できることを確認します。
- SSH プロトコル資格情報を設定します。この情報の詳細については、『HP UCMDB Discovery and Integrations Content Guide - Supported Content』で SSH プロトコル資格情報について説明する項を参照してください。

### HP IVM トポロジ・ディスカバリの実行

次のジョブを実行することによって、HP IVM Topology ディスカバリを実行します。

1. **Range IPs by ICMP**(ターゲット IP を検出)
2. **Host Connection by Shell**(ターゲット・ホストおよびそのホストへのシェル接続を検出)
3. **HP IVM by Shell**(HP IVM が管理する仮想化環境を検出)

## HP IVM by Shell ジョブ

このジョブは、HP IVM トポロジを検出します。

### アダプタ

**ID** : HP Integrity Virtual Machine by Shell

**表示名** : HP Integrity Virtual Machine by Shell

### トリガ TQL

CIT 名	条件
Computer	ExtendedOsFamily 等しい hp_ux

CIT 名	条件
IpAddress	Not Ip Probe Name null
Shell	NOT Reference to credentials dictionary entry null

## ディスカバリ・フロー

HP IVM by Shell ジョブのディスカバリ・フローは次のとおりです。

1. 実行中の仮想マシンの現在の状態を **hvvmstatus -V** コマンドを使用して取得します。
2. 仮想マシンの構成データを **hvvmstatus -d -P <仮想マシン名>** コマンドを使用して取得します。
3. バージョン関連情報を **hvvminfo -v** コマンドを使用して取得します。

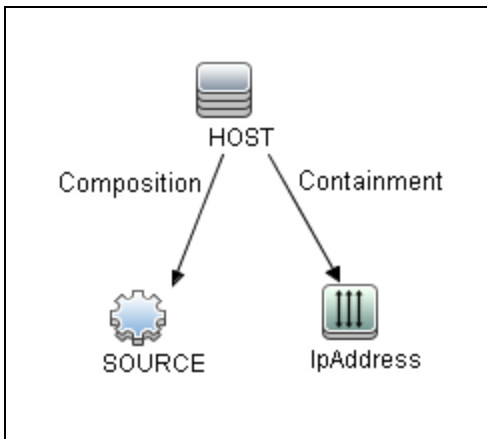
## HP Integrity Virtual Machine by Shell アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Shell

### 入力 TQL



CIT 名	条件
HOST	NOT CI Type 等しい nt

## トリガ CI データ

名前	値
Protocol	SOURCE.root_class
credentialsId	SOURCE.credentials_id
hostId	HOST.root_id
ip_address	SOURCE.application_ip

## 使用スクリプト

- ivm.py
- ivm\_by\_shell.py
- ivm\_discoverer.py

## 検出された CIT

- Composition
- ExecutionEnvironment
- HP IVM Config
- Node
- Virtualization Layer Software

注: トポロジを表示するには, 「[トポロジ](#)」(61ページ)を参照してください。

## パラメータ

```
reportHostNameAsVmName = false
```

## 第5章: HP Partitioning Solution ディスカバリ

### 本章の内容

概要 .....	66
サポートされるバージョン .....	66
トポロジ .....	67
HP vPars および nPars の検出方法 .....	70
HP nPars and vPars by Shell ジョブ .....	71
ディスカバリ・メカニズム .....	75
トラブルシューティングおよび制限事項 .....	99

## 概要

### • HP nPartitions

セルベースの HP サーバでは、**nPartitions** を設定することで、単一のサーバ・コンプレックスを1つの大規模システムまたは複数の小規模システムの集まりとして構成できます。各 nPartition は、独立したシステム環境として使用するサーバ・ハードウェア・リソースのサブセットを定義します。1つの nPartition には、それに割り当てた1つ以上のセル(プロセッサとメモリを搭載)と、それらのセルに接続したすべての I/O シャーシが含まれます。nPartition 内のすべてのプロセッサ、メモリ、I/O は、nPartition で実行されるソフトウェアによってのみ使用されます。そのため、各 nPartition には独自のシステム起動インタフェースがあり、各 nPartition の起動と再起動は独立して行われます。各 nPartition ではハードウェアとソフトウェアが分離しているため、ある nPartition でハードウェアまたはソフトウェアの障害が起きても、同じサーバ・コンプレックス内のほかの nPartition には影響しません。サーバの nPartition 定義は、HP のソフトウェアベース nPartition 管理ツールを使用することで再設定できます。サーバ・ハードウェア構成を物理的に変更する必要はありません。

### • HP vPartitions

vPars は仮想パーティション製品です。これを使用して、ハード・パーティションをさらに複数の仮想パーティションに分割することにより、1つのハード・パーティションで HP-UX の複数インスタンスを同時に実行できます。各仮想パーティションには独自のハードウェア・サブセットが割り当てられます。また、各仮想パーティションは HP-UX インスタンスを個別に実行し、独自のアプリケーション・セットをホストします。各 HP-UX インスタンスはほかのすべてのインスタンスから分離されるため、vPars ではアプリケーションとオペレーティング・システム(OS)の障害も分離されます。各 HP-UX インスタンスには、異なるパッチやカーネルを適用できます。

## サポートされるバージョン

このディスカバリは、vPars A.03.xx, A.04.xx, A.05.xx の各バージョンをサポートしています。

このパッケージは、HP-UX オペレーティング・システムを実行する vPars を搭載したセルラ・システムで検証済みです。

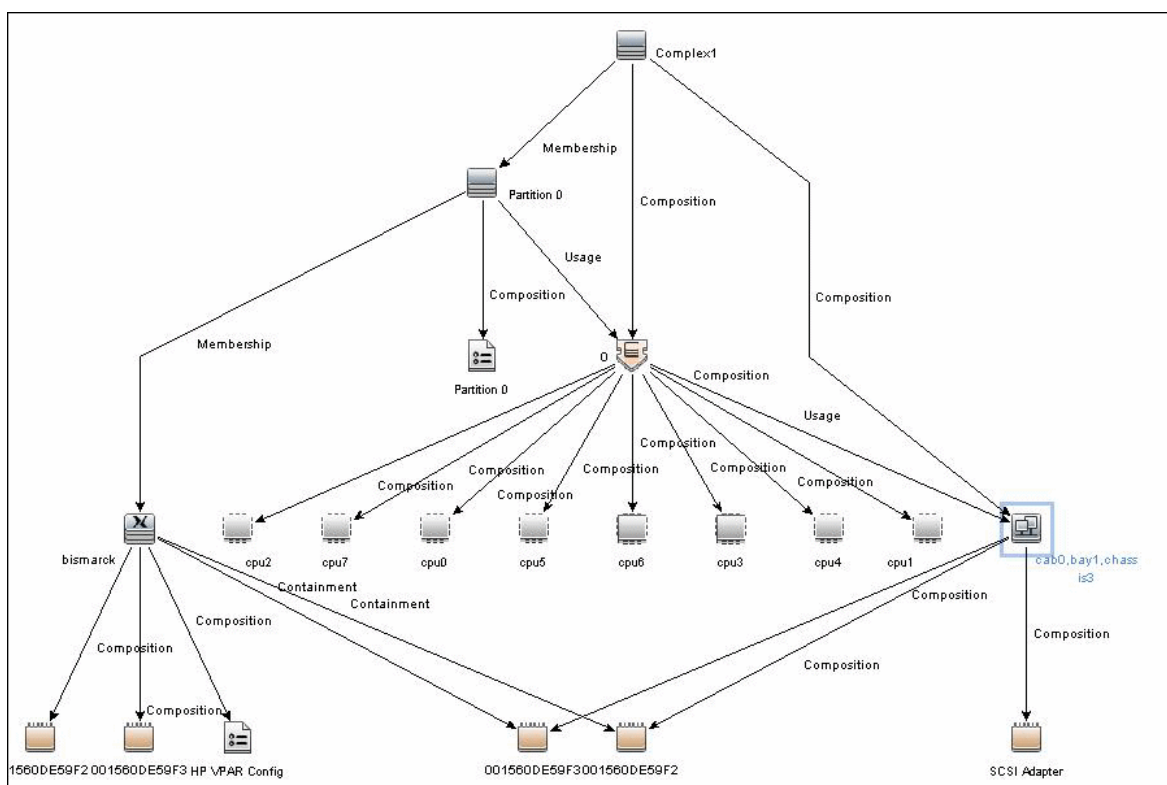
このディスカバリは、HP Blade ベースのコンプレックスをサポートしています。

## トポロジ

### 本項の内容

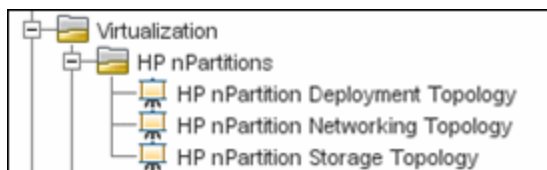
- 「HP vPars および nPars のトポロジ」(67ページ)
- 「HP nPartitions トポロジ・ビュー」(67ページ)

### HP vPars および nPars のトポロジ



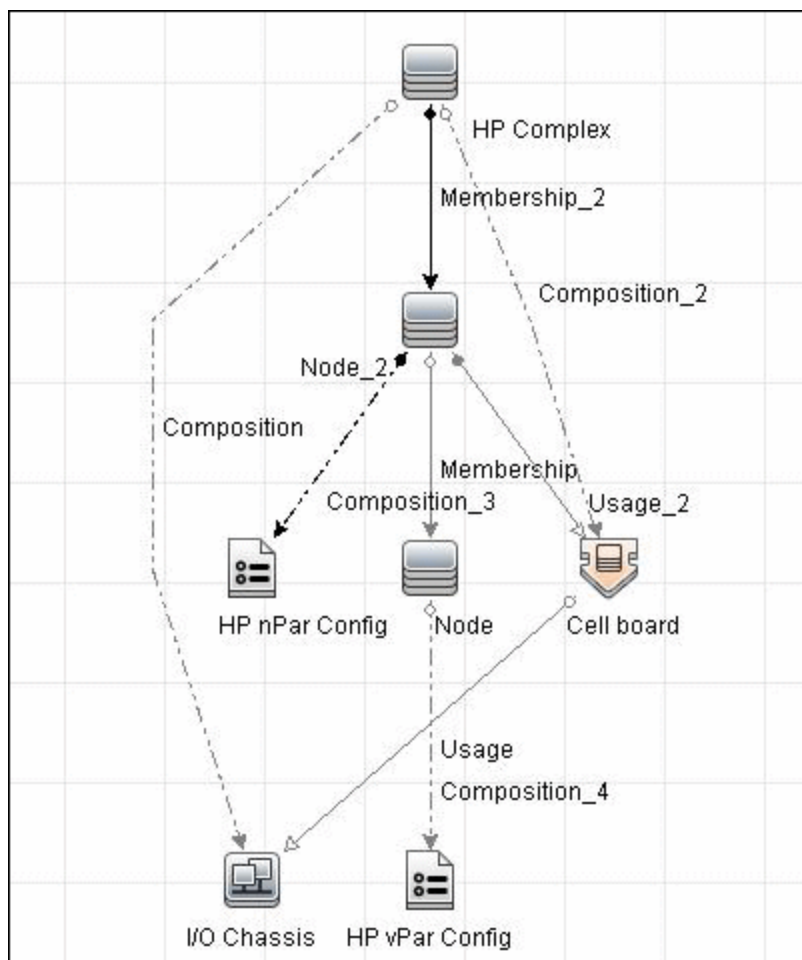
### HP nPartitions トポロジ・ビュー

HP nPartitions トポロジは、Virtualization モジュールの下、次のビューで表示されます。



### • HP nPartition Deployment Topology ビュー

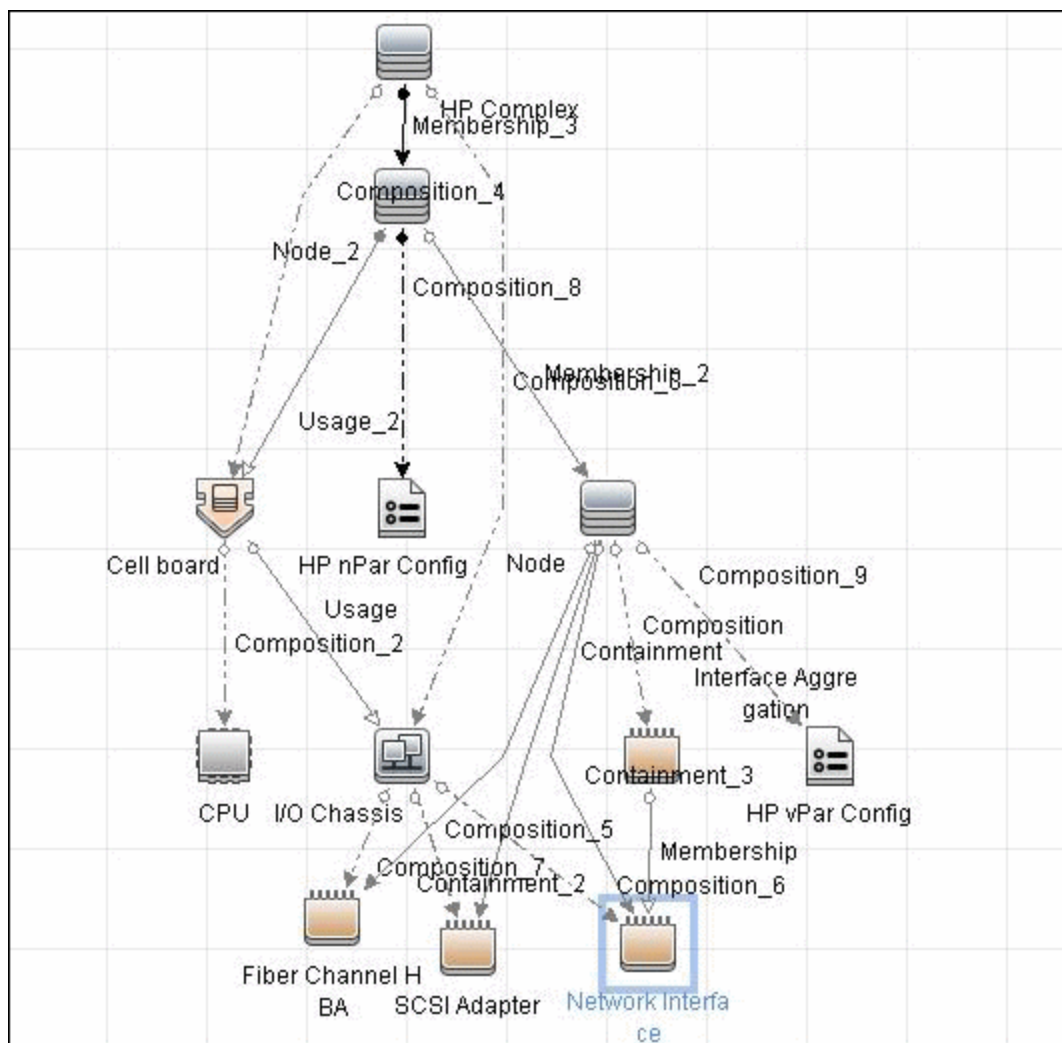
このビューは、nPars、vPars、セル、I/O シャーシのみを含む基本的な仮想化デプロイメントを表示します。





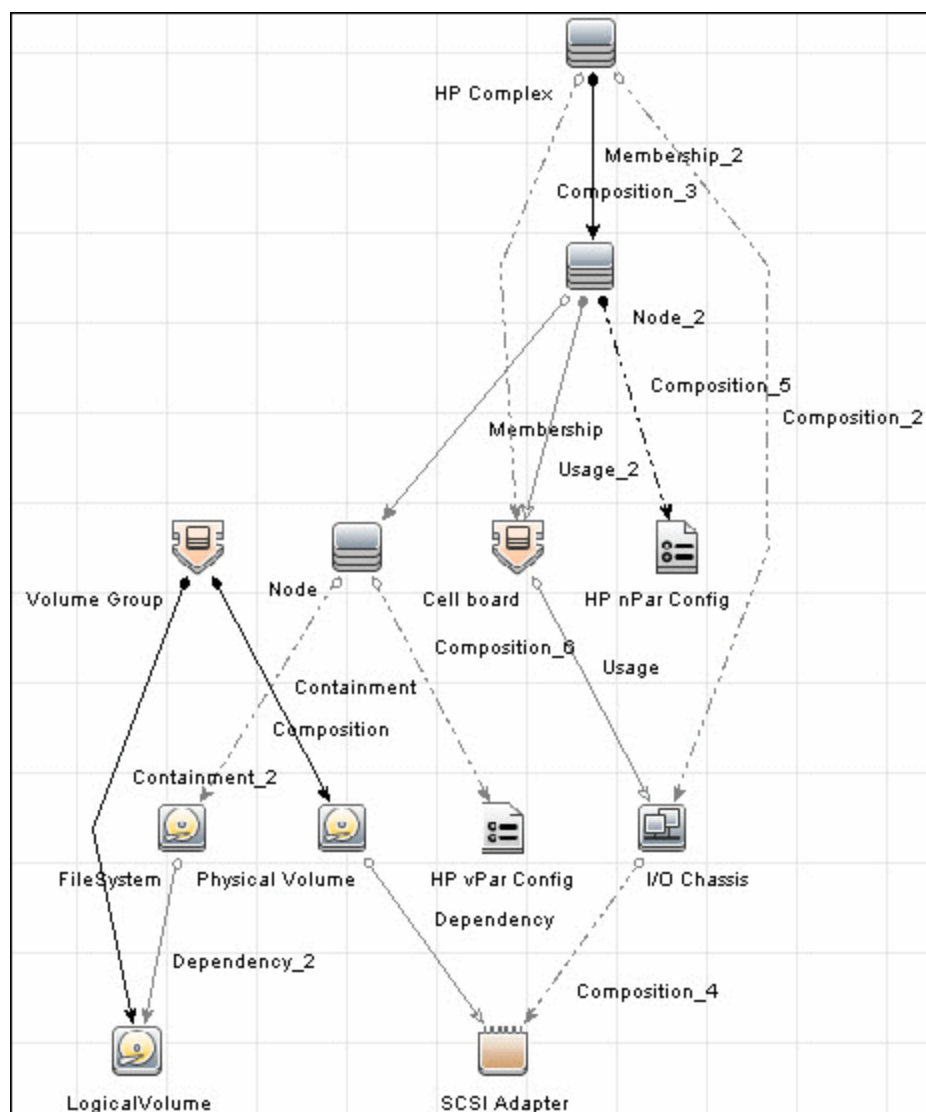
### • HP nPartition Networking Topology ビュー

このビューは、vPars の I/O デバイスと、I/O シャーシ上の物理的位置との関係を含む、nPartition デプロイメントのネットワークの側面を表示します。



### • HP nPartition Storage Topology ビュー

このビューは、ファイル・システムと論理ボリュームの関係を含む、HP nPartitions システムのストレージの側面を表示します。



## HP vPars および nPars の検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

プローブでシェルの資格情報が設定されていることを確認します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. ディスカバリの実行

HP vPars および nPars を検出するには、次の手順を実行します。

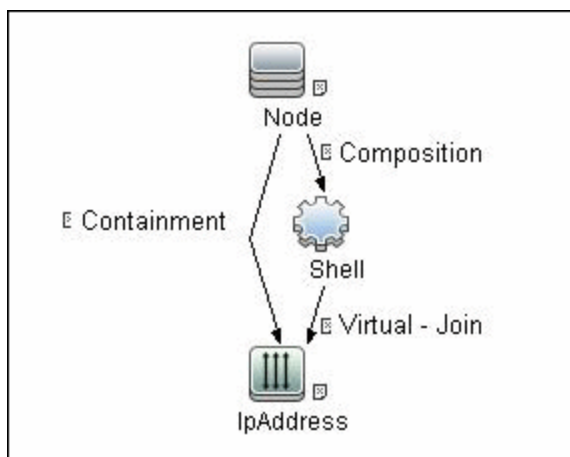
- a. **Range IPs by ICMP** ジョブを実行します。
- b. **Host Connection by Shell** ジョブを実行します。
- c. **HP nPars and vPars by Shell** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

# HP nPars and vPars by Shell ジョブ

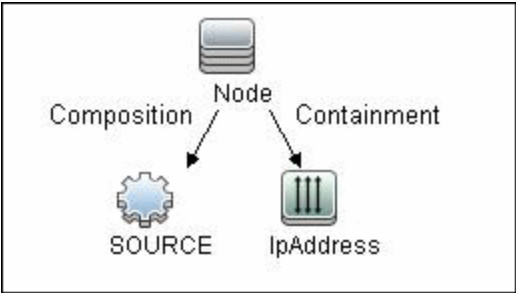
本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ



## アダプタ

- hp\_npar\_by\_shell アダプタの入力クエリ



作成または変更されたエンティティ

新しいクラス

- hp\_complex
- cell\_board
- io\_chassis
- hp\_npar\_config
- hp\_vpar\_config

End1	関係タイプ	End2
node	containment	fchba
node	containment	interface
node	containment	scsi_adapter
cell_board	composition	cpu
cell_board	composition	memory
hp_complex	composition	io_chassis
io_chassis	composition	fchba
io_chassis	composition	interface
io_chassis	composition	scsi_adapter

End1	関係タイプ	End2
cell_board	usage	io_chassis
node	usage	cell_board
node	usage	fchba
node	usage	interface

## 検出された CIT

- CellBoard
- Composition
- Containment
- Cpu
- Dependency
- Fibre Channel HBA
- FileSystem
- HP Complex
- HP nPar Config
- HP vPar Config
- I/O Chassis
- Interface
- Interface Aggregation
- LogicalVolume
- Membership

- Node
- Physical Volume
- SCSI Adapter
- Usage
- Volume Group

## ディスカバリ・メカニズム

### 本項の内容

- 「vPartition でのディスカバリの確認」(76ページ)
- 「nPartition でのディスカバリの確認」(76ページ)
- 「Complex に関する情報の取得」(76ページ)
- 「すべてのセルに関する一般情報の一覧表示」(78ページ)
- 「各セルの詳細情報の一覧表示」(78ページ)
- 「I/O シャーシに関する情報の取得」(84ページ)
- 「システム上の nPartition の名前一覧の取得」(85ページ)
- 「nPartition の詳細情報の取得」(85ページ)
- 「現在の vPartition の名前の取得」(89ページ)
- 「vPartition の詳細情報の取得」(89ページ)
- 「ファイバ・チャネル・アダプタの取得」(91ページ)
- 「ディスク・デバイスの取得」(92ページ)
- 「ネットワーク・インタフェースの取得」(93ページ)
- 「ファイル・システムの取得」(94ページ)
- 「論理ボリューム、ボリューム・グループ、物理ボリュームの取得」(94ページ)
- 「ネットワーク・インタフェースの取得」(96ページ)
- 「リンク・アグリゲーション・インタフェースに関する情報の取得」(97ページ)
- 「アグリゲーションされたインタフェースの MAC アドレスの取得」(97ページ)
- 「アグリゲーションされたインタフェースのハードウェア・パスの取得」(98ページ)
- 「アグリゲーションされたインタフェースの IP アドレスの取得」(98ページ)

## vPartition でのディスカバリの確認

目的	<ol style="list-style-type: none"> <li>1. ディスカバリがvPartition に接続したかどうかを確認します。</li> <li>2. コマンドをさらに実行して、サポートされている出力が生成されるかどうかを確認します。</li> </ol>
コマンド	vparstatus -V
出力	Version 2.0
取得された値	<ol style="list-style-type: none"> <li>1. 2.0 :vparstatus 実行可能ファイルのバージョン</li> <li>2. 戻りコード</li> </ol>
コメント	出力がサポートされるバージョンは 2.0 および 1.3

## nPartition でのディスカバリの確認

目的	ディスカバリがパーティション分割可能なサーバに接続したかどうかを確認します。
コマンド	parstatus -s
出力	なし
取得された値	戻りコード
コメント	戻りコードが0の場合、ディスカバリはパーティション分割可能なシステムに接続済み

## Complex に関する情報の取得

目的	HP Complex CIT のプロパティを取得します。
コマンド	parstatus -X
出力 (rp8420)	[Complex] Complex Name :Complex 01 Complex Capacity Compute Cabinet (4 cell capable) :1 Active GSP Location : cabinet 0



	<p>Model :9000/800/rp8420</p> <p>Serial Number :DEH45419K0</p> <p>Current Product Number :A6912A</p> <p>Original Product Number :A6912A</p> <p>Complex Profile Revision :1.0</p> <p>The total number of Partitions Present :2</p>
<b>出力(rx8640)</b>	<p>[Complex]</p> <p>Complex Name :Complex 01</p> <p>Complex Capacity</p> <p>Compute Cabinet (4 cell capable) :1</p> <p>Active MP Location : cabinet 0</p> <p>Original Product Name : server rx8640</p> <p>Original Serial Number :DEH4831H1Y</p> <p>Current Product Order Number :AB297A</p> <p>OEM Manufacturer :</p> <p>Complex Profile Revision :1.0</p> <p>The total number of partitions present :1</p>
<b>出力(sx3000)</b>	<p>[Complex]</p> <p>Complex Name:Complex 1</p> <p>Complex UUID: ab111111-2222-3333-4444-555555555555</p> <p>Complex Capacity Compute Enclosure:1 IO Enclosure:8</p> <p>Monarch OA Location:1</p> <p>Model:N/A</p> <p>Original Serial Number:AB11111111</p> <p>Current Product Order Number:CD2222</p> <p>OEM Manufacturer:N/A</p> <p>Total number of partitions present:2</p>
<b>取得された値</b>	<ul style="list-style-type: none"> <li>• Complex Name : 名前</li> <li>• Serial number/Original Serial Number : シリアル番号, ホスト・キー</li> </ul>
<b>コメント</b>	<p>HP Complex CIT は Host CIT から派生</p>

## すべてのセルに関する一般情報の一覧表示

目的	Complex 内にある全キャビネットのすべてのセル名の一覧を取得します。
コマンド	parstatus -C -M
出力 (rp8420)	<pre>cell: cab0, cell0: active core : 8/0/8 : 48.0/ 0.0: cab0, bay0, chassis0 : yes : yes : 0  cell: cab0, cell1: active core : 4/0/8 : 32.0/ 0.0: cab0, bay0, chassis1 : yes : yes : 1  cell: cab0, cell2: active base : 8/0/8 : 40.0/ 0.0: - : no : yes : 0  cell: cab0, cell3: active base : 4/0/8 : 32.0/ 0.0: - : no : yes : 1</pre>
出力 (rx8640)	<pre>cell: cab0, cell0: Active Core : 8/0/8 : 80.0/0.0 : cab0, bay0, chassis0 : yes : yes : 0  cell: cab0, cell1: Active Base : 8/0/8 : 80.0/0.0 : cab0, bay0, chassis1 : yes : yes : 0  cell: cab0, cell2: Active Base : 4/0/8 : 64.0/0.0 : - : no : yes : 0  cell: cab0, cell3: Absent : - : - : - : -</pre>
取得された値	セルの名前
コメント	セル名はこのあと各セルの詳細情報を取得するために使用

## 各セルの詳細情報の一覧表示

目的	Cell CI および対応する CPU CI と Memory CI に関するプロパティを取得します。
コマンド	parstatus -v -c <cell_number>
出力 (rp8420)	<pre>[Cell]  Hardware Location : cab0, cell0  Global Cell Number : 0  Actual Usage : active core  Normal Usage : base  Connected To : cab0, bay0, chassis0  Core Cell Capable : yes  Firmware Revision : 24.1</pre>

```

Failure Usage : activate
Use On Next Boot : yes
Partition Number :0
Partition Name : db01_ap02_db03_db04
[CPU Details]
Type :88E0
Speed :1100 MHz
CPU Status
=== =====
0 ok
1 ok
2 ok
3 ok
4 ok
5 ok
6 ok
7 ok
CPU
=====
OK :8
Deconf :0
Max :8
[Memory Details]
DIMM Size (MB) Status
==== =====
0A 4096 ok
4A 4096 ok
0B 4096 ok
4B 4096 ok
1A 4096 ok
5A 4096 ok
    
```

	<pre> 1B 4096 ok 5B 4096 ok 2A 4096 ok 2B 4096 ok 3A 4096 ok 3B 4096 ok Memory ===== DIMM OK :12 DIMM Deconf :0 Max DIMMs :16 Memory OK :48.00 GB Memory Deconf :0.00 GB </pre>
出力(rx8640)	<pre> [Cell] Hardware Location : cab0,cell0 Global Cell Number :0 Actual Usage :Active Core Normal Usage :Base Connected To : cab0,bay0,chassis0 Core Cell Capable : yes Firmware Revision :9.48 Failure Usage :Normal Use On Next Boot : yes Partition Number :0 Partition Name : db10_ap13_ap14_db15_db16_ap17_ap18_ap20 Requested CLM value :0.0 GB Allocated CLM value :0.0 GB Cell Architecture Type :Itanium(R)-based CPU Compatibility :CDH-640 Hyperthreading Capable : yes [CPU Details] </pre>

	Type :FFFF
	Speed :1598 MHz
	CPU Status
	=== =====
	0 OK
	1 OK
	2 OK
	3 OK
	4 OK
	5 OK
	6 OK
	7 OK
	CPU
	=====
	OK :8
	Deconf :0
	Max :8
	[Memory Details]
	DIMM Size (MB) Status
	==== =====
	3A 8192 OK
	3B 8192 OK
	1A 8192 OK
	1B 8192 OK
	4A 8192 OK
	4B 8192 OK
	0A 8192 OK
	0B 8192 OK
	2A 8192 OK
	2B 8192 OK
	Memory

	<pre>===== DIMM OK :10 DIMM Deconf :0 Max DIMMs :16 Memory OK :80.00 GB Memory Deconf :0.00 GB</pre>
--	--

取得された値	Global Cell Number > name		
	Hardware Location > hardware_path		
	Actual Usage > is_core	<b>Actual Usage</b> の値に <b>Core</b> という語が含まれている場合	
	Core Cell Capable > core_capable	<b>yes / no</b> をブール値に変換	
	Requested CLM value > requested_clm_value	<ul style="list-style-type: none"> <li>rp8420 サーバについてはこのパラメータなし</li> <li>GB から MB への変換が必要</li> </ul>	
	Allocated CLM value > allocated_clm_memory	<ul style="list-style-type: none"> <li>rp8420 サーバについてはこのパラメータなし</li> <li>GB から MB への変換が必要</li> </ul>	
	Use On Next Boot > use_on_next_boot	<b>yes / no</b> をブール値に変換	
	Failure Usage > failure_usage		
	Firmware Revision > firmware_revision		
	Cell Architecture Type > architecture_type	rp8420 サーバについてはこの値なし	
	CPU Compatibility > cpu_compatibility	rp8420 サーバについてはこの値なし	
	Hyperthreading Capable > is_hyperthreading_capable	<b>yes / no</b> をブール値に変換	
	CPU ===== OK :8	構成解除された CPU 数 : 0	

	Deconf :0 Max :8	CPU 最大数 : 8	
	Memory =====	メモリのサイズ : 80.00 GB	GB から MB への変換が必要
	DIMM OK :10	構成解除されたメモリ : 0.00 GB	
	DIMM Deconf :0	DIMM 最大数 : 16	
	Max DIMMs :16	構成解除された DIMM : 0	
	Memory OK :80.00 GB		
	Memory Deconf :0.00 GB		
コメント	UCMDB 9.x については Memory CI は作成されません(そのような CIT が無いため)。パーティション番号は、セルを nPartition に接続する際に使用されます(ホストとして表現)。		

## I/O シャーシに関する情報の取得

目的	Complex 内のすべての I/O シャーシ(I/O 拡張キャビネットを含む)のデータを取得します。		
コマンド	parstatus -l -M		
出力(rp8420)	chassis: cab0, bay0, chassis0 :active :yes : cab0, cell0:0 chassis: cab0, bay0, chassis1 :active :yes : cab0, cell1:1		
出力(rx8640)	chassis: cab0, bay0, chassis0 :Active :yes : cab0, cell0:0 chassis: cab0, bay0, chassis1 :Active :yes : cab0, cell1:0		
取得された値	名 前 : cab0, bay0, chassis0		
	使用法 : アクティブ		
	コアかどうか : yes	ブール値に変換	
コメント	セル・ハードウェアのパスは、シャーシをセルに接続する際に使用されます。		



## システム上の nPartition の名前一覧の取得

目的	システムで設定されている nPartition 番号の一覧を取得します。
コマンド	parstatus -P -M
出力 (rp8420)	partition:0 :active :2 :1 :cab0,cell0:db01_ap02_db03_db04 partition:1 :active :2 :1 :cab0,cell1:wdb1_wdb4
出力 (rx8640)	partition:0 :Active :3 :2 :cab0,cell0:db10_ap13_ap14_db15_ db16_ap17_
取得された値	nPartition 番号の一覧
コメント	これらの番号は、各 nPartition の詳細情報を取得するために使用されます。

## nPartition の詳細情報の取得

目的	各 nPartition の詳細情報を取得し、セルおよび HP nPar Config CI に接続するホストを作成します。
コマンド	parstatus -v -p <npartition_number>
出力 (rp8420)	[Partition]  Partition Number :0  Partition Name : db01_ap02_db03_db04  Status : active  IP address :0.0.0.0  Primary Boot Path :0/0/0/2/0.6.0  Alternate Boot Path :0/0/0/2/1.2.0  HA Alternate Boot Path :0/0/0/3/0.6.0  PDC Revision :24.1  IODCH Version :88E0  CPU Speed :1100 MHz  Core Cell : cab0,cell0  [Cell]  CPU Memory Use

	<pre> OK/ (GB) Core On  Hardware Actual Deconf/ OK/ Cell Next Par  Location Usage Max Deconf Connected To Capable Boot Num  ===== cab0,cell0 active core 8/0/8 48.0/ 0.0 cab0,bay0,chassis0 yes yes 0  cab0,cell2 active base 8/0/8 40.0/ 0.0 - no yes 0  [Chassis]  Core Connected Par  Hardware Location Usage IO To Num  ===== cab0,bay0,chassis0 active yes cab0,cell0 0 </pre>
出力(rx8640)	<pre> [Partition]  Partition Number :0  Partition Name : db10_ap13_ap14_db15_db16_ap17_ap18_ap20  Status :Active  IP Address :  Primary Boot Path :0/0/8/1/0/4/0.8.0.255.0.12.0  Alternate Boot Path :0/0/8/1/0/4/1.8.0.255.0.13.0  HA Alternate Boot Path :  PDC Revision :9.48  IODCH Version : ffff  Cell Architecture :Itanium(R)-based  CPU Compatibility :CDH-640  CPU Speed :1598 MHz  Core Cell : cab0,cell0  Core Cell Choice [0] : cab0,cell0  Total Good Memory Size :224.0 GB  Total Interleave Memory:224.0 GB  Total Requested CLM :0.0 GB  Total Allocated CLM :0.0 GB </pre>

	<p>Hyperthreading Enabled : no</p> <p>[Cell]</p> <p>CPU Memory Use</p> <p>OK/ (GB) Core On</p> <p>Hardware Actual Deconf/ OK/ Cell Next Par</p> <p>Location Usage Max Deconf Connected To Capable Boot Num</p> <p>=====</p> <p>cab0,cell0 Active Core 8/0/8 80.0/0.0 cab0,bay0,chassis0 yes yes 0</p> <p>cab0,cell1 Active Base 8/0/8 80.0/0.0 cab0,bay0,chassis1 yes yes 0</p> <p>cab0,cell2 Active Base 4/0/8 64.0/0.0 - no yes 0</p> <p>Notes:* = Cell has no interleaved memory.</p> <p>[Chassis]</p> <p>Core Connected Par</p> <p>Hardware Location Usage IO To Num</p> <p>=====</p> <p>cab0,bay0,chassis0 Active yes cab0,cell0 0</p> <p>[Chassis]</p> <p>Core Connected Par</p> <p>Hardware Location Usage IO To Num</p> <p>=====</p> <p>cab0,bay0,chassis1 Active yes cab0,cell1 0</p>
--	--

取得された値	ホスト(nPartition)	
	hostkey	ホスト・キーは nPartition 名と Complex のシリアル番号で構成
	Partition Name > tname	
	HP nPar Config	
	Constant "nPar Config" > name	

	Partition Name > npar_name	
	Status > npar_status	
	PDC Revision > pdc_revision	
	Hyperthreading Enabled > hyperthreading_mode	rp8420 サーバについてはこの値なし
	Partition Number > partition_number	
	Primary Boot Path > primary_boot_path	
	Alternate Boot Path > alternate_boot_path	

## 現在の vPartition の名前の取得

目的	現在の vPartition の名前を取得します。
コマンド	vparstatus -w -M
出力	doidb01
取得された値	ディスカバリが接続した vPartition の名前
コメント	一覧に含まれるのは、現在の vPartition に関する詳細情報のみです。 nPartition にあるすべての vPartition の詳細情報を取得することも可能ですが、それらの IP アドレスや下位の MAC アドレスを取得して UCMDB にホストを作成することはできません。

## vPartition の詳細情報の取得

目的	vPartition の詳細情報を取得し、Host CI と HP vPar Config CI を作成します。
コマンド	vparstatus -v -p <vPartition 名>
出力 (rp8420)	<pre>[Virtual Partition Details] Name: doidb01 State:Up Attributes:Dynamic,Autoboot,Nosearch Kernel Path:/stand/vmunix Boot Opts:-lq [CPU Details] Min/Max:3/16 Bound by User [Path]:0.15     0.16     0.17 Bound by Monitor [Path]: Unbound [Path]:2.14     2.15 [IO Details]</pre>

	<p>0.0.12</p> <p>0.0.14</p> <p>0.0.12.1.0.4.0.8.0.255.0.0.0</p> <p>0.0.14.1.0.4.0.8.0.255.0.1.0</p> <p>0.0.12.1.0.4.0.111.128.19.4.0.0</p> <p>0.0.12.1.0.4.0.111.88.19.5.0.0 BOOT</p> <p>0.0.14.1.0.4.0.112.88.19.5.0.0, ALTBOOT</p> <p>[Memory Details]</p> <p>Specified [Base /Range]:</p> <p>(bytes) (MB)</p> <p>Total Memory (MB):24448</p>
<b>出力(rx8640)</b>	<p>[Virtual Partition Details]</p> <p>Name: doiap17</p> <p>State:Up</p> <p>Attributes:Dynamic,Autoboot,Nosearch</p> <p>Kernel Path:/stand/vmunix</p> <p>Boot Opts:-lq</p> <p>[CPU Details]</p> <p>Min/Max:1/12</p> <p>User assigned [Path]:</p> <p>Boot processor [Path]:1.122</p> <p>Monitor assigned [Path]:</p> <p>Non-cell-specific:</p> <p>User assigned [Count]:1</p> <p>Monitor assigned [Count]:0</p> <p>Cell-specific [Count]:Cell ID/Count</p> <p>&lt;none&gt;</p> <p>[IO Details]</p> <p>0.0.8</p> <p>0.0.8.1.0.4.0.8.0.255.0.13.0</p> <p>0.0.8.1.0.4.0.8.0.255.0.12.0 BOOT</p>

	<p>0.0.8.1.0.4.1.8.0.255.0.13.0,ALTB00T</p> <p>[Memory Details]</p> <p>ILM, user-assigned [Base /Range]:</p> <p>(bytes) (MB)</p> <p>ILM, monitor-assigned [Base /Range]:0x11c0000000/8192</p> <p>(bytes) (MB)</p> <p>ILM Total (MB):8192</p> <p>ILM Granularity (MB):512</p> <p>CLM, user-assigned [CellID Base /Range]:</p> <p>(bytes) (MB)</p> <p>CLM, monitor-assigned [CellID Base /Range]:</p> <p>(bytes) (MB)</p> <p>CLM (CellID MB):</p> <p>CLM Granularity (MB):128</p>	
取得された値	Const "HP vPar Config" > name	
	Name > vpar_name	
	Boot Opts > boot_options	
	Boot processor [Path] > boot_processor_path	rp8420 サーバについてはこの値なし
	State > vpar_status	
	Attributes:Dynamic, Autoboot, Nosearch	<ul style="list-style-type: none"> <li>• 自動起動モード : Autoboot</li> <li>• 自動検索モード : Nosearch</li> <li>• 変更モード : Dynamic</li> </ul>
	Bound by User [Path]/User assigned [Path] > cpus_bound_by_user	実際のパラメータはサーバのバージョンに応じて異なる
	Unbound [Path] > unbound_cpus	
コメント	cpus_bound_by_userなどの属性の形式については、データ・モデルの仕様を参照してください。	

## ファイバ・チャネル・アダプタの取得

目的	ファイバ・チャネル・アダプタをモデル化します。
----	-------------------------

コマンド	ioscan -FnkCfc	
出力	<pre>pci:wsio:F:T:F:-1:50:4294967295:fc:fcd: 0/0/12/1/0/4/0:16 119 35 18 0 0 0 0 :0: root.cell.sba.lba.PCItoPCI.fcd:fcd: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC /2-port 1000B-T Combo Adapter (FC Port 1):0  /dev/fcd0  pci:wsio:F:T:F:-1:50:4294967295:fc:fcd: 0/0/12/1/0/4/1:16 119 35 18 0 0 0 0 :1: root.cell.sba.lba.PCItoPCI.fcd:fcd: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 2):1  /dev/fcd1  pci:wsio:F:T:F:-1:50:4294967295:fc:fcd: 0/0/14/1/0/4/0:16 119 35 18 0 0 0 0 : 2:root.cell.sba.lba.PCItoPCI.fcd:fcd: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 1):2  /dev/fcd2  pci:wsio:F:T:F:-1:50:4294967295:fc:fcd: 0/0/14/1/0/4/1:16 119 35 18 0 0 0 0 :3: root.cell.sba.lba.PCItoPCI.fcd:fcd: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X Fibre Channel 2-port 2Gb FC/2-port 1000B-T Combo Adapter (FC Port 2):3  /dev/fcd3</pre>	
取得された値	name	/dev/fcd0
	data_description	HP AB465-60001 PCI/PCI-X ファイバ・チャネル 2 ポート 2Gb FC/2 ポート 1000B-T コンボ・アダプタ(FC ポート 2)
コメント	ハードウェア・パスはセルの場所を示します。これは FC HBA のコンテナとして使用します。値の例は 0/0/14/1/0/4/0 です。最初の整数値はセルのグローバル ID で、2 番目の値は I/O シャーシの ID です。	

## ディスク・デバイスの取得

目的	I/O シャーシ、物理ディスク、SCSI アダプタ間の依存関係に関する情報を取得します。
----	--



<b>コマンド</b>	ioscan -FnkCdisk	
<b>出力</b>	<pre> scsi:wsio:T:T:F:31:188:2031616: disk:sdisk:0/0/12/1/0/4/0. 111.88.19.5.0.0:0 0 4 50 0 0 0 0 51 248 164 14 99 72 178 210 :3:root.cell.sba.lba.PCItoPCI.fcd.fcd_fcp.fcd_ vbus.tgt.sdisk: sdisk:CLAIMED:DEVICE:EMC SYMMETRIX:31  /dev/dsk/c31t0d0 /dev/rdisk/c31t0d0  scsi:wsio:T:T:F:31:188:2031872: disk:sdisk:0/0/12/1/0/4/0. 111.88.19.5.0.1:0 0 4 50 0 0 0 0 51 248 164 14 76 238 217 30 :59:root.cell.sba.lba. PCItoPCI.fcd.fcd_fcp.fcd_vbus.tgt. sdisk:sdisk:CLAIMED:DEVICE:EMC SYMMETRIX:31  /dev/dsk/c31t0d1 /dev/rdisk/c31t0d1  scsi:wsio:T:T:F:31:188:2032128: disk:sdisk:0/0/12/1/0/4/0. 111.88.19.5.0.2:0 0 4 50 0 0 0 0 51 248 164 14 101 17 172 238 :61:root.cell.sba.lba. PCItoPCI.fcd.fcd_fcp.fcd_vbus.tgt.sdisk:sdisk: CLAIMED:DEVICE:EMC SYMMETRIX:31  /dev/dsk/c31t0d2 /dev/rdisk/c31t0d2 </pre>	
<b>取得された値</b>	slot_number	0/0/12/1/0/4/0.111.88.19.5.0.0
	name	/dev/dsk/c31t0d2
	Cell ID	0/0/12/1/0/4/0.111.88.19.5.0.0
	IO chassis ID	0/0/12/1/0/4/0.111.88.19.5.0.0

## ネットワーク・インタフェースの取得

<b>目的</b>	ネットワーク・インタフェースとI/O シャーシ間の依存関係に関する情報を取得します。
<b>コマンド</b>	ioscan -FnkClan
<b>出力</b>	<pre> pci:wsio:F:F:F:-1:-1:4294967295:lan: igelan:0/0/12/1/0/6/0:20 228 22 72 0 0 0 0 :0: root.cell.sba.lba.PCItoPCI.igelan:igelan: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X </pre>

	<pre> 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:0 pci:wsio:F:F:F:-1:-1:4294967295:lan: igelan:0/0/12/1/0/6/1:20 228 22 72 0 0 0 0 :1: root.cell.sba.lba.PCItoPCI.igelan:igelan: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:1 pci:wsio:F:F:F:-1:-1:4294967295:lan: igelan:0/0/14/1/0/6/0:20 228 22 72 0 0 0 0 :2: root.cell.sba.lba.PCItoPCI.igelan:igelan: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:2 pci:wsio:F:F:F:-1:-1:4294967295:lan: igelan:0/0/14/1/0/6/1:20 228 22 72 0 0 0 0 :3: root.cell.sba.lba.PCItoPCI.igelan:igelan: CLAIMED:INTERFACE:HP AB465-60001 PCI/PCI-X 1000Base-T 2-port 2Gb FC/2-port 1000B-T Combo Adapter:3 </pre>
取得された値	このインタフェースが属するセルとI/O シャーシを反映したハードウェア・パス

## ファイル・システムの取得

目的	ファイル・システムとそれに対応する論理ボリュームに関する情報を取得します。
コマンド	df -P
出力	<pre> Filesystem 512-blocks Used Available Capacity Mounted on /dev/vg01/lv106 9837710 115094 9722616 2% /usr/vw/rvs /dev/vg01/lv124 7915344 814616 7100728 11% /home/kdov12 /dev/vg01/lv125 10222640 6275190 3947450 62% /home/ebrev /dev/vg01/lv123 20829536 2796208 18033328 14% /home/temp /dev/vg01/lv110 2080832 4608 2076224 1% /oracle2/arch/inst_ aebp </pre>
取得された値	<p>FileSystem CIT の名前 : /usr/vw/rvs</p> <p>論理ボリュームの名前 : /dev/vg01/lv106</p>

## 論理ボリューム, ボリューム・グループ, 物理ボリュームの取得

目的	論理ボリューム, ボリューム・グループ, 物理ボリュームをモデル化するためのデータを取得します。
----	--

コマンド	vgdisplay -v
出力	<pre>--- Volume groups --- VG Name /dev/vg00 VG Write Access read/write VG Status available Max LV 255 Cur LV 10 Open LV 10 Max PV 16 Cur PV 1 Act PV 1 Max PE per PV 4384 VGDA 2 PE Size (Mbytes) 16 Total PE 4315 Alloc PE 4156 Free PE 159 Total PVG 0 Total Spare PVs 0 Total Spare PVs in use 0 --- Logical volumes --- LV Name /dev/vg00/lvol1 LV Status available/syncd LV Size (Mbytes) 256 Current LE 16 Allocated PE 16 Used PV 1 --- Physical volumes --- PV Name /dev/dsk/c31t0d0</pre>

	PV Name /dev/dsk/c32t0d0 Alternate Link PV Status available Total PE 4315 Free PE 159 Autoswitch On Proactive Polling On	
取得された値	ボリューム・グループ	
	VG Name > name	
	VG Write Access > write_access	
	VG Status > vg_status	この値は、物理ボリュームのサイズを計算するために使用されます。
	PE Size (Mbytes)	
	Logical Volume	
	LV Name > name	
	LV Status > lv_status	
	物理ボリューム	
	PV Name > name	代替リンクを使用することもできます。これは、 <b>ioscan FnkCdisk</b> コマンドの出力結果によって変わります。
	PV Status > pv_status	
	Total PE > pv_size	この属性は、PE Size (Mbytes) の値に基づいて計算されます。

## ネットワーク・インタフェースの取得

目的	ネットワーク・インタフェースに関する情報を取得します。
コマンド	lanscan
出力	Hardware Station Crd Hdw Net-Interface NM MAC HP-DLPI DLPI Path Address In# State NamePPA ID Type Support Mjr# 0/0/4/1/0/6/1 0x0014C254D9BD 1 UP lan1 snap1 2 ETHER Yes 119

	<pre>0/0/6/1/0/6/1 0x0014C254C961 3 UP lan3 snap3 4 ETHER Yes 119 LinkAgg0 0x0014C254D9BC 900 UP lan900 snap900 6 ETHER Yes 119 LinkAgg1 0x000000000000 901 DOWN lan901 snap901 7 ETHER Yes 119 LinkAgg2 0x000000000000 902 DOWN lan902 snap902 8 ETHER Yes 119 LinkAgg3 0x000000000000 903 DOWN lan903 snap903 9 ETHER Yes 119 LinkAgg4 0x000000000000 904 DOWN lan904 snap904 10 ETHER Yes 119</pre>
取得された値	<ul style="list-style-type: none"> <li>ネットワーク・インタフェースとI/O シャーシとの間のリンクを作成するためのハードウェア・パス</li> <li>ネットワーク・インタフェースを作成するための MAC アドレス</li> <li>リンク・アグリゲーション・インタフェースの MAC アドレス, インタフェースが稼働していることを示すインジケータ, デバイス名</li> </ul>

## リンク・アグリゲーション・インタフェースに関する情報の取得

目的	インタフェースとリンク・アグリゲーションとの間のリンクをモデル化します。
コマンド	lanscan -q
出力	<pre>1 3 900 0 2 901 902 903 904</pre>
取得された値	アグリゲーションされたインタフェースのインタフェース番号とID

## アグリゲーションされたインタフェースの MAC アドレスの取得

目的	アグリゲーションされたインタフェースの MAC アドレスを取得します。
----	-------------------------------------

コマンド	lanadmin -a <interface_id>
例	lanscan -a 0
出力	Station Address = 0x0014c254d9bc
取得された値	アグリゲーションされたインタフェースの MAC アドレス

## アグリゲーションされたインタフェースのハードウェア・パスの取得

目的	アグリゲーションされたインタフェースのハードウェア・パスを取得する
コマンド	lanscan -v   grep -E <list_of_aggregated_interfaces>
例	lanscan -v   grep -E "lan0 lan2"
出力	0/0/4/1/0/6/0 0 UP lan0 snap0 1 ETHER Yes 119 igelan 0/0/6/1/0/6/0 2 UP lan2 snap2 3 ETHER Yes 119 igelan
取得された値	このインタフェースを保持する I/O シャーシを割り当てるハードウェア・パス

## アグリゲーションされたインタフェースの IP アドレスの取得

目的	インタフェースの IP アドレスを取得します。
コマンド	netstat -m
出力	Routing tables  Destination Gateway Flags Refs Interface Pmtu 127.0.0.1 127.0.0.1 UH 0 lo0 4136  10.186.112.115 10.186.112.115 UH 0 lan0 4136 10.186.116.13 10.186.116.13 UH 0 lan1 4136 192.168.121.1 192.168.121.1 UH 0 lan2 4136 10.186.115.18 10.186.115.18 UH 0 lan3 4136 10.186.116.19 10.186.116.19 UH 0 lan1:1 4136 10.186.116.0 10.186.116.13 U 3 lan1 1500 10.186.116.0 10.186.116.19 U 3 lan1:1 1500 10.186.115.0 10.186.115.18 U 2 lan3 1500 10.186.112.0 10.186.112.115 U 2 lan0 1500

	<pre>192.168.121.0 192.168.121.1 U 2 lan2 1500 10.186.86.0 10.186.115.1 UG 0 lan3 1500 127.0.0.0 127.0.0.1 U 0 lo0 4136 default 10.186.116.1 UG 0 lan1 1500</pre>
取得された値	<p>インタフェースの IP アドレス</p> <p><b>ifconfig</b> とは異なり, <b>netstat</b> コマンドにルート権限は必要ありません。</p>

## トラブルシューティングおよび制限事項

- ・ ターゲット・ホストが HP nPartition システムの一部ではない

DFM は, ターゲット・ホストを HP のパーティション分割可能なシステムの一部ではないとみなします。基準は, **parstatus -s** コマンドの実行に基づきます。

- ・ vPartition の詳細を検出できない

**vparstatus** コマンドが正常に実行されませんでした。このコマンドがアクセス可能であること, および DFM に, 十分な実行権限があることが必要です。**sudo** の実行がこのコマンドに必要な場合は, SSH 資格情報を設定します。

資格情報の詳細については, 『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

- ・ ストレージ・トポロジを検出できない

**vgdisplay** コマンドが正常に実行されませんでした。

- ・ ファイル・システムとディスクをリンクできない

**df** コマンドが正常に実行されませんでした。

- ・ SCSI アダプタ, ファイバ・チャネル・アダプタ, またはネットワーク・カードを検出できない

**ioscan** コマンドが正常に実行されませんでした。

## 第6章: Hyper-V ディスカバリ

### 本章の内容

概要 .....	101
サポートされるバージョン .....	101
トポロジ .....	101
Hyper-V の検出方法 .....	102
ディスカバリ・メカニズム .....	103
\\root\virtualization 名前空間のディスカバリ・メカニズム .....	104
\\root\virtualization\v2 名前空間のディスカバリ・メカニズム .....	110
Hyper-V Topology by Shell ジョブ .....	117
Hyper-V Topology by WMI ジョブ .....	119
作成または変更されたエンティティ .....	121
トラブルシューティングおよび制限事項 .....	122



## 概要

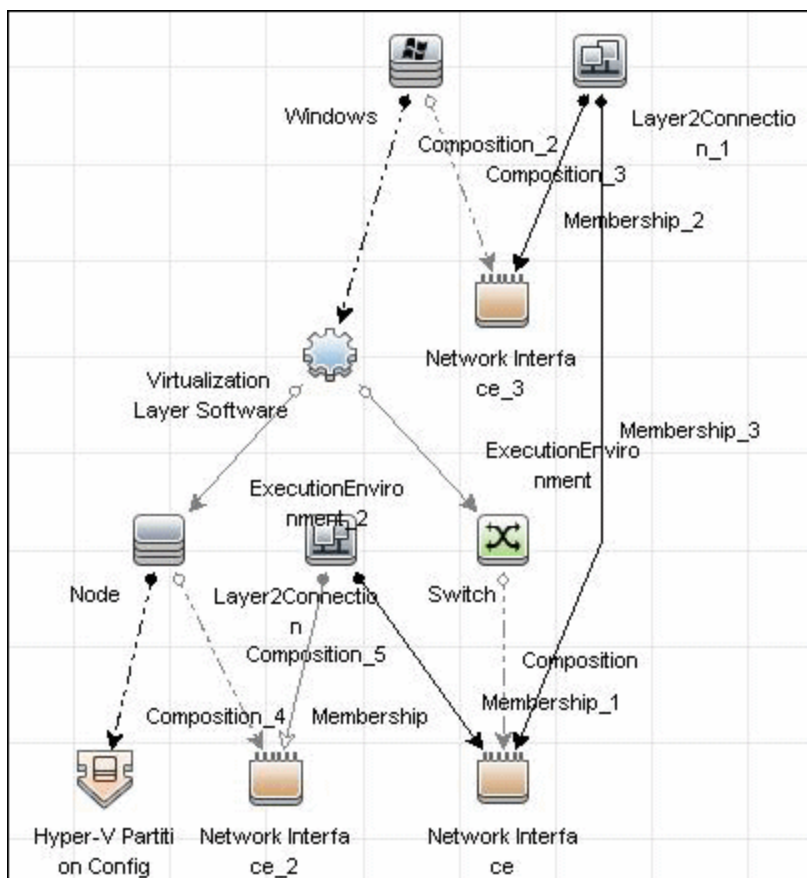
**Hyper-V** パッケージでは、WMI および NTCMD を介して Hyper-V 対応の Windows サーバが検出されます。検出対象はリソース・プール、仮想スイッチ、仮想 NIC、仮想マシンです。

## サポートされるバージョン

**Hyper-V** パッケージは、Windows 2008、Windows 2008 R2、Windows Server 2012、Windows Server 2012 R2 をサポートしています。

## トポロジ

次の図に、Hyper-V ディスカバリのトポロジを示します。



# Hyper-V の検出方法

本項の内容

## 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、NTCMD プロトコルおよび WMI プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - 検証

ターゲット・マシン上の `\\root\virtualization` 名前空間または `\\root\virtualization\v2` 名前空間で WMI を介して、またはシェル・プロトコルでの接続中に `wmic` コマンドを介して WMI クエリを実行できることを確認します。

## 3. ディスカバリの実行

シェルを介して Hyper-V トポロジを検出するには、次の手順を実行します。

- a. **Range IPs by ICMP** ジョブを実行し、IP 範囲内で稼働中のマシンを検出します。
- b. **Host Connection by Shell** ジョブを実行し、シェル接続と、ホストに関する基本情報を検出します。
- c. **Host Applications by Shell** ジョブを実行し、ターゲット・マシン上のプロセスを検出します。
- d. **Hyper-V Topology by Shell** ジョブを実行し、Hyper-V トポロジを検出します。

WMI を介して Hyper-V トポロジを検出するには、次の手順を実行します。

- a. **Range IPs by ICMP** ジョブを実行し、IP 範囲内で稼働中のマシンを検出します。
- b. **Host Connection by WMI** ジョブを実行し、WMI 接続と、ホストに関する基本情報を検出します。
- c. **Host Applications by WMI** ジョブを実行し、ターゲット・マシン上のプロセスを検出します。
- d. **Hyper-V Topology by WMI** ジョブを実行して Hyper-V トポロジを検出します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

# ディスカバリ・メカニズム

サポートされている名前空間の WMI クラスおよび属性の列挙が含まれます。

\\root\virtualization 名前空間のディスカバリ・メカニズム .....	104
\\root\virtualization\v2 名前空間のディスカバリ・メカニズム .....	110

## \\root\virtualization 名前空間のディスカバリ・メカニズム

### 本項の内容

- ・「Hyper-V ホスト名の取得」(104ページ)
- ・「仮想マシンの取得」(105ページ)
- ・「仮想マシンのグローバル設定の取得」(105ページ)
- ・「仮想マシンの設定の取得」(105ページ)
- ・「仮想マシンから設定 (VSSD) へのリファレンスの取得」(106ページ)
- ・「仮想マシン設定 (VSSD) からコンポーネントへのリファレンスの取得」(106ページ)
- ・「仮想マシンのメモリ設定の取得」(107ページ)
- ・「仮想マシンのプロセッサ設定の取得」(107ページ)
- ・「仮想スイッチの取得」(107ページ)
- ・「仮想スイッチのポートの取得」(108ページ)
- ・「仮想スイッチからポートへのリファレンスの取得」(108ページ)
- ・「仮想マシンのインタフェースの取得」(108ページ)
- ・「管理パーティションのインタフェースの取得」(109ページ)
- ・「仮想マシンからインタフェースへのリファレンスの取得」(109ページ)
- ・「仮想スイッチのポートからインタフェースへのリファレンスの取得」(109ページ)

### Hyper-V ホスト名の取得

クエリされるオブジェクト	Msvm_ComputerSystem
条件	Description = 'Microsoft Hosting Computer System'
クエリされるプロパティ	ElementName

コメント	Hyper-V 名前空間 <b>\\root\virtualization</b> にアクセスできることを確認し、Hyper-V ホストの名前を取得します。
------	--

## 仮想マシンの取得

クエリされるオブジェクト	Msvm_ComputerSystem
条件	Description = 'Microsoft Virtual Machine'
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Name</li> <li>• ElementName</li> <li>• EnabledState</li> <li>• HealthState</li> </ul>
コメント	Hyper-V ホスト上にある仮想マシンを取得し、各仮想マシンの GUID、名前、状況、有効化状態を取得します。

## 仮想マシンのグローバル設定の取得

クエリされるオブジェクト	Msvm_VirtualSystemGlobalSettingData
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• SystemName</li> <li>• SnapshotDataRoot</li> <li>• ExternalDataRoot</li> <li>• AutomaticRecoveryAction</li> <li>• AutomaticShutdownAction</li> <li>• AutomaticStartupAction</li> </ul>
コメント	すべての仮想マシンのグローバル設定を取得します。

## 仮想マシンの設定の取得

クエリされるオブジェクト	Msvm_VirtualSystemSettingData
--------------	-------------------------------

条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• InstanceID</li> <li>• BaseBoardSerialNumber</li> <li>• BIOSGUID</li> <li>• BIOSSerialNumber</li> <li>• ChassisAssetTag</li> <li>• ChassisSerialNumber</li> </ul>
コメント	<p>仮想マシンの追加設定を保持する仮想マシンの <b>VirtualSystemSettingData</b>(VSSD)オブジェクトを取得します。</p> <p><b>BIOSGUID</b> プロパティに、仮想マシンの BIOS UUID が保持されます。このプロパティから、先頭と末尾の中間の括弧({})は削除されます。</p>

### 仮想マシンから設定 (VSSD)へのリファレンスの取得

クエリされるオブジェクト	Msvm_SettingsDefineState
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• ManagedElement</li> <li>• SettingData</li> </ul>
コメント	仮想マシンとそれらの設定 ( <b>VirtualSystemSettingData</b> )を関連付けます。

### 仮想マシン設定 (VSSD)からコンポーネントへのリファレンスの取得

クエリされるオブジェクト	Msvm_VirtualSystemSettingDataComponent
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• GroupComponent</li> <li>• PartComponent</li> </ul>
コメント	<b>VirtualSystemSettingData</b> オブジェクトからそのコンポーネントへのリファレンスを取得します。

## 仮想マシンのメモリ設定の取得

クエリされるオブジェクト	Msvm_MemorySettingData
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"><li>• InstanceID</li><li>• Limit</li><li>• Reservation</li></ul>
コメント	仮想マシンのメモリ設定(予約および制限)を取得します。前の手順( <a href="#">「ディスカバリ・メカニズム」(103ページ)</a> )で取得したリファレンスによって、これらの設定を関連する仮想マシンに適切に関連付けることができます。

## 仮想マシンのプロセッサ設定の取得

クエリされるオブジェクト	Msvm_ProcessorSettingData
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"><li>• InstanceID</li><li>• Limit</li><li>• Reservation</li><li>• Weight</li></ul>
コメント	仮想マシンのプロセッサ設定(予約, 制限, 重み)を取得します。前の手順( <a href="#">「ディスカバリ・メカニズム」(103ページ)</a> )で取得したリファレンスによって、これらの設定を関連する仮想マシンに適切に関連付けることができます。

## 仮想スイッチの取得

クエリされるオブジェクト	Msvm_VirtualSwitch
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"><li>• ElementName</li><li>• Name</li></ul>

コメント	Hyper-V ホストに構成された仮想スイッチを取得します。
------	--------------------------------

### 仮想スイッチのポートの取得

クエリされるオブジェクト	Msvm_SwitchPort
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• ElementName</li> <li>• Name</li> </ul>
コメント	仮想スイッチのポートを取得します。

### 仮想スイッチからポートへのリファレンスの取得

クエリされるオブジェクト	Msvm_HostedAccessPoint
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Antecedent</li> <li>• Dependent</li> </ul>
コメント	仮想スイッチとそれらのポートとを関連付けられるようにするリファレンスを取得します。

### 仮想マシンのインタフェースの取得

クエリされるオブジェクト	Msvm_VmLANEndpoint
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Name</li> <li>• ElementName</li> <li>• MACAddress</li> </ul>
コメント	仮想マシンのインタフェースに接続されているエンドポイントを取得します。これらのエンドポイントは、これら自体がインタフェースではありませんが、インタフェースをレポートするために十分な情報を保持しています。



**管理パーティションのインタフェースの取得**

クエリされるオブジェクト	Msvm_SwitchLANEndpoint
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Name</li> <li>• ElementName</li> <li>• MACAddress</li> </ul>
コメント	Hyper-V ホスト上の管理パーティションのインタフェースに接続されているエンドポイントを取得します。これらのエンドポイントは、これら自体がインタフェースではありませんが、インタフェースをレポートするために十分な情報を保持しています。これらには、仮想マシンへの内部接続に使用されているパーティションの物理インタフェースと仮想インタフェースの両方が含まれます。

**仮想マシンからインタフェースへのリファレンスの取得**

クエリされるオブジェクト	Msvm_DeviceSAPImplementation
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Antecedent</li> <li>• Dependent</li> </ul>
コメント	仮想エンドポイントから仮想マシンへのリファレンスを取得します。これにより、関連付けが可能になります。

**仮想スイッチのポートからインタフェースへのリファレンスの取得**

クエリされるオブジェクト	Msvm_ActiveConnection
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Antecedent</li> <li>• Dependent</li> </ul>
コメント	仮想スイッチのポートからエンドポイントへのリファレンスを取得します。これにより、関連付けが可能になります。

## \\root\virtualization\v2 名前空間のディスカバリ・メカニズム

### 本項の内容

- 「Hyper-V ホスト名の取得」(111ページ)
- 「仮想マシンの取得」(111ページ)
- 「仮想マシンのグローバル設定の取得」(111ページ)
- 「仮想マシンの設定の取得」(112ページ)
- 「仮想マシンから設定 (VSSD) へのリファレンスの取得」(112ページ)
- 「仮想マシン設定 (VSSD) からコンポーネントへのリファレンスの取得」(112ページ)
- 「仮想マシンのメモリ設定の取得」(113ページ)
- 「仮想マシンのプロセッサ設定の取得」(113ページ)
- 「仮想スイッチの取得」(114ページ)
- 「仮想スイッチのポートの取得」(114ページ)
- 「仮想マシンのインターフェースの取得」(114ページ)
- 「\\root\virtualization\v2 名前空間のディスカバリ・メカニズム」(110ページ)
- 「仮想マシンからインターフェースへのリファレンスの取得」(115ページ)
- 「仮想スイッチのポートからインターフェースへのリファレンスの取得」(115ページ)
- 「Hyper-V ホストのインターフェースの取得」(115ページ)
- 「統合イーサネット・アダプタの取得」(116ページ)
- 「エミュレートされたイーサネット・アダプタの取得」(116ページ)
- 「内部イーサネット・アダプタ(ネットワーク・アダプタ)の取得」(116ページ)
- 「外部イーサネット・アダプタ(ネットワーク・アダプタ)の取得」(117ページ)
- 「LAN エンドポイントからグローバル・イーサネット・ポートへのリファレンスの取得」(117ページ)

## Hyper-V ホスト名の取得

クエリされるオブジェクト	Msvm_ComputerSystem
条件	Description = 'Microsoft Hosting Computer System'
クエリされるプロパティ	ElementName
コメント	Hyper-V 名前空間 <code>\\root\virtualization\v2</code> にアクセスできることを確認し、Hyper-V ホストの名前を取得します。

## 仮想マシンの取得

クエリされるオブジェクト	Msvm_ComputerSystem
条件	Description = 'Microsoft Virtual Machine'
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Name</li> <li>• ElementName</li> <li>• EnabledState</li> <li>• HealthState</li> </ul>
コメント	Hyper-V ホスト上にある仮想マシンを取得し、各仮想マシンの GUID、名前の状況、有効化状態を取得します。

## 仮想マシンのグローバル設定の取得

クエリされるオブジェクト	Msvm_VirtualSystemSettingData
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• InstanceID</li> <li>• SnapshotDataRoot</li> <li>• ExternalDataRoot</li> <li>• AutomaticRecoveryAction</li> <li>• AutomaticShutdownAction</li> </ul>

	<ul style="list-style-type: none"> <li>AutomaticStartupAction</li> </ul>
コメント	すべての仮想マシンのグローバル設定を取得します。

## 仮想マシンの設定の取得

クエリされるオブジェクト	Msvm_VirtualSystemSettingData
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>InstanceID</li> <li>BaseBoardSerialNumber</li> <li>BIOSGUID</li> <li>BIOSSerialNumber</li> <li>ChassisAssetTag</li> <li>ChassisSerialNumber</li> </ul>
コメント	<p>仮想マシンの追加設定を保持する仮想マシンの <b>VirtualSystemSettingData</b>(VSSD)オブジェクトを取得します。</p> <p><b>BIOSGUID</b> プロパティに、仮想マシンの BIOS UUID が保持されます。このプロパティから、先頭と末尾の中かっこ({})は削除されます。</p>

## 仮想マシンから設定(VSSD)へのリファレンスの取得

クエリされるオブジェクト	Msvm_SettingsDefineState
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>ManagedElement</li> <li>SettingData</li> </ul>
コメント	仮想マシンとそれらの設定 ( <b>VirtualSystemSettingData</b> )を関連付けます。

## 仮想マシン設定(VSSD)からコンポーネントへのリファレンスの取得

クエリされるオブジェクト	Msvm_VirtualSystemSettingDataComponent
--------------	--

条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• GroupComponent</li> <li>• PartComponent</li> </ul>
コメント	<b>VirtualSystemSettingData</b> オブジェクトからそのコンポーネントへのリファレンスを取得します。

## 仮想マシンのメモリ設定の取得

クエリされるオブジェクト	Msvm_MemorySettingData
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• InstanceID</li> <li>• Limit</li> <li>• Reservation</li> </ul>
コメント	仮想マシンのメモリ設定(予約および制限)を取得します。前の手順( <a href="#">「ディスカバリ・メカニズム」(103ページ)</a> )で取得したリファレンスによって、これらの設定を関連する仮想マシンに適切に関連付けることができます。

## 仮想マシンのプロセッサ設定の取得

クエリされるオブジェクト	Msvm_ProcessorSettingData
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• InstanceID</li> <li>• Limit</li> <li>• Reservation</li> <li>• Weight</li> </ul>
コメント	仮想マシンのプロセッサ設定(予約, 制限, 重み)を取得します。前の手順( <a href="#">「ディスカバリ・メカニズム」(103ページ)</a> )で取得したリファレンスによって、これらの設定を関連する仮想マシンに適切に関連付けることができます。

## 仮想スイッチの取得

クエリされるオブジェクト	Msvm_VirtualEthernetSwitch
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• ElementName</li> <li>• Name</li> </ul>
コメント	Hyper-V ホストに構成された仮想スイッチを取得します。

## 仮想スイッチのポートの取得

クエリされるオブジェクト	Msvm_EthernetSwitchPort
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• ElementName</li> <li>• Name</li> </ul>
コメント	仮想スイッチのポートを取得します。

## 仮想マシンのインタフェースの取得

クエリされるオブジェクト	Msvm_LANEndpoint
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Name</li> <li>• ElementName</li> <li>• MACAddress</li> </ul>
コメント	仮想マシンのインタフェースに接続されているエンドポイントを取得します。これらのエンドポイントは、これら自体がインタフェースではありませんが、インタフェースをレポートするために十分な情報を保持しています。

### 仮想マシンからインタフェースへのリファレンスの取得

クエリされるオブジェクト	Msvm_DeviceSAPImplementation
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Antecedent</li> <li>• Dependent</li> </ul>
コメント	仮想エンドポイントから仮想マシンへのリファレンスを取得します。これにより、関連付けが可能になります。

### 仮想スイッチのポートからインタフェースへのリファレンスの取得

クエリされるオブジェクト	Msvm_ActiveConnection
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Antecedent</li> <li>• Dependent</li> </ul>
コメント	仮想スイッチのポートからエンドポイントへのリファレンスを取得します。これにより、関連付けが可能になります。

### Hyper-V ホストのインタフェースの取得

クエリされるオブジェクト	Msvm_LANEndpoint
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Name</li> <li>• ElementName</li> <li>• MACAddress</li> </ul>
コメント	Hyper-V ホストのインタフェースを取得します。

### 統合イーサネット・アダプタの取得

クエリされるオブジェクト	Msvm_SyntheticEthernetPort
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"><li>• DeviceID</li><li>• ElementName</li><li>• PermanentAddress</li><li>• SystemName</li></ul>
コメント	統合イーサネット・インタフェースを取得します。

### エミュレートされたイーサネット・アダプタの取得

クエリされるオブジェクト	Msvm_EmulatedEthernetPort
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"><li>• DeviceID</li><li>• ElementName</li><li>• PermanentAddress</li><li>• SystemName</li></ul>
コメント	エミュレートされたイーサネット・インタフェースを取得します。

### 内部イーサネット・アダプタ(ネットワーク・アダプタ)の取得

クエリされるオブジェクト	Msvm_InternalEthernetPort
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"><li>• DeviceID</li><li>• ElementName</li><li>• PermanentAddress</li></ul>



	<ul style="list-style-type: none"> <li>• SystemName</li> </ul>
コメント	内部イーサネット・インタフェースを取得します。

### 外部イーサネット・アダプタ(ネットワーク・アダプタ)の取得

クエリされるオブジェクト	Msvm_ExternalEthernetPort
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• DeviceID</li> <li>• ElementName</li> <li>• PermanentAddress</li> <li>• SystemName</li> </ul>
コメント	外部イーサネット・インタフェースを取得します。

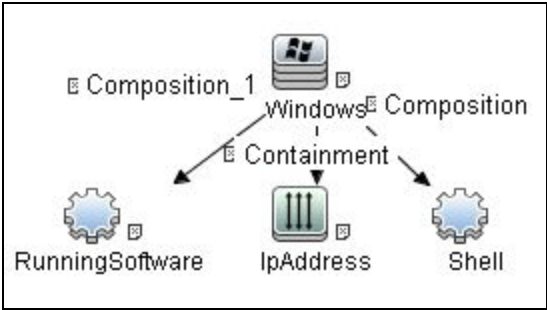
### LAN エンドポイントからグローバル・イーサネット・ポートへのリファレンスの取得

クエリされるオブジェクト	Msvm_EthernetDeviceSAPImplementation
条件	なし
クエリされるプロパティ	<ul style="list-style-type: none"> <li>• Antecedent</li> <li>• Dependent</li> </ul>
コメント	LAN エンドポイントからグローバル・イーサネット・ポートへのリファレンスを取得します。

## Hyper-V Topology by Shell ジョブ

本項では、このジョブのトリガ・クエリおよびアダプタについて説明します。

トリガ・クエリ



ノード名	条件
RunningSoftware	DiscoveredProductName 等しい Microsoft Hyper-V Hypervisor
IpAddress	NOT IP Probe Name null
Shell	CI Type 等しい ntcmd OR CI Type 等しい uda

アダプタ

このジョブでは, **hyperv\_topology\_by\_shell** アダプタを使用します。

- 入力クエリ



• トリガ CI データ

名前	値
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

• 使用スクリプト

- hyperv.py
- hyperv\_topology\_by\_shell.py

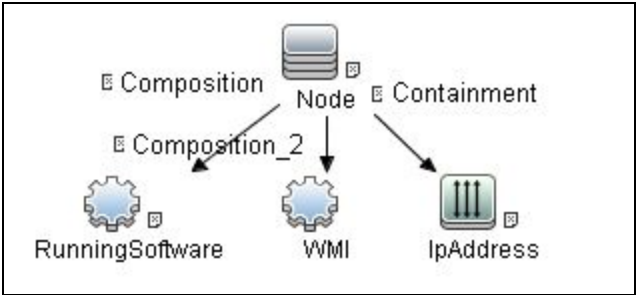
• 検出された CIT

- Composition
- ExecutionEnvinroment
- Hyper-V Partition Config
- Interface
- Layer2Connection
- Membership
- Node
- Switch
- Virtualization Layer Software

# Hyper-V Topology by WMI ジョブ

本 項 では、このジョブのトリガ・クエリおよびアダプタについて説明します。

トリガ・クエリ



ノード名	条件
RunningSoftware	DiscoveredProductName 等しい Microsoft Hyper-V Hypervisor
WMI	NOT Reference to the credentials dictionary entry null
IpAddress	NOT IP Probe Name null

アダプタ

このジョブでは、hyperv\_topology\_by\_wmi アダプタを使用します。

- 入カクエリ



- トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

- 使用スクリプト
  - hyperv.py
  - hyperv\_topology\_by\_wmi.py
- 検出された CIT
  - Composition
  - ExecutionEnvinroment
  - Hyper-V Partition Config
  - Interface
  - Layer2Connection
  - Membership
  - Node
  - Switch
  - Virtualization Layer Software

## 作成または変更されたエンティティ

エンティティ	新規または変更	エンティティ名
CIT	新規作成	Hyper-V Partition Config(hyperv_partition_config)
有効なリンク	新規作成	なし
ビュー	新規作成	Hyper-V Topology 9.0
使用スクリプト	新規作成	<ul style="list-style-type: none"> <li>• hyperv_topology_by_shell.py</li> <li>• hyperv_topology_by_wmi.py</li> <li>• hyperv.py</li> </ul>

エンティティ	新規または変更	エンティティ名
アダプタ	新規作成	<ul style="list-style-type: none"> <li>hyperv_topology_by_shell</li> <li>hyperv_topology_by_wmi</li> </ul>
ジョブ	新規作成	<ul style="list-style-type: none"> <li>Hyper-V Topology by Shell</li> <li>Hyper-V Topology by WMI</li> </ul>
トリガ・クエリ		<ul style="list-style-type: none"> <li>ntcmd_on_hyperv_host</li> <li>wmi_on_hyperv_host</li> </ul>

## トラブルシューティングおよび制限事項

オフラインの仮想マシンは、それらのマシンの MAC アドレスが使用できないために検出できません。

## 第7章: IBM Virtualization ディスカバリ

### 本章の内容

概要 .....	124
サポートされるバージョン .....	124
トポロジ .....	125
IBM 仮想化の検出方法 .....	131
IBM Virtualization by Shell ジョブ .....	133
アダプタ情報 .....	134
IBM LPAR and VIO Server Topology by Shell ジョブ .....	136
はじめに .....	136
アダプタ情報 .....	137
IBM PureFlex Topology by Shell ジョブ .....	140
IBM 仮想化コマンド .....	143
VIO サーバ側のコマンド .....	155
LPAR 側のコマンド .....	165
作成または変更されたエンティティ .....	166
トラブルシューティングおよび制限事項 .....	168

## 概要

本章では、IBM Virtualization ディスカバリ・パッケージの使用法と機能について説明します。このパッケージでは、データ・ソースとして IBM の仮想化マネージャのいずれかに基づいて、IBM Virtualization トポロジのディスカバリがサポートされます。仮想化マネージャには HMC(Hardware Management Console)、IVM(Integrity Virtual Machines)、FSM(Flex System Manager)があります。

IBM HMC は、パーティション分割されたシステム(LPAR または仮想化システムとも呼ばれる)または SMP システム(IBM System I または IBM System p シリーズなど)の構成および運用のための標準インタフェースを提供する目的で、IBM が開発したものです。

IBM IVM は簡単に使い、クライアントが複数のワークロードをポイント、クリックして 1 つの IBM Power System に統合できるブラウザ・ベースのツールです。

IBM FSM はシステム管理アプライアンスであり、データ・センタの効率性とコスト削減が促進されます。IBM FSM ではサーバ、ストレージ、ネットワークが事前に統合された仮想化管理環境を提供し、この環境は単一のインタフェースから簡単に管理できます。

## サポートされるバージョン

このディスカバリ・ソリューションでは、次をサポートします。

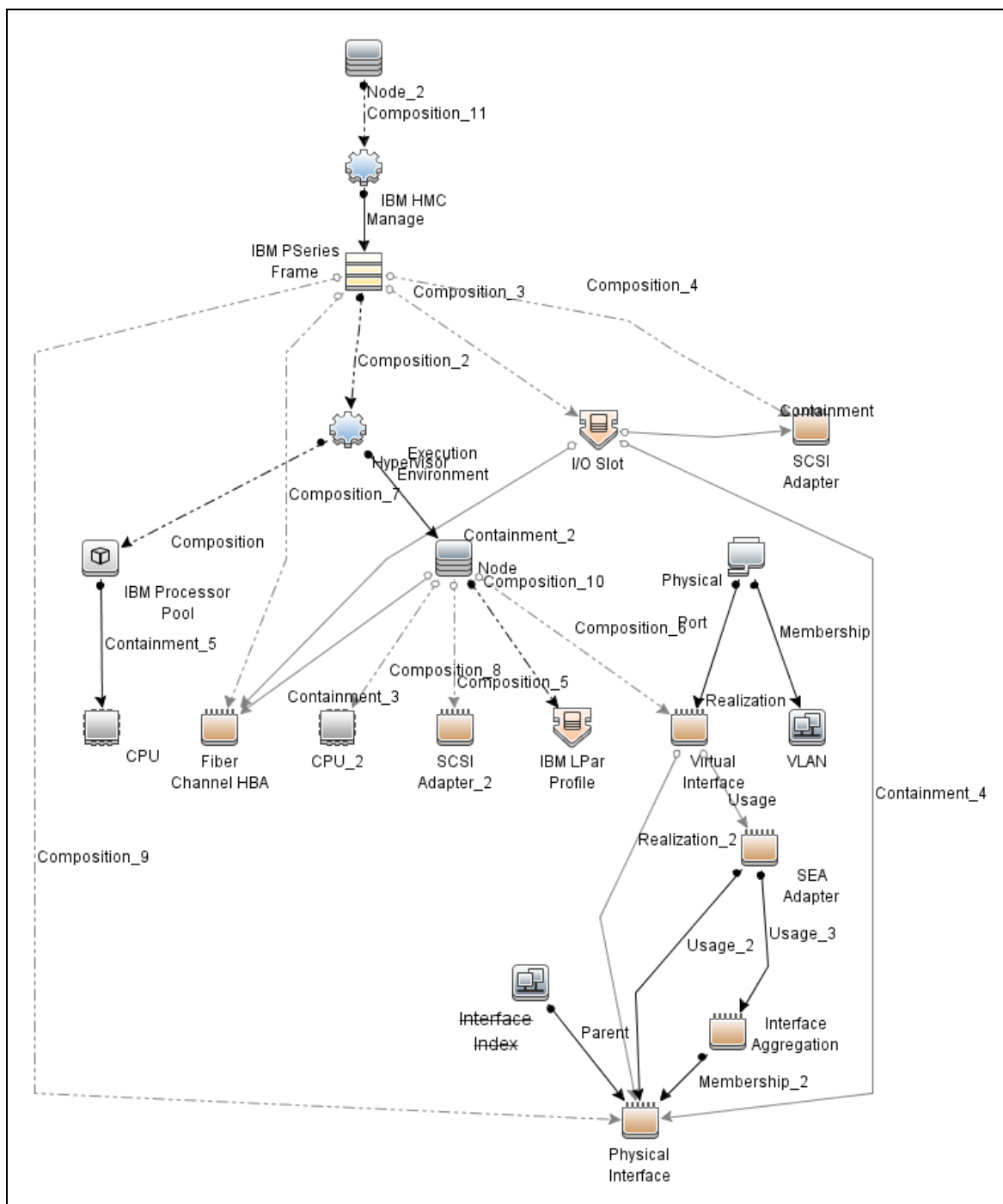
- AIX および Linux 上の IBM HMC バージョン 3.x, 5.x, 6.x, 7.x
- IBM FSM バージョン 1.x



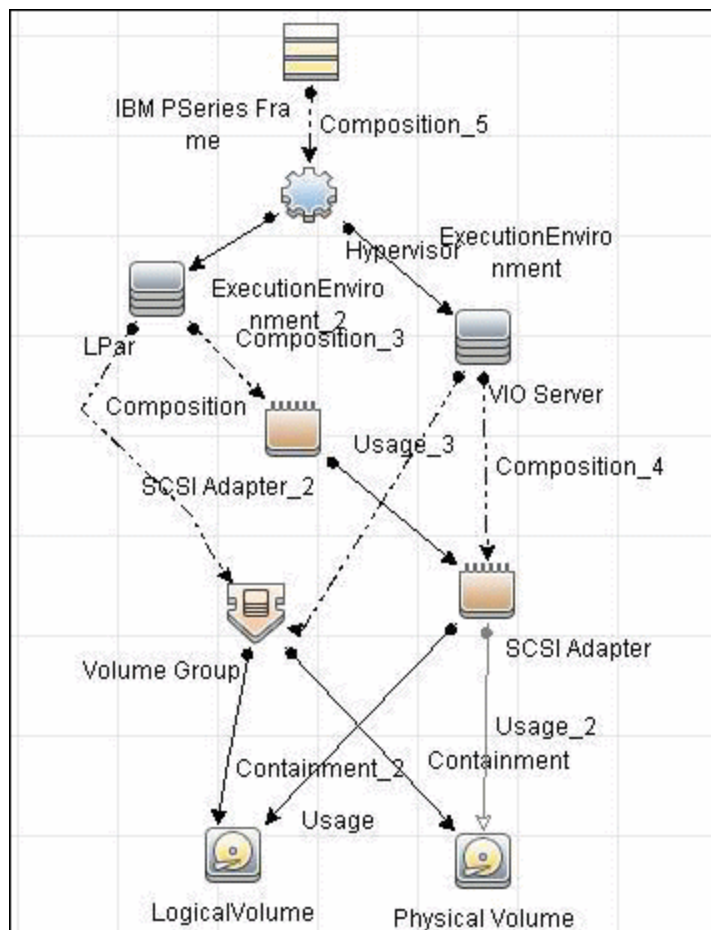
## トポロジ

注: 検出された CIT のリストについては, 「[検出された CIT](#)」([139ページ](#))を参照してください。

## IBM Virtualization by Shell トポロジ



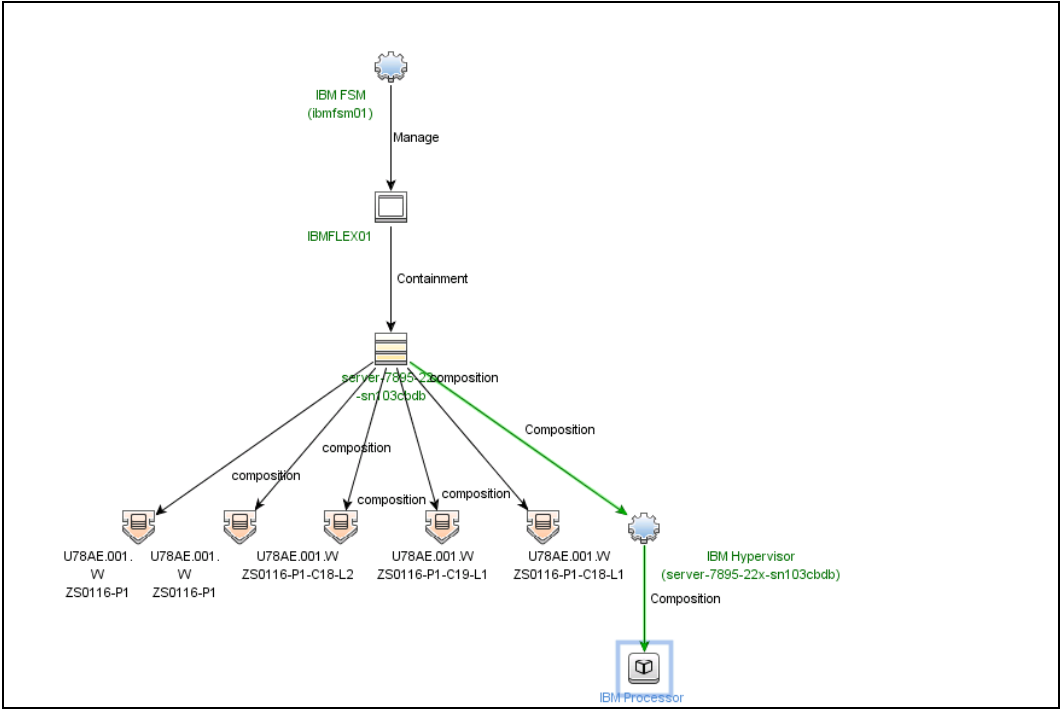
## IBM Storage Topology



## IBM IVM Topology



IBM PureFlex Topology by Shellトポロジ



## IBM 仮想化の検出方法

### 本項の内容

#### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、SSH プロトコルおよび Telnet シェル・プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

コマンドの一部が、ターゲット・ホスト上で **sudo** を使用して実行されるように構成されている場合、**[プロトコル パラメータ]** ダイアログ・ボックスで、次のフィールドを入力します。

- **Sudo パス** : sudo 実行可能ファイルへのフル・パスと実行可能ファイルの名前を合わせて入力します。実行可能ファイルがターゲット・オペレーティング・システム上の複数の場所に配置されている場合、複数のエントリを追加できます。

例 : sudo, /usr/bin/sudo, /bin/sudo

- **コマンドリスト** : **sudo** (プレフィックス) を付けるコマンドのリストを入力します。

例 : lspath, ifconfig

詳細については、『HP Universal CMDB Data Flow Management Guide』の「Protocol Parameter Dialog Box」を参照してください。

#### 2. 前提条件 - 権限の設定

ディスカバリをアクティブ化する前に、ディスカバリ・ユーザが次のコマンドを実行するためのすべての必要な権限を持っていることを確認します。

**注** : これらのコマンドの詳細については、次を参照してください。

- 「IBM 仮想化コマンド」(143ページ)
- 「VIO サーバ側のコマンド」(155ページ)
- 「LPAR 側のコマンド」(165ページ)

- hostname
- lscfg
- lsdev -dev <デバイス>
- lshmc -b
- lshmc -n
- lshmc -v
- lshmc -V
- lshwres -r io --subtype slot -m <pSeriesName>
- lshwres -r mem --level lpar -m <pSeriesName>
- lshwres -r mem --level sys -m <pSeriesName>
- lshwres -r proc --level lpar -m <pSeriesName>
- -r proc --level pool -m <pSeriesName>
- lshwres -r proc --level sys -m <pSeriesName>
- lshwres -r virtualio --subtype eth --level lpar -m <pSeriesName>
- lshwres -r virtualio --subtype scsi -m <pSeriesName>
- lsivm
- lsiv
- lsiv -v <論理ボリューム名>
- lsmapi -all
- lsmapi -all -net
- lspartition
- lspath



- lspv
- lssyscfg -r lpar -m <pSeriesName>
- lssyscfg -r prof -m <pSeriesName> --filter <lparName>
- lssyscfg -r sys
- lstcpip
- lsvg
- lsvg -l <ボリューム・グループ名>
- lsvio -e
- lsvio -s
- lvdisplay
- pvdisplay
- vgdisplay

### 3. ディスカバリの実行

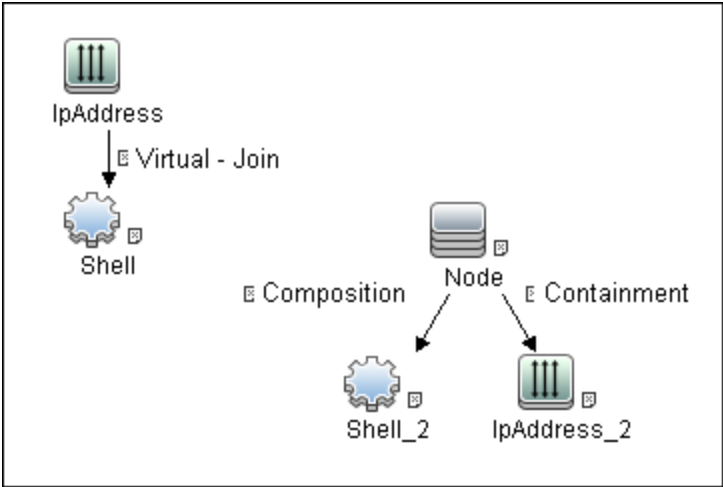
- a. **Range IPs by ICMP** ジョブを実行します。
- b. **Host Connection by Shell** ジョブを実行します。
- c. **IBM Virtualization by Shell** ジョブまたは **IBM PureFlex Topology by Shell** ジョブを実行します。
- d. **IBM LPAR and VIO Server Topology by Shell** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## IBM Virtualization by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ・クエリ



アダプタ情報

このジョブでは、IBM Virtualization by Shell アダプタを使用します。

アダプタ・タイプ

Jython

入力 CIT

IpAddress

入力クエリ



トリガ CI データ

名前	値
ip_address	\${SOURCE.name}

名前	値
ip_domain	\${SOURCE.routing_domain}

## 使用スクリプト

- ibm\_hmc\_lib.py
- storage\_topology.py
- ibm\_hmc\_discoverer.py
- ibm\_hmc\_by\_shell.py

## 検出された CIT

- Composition
- Containment
- Cpu
- ExecutionEnvironment
- I/O Slot
- IBM Frame
- IBM HMC
- IBM IVM
- IBM LPar Profile
- IBM Processor Pool
- Interface
- IpAddress
- Manage

- Membership
- Node
- PhysicalPort
- Realization
- SCSI Adapter
- Shell
- Usage
- Virtualization Layer Software
- Vlan

注: トポロジを表示するには, 「[IBM Virtualization by Shell トポロジ](#)」(126ページ)を参照してください。

アダプタ・パラメータ

パラメータ	標準設定値	説明
reportCPUs	true	CPU CI をレポートするかどうかを示す。
reportLparNameAsHostName	false	仮想ホスト名を LPAR 名としてレポートするかどうかを示す。

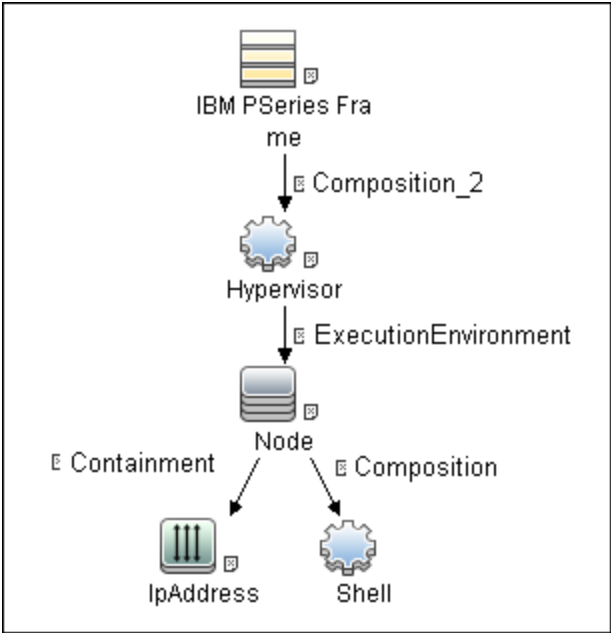
IBM LPAR and VIO Server Topology by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

はじめに

このジョブは, IBM VIO サーバまたは IBM LPAR の実行中のシステムを検出します。

トリガ・クエリ



アダプタ情報

このジョブでは, IBM LPar And VIO Server by Shell アダプタを使用します。

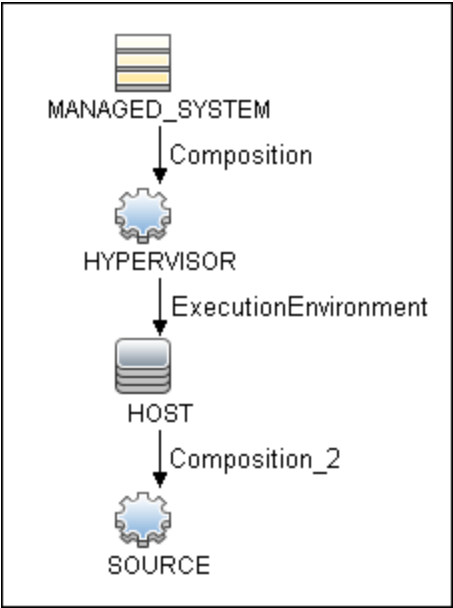
アダプタ・タイプ

Jython

入力 CIT

Shell

入力クエリ



トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${SOURCE.root_container}
ip_address	\${SOURCE.application_ip}
managedSystemId	\${MANAGED_SYSTEM.root_id}
osType	\${HOST.discovered_os_name}

使用スクリプト

- hostresource.py
- host\_win.py
- networking.py

- hostresource\_win.py
- networking\_win\_shell.py
- hostresource\_win\_wmi.py
- host\_win\_shell.py
- host\_discoverer.py
- host\_win\_wmi.py
- ibm\_hmc\_lib.py
- TTY\_HR\_CPU\_Lib.py
- storage\_topology.py
- ibm\_lpar\_or\_vio\_by\_shell.py

## 検出された CIT

- Composition
- Containment
- Dependency
- Fibre Channel HBA
- FileSystem
- I/O Slot
- Interface
- Interface Aggregation
- IpAddress
- LogicalVolume
- Membership

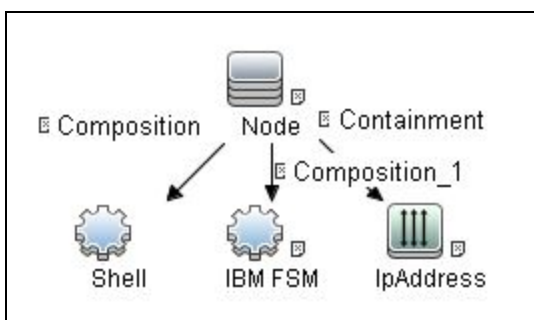
- Node
- Parent
- Physical Volume
- Realization
- SCSI Adapter
- SEA Adapter
- Usage
- Volume Group

注: トポロジを表示するには, [「トポロジ」\(125ページ\)](#)を参照してください。

## IBM PureFlex Topology by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

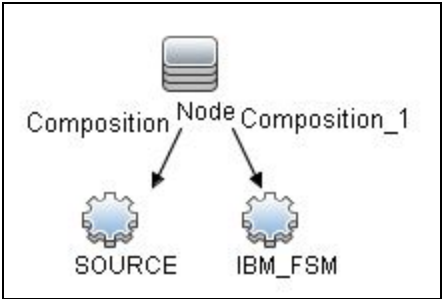


### アダプタ

このジョブでは, **PureFlex Discovery** アダプタを使用します。



• 入力クエリ



• トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
credentialsId	\${SOURCE.credentials_id}
fsmId	\${IBM_FSM.root_id}
hostId	\${SOURCE.root_container}
ip_address	\${SOURCE.application_ip}

• 使用スクリプト

- storage\_topology.py
- ibm\_hmc\_lib.py
- ibm\_fsm.py
- ibm\_fsm\_discoverer.py
- ibm\_fsm\_by\_shell.py

検出された CIT

- Chassis
- Composition

- Containment
- IBM FSM
- IBM Frame
- Interface
- IpAddress
- Management
- Node
- Realization
- Storage Array
- Switch

注: トポロジを表示するには, 「[IBM PureFlex Topology by Shell トポロジ](#)」([130ページ](#))を参照してください。

## IBM 仮想化コマンド

### 本項の内容

- [「lshmc -V」\(143ページ\)](#)
- [「lshmc -v」\(144ページ\)](#)
- [「lshmc -b」\(144ページ\)](#)
- [「lshmc -n」\(144ページ\)](#)
- [「lspartition -c <タイプ>\\_<バージョン> -i」\(145ページ\)](#)
- [「lssyscfg -r sys」\(146ページ\)](#)
- [「lshwres -r proc -level sys -m '<管理対象システム名>」\(147ページ\)](#)
- [「lshwres -r proc -level pool -m '<管理対象システム名>」\(148ページ\)](#)
- [「lssyscfg -r lpar -m '<管理対象システム名>」\(149ページ\)](#)
- [「lssyscfg -r prof -m '<管理対象システム名>」\(150ページ\)](#)
- [「lshwres -r virtualio -rsubtype eth -level lpar -m '<管理対象システム名>」\(151ページ\)](#)
- [「lshwres -r virtualio -rsubtype scsi -m '<管理対象システム名>」\(152ページ\)](#)
- [「lshwres -r proc -level lpar -m '<管理対象システム名>」\(153ページ\)](#)
- [「lshwres -r io -rsubtype slot -m '<管理対象システム名>」\(153ページ\)](#)

### lshmc -V

#### 出力

```
version= Version:7 Release:3.5.0 Service Pack:0 HMC Build level 20091201.1  
MH01195:Required fix for HMC V7R3.5.0 (10-16-2009) MH01197:Fix for HMC V7R3.5.0  
(11-12-2009) MH01204:Fix for HMC V7R3.5.0 (12-11-2009) ", "base_version=V7R3.5.0 "
```

#### マッピング

このコマンドの出力は、**IBM HMC CI**の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
Version	IBM HMC	Version_number
Base_version	IBM HMC	Application_version_description

## lshmc -v

### 出力

```
vpd=*FC ????????*VC 20.0 *N2 Tue Apr 27 13:05:33 CEST 2010 *FC ????????*DS Hardware
Management Console *TM eserver xSeries 335 -[XXXXCR2]- *SE XXXXXXX *MN IBM *PN
Unknown *SZ 1059495936 *OS Embedded Operating Systems *NA 192.168.1.10 *FC
????????*DS Platform Firmware *RM V7R3.5.0.0
```

### マッピング

このコマンドの出力は、**IBM HMC** CI の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
SE	IBM HMC	HMC Serial Number
TM	IBM HMC	HMC TYPE

## lshmc -b

### 出力

```
bios=T2E139AUS-1.15
```

### マッピング

このコマンドの出力は、**IBM HMC** CI の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
Bios	IBM HMC	HMC BIOS

## lshmc -n

### 出力

```
hostname=hmc01, domain=somedomain.com,
"ipaddr=192.168.1.10,0.0.0.0,192.168.128.1",
"networkmask=255.255.254.0,255.255.255.0,255.255.128.0",
```

```
gateway=192.168.1.1,nameserver=,domainsuffix=,
slipipaddr=192.168.1.1,slipnetmask=255.255.0.0,
"ipaddr_lpar=192.168.80.1,192.168.128.1",
"networkmask_lpar=255.255.254.0,255.255.128.0",
clients=,ipv6addr_lpar=,ipv4addr_eth0=192.168.1.10,
ipv4netmask_eth0=255.255.254.0,ipv4dhcp_eth0=off,ipv6addr_eth0=,
ipv6auto_eth0=off,ipv6privacy_eth0=off,ipv6dhcp_eth0=off,
lparcomm_eth0=off,jumboframe_eth0=off,speed_eth0=100,
duplex_eth0=full,tso_eth0=off,ipv4addr_eth1=0.0.0.0,
ipv4netmask_eth1=255.255.255.0,ipv4dhcp_eth1=off,
ipv6addr_eth1=,ipv6auto_eth1=off,ipv6privacy_
eth1=off,ipv6dhcp_eth1=off,lparcomm_eth1=off,jumboframe_
eth1=off,speed_eth1=auto,duplex_eth1=auto,tso_
eth1=off,ipv4addr_eth2=192.168.128.1,ipv4netmask_
eth2=255.255.128.0,ipv4dhcp_eth2=off,ipv6addr_
eth2=,ipv6auto_eth2=off,ipv6privacy_eth2=off,ipv6dhcp_
eth2=off,lparcomm_eth2=off,jumboframe_eth2=off,speed_
eth2=auto,duplex_eth2=auto,tso_eth2=off
```

### マッピング

このコマンドの出力は、特定の HMC マシン用の次のネットワーク情報の入力に使用されます。HMC が実行されているホストは、常に不完全なホストとしてレポートされます。これは、インタフェース MAC アドレスに関する情報が存在しないこと、および、この環境では標準 UNIX コマンドが動作しないためです。

CMD 出力属性	CI 名	CI 属性
constant AIX	Unix	Host Operating System
Hostname	Unix	Host Name
Hostname	Unix	Name
Domain	Unix	OS Domain Name
Ipv4addr_eth<0..N>	IpAddress	Ip Address

### lspartition -c <タイプ>\_<バージョン> -i

#### 出力

```
2,192.168.80.52,3;1,192.168.80.62,3;3,192.168.80.53,3
```

### マッピング

出力内の各ブロックは、セミコロン文字 (;) で区切られます。最初の値は LPAR ID であり、2 番目の値は LPAR IP アドレスです。LPAR の ID をほかのコマンドからの出力と照合することで、不完全なホストが作成され、割り当てられた LPAR プロファイル CI と一緒にレポートされます。

## Issyscfg -r sys

### 出力

```
name=XXXXXXXX-XXXX-XXX-XXXXXXXXXX-XX,type_model=XXXX-XXX, serial_
num=XXXXXX,ipaddr=192.168.1.10,state=Operating,sys_time=04/27/2010 12:55:23,power_
off_policy=1,active_lpar_mobility_capable=0,inactive_lpar_mobility_
capable=0,active_lpar_share_idle_procs_capable=0,active_mem_sharing_capable=0,bsr_
capable=0,cod_mem_capable=0,cod_proc_capable=1,electronic_err_reporting_
capable=0,firmware_power_saver_capable=0,hardware_power_saver_capable=0,hardware_
discovery_capable=0,addr_broadcast_perf_policy_capable=0,hca_capable=1,huge_page_
mem_capable=1,lhea_capable=0,lpar_avail_priority_capable=0,lpar_proc_compat_mode_
capable=0,micro_lpar_capable=1,os400_capable=0,5250_application_
capable=0,redundant_err_path_reporting_capable=1,shared_eth_failover_capable=1,sn_
msg_passing_capable=0,sp_failover_capable=1,vet_activation_capable=1,virtual_fc_
capable=0,virtual_io_server_capable=1,virtual_switch_capable=0,assign_5250_cpw_
percent=0,max_lpars=40,max_power_ctrl_lpars=1,hca_bandwidth_
capabilities=null,service_lpar_id=none,curr_sys_keylock=norm,pend_sys_
keylock=norm,curr_power_on_side=temp,pend_power_on_side=temp,curr_power_on_
speed=fast,pend_power_on_speed=fast,curr_power_on_speed_override=none,pend_power_
on_speed_override=none,power_on_type=power on,power_on_option=standby,power_on_
lpar_start_policy=userinit,pend_power_on_option=standby,pend_power_on_lpar_start_
policy=userinit,power_on_method=02,power_on_attr=0000,sp_boot_attr=0000,sp_boot_
major_type=08,sp_boot_minor_type=01,sp_version=00030030,mfg_default_config=0,curr_
mfg_default_ipl_source=a,pend_mfg_default_ipl_source=a,curr_mfg_default_boot_
mode=norm,pend_mfg_default_boot_mode=norm
```

### マッピング

検出された各 IBM Pseries Frame について、name 属性に IBM Hypervisor と設定された Hypervisor CI が作成されます。

このコマンドの出力は、**IBM PSeries Frame CI** の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
Name	IBM PSeries Frame	Name
serial_number	IBM PSeries Frame	Host Key
cod_proc_capable	IBM PSeries Frame	CPU Capacity on Demand Capable
cod_mem_capable	IBM PSeries Frame	Memory Capacity on Demand Capable
huge_page_mem_capable	IBM PSeries Frame	Huge Memory Page Capable
max_lpars	IBM PSeries Frame	Max LPARs

CMD 出力属性	CI 名	CI 属性
Status	IBM PSeries Frame	Frame State
micro_lpar_capable	IBM PSeries Frame	Micro LPAR Capable
service_lpar_id	IBM PSeries Frame	Service LPAR ID
service_lpar_name	IBM PSeries Frame	Service LPAR Name

## lshwres -r proc --level sys -m '<管理対象システム名>'

### 出力

configurable\_sys\_proc\_units=4.0,curr\_avail\_sys\_proc\_units=1.4, pend\_avail\_sys\_proc\_units=1.4,installed\_sys\_proc\_units=4.0, max\_capacity\_sys\_proc\_units=deprecated,deconfig\_sys\_proc\_units=0, min\_proc\_units\_per\_virtual\_proc=0.1,max\_virtual\_procs\_per\_lpar=64,max\_procs\_per\_lpar=4,max\_curr\_virtual\_procs\_per\_aixlinux\_lpar=64,max\_curr\_virtual\_procs\_per\_vios\_lpar=64, max\_curr\_virtual\_procs\_per\_os400\_lpar=64,max\_curr\_procs\_per\_aixlinux\_lpar=4, max\_curr\_procs\_per\_vios\_lpar=4,max\_curr\_procs\_per\_os400\_lpar=4, max\_shared\_proc\_pools=1

### マッピング

このコマンドの出力は、**IBM PSeries Frame** CI の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
min_proc_units_per_virtual_proc	IBM PSeries Frame	Min CPU Units per Virtual CPU
curr_avail_sys_proc_units	IBM PSeries Frame	Current Available CPU Units
max_shared_proc_pools	IBM PSeries Frame	Max Shared CPU Pools
configurable_sys_proc_units	IBM PSeries Frame	Configurable CPU Units
installed_sys_proc_units	IBM PSeries Frame	Installed CPU Units
pend_avail_sys_proc_units	IBM PSeries Frame	Pending Available CPU Units
max_procs_per_lpar	IBM PSeries Frame	Max CPUs per LPAR
max_virtual_procs_per_lpar	IBM PSeries Frame	Max Virtual CPUs per LPAR

## lshwres -r mem --level sys -m '<管理対象システム名>'

### 出力

```
configurable_sys_mem=32768,curr_avail_sys_mem=1344,pend_avail_sys_mem=1344,
installed_sys_mem=32768,max_capacity_sys_mem=deprecated,deconfig_sys_mem=0, sys_
firmware_mem=704,mem_region_size=64,configurable_num_sys_huge_pages=0, curr_avail_
num_sys_huge_pages=0,pend_avail_num_sys_huge_pages=0, max_num_sys_huge_
pages=1,requested_num_sys_huge_pages=0,huge_page_size=16384, max_mem_pools=0
```

### マッピング

このコマンドの出力は, **IBM PSeries Frame CI** の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
configurable_sys_mem	IBM PSeries Frame	Configurable System Memory
max_num_sys_huge_pages	IBM PSeries Frame	Max Number of Huge Pages
huge_page_size	IBM PSeries Frame	Huge Page Size
sys_firmware_mem	IBM PSeries Frame	Firmware Memory
mem_region_size	IBM PSeries Frame	Memory Region Size
curr_avail_sys_mem	IBM PSeries Frame	Current Available Memory
installed_sys_mem	IBM PSeries Frame	Installed Memory
requested_num_sys_huge_pages	IBM PSeries Frame	Requested Number of Huge Pages
pend_avail_sys_mem	IBM PSeries Frame	Pending Available Memory

## lshwres -r proc --level pool -m '<管理対象システム名>'

### 出力

```
configurable_pool_proc_units=4.0,curr_avail_pool_proc_units=1.4,pend_avail_pool_
proc_units=1.4
```

### マッピング

ユーザ定義プールがない場合, **pool\_id** パラメータは出力に表示されません(システムは標準設定で **pool\_id** をゼロとみなす)。

このコマンドの出力は, **IBM Processor Pool CI** の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
curr_avail_pool_proc_units	IBM Processor Pool	CPU Pool Available Physical CPUs



CMD 出力属性	CI 名	CI 属性
configurable_pool_proc_units	IBM Processor Pool	CPU Pool Configurable Physical CPUs
pend_avail_pool_proc_units	IBM Processor Pool	CPU Pool Pending Available Physical CPUs
pool_id	IBM Processor Pool	Name

### lssyscfg -r lpar -m '<管理対象システム名>'

#### 出力

```
name=somelparname1,lpar_id=5,lpar_env=aixlinux,state=Running,resource_config=1,os_
version=Unknown,logical_serial_num=65B922G5,default_
profile=somedefaultprofilename1,curr_profile=somelparprofilename1,work_group_
id=none,shared_proc_pool_util_auth=1,allow_perf_collection=1,power_ctrl_lpar_
ids=none,boot_mode=sms,lpar_keylock=norm,auto_start=0,redundant_err_path_
reporting=0
```

#### マッピング

このコマンドの出力は、**IBM LPAR Profile** CI の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
logical_serial_num	IBM LPAR Profile	LPAR Serial Number
boot_mode	IBM LPAR Profile	LPAR Profile Boot Mode
auto_start	IBM LPAR Profile	LPAR Profile Auto Start
work_group_id	IBM LPAR Profile	LPAR Profile Workgroup ID
default_profile	IBM LPAR Profile	LPAR default profile name
curr_profile	IBM LPAR Profile	LPAR profile name
power_ctrl_lpar_ids	IBM LPAR Profile	LPAR power control ids
State	IBM LPAR Profile	Lpar state
lpar_env	IBM LPAR Profile	Lpar type
lpar_id	IBM LPAR Profile	LPAR ID
Name	IBM LPAR Profile	LPAR Name

**Issyscfg -r prof -m '<管理対象システム名>'****出力**

```

name=name1,lpar_name=name2,lpar_id=5,lpar_env=aixlinux,
all_resources=0,min_mem=4096,desired_mem=8192,max_mem=8192,
min_num_huge_pages=0,desired_num_huge_pages=0,
max_num_huge_pages=0,proc_mode=shared,min_proc_units=0.3,
desired_proc_units=0.5,max_proc_units=1.0,min_procs=1,
desired_procs=2,max_procs=2,sharing_mode=uncap,
uncap_weight=128,io_slots=none,lpar_io_pool_ids=none,
max_virtual_slots=10,"virtual_serial_adapters=0/server/1/
any//any/1,1/server/1/any//any/1","virtual_scsi_adapters=5/
client/1/l11s12vio1/13/1,6/client/1/l11s12vio1/14/1,7/client
/1/l11s12vio1/15/1",virtual_eth_adapters=2/0/1//0/1,
hca_adapters=none,boot_mode=norm,conn_monitoring=1,auto_start=0,
power_ctrl_lpar_ids=none,work_group_id=none,redundant_err_path_reporting=0
name=name3,lpar_name=name4,lpar_id=4,lpar_env=aixlinux,all_resources=0,
min_mem=4096,desired_mem=10240,max_mem=10240,min_num_huge_pages=0,
desired_num_huge_pages=0,max_num_huge_pages=0,proc_mode=shared,
min_proc_units=0.3,desired_proc_units=0.7,max_proc_units=1.0,
min_procs=1,desired_procs=2,max_procs=2,sharing_mode=uncap,
uncap_weight=128,io_slots=none,lpar_io_pool_ids=none,
max_virtual_slots=10,"virtual_serial_adapters=0/server
/1/any//any/1,1/server/1/any//any/1",
"virtual_scsi_adapters=5/client/1/l11s12vio1/10/1,6/
client/1/l11s12vio1/11/1,7/client/1/l11s12vio1/12/1",
virtual_eth_adapters=2/0/2//0/1,hca_adapters=none,boot_mode=norm,
conn_monitoring=1,auto_start=0,power_ctrl_lpar_ids=none,
work_group_id=none,redundant_err_path_reporting=0

```

**マッピング**

このコマンドの出力は、**IBM LPAR Profile CI**の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
sharing_mode	IBM LPAR Profile	LPAR Profile Sharing Mode
proc_mode	IBM LPAR Profile	LPAR Profile CPU Mode
uncap_weight	IBM LPAR Profile	LPAR Profile Uncapped Weight
desired_num_huge_pages	IBM LPAR Profile	LPAR Profile Desired Number of Huge Memory Pages
min_num_huge_pages	IBM LPAR Profile	LPAR Profile Minimum Number of Huge Memory Pages

CMD 出力属性	CI 名	CI 属性
max_procs	IBM LPAR Profile	LPAR Profile Maximum Number of CPUs
desired_procs	IBM LPAR Profile	LPAR Profile Desired Number of CPUs
min_proc_units	IBM LPAR Profile	LPAR Profile Minimum Physical CPUs
max_mem	IBM LPAR Profile	LPAR Profile Maximum memory
conn_monitoring	IBM LPAR Profile	LPAR Profile Connection Monitoring Enabled
min_mem	IBM LPAR Profile	LPAR Profile Minimum Memory on this LPAR
max_virtual_slots	IBM LPAR Profile	LPAR Profile Maximum Number of Virtual Slots
redundant_err_path_reporting	IBM LPAR Profile	LPAR Profile Redundant Error Path Reporting
max_num_huge_pages	IBM LPAR Profile	LPAR Profile Maximum Number of Huge Memory Pages
min_procs	IBM LPAR Profile	LPAR Profile Minimum Number of CPUs
max_proc_units	IBM LPAR Profile	LPAR Profile Maximum Physical CPUs
io_slots	IBM LPAR Profile	LPAR Profile IO Slots
lpar_io_pool_ids	IBM LPAR Profile	LPAR Profile IO Pool IDs
desired_proc_units	IBM LPAR Profile	LPAR Profile Desired Physical CPUs
desired_mem	IBM LPAR Profile	LPAR Profile Memory Requested by this LPAR
virtual_serial_adapters	IBM LPAR Profile	LPAR Profile Virtual Serial Adapters

**lshwres -r virtualio --subtype eth --level lpar -m '<管理対象システム名>'**

#### 出力

```
lpar_name=name1,lpar_id=1,slot_num=2,state=1,is_required=1,is_trunk=1,trunk_
priority=1,ieee_virtual_eth=0,port_vlan_id=1,addl_vlan_ids=,mac_addr=765920001002
lpar_name=l11s12vio1,lpar_id=1,slot_num=3,state=1,is_required=1,is_trunk=1,trunk_
priority=1,ieee_virtual_eth=0,port_vlan_id=2,addl_vlan_ids=,mac_addr=765920001003
lpar_name=name2,lpar_id=2,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual_
eth=0, port_vlan_id=1,addl_vlan_ids=,mac_addr=765920002002
lpar_name=name3,lpar_id=3,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual_
eth=0, port_vlan_id=1,addl_vlan_ids=,mac_addr=765920003002
lpar_name=name4,lpar_id=4,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual_
```

```
eth=0, port_vlan_id=2,addl_vlan_ids=,mac_addr=765920004002
lpar_name=name5,lpar_id=5,slot_num=2,state=1,is_required=1,is_trunk=0,ieee_virtual_
eth=0, port_vlan_id=1,addl_vlan_ids=,mac_addr=765920005002
```

### マッピング

mac\_addr 属性は、先頭のゼロなしの 10 進数で表されます。この値は、CMDB で MAC アドレスの正しい表現になるように、足りない桁数がゼロ埋めされた右詰めの 16 進数に変換されます。

この MAC アドレスに基づいて仮想 NIC が作成され、対応する LPAR または VIO サーバに接続されます。これは **Lpar\_name** または **Lpar\_id** によって示されます。**Vlan CI** が **vlan\_id** または **addl\_vlan\_ids** に基づいて作成され、インタフェースのポートにリンクされます。VLAN 用のルート・コンテナは、固有の IBM Pseries Frame です(管理対象システム)。

CMD 出力属性	CI 名	CI 属性
port_vlan_id/addl_vlan_ids	VLAN	Vlan Number
IBM PSeries Frame CMDB ID	VLAN	Root Container
mac_addr(必要に応じて標準化される場合は 16 進数に変換)	Interface	MAC Address

## lshwres -r virtualio --subtype scsi -m '<管理対象システム名>'

### 出力

```
lpar_name=vioname1,lpar_id=1,slot_num=15,state=1,is_required=0,adapter_
type=server,remote_lpar_id=5,remote_lpar_name=lparname1,remote_slot_num=7
lpar_name=vioname1,lpar_id=1,slot_num=14,state=1,is_required=0,adapter_
type=server,remote_lpar_id=5,remote_lpar_name=lparname2,remote_slot_num=6
lpar_name=vioname1,lpar_id=1,slot_num=13,state=1,is_required=0,adapter_
type=server,remote_lpar_id=5,remote_lpar_name=lparname2,remote_slot_num=5
```

### マッピング

lpar\_name 属性と lpar\_id 属性は、常に LPAR への仮想 SCSI を作成して付与する VIO サーバの名前と ID です。LPAR 上の SCSI アダプタは、スロット番号とそのスロットが属する LPAR の名前によって識別されます。

CMD 出力属性	CI 名	CI 属性
Slot_num/remote_slot_num	SCSI	Slot Number
<lpar_name> または <リモート LPAR 名> が付加されたホスト ID	SCSI	Root Container

**lshwres -r proc --level lpar -m '<管理対象システム名>'****出力**

```
lpar_name=name1,lpar_id=5,curr_shared_proc_pool_id=0,curr_proc_mode=shared,curr_min_proc_units=0.3,curr_proc_units=0.5,curr_max_proc_units=1.0,curr_min_procs=1,curr_procs=2,curr_max_procs=2,curr_sharing_mode=uncap,curr_uncap_weight=128,pend_shared_proc_pool_id=0,pend_proc_mode=shared,pend_min_proc_units=0.3,pend_proc_units=0.5,pend_max_proc_units=1.0,pend_min_procs=1,pend_procs=2,pend_max_procs=2,pend_sharing_mode=uncap,pend_uncap_weight=128,run_proc_units=0.5,run_procs=2,run_uncap_weight=128
```

**マッピング**

"lpar\_name"/"lpar\_id" を出力からの "curr\_shared\_proc\_pool\_id" と組み合わせて使用することで、LPAR が使用する特定の共有プロセッサ・プール("IBM Processor Pool")への対応するリンクを作成できます。専用("ded")CPU の場合、予備のプロセッサへのリンクを作成します。

**lshwres -r io --subtype slot -m '<管理対象システム名>'****出力**

```
unit_phys_loc=XXXXX.XXX.XXXXXXX,bus_id=2,phys_loc=C3,drc_index=21010002,lpar_name=name1,lpar_id=1,slot_io_pool_id=none,description=RAID Controller,feature_codes=none,pci_vendor_id=1069,pci_device_id=B166,pci_subs_vendor_id=1014,pci_subs_device_id=0278,pci_class=0104,pci_revision_id=04,bus_grouping=0,iop=0,parent_slot_drc_index=none,drc_name=XXXXX.XXX.XXXXXXX-XX-XX
```

**マッピング**

このコマンドの出力は、**I/O Slot CI** の作成に使用されます。LPAR の名前とIDを使用することで、デイスカバリはスロットを使用している特定の LPAR への関係を作成します。

CMD 出力属性	CI 名	CI 属性
Description	I/O Slot	Name of the Slot
bus_id	I/O Slot	Slot Bus ID
phys_loc	I/O Slot	Slot Physical Location on Bus
pci_revision_id	I/O Slot	Slot PCI Revision ID
bus_grouping	I/O Slot	Slot Bus Grouping
pci_device_id	I/O Slot	Slot PCI Device ID

CMD 出力属性	CI 名	CI 属性
unit_phys_loc	I/O Slot	Slot Physical Location
parent_slot_drc_index	I/O Slot	Slot Parent Slot DRC Index
drc_index	I/O Slot	Slot DRC Index
pci_subs_vendor_id	I/O Slot	Slot PCI Subslot Vendor ID
pci_class	I/O Slot	Slot PCI Class
slot_io_pool_id	I/O Slot	Slot IO Pool ID
pci_vendor_id	I/O Slot	Slot PCI Vendor ID
drc_name	I/O Slot	Slot DRC Name
feature_codes	I/O Slot	Slot Feature Codes
pci_subs_device_id	I/O Slot	Slot PCI Subslot Device ID

## VIO サーバ側のコマンド

### 本項の内容

- 「`/usr/ios/cli/ioscli lsdev -dev 'ent*' -field name physloc -fmt`」(155ページ)
- 「`ioscli entstat -all '<インタフェース名>' | grep -E "ETHERNET STATISTICS|Device Type|Hardware Address`」(156ページ)
- 「`ioscli lsdev -dev '<インタフェース名>' -attr`」(157ページ)
- 「`ioscli lsmmap -all -net`」(157ページ)
- 「`ioscli lsdev -dev fcs* -field name physloc description -fmt`」(158ページ)
- 「`ioscli lsdev | grep proc`」(159ページ)
- 「`ioscli lsdev -dev sysplanar0 -vpd | grep PROC`」(159ページ)
- 「`ioscli lsdev -dev proc<インデックス> -attr`」(159ページ)
- 「`lspv`」(160ページ)
- 「`lsvg`」(160ページ)
- 「`lsvg <ボリューム・グループ名>`」(160ページ)
- 「`lsvg -lv <ボリューム・グループ名>`」(161ページ)
- 「`lsvg -pv <論理ボリューム・グループ>`」(162ページ)
- 「`lslv <論理ボリューム名>`」(162ページ)
- 「`ioscli lsmmap -all`」(163ページ)

**`/usr/ios/cli/ioscli lsdev -dev 'ent*' -field name physloc -fmt`**

### 出力

```
ent0:U100C.001.DQDE777-P1-C4-T1
ent1:U100C.001.DQDE777-P1-C4-T2
ent2:U100C.001.DQDE777-P1-C4-T3
ent16:
```

ent17:

ent18:

ent19:

ent20:

## マッピング

特定のインタフェースのインタフェース名および物理的な場所が、このコマンドの出力です。出力は行ごとにコロン文字(:)で区切られ、最初の部分はインタフェース名、最後の部分は物理的な場所です。物理的な場所は常にあるわけではありません。たとえば、SEA やリンク・アグリゲーション・インタフェースの場合は設定されていません。物理的な場所の値は、物理 NIC から I/O スロットへのリンクの作成に使用されます。

## ioscli entstat -all '<インタフェース名>' | grep -E "ETHERNET STATISTICS|Device Type|Hardware Address

例 : ioscli entstat -all 'ent16' | grep -E "ETHERNET STATISTICS|Device Type|Hardware Address

## 出力

```
ETHERNET STATISTICS (ent16) :
Device Type:Shared Ethernet Adapter
Hardware Address:00:1B:64:91:74:55
ETHERNET STATISTICS (ent14) :
Device Type:EtherChannel
Hardware Address:00:1B:64:91:74:55
ETHERNET STATISTICS (ent0) :
Device Type:2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address:00:1a:64:91:74:44
ETHERNET STATISTICS (ent2) :
Device Type:2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
Hardware Address:00:1B:64:91:74:55
ETHERNET STATISTICS (ent4) :
Device Type:Virtual I/O Ethernet Adapter (1-lan)
Hardware Address:46:61:fa:d4:bf:0b
```

## マッピング

UCMDB バージョン 8.0x : 単一のマシンに、同じ MAC の2つのインタフェースを持たせることはできません。この場合、最初のインタフェースの MAC アドレス属性はその MAC アドレスの値のみを取得し、2番目のインタフェースにはアンダースコア(\_)とインタフェース・インデックスが付加されます。たとえば、上記の出力のインタフェース **ent0** は、**00:1B:64:91:74:55** に設定された MAC アドレスでレポートされ、インタフェース **ent2** は、**00:1B:64:91:74:55\_2** に設定された MAC アドレスでレポートされます。

UCMDB バージョン 9.0x : この制限には該当しないため、トポロジはそのままレポートされます。



CMD 出力属性	CI 名	CI 属性
ETHERNET STATISTICS 行	Interface	Name
Hardware Address	Interface	Mac Address
Device Type	Interface	Description
Device Type の値が EtherChannel の場合は ETHERNET STATISTICS 行	Interface Aggregation	Name
Device Type の値が Shared Ethernet Adapter の場合は ETHERNET STATISTICS 行	IBM SEA	Name

## ioscli lsdev -dev '<インタフェース名>' -attr

例 : ioscli lsdev -dev 'ent16' -attr

### 出力

```
attribute value description user_settable
adapter_names ent0,ent4 EtherChannel Adapters True
alt_addr 0x00000000000000 Alternate EtherChannel Address True
auto_recovery yes Enable automatic recovery after failover True
backup_adapter NONE Adapter used when whole channel fails True
hash_mode default Determines how outgoing adapter is chosen True
mode standard EtherChannel mode of operation True
netaddr 0 Address to ping True
noloss_failover yes Enable lossless failover after ping failure True
num_retries 3 Times to retry ping before failing True retry_time 1 Wait time (in
seconds) between pings True
use_alt_addr no Enable Alternate EtherChannel Address True
use_jumbo_frame no Enable Gigabit Ethernet Jumbo Frames True
```

### マッピング

adapter\_names 属性値は、バックアップ・デバイスへのリンクの作成に使用されます。

Media Speed の値は、二重および接続速度の両方を表します。

CMD 出力属性	CI 名	CI 属性
media_speed	Interface Index	Speed

## ioscli lsmmap -all -net

### 出力

SVEA Physloc

-----

ent4 U1000.E4A.06FB0D1-V1-C11-T1

SEA ent16

Backing device ent14

Status Available

Physloc

SVEA Physloc

-----

ent9 U1000.E4A.06FB0D1-V1-C16-T1

SEA ent21

Backing device ent12

Status Available

Physloc U1000.001.DQD3693-P1-C7-T3

### マッピング

このコマンドは、インタフェース間の関係を判断し、それらのインタフェースのタイプを識別するために使用されます。

CMD 出力属性	CI 名	CI 属性
SEA	SEA Adapter	Name
Backing Device	Link Aggregation / Interface	Name
SVEA	Interface(仮想)	Name

### ioscli lsdev -dev fcs\* -field name physloc description -fmt

#### 出力

fcs0:U1000.001.DQDE996-P1-C1-T1:4Gb FC PCI Express Adapter (df1000fe)

fcs1:U1000.001.DQDE996-P1-C1-T2:4Gb FC PCI Express Adapter (df1000fe)

fcs2:U1000.001.DQDE996-P1-C2-T1:4Gb FC PCI Express Adapter (df1000fe)

fcs3:U1000.001.DQDE996-P1-C2-T2:4Gb FC PCI Express Adapter (df1000fe)

### マッピング

このコマンドの出力は、VIO サーバ上の Fibre Channel Host アダプタを表します。この出力では、FC 名と FC の物理パスが取得され、これらは PFrame 上の I/O スロットへのリンクと FC インタフェースの説明の作成に使用されます。

CMD 出力属性	CI 名	CI 属性
最初のトークン	Fibre Channel HBA	Name
3 番目のトークン	Fibre Channel HBA	Description

## ioscli lsdev | grep proc

### 出力

```
proc0    Available    Processor
proc2    Available    Processor
proc4    Available    Processor
proc6    Available    Processor
```

### マッピング

このコマンドの出力は、検出された CPU インデックスを示します。ここでは 0, 2, 4, 6 です。

## ioscli lsdev -dev sysplanar0 -vpd | grep PROC

### 出力

```
2-WAY PROC CUOD :
2-WAY PROC CUOD :
2-WAY PROC CUOD :
2-WAY PROC CUOD :
```

### マッピング

このコマンドの出力には、検出されたコア数が示されます。出力内の "2" は、CPU コア数が 2 であることを示します。

## ioscli lsdev -dev proc<インデックス> -attr

### 出力

```
attribute value description user_settable
frequency 1654344000 Processor Speed False
smt_enabled true Processor SMT enabled False
smt_threads 2 Processor SMT threads False
state enable Processor state False
type PowerPC_POWER5 Processor type False
```

### マッピング

CMD 出力属性	CI 名	CI 属性
frequency	CPU	speed
type	CPU	model

## lspv

### 出力

```
NAME PVID VG STATUS
hdisk0 001fb2d15d794e0d rootvg active
hdisk1 001fb2d18f1f7f0c clientvg active
```

### マッピング

このコマンドでは、物理ボリュームとボリューム・グループ間の関係を取得し、ボリューム・グループから物理ボリュームへのリンクが作成されます。

CMD 出力属性	CI 名	CI 属性
VG	Physical Volume	Name
VG	Fibre Channel HBA	Name

## lsvg

### 出力

```
rootvg clientvg
```

### マッピング

このコマンドでは、VIO サーバ上に存在するすべてのボリューム・グループのリストを取得します。

## lsvg <ボリューム・グループ名>

### 出力

```
VOLUME GROUP: rootvg
VG IDENTIFIER:001fb2d10005d90000000011a5d795185
VG STATE: active
PP SIZE:256 megabyte(s)
VG PERMISSION: read/write
TOTAL PPs:520 (133120 megabytes)
MAX LVs:256
FREE PPs:372 (95232 megabytes)
```

```

LVs:13
USED PPs:148 (37888 megabytes)
OPEN LVs:11
QUORUM:2 (Enabled)
TOTAL PVs:1
VG DESCRIPTORS:2
STALE PVs:0
STALE PPs:0
ACTIVE PVs:1
AUTO ON: yes
MAX PPs per VG:32512
MAX PPs per PV:1016
MAX PVs:32
LTG size (Dynamic):256 kilobyte(s)
AUTO SYNC: no
HOT SPARE: no
BB POLICY: relocatable
    
```

### マッピング

このコマンドでは、ボリューム・グループ CI 属性の値を取得します。

CMD 出力属性	CI 名	CI 属性
VOLUME GROUP	Volume Group	Name
STATE	Volume Group	Volume Group State
VG IDENTIFIER	Volume Group	Volume Group ID

### lsvg -lv <ボリューム・グループ名>

#### 出力

```

rootvg:
LV NAME TYPE LPS PPs PVs LV STATE MOUNT POINT
hd5 boot 1 1 1 closed/syncd N/A
hd6 paging 2 2 1 open/syncd N/A
paging00 paging 4 4 1 open/syncd N/A
hd8 jfs2log 1 1 1 open/syncd N/A
hd4 jfs2 1 1 1 open/syncd /
hd2 jfs2 10 10 1 open/syncd /usr
hd9var jfs2 3 3 1 open/syncd /var
hd3 jfs2 10 10 1 open/syncd /tmp
hd1 jfs2 40 40 1 open/syncd /home
hd10opt jfs2 4 4 1 open/syncd /opt
lg_dump1v sysdump 4 4 1 open/syncd N/A
    
```

```
VMLib_LV jfs2 56 56 1 open/syncd /var/vio/VMLib
Ilv jfs2 12 12 1 closed/syncd /export/lbm
```

### マッピング

このコマンドでは、特定の論理ボリューム・グループに属するすべての論理ボリュームのリストと、マウント・ポイントがある場合はマウント・ポイントのリストを取得します。この情報によって、ボリューム・グループから論理ボリュームへのリンクの作成が可能になります。

CMD 出力属性	CI 名	CI 属性
LV Name	Logical Volume	Name
Mount Point	Disk(FS)	Name
Type	Disk	Type

### lsvg -pv <論理ボリューム・グループ>

#### 出力

```
rootvg:
PV_NAME PV STATE TOTAL PPs FREE PPs FREE DISTRIBUTION
hdisk0 active 520 372 103..30..31..104..104
```

### マッピング

このコマンドでは、ボリューム・グループ内の物理ボリュームのリストを取得します。この情報によって、物理ボリュームとボリューム・グループとの間のリンクの作成が可能になります。

### lslv <論理ボリューム名>

#### 出力

```
LOGICAL VOLUME: lv1
VOLUME GROUP: clientvg
LV IDENTIFIER:000fb1d10230d9000000011b8f1f8187.1
PERMISSION: read/write
VG STATE: active/complete
LV STATE: opened/syncd
TYPE: jfs
WRITE VERIFY: off
MAX LPS:32512
PP SIZE:512 megabyte(s)
COPIES:1
SCHED POLICY: parallel
LPs:70
```

```
PPs:70
STALE PPs:0
BB POLICY: non-relocatable
INTER-POLICY: minimum
RELOCATABLE: yes
INTRA-POLICY: middle
UPPER BOUND:1024
MOUNT POINT:N/A
LABEL:None
MIRROR WRITE
CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:NO
DEVICESUBTYPE :DS_LVZ
```

### マッピング

このコマンドでは、論理ボリューム・パラメータに関する情報を取得します。この情報は、論理ボリューム CI の属性にマッピングされます。

CMD 出力属性	CI 名	CI 属性
LOGICAL VOLUME	Logical Volume	Name
LV IDENTIFIER	Logical Volume	Logical Volume ID
LV STATE	Logical Volume	Logical Volume Status
Type	Logical Volume	Logical Volume File System Type

### ioscli lsmmap -all

#### 出力

```
SVSA Physloc Client Partition ID
-----
vhost0 U1000.E4A.06FB0D1-V1-C21 0x00000002
VTD vtopt0
Status Available
LUN 0x8100000000000000
Backing device /var/vio/VMLib/bootcd_rh5
Physloc
SVSA Physloc Client Partition ID
-----
vhost3 U1000.E4A.06FB0D1-V1-C31 0x00000002
VTD vtscsi0
Status Available
LUN 0x8100000000000000
```

```
Backing device os_ lv1
Physloc
VTD vtscsi1
Status Available
LUN 0x8200000000000000
Backing device p01_lv1
Physloc
VTD vtscsi8
Status Available
LUN 0x8300000000000000
Backing device p01_lv2
Physloc
```

マッピング

このコマンドでは、vSCSI から厳密なバックিং・デバイスへの関係を取得します。これは通常 ボリュームまたはボリューム・グループです。

CMD 出力属性	CI 名	CI 属性
SVSA	SCSI	Name
C<番号>	SCSI	Slot Number
Backing Device	LV / PV / FS	Name



## LPAR 側のコマンド

本項の内容

### lscfg

出力

```
INSTALLED RESOURCE LISTThe following resources are
installed on the machine.+/- = Added or deleted from
Resource List.*= Diagnostic support not available.
Model Architecture: chrp
Model Implementation:Multiple Processor, PCI bus + sys0
System Object+ sysplanar0 System Planar* vio0
Virtual I/O Bus* vsa0 U1000.505.062136A-V1-C0
LPAR Virtual Serial Adapter* vty0 U1000.505.062136A-V1-C0-L0
Asynchronous Terminal* pci2 U1000.001.AAA0757-P1
PCI Bus* pci1 U1000.001.AAA0757-P1
PCI Bus* pci0 U1000.001.AAA0757-P1
PCI Bus* pci3 U1000.001.AAA0757-P1
PCI Bus+ ent0 U1000.001.AAA0757-P1-T1
2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)+ ent1
U1000.001.AAA0757-P1-T2
2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)* pci4
U1000.001.AAA0757-P1
PCI Bus+ usbhc0 U1000.001.AAA0757-P1
USB Host Controller (33103500)+ usbhc1 U1000.001.AAA0757-P1
USB Host Controller (33103500)* pci5 U1000.001.AAA0757-P1
PCI Bus* ide0 U1000.001.AAA0757-P1-T10
ATA/IDE Controller Device+ cd0 U1000.001.AAA0757-P1-D3
IDE DVD-ROM Drive* pci6 U1000.001.AAA0757-P1
PCI Bus+ sisscsia0 U1000.001.AAA0757-P1
PCI-X Dual Channel Ultra320
SCSI Adapter+ scsi0 U1000.001.AAA0757-P1-T5
PCI-X Dual Channel Ultra320
SCSI Adapter bus+ scsi1 U1000.001.AAA0757-P1-T9
PCI-X Dual Channel Ultra320
SCSI Adapter bus+ hdisk0 U1000.001.AAA0757-P1-T9-L5-L0 16 Bit LVD
SCSI Disk Drive (146800 MB)+ hdisk1 U1000.001.AAA0757-P1-T9-L8-L0
16 Bit LVD
SCSI Disk Drive (146800 MB)+
ses0 U1000.001.AAA0757-P1-T9-L15-L0
SCSI Enclosure Services Device+
L2cache0 L2 Cache+ mem0 Memory+ proc0 Processor
```

## 作成または変更されたエンティティ

エンティティ名	エンティティ・タイプ	エンティティの説明
IBM HMC	CI タイプ	HMC ソフトウェア
IBM LPar Profile	CI タイプ	LPAR 構成
IBM Processor Pool	CI タイプ	共有プロセッサ・プール
IBM PSeries Frame	CI タイプ	PSeries フレーム / 管理対象システム
Interface Aggregation	CI タイプ	リンク・アグリゲーション
I/O Slot	CI タイプ	フレーム上の I/O スロット
SEA Adapter	CI タイプ	VIO サーバ上の仮想イーサネット・インタフェース
IBM Processor Pool > containment > CPU	有効なリンク	
I/O Slot > containment > ファイバ・チャネル HBA	有効なリンク	
I/O Slot > containment > ネットワーク・インタフェース	有効なリンク	
I/O Slot > containment > SCSI アダプタ	有効なリンク	
IBM HMC > manage > IBM PSeries Frame	有効なリンク	
Interface Aggregation > membership > ネットワーク・インタフェース	有効なリンク	
ネットワーク・インタフェース > realization > ネットワーク・インタフェース	有効なリンク	
ネットワーク・インタフェース > usage > SEA Adapter	有効なリンク	
SEA Adapter > usage > ネットワーク・インタフェース	有効なリンク	
IBM Virtualization by Shell	ジョブ	仮想化ベースのディスカバリを実行

エンティティ名	エンティティ・タイプ	エンティティの説明
IBM LPAR and VIO Server Topology by Shell	ジョブ	LPAR および VIO サーバ側のディスカバリを実行
IBM_VIRTUALIZATION_BY_SHELL_PATTERN	アダプタ	IBM Virtualization by Shell ジョブ用のアダプタ
IBM LPar And VIO Server by Shell	アダプタ	IBM LPAR and VIO Server Topology by Shell ジョブ用のアダプタ
ibm_hmc_by_shell	スクリプト	汎用の HMC 側 ディスカバリ・スクリプト
ibm_hmc_lib	スクリプト	両方の新しいジョブ用の共通データ・オブジェクトおよび手順
ibm_lpar_or_vio_by_shell	スクリプト	汎用 VIO サーバおよび LPAR ディスカバリ・スクリプト
ibm_hmc_by_shell.xml	クエリ	IBM Virtualization by Shell ジョブ用のトリガ・クエリ
ibm_lpar_or_vio_trigger_tql.xml	クエリ	IBM LPAR and VIO Server Topology by Shell ジョブ用のトリガ・クエリ
IBM HMC Topology.xml	クエリ	IBM HMC Topology ビュー用のクエリ(TQL)
IBM Storage Topology.xml	クエリ	IBM Storage Topology ビュー用のクエリ(TQL)
IBM HMC Topology.xml	ビュー	
IBM Storage Topology.xml	ビュー	
lpar_boot_mode	タイプ	サポートされている起動モード
lpar_cpu_mode	タイプ	CPU 共有モード
lpar_sharing_mode	タイプ	LPAR キャップ / アンキャップ共有モード
lpar_state	タイプ	可能性のある LPAR 状態
lpar_type	タイプ	可能性のある LPAR タイプ

## トラブルシューティングおよび制限事項

本項では、IBM Virtualization ディスカバリのトラブルシューティングと制限事項について説明します。

- LPAR から PFrame へのパーティション移行を構成できます。これは、Power Series v6 でのみサポートされており、IBM Virtualization では現在サポートされていません。
- Linux OS 上の VIO サーバはサポートされません。

## 第8章: Oracle VM Server for SPARC Technology ディスカバリ

概要 .....	170
サポートされるバージョン .....	170
トポロジ .....	171
Oracle VM Server for SPARC テクノロジの検出方法 .....	173
Oracle VM Server for SPARC Technology by Shell アダプタ .....	174
Oracle VM Server for SPARC Technology by Shell ジョブ .....	177
ディスカバリ・フロー .....	178
全般 .....	178
Oracle VM Server for SPARC Technology by Shell ジョブのフロー .....	178
コマンド .....	179
トラブルシューティングおよび制限事項 .....	183

## 概要

Oracle VM Server for SPARC Technology ディスカバリでは、Oracle LDOM(Logical Domains)または Oracle VM Server for SPARC テクノロジを検出できます。

## サポートされるバージョン

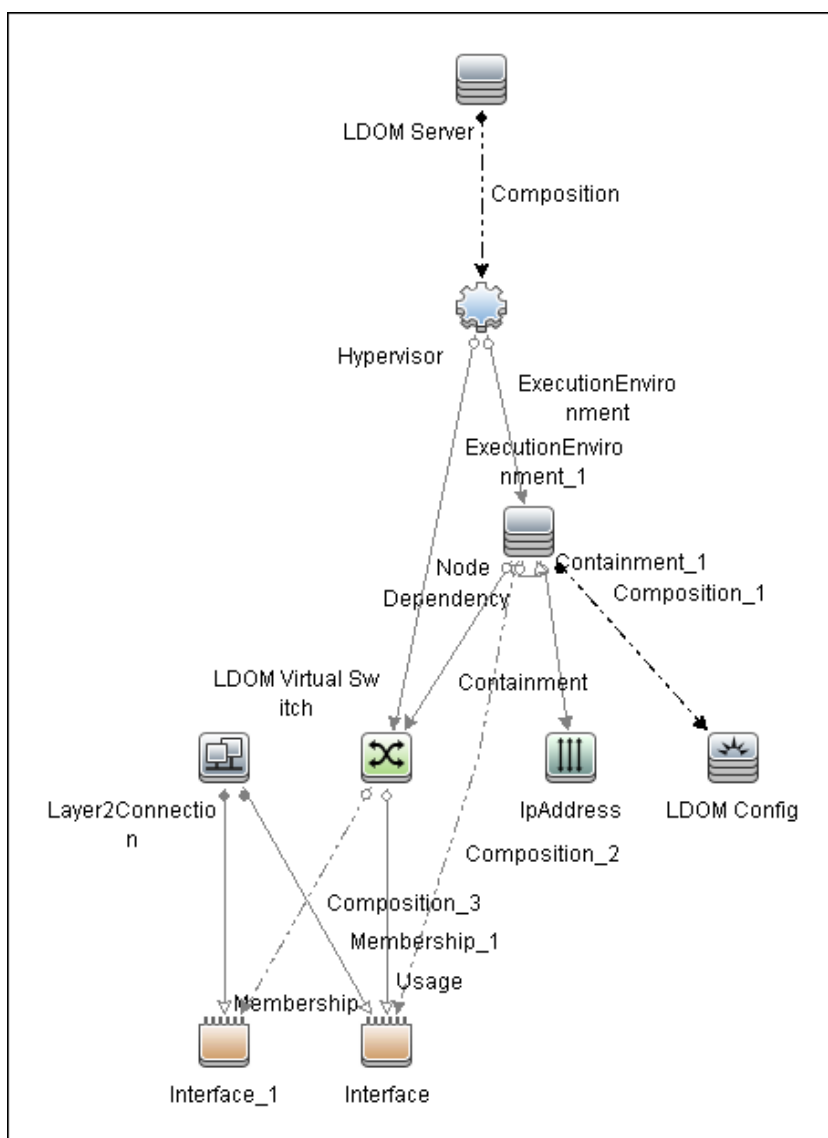
Oracle VM Server for SPARC Technology ディスカバリは、LDOM バージョン 1.0～1.3 および 2.0、Oracle VM Server for SPARC バージョン 2.0～2.1 をサポートしています。

## トポロジ

本項では、次のトポロジ・マップを示します。

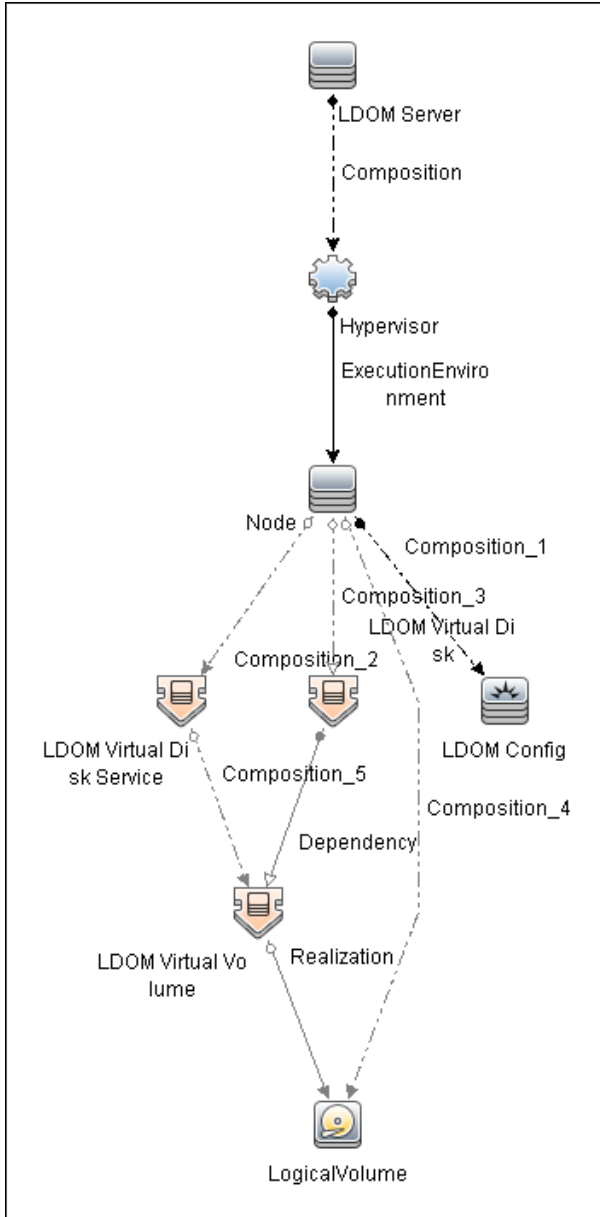
- [「LDOM\\_Networking トポロジ」\(171ページ\)](#)
- [「LDOM Storage トポロジ」\(172ページ\)](#)

### LDOM\_Networking トポロジ



注: 検出された CIT のリストについては、「[検出された CIT](#)」(176ページ)を参照してください。

## LDOM Storage トポロジ



注: 検出された CIT のリストについては、「[検出された CIT](#)」(176ページ)を参照してください。



## Oracle VM Server for SPARC テクノロジーの検出方法

### 1. 前提条件 - 一般

- a. コントロールドメインへのシェル接続が必要です。
- b. 必要な場合、次のコマンドの実行を許可するように **sudo** を各ターゲット・ホスト上で構成します。

```
/opt/SUNWldm/bin/ldm list*
```

```
/usr/sbin/ldm list*
```

このパスは、ldm コマンドの場所に応じて異なります。

### 2. 前提条件 - プロトコル資格情報の設定

次のいずれかのプロトコルを設定します。

- SSH
- Telnet

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 3. ディスカバリの実行

- **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- **Host Connection by Shell** ジョブを実行し、ターゲット・ホストおよびそのホストへのシェル接続を検出します。
- **Host Applications by Shell** ジョブを実行し、**Logical Domains Manager** アプリケーションを含むターゲット・ホストのアプリケーションを検出します。
- ターゲット LDOM サーバのトポロジを検出するため、**Oracle VM Server for SPARC Technology by Shell** ジョブを実行します。

## Oracle VM Server for SPARC Technology by Shell アダプタ

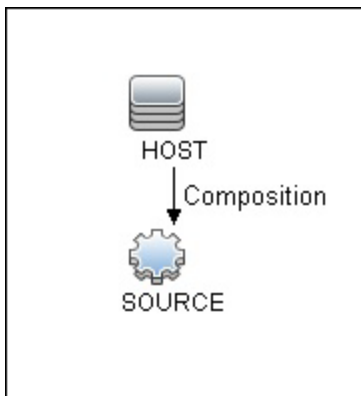
本項の内容

- [「入力 CIT」\(174ページ\)](#)
- [「入力クエリ」\(174ページ\)](#)
- [「トリガ CI データ」\(175ページ\)](#)
- [「使用スクリプト」\(175ページ\)](#)
- [「検出された CIT」\(176ページ\)](#)
- [「パラメータ」\(176ページ\)](#)

### 入力 CIT

Shell

### 入力クエリ



## トリガ CI データ

名前	値
<b>connected_os_credentials_id</b>	\${SOURCE.connected_os_credentials_id:NA}
<b>credentialsId</b>	\${SOURCE.credentials_id}
<b>hostId</b>	\${SOURCE.root_container}
<b>ip_address</b>	\${SOURCE.application_ip}
<b>protocol</b>	\${SOURCE.root_class}

## 使用スクリプト

- TTY\_HR\_CPU\_Lib.py
- networking.py
- host\_win\_wmi.py
- host\_discoverer.py
- host\_win\_shell.py
- hostresource\_win\_wmi.py
- networking\_win\_shell.py
- hostresource\_win.py
- host\_win.py
- hostresource.py
- solaris\_networking.py
- ldom.py
- ldom\_discover.py
- ldom\_report.py

- Idom\_by\_shell.py

検出された CIT

- Composition
- Containment
- Dependency
- ExecutionEnvironment
- Hypervisor
- Interface
- IpAddress
- Layer2Connection
- LDOM Resource
- Logical Volume
- Membership
- Node
- Realization

パラメータ

名前	説明
match_domain_names_to_hostnames	有効な場合、このディスカバリは、ドメイン名に設定されたホスト名を使用してゲスト LDOM をレポートします。この情報は、ホストの調整に役立つ場合があります。  標準設定 : false

# Oracle VM Server for SPARC Technology by Shell ジョブ

本 項 の 内 容

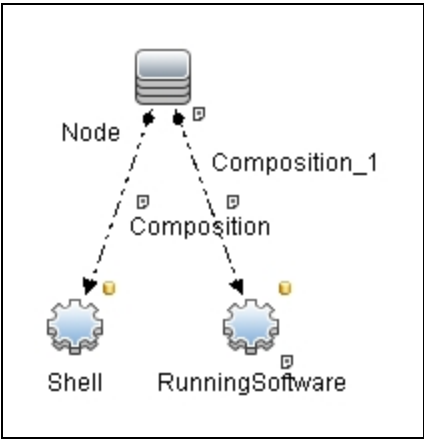
- ・「アダプタ」(177ページ)
- ・「トリガ・クエリ」(177ページ)

## アダプタ

このジョブでは、Oracle VM Server for SPARC Technology by Shell アダプタを使用します。

## トリガ・クエリ

名 前 : ldom\_control\_domain\_by\_shell



ノード名	条件
Node	NOT CI Type 等しい nt
Shell	なし
RunningSoftware	DiscoveredProductName 等しい "Logical Domains Manager"

## ディスカバリ・フロー

本項では、Oracle VM Server for SPARC Technology by Shell ジョブのディスカバリ・フローについて説明します。

### 全般

- ディスカバリは、コントロール・ドメインのシェルを使用することによって実行されます。
- コントロール・ドメインの単一のコマンド **ldm** によって、必要な構成情報のほとんどが提供されます。
- ゲスト・ドメインは次のとおりです。
  - 完全に分離されている
  - コントロール・ドメインへのネットワーク接続がない場合がある
  - OS が Solaris と異なる場合がある

注: LDOM のバージョンが 2.0 以前であり、ゲスト OS が Solaris と異なる場合、ゲスト・ドメインと通常のホストのどちらであるかを識別することはできません。

そのため、ゲスト・ドメインによって特定のディスカバリが実行されることはありません。

- アクティブまたはバインド状態にあるドメインのみが検出されます。その他の状態にあるドメインは、構成が不完全であるか、古くなっている可能性があるためです。

## Oracle VM Server for SPARC Technology by Shell ジョブのフロー

- **Logical Domains Manager のバージョンの取得**

**ldm** コマンドが実行され、このコマンドによって **Logical Domains Manager** のバージョンが取得されます。「[Logical Domains Manager のバージョン情報の取得](#)」(179ページ)を参照してください。**ldm** を実行するには、次の手順を実行します。

- **ldm** コマンドが存在することを確認します。存在しない場合、そのドメインはコントロール・ドメインではなく、ディスカバリ手順を進めることはできません。

- **ldm** コマンド への適切なパスを取得します。このパスは **/opt/SUNWldm/bin/ldm** の下または **/usr/sbin/ldm** の下である可能性があります。

#### • バインドされたすべてのドメインの構成の取得

**ldm** コマンドが実行され、バインドされたすべてのドメインの構成全体が取得されます。[「バインドされたドメインの構成のリステイング」\(180ページ\)](#)を参照してください。

#### • 汎用ネットワーク構成の取得

標準ネットワーク・ディスカバリが実行されます。これには次のコマンドが含まれます。

- netstat
- ifconfig
- dladm

詳細については、[「UNIX ベースのプロセス」\(976ページ\)](#)を参照してください。

#### • ドメイン内で仮想スイッチが作成したインタフェース名の取得

ドメイン内で作成された仮想スイッチは、それぞれが追加の仮想インタフェースを作成します(通常 **vsw<番号>** という名前)。これらのインタフェースを有効にすることによって、親ドメインはそのスイッチへの接続を確立できます。このようなインタフェースの名前を取得するため、追加の **find** コマンドが実行されます。[「ドメイン内の仮想スイッチが作成したインタフェースの検出」\(183ページ\)](#)を参照してください。

#### • 物理 CPU ごとのコア数の取得

物理コアに関する情報は、**/usr/sbin/prtpicl -c other | grep CORE** コマンドで取得されます。このあとは、Solaris 上での通常の CPU ディスカバリ方法が実行されます。詳細については、[「ホストのリソースおよびアプリケーションの検出方法」](#)を参照してください。

## コマンド

本項では、このディスカバリで使用するコマンドの例を示します。

### Logical Domains Manager のバージョン情報の取得

#### コマンド

```
/usr/sbin/ldm -V
```

## 出力

```
Logical Domains Manager (v 2.1)
  Hypervisor control protocol v 1.6
  Using Hypervisor MD v 1.3

System PROM:
  Hostconfig      v. 1.0.0.      @(#)Hostconfig 1.0.0.b 2010/09/15 03:03
[serpa:release]
  Hypervisor      v. 1.9.0.      @(#)Hypervisor 1.9.0.b 2010/09/15 01:48
  OpenBoot        v. 4.32.0.     @(#)OpenBoot 4.32.0.b 2010/09/29 19:13
```

## バインドされたドメインの構成のリスティング

### コマンド

```
/usr/sbin/ldm list-bindings -p
```

## 出力

出力は簡潔に示すために一部省略

```
VERSION 1.5
DOMAIN|name=primary|state=active|flags=normal,control,vio-
service|cons=SP|ncpu=8|mem=4294967296|util=2.4|uptime=10178475
UUID|uuid=11111111-1e91-c63f-99c7-e7484ec50000
MAC|mac-addr=00:21:28:11:73:a0
HOSTID|hostid=0x85117333
CONTROL|failure-policy=ignore
DEPENDENCY|master=
CORE
|cid=0|cpuset=0,1,2,3,4,5,6,7
VCPU
|vid=0|pid=0|util=0.7%|strand=100|cid=0
|vid=1|pid=1|util=0.6%|strand=100|cid=0
|vid=2|pid=2|util=0.9%|strand=100|cid=0
|vid=3|pid=3|util=0.8%|strand=100|cid=0
|vid=4|pid=4|util=2.1%|strand=100|cid=0
|vid=5|pid=5|util=0.5%|strand=100|cid=0
|vid=6|pid=6|util=0.5%|strand=100|cid=0
|vid=7|pid=7|util=3.3%|strand=100|cid=0
MAU
|id=0|cpuset=0,1,2,3,4,5,6,7
MEMORY
|ra=0x8000000|pa=0x8000000|size=4294967296
VARIABLES
|auto-boot?=false
```



```
|boot-device=disk0 disk1
|keyboard-layout=US-English
IO
|dev=pci@0|alias=pci
|dev=niu@80|alias=niu
|dev=pci@0/pci@0/pci@8/pci@0/pci@9|alias=MB/RISER0/PCIE0
|dev=pci@0/pci@0/pci@8/pci@0/pci@1|alias=MB/RISER1/PCIE1
|dev=pci@0/pci@0/pci@9|alias=MB/RISER2/PCIE2
|dev=pci@0/pci@0/pci@8/pci@0/pci@a|alias=MB/RISER0/PCIE3
|dev=pci@0/pci@0/pci@8/pci@0/pci@2|alias=MB/RISER1/PCIE4
|dev=pci@0/pci@0/pci@8/pci@0/pci@8|alias=MB/RISER2/PCIE5
|dev=pci@0/pci@0/pci@1/pci@0/pci@2|alias=MB/NET0
|dev=pci@0/pci@0/pci@1/pci@0/pci@3|alias=MB/NET2
|dev=pci@0/pci@0/pci@2|alias=MB/SASHBA
VCC|name=vcc|port-range=5001-5010
|client=guest1@vcc|port=5001
VSW|name=vsw1|mac-addr=00:21:28:11:73:a2|net-dev=e1000g2|dev=switch@1|default-vlan-
id=1|pvid=1|vid=|mode=|mtu=1500|linkprop=|id=1
|peer=vnet0@guest1|mac-addr=00:14:4f:f9:6f:4d|pvid=1|vid=|mtu=1500
VDS|name=vds0
|vol=guest1os|opts=|dev=/dev/zvol/dsk/ldoms/guest1os|mpgroup=
|vol=guest1ap|opts=|dev=/dev/zvol/dsk/ldoms/guest1ap|mpgroup=
|vol=L1_2234|opts=|dev=/dev/dsk/c6t6006048000029010117753303232334d0s2|mpgroup=
|vol=L1_2228|opts=|dev=/dev/dsk/c6t60060480000290101177533032323238d0s2|mpgroup=
|vol=L1_221C|opts=|dev=/dev/dsk/c6t60060480000290101177533032323143d0s2|mpgroup=
|client=vdisk0@guest1|vol=guest1os
|client=vdisk1@guest1|vol=guest1ap
|client=vdisk2@guest1|vol=L1_2234
|client=vdisk3@guest1|vol=L1_2228
|client=vdisk4@guest1|vol=L1_221C
VCONS|type=SP
DOMAIN|name=guest1|state=active|flags=normal|cons=5001|ncpu=32|
mem=19327352832|util=0.0|uptime=8584562
UUID|uuid=22222222-8dfb-6742-9705-d2f4d4310000
MAC|mac-addr=00:14:4f:f9:35:8f
HOSTID|hostid=0x84f93555
CONTROL|failure-policy=ignore
DEPENDENCY|master=
CORE
|cid=1|cpuset=8,9,10,11,12,13,14,15
|cid=2|cpuset=16,17,18,19,20,21,22,23
|cid=3|cpuset=24,25,26,27,28,29,30,31
|cid=4|cpuset=32,33,34,35,36,37,38,39
VCPU
|vid=0|pid=8|util=0.3%|strand=100|cid=1
|vid=1|pid=9|util=0.1%|strand=100|cid=1
|vid=2|pid=10|util=0.0%|strand=100|cid=1
|vid=3|pid=11|util=0.0%|strand=100|cid=1
```

```

|vid=4|pid=12|util=0.3%|strand=100|cid=1
|vid=5|pid=13|util=0.0%|strand=100|cid=1
|vid=6|pid=14|util=0.0%|strand=100|cid=1
|vid=7|pid=15|util=0.0%|strand=100|cid=1
|vid=8|pid=16|util=0.0%|strand=100|cid=2
|vid=9|pid=17|util=0.0%|strand=100|cid=2
|vid=10|pid=18|util=0.0%|strand=100|cid=2
|vid=11|pid=19|util=0.0%|strand=100|cid=2
|vid=12|pid=20|util=0.0%|strand=100|cid=2
|vid=13|pid=21|util=0.0%|strand=100|cid=2
|vid=14|pid=22|util=0.3%|strand=100|cid=2
|vid=15|pid=23|util=0.1%|strand=100|cid=2
|vid=16|pid=24|util=0.0%|strand=100|cid=3
|vid=17|pid=25|util=0.0%|strand=100|cid=3
|vid=18|pid=26|util=0.1%|strand=100|cid=3
|vid=19|pid=27|util=0.1%|strand=100|cid=3
|vid=20|pid=28|util=0.0%|strand=100|cid=3
|vid=21|pid=29|util=0.0%|strand=100|cid=3
|vid=22|pid=30|util=0.0%|strand=100|cid=3
|vid=23|pid=31|util=0.0%|strand=100|cid=3
|vid=24|pid=32|util=3.6%|strand=100|cid=4
|vid=25|pid=33|util=0.0%|strand=100|cid=4
|vid=26|pid=34|util=0.0%|strand=100|cid=4
|vid=27|pid=35|util=0.0%|strand=100|cid=4
|vid=28|pid=36|util=0.2%|strand=100|cid=4
|vid=29|pid=37|util=0.0%|strand=100|cid=4
|vid=30|pid=38|util=0.0%|strand=100|cid=4
|vid=31|pid=39|util=0.0%|strand=100|cid=4
MAU
|id=1|cpuset=8,9,10,11,12,13,14,15
|id=2|cpuset=16,17,18,19,20,21,22,23
|id=3|cpuset=24,25,26,27,28,29,30,31
|id=4|cpuset=32,33,34,35,36,37,38,39
MEMORY
|ra=0x8000000|pa=0x108000000|size=19327352832
VARIABLES
|boot-device=/virtual-devices@100/channel-devices@200/disk@0:a disk net
|keyboard-layout=US-English
VNET|name=vnet0|dev=network@0|service=vsw1@primary|mac-
addr=00:14:4f:f9:6f:4d|mode=|pvid=1|vid=|mtu=1500|linkprop=|id=0
|peer=vsw1@primary|mac-addr=00:21:28:11:73:a2|mode=|pvid=1|vid=|mtu=1500
VDISK|name=vdisk0|vol=guest1os@vds0|timeout=|dev=disk@0|
server=primary|mpgroup=|id=0
VDISK|name=vdisk1|vol=guest1ap@vds0|timeout=|dev=disk@1|
server=primary|mpgroup=|id=1
VDISK|name=vdisk2|vol=L1_2234@vds0|timeout=|dev=disk@2|
server=primary|mpgroup=|id=2
VDISK|name=vdisk3|vol=L1_2228@vds0|timeout=|dev=disk@3|
server=primary|mpgroup=|id=3

```

```
VDISK|name=vdisk4|vol=L1_221C@vds0|timeout=|dev=disk@4|
server=primary|mpgroup=|id=4
VCONS|group=guest1|service=vcc@primary|port=5001
```

## ドメイン内の仮想スイッチが作成したインタフェースの検出

### コマンド

```
find /devices/virtual-devices@100 -type c -name virtual-network-switch*
```

### 出力

```
/devices/virtual-devices@100/channel-devices@200/virtual-network-switch@0:vsw0
/devices/virtual-devices@100/channel-devices@200/virtual-network-switch@1:vsw1
```

## CPU ごとのコア数の検出

### コマンド

```
/usr/sbin/prtpicl -c other | grep CORE
```

### 出力

```
CORE0 (other, b2333315a2)
CORE1 (other, b2333315cc)
CORE2 (other, b2333315f6)
CORE3 (other, b233331620)
CORE0 (other, b2333315a2)
CORE1 (other, b2333315cc)
CORE2 (other, b2333315f6)
CORE3 (other, b233331620)
```

この出力には、それぞれ4個のコアを持つ2つの物理CPUが示されています。

## トラブルシューティングおよび制限事項

- LDOMの技術的な制限事項とアーキテクチャのため、すべてのゲスト・ドメインをディスカバリ・ジョブでレポートできるわけではありません。このコントロール・ドメインに配置された仮想スイッチへのネットワーク接続がないゲスト・ドメインは、識別情報が不足しているためにレポートの対象にできません。
- LDOMによって作成される仮想ネットワーク・デバイスの中には、MACアドレスが割り当てられるものがあります。これらのMACは自動生成されるようにする、または手動で割り当てることができます。場合によって、異なる複数のLDOMサーバがMACを生成することがあります。ゲスト・ドメインに関し、それらの仮想インタフェースのMACアドレス以外に使用可能な識別情報がないため、異なるLDOM上のMACが一致した場合、ドメインの対応するノードがCMDB内で統合されることがあります。

## 第9章: Oracle VM for x86 ディスカバリ

### 本章の内容

概要 .....	185
サポートされるバージョン .....	185
トポロジ .....	186
Oracle VM for x86 トポロジの検出方法 .....	187
Oracle VM for x86 by Manager Main CLI ジョブ .....	187
Oracle VM Manager Discovery By MainCli アダプタ .....	189

## 概要

Oracle VM は、仮想化テクノロジーの最新機能をすべて備えた、機能完備の環境を実現するプラットフォームです。Oracle VM を導入することによって、サポートされている仮想化環境内でオペレーティング・システムおよびアプリケーション・ソフトウェアをデプロイできます。Oracle VM は、基盤となっている仮想化テクノロジーからユーザと管理者を隔離し、目標指向の GUI インタフェースを使用して日常的な運用業務を実施できるようにします。

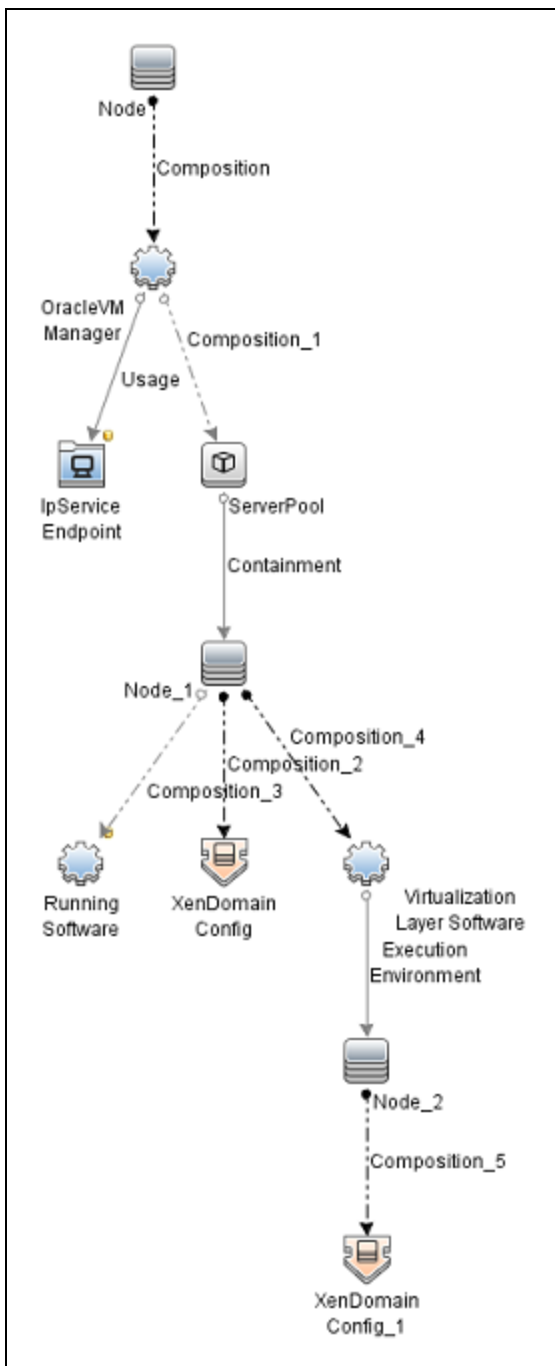
Oracle VM はエンタープライズ・クラス・サーバの仮想化ソリューションであり、Oracle VM Server for x86, Oracle VM Server for SPARC, Oracle VM Manager から構成されます。x86 サーバ用に、Oracle VM には Oracle VM Manager と Oracle VM Server for x86 が含まれます。

## サポートされるバージョン

Oracle VM for x86 デイスカバリは x86、バージョン 3.2.1 以降の Oracle VM サーバをサポートしています。

## トポロジ

Oracle VM for x86 ディスカバリのトポロジを次に示します。



## Oracle VM for x86 トポロジの検出方法

本項では、Oracle VM for x86 トポロジの検出方法について説明します。

### 前提条件

- Oracle VM Manager にシェル接続できることを確認します。
- SSH プロトコル資格情報を設定します。資格情報にポートが指定された場合、その資格情報は、ポートがリスン中の宛先にのみ適用されます。指定されていない場合、資格情報はスキップされます。

### ディスカバリの実行

**注:** Oracle VM for x86 Topology ディスカバリは、シャロー・モードまたはディープ・モードで実行できます。シャロー・ディスカバリを実行するには、次の手順 1～3 を実行します。ディープ・ディスカバリを実行するには、次のすべての手順を実行します。

1. **Range IPs by ICMP** ジョブを実行します(到達可能な IP を検出)。
2. **Host Connection by Shell** ジョブを実行します(ターゲット・ホストおよびそのホストへのシェル接続を検出)。
3. **Host Applications by Shell** ジョブを実行します(Oracle VM エージェント、Oracle VM Manager、Oracle VM CLI のリスン中のポートを検出)。

**注:** シャロー・ディスカバリのみを実行する場合、ここで終了し、次の手順に進む必要はありません。

4. **Oracle VM for x86 by Manager Main CLI** ジョブを実行します(Oracle VM Manager が管理する仮想化トポロジを検出)。

## Oracle VM for x86 by Manager Main CLI ジョブ

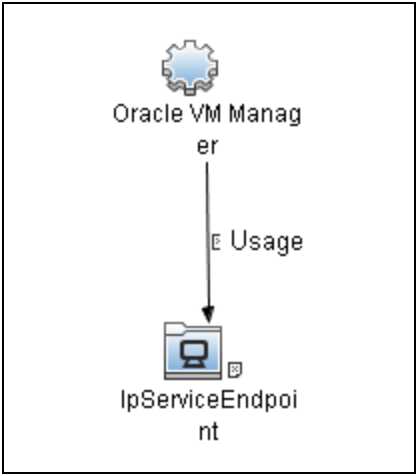
本項ではこのジョブに関する詳細を説明します。

アダプタ

ID : oracle\_vm\_manager\_by\_maincli

トリガ TQL

- トリガ CI : **OracleVMMManager**(オプションのエンド ポイント **ovm\_manager\_cli\_ssh** が接続)
- トリガ・クエリ:



- CI 属性条件 :

CI	属性値
IpServiceEndpoint	ServiceNames には ovm_manager_cli_ssh が含まれます。

ディスカバリ・フロー

このジョブは、利用可能なすべてのSSH 資格情報を反復し、次の場所のいずれかに指定されたポートを使用して、マネージャの宛先に接続を試みます。

- メインのCLI エンドポイント・ポート
- SSH 資格情報ポート



- 標準設定ポート : 10000(スクリプト内に指定)

接続が確立されている場合、ディスカバリは OVM CLI を使用して次の順序で実行されます。

1. このジョブは、CLI コマンド出力を XML に設定します(コマンド : **set output=xml**)。
2. このジョブは、利用可能なすべてのサーバの一覧を示します(コマンド : **list Server**)。
3. このジョブは、すべての仮想マシンの一覧を示します(コマンド : **list VM**)。
4. このジョブは、各サーバに関する詳細情報を取得します(コマンド : **show Server**)。
5. このジョブは、各仮想マシンに関する詳細情報を取得します(コマンド : **show Vm**)。
6. このジョブは、CLI / Oracle VM Manager のバージョンを取得します(コマンド : **showversion**)。

## Oracle VM Manager Discovery By MainCli アダプタ

本項ではこのアダプタに関する詳細を説明します。

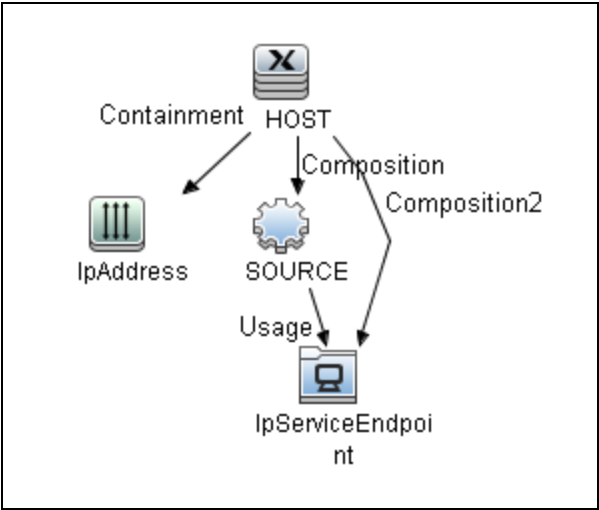
### ID

oracle\_vm\_manager\_by\_maincli

### 入力 CIT

OracleVMManager

入力 TQL



トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id:NA}
ip_address	\${SOURCE.application_ip:NA}
protocol_port	\${IpServiceEndpoint.network_port_number:NA}

使用スクリプト

- `iteratortools.py`
- `ovm_arg_validator.py`
- `ovm_reportage.py`
- `ovm_linkage.py`
- `ovm_software.py`

- **ovm\_virtualization.py**
- **ovm\_xen\_domain.py**
- **ovm\_flow.py**
- **ovm\_node.py**
- **ovm\_domain.py**
- **ovm\_cli.py**
- **manager\_by\_ovm\_cli.py** : エントリ・ポイント・スクリプト

## 検出された CIT

- Composition
- Containment
- Dependency
- ExecutionEnvironment
- Interface
- IpAddress
- IpServiceEndpoint
- Node
- Realization
- RunningSoftware
- ServerPool
- Usage
- Virtualization Layer Software

- XenDomainConfig

## パラメータ

- **commandExecutionDurationInMs** : すべての CLI コマンドの実行のために割り当てられた時間(ミリ秒)です。指定される時間の値は、マネージャ・ホストのロード係数に応じて異なります。標準設定値は 2000 ミリ秒 (2 秒) です。
- **reportStoppedVMs** : 停止した仮想マシンのディスカバリを実行します。標準設定値は **false**。

## 第10章: Solaris Zones ディスカバリ

### 本章の内容

概要 .....	194
サポートされるバージョン .....	194
トポロジ .....	195
Solaris Zones の検出方法 .....	196
Solaris Zones by TTY ジョブ .....	197
トリガ・クエリ .....	197
アダプタ .....	197
パラメータ .....	197
作成または変更されたエンティティ .....	198
ディスカバリ・メカニズム .....	200
トラブルシューティングおよび制限事項 .....	211

## 概要

Solaris Zones パーティション分割テクノロジーは、オペレーティング・システム・サービスを仮想化し、アプリケーションを実行するための分離されたセキュアな環境を提供するために使用されます。ゾーンとは、Solaris オペレーティング・システムの単一のインスタンス内に作成される仮想オペレーティング・システム環境です。ゾーンを作成すると、システム内のその他の部分からプロセスが分離されるアプリケーション実行環境を構築できます。この分離によって、1つのゾーン内で実行されるプロセスがゾーン外から監視されることがなく、ほかのゾーン内で実行されているプロセスに影響することがありません。スーパーユーザ資格情報を使用して実行されているプロセスであっても、ほかのゾーン内のアクティビティを参照する、または影響を与えることはできません。

ゾーンは、アプリケーションがデプロイされているマシンの物理的な属性から、アプリケーションを分離する抽象レイヤを設けるためにも使用されます。このような属性の例として、物理デバイス・パスがあります。

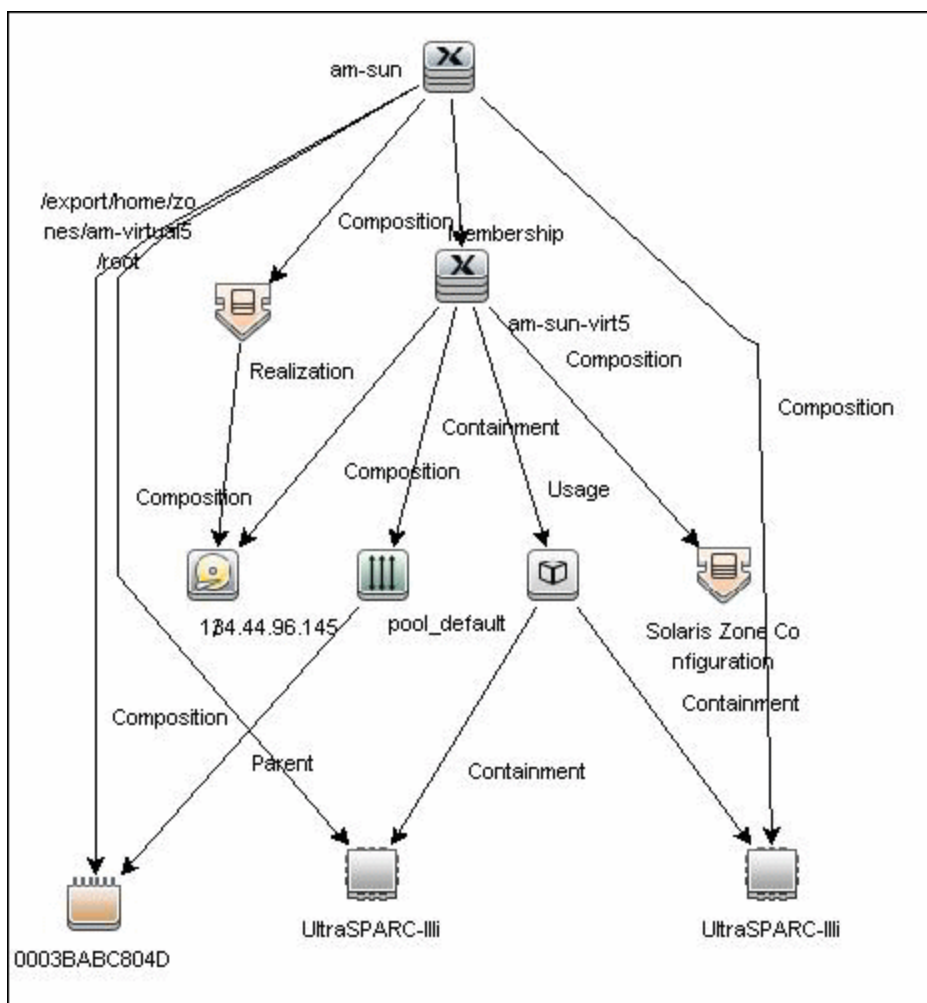
## サポートされるバージョン

Solaris Zones ディスカバリは Solaris 10 以降をサポートしています。

## トポロジ

次の図は、Solaris Zones ディスカバリのトポロジとサンプル出力を示します。

注: 検出された CIT のリストについては、[「Solaris Zones by TTY ジョブ」\(197ページ\)](#)を参照してください。



## Solaris Zones の検出方法

### 本項の内容

#### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、SSH プロトコルおよび Telnet プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

#### 2. 前提条件 - 権限の設定

ゾーンはマシンのグローバル・ゾーンから検出されます。そのため、次を実行するための適切な権限を持っている必要があります。

- グローバル・ゾーンへのアクセスとディスカバリの実行
- **zlogin** コマンドを介したノン・グローバル・ゾーンへのログイン

**注:** zlogin コマンドは次のいずれかを使用して実行できます。

- ルート・ユーザ(標準設定値)
- グローバル・ゾーン・ユーザへの接続 ディスカバリ・パターンの **zloginWithConnectedUser** パラメータを使用することで、このオプションを構成できます。

#### 3. ディスカバリの実行

- Range IPs by ICMP** ジョブを実行し、IP 範囲内で稼働中のマシンを検出します。
- Host Connection by Shell** ジョブを実行し、シェル接続と、ホストに関する基本情報を検出します。
- Solaris Zones by TTY** ジョブを実行し、ゾーン構成を検出します。

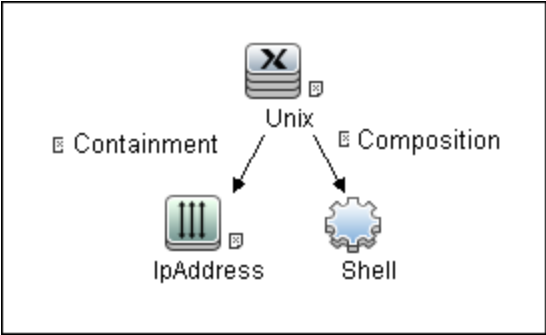
ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。



# Solaris Zones by TTY ジョブ

本 項 ではこのジョブに関する詳 細を説 明します。

## トリガ・クエリ

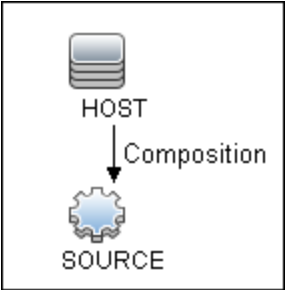


## アダプタ

Solaris Zones by TTY ジョブでは, **SolarisZone\_Disc\_By\_TTY** アダプタを使用します。

### ・ 入カクエリ

入カクエリには 1 つの Shell CI のみが含まれます。



## パラメータ

パラメータ	説 明
reportBasicTopology	基 本トポロジ(true)または完 全トポロジ(false)のど ちらをレポートするかを指 定します。  標準設定 : false

パラメータ	説明
<b>zloginWithConnectedUser</b>	<p><b>true</b> の場合、グローバル・ユーザ・アカウントへの接続を使用して zlogin が実行されます。<b>false</b> の場合、ルート・アカウントを使用して zlogin が実行されます。</p> <p>標準設定 : false</p>

## 作成または変更されたエンティティ

### • 追加の CI タイプ :

- Solaris Zones Config
- Solaris Resource Pool

### • 追加の有効なリンク :

- Solaris Resource Pool > **Containment** > CPU
- Unix > **Usage** > Solaris Resource Pool
- Unix > **Composition** > Solaris Resource Pool

### • 変更されたビュー :

- Solaris Zones ビュー

### • 変更されたスクリプト :

- SolarisZone\_Disc\_By\_TTY.py

### • 追加のエンリッチメント :

- Solaris Zones ネットワーク

## 検出された CIT

- Composition
- Containment
- Cpu
- ExecutionEnvironment

- Fibre Channel HBA
- FileSystem
- FileSystemExport
- Hypervisor
- IPMP Group
- Interface
- IpAddress
- IpSubnet
- Membership
- Node
- Parent
- Realization
- Solaris Resource Pool
- Solaris Zone Config
- Usage

注: トポロジを表示するには、[「トポロジ」\(195ページ\)](#)を参照してください。

## ディスカバリ・メカニズム

### 本項の内容

- ・「接続された OS が、ゾーン準拠であることの確認」(200ページ)
- ・「ゾーン・リストの取得および接続されたホストがグローバル・ゾーンであることの確認」(201ページ)
- ・「各ノン・グローバル・ゾーンの構成の取得」(202ページ)
- ・「グローバル・ゾーン・インタフェースの MAC アドレスの取得」(204ページ)
- ・「グローバル・ゾーンの IP 情報の取得」(204ページ)
- ・「排他的ゾーンの IP 情報の取得」(206ページ)
- ・「排他的ゾーンの専用インタフェースの MAC アドレスの取得」(206ページ)
- ・「グローバル・ゾーンの CPU 情報の取得」(207ページ)
- ・「リソース・プールの取得」(208ページ)
- ・「ファイバ・チャネル・アダプタの取得」(210ページ)

### 接続された OS が、ゾーン準拠であることの確認

コマンド	uname -r
出力例	5.10
取得された値	5.10
コメント	このコマンドでは、Solaris OS バージョンを取得します。バージョンが 5.10 の場合、そのバージョンでゾーンがサポートされているとみなされ、ディスカバリを続行します。5.10 と異なる場合 (たとえば 5.9)、ホストはゾーン準拠でないとみなされ、Server does not support zones というメッセージが表示されてディスカバリは終了します。

## ゾーン・リストの取得および接続されたホストがグローバル・ゾーンであることの確認

コマンド	<code>/usr/sbin/zoneadm list -cp</code>
出力例 1	<pre>0:global:running:/::native:shared 27:zone1:running:/var/opt/zones/zone1 :11559a59-3c6f-6a6e-a723-cc8159351247: native:excl -:zone2:configured:/var/opt/zones/ zone2::native:shared</pre>
出力例 2(ルート権限なし)	<pre>0:global:running:/ 1:am-virtual6:running:/export/home/ zones/am-virtual6 5:am-virtual5:running:/export/home/ zones/am-virtual5 7:am-virtual3:running:/virtual/3 9:am-virtual1:running:/am-virtual/1</pre>
取得された値	<p>ゾーンの名称 : zone1</p> <p>ゾーンの状態 : 実行中</p> <p>ゾーン・パス : /var/opt/zones/zone1</p>
コメント	<p>このコマンドでは、ゾーンのリストと、ゾーンの名称、状態、パスなどのゾーン構成を取得します。次のことが確認されます。</p> <ul style="list-style-type: none"> <li>• <b>global</b> が出力に含まれていること。含まれていない場合、ディスカバリの接続先ゾーンはグローバルではありません。</li> <li>• グローバル・ゾーンのほかに、少なくとも 1 つのノン・グローバル・ゾーンがあること。</li> </ul> <p>ない場合、<b>Server does not have zones defined</b> というメッセージが表示されてディスカバリは終了します。</p>

## 各ノン・グローバル・ゾーンの構成の取得

コマンド	<code>/usr/sbin/zonecfg -z &lt;ゾーン名&gt; info</code>
出力例 1	<pre>zonename: zone1 zonepath: /var/opt/zones/zone1 brand: native autoboot: true bootargs: -m verbose pool: limitpriv: default,sys_time scheduling-class: ip-type: exclusive fs: dir: /mnt/globalzone special: /var/opt/zone1-data raw not specified type: lofs options: [] net: address not specified physical: bge2 defrouter not specified device match: /dev/bge2 dedicated-cpu: ncpus: 1 importance: 1 capped-cpu: [ncpus: 1.00] capped-memory: physical: 16G [swap: 8G] [locked: 12G]</pre>
出力例 2	<pre>zonename: zone2 zonepath: /var/opt/zones/zone2 brand: native autoboot: true bootargs: -m verbose pool: limitpriv: default scheduling-class: FSS ip-type: shared</pre>

	<pre> fs: dir:/mnt special:/var/opt/zone2-data raw not specified type: lofs options:[] net: address:134.44.0.100 physical: bge0 defrouter not specified device match:/dev/pts* rctl: name: zone.cpu-shares value:(priv=privileged,limit=5,action=none) </pre>
取得された値	<p>出力から次の情報が取得されます。</p> <ul style="list-style-type: none"> <li>• ブランド (指定されていない場合はネイティブと仮定)</li> <li>• 自動起動</li> <li>• リソース・プール名</li> <li>• 権限の制限</li> <li>• スケジューリング・クラス</li> <li>• IP タイプ</li> <li>• マウントされたすべてのファイル・システム</li> <li>• ネットワーク情報 (IP またはネットワーク・インタフェース, またはその両方)</li> <li>• 専用 CPU およびそれらの重要度</li> <li>• メモリ・キャップ</li> <li>• CPU キャップ</li> <li>• CPU 共有</li> </ul>
コメント	<p>このコマンドは、見つかった各ノン・グローバル・ゾーンに対して実行されます。これらのプロパティのほとんどは、<b>Solaris Zone Config</b> CI に格納されます。ファイル・システムは、グローバル・ゾーンからノン・グローバル・ゾーンへのファイル・システム・エクスポートとしてレポートされます。リソース・プール名は、対応するリソース・プール CI へのリンクの作成に使用されます。</p>

## グローバル・ゾーン・インタフェースの MAC アドレスの取得

コマンド	/usr/bin/netstat -np
出力例	<pre> Net to Media Table:IPv4  Device IP Address Mask Flags Phys Addr ----- -----  bge0 134.44.0.101 255.255.255.255 o 00:15:f2:05:9e:ff bge0 134.44.1.150 255.255.255.255 o 00:15:f2:9b:2d:96 bge0 134.44.0.100 255.255.255.255 SPLA 00:14:4f:82:74:a4 bge0 134.44.98.135 255.255.255.255 o 00:1c:c0:2b:57:35 bge0 224.0.0.0 240.0.0.0 SM 01:00:5e:00:00:00 </pre>
取得された値	対応するインタフェースの MAC アドレス
コメント	<p>このコマンドでは、排他的ゾーンで使用される専用インタフェース以外のすべてのインタフェースのリストを取得します。</p> <p>グローバル・ゾーンのインタフェースは、共有ゾーンと共有されるため、このコマンドは 1 回のみ実行されます。</p> <p>zonecfg の出力内の MAC アドレスおよび情報によって、共有ノン・グローバル・ゾーンの Host CI の作成が可能になります。</p>

## グローバル・ゾーンの IP 情報の取得

コマンド	/usr/sbin/ifconfig -a
出力例	<pre> lo0: flags=2001000849&lt;UP,LOOPBACK, RUNNING,MULTICAST,IPv4,VIRTUAL&gt; mtu 8232 index 1 inet 127.0.0.1 netmask ff000000 lo0:1: flags=2001000849&lt;UP,LOOPBACK,RUNNING, MULTICAST,IPv4,VIRTUAL&gt; mtu 8232 index 1 zone zone2 inet 127.0.0.1 netmask ff000000 e1000g1: flags=1000843&lt;UP,BROADCAST,RUNNING, MULTICAST,IPv4&gt; mtu 1500 index 2 inet 134.44.0.50 netmask ffffffff00 broadcast 134.44.0.255 </pre>



	<pre>e1000g1:1: flags=1000843&lt;UP,BROADCAST,RUNNING, MULTICAST,IPv4&gt; mtu 1500 index 2 zone zone2 inet 134.44.0.100 netmask ffffffff00 broadcast 134.44.0.255</pre>
取得された値	対応するインタフェースの MAC アドレス
コメント	<p>このコマンドでは、対応する共有ノン・グローバル・ゾーンと共有されているグローバル・ゾーンの IP 構成を取得します。</p> <p>この情報は、IP アドレスのレポートと、それらに対応するネットワーク・インタフェースにリンクするために使用されます。</p>

## 排他的ゾーンの IP 情報の取得

コマンド	<code>/usr/sbin/zlogin -l &lt;ユーザ名&gt; &lt;ゾーン名&gt; /usr/sbin/ifconfig -a</code>
出力例	<pre>lo0: flags=2001000849&lt;UP,LOOPBACK,RUNNING, MULTICAST,IPv4,VIRTUAL&gt; mtu 8232 index 1 inet 127.0.0.1 netmask ff000000 bge2: flags=201004843&lt;UP,BROADCAST,RUNNING, MULTICAST,DHCP,IPv4,CoS&gt; mtu 1500 index 2 inet 134.44.0.200 netmask fffffffc00 broadcast 134.44.0.255 ether 0:14:4f:82:74:a6</pre>
取得された値	ループバックを除く、存在するすべての IP
コメント	<p>このコマンドでは、排他的ノン・グローバル・ゾーンの IP 情報を取得します。zlogin 用に sudo パターンの設定を簡略化するために -l &lt;ユーザ&gt; スイッチが追加されていますが、これはジョブ・パラメータから削除できます。</p> <p><b>注 :</b> ディスカバリは、実行中状態のゾーンに対してのみ zlogin を実行します。</p>

## 排他的ゾーンの専用インタフェースの MAC アドレスの取得

コマンド	<code>/usr/sbin/zlogin -l &lt;ユーザ名&gt; &lt;ゾーン名&gt; /usr/bin/netstat -np</code>
出力例	<pre>Net to Media Table:IPv4  Device IP Address Mask Flags Phys Addr ----- -----  bge2 134.44.0.200 255.255.255.255 SPLA 00:14:4f:82:74:a6 bge2 224.0.0.0 240.0.0.0 SM 01:00:5e:00:00:00</pre>
取得された値	MAC アドレス
コメント	<p>インタフェースの MAC アドレスが、インタフェース名と一緒に取得されます。</p> <p><b>注 :</b> ディスカバリは、実行中状態のゾーンに対してのみ zlogin を実行します。</p>

グローバル・ゾーンの CPU 情報の取得

コマンド	/usr/sbin/psrinfo -v
出力例	Status of virtual processor 0 as of:05/03/2010 16:00:15 on-line since 04/26/2010 19:45:40.  The sparcv9 processor operates at 1200 MHz, and has a sparcv9 floating point processor.  Status of virtual processor 1 as of:05/03/2010 16:00:15 on-line since 04/26/2010 19:45:42.  The sparcv9 processor operates at 1200 MHz, and has a sparcv9 floating point processor.
取得された値	仮想 CPU 数とID 仮想プロセッサ名 (sparcv9) プロセッサ速度 (1200)
コメント	仮想プロセッサの各インスタンスにつき、ディスカバリは CPU を名前 (sparcv9) と速度 (1200) 付きで作成します。これらはグローバル・ゾーンにリンクされます。これら是对应するリソース・プールにもリンクされます。

## リソース・プールの取得

コマンド	/usr/sbin/pooladm
出力例	<pre> system default   string system.comment   int system.version 1   boolean system.bind-default true   string system.poold.objectives wt-load   pool SUNWtmp_zone1     int pool.sys_id 1     boolean pool.active true     boolean pool.default false     int pool.importance 1     string pool.comment     boolean pool.temporary true     pset SUNWtmp_zone1   pool pool_default     int pool.sys_id 0     boolean pool.active true     boolean pool.default true     int pool.importance 1     string pool.scheduler FSS     string pool.comment     pset pset_default </pre>
出力例(つづき)	<pre> pset SUNWtmp_zone1   int pset.sys_id 1   boolean pset.default false   uint pset.min 1   uint pset.max 1 </pre>

	<pre> string pset.units population uint pset.load 0 uint pset.size 1 string pset.comment boolean pset.temporary true  cpu int cpu.sys_id 0 string cpu.comment string cpu.status on-line </pre>
取得された値	<ul style="list-style-type: none"> <li>• プール： <ul style="list-style-type: none"> <li>◦ 名前</li> <li>◦ 標準設定かどうか</li> <li>◦ アクティブかどうか</li> <li>◦ 重要度</li> <li>◦ スケジューラ</li> </ul> </li> <li>• Pset： <ul style="list-style-type: none"> <li>◦ 名前</li> <li>◦ 最小 CPU 数</li> <li>◦ 最大 CPU 数</li> <li>◦ 対象物</li> </ul> </li> </ul> <p><b>プールから Pset および Pset から割り当てられた CPU への ID による関係</b></p>
コメント	<p>この情報によって、プールのレポートが可能になり、これらのプールがグローバル・ゾーンの対応する CPU に ID によってリンクされます。現在、ディスカバリはプールとその Pset を 1 つのエントリとしてレポートします。</p> <p>リソース・プール機能を使用されていない、またはアクティブでない場合、ディスカバリは構成を読み取ることができません。その場合も、標準設定(ダミー)プールが属性なしでレポートされ、すべての CPU はそこにリンクされます。</p> <p>ノン・グローバル・ゾーンに、構成内のプール名が含まれている場合、ディスカバリはゾーンをそのプールにリンクします。</p> <p>ノン・グローバル・ゾーンに専用の cpu プロパティ・セットがある場合、ディスカ</p>

	<p>バリは一時的な動的プールの名前をリンクのために計算します。名前の形式は SUNWtmp_&lt;ゾーン名&gt; になります。</p>
--	--

## ファイバ・チャネル・アダプタの取得

コマンド	/usr/sbin/fcinfo hba-port
出力例	<pre> HBA Port WWN:2100001c3491b18a OS Device Name:/dev/cfg/c1 Manufacturer:QLogic Corp. Model:555-1156-02 Firmware Version:05.01.00 FCode/BIOS Version:BIOS:2.2;   fcode:2.1; EFI:2.0; Serial Number:0708R00-4259732555 Driver Name: qlc Driver Version:20090610-3.21 Type:N-port State: online Supported Speeds:1Gb 2Gb 4Gb Current Speed:2Gb Node WWN:2000001c3491b18a HBA Port WWN:2101001c34b1b18a OS Device Name:/dev/cfg/c2 Manufacturer:QLogic Corp. Model:555-1156-02 Firmware Version:05.01.00 FCode/BIOS Version:BIOS:2.2;   fcode:2.1; EFI:2.0; Serial Number:0708R00-4259732555 Driver Name: qlc Driver Version:20090610-3.21 Type:N-port State: online Supported Speeds:1Gb 2Gb 4Gb Current Speed:2Gb Node WWN:2001001c34b1b18a                     </pre>
取得された値	<ul style="list-style-type: none"> <li>• ポート WWN</li> <li>• OS デバイス名</li> <li>• メーカー</li> <li>• モデル</li> </ul>

	<ul style="list-style-type: none"><li>• タイプ</li><li>• シリアル</li><li>• ドライバ・バージョン</li></ul>
コメント	この情報によって、ディスカバリはファイバ・チャネルの HBA をレポートできません。OS デバイス名は <b>name</b> 属性に保持されます。ポート WWN は <b>HBA WWN</b> 属性に保持されます。

## トラブルシューティングおよび制限事項

**問題 :** ディスカバリ中に、次の警告メッセージが表示されます : `Not enough permissions to execute command, zone is skipped.`

**理由 :** これは、ディスカバリを実行しているユーザに権限がなかったために、スクリプトが **zlogin** を使用して排他的ゾーンのネットワーク情報を取得できないことを示している可能性があります。

**解決策 :** 必要な権限をそのユーザに付与します。

# 第11章: VMware

## 本章の内容

VMware の前提条件 .....	212
VMware Infrastructure ディスカバリ .....	215
サポートされるバージョン .....	217
VIM プロトコルの SSL サポート .....	217
トポロジ .....	217
VMware VIM Topology の検出方法 .....	226
Manual VMware VIM Connection ジョブの実行方法 .....	229
CIM を介した VMware ESX サーバトポロジの検出方法 .....	229
Manual VMware VIM Connection ジョブ .....	231
VMware ESX Connection by CIM ジョブ .....	232
VMware ESX Connection by VIM ジョブ .....	235
VMware ESX Topology by CIM ジョブ .....	238
VMware ESX Topology by VIM ジョブ .....	241
VMware vCenter Connection by VIM ジョブ .....	245
VMware vCenter Topology by VIM ジョブ .....	247
VMware VMotion ディスカバリおよびイベント追跡 .....	253
概要 .....	254
サポートされるバージョン .....	254
VMware VMotion の検出方法とイベントの追跡方法 .....	254
VMware vMotion Monitor by VIM ジョブ .....	255
VMware ディスカバリのトラブルシューティングおよび制限事項 .....	258
トラブルシューティング .....	259
制限事項 .....	259

## VMware の前提条件

VMware ジョブを実行する前に、次の前提条件を満たす必要があります。



## 1. 資格情報と権限

vCenter Server と ESX サーバでは、ログインに資格情報(ユーザ名とパスワード)が必要です。さらに、検出されたすべてのエンティティに対し、サーバからの取得を実行するための権限も必要です。VMware vSphere Client の各エンティティ(ホスト、クラスタ、仮想マシンなど)の[権限]タブに、認証された各ユーザが読み取り専用以上のロールで表示されていることを確認してください。

## 2. JAR ファイルのインストール

適切な JAR ファイルがプローブ・マシンにインストールされていることを確認する必要があります。これは、管理ゾーンを使用して行うことも、手動で行うこともできます。

### 管理ゾーンを使用した JAR ファイルのインストール方法

このタスクは次の手順で構成されます。

- a. **VMware Infrastructure SDK** を次の場所からダウンロードします。

<http://www.vmware.com/support/developer/vc-sdk/>

必要なバージョンは 5.0 以降です。

- b. ファイルをローカル・フォルダに展開します。例 :C:\VMware-vSphere-SDK

jar ファイルは、インベントリ・デイスカバリ・アクティビティを介してインストールします。インベントリ・デイスカバリ・アクティビティは、新規作成するか、既存のものを編集します。

- c. インベントリ・デイスカバリ・アクティビティを新規作成します。

- i. [データフロー管理]>[Universal Discovery]>[ゾーンベース デイスカバリ]>[管理ゾーン]に移動します。

- ii. 適切な管理ゾーンを選択します。

- iii.  ボタンをクリックし、[新規 デイスカバリ アクティビティ]>[インベントリ]を選択します。

[インベントリ デイスカバリ アクティビティの新規作成]ダイアログ・ボックスが表示されます。

- iv. アクティビティ・ウィザードのオンライン・プロンプトに従ってアクティビティを作成し、アクティブ化します。

[仮想化]ページで次の手順を実行します。


- A. [仮想化トポロジを含む]を選択します。
- B. [ディスカバリ オプション]で[VMware SDK ライブラリ]を選択します。
- C. [ファイルをインポート]をクリックします。[ファイルをインポート]ダイアログ・ボックスが表示されます。
- D. 選択肢として、Axis ライブラリと JAX-WS ライブラリがあります。不明な場合は、JAX-WS を選択します。インポートするファイルを選択します。

- Axis ライブラリを使用する場合は、**vsphere-ws\java\Axis** フォルダ(例 : C:\VMware-vSphere-SDK\vsphere-ws\java\Axis)に移動し、**vim.jar** と **vim25.jar** を選択します。

注 : Axis ライブラリは VMware Infrastructure SDK バージョン 5.1 以降には含まれません。

- JAX-WS ライブラリを使用する場合は、**vsphere-ws\java\JAXWS\lib** フォルダ(例 : C:\VMware-vSphere-SDK\vsphere-ws\java\JAXWS\lib)に移動し、**vim25.jar** を選択します。

既存のインベントリ・ディスカバリ・アクティビティを編集するには、次の手順を実行します。

- i. [データフロー管理] > [Universal Discovery] > [ゾーンベース ディスカバリ] > [管理ゾーン]に移動します。
- ii. 適切な管理ゾーンを選択します。
- iii. 既存のインベントリ・ディスカバリ・アクティビティを選択します。
- iv.  ボタンをクリックします。

[インベントリ ディスカバリ アクティビティの編集]ダイアログ・ボックスが表示されます。

- v. [仮想化]ページで、上記の手順 iv に示した内容を実行します。

## 手動による JAR ファイルのインストール方法

- a. VMware Infrastructure SDK を次の場所からダウンロードします。

<http://www.vmware.com/support/developer/vc-sdk/>

必要なバージョンは 5.0 以降です。

- b. ファイルをローカル・フォルダに展開します。例 :C:\VMware-vSphere-SDK
- c. jar ファイルを **VMware Infrastructure SDK** から取得します。選択肢として、Axis ライブラリと JAX-WS ライブラリがあります。不明な場合は、JAX-WS を選択します。
  - Axis ライブラリを使用する場合は、**vsphere-ws\java\Axis** フォルダ(例 :C:\VMware-vSphere-SDK\vsphere-ws\java\Axis)に移動し、**vim.jar** と **vim25.jar** を選択します。

**注:** Axis ライブラリは VMware Infrastructure SDK バージョン 5.1 以降には含まれません。

- JAX-WS ライブラリを使用する場合は、**vsphere-ws\java\JAXWS\lib** フォルダ(例 :C:\VMware-vSphere-SDK\vsphere-ws\java\JAXWS\lib)に移動し、**vim25.jar** を選択します。
- d. jar ファイルを次の場所に配置します。

<プローブ>/runtime/probeManager/discoveryResources/vmware

**注:** これは、VMware ディスカバリを実行する各プローブに対して実行する必要があります。

- e. プローブを再起動します。

## VMware Infrastructure ディスカバリ

### 本章の内容

サポートされるバージョン .....	217
VIM プロトコルでサポートされるバージョン .....	217
CIM プロトコルでサポートされるバージョン .....	217
VIM プロトコルの SSL サポート .....	217
トポロジ .....	217
VMware VIM Topology の検出方法 .....	226
Manual VMware VIM Connection ジョブの実行方法 .....	229

CIM を介した VMware ESX サーバトポロジの検出 方法 .....	229
Manual VMware VIM Connection ジョブ .....	231
VMware ESX Connection by CIM ジョブ .....	232
VMware ESX Connection by VIM ジョブ .....	235
VMware ESX Topology by CIM ジョブ .....	238
VMware ESX Topology by VIM ジョブ .....	241
VMware vCenter Connection by VIM ジョブ .....	245
VMware vCenter Topology by VIM ジョブ .....	247

## サポートされるバージョン

### VIM プロトコルでサポートされるバージョン

VIM プロトコルを使用した VMware インフラストラクチャのディスカバリは、次のサーバでサポートされます。

- VirtualCenter 2.5, 2.0
- vCenter Server 4, 4.1, 5.0, 5.1, 5.5, 6.0
- ESX Server 3.0, 3.5, 4.0, 4.1, 5.0, 5.1, 5.5, 6.0

### CIM プロトコルでサポートされるバージョン

CIM プロトコルを使用した VMware インフラストラクチャのディスカバリは、ESX サーバ 4.x, 5.x, 6.0 でサポートされます。

## VIM プロトコルの SSL サポート

Web サービスは http 通信を使用します。SSL を介した通信も可能です。VIM プロトコルは標準設定で SSL を使用しますが、SSL を使用しないように設定することも可能です。

VIM プロトコルをサポートする各サーバ(vCenter Server または ESX サーバ)には独自の SSL 証明書があります。SSL を介した接続ではこの証明書を確認して受け入れます。

- サーバからのすべての証明書をトラスト・ストアにインポートし、接続のたびに確認します。一連の信頼済み証明書に含まれないものは拒否します(安全な方法)。
- 確認なしですべての証明書を受け入れます(安全性が低い方法)。

現在、DFM は 1 つの方法(すべての証明書を常に受け入れる)のみをサポートしています。

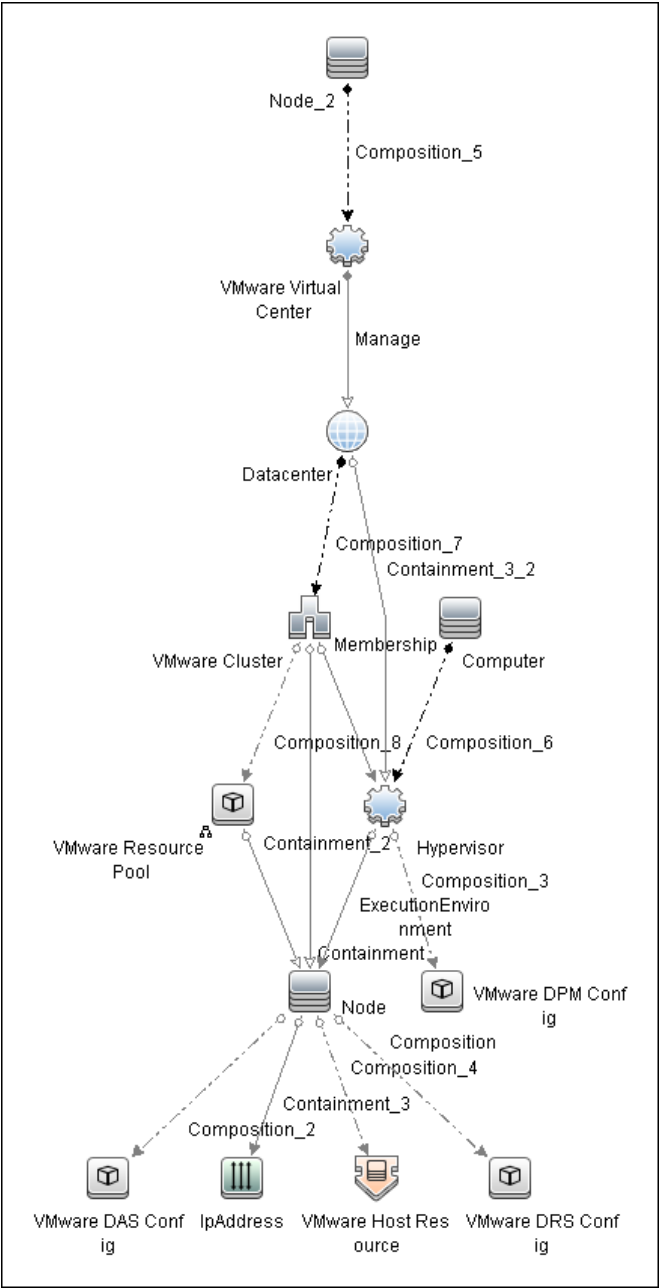
## トポロジ

本項の内容

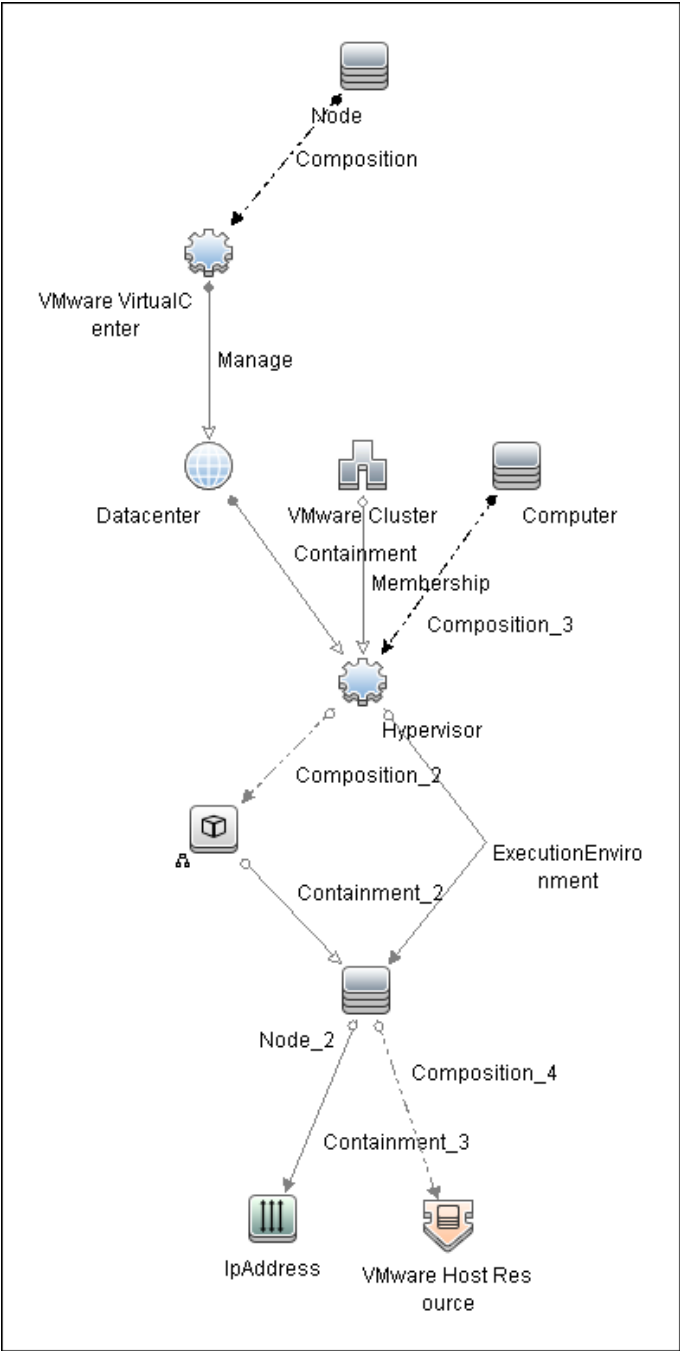
- [「クラスタの仮想トポロジ・ビュー」\(219ページ\)](#)
- [「非クラスタの仮想トポロジ・ビュー」\(220ページ\)](#)

- [「ネットワークの仮想トポロジ・ビュー」\(221ページ\)](#)
- [「ライセンス・トポロジ・マップ」\(222ページ\)](#)
- [「ストレージの仮想トポロジ・ビュー」\(223ページ\)](#)
- [「分散ネットワークのトポロジ」\(224ページ\)](#)
- [「VMware ESX サーバ・インベントリおよび仮想トポロジ・ビュー」\(224ページ\)](#)
- [「VMware ESX サーバ・インベントリおよび仮想トポロジ・インスタンスの例」\(225ページ\)](#)

クラスタの仮想トポロジ・ビュー

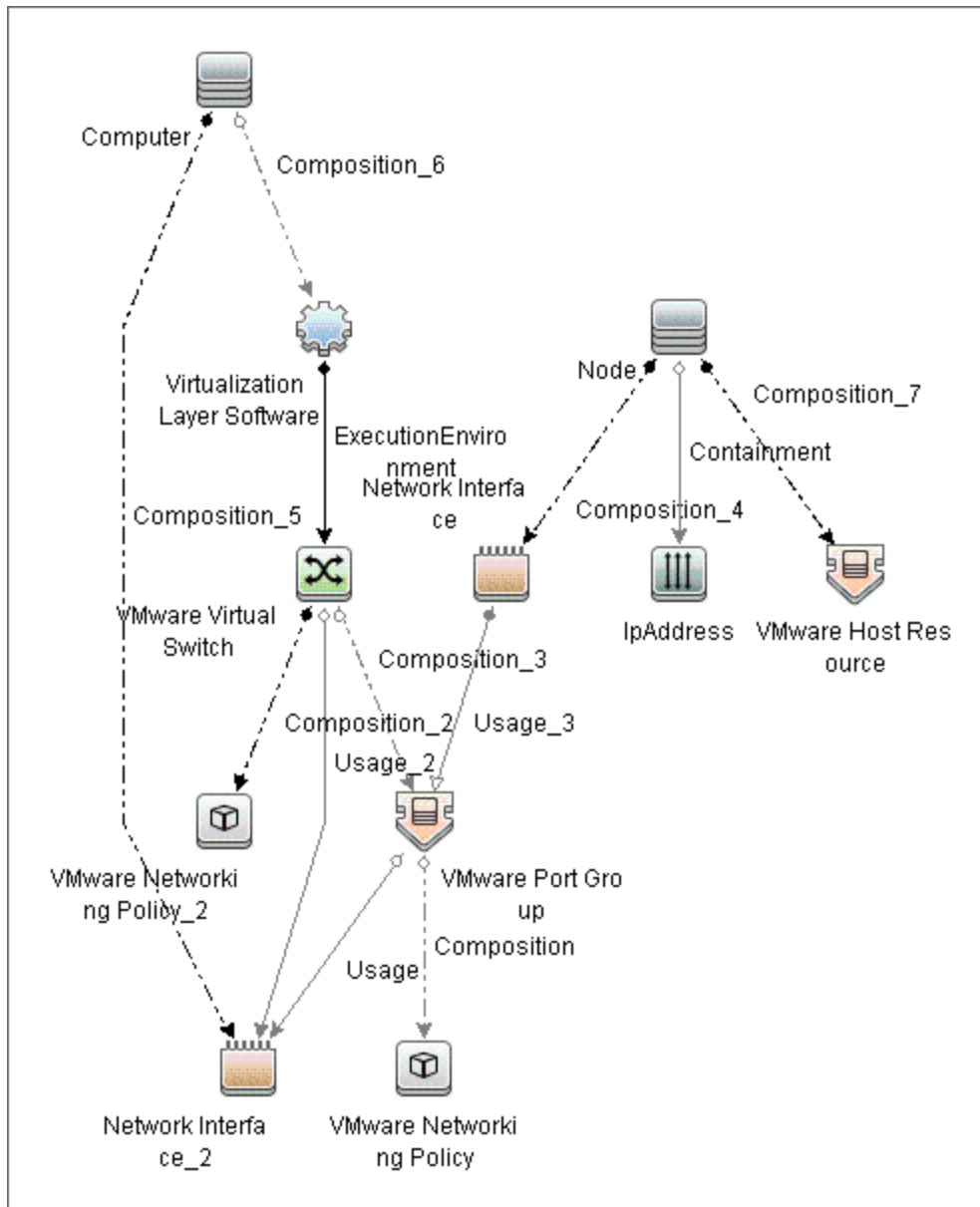


非 クラスタの仮想トポロジ・ビュー

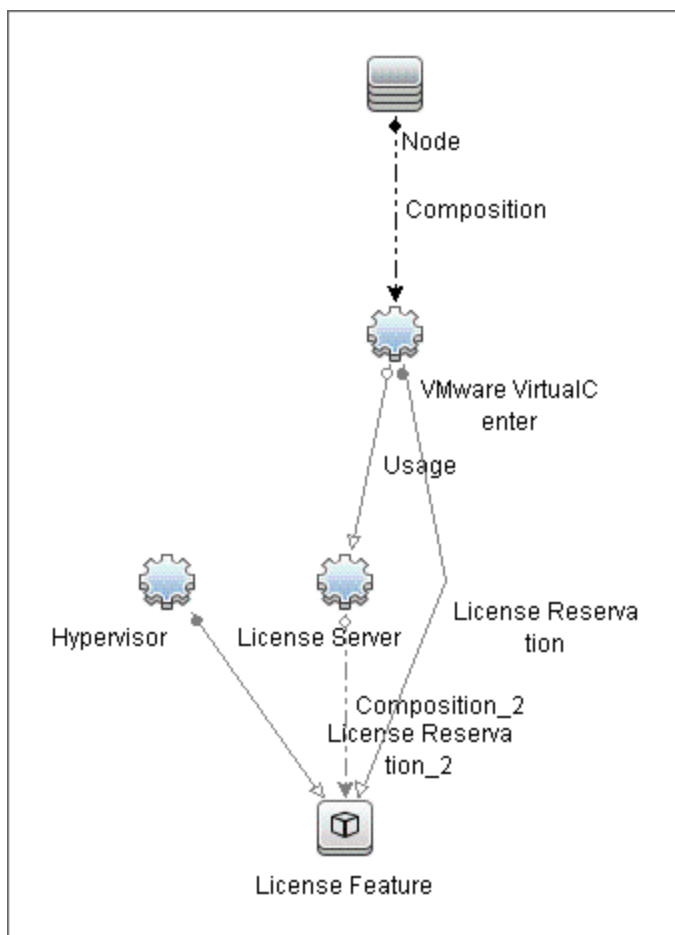




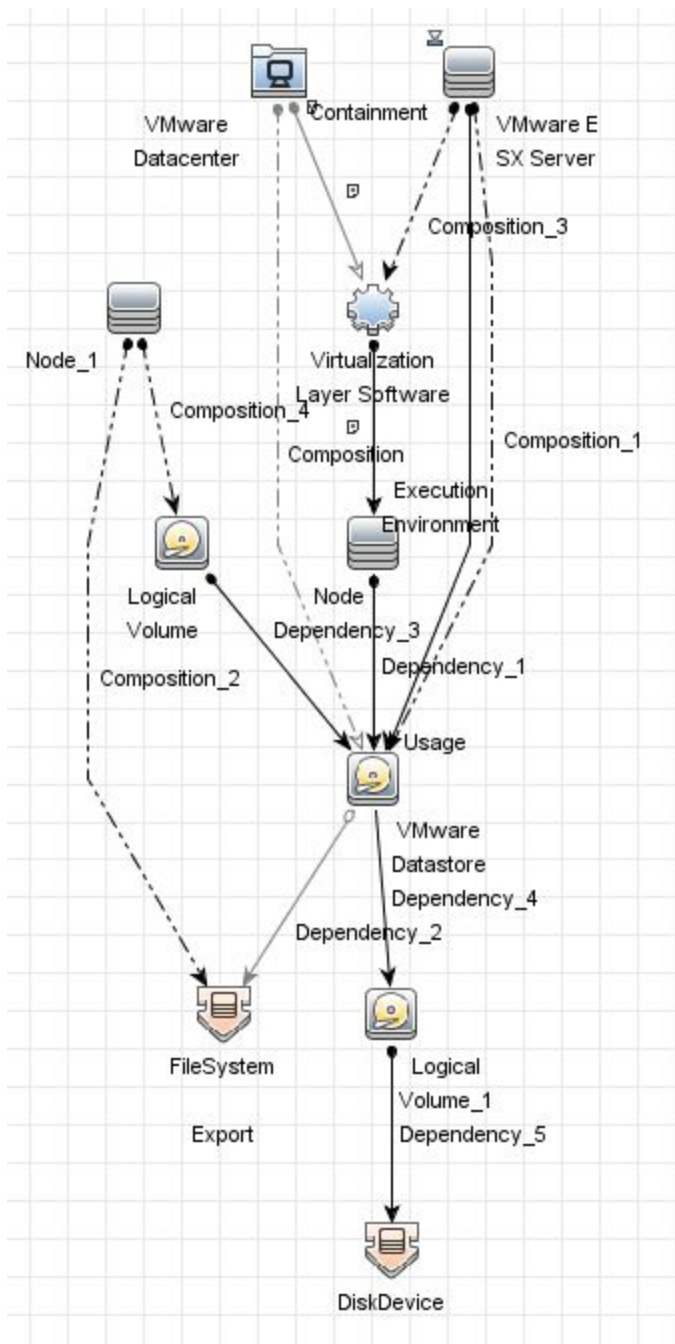
## ネットワークの仮想トポロジ・ビュー



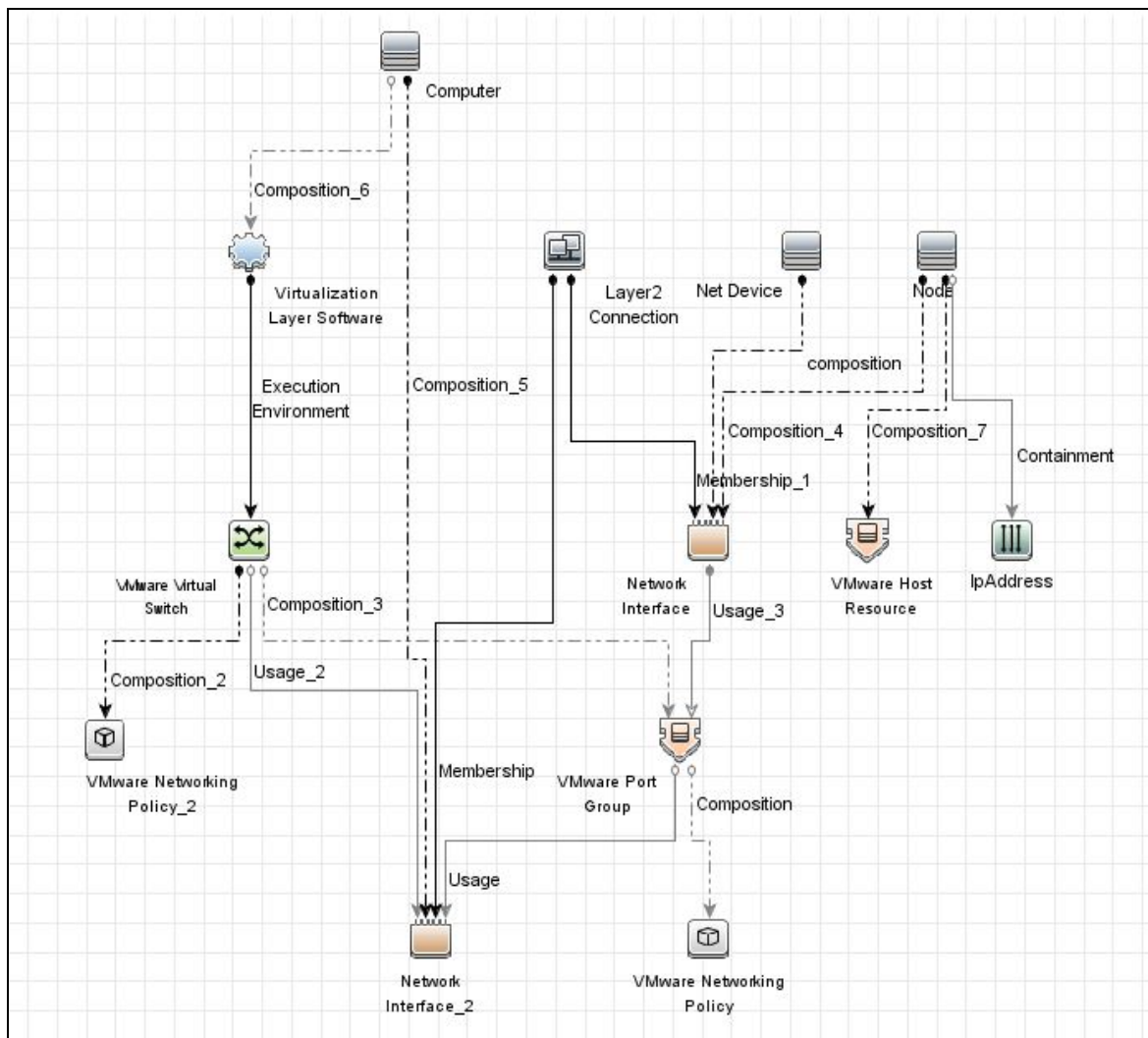
## ライセンス・トポロジ・マップ



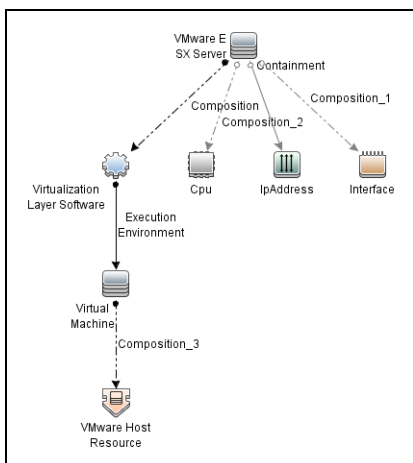
## ストレージの仮想トポロジ・ビュー



## 分散ネットワークのトポロジ

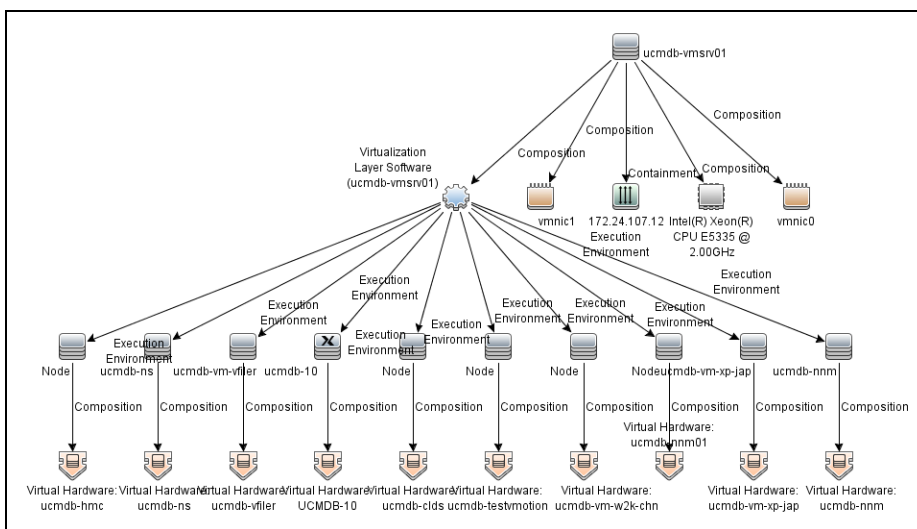


## VMware ESX サーバ・インベントリおよび仮想トポロジ・ビュー



このビューは、CIM プロトコルを使用してレポートされます。

## VMware ESX サーバ・インベントリおよび仮想トポロジ・インスタンスの例



VMware ESX サーバ・インベントリおよび仮想トポロジ・インスタンスは、CIM プロトコルを使用してレポートされます。

## VMware VIM Topology の検出方法

このタスクでは、VMware VIM Topology suite を検出する方法を説明します。VMware で実行中の仮想マシン (VM)、ESX サーバ、ネットワーク・リソース、クラスター・リソースが検出可能です。

注: ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Discovery Control Panel」を参照してください。

### 本項の内容

- [「前提条件 - プロトコル資格情報の設定」\(226ページ\)](#)
- [「前提条件 - VMware Infrastructure の権限の設定」\(226ページ\)](#)
- [「ホスト・ディスカバリの実行」\(227ページ\)](#)
- [「プロセス・ディスカバリの実行」\(227ページ\)](#)
- [「VMware Infrastructure ディスカバリの実行」\(227ページ\)](#)

### 1. 前提条件 - プロトコル資格情報の設定

- WMI, シェル (Telnet, SSH, NTCMD), SNMP プロトコルは、ホストおよびホスト・プロセスの検出に必須です。

これらのプロトコルには、ユーザ名、パスワード、ドメイン名が必要です (ドメイン名は NTCMD では省略可能)。

- VIM プロトコルは、すべての VMware ジョブに必須です。
  - このプロトコルには、ユーザ名とパスワードが必要です。
  - [ポート番号] は省略可能です。
  - [SSL を使用] には、SSL を使用するように VMware サーバが標準設定されている場合は **true** を選択します。セキュアでない http を使用するように VMware サーバが設定されている場合は **false** を選択します。

資格情報の詳細については、『HP UCMDB Discovery and Integrations Content Guide』の「Supported Protocols」を参照してください。

### 2. 前提条件 - VMware Infrastructure の権限の設定

VMware Infrastructure Management(VIM)プロトコルには、次の権限が必要です。

- **System.Read** : ユーザがディスカバリを実行する際に必要な権限です。ユーザは、検出されたすべてのエンティティに対する権限を持っていること、および読み取り専用以上のロールが割り当てられていることが必要です。
- **Global.Licenses** : 各ライセンス機能でのライセンスの合計数と使用可能数を取得するために必要な権限です。ユーザにこれらの権限がない場合、属性は空のままになります。

### 3. ホスト・ディスカバリの実行

各 VMware サーバ(vCenter, VirtualCenter, または ESX)に接続するには、**Host Connection by Shell/WMI** ジョブのいずれかを実行して Host CI を検出します。

### 4. プロセス・ディスカバリの実行

各 VMware サーバ(vCenter, VirtualCenter, または ESX)に接続するには、特定の基準と一致する Process CI を検出し、**Host Applications by Shell/WMI** ジョブのいずれかを実行してアプリケーション署名ディスカバリを実行する必要があります。

### 5. VMware Infrastructure ディスカバリの実行

**Virtualization** モジュールには、vCenter または VirtualCenter Server 検出用の 2 つのジョブと、ESX サーバ検出用の 2 つのジョブが含まれます。

- VMware Infrastructure 環境が vCenter または VirtualCenter Server によって管理されている場合は、**VMware vCenter Connection by VIM** ジョブ、**VMware vCenter Topology by VIM** ジョブの順に実行します。
- VMware Infrastructure 環境に、管理対象外の ESX サーバ(スタンドアロン)が含まれているか、環境全体が管理対象外になっている場合は、**VMware ESX Connection by VIM** ジョブ、**VMware ESX Topology by VIM** ジョブの順に実行します。

**注 :** **Manual VMware VIM Connection** ジョブは、上記のジョブが VMware 環境を検出できない場合のインスタンスで使用することを目的としています。ただし、このジョブは手動で実行する必要があります。[「Manual VMware VIM Connection ジョブの実行方法」](#)(229ページ)を参照してください。

各ジョブの詳細については、次を参照してください。

- [「VMware vCenter Connection by VIM ジョブ」](#)(245ページ)
- [「VMware vCenter Topology by VIM ジョブ」](#)(247ページ)

- [「VMware ESX Connection by VIM ジョブ」\(235ページ\)](#)
- [「VMware ESX Topology by VIM ジョブ」\(241ページ\)](#)



## Manual VMware VIM Connection ジョブの実行方法

このジョブは、ESX サーバまたは vCenter Server に対するシェル・アクセスが存在しないために、通常の接続ジョブ(VMware ESX Connection by VIM ジョブまたは VMware vCenter Connection by VIM ジョブ)を実行できない場合に使用できます。


このタスクには次の手順が含まれます。

### 1. 前提条件

モデリング・スタジオに移動して新規クエリを作成します。新規クエリの作成に関する詳細については、『HP Universal CMDB モデリング・ガイド』の「TQL クエリの定義方法」を参照してください。

#### 注：

- VMware デイスカバリが実行される vCenter Server(または個々の VMware ESX/ESXi ホスト)に属する IPAddress CI を選択してください。
- 表示される[クエリ定義プロパティ]ダイアログ・ボックスでクエリのプロパティを設定する際は、[タイプ]フィールドの[デイスカバリ]を選択し、[OK]をクリックします。
- クエリを Root ディレクトリの **Discovery** フォルダに保存し、[OK]をクリックします。

- [データフロー管理] > [Universal Discovery] > [デイスカバリ モジュール/ジョブ] > [デイスカバリ モジュール] > [Cloud and Virtualization] > [Virtualization] > [VMware] に移動します。
- [Manual VMware VIM Connection] を選択します。
- [プロパティ] タブ > [トリガ クエリ] 表示枠で、 ボタンをクリックします。表示される[デイスカバリ クエリの選択]ダイアログ・ボックスで、手順 1 で作成したクエリ名を選択し、[OK]をクリックします。
- [OK] をクリックして変更を保存します。
- [Manual VMware VIM Connection] ジョブを右クリックし、[アクティブ化] を選択してジョブをアクティブ化します。

## CIM を介した VMware ESX サーバ・トポロジの検出方法

このタスクでは、CIM を介して VMware ESX サーバ・インベントリと仮想トポロジを検出する方法を説明します。

### 前提条件

- ターゲット ESX サーバ上で CIM エージェントが有効になっていることを確認します。
- CIM の資格情報を定義します。

このタスクには次の手順が含まれます。

1. ターゲット IP を検出するため、**Range IPs by ICMP** ジョブを実行します。
2. ターゲット ESX サーバとその基本トポロジを検出するため、**VMware ESX Connection by CIM** ジョブを実行します。
3. サーバ・インベントリと仮想化トポロジを検出するため、**VMware ESX Topology by CIM** ジョブを実行します。

**注:**

- ディスカバリは、SBLIM ライブラリ(<http://sblim.wiki.sourceforge.net>)に基づいた CIM クライアントを使用します。
- ディスカバリは、必要なすべての情報を収集するため、次の 3 つの名前空間にアクセスします。
  - root/interop : ターゲット・サーバが基本のサーバ・プロファイルを実装したことを確認し、UnitaryComputerSystem インスタンスへの参照を検索します。
  - root/cimv2 : サーバ・インベントリの情報を収集します。
  - vmware/esxv2 : 仮想化トポロジの情報を収集します。

## Manual VMware VIM Connection ジョブ

このジョブは、ESX サーバとvCenter Server への接続をサポートします。

### アダプタ

このジョブでは、VMware\_VIM\_Connection\_Manual アダプタを使用します。

### パラメータ

プロパティ	説明
remoteJVMArgs	リモート・プロセスに渡される必要がある JVM パラメータ
remoteJVMClasspath	外部の java プロセスで使用されるクラス・パス  注: このプロパティの標準設定は変更しないでください。
runInSeparateProcess	<b>true</b> にすると、外部の java 仮想マシンでのジョブ実行が可能になります。 標準設定 : true  注: このプロパティの標準設定は変更しないでください。

### ディスカバリ・フロー

1. ディスカバリ・ジョブがIP アドレスでトリガされます。
2. そのIP アドレスに該当する資格情報の一覧が取得されます。
3. 各資格情報を使用して、ターゲットへの接続が試みられます。これは、上記のIPアドレスを使用し、資格情報のエントリからポートとプロトコルを取得してURLを作成することによって実行されます。
4. 接続に成功すると、接続したサーバがESX サーバとvCenter Server のどちらであるかが確認されます。
5. ジョブによって、検出されたサーバがレポートされます。

## VMware ESX Connection by CIM ジョブ

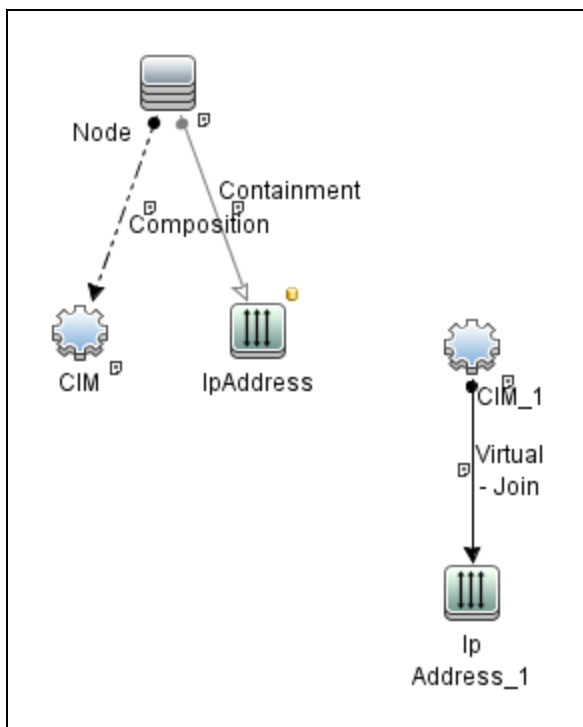
このジョブは、CIM プロトコルを使用して、対象の ESX サーバへの接続とその基本トポロジを検出します。

本項の内容

- [「トリガ・クエリ」\(232ページ\)](#)
- [「アダプタ」\(233ページ\)](#)
- [「検出された CIT」\(233ページ\)](#)

### トリガ・クエリ

- トリガ CI : IpAddress
- トリガ・クエリ:



ノード名	条件
Node	なし
CIM	なし
IpAddress	NOT IP Probe Name null AND NOT IP Is Broadcast 等しい True

## アダプタ

このジョブでは、**VMware\_ESX\_Connection\_by\_CIM** アダプタを使用します。

- トリガ CI データ

名前	値
ip_address	\${SOURCE.name}

- アダプタ・パラメータ: なし
- 使用スクリプト:
  - cim.py
  - cim\_discover.py
  - host\_discoverer.py
  - vmware\_cim\_discover.py
  - vmware\_cim\_report.py
  - vmware\_esx\_connection\_by\_cim.py

## 検出された CIT

- CIM
- Composition
- Containment
- IpAddress

- Interface
- Virtualization Layer Software
- VMware ESX Server

## VMware ESX Connection by VIM ジョブ

このジョブは、VIM プロトコルを使用して、VMware ESX サーバへの接続を検出します。

本項の内容

- [「ディスカバリ・メカニズム」\(235ページ\)](#)
- [「トリガ・クエリ」\(236ページ\)](#)
- [「アダプタ」\(236ページ\)](#)
- [「検出された CIT」\(236ページ\)](#)
- [「トラブルシューティングおよび制限事項」\(237ページ\)](#)

### ディスカバリ・メカニズム

データ・フロー管理 は次の手順を実行します。

- VIM プロトコルの資格情報を確認します。
- 定義されているポートが現在の資格情報に含まれている場合、DFM はこのポートを使用します。

そうでない場合、生成される接続 URL にポートは指定されません。

プレフィックスは、現在の資格情報の[**SSL を使用**]の値によって決まります。

- 次の形式で接続 URL を生成します: <プレフィックス>://<IP アドレス>:<ポート>/sdk
- VMware Infrastructure Client を作成し、生成された URL、および資格情報から得たユーザ名とパスワードを使用して接続します。
- 接続に成功すると、DFM は ESX サーバの製品詳細情報(バージョン、ビルド、説明)を取得します。この情報は、**Virtualization Layer Software CI** の属性として使用されます。

さらに DFM は、ESX サーバの UUID と名前も取得します。ESX UUID は **VMware ESX Server CI** の host\_key 属性に格納されます。

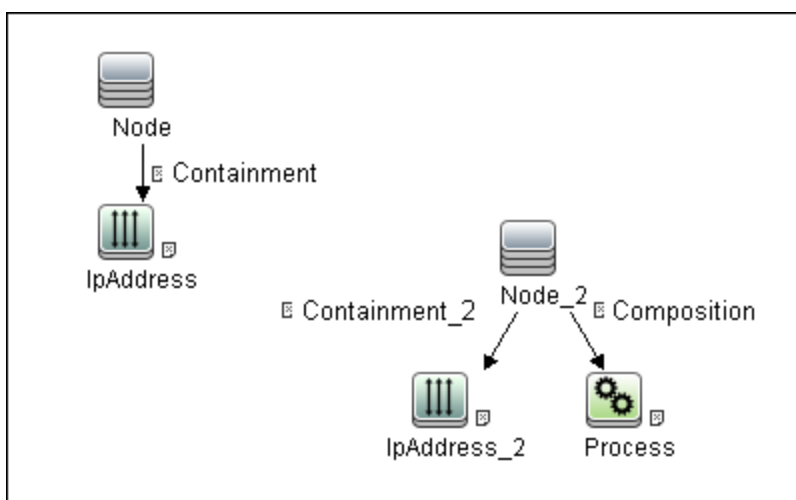
ESX サーバのホスト名は、**VMware ESX Server CI** の name(キー)属性に格納されます。

- すべてのエラーと警告をクリアし、検出されたすべての結果を返します。

そうでない場合、接続に失敗すると、DFM は次の VIM プロトコルの資格情報を試し、すべてを試すまで続けます。

## トリガ・クエリ

- トリガ CI : Node
- トリガ・クエリ:



## アダプタ

このジョブでは、**VMware\_ESX\_Connection\_by\_VIM** アダプタを使用します。

- **アダプタ・パラメータ**: このジョブは、**runInSeparateProcess** と **remoteJVMClasspath** という 2 つの内部パラメータ(パラメータ値は変更不可)を使用します。

## 検出された CIT

- Composition
- VMware ESX Server



- Virtualization Layer Software

## トラブルシューティングおよび制限事項

- **問題** : 十分な権限がないことが原因で操作を実行できない場合は、次のエラー・メッセージが表示されます。

User does not have required '<権限>' permission

**解決策** : 権限が **System.Read** に設定されていることを確認します。

- **問題** : 資格情報が正しくない場合は、次のエラー・メッセージが表示されます。

無効なユーザ名またはパスワード

- **問題** : ジョブの完了時に、タイムアウトの警告メッセージが表示されます。

<<進捗メッセージ, 重大度 : エラー>>

VMware VIM: リモート エージェントに接続しようとしているときにタイムアウトになりました。タイムアウト値を増やしてみてください。

**制限事項** : ジョブの接続タイムアウト値は、VMware API の制限事項が原因で設定できません。標準設定の 60 秒がタイムアウト値として常に使用されます。

## VMware ESX Topology by CIM ジョブ

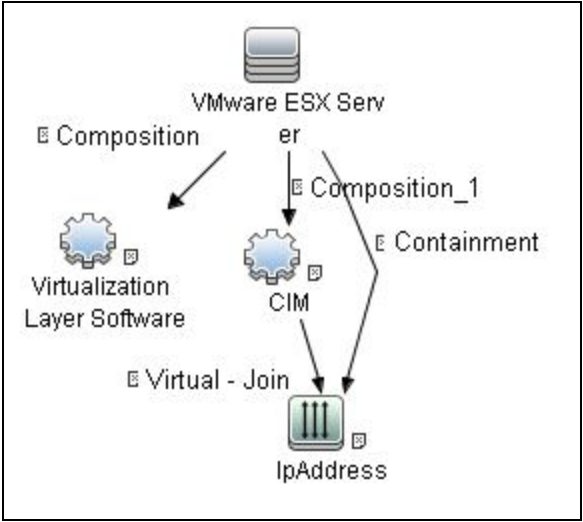
このジョブは、CIM プロトコルを使用してサーバ・インベントリと仮想化トポロジを検出します。

本項の内容

- ・ [「トリガ・クエリ」\(238ページ\)](#)
- ・ [「アダプタ」\(239ページ\)](#)
- ・ [「検出された CIT」\(239ページ\)](#)
- ・ [「制限事項」\(240ページ\)](#)

### トリガ・クエリ

- ・ トリガ CI : VMware ESX Server
- ・ トリガ・クエリ:



ノード名	条件
CIM	NOT Reference to the credentials dictionary entry null AND CimCategory 含む "VMware"
VMware ESX Server	NOT BiosUuid null

ノード名	条件
IpAddress	NOT IP Probe Name null

## アダプタ

このジョブでは, **VMware\_ESX\_Topology\_by\_CIM** アダプタを使用します。

### • トリガ CI データ

名 前	値
credentialsId	\${CIM.credentials_id}
esx_bios_uuid	\${SOURCE.bios_uuid}
esx_cmdb_id	\${SOURCE.root_id}
hypervisor_cmdb_id	\${HYPERVISOR.root_id}
ip_address	\${IP.name}

### • アダプタ・パラメータ : なし

### • 使用スクリプト :

- memory.py
- host\_discoverer.py
- cim.py
- cim\_discover.py
- vmware\_cim\_discover.py
- vmware\_cim\_report.py
- vmware\_esx\_topology\_by\_cim.py

## 検出された CIT

- Composition

- Containment
- Cpu
- IpAddress
- Virtualization Layer Software
- VMware ESX Server
- Node
- VMware Host Resource
- ExecutionEnvironment

### 制限事項

仮想マシンの仮想名前空間では、BIOS UUID、プライマリIP アドレス、ホスト名の値のみ取得できません。これらの値が存在しない場合、仮想マシンはレポートできないためスキップされます。

## VMware ESX Topology by VIM ジョブ

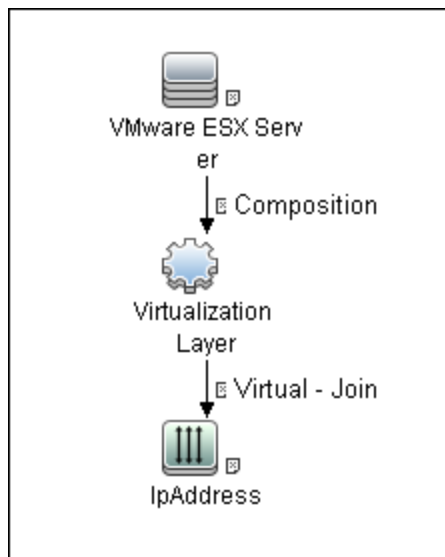
このジョブは、VIM プロトコルを使用して ESX サーバに接続し、そのトポロジを検出します。

本項の内容

- [「トリガ・クエリ」\(241ページ\)](#)
- [「アダプタ」\(241ページ\)](#)
- [「検出された CIT」\(243ページ\)](#)
- [「トラブルシューティング」\(244ページ\)](#)

### トリガ・クエリ

- トリガ CI :Virtualization Layer Software
- トリガ・クエリ:



### アダプタ

このジョブでは、**VMware\_ESX\_Topology\_by\_VIM** アダプタを使用します。

• トリガ CI データ :

<b>credentialsId</b>	ESX サーバ属性に保存された VMware Infrastructure(VIM)プロトコルの資格情報 ID
<b>server_url</b>	ESX サーバの <b>connection_url</b> 属性から取得した接続の URL
<b>ip_address</b>	ESX サーバの IP アドレス

• アダプタ・パラメータ :

プロパティ	説明
<b>discoverOutOfProbeRangeIps</b>	プローブ範囲に含まれない IP アドレスに対してディスカバリを実行するかどうかを指定  標準設定 : true。
<b>remoteJVMArgs</b>	リモート・プロセスに渡される必要がある JVM パラメータを示します。
<b>remoteJVMClasspath</b>	外部の Java プロセスで使用するクラス・パス  注: このプロパティの標準設定は変更しないでください。
<b>reportBasicTopology</b>	基本トポロジ(true)または完全トポロジ(false)のどちらかをレポートするかを指定  標準設定 : false
<b>reportLastBootTime</b>	仮想マシンの前回の起動時刻をレポートするかどうかを指定  標準設定 : true
<b>reportPoweredOffVMs</b>	パワーオフ状態の仮想マシンをレポートするかどうかを指定。レポートする場合は true, レポートしない場合は false。  標準設定 : false
<b>runInSeparateProcess</b>	true にすると, 外部の java 仮想マシンでのジョブ実行が可能になります。  標準設定 : true  注: このプロパティの標準設定は変更しないでください。

## 検出された CIT

- Composition
- Containment
- Cpu
- Dependency
- ExecutionEnvironment
- FileSystemExport
- Interface
- IpAddress
- License Feature
- License Reservation
- License Server
- Node
- Usage
- VMware Datastore
- VMware ESX Server
- VMware Host Resource
- VMware Networking Policy
- VMware Port Group
- VMware Resource Pool
- VMware Virtual Switch
- Virtualization Layer Software

## トラブルシューティング

- **問題** : 十分な権限がないことが原因で操作を実行できない場合は、次のエラー・メッセージが表示されます。

User does not have required '<権限>' permission

権限が **System.Read** に設定されていることを確認します。

- **問題** : 資格情報が正しくない場合は、次のエラー・メッセージが表示されます。

無効なユーザ名またはパスワード

- **問題** : 権限が十分ないためにDFMがライセンス情報を取得できない場合、次の警告メッセージが表示されます。

User does not have required '<権限>' permission, licensing information won't be reported



## VMware vCenter Connection by VIM ジョブ

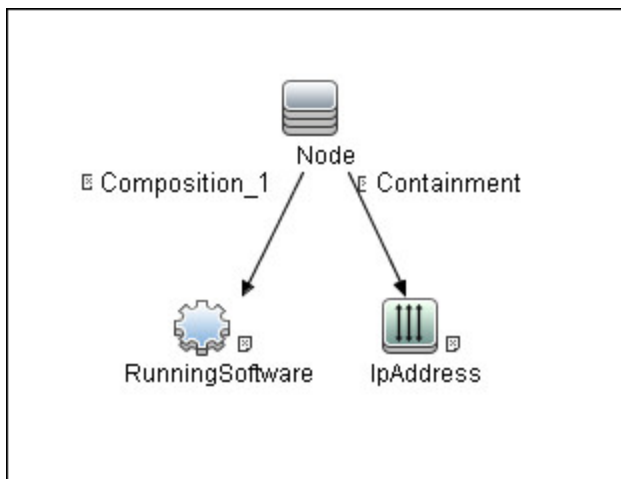
このジョブは, vCenter または VirtualCenter Server を検出します。

本項の内容

- [「トリガ・クエリ」\(245ページ\)](#)
- [「アダプタ」\(245ページ\)](#)
- [「検出された CIT」\(246ページ\)](#)
- [「トラブルシューティング」\(247ページ\)](#)

### トリガ・クエリ

- トリガ CI : Node
- トリガ・クエリ:



### アダプタ

このジョブでは, **VMware\_VirtualCenter\_Connection\_by\_VIM** アダプタを使用します。

• トリガ CI データ :

名前	説明
<b>connection_url</b>	vCenter Server の接続 URL
<b>credentialsId</b>	WMI agent CI の資格情報 ID
<b>ip_addresses</b>	ホストに接続しているすべての IP の一覧
<b>rs_id</b>	RunningSoftware としてレポートされた vCenter Server の CMDB ID
<b>vc_id</b>	vCenter Server の CMDB ID

• アダプタ・パラメータ

プロパティ	説明
<b>remoteJVMArgs</b>	リモート・プロセスに渡される必要がある JVM パラメータ
<b>remoteJVMClasspath</b>	外部の java プロセスで使用されるクラス・パス  注: このプロパティの標準設定は変更しないでください。
<b>runInSeparateProcess</b>	true にすると, 外部の java 仮想マシンでのジョブ実行が可能になります。  標準設定 : true  注: このプロパティの標準設定は変更しないでください。

## 検出された CIT

- Composition
- Containment
- IpAddress
- Node

- VMware VirtualCenter

## トラブルシューティング

- **問題** : 十分な権限がないことが原因で操作を実行できない場合は、次のエラー・メッセージが表示されます。

User does not have required '<権限>' permission

**解決策** : 検出対象のすべてのエンティティに対する権限がユーザにあることを確認します。**VMware Infrastructure Client** で、各エンティティ(ホスト、クラスタ、仮想マシンなど)の[権限]タブにアクセスします。読み取り専用以上のロールがユーザに割り当てられていることを確認します。

注: 必要な権限は、[デイスカバリ ジョブの詳細]表示枠([**Universal Discovery**] > [デイスカバリ モジュールジョブ]タブ > <ジョブ>の選択 > [詳細]タブ)で確認できます。

- **問題** : 資格情報が正しくない場合は、次のエラー・メッセージが表示されます。

無効なユーザ名またはパスワード

- **問題** : **VMware vCenter Connection by VIM** ジョブがリモート・プロセスへの接続に失敗し、ジョブを FIPS モードまたは非 FIPS モードで実行できません。

**解決策** :

- プローブを再インストールした場合は、手動によるこれ以上の変更は必要ありません。
- プローブを古いバージョン(10.20 から 10.21 など)からアップグレードした場合は、次の値を **DataFlowProbe.properties** ファイルの **basic\_discovery\_minimal\_classpath** 設定に追加する必要があります。

`../lib/cryptojce.jar;../lib/cryptojcommon.jar;../lib/jcmFIPS.jar;`

## VMware vCenter Topology by VIM ジョブ

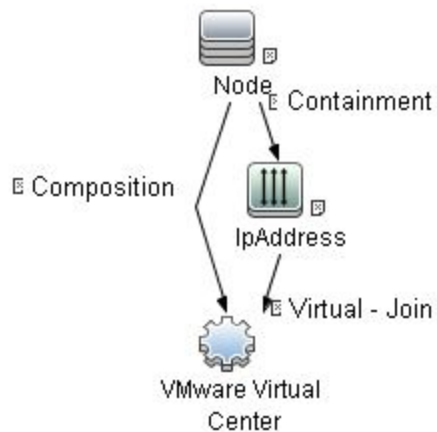
このジョブは、vCenter または VirtualCenter Server に接続し、完全な VMware Infrastructure トポロジを検出します。

本項の内容

- [「トリガ・クエリ」\(248ページ\)](#)
- [「アダプタ」\(248ページ\)](#)
- [「検出された CIT」\(250ページ\)](#)
- [「トラブルシューティング」\(251ページ\)](#)

## トリガ・クエリ

- トリガ CI :VMware VirtualCenter
- トリガ TQL クエリ:



## アダプタ

このジョブでは, **VMware\_VirtualCenter\_Topology\_by\_VIM** アダプタを使用します。

• トリガ CI データ :

名前	説明
<b>credentialsId</b>	vCenter または VirtualCenter Server の属性に保存されている VMware Infrastructure Management(VIM)プロトコルの資格情報 ID
<b>connection_url</b>	vCenter または VirtualCenter Server の <b>connection_url</b> 属性から取得された、VMware Infrastructure に接続するための URL

• アダプタ・パラメータ

プロパティ	説明
<b>discoverOutOfProbeRangeIps</b>	<p>プローブ範囲に含まれない IP アドレスに対してディスカバリを実行するかどうかを指定</p> <p><b>標準設定</b> : true。</p>
<b>remoteJVMArgs</b>	リモート・プロセスに渡される必要がある JVM パラメータを示します。
<b>remoteJVMClasspath</b>	<p>外部の Java プロセスで使用するクラス・パス</p> <p><b>注</b>: このプロパティの標準設定は変更しないでください。</p>
<b>reportBasicTopology</b>	<p>基本トポロジ(true)または完全トポロジ(false)のどちらをレポートするかを指定</p> <p><b>標準設定</b> : false</p>
<b>reportDiscoveredOsName</b>	<p><b>true</b> の場合、Node CIT の <b>DiscoveredOsName</b> 属性が入力されます。</p> <p><b>標準設定</b> : false</p> <p><b>注</b> : VM インスタンスの重複する OS 名がレポートされるため、このパラメータを有効にすることは推奨されません。</p>
<b>reportPoweredOffVMs</b>	<p>パワーオフ状態の仮想マシンをレポートするかどうかを指定。レポートする場合は true、レポートしない場合は false。</p> <p><b>標準設定</b> : false</p>
<b>runInSeparateProcess</b>	<b>true</b> の場合、外部の Java 仮想マシンでのジョブの実行を有

	<p>効にします。</p> <p><b>標準設定 : true</b></p> <p><b>注 :</b> このプロパティの標準設定は変更しないでください。</p>
--	--

検出された CIT

- Composition
- Containment
- Cpu
- Dependency
- ExecutionEnvironment
- FileSystemExport
- Interface
- IpAddress
- Licence Feature
- License Reservation
- License Server
- Manage
- Membership
- Node
- Usage
- VMware Cluster

- VMware DAS Config
- VMware DPM Config
- VMware DRS Config
- VMware Datacenter
- VMware Datastore
- VMware Distributed Virtual Switch
- VMware ESX Server
- VMware Host Resource
- VMware Networking Policy
- VMware Port Group
- VMware Resource Pool
- VMware Uplink
- VMware Virtual Switch
- VMware VirtualCenter
- Virtualization Layer Software

## トラブルシューティング

- **問題** : 十分な権限がないことが原因で操作を実行できない場合は、次のエラー・メッセージが表示されます。

User does not have required '<権限>' permission

**解決策** : 権限が **System.Read** に設定されていることを確認します。

- **問題** : 資格情報が正しくない場合は、次のエラー・メッセージが表示されます。

無効なユーザ名またはパスワード

- **問題:** ディスカバリの実行中、次の警告メッセージが通信ログに表示されます。

VM '<名前>': powered off, VM is skipped

**解決策:** このメッセージは、ディスカバリでパワー・オフ状態の仮想マシンが見つかったことを示します。標準設定では、パワー・オフ状態の仮想マシンはレポートされません。このようなパワー・オフ状態の仮想マシンは、設定が古くなっている場合があります。こうした古い情報が仮想マシンのIDに影響を及ぼし、正確でないトポロジがレポートされる可能性があります。

例:

- インタフェースのMACアドレスが別の仮想マシンに割り当てられても、パワー・オフ状態の仮想マシンとしてリストされたままになることがあります。
- パワー・オフ状態の仮想マシンのものとしてリストされたままになっているIPアドレスが、実際にはディスカバリの開始前にDHCPサーバによって別のマシンに再割り当てされていることがあります。

パワー・オフ状態の仮想マシンが引き続きレポートされるようにするには、トポロジ・ジョブの **reportPoweredOffVMs** パラメータを **true** に設定します。

- **問題:** ディスカバリの実行中、次の警告メッセージが通信ログに表示されます。

Host '<名前>': cannot find UUID, Host is skipped

**解決策:** ESX サーバのUUIDはESX Server CIのキー属性です。有効なUUIDなしでESXサーバをレポートすることはできません。ゼロだけで構成されたESXサーバのUUIDも無効とみなされます。通信ログ内のメッセージは、指定したESXサーバは検出されましたが、UUIDがないか無効であるためにスキップされたことを示します。

- **問題:** ディスカバリの実行中、次の警告メッセージが通信ログに表示されます。

VM '<名前>': duplicate host key '<キー>' found in another VM '<名前>' which was preferred, VM is skipped

**解決策:** すべての仮想マシンの検出後、重複するホスト・キーを含む仮想マシンは除外されます。**host\_key** は仮想マシンのキー属性なので、同じホスト・キーを持つ2台の仮想マシンをレポートすることはできません。通信ログ内のメッセージは、重複が見つかったこと、および重複する仮想マシンのどちらかがスキップされたことを示します。

**reportPoweredOffVMs** パラメータが **true** に設定されている場合、パワー・ステータスが異なる仮想マシンが2台あると、パワーオン状態の仮想マシンの方がパワーオフ状態の仮想マシンより優先されます。



# VMware VMotion ディスカバリおよびイベント 追跡

## 本章の内容

概要 .....	254
サポートされるバージョン .....	254
VMware VMotion の検出方法とイベントの追跡方法 .....	254
VMware vMotion Monitor by VIM ジョブ .....	255

## 概要

VMware VMotion は、実行中の仮想マシンをあるサーバから別のサーバに瞬時に移動するテクノロジーです。VMware vCenter Server は管理インタフェースを公開しており、DFM はこれを使用して次のことを実行できます。

- VIM プロトコルを使用して vCenter Server に接続し、トポロジ(データセンタ、クラスタ、ESX サーバ、リソース・プール、仮想マシンなど)を検出します。
- ESX サーバに接続して、フル・トポロジを検出します。この場合、検出対象となるのはサーバのみです。
- インベントリ構造で発生するイベントをリッスンします。現在追跡およびレポートされるのは、次の2種類のイベントです。
  - VMotion イベント(サーバ間で仮想マシンを移行する場合)
  - 仮想マシンのパワーオン・イベント(仮想マシンがパワーオン状態の場合)

VMware は、このインタフェースを記述するための SDK(ドキュメント、API リファレンス、ライブラリ、サンプルを含む)を提供しています。VMware Infrastructure SDK は、<http://www.vmware.com/support/developer/vc-sdk/> からダウンロードできます。

## サポートされるバージョン

このディスカバリでは次がサポートされます。

- VirtualCenter 2.5, 2.0
- vCenter Server 4, 4.1, 5.0, 5.1, 5.5, 6.0
- ESX Server 3.0, 3.5, 4.0, 4.1, 5.0, 5.1, 5.5, 6.0

## VMware VMotion の検出方法とイベントの追跡方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

VIM プロトコルを使用したサーバに接続するには、次を準備します。

- 接続 URL(例 : <https://vcserver/sdk>)
- ユーザ名とパスワードの資格情報 (VMware サーバで自分のユーザ・アカウントを作成する必要がある)

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - 権限の設定

VMotion イベントによるディスカバリには、使用するプロトコル用の特別な権限が必要です。

- **System.Read** 権限は、すべての DFM アクションで、ユーザがログインを実行するために必要です。ユーザは **Read-Only** ユーザ・グループのメンバーである必要があります。

## 3. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

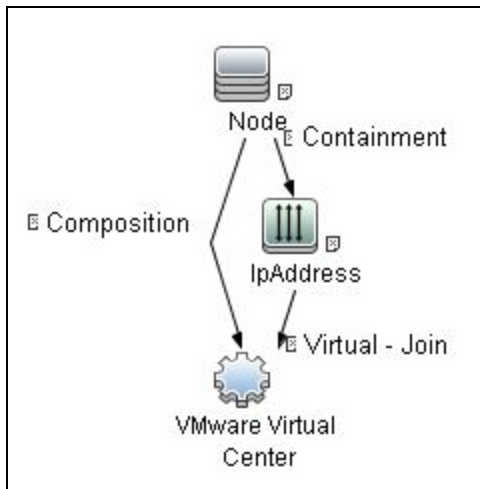
- a. **VMware vCenter Connection by VIM** ジョブと **VMware vCenter Topology by VIM** ジョブを実行します。
- b. **VMware vMotion Monitor by VIM** ジョブを実行します。このジョブには、VirtualCenter Server によって収集された仮想マシンの移行イベントをリッスンする **VMware\_vMotion\_discovery\_by\_VIM** アダプタが含まれています。

# VMware vMotion Monitor by VIM ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ

- トリガ CI : VMware VirtualCenter
- トリガ・クエリ :



## アダプタ

このジョブでは、**VMware\_VMotion\_discovery\_by\_VIM** アダプタを使用します。

- トリガ CI データ：

名前	値	説明
<b>credentialsId</b>	<code>\${SOURCE.credentials_id}</code>	VirtualCenter 属性に保存された VIM プロトコルの資格情報 ID
<b>ip_address</b>	<code>\${SOURCE.application_ip}</code>	vCenter Server の <b>application_ip</b> から取得された IP アドレス
<b>server_url</b>	<code>\${SOURCE.connection_url}</code>	vCenter Server の <b>connection_url</b> 属性から取得された接続 URL

- アダプタ・パラメータ：

プロパティ	説明
<b>connectionRetryNumber</b>	DFM が接続の復元を試行する最大回数  標準設定：0(ゼロ)(試行回数に上限がないことを示す)
<b>eventBasedDiscoveryEnabled</b>	<b>true</b> に設定した場合に、ジョブをアクティブ化するたびにターゲット・マシンに接続し、ジョブが停止するまで VMotion イベントを

プロパティ	説明
	<p>リスンするようになるパラメータ</p> <p><b>標準設定 : true</b></p>
<b>historyHours</b>	<p>未追跡のVMotion イベントを DFM がチェックする期間を示し、DFM は、ジョブがアクティブ化された時点から過去に遡って期間を算出</p> <p><b>標準設定 : 24 時間</b></p>
<b>remoteJVMArgs</b>	リモート・プロセスに渡される必要がある JVM パラメータ
<b>remoteJVMClasspath</b>	<p>外部の java プロセスで使用するクラス・パス</p> <p><b>注:</b> このプロパティの標準設定は変更しないでください。</p>
<b>runInSeparateProcess</b>	<p><b>true</b> にすると、外部の java 仮想マシンでのジョブ実行が可能になります。</p> <p><b>標準設定 : true</b></p> <p><b>注:</b> このプロパティの標準設定は変更しないでください。</p>

## 検出された CIT

- Composition
- Containment
- ExecutionEnvironment
- Interface
- IpAddress
- Node
- Usage

- VMware Host Resource
- VMware Port Group
- VMware Virtual Switch
- Virtualization Layer Software

# VMware ディスカバリのトラブルシューティングおよび制限事項

## 本章の内容

トラブルシューティング .....	259
制限事項 .....	259

## トラブルシューティング

- **問題** : 次のエラー・メッセージが表示されます。

Required class %s not found.Verify VMware SDK jar files are present in probe.See documentation for details.

**原因** : SDK \*.jar ファイルが Data Flow Probe にコピーされていません。

**解決策** : 「[VMware の前提条件](#)」(212ページ)の説明に従って \*.jar ファイルをコピーします。

- **問題** : 次のエラー・メッセージが表示されます。

User does not have required 'System.Read' permission

**原因** : DFM から ESX サーバの vCenter Server に接続する際の、ユーザ・アカウントの権限が十分ではありません。

**解決策** :

- VMware Infrastructure Management(VIM)プロトコルの資格情報が正しい優先順位で定義されていることを確認します。完全な権限のある資格情報の方が、権限レベルの低い資格情報よりインデックス番号が小さくなります。詳細については、『HP Universal CMDB Data Flow Management Guide』の「Index」を参照してください。
- DFM が以前に、完全な権限より低い権限を持つ資格情報を使用して接続を検出している場合は、接続ジョブを再実行する必要があります。
  - ESX の接続とトポロジの場合 : **VMware ESX Connection by VIM** を実行して ESX サーバの credentials ID 属性を更新してから、トポロジ・ジョブ **VMware ESX Topology by VIM** を実行します。
  - vCenter トポロジの場合 : 統合ポイントを編集し、よりレベルの高い権限を持つ資格情報を選択します。

## 制限事項

- DFM は、ライセンスの合計数と、各機能についての使用可能ライセンスを検出できます。ただし、これが可能なのは、ユーザに **Global.Licenses** 権限がある場合のみです。この権限がユーザにない場合、License Feature CI のこれらの属性はポピュレートされません。
- ESX サーバのバージョン(バージョン 3.0 と 3.5)ごとに、esxFull 機能についてレポートされる feature\_is\_edition フラグが異なります。古いバージョンでは false としてレポートされ、新しいバージョンでは

true としてレポートされます。この違いにより、DFM はこの属性をレポートしません。

- ESX サーバのバージョン(バージョン 3.0 と 3.5)ごとに、esxFull エディション・ライセンスに含まれる ESX 固有の機能(nas, iscsi, vsmp, san)についてレポートされるライセンスの合計数と使用可能数が異なります。こうした機能について、DFM はこれらの属性をレポートしません。
- VMware プロトコルのバージョン間には差異があります。特定の属性が新しいバージョンにのみ表示され、以前のバージョンには表示されません。その結果、古いプロトコルを使用している場合に、特定の属性(特にクラスタとライセンスについての属性)を検出できません。
- DFM は、バージョン 4.0 以降の vCenter\ESX サーバに関するライセンス情報について検出やレポートを実行しません。
- DFM は、チーミング・インタフェースの順序に関する情報をレポートしません。ESX サーバのサーバ物理インタフェースは、NIC チーミング・グループにグループ化できます。また、グループ内でのインタフェースの順序も指定できます(1 番目、2 番目など)。チーミングするインタフェースの情報はレポートされますが、インタフェースの順序はレポートされません。



## 第12章: Xen and KVM ディスカバリ

### 本章の内容

概要 .....	262
サポートされるバージョン .....	262
トポロジ .....	262
ディスカバリ・メカニズム .....	263
KVM および Xen の検出方法 .....	264
Xen and KVM by Shell アダプタ .....	264
Xen and KVM by Shell ジョブ .....	267

## 概要

Xen は、同じコンピュータ・ハードウェア上での複数コンピュータ・オペレーティング・システムの並列稼働を実現するサービスを提供するハイパーバイザです。Xen は現在 IA-32, x86-64, ARM コンピュータ・アーキテクチャで使用できます。

カーネル・ベースの仮想マシン(KVM)は、Linux カーネル用の仮想化インフラストラクチャです。KVM は、ハードウェア仮想化拡張機能を使用することで、プロセッサ上でのネイティブの仮想化をサポートします。KVM は読み込み可能なカーネル・モジュールとして、FreeBSD および Illumos にも移植されています。

元々 KVM は x86 プロセッサおよび x86-64 プロセッサ向けにサポートされており、S/390, PowerPC, IA-64 に移植されていました。ARM の移植が進行中であり、KVM ハイパーバイザの ARM Cortex-A15 への移植が仮想オープン・システムによって実現しています。

オープン・ソース API である libvirt は、プラットフォームの仮想化を管理するためのデーモン兼管理ツールです。Linux KVM, Xen, VMware ESX, その他の仮想化テクノロジーの管理に使用できます。この API は、仮想マシン・マネージャなどのグラフィカル・インタフェースで使用できるだけでなく、コマンド・ライン・インタフェース(virsh)や、oVirt などの上位のツールでも使用できます。

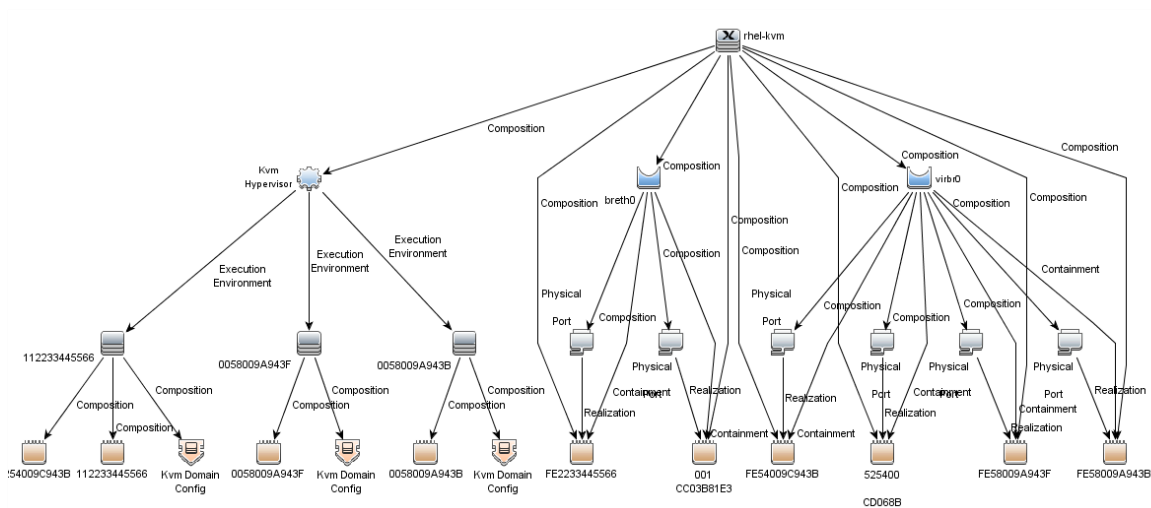
このパッケージは、シェル接続を介して libvirt を API として使用することによって、Xen および KVM 仮想化ソリューションを検出します。

## サポートされるバージョン

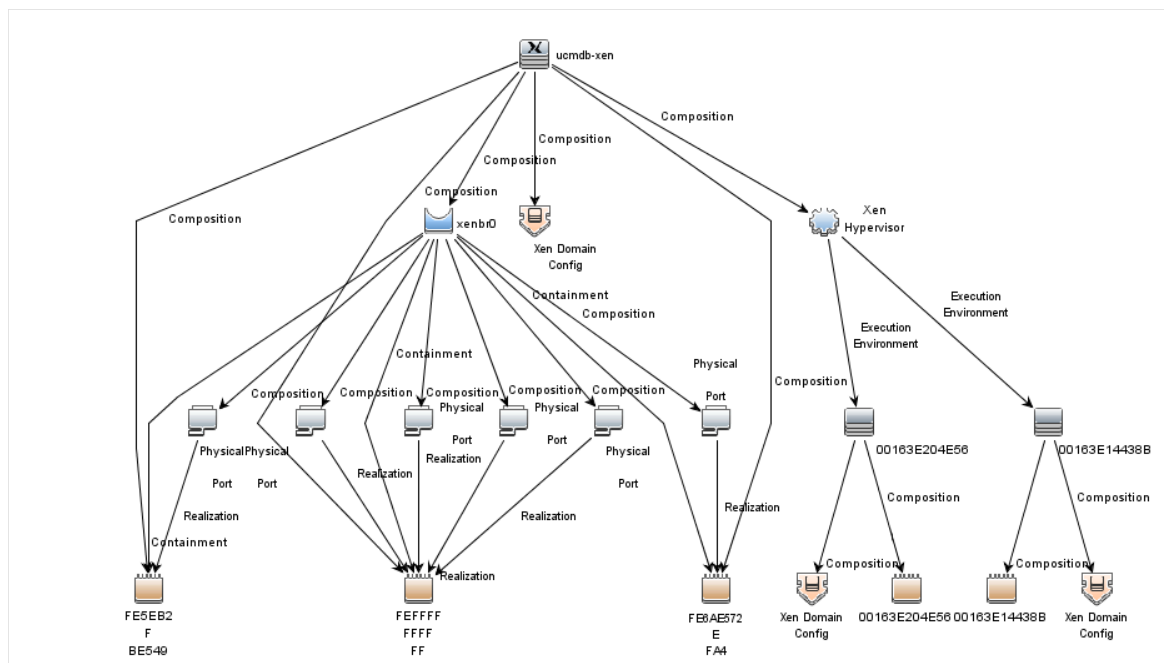
libvirt には、Xen および KVM を管理するための統合 API が用意されています。このディスカバリは、libvirt バージョン 0.8.2 から 0.8.x とバージョン 0.9.1 から 0.9.6 をサポートしています。

## トポロジ

次の図に、KVM ディスカバリのトポロジを示します。



次の図に、Xen ディスカバリのトポロジを示します。



## ディスカバリ・メカニズム

シェルによる通常の接続は、Xen または KVM が稼働する対象マシンに対し、インストールされた libvirt 管理ツールを介して実行されます。libvirt CLI を使用することで、対象マシンに関する詳細を検出できます。

## KVM および Xen の検出方法

このディスカバリは、次の段階で実行されます。

- 汎用 Linux ホストの検出
- XEN および KVM の仮想化の検出

### 1. 前提条件 - プロトコル資格情報の設定

SSH プロトコルまたは Telnet プロトコルを設定する必要があります。どちらのプロトコルの場合も、ユーザ名とパスワードを準備する必要があります。

### 2. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行し、IP CI を検出します。
- b. **Host Connection by Shell** ジョブを実行し、ターゲット・ホストおよびそのホストへのシェル接続を検出します。
- c. **Xen and KVM by Shell** ジョブを実行し、Xen および KVM のトポロジを検出します。

## Xen and KVM by Shell アダプタ

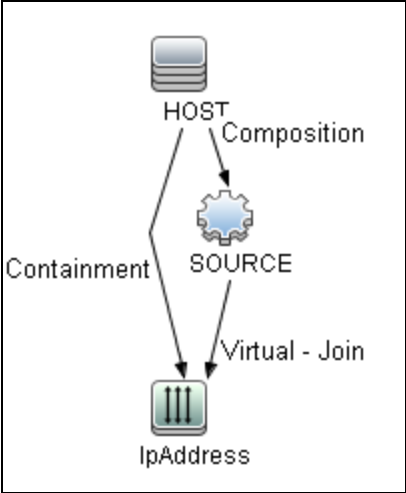
このディスカバリでは、Xen and KVM by Shell アダプタが使用されます。

### 入力 CIT

Shell

### 入力 TQL クエリ

次の図は、このジョブの入力 TQL クエリを示しています。



属性	条件
HOST	NOT CI Type 等しい nt
IpAddress	NOT IP Probe Name null

トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${SOURCE.root_container}
ip_address	\${SOURCE.application_ip}

使用スクリプト

xen\_by\_tty.py

検出された CIT

- Bridge
- Composition
- Containment
- ExecutionEnvironment
- FileSystem
- FileSystemExport
- Interface
- Kvm domain config
- Layer2Connection
- Node
- PhysicalPort
- Realization
- Virtualization Layer Software
- XenDomainConfig

パラメータ

パラメータ	タイプ	説明
virsh_path	文字列	virsh 管理ユーティリティへのフルパスです。

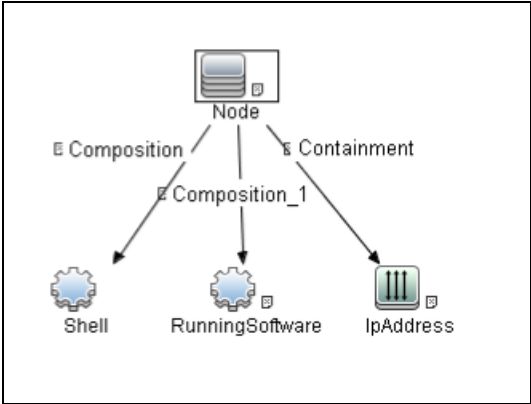
## Xen and KVM by Shell ジョブ

本 項 ではこのジョブに関する詳 細を説 明します。

### アダプタ

このジョブでは、Xen and KVM by Shell アダプタが使用されます。

### トリガ・クエリ



### パラメータ

パラメータ	タイプ	説 明
virsh_path	文 字 列	virsh 管 理 ユーティリティへのフル・パスです。

### ディスカバリ・フロー

#### 1. ドメイン・リストの検出

**virsh list** のサンプル出 力は次のとおりです。

Id	Name	State
-----		
0	Domain-0	running
15	ucmdb-vm-vista	idle
16	ucmdb-vm-xp	idle

## 2. ドメイン構成リストの検出

次のコマンドでは、構成を使用して xml ファイルが作成されます。

**virsh dumpxml <domain\_name>**

### a. 特徴的なハイパーバイザ:

Xen : <domain type='xen' id='15'>

KVM : <domain type='kvm' id='12'>

### b. 仮想マシン・メモリ構成

Xen および KVM : <memory>1572864</memory>

<currentMemory>1572864</currentMemory>

### c. 仮想マシン CPU 構成

Xen および KVM : <vcpu>1</vcpu>

### d. 仮想マシン・ネットワーク構成

Xen および KVM : <interface type='bridge'><mac address='00:16:3e:20:4e:56'><source  
bridge='xenbr0'><script path='vif-bridge'><target dev='vif15.0'></interface>



## 第3部: Clustering and Load Balancing > Failover Clusters

## 第13章: EMC AutoStart ディスカバリ

### 本章の内容

概要 .....	271
サポートされるバージョン .....	271
トポロジ .....	271
EMC AutoStart の検出方法 .....	273
EMC AutoStart by Shell ジョブ .....	274
アダプタ .....	274
トリガ・クエリ .....	274
パラメータ .....	275
EMC_AutoStart_by_Shell アダプタ .....	276
ディスカバリ・フロー .....	279
EMC AutoStart ディスカバリ・コマンド .....	281

## 概要

EMC AutoStart は、複数のオペレーティング・システムでのハイアベイラビリティ(高可用性)を実現し、計画的または計画外のサービス停止に対応します。

EMC AutoStart のディスカバリ・プロセスにより、フル・トポロジを検出できます。

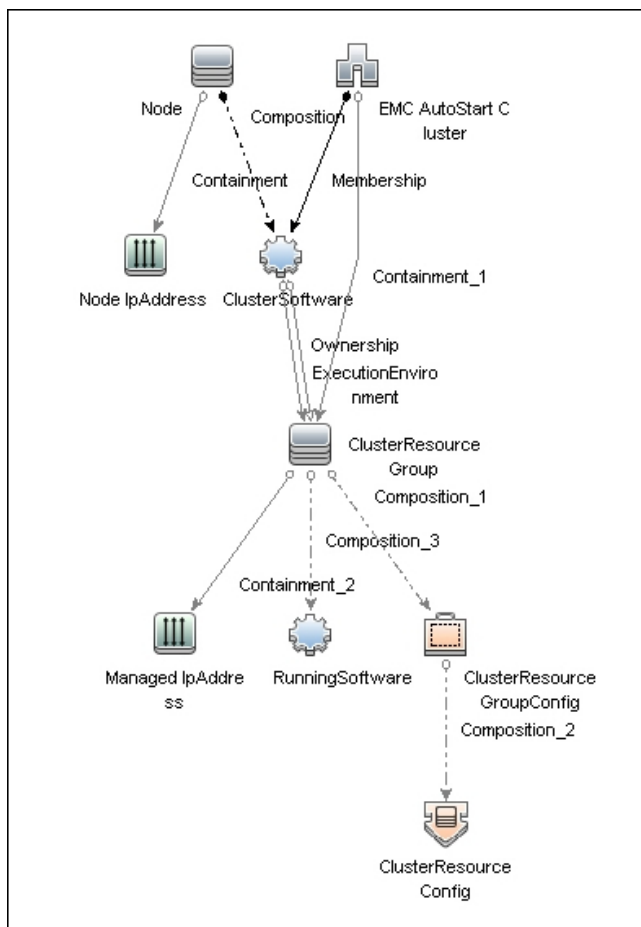
## サポートされるバージョン

EMC AutoStart ディスカバリは、EMC AutoStart のバージョン 5.x をサポートしています。

## トポロジ

次の図に、EMC AutoStart トポロジを示します。

注: 検出された CIT のリストについては、「[検出された CIT](#)」(277 ページ)を参照してください。



## EMC AutoStart の検出方法

### 本項の内容

#### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、次のプロトコルを使用します。

- **SSH**
- **Telnet**
- **NTCMD**

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

#### 2. 前提条件 - その他

- a. AutoStart ドメインの 1 つ以上のノードにシェル接続できることを確認します。
- b. 必要に応じて、使用される全コマンドの実行を許可する sudo を各ターゲット・ホストで構成します。[「EMC AutoStart ディスカバリ・コマンド」\(281ページ\)](#)を参照してください。
- c. Windows で **NTCMD** を使用して接続する場合は、AutoStart で認識されるユーザとして **HPCmdSvc** サービスを実行します。そうしないと、構成情報を使用できません。

#### 3. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by Shell** ジョブを実行して、ターゲット・ホストとそれに対するシェル接続を検出します。
- c. **Host Applications by Shell** ジョブを実行して、ターゲット・ホストのアプリケーション(EMC AutoStart Cluster ソフトウェアとエージェント・プロセスを含む)を検出します。
- d. **EMC AutoStart by Shell** ジョブを実行して、ターゲットの EMC AutoStart クラスタのトポロジを検出します。

## EMC AutoStart by Shell ジョブ

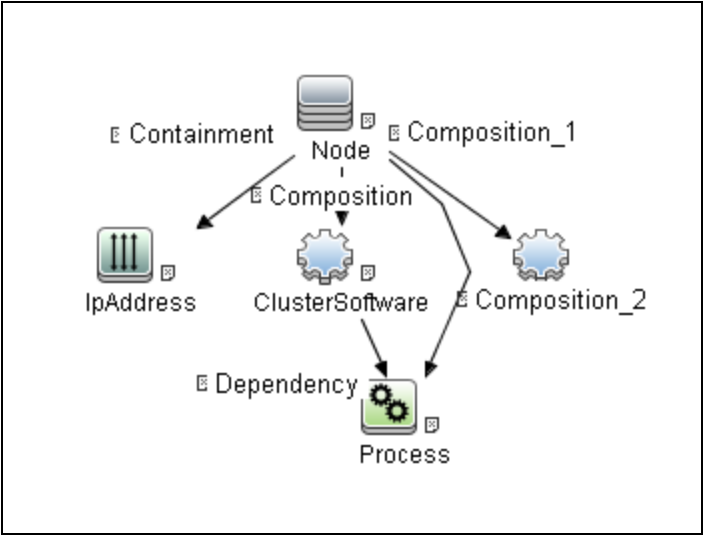
本項では、EMC AutoStart by Shell ジョブに関する詳細を説明します。

### アダプタ

このジョブでは、**EMC\_AutoStart\_by\_Shell** アダプタを使用します。

### トリガ・クエリ

**emc\_autostart\_with\_shell**



ノード名	条件
Shell	NOT Reference to the credentials dictionary entry null AND NOT Application IP null
IpAddress	NOT IP Probe Name null
Process	NOT Process Path null AND Name 類似 "ftAgent%"
ClusterSoftware	DiscoveredProductName 等しい EMC AutoStart Cluster SW AND NOT Name null
Node	なし

## パラメータ

このジョブは、アダプタのパラメータ値を使用します。標準設定では、パラメータは上書きされません。

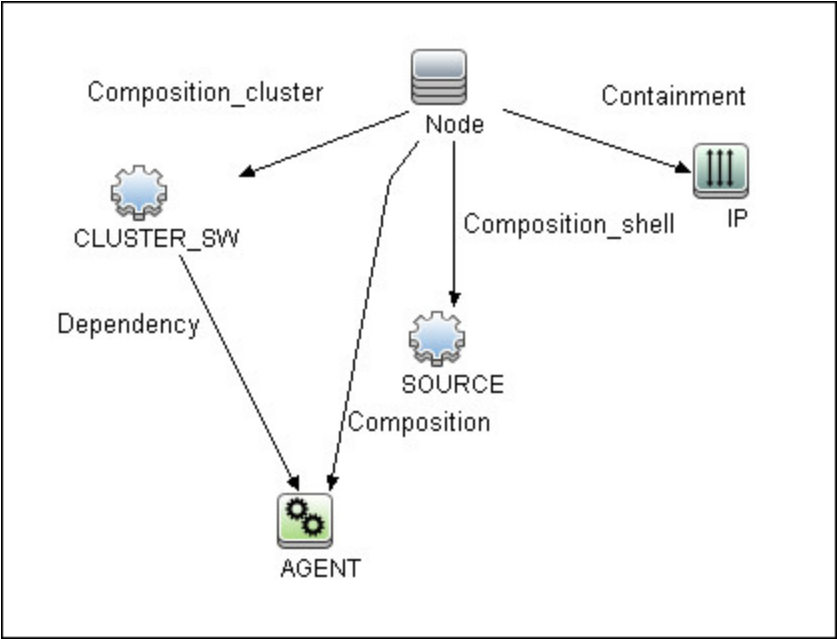
## EMC\_AutoStart\_by\_Shell アダプタ

本項では EMC\_AutoStart\_by\_Shell アダプタに関する詳細を説明します。

### 入力 CIT

Shell

### 入力 TQL クエリ



ノード名	条件
SOURCE	NOT Reference to the credentials dictionary entry null AND NOT Application IP null
IP	NOT IP Probe Name null
AGENT	NOT Process Path null AND Name 類似 "ftAgent%"
CLUSTER_SW	DiscoveredProductName 等しい EMC AutoStart Cluster SW AND NOT Name null
Node	なし



## トリガ CI データ

名前	値
<b>Protocol</b>	\${SOURCE.root_class}
<b>agentPath</b>	\${AGENT.process_path}
<b>connected_os_credentials_id</b>	\${SOURCE.connected_os_credentials_id:NA}
<b>credentialsId</b>	\${SOURCE.credentials_id}
<b>domainName</b>	\${CLUSTER_SW.name}
<b>ip_address</b>	\${SOURCE.application_ip}

## 使用スクリプト

- emc\_autostart.py
- emc\_autostart\_discover.py
- emc\_autostart\_report.py
- emc\_autostart\_by\_shell.py

## 検出された CIT

- ClusterResourceConfig
- ClusterResourceGroup
- ClusterResourceGroupConfig
- ClusterSoftware
- Composition
- Containment
- EMC AutoStart Cluster
- ExecutionEnvironment

- IpAddress
- Membership
- Node
- Ownership

## ディスカバリ・フロー

本項では、**EMC Autostart by Shell** ジョブのディスカバリ・フローを説明します。

### 1. パスの計算

**アプリケーション署名** で検出された **ftAgent** プロセスのパスが解析されます。計算されるパスは次のとおりです。

- デプロイメントのルート
- 実行可能ファイルがあるフォルダへのパス(bin)

### 2. ftcli コマンドの存在確認

次の目的で、**ftcli** コマンドに **-version** 引数を指定して実行します。

- 計算したパスを使って、コマンドが使用可能であることを確認します。
- インストールされている EMC AutoStart ソフトウェアのバージョン情報を取得します。

### 3. ドメイン名の確認

EMC AutoStart ソフトウェア・プロセスのコマンド・ラインから計算されたドメイン名を確認します。

- ジョブは、構成ファイル **<ルート>/config/<domain-name>-sites** の読み取りを試みます。
- このファイルがない場合、ドメイン名は無効とみなされ、ジョブは終了します。

### 4. クラスタ・トポロジの検出

コマンド **ftcli** を使用して、次を含むクラスタ構成を読み取ります。

- ノード (listNodes, getNode)
- 管理対象 IP (listManagedIPS, getIP)
- 管理対象 NIC (listManagedNics, getNic)

- **リソース・グループ**(listResourceGroups, getResourceGroup)
- **データ・ソース**(getDataSource)
- **プロセス**(getProc)

## EMC AutoStart ディスカバリ・コマンド

本項では、EMC AutoStart ディスカバリで使用するコマンドについて説明します。

### コマンド `ftcli.exe -version`

```
"C:\Program Files\EMC\AutoStart\DDM_dom\bin\ftcli.exe" -version
```

#### 出力

```
Version 5.4.1 Build 82
```

```
                EMC AutoStart
              Version 5.4.1 build 82
        Built:Thu Nov 3 16:09:59 EDT 2011
```

### コマンド `ftcli.exe -cmd "listNodes"`

```
"C:\Program Files\EMC\AutoStart\DDM_dom\bin\ftcli.exe" -cmd "listNodes"
```

#### 出力

Node	Type	State
-----	-----	-----
ddm-autostart	Primary	Agent Running
ddm-autostart2	Primary	Agent Running

### コマンド `ftcli -cmd "getNode node1"`

```
/opt/EMCas/bin/ftcli -cmd "getNode node1"
```

#### 出力

```
Description      :Entry for node node1
System Name      : node1
Operating System  :HP-UX 11.31
Kernel Arch      : ia64
Main Memory (MB) :4076
Swap space (MB)  :24506
Supported DS      :
IP Address(es)   :10.20.30.136
                  10.20.30.137
Node Attributes   : name=Ticket                      value=1
LAAM Version      :5.4.1
LAAM Version Info :Version 5.4.1 build 82
```

Build Date	:Thu Nov 3 16:09:30 EDT 2011
State	:Agent Running

## 第14章: IBM High Availability Cluster Multiprocessing(HACMP)ディスカバリ

### 本章の内容

概要 .....	284
サポートされるバージョン .....	284
トポロジ .....	285
IBM HACMP の検出方法 .....	286
ディスカバリ・メカニズム .....	288
HACMP Topology Discovery ジョブ .....	294
HACMP Application Discovery ジョブ .....	295

## 概要

HACMP(High Availability Cluster Multiprocessing)は、IBM System p プラットフォームを対象とした、AIX UNIX および Linux でのハイアベイラビリティ・クラスタ用 IBM ソリューションです。

HACMP は、アプリケーションをアクティブに実行しているか(アクティブ)、障害が発生したノードからの引き継ぎを待機している(パッシブ)、最大 32 台のコンピュータまたはノードで実行できます。ファイル・システム上のデータは、クラスタ内のシステム間で共有できます。

HACMP は、IBM の RSCT(Reliable Scalable Cluster Technology)に大きく依存しています。RSCT には、クラスタ状態(ノード、NIC、またはネットワークのクラッシュなど)の監視と、これらのイベントに対する応答の調整を担当するデーモンが含まれます。HACMP は RSCT 対応クライアントです。RSCT は AIX を使用して分散されます。

**IBM\_HACMP** パッケージは、TTY(SSH または Telnet プロトコル)を介して AIX 上の HACMP を検出します。このパッケージは、ディスカバリ・モデルに従って HACMP トポロジ(構成済みのネットワーク、ノード・インタフェース(パブリック TCP/IP とシリアル・ハートビートの両方)、サービス IP)と、アプリケーション・リソース(構成済みのリソース・グループ、アプリケーション・サーバ、ボリューム・グループ)を検出します。また、構成済みのパブリック・インタフェースを UCMDB IP に、シリアル・インタフェースを UCMDB ホストの下のディレクトリにマッピングするほか、ボリューム・グループを UCMDB ホストの下の論理ディスクに、そしてアプリケーション・リソースをトポロジにマッピングします。

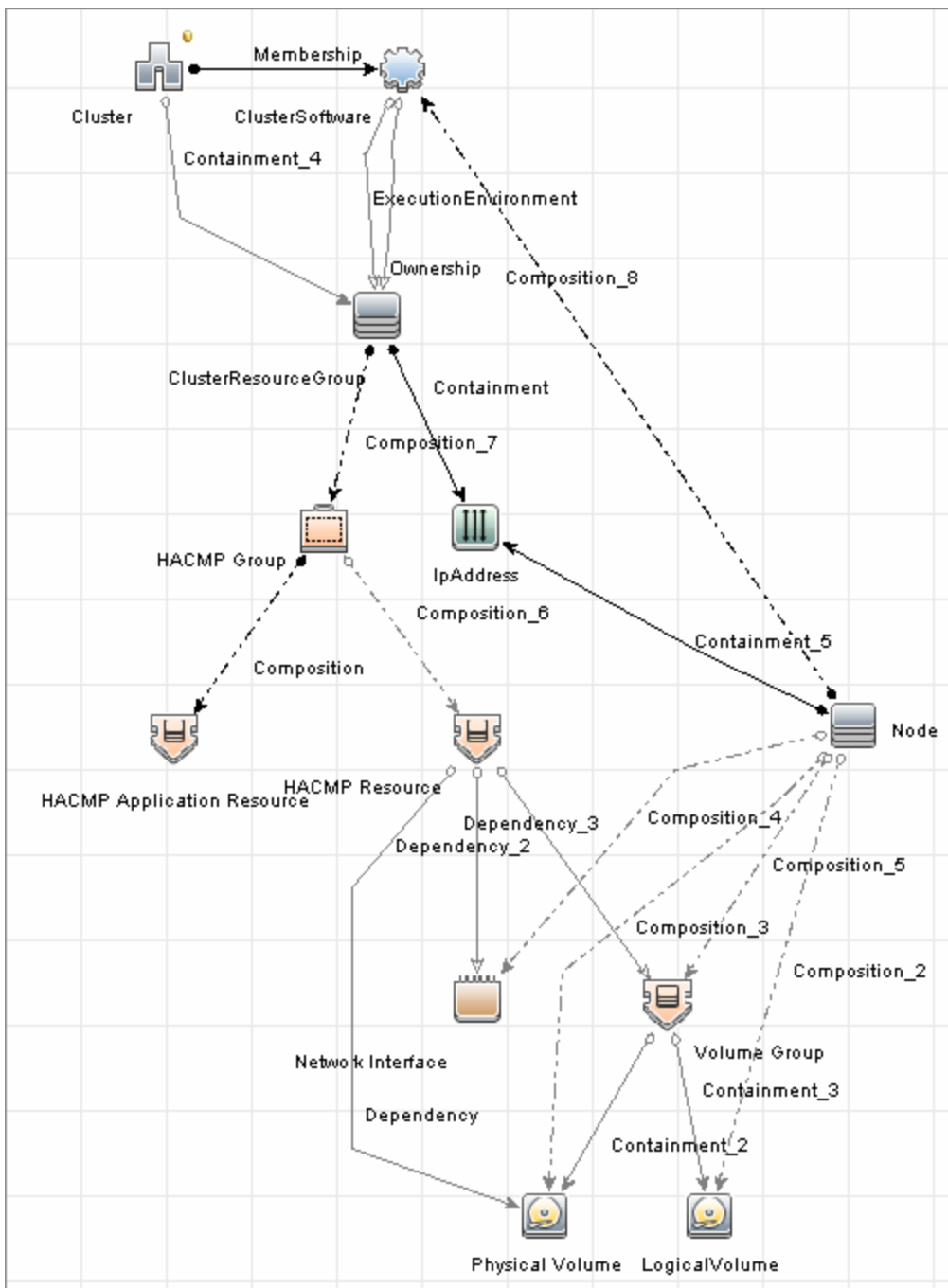
## サポートされるバージョン

このディスカバリは、AIX 5.3 での HACMP 5.4 をサポートしています。



## トポロジ

次の図に、HACMP ディスカバリのトポロジを示します。



## IBM HACMP の検出方法

### 本項の内容

#### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、次のシェル・プロトコルが使用されます。

- SSH プロトコル
- Telnet プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

#### 2. 前提条件 - その他

- クラスタに関与するノードで **Host Connection** ジョブが正常に実行されていることを確認します。

詳細については、[「ネットワーク - 基本 ディスカバリ」\(965ページ\)](#)を参照してください。

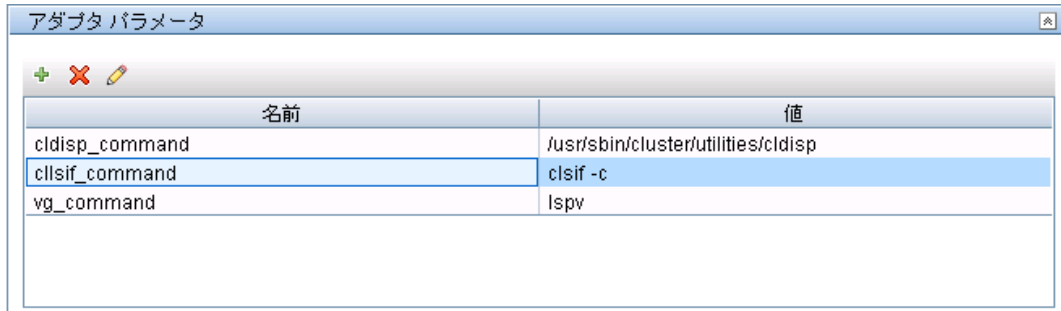
- HACMP パッケージをデプロイする前に、ストレージ・トポロジのアドオン・パッケージをロードします。

#### 3. ディスカバリの実行

- プローブに IP 範囲 (IBM HACMP Cluster を実行しているターゲット・マシンの IP を含む) が割り当てられていることを確認します。
- シェル (SSH または Telnet) の資格情報が指定されていることを確認します。詳細については、[「前提条件 - プロトコル資格情報の設定」\(286ページ\)](#)を参照してください。
- Range IPs by ICMP** ジョブを実行し、IP 範囲内で稼働中のマシンを検出します。
- Host Connection by Shell** ジョブを実行し、シェル接続と、ホストに関する基本情報を検出します。
- Host Connection** ジョブが、HACMP クラスタの一部とするホストを以前に検出したことがあるかどうかを確認します。詳細については、[「前提条件 - プロトコル資格情報の設定」\(286ページ\)](#)を参照してください。これらのジョブを実行したことがない場合は、ここでアクティブ化できます。
- TTY\_HACMP\_Applications** アダプタと **TTY\_HACMP\_Topology** アダプタのアダプタ・パラメータを確認します。コマンドで **sudo** を使用するため、パラメータを適宜調整します。これらはジョブ

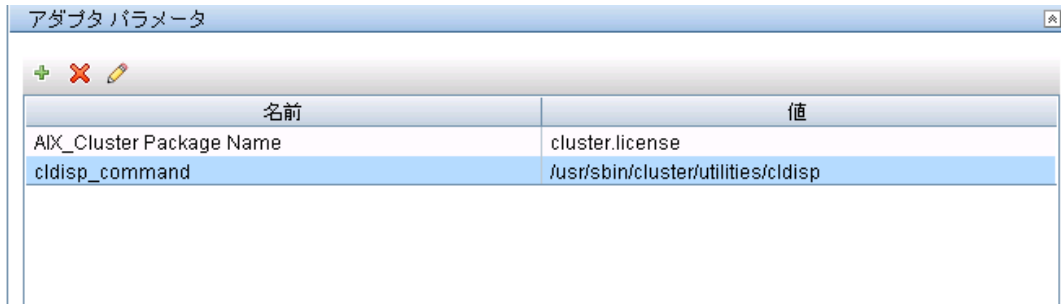
で調整することも可能です。

#### TTY\_HACMP\_Applications アダプタ



名前	値
cldisp_command	/usr/sbin/cluster/utilities/cldisp
clicif_command	clicif -c
vg_command	lspv

#### TTY\_HACMP\_Topology アダプタ



名前	値
AIX_Cluster Package Name	cluster.license
cldisp_command	/usr/sbin/cluster/utilities/cldisp

- g. **HACMP Topology Discovery** ジョブをアクティブにします。ジョブが完了したら, **HACMP** CI が作成されていることを[ディスカバリ結果]表示枠で確認します。検出された CI の詳細については、『HP Universal CMDB Data Flow Management Guide』で, ディスカバリの進行状況と結果について解説した項を参照してください。
- h. **HACMP Application Discovery** ジョブをアクティブにします。このジョブは, HACMP アプリケーションとリソース CI を作成します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## ディスカバリ・メカニズム

本項では、次のコマンドについて説明します。

- [「接続した OS が HACMP をサポートしていることの確認」\(288ページ\)](#)
- [「HACMP のバージョンの取得」\(288ページ\)](#)
- [「クラスタ情報の取得」\(289ページ\)](#)
- [「ホスト・ファイルからの DNS 情報の取得」\(289ページ\)](#)
- [「ボリューム・グループ情報の取得」\(290ページ\)](#)
- [「HACMP アプリケーション情報の取得」\(291ページ\)](#)

### 接続した OS が HACMP をサポートしていることの確認

コマンド	uname
出力例	aix
取得された値	aix
コメント	このコマンドは OS を取得します。このパッケージは、AIX プラットフォームでのみ実行するため、ディスカバリでは OS を確認する必要があります。

### HACMP のバージョンの取得

コマンド	lspp -l cluster.license
出力例	cluster.license 5.4.0.0 COMMITTED HACMP Electronic License
取得された値	5.4.x.x
コメント	このコマンドは HACMP のバージョンを提供します。ディスカバリは、HACMP のバージョンが有効であることを確認します。

## クラスタ情報の取得

コマンド	/usr/sbin/cluster/utilities/cldisp
出力例	## ===== ## Cluster: db590_db591 ## Cluster services: active ## State of cluster: up ## Substate: stable ## ## ##### ## APPLICATIONS ## ##### ## ... ## =====
取得された値	クラスタ : db590_db591
コメント	このコマンドは、HACMP のクラスタ名を取得します。

## ホスト・ファイルからの DNS 情報の取得

コマンド	cat /etc/hosts
出力例	## Sample output... ## ===== ## # Do not remove the following line, or various programs ## # that require network functionality will fail. ## 127.0.0.1 testserver localhost.localdomain localhost ## 12.20.30.3 server1 server1.compay.net ## 12.20.20.3 server1-backup server1-backup.company.net ## 192.168.1.103 server1-local server1-local.company.net ## 12.20.30.4 server2 server1.compay.net

	## 12.20.20.4 server2-backup server2-backup.company.net ## 192.168.1.104 server2-local server2-local.company.net ## =====
取得された値	IP アドレスと名 前
コメント	このコマンドは、ホスト名とIPを取得します。

## ボリューム・グループ情報の取得

コマンド	lspv
出力例	## Sample output... # dwscmdb : lspv # hdisk1 00ca4bbeeb84bdab4f rootvg active # hdisk0 00ca4bbeeb84bdac14 rootvg active # hdisk2 00ca4bbeeeb6b3c2 QSWIQ9A0_vg concurrent # hdisk3 00ca4bbeeeb3c581 None # hdisk4 00ca4bbeeeb6b499 QSWIQ9A0_vg concurrent # hdisk5 00ca4bbeeeb3c403 None # hdisk6 00ca4bbeeeb6b60d QSWIQ9B0_vg concurrent # hdisk7 00ca4bbeeeb3c4c2 QSWIQ9B0_vg concurrent # hdisk8 00ca4bbeeeb6b84f QSWIQ9A0_vg concurrent # hdisk9 00ca4bbeeeb6b920 QSWIQ9A0_vg concurrent # hdisk10 00ca4bbeeeb3c641 None # hdisk11 00ca4bbeeeb3c7c0 None # hdisk12 00ca4bbeeeb6b6e5 QSWIQ9B0_vg concurrent # hdisk13 00ca4bbeeeb3c700 QSWIQ9B0_vg concurrent
取得された値	ボリューム・グループ名
コメント	このコマンドは、ボリューム・グループを取得します。

## HACMP アプリケーション情報の取得

コマンド	cldisp
出力例	<pre> ## Sample output...  ## =====  ## Cluster: db590_db591 ## Cluster services: active ## State of cluster: up ## Substate: stable ##  ## ##### ## APPLICATIONS ## #####  ## Cluster sy008_sy015 provides the following applications: assy008  ## Application: assy008 {online} ## This application is part of resource group 'ressy008'. ## Resource group policies: ## Startup: on home node only ## Fallover: to next priority node in the list ## Fallback: never  ## Nodes configured to provide assy008: a_wwasy008 {up} b_ ddasy015 {up}  ## Node currently providing assy008: a_wwasy008 {up}  ## The node that will provide assy008 if a_wwasy008 fails is: b_ddasy015  ## assy008 is started by /usr/local/bin/start_assy008 ## assy008 is stopped by /usr/local/bin/stop_assy008 ## Resources associated with assy008: ## Service Labels ## wwasy008(141.122.74.142) {online} </pre>

```
## Interfaces configured to provide wwasy008:
## wwasy008-boot {down}
## with IP address:141.122.74.149
## on interface: en1
## on node: a_wwasy008 {up}
## on network: net_ether_01 {up}
## wwasy008-stdby {up}
## with IP address:192.168.2.40
## on interface: en2
## on node: a_wwasy008 {up}
## on network: net_ether_01 {up}
## ddasy015 {up}
## with IP address:141.122.74.154
## on interface: en1
## on node: b_ddasy015 {up}
## on network: net_ether_01 {up}
## ddasy015-stdby {up}
## with IP address:192.168.2.10
## on interface: en2
## on node: b_ddasy015 {up}
## on network: net_ether_01 {up}
## Shared Volume Groups:
## vg100
## vg199
## No application monitors are configured for assy008.
##
## #####
## TOPOLOGY
## #####
## ...
```



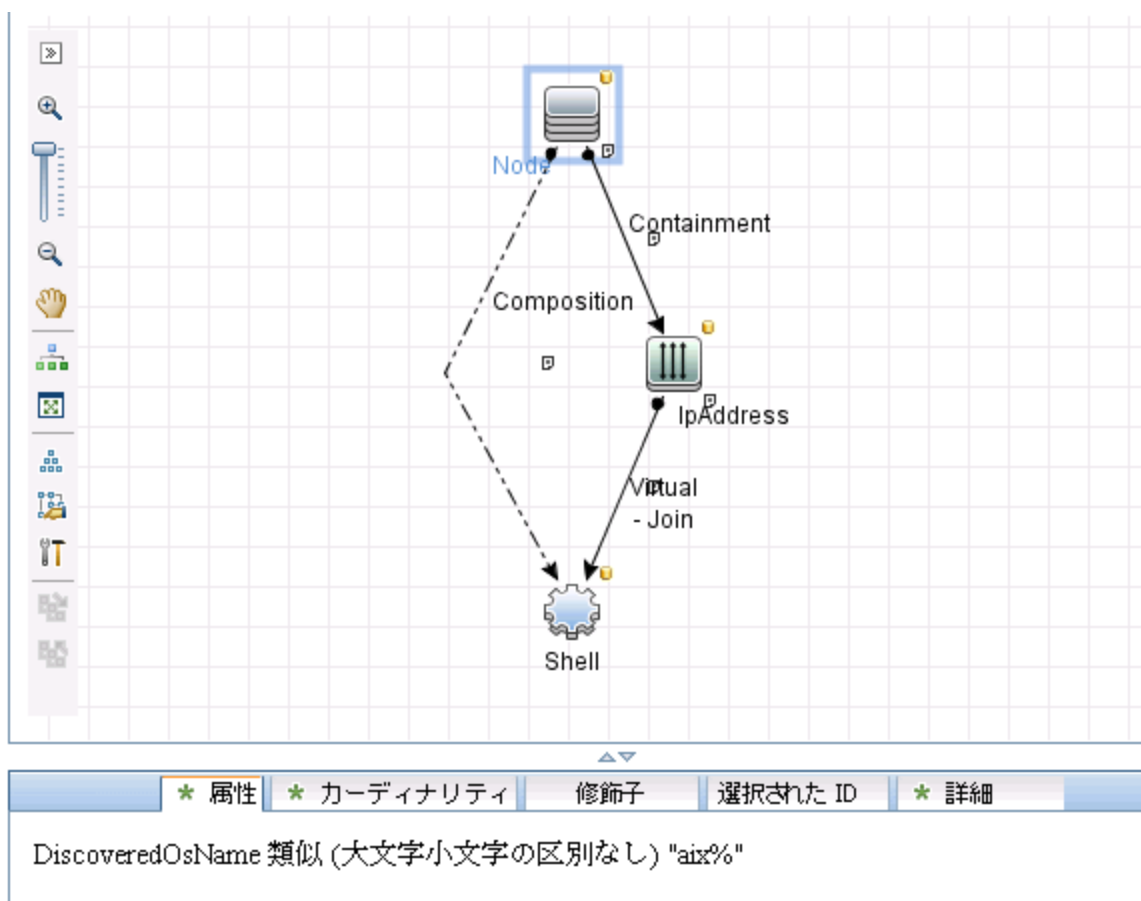
	## =====
取得された値	アプリケーション情報
コメント	このコマンドは、HACMP アプリケーション情報を取得します。

## HACMP Topology Discovery ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ(shell\_not\_ntcmd\_HACMP)

このトリガには、NTCMD シェルではない TTY シェルが必要です。



### アダプタ

### 使用スクリプト

- storage\_topology.py
- TTY\_HACMP\_Topology.py

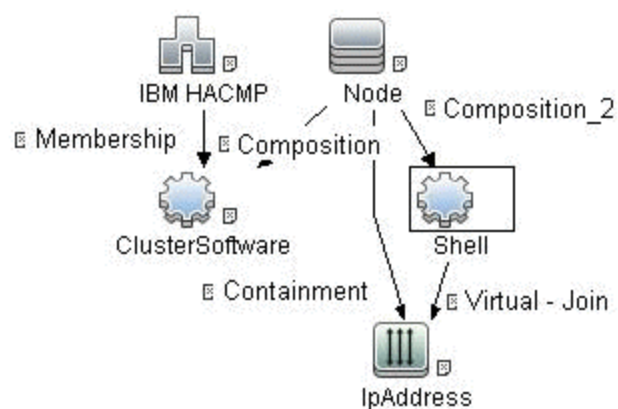
## 検出された CIT

- ClusterSoftware
- Composition
- Containment
- HACMP Cluster
- Interface
- IpAddress
- LogicalVolume
- Membership
- Node
- Physical Volume
- Volume Group

## HACMP Application Discovery ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ・クエリ(shell\_in\_HACMP\_cluster)

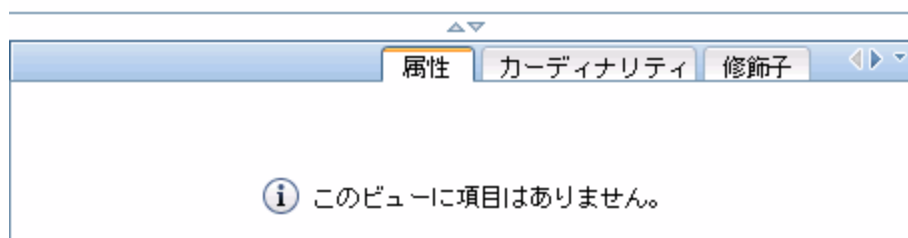
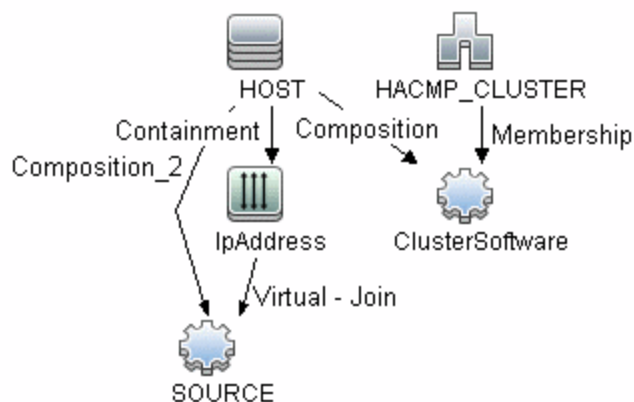


\* 属性 \* カーディナリティ 修飾子 選択された ID \* 詳細

NOT Reference to the credentials dictionary entry null

アダプタ

- ・ 入力クエリ



## 使用スクリプト

- TTY\_HACMP\_Applications.py

## 検出された CIT

- ClusterResourceGroup
- ClusterSoftware
- Composition
- Containment
- Dependency
- ExecutionEnvironment

- HACMP Cluster
- HACMP Resource
- HACMP Resource Group
- Interface
- IpAddress
- Membership
- Node
- Ownership
- Physical Volume
- RunningSoftware
- Usage
- Volume Group

## 第15章: Microsoft Cluster ディスカバリ

### 本章の内容

Microsoft Cluster Server View トポロジ .....	300
サポートされるバージョン .....	301
Microsoft Cluster Server の検出方法 .....	302
MS Cluster by NTCMD or UDA ジョブ .....	302

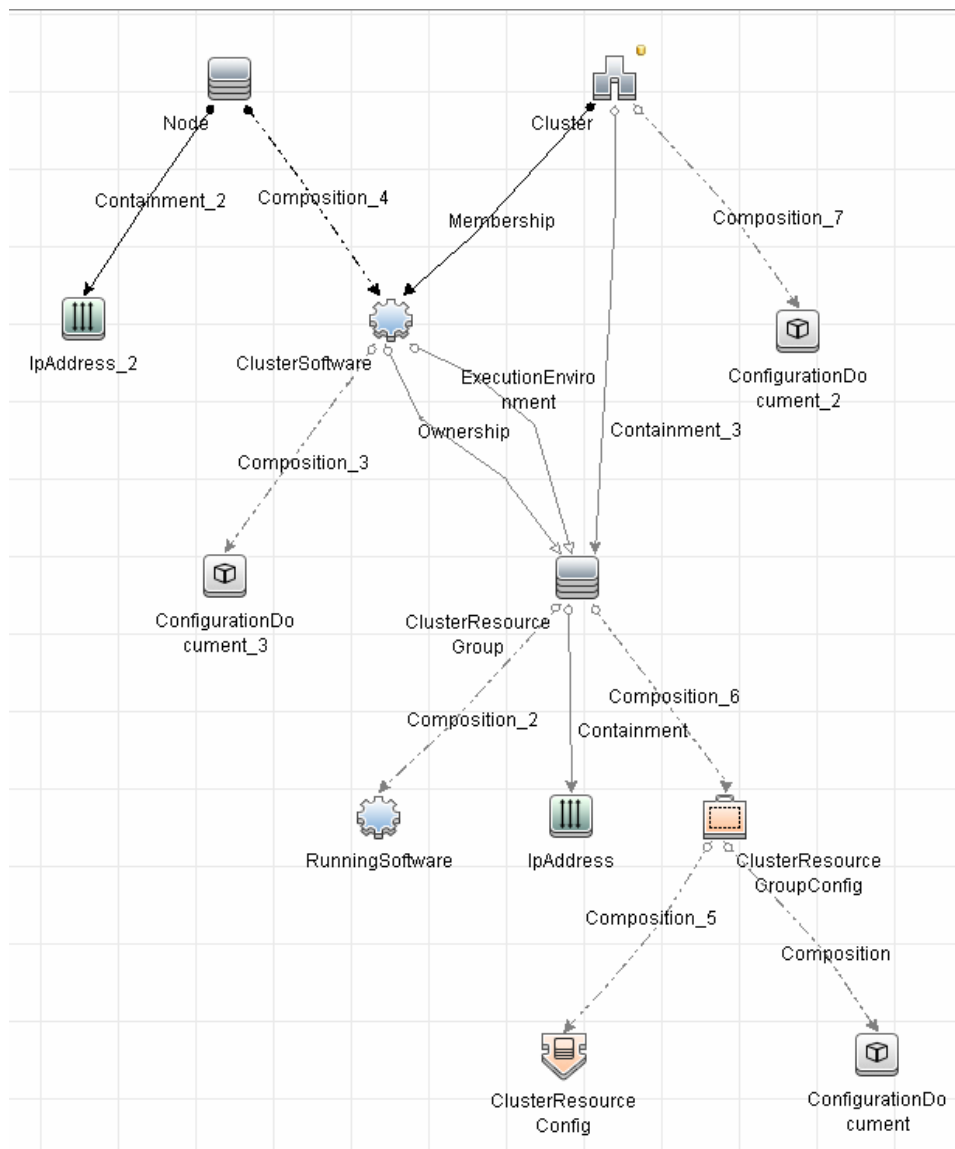
## Microsoft Cluster Server View トポロジ

Microsoft Cluster Server View には、MS Cluster と、そのメンバであるクラスタ・ソフトウェア(実際のホスト上で実行中のエージェント)が表示されます。

クラスタは、数台のクラスタ・サーバ(クラスタ・クライアントで仮想 IP を介して使用される仮想サービスのプラットフォームとなる仮想ホストまたは仮想サーバ)で構成されます。クラスタは Microsoft Cluster グループで構成され、各グループには Microsoft Cluster リソースがあります。各クラスタ・リソース・グループには、それぞれ別の専用仮想 IP が割り当てられているとみなされます。これらの IP は、クラスタ・クライアントによる使用を目的として構成されています。

注: 検出された CIT のリストについては、[「検出された CIT」\(304ページ\)](#)を参照してください。





## サポートされるバージョン

- Windows Server 2003
- Windows Server 2008
- Windows Server 2012

## Microsoft Cluster Server の検出方法

MS Cluster ディスカバリ・プロセスにより、ネットワーク上にある Microsoft Cluster Server のトポロジを検出できます。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、WMI、NTCMD、または PowerShell プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

関連するジョブを次の順序でアクティブ化します。

- a. **Host Connection by Shell** または **Host Connection by PowerShell**
- b. **Host Applications by Shell/SNMP/WMI/PowerShell** および **Host Resources by Shell/SNMP/WMI/PowerShell**
- c. **MS Cluster by NTCMD or UDA**

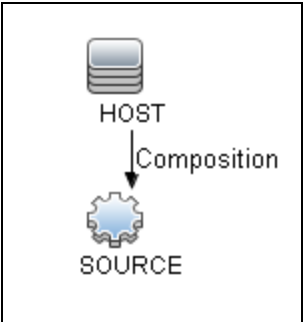
ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## MS Cluster by NTCMD or UDA ジョブ

### 入力 CI タイプ

Agent

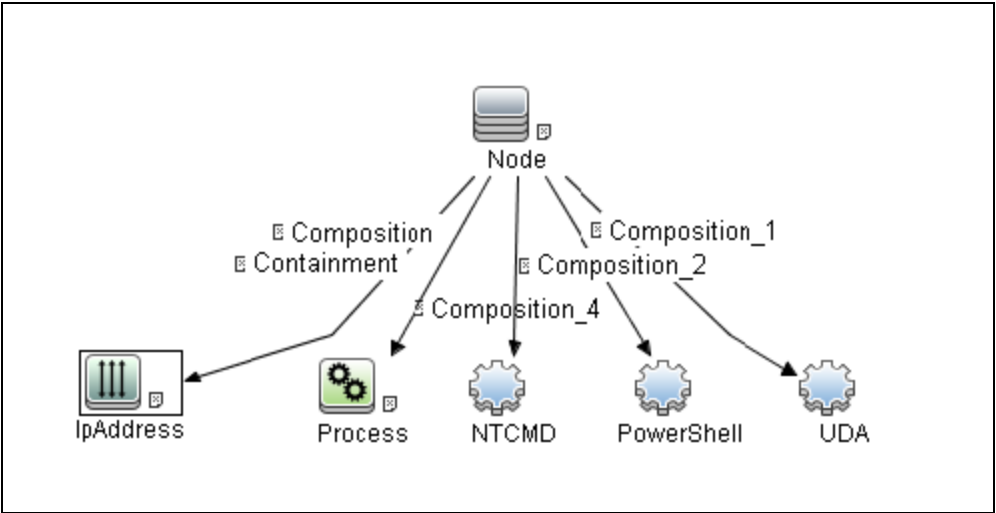
入力 TQL クエリ



トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

トリガ・クエリ



CI 属性条件

CI	条件
IpAddress	NOT IP Probe Name null

CI	条件
Process	Name 等しい (大文字小文字の区別なし) clussvc.exe
NTCMD	NOT Reference to the credentials dictionary entry null
PowerShell	NOT Reference to the credentials dictionary entry null

## 使用スクリプト

- entity.py
- ms\_cluster.py
- ms\_cluster\_discoverer.py
- MS\_Cluster\_Topology.py

## 検出された CIT

検出された CI の詳細については、[詳細]タブの統計テーブルを参照してください。

- ClusterResourceGroup
- ClusterSoftware
- Composition
- ConfigurationDocument
- Containment
- Dependency
- ExecutionEnvironment
- IpAddress
- MS Cluster
- MSCS Resource Group
- MSCS resource

- Membership
- Node
- Ownership
- Virtual

**注:** トポロジを表示するには、[「Microsoft Cluster Server View トポロジ」\(300ページ\)](#)を参照してください。

# 第16章: Red Hat Cluster Suite ディスカバリ

## 本章の内容

- 概要 ..... 307
- サポートされるバージョン ..... 307
- トポロジ ..... 307
- Red Hat Cluster Suite トポロジの検出方法 ..... 308
- Red Hat Cluster by Shell ジョブ ..... 308
- Red Hat Cluster by Shell アダプタ ..... 310
- 制限事項 ..... 312

## 概要

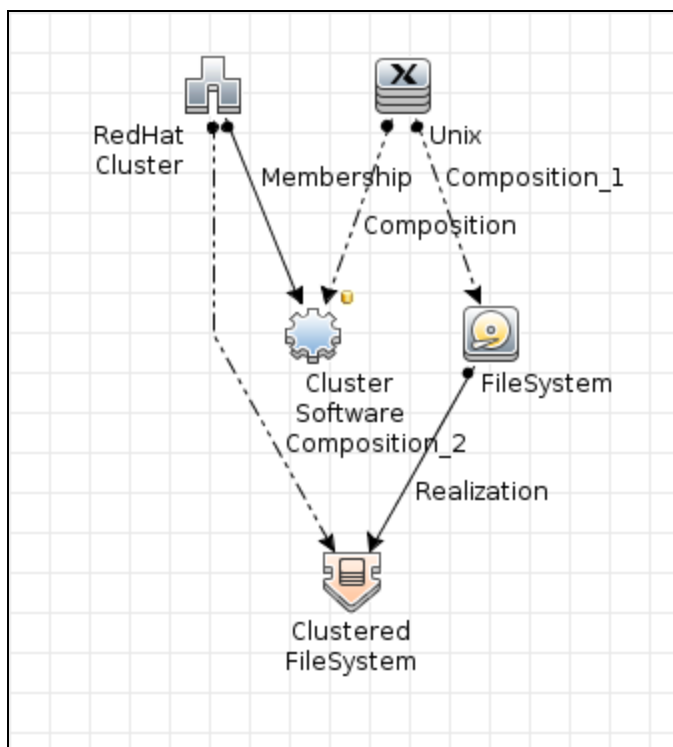
Red Hat Cluster Suite(RHCS)は、ハイアベイラビリティな負荷分散クラスタの設定を可能にするソフトウェア・コンポーネントのセットです。Global File System 2(GFS2)によって、Red Hat Cluster Suite と一緒に使用するためのクラスタ化されたファイル・システムが提供されます。GFS2 では、複数ノードによるブロック・レベルでのストレージ共有が可能です、ストレージがローカルで各クラスタ・ノードに接続されているように扱うことができます。

## サポートされるバージョン

Red Hat Cluster Suite ディスカバリは、Red Hat バージョン 6.3 でサポートされます。

## トポロジ

次の図は、Red Hat Cluster Suite ディスカバリのトポロジを示しています。



## Red Hat Cluster Suite トポロジの検出方法

本項では、Red Hat Cluster Suite が管理するトポロジの検出方法について説明します。

### 前提条件

- クラスタ・ノードにシェル接続できることを確認します。
- SSH プロトコル資格情報を設定します。

### ディスカバリの実行

注: Red Hat Cluster Suite ディスカバリは、シャロー・モードまたはディープ・モードで実行できます。シャロー・ディスカバリを実行するには、次の手順 1~3 を実行します。ディープ・ディスカバリを実行するには、次のすべての手順を実行します。

Red Hat Cluster Suite ディスカバリを実行するには、次のジョブを次の順序で実行します。

1. **Range IPs by ICMP**(ターゲット IP を検出)
2. **Host Connection by Shell**(ターゲット・ホストおよびそのホストへのシェル接続を検出)
3. **Host Applications by Shell**(Red Hat クラスタ・ソフトウェアを検出)

注: シャロー・ディスカバリのみを実行する場合、ここで終了し、次の手順に進む必要はありません。

4. **Red Hat Cluster by Shell**(Red Hat クラスタ・リソースを検出)

## Red Hat Cluster by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

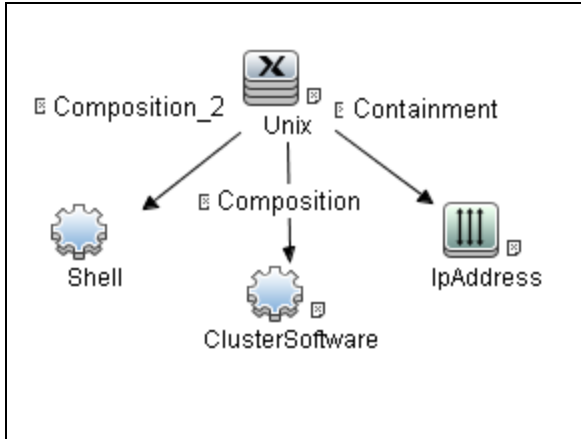
### アダプタ

ID : red\_hat\_cluster\_by\_shell



## トリガ TQL

このジョブは、シェル・アクセスがあるノード上の **ClusterSoftware** が **ProductName=redhat\_cluster** であり、そのノードの IP アドレスがプローブの範囲に含まれる場合にトリガされます。シェルに有効な資格情報がある必要があります。



ノード名	条件
IpAddress	NOT IP Probe Name null
Shell	NOT Reference to the credentials dictionary entry null
Unix	なし
ClusterSoftware	ProductName 等しい redhat_cluster

## デイスカバリ・フロー

**Red Hat Cluster by Shell** ジョブのデイスカバリ・フローは次のとおりです。

1. ジョブはクラスタ情報 (**clustat -x**) を取得し、必要なクラスタ・トポロジを作成します。
2. ジョブは **nslookup** または **hosts** ファイルを使用することでホストを解決します。
3. ジョブは **blkid** コマンドを実行し、GFS または GFS2 が使用中であるかどうかを判断します。
4. ジョブは **gfs\_edit** コマンドおよび **blkid** コマンドを実行し、GFS 構成を取得します。
5. ジョブは **cat/etc/mtab** コマンドを実行し、マウント・ポイントを取得します。

## Red Hat Cluster by Shell アダプタ

本項ではこのアダプタに関する詳細を説明します。

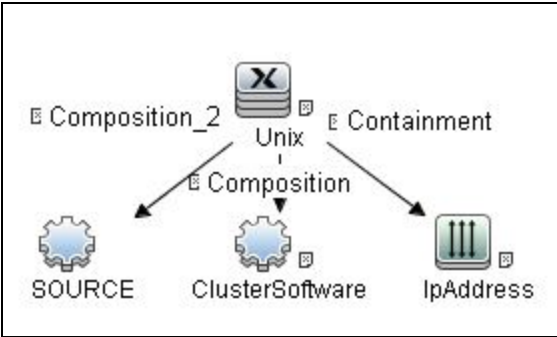
### ID

red\_hat\_cluster\_by\_shell

### 入力 CIT

Shell

### 入力 TQL



このジョブは、**ClusterSoftware** が **ProductName=redhat\_cluster** であり、シェル・アクセスがあるノードを持ち、そのノードの IP アドレスがプローブの範囲に含まれる場合にトリガされます。シェルに有効な資格情報がある必要があります。

ノード名	条件
Unix	なし
SOURCE	NOT Reference to the credentials dictionary entry null
ClusterSoftware	ProductName 等しい redhat_cluster
IpAddress	NOT IP Probe Name null

トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

- **Protocol**。シェルの root\_class です。
- **CredentialsId**。シェルの credentials\_id です。
- **IP アドレス**。シェルの application\_ip です。

使用スクリプト

- red\_hat\_cluster\_by\_shell.py

検出された CIT

- ClusteredFileSystem
- ClusterSoftware
- Composition
- FileSystem
- Membership
- Node
- Realization
- RedHatCluster

パラメータ

なし

## 制限事項

Red Hat Cluster Suite ディスカバリには、次の制限事項があります。

- 現在、GFS および基本クラスタ情報のディスカバリのみがサポートされています。
- クラスタ化サービスのディスカバリはサポートされていません。
- LVM のディスカバリはサポートされていません。

## 第17章: HP ServiceGuard Cluster ディスカバリ

### 本章の内容

概要 .....	314
サポートされるバージョン .....	314
トポロジ .....	314
HP ServiceGuard クラスタ・トポロジの検出方法 .....	315
Service Guard Cluster Topology by TTY ジョブ .....	316
Service Guard Cluster Topology アダプタ .....	318
HP ServiceGuard クラスタのコマンド .....	319

## 概要

HP ServiceGuard は、HP-UX 向けのクラスターソリューションです。HP Global Workload Management は、負荷を調整してパフォーマンスを最適化し、Instant Capacity on Demand を使用して統合を実現します。HP ServiceGuard により、インストール済みサービスを使用したファイルシステムのクラスタ化が可能です。 **Service Guard Cluster Topology by TTY** ジョブは、パッケージ、ファイルシステム要素、実行中のサービスなどの CI を、対応する論理リンクを使用して検出します。

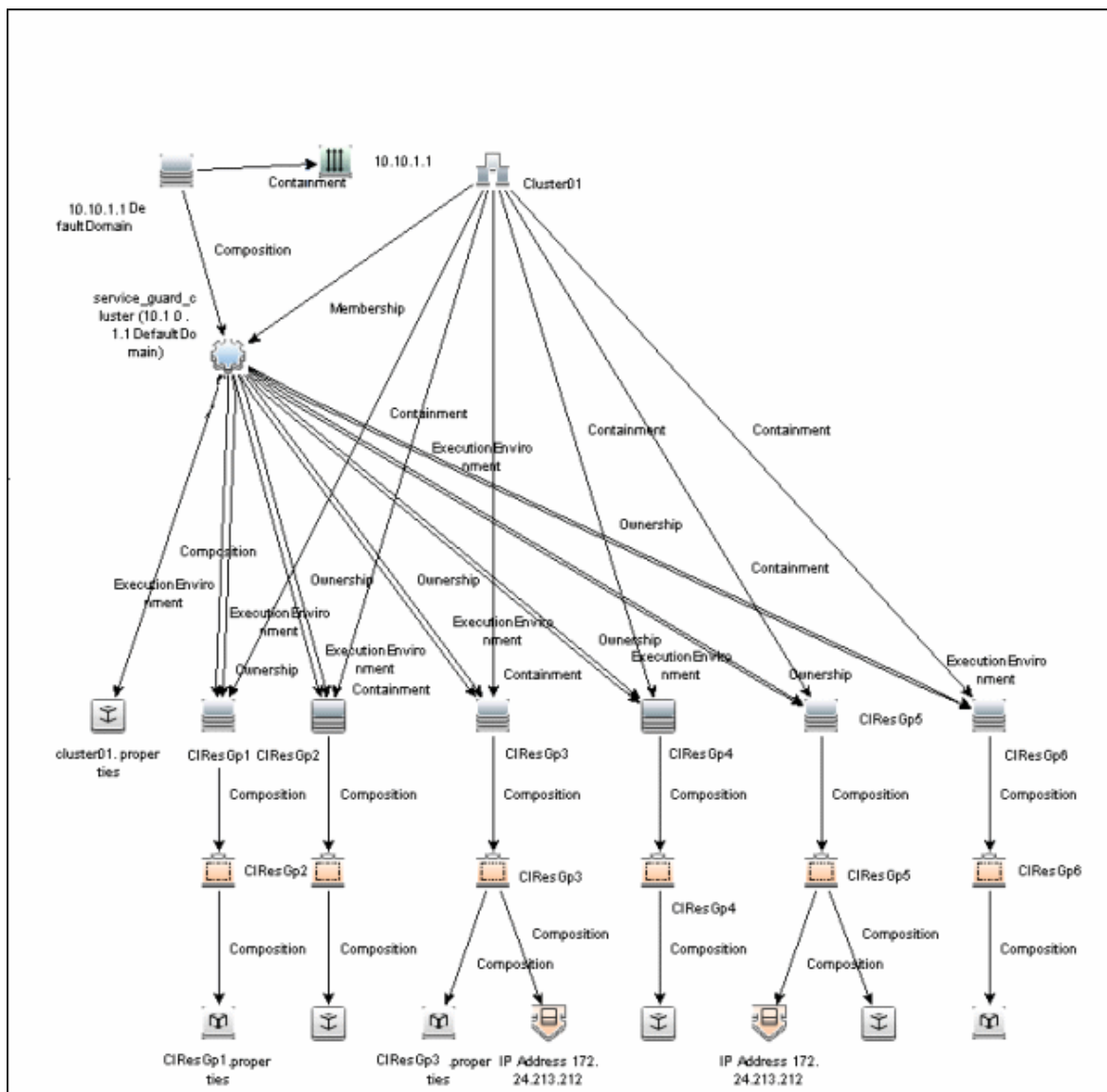
## サポートされるバージョン

このディスカバリ・ソリューションは、HP-UX 10.xx と 11.xx での HP ServiceGuard Cluster をサポートしています。

## トポロジ

次の図に、HP ServiceGuard Cluster ディスカバリのトポロジを示します。

注: 検出された CIT のリストについては、[「Service Guard Cluster Topology アダプタ」\(318ページ\)](#)を参照してください。



## HP ServiceGuard クラスタ・トポロジの検出方法

このタスクでは、ServiceGuard クラスタ・トポロジの検出方法を説明します。

### 1. 前提条件 - 権限

ディスカバリを開始する前に、次のコマンドを実行するために必要な権限がユーザにあることを確認します。

- `/usr/sbin/cmviewcl -v`
- `cat` <パッケージ構成またはログ>
- `uname`
- `ps -ef`
- `lsnrctl status`
- `pfiles`
- `lsnf`

## 2. 前提条件 - プロトコル資格情報の設定

HP ServiceGuard クラスタ・トポロジを検出するには、適切なシェル・プロトコル(SSH, Telnet, またはその両方)を、アクセス対象のシステムに応じて設定する必要があります。シェル・プロトコル用の情報として、**ユーザ名**、**パスワード**、**ドメイン名**を準備します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 3. ディスカバリの実行

次のジョブを実行します。

- **Range IPs by ICMP**(HP ServiceGuard クラスタ IP アドレスを検出するため)
- **Host Connection by Shell**(SSH エージェントとネットワーク・トポロジが接続された HP ServiceGuard システムを検出するため)
- **Host Applications by Shell**(HP ServiceGuard が設定され、検出対象で実行されているかどうかを検出するため)
- **Service Guard Cluster Topology by TTY**

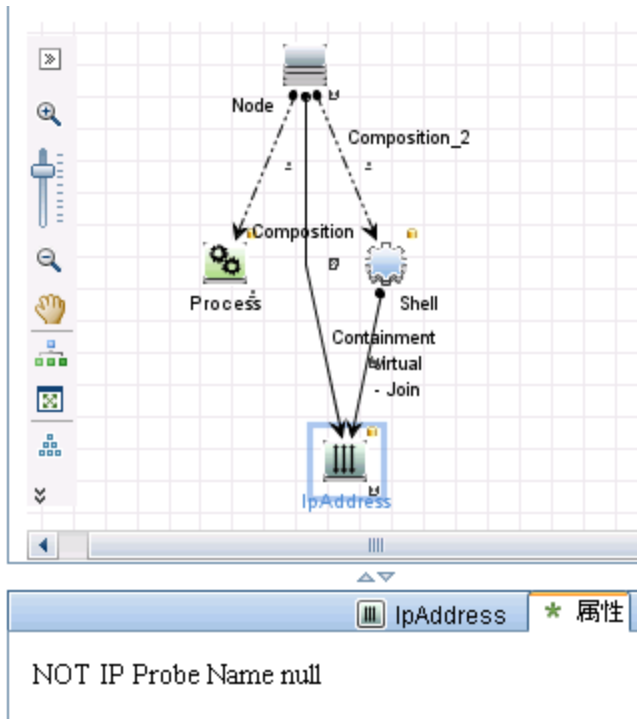
ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

# Service Guard Cluster Topology by TTY ジョブ

本項ではこのジョブに関する詳細を説明します。



トリガ・クエリ



CIT	属性
IpAddress	NOT IP Probe Name null
Shell	なし
Node	なし
Process	Name 等しい (大文字小文字の区別なし) cmcld OR 等しい (大文字小文字の区別なし) cmclconfd

アダプタ

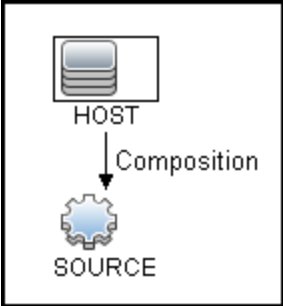
このジョブでは、**Service Guard Cluster Topology** アダプタを使用します。

詳細については、「[Service Guard Cluster Topology アダプタ](#)」(318ページ)を参照してください。

# Service Guard Cluster Topology アダプタ

本項ではこのアダプタに関する詳細を説明します。

## 入力クエリ



## 入力 CIT

Shell

## トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
cmclconfig_path	\${PARAMETERS.cmclconfig_path}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

## 使用スクリプト

- file\_mon\_utils.py
- file\_ver\_lib.py
- entity.py
- networking.py

- oracle\_shell\_utils.py
- service\_guard.py
- service\_guard\_discoverers.py
- Service\_Guard\_Cluster\_Topology.py

## 検出された CIT

- ClusterSoftware
- Composition
- ConfigurationDocument
- Containment
- Dependency
- ExecutionEnvironment
- IpAddress
- Membership
- Node
- Ownership
- SG Package
- SG Resource
- Service Guard Cluster

## HP ServiceGuard クラスタのコマンド

本項では、ServiceGuard のクラスタリング・コマンドを説明します。

```
/usr/local/bin/sudo /usr/sbin/cmviewcl -v
```

出力 :

CLUSTER            STATUS

SomeClusterName    up

NODE        STATUS        STATE

Node1       up        running

Quorum\_Server\_Status:

NAME            STATUS        STATE

172.24.0.5       up        running

Network\_Parameters:

INTERFACE    STATUS    PATH                    NAME

PRIMARY       up        0/2/2/1                lan3

PRIMARY       up        0/1/1/1                lan1

PRIMARY       up        0/2/2/0                lan2

PRIMARY       up        0/1/1/0                lan0

STANDBY       up        0/3/0/0/0/0/4/0/0/    lan7

STANDBY       up        0/3/0/0/0/0/2/0/0/    lan5

STANDBY       up        0/3/0/0/0/0/4/0/0/    lan6

STANDBY       up                        0/3/0/0/0/0/2/0/0/    lan4

PACKAGE    STATUS    STATE    AUTO\_RUN    NODE

PackageName1 up        running    enabled    Node1

Policy\_Parameters:

POLICY\_NAME CONFIGURED\_VALUE

Failover       configured\_node

Failback       manual

Node\_Switching\_Parameters:

NODE\_TYPE    STATUS    SWITCHING    NAME

Primary       up        enabled    Node1 (current)

PACKAGE    STATUS    STATE    AUTO\_RUN    NODE

PackageName2 up        running    enabled    Node1

Policy\_Parameters:

POLICY\_NAME CONFIGURED\_VALUE

Failover       configured\_node

Failback       manual

Script\_Parameters:

ITEM        STATUS    MAX\_RESTARTS    RESTARTS    NAME

Subnet       up        192.168.62.0

Subnet       up        172.24.0.0

```
Node_Switching_Parameters:
NODE_TYPE  STATUS  SWITCHING NAME
Primary    up      enabled Node1 (current)

PACKAGE STATUS STATE AUTO_RUN NODE
PackageName3 up running enabled Node1

Policy_Parameters:
POLICY_NAME CONFIGURED_VALUE
Failover configured_node
Failback manual

Node_Switching_Parameters:
NODE_TYPE STATUS SWITCHING NAME
Primary    up      enabled Node1 (current)
```

### マッピング

このコマンドの出力は、CI の次の属性の入力に使用されます。

CMD 出力属性	CI 名	CI 属性
SomeClusterName	Service Guard Cluster	Name
PackageName1,..., PackageName3	SG Package	Name
IP Address	SG Resource	Name
Subnet value	Network	Name
Node1	Node	Name

```
find /etc/cmcluster/ -name '*.cfg'
```

出力:

```
/etc/cmcluster/scripts/exampleapplicatie.cfg
/etc/cmcluster/package1/package1.cfg
/etc/cmcluster/package2/package2.cfg
/etc/cmcluster/package3/package3.cfg
```

マッピング:

このコマンドは、SG Cluster 構成ディレクトリ内のパッケージ構成ファイルを検索するために使用されます。

```
find /etc/cmcluster/ -name '*.config'
```

出力:

```
/etc/cmcluster/scripts/exampleapplicatie.config  
/etc/cmcluster/package1/package1.config  
/etc/cmcluster/package2/package2.config  
/etc/cmcluster/package3/package3.config
```

マッピング:

このコマンドは、SG Cluster 構成ディレクトリ内のパッケージ構成ファイルを検索するために使用されます。

```
cat "/etc/cmcluster/package1/package.cfg" | grep -iE "PACKAGE_  
NAME|SCRIPT_LOG_FILE|RUN_SCRIPT|FS_DIRECTORY"
```

出力:

```
# "PACKAGE_NAME" is the name that is used to identify the package.  
# Legal values for PACKAGE_NAME:  
PACKAGE_NAME package1  
# "RUN_SCRIPT" is the script that starts a package.  
# Legal values for RUN_SCRIPT:  
RUN_SCRIPT /etc/cmcluster/package1/package1.cntl  
# "RUN_SCRIPT_TIMEOUT" is the number of seconds allowed for the package to start.
```

マッピング:

PACKAGE\_NAME 変数とRUN\_SCRIPT 変数の値は、IP とマウント・ポイントを検出するこのパッケージで管理される別のコマンドでも使用されます。

```
cat "/etc/cmcluster/package1/package1.cntl.log" | grep -E  
"Mounting"
```

出力:

```
Jul 11 09:27:10 - Node "Node1":  
Mounting /dev/vg1/lvol1 at /oracle/somename1  
Jul 11 09:27:22 - Node "Node1":
```

```
Mounting /dev/vg1/lvol2 at /oracle/somename2
Jul 11 09:27:53 - Node "Node1":
Mounting /dev/vg1/lvol3 at /oracle/somename3
```

#### マッピング:

検出されたマウント・ポイントのデータは, RunningSoftware を適切なクラスタ化 サービス(実際はパッケージ)にリンクするために使用されます。このリンク付け手法は, 実行中のプロセス・パスに依存します。

```
cat "/etc/cmcluster/package1/package1.cnt1.log" | grep -E "Adding
IP"
```

#### 出力:

```
Jun 12 09:27:11 - Node "Node1":
Adding IP address 192.168.62.146 to subnet 192.168.62.0
Jun 12 09:27:11 - Node "Node1":
Adding IP address 172.24.10.142 to subnet 172.24.0.0
```

#### マッピング:

検出された IP アドレスとネットワークは, 対応する CI としてレポートされます。これが行われるのは, すべての IP リソースが cmviewcl 出力にあるとはかぎらないからです。

```
ps -ef | grep "tnslsnr"
```

#### 出力:

```
orauser 21926 1 0 Jun 9 ?
6:09 /oracle/somename1/applic/oracle/db/
10.2.0/instns1/bin/tnslsnr listener_name1 -inherit
```

#### マッピング:

取得された Oracle リスナのプロセス情報 ORACLE\_HOME 値から, リスナ名と pid が解析されます。ORACLE\_HOME とリスナ名は, 別のディスカバリでも, リスナ・ステータスと Oracle DB SID を解析するために使用されます。

```
/oracle/somename1/applic/oracle/db/10.2.0/instns1/bin/lsnrctl  
status listener_name1
```

**出力:**

```
LSNRCTL for HPUX:Version 10.2.0.5.0 -  
Production on 20-JUL-2011 06:44:11  
Copyright (c) 1991, 2010, Oracle.All rights reserved.  
  
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=oic6)))  
STATUS of the LISTENER  
-----  
Alias listener_name1  
Version TNSLSNR for HPUX:Version 10.2.0.5.0 -  
Production Start Date 09-JUN-2011 21:56:34  
Uptime 40 days 8 hr.47 min. 36 sec  
Trace Level off  
Security ON:Local OS Authentication  
SNMP OFF  
Listener Parameter File  
/oracle/somename1/applic/oracle/db/10.2.0/  
instns1/network/admin/listener.ora  
Listener Log File  
/oracle/somename1/applic/oracle/db/10.2.0/  
instns1/network/log/listener_name1.log  
Listening Endpoints Summary...  
DESCRIPTION=  
ADDRESS=(PROTOCOL=ipc)(KEY=sid1)))  
DESCRIPTION=  
ADDRESS=(PROTOCOL=tcp)(HOST=192.168.80.24)(PORT=1521)))  
DESCRIPTION=  
ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROCsidas)))  
  
Services Summary...  
Service "PLSExtProcsid1" has 1 instance(s).  
Instance "PLSExtProcdis1", status UNKNOWN,  
has 1 handler(s) for this service...  
Service "sid1.somedomain" has 1 instance(s).  
Instance "sid1", status UNKNOWN,
```



```
has 1 handler(s) for this service...
```

```
The command completed successfully
```

#### マッピング:

インスタンス値は解析され、SID として扱われます。PLSExtProc で始まるインスタンス名は RPC 呼び出しサービスであるため、除外されます。

```
nice lsof -i 4 -a -P -n -p <実際の PID>
```

#### または

```
nice pfiles 21926 2>&1 | awk "/S_IFSOCK|SOCK_STREAM|SOCK_
DGRAM|port/ { print }"
```

#### マッピング:

検出された IP とポートの情報は、レポートされる実行中ソフトウェアのアプリケーション IP とポートを設定するために使用されます。

```
cat
```

```
"/oracle/somename2/applic/oracle/oas/10.1.2/somename5/opmn/conf/o
pmn.xml"
```

#### 出力:

```
.....skip.....
<ias-instance id="somename.somedomain">
<environment>
<variable id="TMP" value="/tmp"/>
<variable id="LD_LIBRARY_PATH" value="/usr/lib"/>
<variable id="LD_PRELOAD" value="libloghost.so.1"/>
.....skip.....
```

#### マッピング:

Oracle iAS CI 名は、ias-instance の値に基づいてパラメータ名、パラメータ ID、標準設定サーバの順で構成されます。

## 第18章: Solaris Cluster ディスカバリ

### 本章の内容

概要 .....	327
サポートされるバージョン .....	327
トポロジ .....	327
Sun クラスタの検出方法 .....	329
Sun Cluster by Shell ジョブ .....	329
Sun Cluster by Shell アダプタ .....	331
Solaris Cluster ディスカバリ・コマンド .....	333
クラスタの名前の取得 .....	333
クラスタのノードの取得 .....	335
ノード名のIPへの解決 .....	335
ノードのステータスの取得 .....	336
リソース・グループおよびリソースの取得 .....	336
リソース・グループおよびリソースの詳細の取得 .....	337
クラスタ相互接続情報の取得 .....	348
定足数構成の取得 .....	352

## 概要

Sun Cluster 製品は、可用性と拡張性が高いサービスの作成に使用されるハードウェアとソフトウェアを統合したソリューションです。Sun Cluster 環境によって、Solaris オペレーティング・システムはクラスタ・オペレーティング・システムに拡張されます。クラスタとは、そのクラスタのみに属する 1 つまたは複数のノードの集まりです。

## サポートされるバージョン

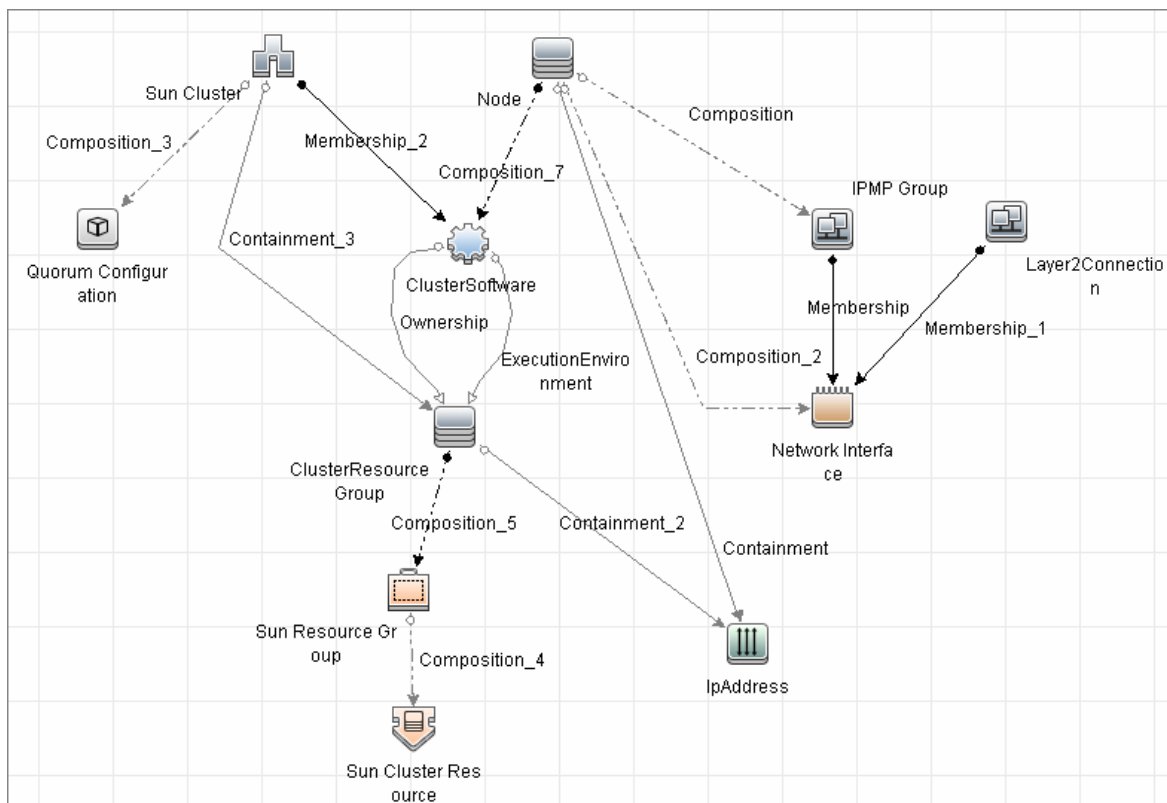
Sun Cluster パッケージは、Sun Cluster 3.2 をサポートしています。以前のバージョンの Sun Cluster のサポートは検証されていません。

Sun Cluster ソフトウェアは、Solaris オペレーティング・システムと統合されているため、このオペレーティング・システムのみがサポートされます。

## トポロジ

次の図に、Solaris Cluster ディスカバリのトポロジを示します。

注: 検出された CIT のリストについては、[「Sun Cluster by Shell ジョブ」\(329ページ\)](#)を参照してください。



## Sun クラスタの検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報および権限の設定

- このディスカバリでは、Telnet プロトコルおよび SSH プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

- Solaris Cluserter ディスカバリでクラスタリング・コマンド (**scrgadm**, **scstat**, **scconf** など)を実行できるように、ユーザの権限を設定します。コマンドの完全なリストについては、「[Solaris Cluster ディスカバリ・コマンド](#)」(333ページ)を参照してください。

### 2. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次のジョブを次の順序で実行します。

- Range IPs by ICMP** ジョブを実行し、IP 範囲内で稼働中のマシンを検出します。
- Host Connection by Shell** ジョブを実行し、シェル接続と、ホストに関する基本情報を検出します。
- Host Applications by Shell** ジョブを実行し、ターゲット・マシン上のプロセスを検出します。
- Sun Cluster by Shell** ジョブを実行し、Sun Cluster トポロジを検出します。ジョブの詳細については、「[Sun Cluster by Shell ジョブ](#)」(329ページ)を参照してください。

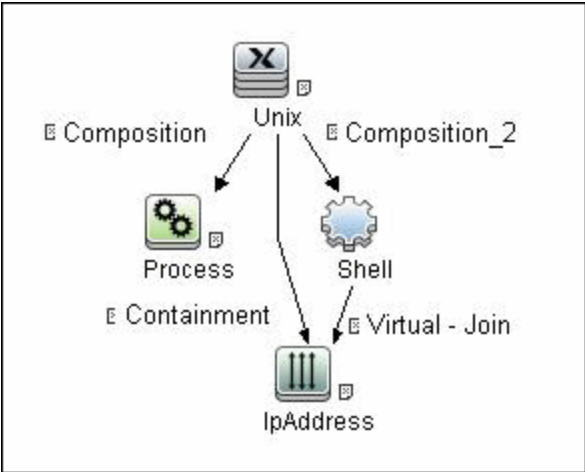
## Sun Cluster by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

ID :Sun\_Cluster\_by\_Shell

トリガ TQL



CI 属性条件

属性	条件
Process	Name 等しい (大文字小文字の区別なし) cluster
Shell	NOT Reference to the credentials dictionary entry null
IpAddress	Not IP Probe Name null

パラメータ

なし

前提条件

「[Sun クラスタの検出方法](#)」(329ページ)の前提条件を参照してください。

ディスカバリ・フロー

Sun Cluster by Shell ジョブのディスカバリ・フローは次のとおりです。

1. 次を含むクラスタ構成を `/usr/cluster/bin/scconf -pv` コマンドを使用して取得します。
  - クラスタ名
  - クラスタ・ノードの名前

- トランスポート・アダプタ
  - 定足数デバイス
- 2. クラスタ・バージョンを `/usr/cluster/bin/scinstall -p` コマンドを使用して取得します。
- 3. 定足数ステータスを `/usr/cluster/bin/scstat -q` コマンドを使用して取得します。
- 4. 検出されたクラスタ・ノードを分析し、それらのホスト名を IP として解決します。
- 5. クラスタ・ノードのステータスを `/usr/cluster/bin/scstat -n` コマンドを使用して取得します。
- 6. 標準ネットワーク情報(特に `netstat` 情報)を `/usr/bin/netstat -np` コマンドを使用して取得します。  
この情報は、ジョブの接続先 クラスタ・ノード上のトランスポート・アダプタの MAC アドレスの解決に使用されます。
- 7. ジョブの現在の接続先でないクラスタ・ノードの ARP テーブルを介して、`/usr/sbin/arp <IP>` コマンドを使用してトランスポート・アダプタの IP アドレスを MAC として解決します。
- 8. 次を含むクラスタ・リソースを `/usr/cluster/bin/scrgadm -pvv` コマンドを使用して取得します。
  - リソース・グループ
  - リソース : `LogicalHostname` タイプまたは `SharedAddress` タイプのリソースのホスト名が IP として解決されます。
- 9. リソース・グループのステータスを `/usr/cluster/bin/scstat -g` コマンドを使用して取得します。
- 10. トランスポート・パスを `/usr/cluster/bin/scstat -W` コマンドを使用して取得します。
- 11. 検出されたトポロジを使用して結果ベクトルを作成し、それを UCMDB に送信します。

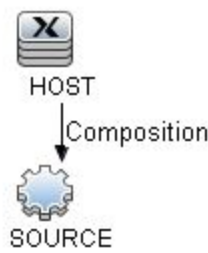
## Sun Cluster by Shell アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Shell

入力クエリ



トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

使用スクリプト

- networking.py
- solaris\_networking.py
- sun\_cluster\_by\_shell.py

検出された CIT

- ClusterSoftware
- Composition
- ConfigurationDocument
- Containment
- ExecutionEnvironment



- Interface
- IpAddress
- Layer2Connection
- Membership
- Node
- Sun Cluster
- Sun Cluster Resource
- Sun Resource Group

注: トポロジを表示するには, [「トポロジ」\(327ページ\)](#)を参照してください。

## Solaris Cluster ディスカバリ・コマンド

本項では, 次の Sun クラスタ化コマンドについて説明します。

- [「クラスタの名前の取得」\(333ページ\)](#)
- [「クラスタのノードの取得」\(335ページ\)](#)
- [「ノード名の IP への解決」\(335ページ\)](#)
- [「ノードのステータスの取得」\(336ページ\)](#)
- [「リソース・グループおよびリソースの取得」\(336ページ\)](#)
- [「リソース・グループおよびリソースの詳細の取得」\(337ページ\)](#)
- [「クラスタ相互接続情報の取得」\(348ページ\)](#)
- [「定数構成の取得」\(352ページ\)](#)

### クラスタの名前の取得

コマンド	/usr/cluster/bin/scconf -p
------	----------------------------

出力例	<pre>Cluster name: cluster1 Cluster ID:0x4A7BDCD3 Cluster install mode: disabled Cluster private net:172.2.0.0 Cluster private netmask:255.255.255.192 Cluster maximum nodes:6 Cluster maximum private networks:4 Cluster new node authentication: unix Cluster authorized-node list:&lt;.- Exclude all nodes&gt; Cluster transport heart beat timeout:10000 Cluster transport heart beat quantum:1000 Round Robin Load Balancing UDP session timeout:480 Cluster nodes: node1 node2 Cluster node name: node1 ...</pre>
取得された値	クラスタの名 前 : cluster1
コメント	クラスタの名 前によって, Sun Cluster CI の作 成 が可 能 になります。

## クラスタのノードの取得

コマンド	/usr/cluster/bin/scconf -p
出力例	Cluster name: cluster1 Cluster ID:0x4A7BDCD3 Cluster install mode: disabled Cluster private net:172.2.7.0 Cluster private netmask:255.255.255.192 Cluster maximum nodes:6 Cluster maximum private networks:4 Cluster new node authentication: unix Cluster authorized-node list:<.- Exclude all nodes> Cluster transport heart beat timeout:10000 Cluster transport heart beat quantum:1000 Round Robin Load Balancing UDP session timeout:480 Cluster nodes: node1 node2 ...
取得された値	ノード名

## ノード名の IP への解決

コマンド	/usr/sbin/nslookup node1
出力例	Server: 134.44.0.10 Address: 134.44.0.10#53 Name: node1.example.com Address: 134.44.0.75
取得された値	ノードの IP
コメント	IP によって、クラスタ内の各ノードに対する不完全なホストの作成が可能になります。

## ノードのステータスの取得

コマンド	/usr/cluster/bin/scstat -n
出力例	-- Cluster Nodes --  Node name Status  -----  Cluster node: node1 Online  Cluster node: node2 Online
取得された値	ノードのステータス
コメント	ステータスはレポートされませんが、ディスカバリがこのステータスを必要とします。たとえば、ノードがオフラインの場合は、ディスカバリが MAC アドレスの解決に arp コマンドを発行しないようにする必要があります。

## リソース・グループおよびリソースの取得

コマンド	/usr/cluster/bin/scstat -g
出力例	-- Resource Groups and Resources --  Group Name Resources  -----  Resources: oracle1 oracle1-zfs oracle1-lh oracle1-ora oracle1-cron oracle1-lsnr_ano_10  -- Resource Groups --  ...
取得された値	グループのリスト  グループ内のリソースのリスト  各ノード上のグループのステータス(これに基づいて実行リンクを作成)

## リソース・グループおよびリソースの詳細の取得

コマンド	/usr/cluster/bin/scrgadm -pvv
出力例	<pre> Res Group name: oracle1 (oracle1) Res Group RG_description:&lt;NULL&gt;  (oracle1) Res Group mode:Failover  (oracle1) Res Group management state:Managed  (oracle1) Res Group RG_project_name: user.oracle  (oracle1) Res Group RG_SLM_type: manual  (oracle1) Res Group RG_affinities:&lt;NULL&gt;  (oracle1) Res Group Auto_start_on_new_cluster:True  (oracle1) Res Group Failback:False  (oracle1) Res Group Nodelist: node1 node2  (oracle1) Res Group Maximum primaries:1  (oracle1) Res Group Desired primaries:1  (oracle1) Res Group RG_dependencies:&lt;NULL&gt;  (oracle1) Res Group network dependencies:True  (oracle1) Res Group Global_resources_used:&lt;All&gt;  (oracle1) Res Group Pingpong_interval:3600  (oracle1) Res Group Pathprefix:&lt;NULL&gt;  (oracle1) Res Group system:False  (oracle1) Res Group Suspend_automatic_recovery:False  (oracle1) Res name: oracle1-zfs  (oracle1:oracle1-zfs) Res R_description:  (oracle1:oracle1-zfs) Res resource type:SUNW.HAStoragePlus:8  (oracle1:oracle1-zfs) Res type version:8  (oracle1:oracle1-zfs) Res resource group name: oracle1  (oracle1:oracle1-zfs) Res resource project name: user.oracle  (oracle1:oracle1-zfs{kvsdb1}) Res enabled:True </pre>
出力例	<pre> (oracle1:oracle1-zfs{kvsdb2}) Res enabled:True </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
(つづき)	<pre> (oracle1:oracle1-zfs{kvsdb1}) Res monitor enabled:True (oracle1:oracle1-zfs{kvsdb2}) Res monitor enabled:True (oracle1:oracle1-zfs) Res strong dependencies:&lt;NULL&gt; (oracle1:oracle1-zfs) Res weak dependencies:&lt;NULL&gt; (oracle1:oracle1-zfs) Res restart dependencies:&lt;NULL&gt; (oracle1:oracle1-zfs) Res offline restart dependencies:&lt;NULL&gt; (oracle1:oracle1-zfs) Res property name:Retry_interval (oracle1:oracle1-zfs:Retry_interval) Res property class: standard (oracle1:oracle1-zfs:Retry_interval) Res property description:Time in which monitor attempts to restart a failed resource Retry_count times. (oracle1:oracle1-zfs:Retry_interval) Res property type: int (oracle1:oracle1-zfs:Retry_interval) Res property value:300 (oracle1:oracle1-zfs) Res property name:Retry_count (oracle1:oracle1-zfs:Retry_count) Res property class: standard (oracle1:oracle1-zfs:Retry_count) Res property description:Indicates the number of times a monitor restarts the resource if it fails. (oracle1:oracle1-zfs:Retry_count) Res property type: int (oracle1:oracle1-zfs:Retry_count) Res property value:2 (oracle1:oracle1-zfs) Res property name:Failover_mode (oracle1:oracle1-zfs:Failover_mode) Res property class: standard (oracle1:oracle1-zfs:Failover_mode) Res property description:Modifies recovery actions taken when the resource fails. (oracle1:oracle1-zfs:Failover_mode) Res property type: enum (oracle1:oracle1-zfs:Failover_mode) Res property value:SOFT (oracle1:oracle1-zfs) Res property name:POSTNET_STOP_ TIMEOUT </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<pre> (oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property class: standard  (oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property description:Maximum execution time allowed for Postnet_stop method.  (oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property type: int  (oracle1:oracle1-zfs:POSTNET_STOP_TIMEOUT) Res property value:1800  (oracle1:oracle1-zfs) Res property name:PRENET_START_TIMEOUT  (oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property class: standard  (oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property description:Maximum execution time allowed for Prenet_Start method.  (oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property type: int  (oracle1:oracle1-zfs:PRENET_START_TIMEOUT) Res property value:1800  (oracle1:oracle1-zfs) Res property name:MONITOR_CHECK_ TIMEOUT  (oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property class: standard  (oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property description:Maximum execution time allowed for Monitor_Check method.  (oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property type: int  (oracle1:oracle1-zfs:MONITOR_CHECK_TIMEOUT) Res property value:90  (oracle1:oracle1-zfs) Res property name:MONITOR_STOP_ TIMEOUT  (oracle1:oracle1-zfs:MONITOR_STOP_TIMEOUT) Res property class: standard  (oracle1:oracle1-zfs:MONITOR_STOP_TIMEOUT) Res property description:Maximum execution time allowed for Monitor_Stop </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<p>method.</p> <p>(oracle1:oracle1-zfs:MONITOR_STOP_TIMEOUT) Res property type: int (oracle1:oracle1-zfs:MONITOR_STOP_TIMEOUT) Res property value:90</p> <p>(oracle1:oracle1-zfs) Res property name:MONITOR_START_TIMEOUT</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property description:Maximum execution time allowed for Monitor_Start method.</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:MONITOR_START_TIMEOUT) Res property value:90</p> <p>(oracle1:oracle1-zfs) Res property name:INIT_TIMEOUT</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property description:Maximum execution time allowed for Init method.</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:INIT_TIMEOUT) Res property value:1800</p> <p>(oracle1:oracle1-zfs) Res property name:UPDATE_TIMEOUT</p> <p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property class: standard</p> <p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property description:Maximum execution time allowed for Update method.</p> <p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:UPDATE_TIMEOUT) Res property value:1800</p> <p>(oracle1:oracle1-zfs) Res property name:VALIDATE_TIMEOUT</p> <p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property class: standard</p>



コマンド	/usr/cluster/bin/scrgadm -pvv
	<p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property description:Maximum execution time allowed for Validate method.</p> <p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property type: int</p> <p>(oracle1:oracle1-zfs:VALIDATE_TIMEOUT) Res property value:1800</p> <p>(oracle1:oracle1-zfs) Res property name:ZpoolsSearchDir</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property class: extension</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property description:Directory location to search devices for zpools</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property pernode:False</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property type: string</p> <p>(oracle1:oracle1-zfs:ZpoolsSearchDir) Res property value:</p> <p>(oracle1:oracle1-zfs) Res property name:FilesystemCheckCommand</p> <p>(oracle1:oracle1-zfs:FilesystemCheckCommand) Res property class: extension</p> <p>(oracle1:oracle1-zfs:FilesystemCheckCommand) Res property description:Command string to be executed for file system checks</p> <p>(oracle1:oracle1-zfs:FilesystemCheckCommand) Res property pernode:False</p> <p>(oracle1:oracle1-zfs:FilesystemCheckCommand) Res property type: stringarray</p> <p>(oracle1:oracle1-zfs:FilesystemCheckCommand) Res property value:&lt;NULL&gt;</p> <p>(oracle1:oracle1-zfs) Res property name:AffinityOn</p> <p>(oracle1:oracle1-zfs:AffinityOn) Res property class: extension</p> <p>(oracle1:oracle1-zfs:AffinityOn) Res property description:For specifying affinity switchover</p>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<pre> (oracle1:oracle1-zfs:AffinityOn) Res property pernode:False (oracle1:oracle1-zfs:AffinityOn) Res property type: boolean (oracle1:oracle1-zfs:AffinityOn) Res property value:TRUE (oracle1:oracle1-zfs) Res property name:FilesystemMountPoints (oracle1:oracle1-zfs:FilesystemMountPoints) Res property class: extension (oracle1:oracle1-zfs:FilesystemMountPoints) Res property description:The list of file system mountpoints (oracle1:oracle1-zfs:FilesystemMountPoints) Res property pernode:False (oracle1:oracle1-zfs:FilesystemMountPoints) Res property type: stringarray (oracle1:oracle1-zfs:FilesystemMountPoints) Res property value:&lt;NULL&gt; (oracle1:oracle1-zfs) Res property name:GlobalDevicePaths (oracle1:oracle1-zfs:GlobalDevicePaths) Res property class: extension (oracle1:oracle1-zfs:GlobalDevicePaths) Res property description:The list of HA global device paths (oracle1:oracle1-zfs:GlobalDevicePaths) Res property pernode:False (oracle1:oracle1-zfs:GlobalDevicePaths) Res property type: stringarray (oracle1:oracle1-zfs:GlobalDevicePaths) Res property value:&lt;NULL&gt; (oracle1:oracle1-zfs) Res property name:Zpools (oracle1:oracle1-zfs:Zpools) Res property class: extension (oracle1:oracle1-zfs:Zpools) Res property description:The list of zpools (oracle1:oracle1-zfs:Zpools) Res property pernode:False (oracle1:oracle1-zfs:Zpools) Res property type: stringarray (oracle1:oracle1-zfs:Zpools) Res property value: oracle1prod </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<pre> (oracle1) Res name: oracle1-lh   (oracle1:oracle1-lh) Res R_description:     (oracle1:oracle1-lh) Res resource type:SUNW.LogicalHostname:2   (oracle1:oracle1-lh) Res type version:2 (oracle1:oracle1-lh) Res resource group name: oracle1   (oracle1:oracle1-lh) Res resource project name: user.oracle (oracle1:oracle1-lh{kvsdb1}) Res enabled:True   (oracle1:oracle1-lh{kvsdb2}) Res enabled:True   (oracle1:oracle1-lh{kvsdb1}) Res monitor enabled:True   (oracle1:oracle1-lh{kvsdb2}) Res monitor enabled:True (oracle1:oracle1-lh) Res strong dependencies:&lt;NULL&gt;   (oracle1:oracle1-lh) Res weak dependencies:&lt;NULL&gt; (oracle1:oracle1-lh) Res restart dependencies:&lt;NULL&gt; (oracle1:oracle1-lh) Res offline restart dependencies:&lt;NULL&gt; (oracle1:oracle1-lh) Res property name:Retry_interval   (oracle1:oracle1-lh:Retry_interval) Res property class: standard (oracle1:oracle1-lh:Retry_interval) Res property description:Time in which monitor attempts to restart a failed resource Retry_count times.   (oracle1:oracle1-lh:Retry_interval) Res property type: int   (oracle1:oracle1-lh:Retry_interval) Res property value:300   (oracle1:oracle1-lh) Res property name:Retry_count   (oracle1:oracle1-lh:Retry_count) Res property class: standard   (oracle1:oracle1-lh:Retry_count) Res property description:Indicates the number of times a monitor restarts the resource if it fails.   (oracle1:oracle1-lh:Retry_count) Res property type: int   (oracle1:oracle1-lh:Retry_count) Res property value:2 </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<pre> (oracle1:oracle1-lh) Res property name:Thorough_probe_interval (oracle1:oracle1-lh:Thorough_probe_interval) Res property class: standard (oracle1:oracle1-lh:Thorough_probe_interval) Res property description:Time between invocations of a high-overhead fault probe of the resource. (oracle1:oracle1-lh:Thorough_probe_interval) Res property type: int (oracle1:oracle1-lh:Thorough_probe_interval) Res property value:60  (oracle1:oracle1-lh) Res property name:Cheap_probe_interval (oracle1:oracle1-lh:Cheap_probe_interval) Res property class: standard (oracle1:oracle1-lh:Cheap_probe_interval) Res property description:Time between invocations of a quick fault probe of the resource. (oracle1:oracle1-lh:Cheap_probe_interval) Res property type: int (oracle1:oracle1-lh:Cheap_probe_interval) Res property value:60  (oracle1:oracle1-lh) Res property name:Failover_mode (oracle1:oracle1-lh:Failover_mode) Res property class: standard (oracle1:oracle1-lh:Failover_mode) Res property description:Modifies recovery actions taken when the resource fails. (oracle1:oracle1-lh:Failover_mode) Res property type: enum (oracle1:oracle1-lh:Failover_mode) Res property value:HARD  (oracle1:oracle1-lh) Res property name:PRENET_START_TIMEOUT (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property description:Maximum execution time allowed for Prenet_Start method. </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<pre> (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property type: int  (oracle1:oracle1-lh:PRENET_START_TIMEOUT) Res property value:300  (oracle1:oracle1-lh) Res property name:MONITOR_CHECK_ TIMEOUT  (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property class: standard  (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property description:Maximum execution time allowed for Monitor_Check method.  (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property type: int  (oracle1:oracle1-lh:MONITOR_CHECK_TIMEOUT) Res property value:300  (oracle1:oracle1-lh) Res property name:MONITOR_STOP_TIMEOUT  (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property class: standard  (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property description:Maximum execution time allowed for Monitor_Stop method.  (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property type: int  (oracle1:oracle1-lh:MONITOR_STOP_TIMEOUT) Res property value:300  (oracle1:oracle1-lh) Res property name:MONITOR_START_ TIMEOUT  (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property class: standard  (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property description:Maximum execution time allowed for Monitor_Start method.  (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property type: int  (oracle1:oracle1-lh:MONITOR_START_TIMEOUT) Res property value:300 </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<pre> (oracle1:oracle1-lh) Res property name:UPDATE_TIMEOUT (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property description:Maximum execution time allowed for Update method. (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property type: int (oracle1:oracle1-lh:UPDATE_TIMEOUT) Res property value:300 (oracle1:oracle1-lh) Res property name:VALIDATE_TIMEOUT (oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property description:Maximum execution time allowed for Validate method. (oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property type: int (oracle1:oracle1-lh:VALIDATE_TIMEOUT) Res property value:300 (oracle1:oracle1-lh) Res property name:STOP_TIMEOUT (oracle1:oracle1-lh:STOP_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:STOP_TIMEOUT) Res property description:Maximum execution time allowed for Stop method. (oracle1:oracle1-lh:STOP_TIMEOUT) Res property type: int (oracle1:oracle1-lh:STOP_TIMEOUT) Res property value:300 (oracle1:oracle1-lh) Res property name:START_TIMEOUT (oracle1:oracle1-lh:START_TIMEOUT) Res property class: standard (oracle1:oracle1-lh:START_TIMEOUT) Res property description:Maximum execution time allowed for Start method. (oracle1:oracle1-lh:START_TIMEOUT) Res property type: int (oracle1:oracle1-lh:START_TIMEOUT) Res property value:500 (oracle1:oracle1-lh) Res property name:CheckNameService </pre>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<pre> (oracle1:oracle1-lh:CheckNameService) Res property class: extension  (oracle1:oracle1-lh:CheckNameService) Res property description:Name service check flag  (oracle1:oracle1-lh:CheckNameService) Res property pernode:False  (oracle1:oracle1-lh:CheckNameService) Res property type: boolean  (oracle1:oracle1-lh:CheckNameService) Res property value:TRUE  (oracle1:oracle1-lh) Res property name:NetIfList  (oracle1:oracle1-lh:NetIfList) Res property class: extension  (oracle1:oracle1-lh:NetIfList) Res property description:List of IPMP groups on each node  (oracle1:oracle1-lh:NetIfList) Res property pernode:False  (oracle1:oracle1-lh:NetIfList) Res property type: stringarray  (oracle1:oracle1-lh:NetIfList) Res property value: ipmp1@1 ipmp1@2  (oracle1:oracle1-lh) Res property name:HostnameList  (oracle1:oracle1-lh:HostnameList) Res property class: extension  (oracle1:oracle1-lh:HostnameList) Res property description:List of hostnames this resource manages  (oracle1:oracle1-lh:HostnameList) Res property pernode:False  (oracle1:oracle1-lh:HostnameList) Res property type: stringarray  (oracle1:oracle1-lh:HostnameList) Res property value: oracle1  ... </pre>
取得された値	<ul style="list-style-type: none"> <li>• グループ: <ul style="list-style-type: none"> <li>◦ 名前</li> </ul> </li> </ul>

コマンド	/usr/cluster/bin/scrgadm -pvv
	<ul style="list-style-type: none"> <li>◦ 説明</li> <li>◦ 管理状態</li> <li>◦ モード (failover / scalable)</li> <li>◦ 最大プライマリ</li> <li>◦ 目的のプライマリ</li> <li>◦ ノード・リスト</li> <li>◦ システムかどうか</li> <li>◦ 新しいクラスタでの自動開始</li> <li>◦ フェイルバック</li> <li>• リソース : <ul style="list-style-type: none"> <li>◦ 名前</li> <li>◦ 説明</li> <li>◦ タイプ</li> <li>◦ フェイルオーバー・モード</li> <li>◦ 再試行間隔</li> <li>◦ 再試行回数</li> </ul> </li> </ul>
コメント	<p>抽出された値に基づき、ディスカバリは属性を使用してリソース・グループを作成し、属性を使用してリソースを作成します。</p> <p>LogicalHostname 処理 : このタイプのリソースについて、ディスカバリはこのリソースが管理するホスト名を含む追加の <b>HostnameList</b> プロパティを抽出します。ホスト名は IP に解決されます。解決された IP は、<b>ClusteredServer</b> CIT に追加されます。</p>

## クラスタ相互接続情報の取得

コマンド	/usr/cluster/bin/scstat -W
出力例	-- Cluster Transport Paths --



コマンド	/usr/cluster/bin/scstat -W
	<pre>Endpoint Endpoint Status ----- Transport path: node1:bge3 node2:nxge11 Path online Transport path: node1:nxge3 node2:nxge3 Path online</pre>
取得された値	出力には、トランスポート・パスとそれらのステータスのリストが含まれます。 オンラインの各パスについて、ソース・ノード上のソース・インタフェースおよび ターゲット・ノード上のターゲット・インタフェースを取得します。
コメント	これらのトランスポート・パスは、ソース・インタフェースからターゲット・インタ フェースへのレイヤ 2 リンクと一緒にレポートされます。 リモート・インタフェース(接続先ノードではないノード上にある)をレポートする には、以下に説明する MAC アドレスが取得されます。

<b>コマンド</b>	/usr/cluster/bin/scconf -p
<b>出力例</b>	<pre> ... Cluster install mode: disabled Cluster private net:172.2.0.0 Cluster private netmask:255.255.255.192 Cluster maximum nodes:6 Cluster maximum private networks:4 Cluster new node authentication: unix Cluster authorized-node list:&lt;.- Exclude all nodes&gt; Cluster transport heart beat timeout:10000 Cluster transport heart beat quantum:1000 Round Robin Load Balancing UDP session timeout:480 Cluster nodes: node1 node2 Cluster node name: node1 Node ID:1 Node enabled: yes Node private hostname: clusternode1-priv Node quorum vote count:1 Node reservation key:0x4A7ADDD300000001 Node zones:&lt;NULL&gt; CPU shares for global zone:1 Minimum CPU requested for global zone:1 Node transport adapters: nxge3 bge3 Node transport adapter: nxge3 Adapter enabled: yes Adapter transport type: dlpi Adapter property: device_name=nxge </pre>
<b>出力例 (つづき)</b>	<pre> Adapter property: device_instance=3 Adapter property: lazy_free=1 </pre>

	<p>Adapter property: dlpi_heartbeat_timeout=10000</p> <p>Adapter property: dlpi_heartbeat_quantum=1000</p> <p>Adapter property: nw_bandwidth=80</p> <p>Adapter property: bandwidth=70</p> <p>Adapter property: ip_address=172.2.0.9</p> <p>Adapter property: netmask=255.255.255.248</p> <p>Adapter port names:0</p> <p>Adapter port:0</p> <p>Port enabled: yes</p> <p>Node transport adapter: bge3</p> <p>Adapter enabled: yes</p> <p>Adapter transport type: dlpi</p> <p>Adapter property: device_name=bge</p> <p>Adapter property: device_instance=3</p> <p>Adapter property: lazy_free=1</p> <p>Adapter property: dlpi_heartbeat_timeout=10000</p> <p>Adapter property: dlpi_heartbeat_quantum=1000</p> <p>Adapter property: nw_bandwidth=80</p> <p>Adapter property: bandwidth=70</p> <p>Adapter property: ip_address=172.2.0.17</p> <p>Adapter property: netmask=255.255.255.248</p> <p>Adapter port names:0</p> <p>Adapter port:0</p> <p>Port enabled: yes</p> <p>...</p>
取得された値	<p>プライベート・ネットワーク・アドレス</p> <p>クラスタ相互接続内で使用されるインタフェースのリスト : 名前と IP アドレスが割り当てられる</p>

コマンド	/usr/sbin/arp 172.2.0.10
出力例	172.2.0.10 (172.2.0.10) at 0:21:a8:39:33:a9
取得された値	MAC
コメント	ディスカバリは、arp を介してリモート・インタフェースの MAC アドレスを解決します。解決できない場合、ディスカバリはトランスポート・パスをレイヤ 2 リンクとしてレポートしません。

## 定足数構成の取得

コマンド	/usr/cluster/bin/scstat -q
出力例	<pre>-- Quorum Summary from latest node reconfiguration -- Quorum votes possible:3 Quorum votes needed:2 Quorum votes present:3 -- Quorum Votes by Node (current status) -- Node Name Present Possible Status ----- Node votes: node1 1 1 Online Node votes: node2 1 1 Online -- Quorum Votes by Device (current status) -- Device Name Present Possible Status ----- Device votes: clusterquo1 1 1 Online</pre>
取得された値	定足数ステータス情報
コメント	定足数デバイスに関する詳細は、定足数構成の config ファイルに追加されます。

# 第 19 章: Veritas ディスカバリ

## 本章の内容

- 概要 ..... 354
- サポートされるバージョン ..... 354
- トポロジ ..... 355
- Veritas Cluster Server の検出方法 ..... 356
- Veritas Cluster by Shell ジョブ ..... 356

## 概要

Veritas Cluster グループは、1 つの構成単位として管理される、依存関係または関連性のあるリソースの集まりです。各 Veritas Cluster グループは、指定されたノードとリンクされ、このノードがグループ内に含まれたリソースをアクティブ化します。指定されたノード内でエラーが発生した場合、リソースをアクティブ化する役割は、別のノードに切り替えられます。

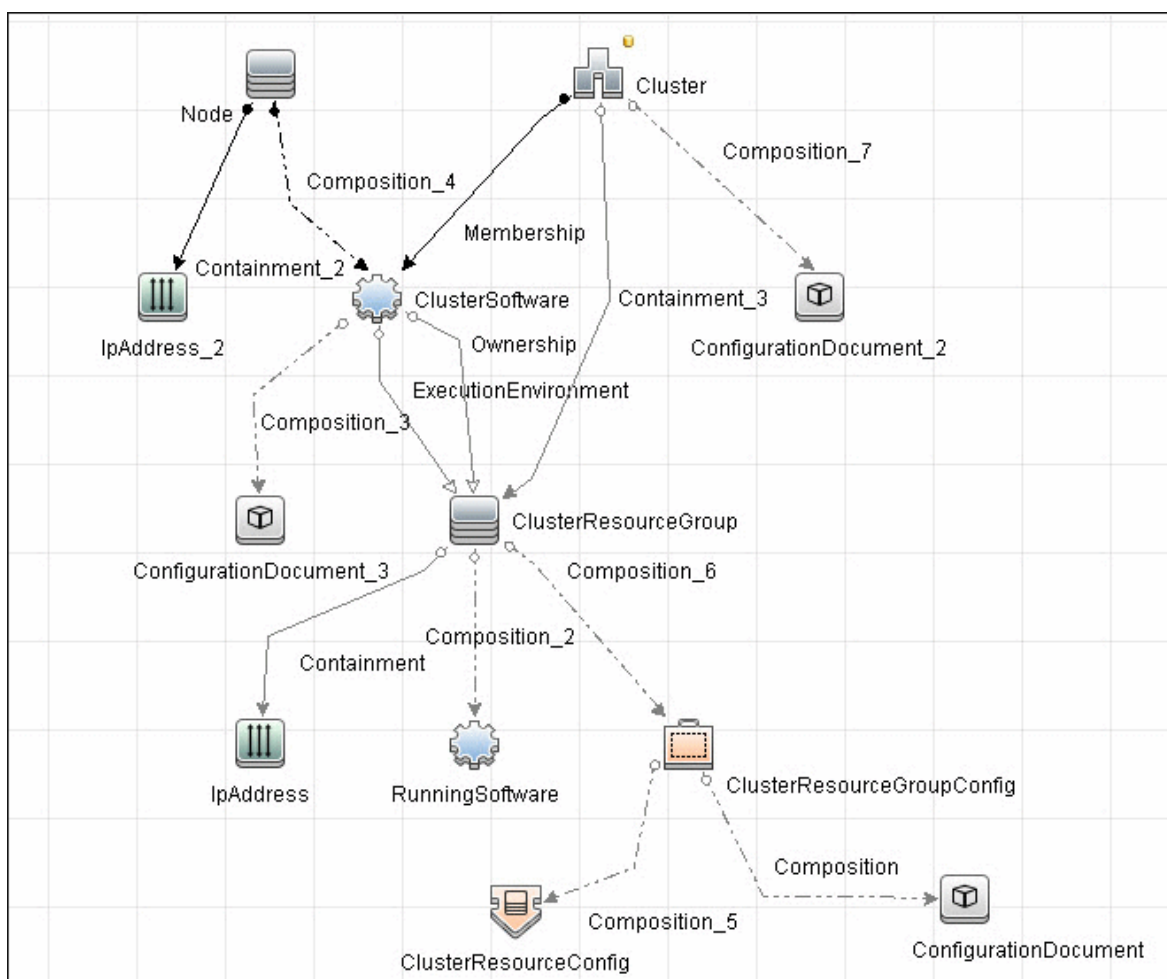
Veritas Cluster はクラスタ化されたいくつかのサーバで構成されます。各サーバは、特定のサービスやアプリケーションを実行します。サーバは互いのバックアップとして使用されます。システム・コンポーネントにエラーが発生した場合、必要なサービスを提供する役割は別のサーバに引き継がれます。

## サポートされるバージョン

UNIX 用 Veritas Cluster Server(VCS)2.x, 3.x, 4.x, 5.x

このビューには、Veritas Cluster トポロジの最上位レイヤが表示されます。ここに、検出された Veritas Cluster と、そのクラスタのメンバであるクラスタ化されたソフトウェア・リソースが表示されます。各ソフトウェア・リソースは、Veritas Cluster への **membership** 関係によってリンクされます。

注: 検出された CIT のリストについては、[「Veritas Cluster by Shell ジョブ」\(356ページ\)](#)を参照してください。



## Veritas Cluster Server の検出方法

Veritas Cluster ディスカバリ・プロセスでは、Veritas Cluster Servers(VCS)とそれらのメンバ・マシン(ノードとも呼ばれる)を検出できます。これにより、クラスタが提供する検出されたリソースがアクティブになります。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、SSH プロトコルまたは Telnet プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次のジョブを次の順序で実行します。

- a. **Host Connection by Shell** ジョブを実行します。
- b. **Host Applications by Shell** ジョブを実行します。
- c. **Veritas Cluster by Shell** ジョブを実行します。ジョブの詳細については、[「Veritas Cluster by Shell ジョブ」\(356ページ\)](#)を参照してください。

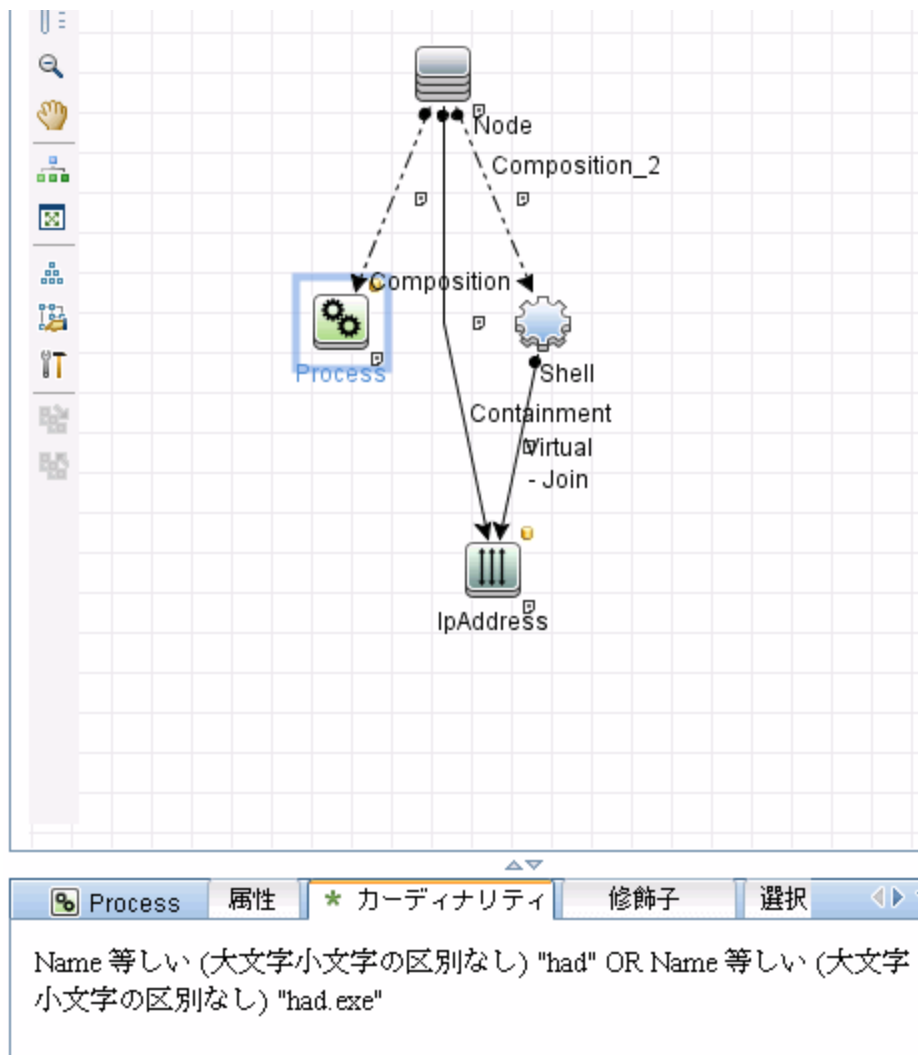
## Veritas Cluster by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

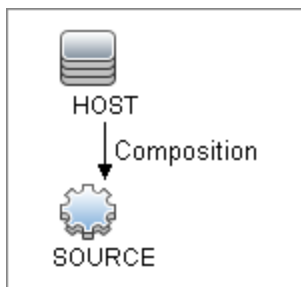
- トリガ・クエリ:





## アダプタ

- ・ 入力クエリ:



## 使用スクリプト

- file\_ver\_lib.py
- Veritas\_Cluster\_Topology.py

## 検出された CIT

- ClusterSoftware
- Composition
- ConfigurationDocument
- Containment
- Dependency
- IpAddress
- IpServiceEndpoint
- Membership
- Node
- Ownership
- RunningSoftware

- Usage
- VCS Resource Group
- VCS resource
- Veritas Cluster

注: トポロジを表示するには, [「トポロジ」\(355ページ\)](#)を参照してください。

## 第4部: Clustering and Load Balancing > Load Balancers

## 第20章: Load Balancer ディスカバリ

### 本章の内容

概要 .....	362
サポートされるバージョン .....	362
トポロジ .....	363
ロード・バランサの検出方法 .....	363
Alteon Application Switch by SNMP ジョブ .....	366
アダプタ情報 .....	367
F5 BIG-IP LTM by SNMP ジョブ .....	369
アダプタ情報 .....	370
F5 BIG-IP LTM by Shell ジョブ .....	372
バージョン .....	372
前提条件 .....	372
アダプタ情報 .....	372
Cisco CSS by SNMP ジョブ .....	376
アダプタ情報 .....	377
Citrix NetScaler by SNMP ジョブ .....	379
アダプタ情報 .....	380
検出された CIT .....	381

## 概要

DFM は次のロード・バランサを検出します。

- F5 BIG-IP Local Traffic Manager(LTM)
- Nortel Application Switch(以前の Alteon Application Switch)
- Cisco Content Services Switch(CSS)
- Citrix NetScaler

## サポートされるバージョン

各ロード・バランサのサポートされるバージョンは次のとおりです。

- **F5 BIG-IP Local Traffic Manager** : バージョン 4, 9, 10
- **Nortel Application Switch** : 既知の制限なし
- **Cisco Content Services Switch** : 既知の制限なし
- **Citrix NetScaler** : 既知の制限なし

[illegible]

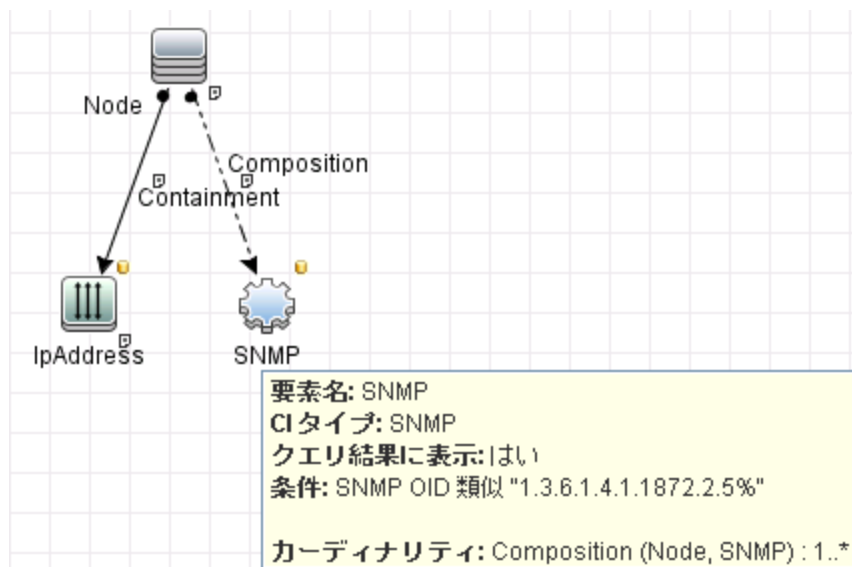
## ロード・ balanサの検出方法

- 「前提条件」(363ページ)
- 「ディスカバリの実行」(365ページ)

**Host Connection by SNMP** ジョブを実行して、次の要件を満たす SNMP CI を検出し、作成しま

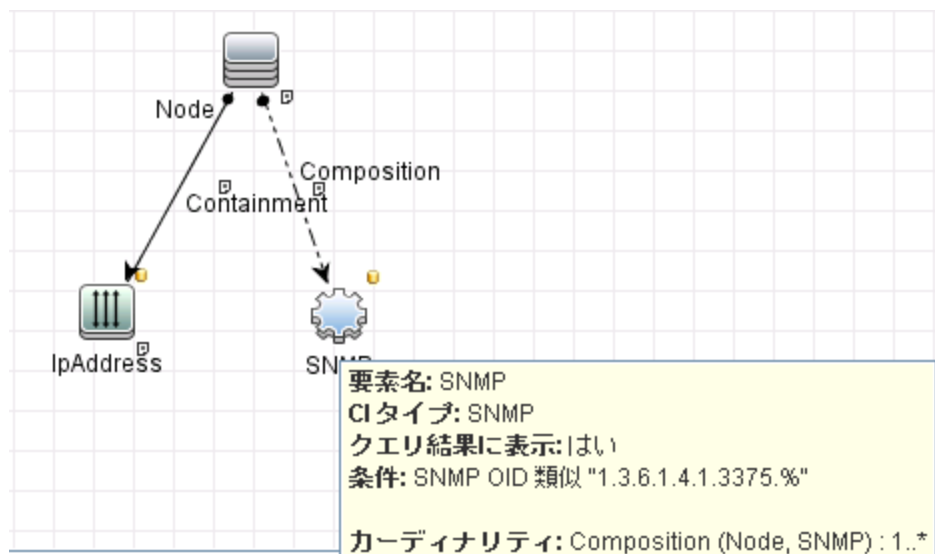
す。

- 次の条件で **Alteon application switch by SNMP** ジョブのトリガ・クエリとなること。



SNMP OID 類似 1.3.6.1.4.1.1872.2.5%

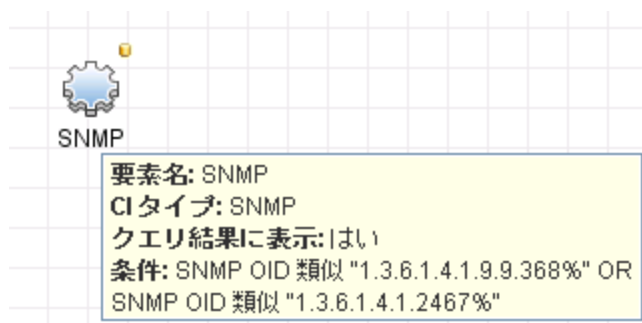
- 次の条件で **F5 BIG-IP LTM by SNMP** ジョブのトリガ・クエリとなること。



SNMP OID 類似 1.3.6.1.4.1.3375%

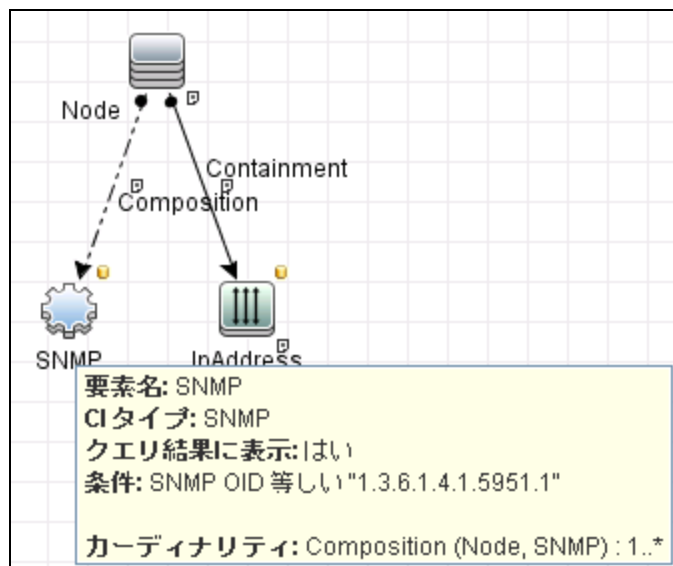


- 次の条件で **Cisco CSS by SNMP** ジョブのトリガ・クエリとなること。



SNMP OID 類似 1.3.6.1.4.1.9.9.368% OR 1.3.6.1.4.1.2467%

- 次の条件で **Citrix NetScaler by SNMP** ジョブのトリガ・クエリとなること。



SNMP OID 等しい 1.3.6.1.4.1.5951.1

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. ディスカバリの実行

- **Host Connection by SNMP を実行します。** ロード・バランサ・ジョブを実行するための前提条件の詳細については、[「前提条件」\(363ページ\)](#)を参照してください。
- 次のいずれかのジョブを実行します。

- **F5 BIG-IP LTM by SNMP**
- **Alteon application switch by SNMP**
- **Cisco CSS by SNMP**
- **Citrix NetScaler by SNMP**

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Alteon Application Switch by SNMP ジョブ

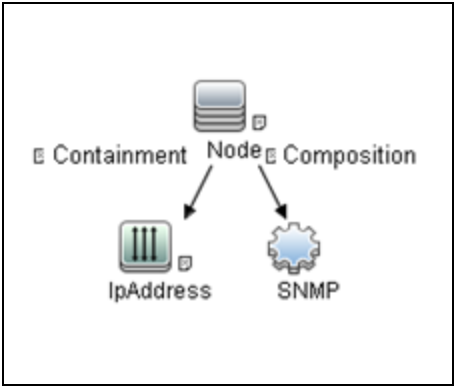
このジョブは、Nortel(Alteon)ロード・バランサとそれに関連するすべての CI を検出します。

### クエリされる SNMP テーブル

次の SNMP テーブルがクエリされます。

テーブル名	MIB の名 前	OID
Virtual servers	slbCurCfgVirtServer Table	1.3.6.1.4.1.1872.2.5.4.1.1.4.2.1
Virtual services	slbCurCfgVirtServices Table	1.3.6.1.4.1.1872.2.5.4.1.1.4.5.1
Real groups	slbCurCfgGroupEntry	1.3.6.1.4.1.1872.2.5.4.1.1.3.3.1
Real servers	slbCurCfgRealServer Table	1.3.6.1.4.1.1872.2.5.4.1.1.2.2.1
Port links	slbCurCfgRealServPortTable	1.3.6.1.4.1.1872.2.5.4.1.1.2.5.1
Ports	slbCurCfgPortTable	1.3.6.1.4.1.1872.2.5.4.1.1.5.2.1

トリガ・クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
SNMP	SNMP OID 類似 1.3.6.1.4.1.1872.2.5%
Node	なし

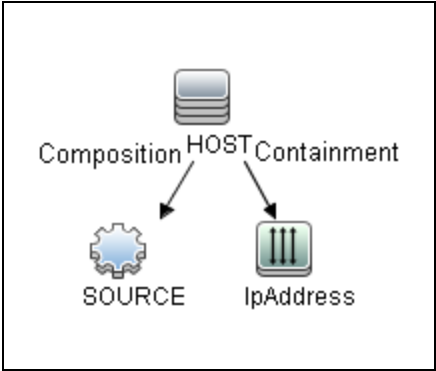
アダプタ情報

このジョブでは、**Alteon\_app\_switch\_by\_SNMP** アダプタを使用します。

入力 CIT

SNMP

入力 TQL クエリ



ノード名	条件
SOURCE	なし
IpAddress	IpAddressType 等しい IPv4
HOST	なし

## トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}
oid	\${SOURCE.snmp_oid}

## 使用スクリプト

- snmputils.py
- Alteon\_app\_switch\_by\_SNMP.py

## 検出された CIT

- Alteon application switch
- Composition
- ConfigurationDocument
- Containment
- IpAddress
- IpServiceEndpoint
- Load Balancer
- Load Balancing Cluster
- Membership

- Node
- Ownership

## F5 BIG-IP LTM by SNMP ジョブ

このジョブは、SNMP を使用して F5 BIG-IP Local Traffic Manager(LTM)を検出します。DFM は、F5 に関連するすべての SNMP を選択し、それらに対して実行します。

このパッケージは F5 BIG-IP LTM のバージョン 11, 10, 9, 4 をサポートしています。

### クエリされる SNMP テーブル

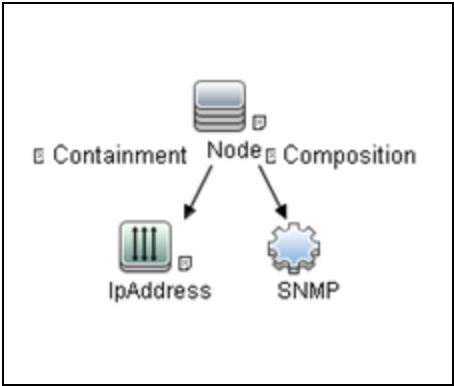
バージョン 10 と 9 では、次の SNMP テーブルがクエリされます。

テーブル名	MIB の名前	OID
General information	sysProduct	1.3.6.1.4.1.3375.2.1.4
Virtual servers	ltmVirtualServTable	1.3.6.1.4.1.3375.2.2.10.1.2.1
Pools	ltmPoolTable	1.3.6.1.4.1.3375.2.2.5.1.2.1
Pools to server	ltmVirtualServPool Table	1.3.6.1.4.1.3375.2.2.10.6.2.1
Pool members	ltmPoolMemberTable	1.3.6.1.4.1.3375.2.2.5.3.2.1
Rules to servers	ltmVirtualServRule Table	1.3.6.1.4.1.3375.2.2.10.8.2.1
Rules	ltmRuleTable	1.3.6.1.4.1.3375.2.2.8.1.2.1

バージョン 4 では、次の SNMP テーブルがクエリされます。

テーブル名	MIB の名前	OID
General information	globalAttributes	1.3.6.1.4.1.3375.1.1.1.1
Virtual servers	virtualServerTable	1.3.6.1.4.1.3375.1.1.3.2.1
Pools	poolTable	1.3.6.1.4.1.3375.1.1.7.2.1
Pool members	poolMemberTable	1.3.6.1.4.1.3375.1.1.8.2.1

トリガ・クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
SNMP	SNMP OID 類似 1.3.6.1.4.1.3375.%
Node	なし

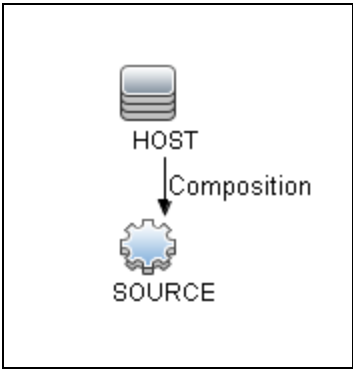
アダプタ情報

このジョブでは、**F5\_BIGIP\_LTM\_by\_SNMP** アダプタを使用します。

入力 CIT

SNMP

入力 TQL クエリ



ノード名	条件
SOURCE	なし
HOST	なし

## トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

## 使用スクリプト

- snmputils.py
- F5\_BIGIP\_LTM\_by\_SNMP.py

## 検出された CIT

- Composition
- ConfigurationDocument
- Containment
- F5 BIG-IP LTM
- IpAddress
- IpServiceEndpoint
- Load Balancer
- Load Balancing Cluster
- Membership
- Node
- Ownership

## F5 BIG-IP LTM by Shell ジョブ

このジョブは、シェルを使用して F5 BIG-IP Local Traffic Manager(LTM)を検出します。DFM は、F5 に関連するすべてのシェルを選択し、それらに対して実行します。

### バージョン

このパッケージは F5 BIG-IP LTM のバージョン 10.x と 11.x をサポートしています。

### 前提条件

このアダプタには、SSH プロトコルの資格情報(ユーザ名とパスワードを含む)が必要です。

注: F5 の構成ファイルは、すべてのユーザによる読み取りと、ルートによる書き込みが可能なため、ユーザはルートとしてログインする必要があります。

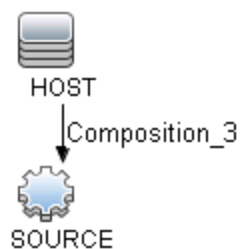
### アダプタ情報

このジョブでは、**F5\_BIGIP\_LTM\_by\_Shell** アダプタを使用します。このアダプタは、シェル・コマンドを使用して F5 BIG-IP LTM の構成ドキュメントを解析し、データを取得します。

#### 入力 CIT

Shell

#### 入力 TQL クエリ





ノード名	条件
SOURCE	なし
HOST	CI Type 等しい lb

## トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}

## 使用スクリプト

- F5\_BIGIP\_LTM\_by\_Shell.py

## 検出された CIT

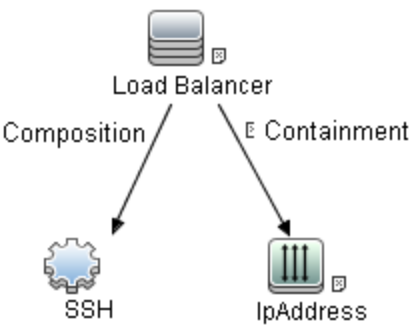
- Composition
- ConfigurationDocument
- Containment
- F5 BIG-IP LTM
- IpAddress
- IpServiceEndpoint
- Load Balancer
- Load Balancing Cluster

- Membership
- Node
- Ownership

## 必要な権限

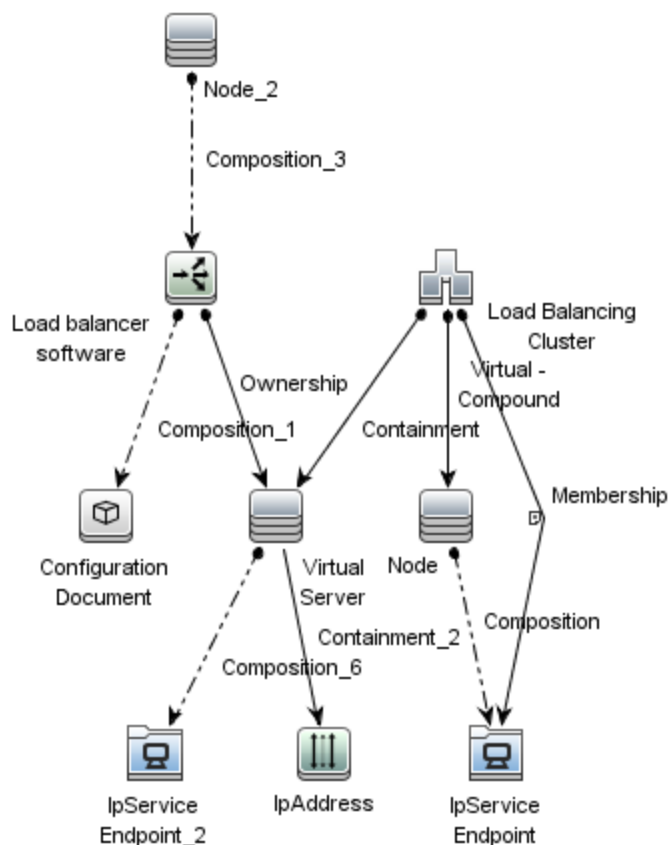
権限	操作	使用状況の詳細	オブジェクトとパラメータ
Shell	exec	Basic Login	UNIX : date +%z UNIX : echo \$? UNIX : echo \$SHELL UNIX : locale -a UNIX : uname -a UNIX : uname -r
Shell	exec	Discover files and F5 details	UNIX : cat <file_path> UNIX : find <folder_path> UNIX : ls -lA <folder_path> UNIX : perl -e UNIX : ps -eo user,pid,lstart,command --cols 2530 --no-headers UNIX : readlink

トリガ TQL



ノード名	条件
Load Balancer	なし
SSH	なし
IpAddress	NOT IP Probe Name null

## トポロジ



## Cisco CSS by SNMP ジョブ

このジョブは、SNMP を使用して Cisco Content Services Switch を検出します。このジョブでは Cisco CSS のすべてのバージョンがサポートされます。

このディスカバリを実行するには、**Cisco CSS by SNMP** ジョブをアクティブ化します。DFM は、Cisco CSS に関連するすべての SNMP を選択し、それらに対して実行します。

**注:** 一部のサービスについては、コンテンツ・ルールが定義されていないと、このパッケージでは検出されない場合があります。

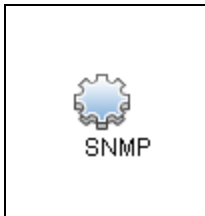
CSS の検出は 3 つのテーブル、**apCntTable**、**apSvcTable**、**apCntsvcTable** に基づいて行われます(次の表を参照)。

- **apCntTable** は、仮想アドレス、仮想サービス、プールに関する情報を提供します。
- **apSvcTable** は、プールに含まれる物理ホストに関する情報を提供します。
- **apCntsvcTable** は、どのホストがどのプールに含まれるかを記述します。

**apSvcTable** には、対応する行が **apCntsvcTable** がないエントリを含めることができます。この場合、該当するホストはスキップされます。

テーブル名	MIB の名前	OID
CNT	apCntTable	1.3.6.1.4.1.2467.1.16.4.1 or 1.3.6.1.4.1.9.9.3681.16.4.1
SVC	apSvcTable	1.3.6.1.4.1.2467.1.15.2.1 or 1.3.6.1.4.1.9.9.3681.15.2.1
CNT to SVC	apCntsvcEntry	1.3.6.1.4.1.2467.1.18.2.1 or 1.3.6.1.4.1.9.9.3681.18.2.1

## トリガ・クエリ



ノード名	条件
SNMP	SNMP OID 類似 1.3.6.1.4.1.9.9.368% OR SNMP OID 類似 1.3.6.1.4.1.2467%

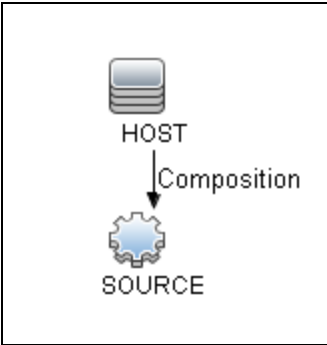
## アダプタ情報

このジョブでは、**Cisco CSS by SNMP** アダプタを使用します。

## 入力 CIT

SNMP

入力 TQL クエリ



ノード名	条件
SOURCE	なし
HOST	なし

トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}
hostId	\${HOST.root_id}

使用スクリプト

- snmputils.py
- Cisco\_CSS\_by\_SNMP.py

検出された CIT

- Cisco CSS
- Composition
- Containment

- IpAddress
- IpServiceEndpoint
- Load Balancer
- Load Balancing Cluster
- Membership
- Node
- Ownership

## Citrix NetScaler by SNMP ジョブ

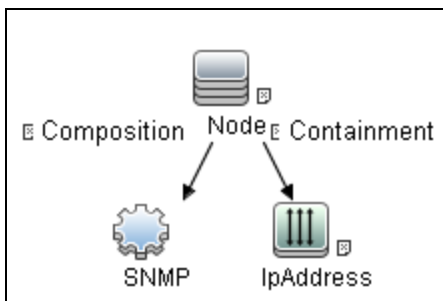
このジョブは、SNMP を使用して Citrix NetScaler トポロジを検出します。

### クエリされる SNMP テーブル

次の SNMP テーブルがクエリされます。

テーブル名	MIB の名 前	OID
Server	serverTable	1.3.6.1.4.1.5951.4.1.2.2
Service	serviceTable	1.3.6.1.4.1.5951.4.1.2.1
Service Group	serviceGroupMemberTable	1.3.6.1.4.1.5951.4.1.2.7
Virtual Server	vserverTable	1.3.6.1.4.1.5951.4.1.3.1
Virtual Service	vserverServiceTable	1.3.6.1.4.1.5951.4.1.3.2

### トリガ TQL



ノード名	条件
SNMP	SNMP OID 等しい 1.3.6.1.4.1.5951.1

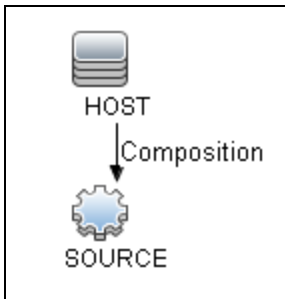
## アダプタ情報

このジョブでは, **Citrix NetScaler by SNMP** アダプタを使用します。

### 入力 CIT

SNMP

### 入力 TQL



ノード名	条件
SOURCE	なし
HOST	なし

### トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

### 使用スクリプト

- snmputils.py
- Citrix\_NetScaler\_by\_SNMP.py



## 検出された CIT

- Composition
- ClusterResourceGroup
- Containment
- Citrix NetScaler
- IpAddress
- IpServiceEndpoint
- Load Balancer
- Load Balancing Cluster
- Membership
- Node
- Ownership

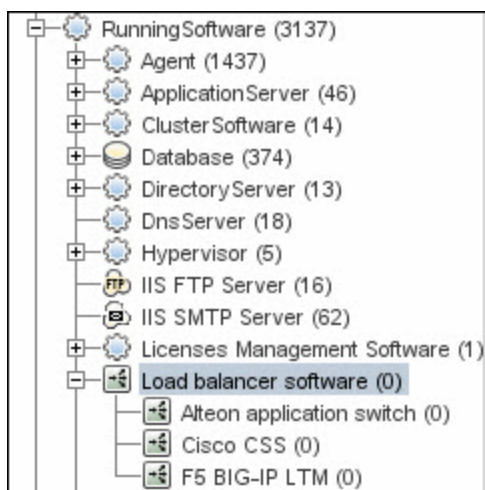
## 検出された CIT

次の CIT は、ロード・バランサ・トポロジをモデリングします。

- **Load Balancer Software**

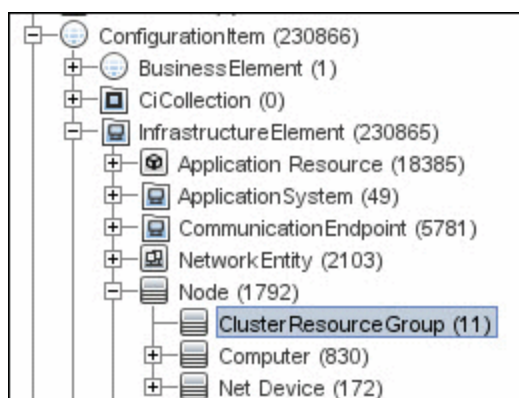
この CIT は、負荷分散ソリューションを提供するソフトウェアを表します。サポートされるロード・バラン

サの詳細については、「[概要](#)」(362ページ)を参照してください。



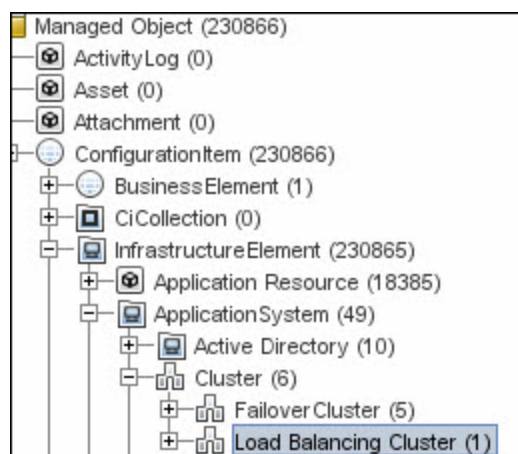
## • クラスタ・サーバ

クラスタ・サーバは、トラフィックの負荷をサーバ・プールに分散する、システム上のトラフィック管理オブジェクトです。クラスタ・サーバにより、クライアント要求の処理に使用するリソースの可用性が高まります。クラスタ・サーバの最大の機能は、要求を受け取り、それらを指定した基準に従ってプール・メンバーに分散することです。



## • Load Balancing Cluster

負荷分散クラスタ(またはプール)は、トラフィックの受信と処理を行うデバイスをグループ化した、デバイスの論理セットです。クライアント要求で指定された宛先 IP アドレスにクライアント・トラフィックを送信する代わりに、仮想サーバが、プール内のいずれかのメンバー・サーバに要求を送信します。これは、サーバ・リソースに対する効率的な負荷分散に役立ちます。



注: トポロジを表示するには, 「[トポロジ](#)」(363ページ)を参照してください。

## 第21章: Microsoft Network Load Balancing (NLB)ディスカバリ

### 本章の内容

概要 .....	385
サポートされるバージョン .....	385
トポロジ .....	386
Microsoft NLB(ネットワーク負荷分散)システムの検出方法 .....	387
コマンド・ライン・ユーティリティを使用した NLB の検出方法 .....	388
MS NLB by NTCMD or UDA ジョブ .....	389
MS NLB by NTCMD or UDA アダプタ .....	391
NLB(ネットワーク負荷分散)アーキテクチャのコンポーネント .....	396
用語集 .....	397

## 概要

ネットワーク負荷分散(NLB)は、クラスタ内のホスト上でそれぞれが実行されている Web サーバなどの TCP/IP サービスの複数のコピー(またはインスタンス)に IP トラフィックを分散します。NLB はクライアント要求をホスト間で透過的に分割し、クライアントが1 つまたは複数の仮想 IP アドレスを使用してクラスタにアクセスできるようにします。クライアント側からは、クラスタはクライアント要求に応答する単一のサーバに見えます。各サーバはすべてのクライアント要求を受信しますが、どのサーバが応答するかは NLB が決定します。

Microsoft NLB クラスタの機能を担うすべてのコンポーネントは、**Microsoft\_NLB\_Cluster.zip** パッケージにバンドルされています。

MS-NLB を検出する方法については、[「Microsoft NLB\(ネットワーク負荷分散\)システムの検出方法」\(387ページ\)](#)を参照してください。

### 関連情報

- [「NLB\(ネットワーク負荷分散\)アーキテクチャのコンポーネント」\(396ページ\)](#)
- [「用語集」\(397ページ\)](#)

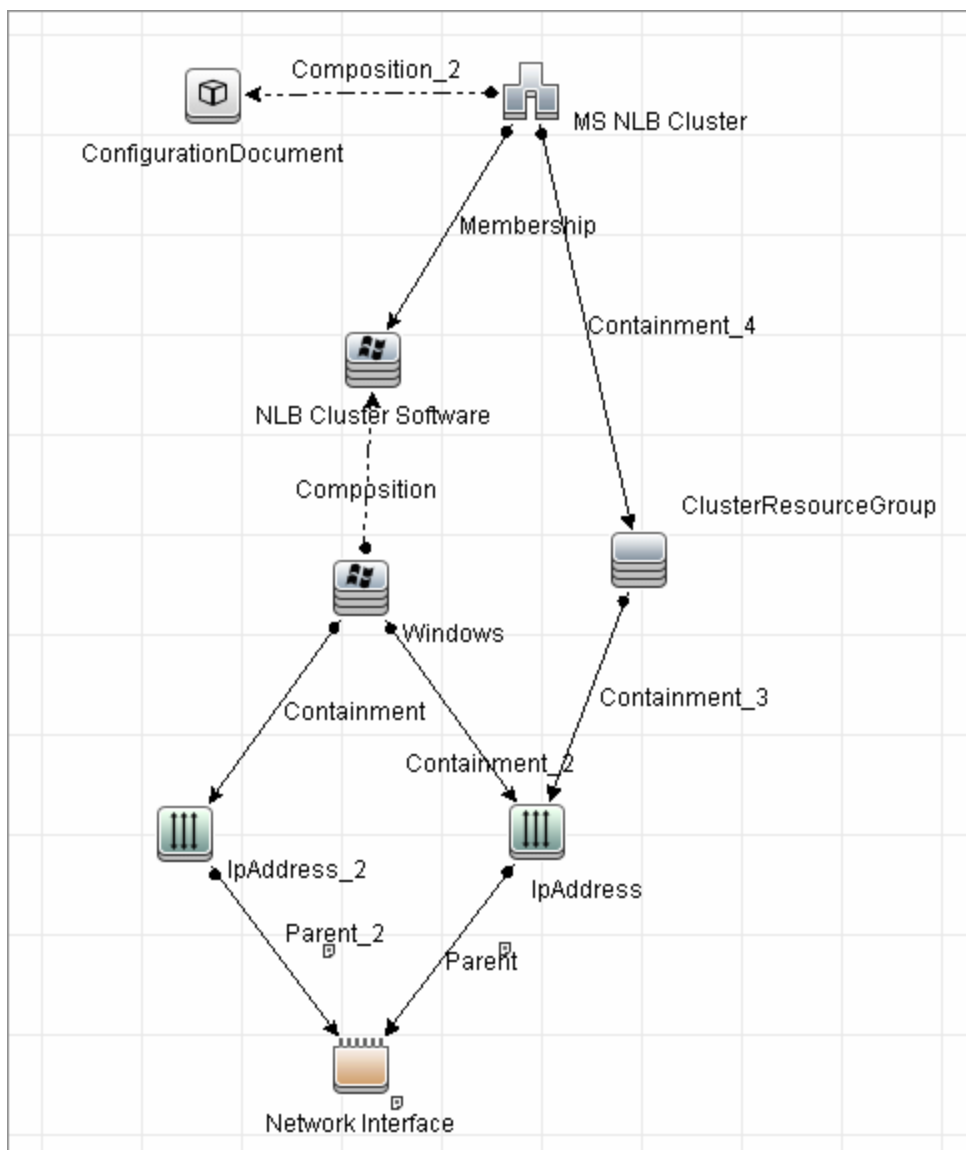
## サポートされるバージョン

このディスカバリは、Microsoft ネットワーク負荷分散バージョン 2000, 2003, 2008, 2012, および 2012 R2 をサポートしています。

## トポロジ

次の図に、MS NLB ディスカバリのトポロジを示します。

注: 検出された CIT のリストについては、[「MS NLB by NTCMD or UDA ジョブ」\(389ページ\)](#)を参照してください。



# Microsoft NLB(ネットワーク負荷分散)システムの検出方法

本項の内容

## 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、NTCMD プロトコル、または Exchange Server に UD エージェントがインストールされている場合は Universal Discovery プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

NTCMD プロトコルで定義されているユーザに、リモート・マシン上でシェルを実行する管理権限が付与されていることを確認します。

**wlbs params** コマンドを実行することで、NTCMD プロトコルによって NLB に関する情報が取得されます。

## 2. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次のジョブを次の順序でアクティブにします。

- NLB ディスカバリのトリガとして機能する Windows マシンを検出するための **Host Connection by Shell** ジョブ。
- NTCMD によってホストに接続し、MS NLB クラスタ・トポロジを取得するための **MS NLB by NTCMD or UDA** ジョブ。ジョブの詳細については、[「MS NLB by NTCMD or UDA ジョブ」\(389 ページ\)](#)を参照してください。

ディスカバリ・メカニズムの詳細については、[「MS NLB by NTCMD or UDA ジョブ」\(389ページ\)](#)を参照してください。

## コマンド・ライン・ユーティリティを使用した NLB の検出方法

NLB は、**nlb.exe** コマンド・ライン・ユーティリティを実行することで検出できます。

このユーティリティは、**params** キーを指定して実行され、検出されたマシン上のすべての NLB クラスタに関する情報を出します。

- NLB が Windows 2003 Server マシン上にインストールされていない場合、出力は次のようになります。

```
WLBS Cluster Control Utility V2.4 (c) 1997-2003 Microsoft Corporation.  
WLBS is not installed on this system or you do not have sufficient privileges to  
administer the cluster.
```

- NLB クラスタがそのマシン上で設定されている場合、出力は次のようになります。

```
Cluster 192.168.0.222  
Retrieving parameters  
Current time           = 9/3/2009 1:02:38 PM  
HostName               = ddmvm-2k3-s  
ParametersVersion      = 4  
CurrentVersion         = 00000204  
EffectiveVersion       = 00000201  
InstallDate            = 4A9E51F5  
HostPriority            = 1  
ClusterIPAddress       = 192.168.0.222  
ClusterNetworkMask     = 255.255.255.0  
DedicatedIPAddress     = 192.168.0.2  
DedicatedNetworkMask   = 255.255.255.0  
McastIPAddress         = 0.0.0.0  
ClusterName            = cluster2.domain.com  
ClusterNetworkAddress  = 03-bf-c0-a8-00-de  
IPToMACEnable          = ENABLED  
MulticastSupportEnable = ENABLED  
IGMPSupport            = DISABLED  
MulticastARPEnable     = ENABLED  
MaskSourceMAC          = ENABLED  
AliveMsgPeriod         = 1000  
AliveMsgTolerance      = 5  
NumActions             = 100  
NumPackets             = 200  
NumAliveMsgs           = 66  
DescriptorsPerAlloc    = 512
```



```

MaxDescriptorAllocs      = 512
TCPConnectionTimeout     = 60
IPSecConnectionTimeout   = 86400

FilterICMP                = DISABLED
ClusterModeOnStart       = STARTED
HostState                 = STARTED
PersistedStates           = NONE
ScaleSingleClient         = DISABLED
NBTSupportEnable          = ENABLED
NetmonAliveMsgs           = DISABLED
IPChangeDelay             = 60000
ConnectionCleanupDelay    = 300000
RemoteControlEnabled      = DISABLED
RemoteControlUDPPort      = 2504
RemoteControlCode         = 00000000
RemoteMaintenanceEnabled = 00000000
BDATeaming                = NO
TeamID                    =
Master                    = NO
ReverseHash               = NO
IdentityHeartbeatPeriod   = 10000
IdentityHeartbeatEnabled  = ENABLED
PortRules (1):
      VIP      Start  End  Prot  Mode  Pri Load Affinity
-----
All           0 65535 Both Multiple      Eql Single

```

出力を CIT にマッピングするために使用される特別なルールはありません。すべての CI 属性が出力データ名ごとに繰り返されます。このデータは、検出済みのクラスタ・ノードと比較することによって検証されます。

## MS NLB by NTCMD or UDA ジョブ

本項ではこのジョブに関する詳細を説明します。

### ディスカバリ・メカニズム

DFM は、複数(2 つ以上)の IP アドレスを持つ Windows マシン上でトリガされ、**nlb.exe** コマンド・ライン・ユーティリティを使用して情報を収集します(Windows 2000 ファミリの初期のバージョンでは **wlbs.exe** を使用)。これらのユーティリティによって、すべての NLB 関連情報を取得できます。詳細については、「[MS NLB by NTCMD or UDA アダプタ](#)」(391 ページ)を参照してください。

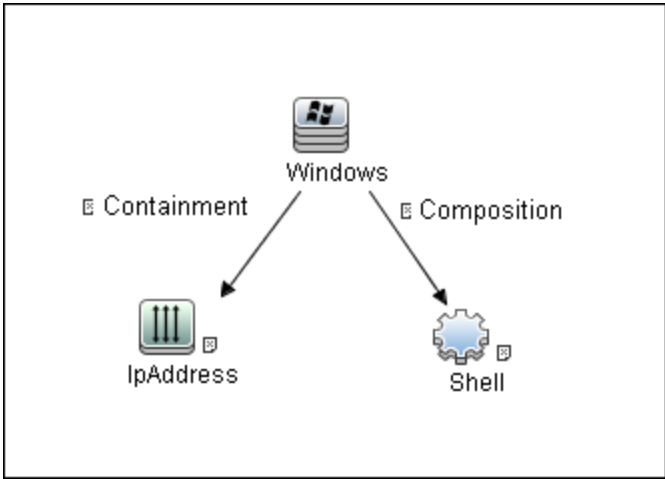
MS NLB クラスタ・システムが存在することを確認するために、クラスタに参加するすべてのノードからの情報を DFM が収集する必要はありません。このソフトウェアを実行中の 1 台のマシンがあれば、それはクラ

スタ・マシンとみなされます。NLB サービスが含まれる複数のマシンが検出された場合 (最初 のマシンと同じ設定を持つ), NLB クラスタは集合 処理を開始します。

なお, クラスタ内に参加しているノードにはほかの参加ノードの情報が含まれないため, クラスタ情報は, 一度に 1 つのノードを検出することによって収集されます。

トリガ・クエリ

- トリガ CIT : **NTCMD**
- トリガ・クエリ:



- **CI 属性条件** : 2 つ以上の IP アドレスを持つ Windows マシン上で実行されている NTCMD または UDA。

名 前	カテゴリ	説 明
ntcmd_with_2_IP	トリガ	<b>MS NLB by NTCMD or UDA</b> ジョブが使用
MS NLB topology	ビュー	<b>MS NLB トポロジ・ビュー</b> が使用

アダプタ

このジョブでは, **MS NLB by NTCMD or UDA** アダプタを使用します。詳細については, 「[MS NLB by NTCMD or UDA アダプタ](#)」(391ページ)を参照してください。

## ビュー

- Microsoft NLB topology

## 使用スクリプト

- ms\_nlb\_report\_utils.py
- ms\_nlb\_ntcmd.py

## 検出された CIT

- Composition
- ConfigurationDocument。詳細については、[「MS NLB by NTCMD or UDA アダプタ」\(391ページ\)](#)を参照してください。
- Containment
- IpAddress
- Membership
- MS NLB Cluster。詳細については、[「MS NLB by NTCMD or UDA アダプタ」\(391ページ\)](#)を参照してください。
- NLB Cluster Software。詳細については、[「MS NLB by NTCMD or UDA アダプタ」\(391ページ\)](#)を参照してください。
- Node

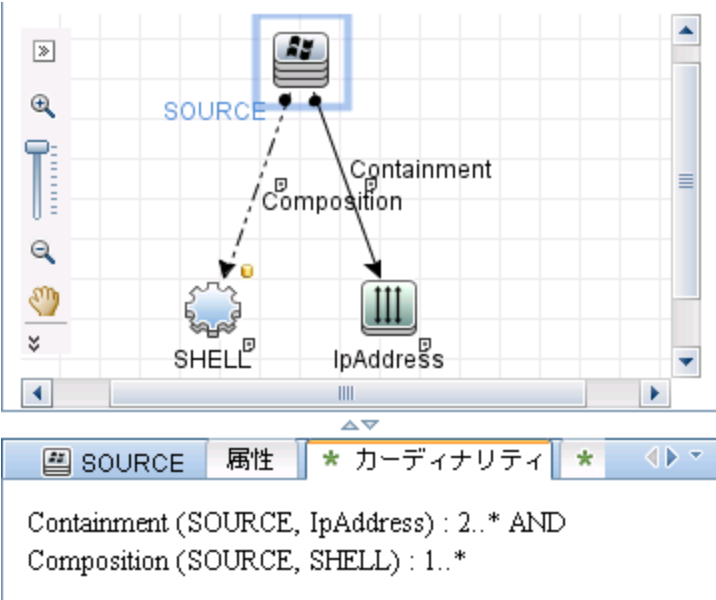
注: トポロジを表示するには、[「トポロジ」\(386ページ\)](#)を参照してください。

## MS NLB by NTCMD or UDA アダプタ

本項ではこのアダプタに関する詳細を説明します。

入カクエリ

下図に示すとおり, 2 つ以上の IP アドレスを持つ Windows マシン上で実行されている NTCMD または UDA の入カクエリです。



トリガ CI データ

名前	値
credentialsId	\${SHELL.credentials_id}
ip_address	\${IpAddress.name}

MS NLB Cluster CIT

この CIT は, NLB クラスタに関する情報を表します。

- CIT 名 : **ms\_nlb\_cluster**
- 親 CIT 名 : **loadbalancecluster**
- 関係

開始ノード	開始ノード・カード ディナリティ	名前	終了ノード	終了ノード・カード ディナリティ
ms_nlb_cluster	1..*	membership	nlb_ clustersoftware	1..*

クラスタ IP アドレスは、NLB を検出する方法として最も信頼性が高いキー・フィールドです。これに対し、クラスタ・ネットワーク・アドレスを介して NLB を検出する方法は、このアドレスが IP アドレスおよびオペレーティング・モード(ユニキャスト、マルチキャスト、IGMP)に依存するために信頼性が低くなります。クラスタ・ドメイン名はクラスタ名から取得されます。

- 属性

次に、MS NLB Cluster CIT に固有の属性を示します。

キー	表示名	属性名	タイプ
X	ClusterIPAddress	cluster_ip_address	文字列型 (15)
	ClusterNetworkMask	cluster_network_ mask	文字列型 (15)
	McastIPAddress	mcast_ip_address	文字列型 (15)
	ClusterDomainName	cluster_domain_name	文字列型 (256)
	ClusterNetworkAddresses	cluster_network_ address	MAC アドレス
	IPToMACEnable	ip_to_mac_enable	ブール型
	MulticastSupportEnable	multicast_support_ enable	ブール型
	IGMPSupport	igmp_support	ブール型
	RemoteControlEnabled	remote_control_ enabled	ブール型
X	Name	name	文字列型 (この CIT 用に変更)

## NLB Cluster Software CIT

この CIT は、NLB クラスタに属する単一のマシン構成の情報を表します。

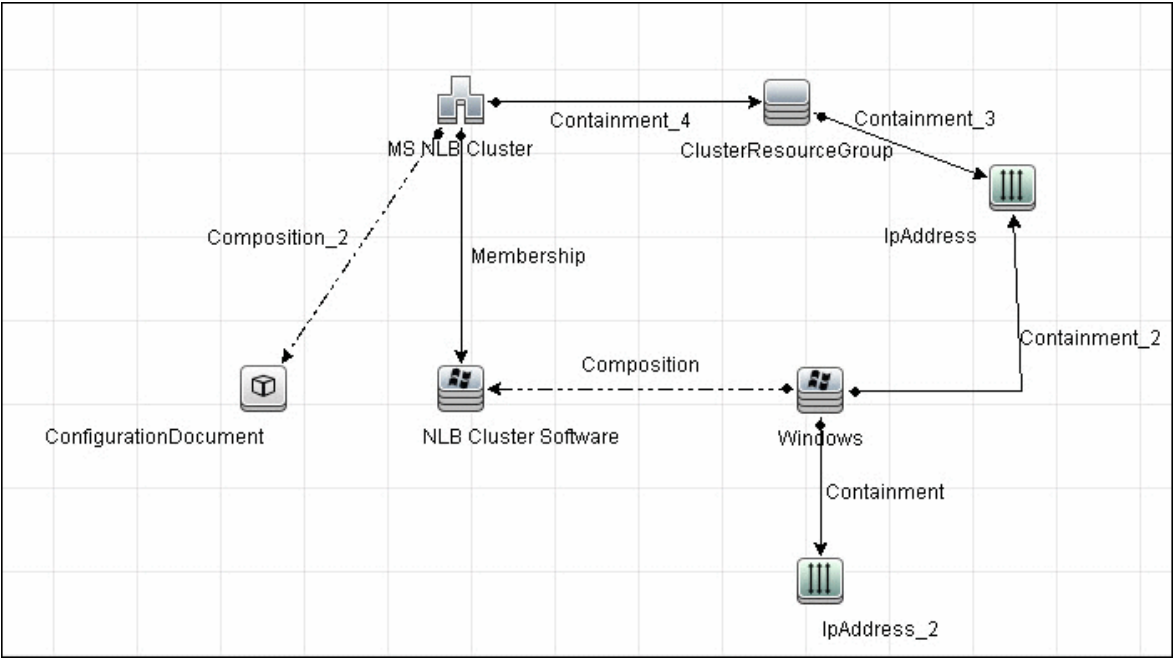
- CIT 名 : **nlb\_clustersoftware**
- 親 CIT 名 : **failoverclustersoftware**
- 関係

開始ノード	開始ノード・カードィナリティ	名前	終了ノード	終了ノード・カードィナリティ
ms_nlb_cluster	1..*	membership	nlb_clustersoftware	1..*
nt	1..*	composition	nlb_clustersoftware	1..*

- 属性

キー	表示名	タイプ
	ClusterIPAddress	文字列型 (15)
	HostPriority	整数型 (1～32)
	ClusterModeOnStart	起動済み, 中断, 停止
	Name	文字列型 (NLB Cluster SW)
	Composition	文字列型 (32)

ConfigurationDocument(NLB ポート・ルール)



この CIT は、NLB クラスタ用に定義された各ポート・ルールに関する情報を取得します。

ポート・ルール・エンティティは、キー属性を明確に定義することはできないため、ポート・ルール・プロパティは次のように キー=値 のペアで プロパティ・ファイルに格納されます。

```
portRule1.ServingIP=All
portRule1.StartPort=0
portRule1.EndPort=100
portRule1.Protocol=Both
portRule1.FilteringMode=Multiple
portRule1.Affinity=Single
portRule1.LoadWeight=40
```

関係

開始ノード	開始ノード・カードィナリティ	名前	終了ノード	終了ノード・カードィナリティ
nt	1..*	composition	nlb_clustersoftware	1..*
ms_nlb_cluster	1..*	membership	nlb_clustersoftware	1..*

## NLB(ネットワーク負荷分散)アーキテクチャのコンポーネント

コンポーネント	説明
Nlb.exe	ネットワーク負荷分散制御プログラムです。Nlb.exe はネットワーク負荷分散を開始、停止、管理するためにコマンド・ラインから使用します。ポートの有効化および無効化、クラスタの状態の照会にも使用できます。
Nlbmgr.exe	ネットワーク負荷分散マネージャ制御プログラムです。このコマンドは、ネットワーク負荷分散マネージャを開始するために使用します。
Wlbs.exe	以前のネットワーク負荷分散制御プログラムです。 <b>Nlb.exe</b> に置き換えられましたが、 <b>Wlbs.exe</b> を参照する既存のスクリプトがある場合など、必要に応じて <b>Nlb.exe</b> の代わりに <b>Wlbs.exe</b> を使用できます。
Wlbsprov.dll	ネットワーク負荷分散 WMI プロバイダです。
Nlbmprov.dll	ネットワーク負荷分散マネージャ WMI プロバイダです。
Wlbsctrl.dll	ネットワーク負荷分散 API DLL です。
Wlbs.sys	ネットワーク負荷分散デバイスドライバです。 <b>Wlbs.sys</b> はクラスタ内の各ホスト上にロードされます。受信する各要求をどのホストが処理するかを決定する目的で、クラスタ・ホストが集合的に使用する統計的マッピング・アルゴリズムがここに含まれます。



## 用語集

### クラスタ

独立したコンピュータの集まりであり、これらが連動して共通するアプリケーション・セットを実行し、クライアントおよびアプリケーションに対して単一のシステムとしてのイメージを提供します。これらのコンピュータはケーブルで物理的に接続されており、クラスタ・ソフトウェアによってプログラム的に接続されます。この接続を介して、コンピュータはサーバ・クラスタ内でのフェイルオーバーやネットワーク負荷分散(NLB)クラスタ内での負荷分散などの問題解決機能を使用できます。詳細については、[http://technet.microsoft.com/en-us/library/cc784941\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc784941(WS.10).aspx) を参照してください。

### 専用 IP アドレス

NLB クラスタと関連付けられていないネットワーク・トラフィックに使用される NLB ホストの IP アドレスを意味します。例として、クラスタ内の特定のホストに対する Telnet アクセスがあります。この IP アドレスは、クラスタ内の各ホストを個別に指定するために使用されます。そのため、ホストごとに一意です。

### NLB ノード

NLB クラスタに属するマシンを示します。詳細については、[http://technet.microsoft.com/en-us/library/cc758834\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc758834(WS.10).aspx) を参照してください。

### オペレーティング・モード

NLB クラスタには、次の2つのオペレーティング・モードがあります。

- 標準設定のユニキャスト・モード・オペレーションでは、NLB が有効化されているネットワーク・アダプタに対し、NLB はステーション(MAC)アドレスの割り当てを変更します。すべてのクラスタ・ホストに同じ MAC(media access control)アドレスが割り当てられます。
- マルチキャスト・モードでは、NLB は、クラスタ・アダプタのステーション・アドレスを変更する代わりに、クラスタ・アダプタにレイヤ2 マルチキャスト・アドレスを割り当てます。詳細については、[http://technet.microsoft.com/en-us/library/cc783135\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc783135(WS.10).aspx) を参照してください。

### ポート・ルール

NLB ドライバはポート・ルールを使用し、このルールに負荷分散対象となる、または無視されるトラフィックが示されます。標準設定では、NLB ドライバによって負荷分散のためのすべてのポートが構成されます。受信ネットワーク・トラフィックの負荷をポート単位で分散する方法を決定する NLB ドライバの構成は、

ポート・ルールを作成することで変更できます。これは、必要に応じてポート・グループごと、または個々のポートに対して作成します。各ポート・ルールに、ポート範囲パラメータが適用されるポートまたは複数のポートを使用するクライアント要求の負荷分散方法を構成します。アプリケーションの負荷分散方法は、ポート・ルールの追加または変更方法を定義することでほぼ決まります。このルールは、各ホスト上で特定のポート範囲に対して作成します。

## 仮想 IP アドレス

NLB クラスターのホスト全体で共有される IP アドレスを意味します。NLB クラスターは、マルチホームの Web サーバのクラスターなどで、複数の仮想 IP アドレスを使用することもあります。詳細については、[http://technet.microsoft.com/en-us/library/cc756878\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc756878(WS.10).aspx) を参照してください。

## 第5部: Database

## 第22章: ホスト 資格 情報によるデータベース接続 ディスカバリ

### 本章の内容

概要 .....	401
サポートされるバージョン .....	401
トポロジ .....	402
Oracle .....	402
Microsoft SQL .....	402
ホスト 資格 情報によるデータベース接続の検出方法 .....	403
DB Connections by Shell ジョブ .....	403
DB Connections by WMI ジョブ .....	407
トラブルシューティングおよび制限事項 .....	408

## 概要

このパッケージの目的は、HP Universal CMDB(UCMDB)にあるホスト・レベルの資格情報を使用してデータベースの自動検出を有効にすることです。DFM ユーザまたは管理者は、データベースに関する詳細情報(データベース名またはSID、リスナ・ポート番号など)を把握していない場合があります。このパッケージのソリューションは、そうした情報を最小限の入力で検出し、エンドツーエンドのデータベース・ディスカバリを可能にします。

DFM は、実行中のプロセス名、Windows サービス名、Windows レジストリ、構成ファイルなど、データベース・サーバや作成された CI にあるさまざまなソースからデータベース情報を抽出します。検出された Database CI は、データベース資格情報をポピュレートするため、Database Connection by SQL ジョブ (**Oracle Database Connection by SQL** ジョブなど)のトリガとして使用できます。これにより、定義済みのデータベース・トポロジ・ディスカバリ・ジョブを使用した、ディープ・ディスカバリが可能になります。

このパッケージのジョブ用の DFM トリガが設定されているので、これらのジョブは UCMDB のスパイラル・ディスカバリ・スケジュールにシームレスに組み込まれます。

このパッケージに含まれる **DB Connections by Shell** ジョブと **DB Connections by WMI** ジョブは、シェル(NTCMD/SSH/Telnet)またはエージェント(WMI)CI をトリガとして使用して、ホスト上のデータベース署名を検索します。これらのジョブは、インスタンス名やSID、データベース・サーバのリスナ・ポートなど、利用できる情報を使用してデータベース CI を作成します。データベース資格情報は使用されないため、ユーザ名と、これらの CI の資格情報 ID の属性は空の状態です。

これらのジョブの詳細については、次を参照してください。

- [「DB Connections by Shell ジョブ」\(403ページ\)](#)
- [「DB Connections by WMI ジョブ」\(407ページ\)](#)

## サポートされるバージョン

ホスト資格情報によるデータベース接続 ディスカバリは、次のデータベース・サーバをサポートしています。

- Oracle 9i, 10g, 11g
- Microsoft SQL Server 2000, 2005, 2008, 2008 R2, 2012, 2014

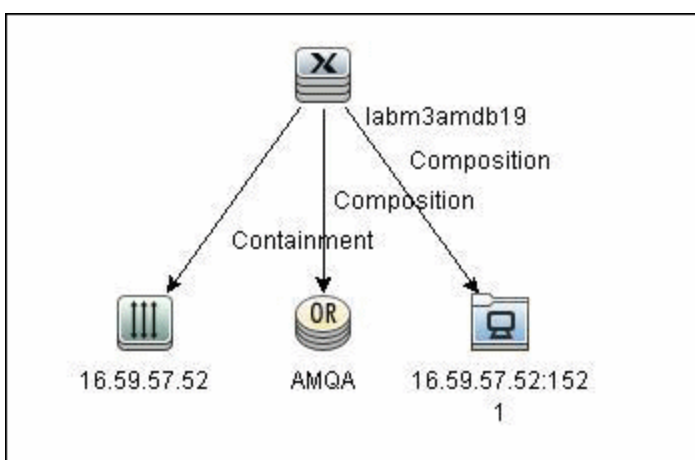
**注:** CP 13.0 以降のバージョンでは、DB2 データベース・サーバのディスカバリはサポートされません。

## トポロジ

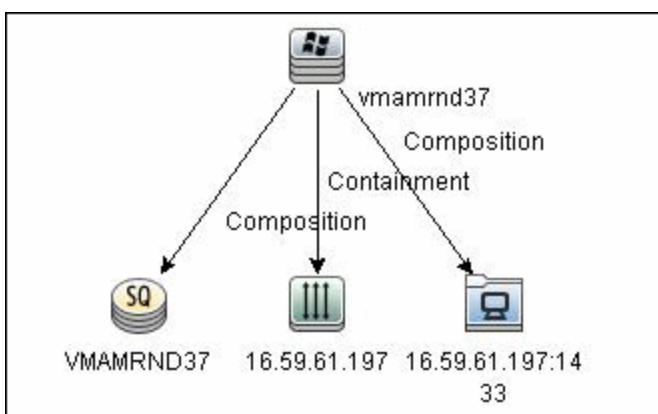
次の図は、ホスト資格情報によるデータベース接続 ディスカバリのトポロジとサンプル出力を示します。

注: 検出された CIT の一覧については、「[DB Connections by Shell ジョブ](#)」(403ページ), 「[DB Connections by WMI ジョブ](#)」(407ページ)を参照してください。

### Oracle



### Microsoft SQL



## ホスト資格情報によるデータベース接続の検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、次のプロトコルを使用します。

- WMI プロトコル
- NTCMD プロトコル
- SSH プロトコル
- Telnet プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ホスト資格情報の検出

- a. **Range IPs by ICMP** ジョブを実行します。
- b. **Host Connection by Shell** ジョブを実行します。
- c. **Host Connection by WMI** ジョブを実行します。
- d. **DB Connections by Shell** ジョブを実行します。詳細については、[「DB Connections by Shell ジョブ」\(403ページ\)](#)を参照してください。
- e. **DB Connections by WMI** ジョブを実行します。詳細については、[「DB Connections by WMI ジョブ」\(407ページ\)](#)を参照してください。

## DB Connections by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### ディスカバリ・メカニズム

このディスカバリ・ジョブは、シェル・クライアント(NTCMD / SSH / Telnet)を使用して、ホスト上にある構成済みデータベースの特定を試みます。接続すると、実行中のプロセスと各プロセスに関連するサーバ・ポー

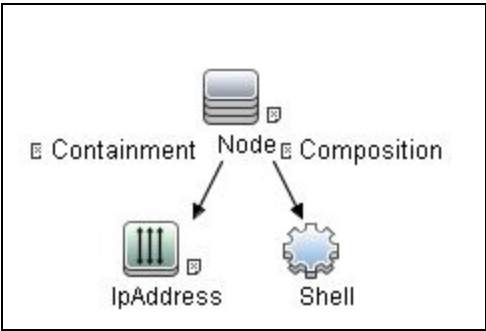
トの一覧を作成します。Microsoft Windows オペレーティング・システムの場合、このジョブはインストール済みの Windows サービスの一覧をリストに追加します。

続いて、Database CI を作成するため、このプロセス/ サービス一覧に対して既知のデータベース署名を検索します。

プロセスへのポートのマッピングには、使用中のオペレーティング・システムに応じて、特定の権限が必要になることがあります。必要な権限を使用できない場合、このジョブは、利用可能な情報を使用して Database CI の作成を試みます。ただし、詳細情報(データベース・ポートなど)がない可能性があります。そのような場合は、必要な権限を使用して新しい資格情報を入力した後で、ジョブの再実行が必要になることがあります。資格情報の追加の詳細については、『HP Universal CMDB Developer Reference Guide』の「Credentials Data Methods」を参照してください。

上の情報を使用してデータベースを特定した後で、このジョブはレジストリ・キー(Microsoft Windows の場合のみ)から、または既知の構成ファイルを解析して、構成済みの(ただし実行していない)インスタンスに関する追加情報の取得を試みます。

トリガ・クエリ



アダプタ

このジョブでは、**Database Connections by Shell** アダプタを使用します。

- 入力クエリ: なし。
- CI 属性の条件 :

CI	属性値
Shell	NOT Reference to the credentials dictionary entry null
IpAddress	NOT IP Probe Name null

- アダプタ・パラメータ



パラメータ	標準設定値	説明
discover_mssql	true	DFM は、Microsoft SQL データベース・サーバを検出します。
discover_oracle	true	DFM は、Oracle Database サーバを検出します。
filterByDiscoveredProcesses	false	このパラメータは常に <b>false</b> に設定します。このスクリプトは定義済みのプロセス・ディスカバリをいくつかのプラットフォームで使用しており、データベース・プロセスがフィルタに含まれないからです。ただし、このジョブは Process CI を作成しないため、このパラメータを <b>false</b> に設定してもマイナスの影響はありません。
use_lsof	false	Solaris プラットフォームと AIX プラットフォームでのプロセスからポートへのマッピングにはルート権限が必要なため、これらのプラットフォームで LSOF プログラムを使用できる場合は、このフラグを <b>true</b> に設定します。LSOF の使用にはルート権限は必要ありません。
use_sudo	false	一部の UNIX プラットフォームでは、プロセスからポートへのマッピングに上位権限が必要なため、 <b>sudo</b> が <b>netstat</b> , <b>ps</b> , <b>pfiles</b> , <b>kdb</b> , または <b>lsof</b> に対して構成されている場合は、このフラグを <b>true</b> に設定します。

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node

- Oracle
- SQL Server
- Unix
- Windows

**注:** トポロジを表示するには、[「トポロジ」\(402ページ\)](#)を参照してください。

## DB Connections by WMI ジョブ

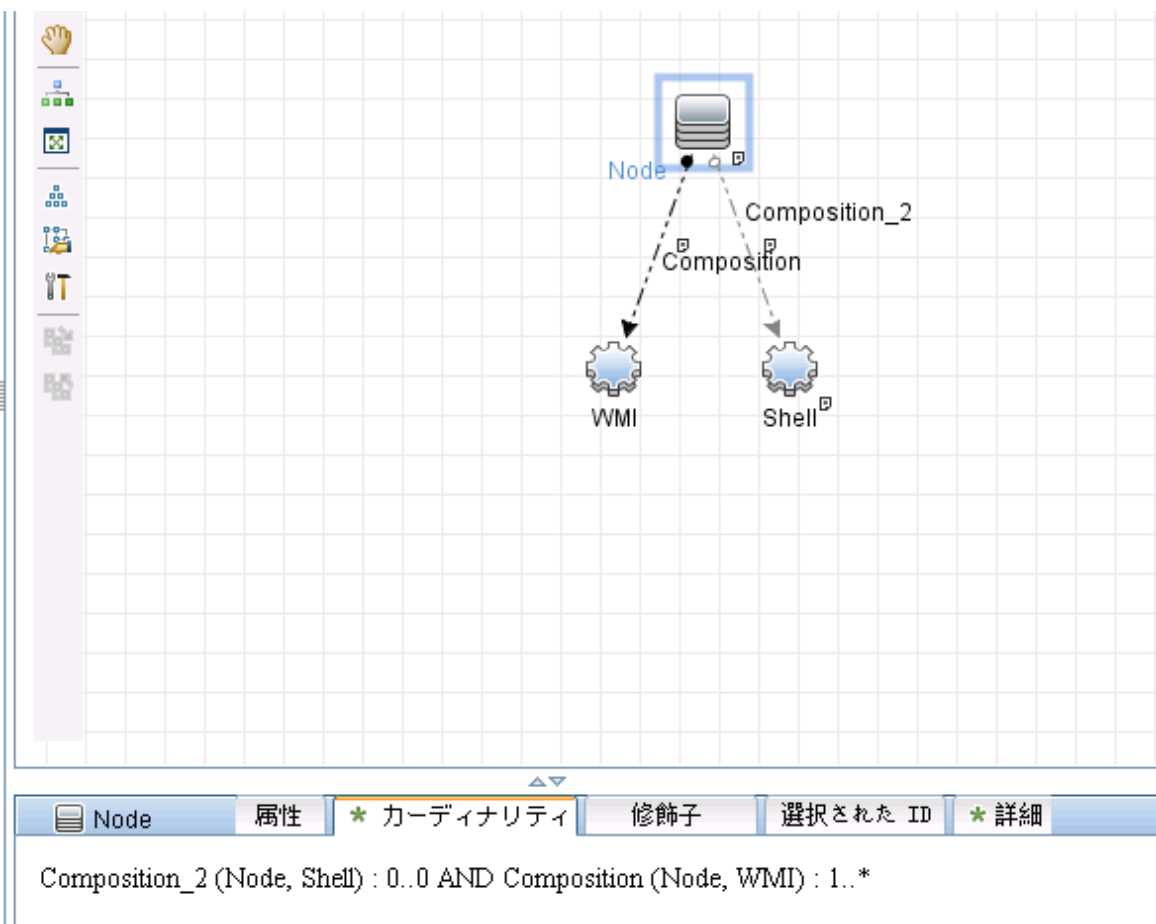
本項ではこのジョブに関する詳細を説明します。

### ディスカバリ・メカニズム

**DB Connections by Shell** ジョブと同じく、このジョブもプロセスとサービスの一覧作成を試み、それらを解析してデータベース署名を確認します。

エージェントには **netstat** などのコマンドの出力に対するアクセス権がないため、このジョブでは、データベース・サーバのリスナ・ポートが常に特定されるわけではないという制限があります。データベースのポート情報 (Microsoft SQL Server など) は Windows レジストリで利用でき、WMI を介して接続するときにこのジョブがクエリを実行してその情報を照会します。

### トリガ・クエリ



## アダプタ

このジョブでは、**Database Connections by Agent** アダプタを使用します。

- 入力クエリ:なし
- アダプタ・パラメータ:

パラメータ	標準設定値	説明
discover_mssql	true	DFM は、Microsoft SQL データベース・サーバを検出します。
discover_oracle	true	DFM は、Oracle Database サーバを検出します。

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node
- Oracle
- SQL Server
- Windows

注: トポロジを表示するには、[「トポロジ」\(402ページ\)](#)を参照してください。

## トラブルシューティングおよび制限事項

本項では、ホスト資格情報によるデータベース接続 ディスカバリのトラブルシューティングと制限事項について説明します。

- **DB Connections by WMI ディスカバリ**: パフォーマンス向上の目的で, DB Connections by WMI ジョブのトリガ・クエリは標準設定で無効になっているため, このジョブを実行するサーバは手動で選択します。

## 第23章: IBM DB2 Database ディスカバリ

### 本章の内容

サポートされるバージョン .....	411
IBM DB2トポロジ .....	411
DB2フルトポロジの検出方法 .....	412
Databases TCP Ports ジョブ .....	416
アダプタ情報 .....	419
DB2 Topology by SQL ジョブ .....	420
DB2 Universal Database Connection by SQL ジョブ .....	423
DB2 Topology by SQL アダプタ .....	426
DB2 Connection by SQL アダプタ .....	431
アプリケーション署名とプラグイン .....	434
アプリケーション署名 .....	434
プラグイン .....	434
PortNumberToPortName 構成 .....	435
トラブルシューティングおよび制限事項 .....	435
トラブルシューティング .....	435
制限事項 .....	435

## サポートされるバージョン

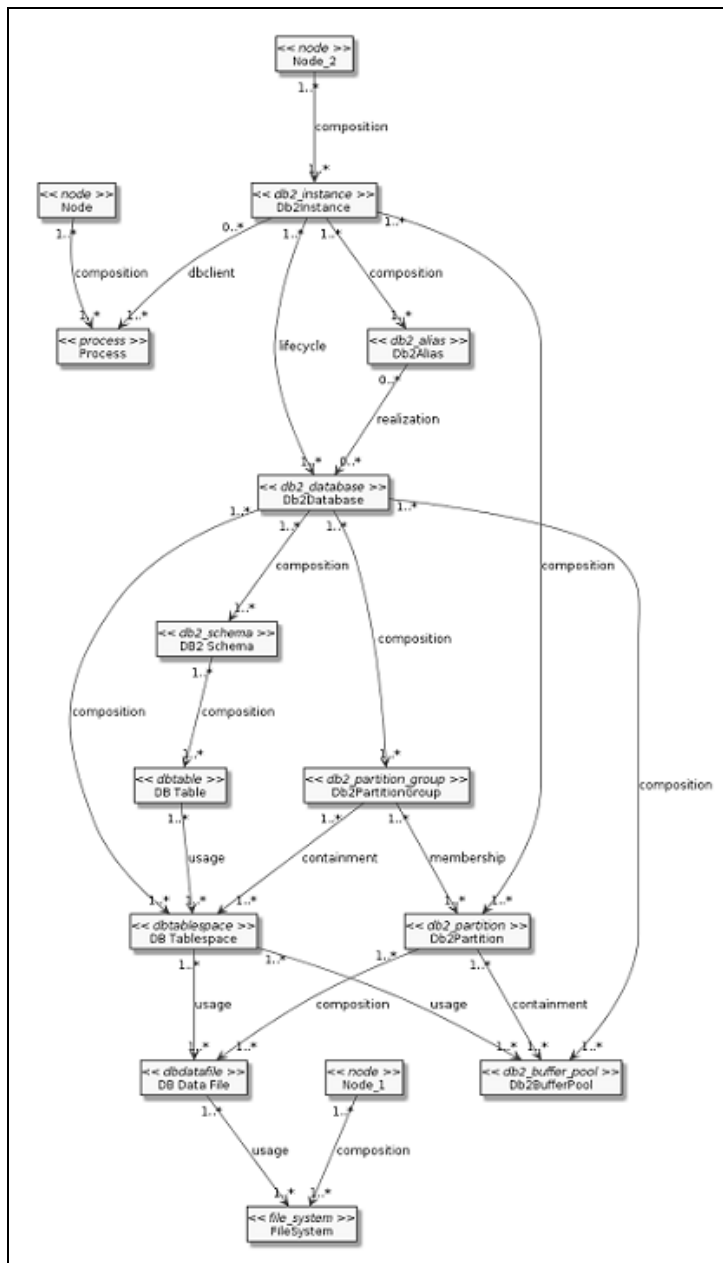
このディスカバリは、次のバージョンをサポートしています。

IBM DB2 Universal Database(UDB)バージョン 9.1, 9.5, 9.7, 9.8, 10.1(Linux, Unix, Windows プラットフォーム向け)。

## IBM DB2 トポロジ

次の図は、IBM DB2 Serverビューのトポロジを表しています。ここでは、IBM DB2 Server とそのリソースがインストールされているホスト(Node\_2)、サーバと通信するプロセス(DB Client リンクで接続)、DB Data File としてモデリングされたテーブルスペース・コンテナが示されています。

注: 検出された CIT の一覧については、[「DB2 Universal Database Connection by SQL ジョブ」](#)(423 ページ)、[「DB2 Topology by SQL ジョブ」](#)(420 ページ)、[「Databases TCP Ports ジョブ」](#)(416 ページ)を参照してください。



## DB2 フル・トポロジの検出方法

このタスクでは、ネットワーク上にある IBM DB2 Server データベースとそのコンポーネントを次の手順で検出します。



## 1. 前提条件 - プロトコル資格情報の設定

- **SQL ベースのディスカバリの場合** : IBM DB2 SQL ベースのディスカバリでは、汎用 DB プロトコル (SQL)を使用します。

プロトコル資格情報を設定する際は次の手順を実行します。

- [データベースタイプ]ボックスで **db2** を選択します。
- IBM DB2 Server で使用するユーザ名、パスワード、ポートを確認します。

汎用 DB プロトコル(SQL)の構成の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

- **シェルベースのディスカバリの場合** :
  - 適切なシェル・プロトコルを構成します。詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。
  - IBM DB2 Database で使用される可能性のあるすべてのポートは、**portNumberToPortName.xml** ファイルに追加する必要があります。**portName="db2"** になっていることを確認してください。

## 2. 前提条件 - その他

- IBM DB2 ディスカバリを実行するには、次のファイルを **<db2 のホーム・ディレクトリ>\IBM\SQLLIB\java**(Windows の場合)または **/opt/ibm/db2/<バージョン>/java**(UNIX 相当のシステムの場合)のディレクトリから Data Flow Probe マシンにコピーします。

- **db2java.zip**
- **db2jcc.jar**
- **db2jcc\_license\_cisuz.jar**
- **db2jcc\_license.jar**

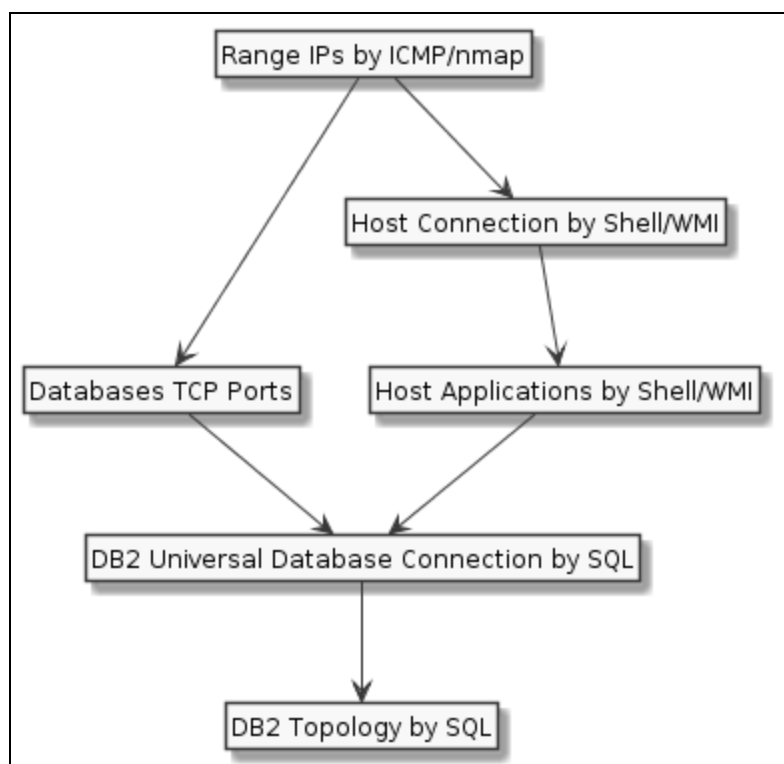
Data Flow Probe マシンで、ファイルを次のフォルダに配置します。

**<hp>\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\db\db2**。Data Flow Probe を再起動します。

### 3. ディスカバリの実行

このディスカバリの実行に必要なジョブは、DB2 の宛先でシェル・アクセスが可能かどうかによって決まります。シェル・アクセスが可能な場合は、下の項にある「[DB2 の宛先でシェル・アクセスが可能な場合の DB2 ディスカバリ](#)」(414 ページ)の手順に従ってディスカバリを実行します。DB2 の宛先でシェル・アクセスが不可能な場合は、下の項にある「[DB2 の宛先でシェル・アクセスが不可能な場合の DB2 ディスカバリ](#)」(415 ページ)の手順に従ってディスカバリを実行します。

次の図は、DB2 フルトポロジを検出するための 2 つの方法を示しています。



#### DB2 の宛先でシェル・アクセスが可能な場合の DB2 ディスカバリ

**注:** DB2 の宛先でシェル・アクセスが可能な場合、シャロー・モードまたはディープ・モードで DB2 ディスカバリを実行できます。基本トポロジを検出するシャロー・ディスカバリを実行するには (Db2Instance, Db2Database, Db2Alias の CIT), 次の手順 a~c を実行します。ディープ・ディスカバリを実行するには、次のすべての手順を実行します。

DB2 ディスカバリを実行するには、次の手順を実行します。

- a. [Universal Discovery] ウィンドウで **Range IPs by ICMP** ジョブを実行します。このジョブは、到達可能なすべての IP を検出します。

- b. **Host Connection By Shell** ジョブを実行します。DB2 サーバへのシェル・プロトコルにより、ホストの接続を検出します。
- c. **Host Applications by Shell** ジョブを実行します。DB2 のインスタンス、データベース、エイリアスを検出します。
- d. さらにディープ・ディスカバリを実行するには、次のジョブをリストされている順に実行します。
  - i. **DB2 Universal Database Connection by SQL**。このジョブは、**Db2Databases** と **Db2Aliases** の **credentials\_id** 属性を検出します。検出された **credentials\_ids** は **DB2 Topology By SQL** ジョブで使用されます(検出された CIT の一覧については、[「検出された CIT」\(434ページ\)](#)を参照してください)。

**注:** このジョブは、次のいずれかの条件を満たす場合にトリガされます。

- **Node** に、Realization リンクを使って **Db2Databases(Host Applications by Shell** ジョブで検出)とリンクされていない **Db2Instance** と **Db2Aliases** がある。
- **Db2Database** に、**DB2 IpServiceEndpoint(Host Applications by Shell** ジョブで検出)がある。
- **Node** が **DB2 IpServiceEndpoint(Host Applications by Shell** ジョブで検出)とリンクされていない。

**DB2 IpServiceEndpoint** は、**service\_names** 属性が **db2** または **ip\_service\_name** 属性が **db2** となっているエンドポイントです。

- ii. **DB2 Topology By SQL**。このジョブは、DB2 のフル・トポロジ([「検出された CIT」\(430ページ\)](#))に一覧表示されている CIT を含む)を検出します。

**注:** このジョブは、次のいずれかの条件を満たす場合にトリガされます。

- **credentials\_id** 属性を伴う **Db2Alias** がレポートされる。
- **credentials\_id** 属性を伴う **Db2Database** がレポートされる。

## DB2 の宛先でシェル・アクセスが可能な場合の DB2 ディスカバリ

DB2 ディスカバリを実行するには、次の手順を実行します。

- a. [Universal Discovery] ウィンドウで **Range IPs by ICMP** ジョブを実行します。このジョブは、到達可能なすべての IP を検出します。
- b. **Databases TCP Ports** ジョブを実行します。このジョブは、DB2 **IpServiceEndpoint** を検出します。これは、**DB2 Universal Database Connection by SQL** ジョブのトリガです。
- c. 次のジョブをリストされている順に実行します。
  - **DB2 Universal Database Connection by SQL**。このジョブは、**Db2Databases** と **Db2Aliases** の **credentials\_ids** を検出します。検出された **credentials\_ids** は **DB2 Topology By SQL** ジョブで使用されます(検出された CIT の一覧については、[「検出された CIT」\(434 ページ\)](#)を参照してください)。

注: このジョブは、**Node** が **DB2 IpServiceEndpoint**(**Databases TCP Ports** ジョブで検出)にリンクされている場合にトリガされます。DB2 **IpServiceEndpoint** は、**service\_names** 属性が **db2** または **ip\_service\_name** 属性が **db2** となっているエンドポイントです。

- **DB2 Topology By SQL**。このジョブは、DB2 のフル・トポロジ([「検出された CIT」\(430 ページ\)](#))に一覧表示されている CIT を含む)を検出します。

注: このジョブは、次のいずれかの条件を満たす場合にトリガされます。

- **credentials\_id** 属性を伴う **Db2Alias** がレポートされる。
- **credentials\_id** 属性を伴う **Db2Database** がレポートされる。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Databases TCP Ports ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、**TCP Ports Discovery** アダプタを使用します。

## トリガ・クエリ



## ノード条件

ノード名	条件
IpAddress	NOT IP Probe Name null

## ジョブ・パラメータ

名前	標準設定値	説明
checkIfIPsReachable	true	このフラグは、ホストのポートが利用可能かどうかの確認をジョブが開始する前に、検出された IP に到達可能であることを確認する必要があるかどうかを示します。
checkOnlyKnownPorts	true	このフラグは、既知のポートのみをこのジョブで検出するかどうかを示します。このフラグによって、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータがキャンセルされることはありません。このフラグの <b>false</b> への設定は、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータ内の実際のポート範囲と合わせて使用する場合にのみ適用されます。
connectTimeout	5000	IP およびポートに接続するときのタイムアウトをミリ秒単位で示します。
nmapPath		nmap 実行可能ファイルのフルパスを示します (例 : C:\Program Files (x86)\Nmap\nmap.exe)。
pingTimeout	2000	ICMP ping のタイムアウトをミリ秒単位で示します。
ports	<b>JEE TCP Ports ジョブの場合 :</b> <ul style="list-style-type: none"> <li>weblogic,</li> <li>weblogicSSL,</li> </ul>	このパラメータには、デイスカバリの実行に使用される TCP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、また

名前	標準設定値	説明
	<p>websphere_ jmx, rmi</p> <p><b>Databases TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>oracle, db2, sybase, sql, mysql</li> </ul> <p><b>SAP TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>sap, sap_ jmx, sap_ http, sap_ https</li> </ul> <p><b>SAP TCP Ports ジョブの場合 : 標準設定値なし</b></p>	<p>は * の値が含まれている場合、ディスカバリはすべての既知の TCP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、<b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。</p>
scanUDP	false	<p>このフラグは、UDP ポートをスキャンするかどうかを示します。</p> <p><b>注:</b> UDP スキャンは、<b>useNMap=true</b> に設定されている場合にのみサポートされます(下記を参照)。</p>
UDPports		<p>このパラメータには、ディスカバリの実行に使用される UDP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など)が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の UDP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、<b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。</p>
useNMap	<p><b>Databases TCP Ports ジョブおよび JEE TCP Ports ジョブの場合 :</b></p> <p>false</p>	<p>このフラグは、ポートのスキャン中に nmap を使用するかどうかを示します。</p> <p><b>注:</b> nmapPath にパスが指定されないと(上記を</p>

名前	標準設定値	説明
	<b>SAP TCP Ports</b> ジョブおよび TCP Ports ジョブの場 合 : true	参照), システム・パスからの nmap が使用されま す。

**注:** `ports` パラメータまたは `UDPports` パラメータ内でポート名が割り当てられており,  
`portNumberToPortName.xml` 構成ファイル内で検出可能 (`isDiscovered=1`)としてマークされてい  
るポートのみが検出対象になります。

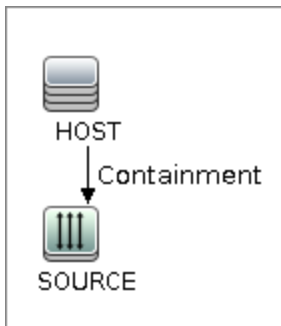
## アダプタ情報

このアダプタでは、TCP ポートを検出します。

### 入力 CIT

IpAddress

### 入力クエリ



### トリガ CI データ

名前	値
<code>ip_address</code>	<code>\${SOURCE.name}</code>
<code>ip_domain</code>	<code>\${SOURCE.routing_domain}</code>

## 使用スクリプト

- TcpPortScanner.py
- nmap.py

## グローバル構成ファイル

portNumberToPortName.xml

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node

## DB2 Topology by SQL ジョブ

このジョブは、DB2 データベース内の物理要素を検出します。

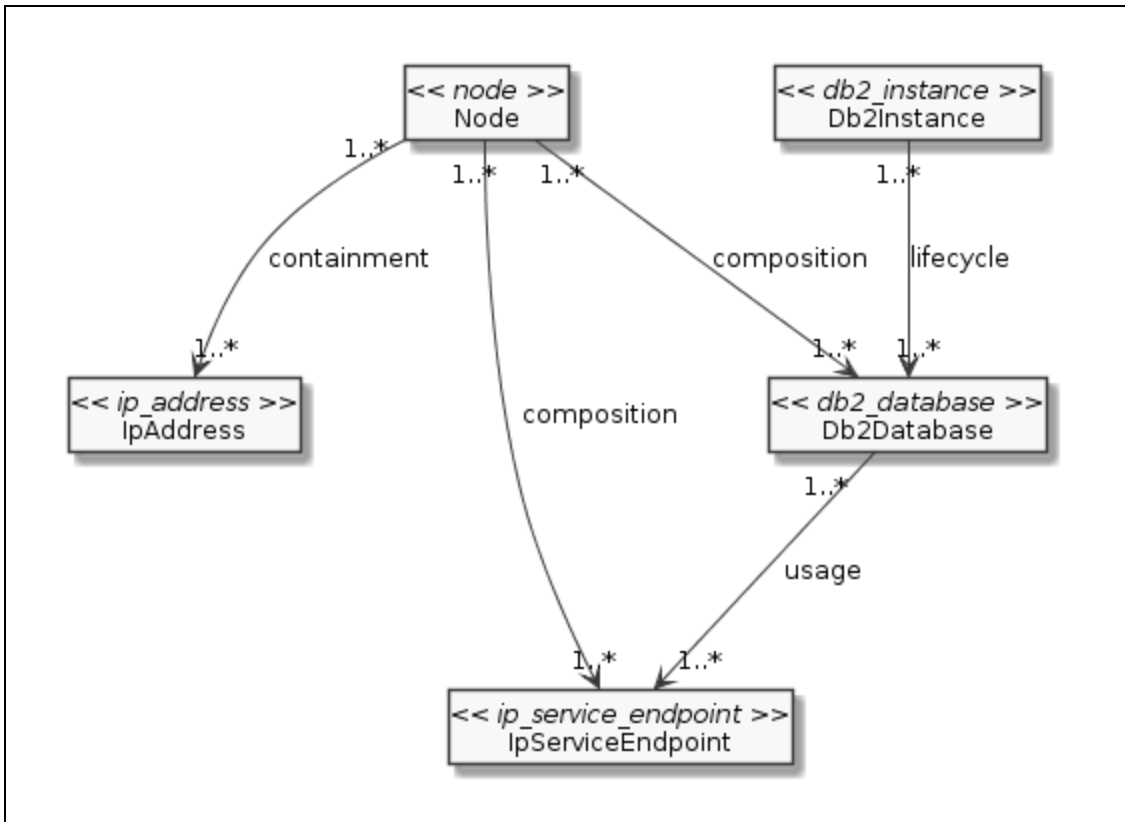
## アダプタ

ID : SQL\_APP\_Dis\_Db2

## トリガ TQL

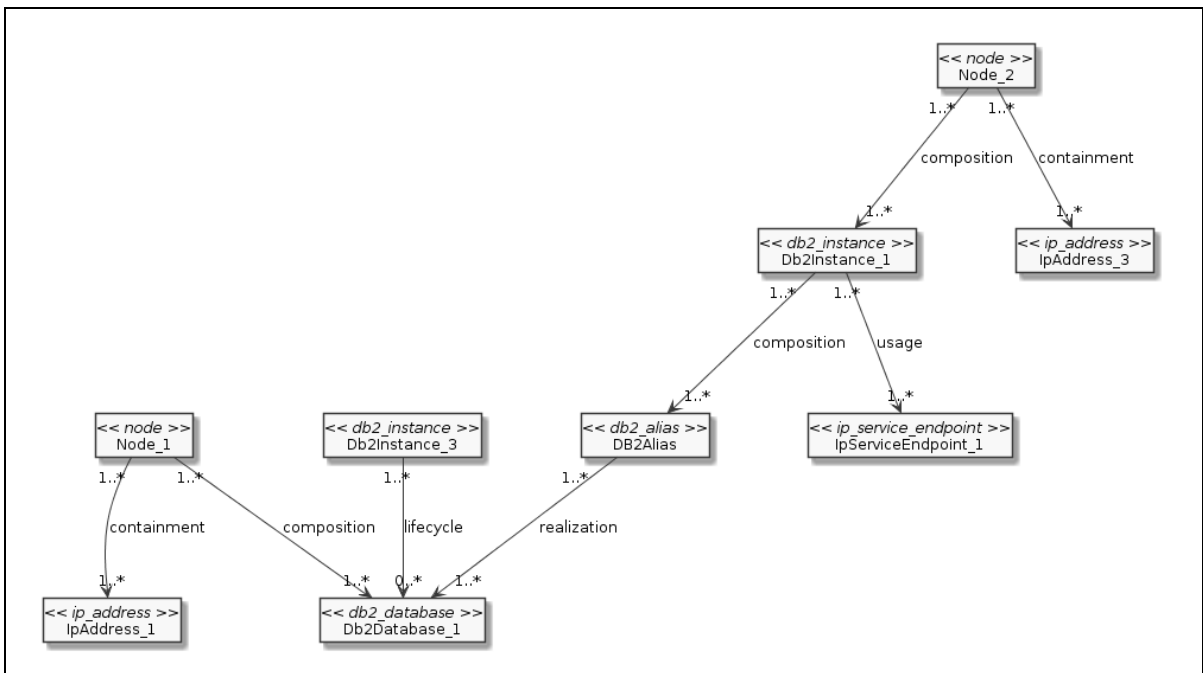
db2withuser.xml





上の図で、**Db2Database** は資格情報の辞書エントリ(**credentials\_id**)をレポートしました。

#### db2\_alias\_with\_user.xml



上の図は次を示しています。

- **Db2Database\_1** は資格情報の辞書エントリ(**credentials\_id**)をレポートしていません。
- **Db2Alias** は資格情報の辞書エントリ(**credentials\_id**)をレポートしました。

## 前提条件

- 資格情報を設定します。

このジョブは、DB2 データベース用に定義された SQL 資格情報を使用します。

## ディスカバリ・フロー

接続が確立されると、次の順序でディスカバリが実行されます。

1. このジョブにより、DB2 のメジャー・バージョンとマイナー・バージョン(`TABLE(sysproc.env_get_inst_info())` の `SERVICE_LEVEL` フィールド)が取得されます。
2. このジョブにより、適切な DB2 バージョンのディスカバラがインスタンス化されます(指定したバージョンのディスカバラを使用できない場合は 9x のディスカバラが使用されます)。
3. このジョブにより、DB2 の完全なバージョン情報(`TABLE(sysproc.env_get_inst_info())` の `RELEASE_NUM`, `SERVICE_LEVEL`, `BLD_LEVEL`, `PTF`, `FIXPACK_NUM` フィールド)が取得され、完全なバージョンが **DB2** の `application_version` 属性にレポートされます。
4. このジョブにより、**DB Data File**(`SYSIBMADM.CONTAINER_UTILIZATION`)が取得されます。**DB Data File**が Windows ホストに配置されている場合は、**FileSystem** もモデリングされます(パスのディスク名は **DB Data File**のパスから取得され、**FileSystem** の `mountpoint` 属性に割り当てられます)。
5. このジョブにより、開かれているセッションが取得されます(`TABLE(SNAP_GET_APPL_INFO(db_name, partition_number))`)。
  - クライアントのアドレスとポートは、`APPL_ID` フィールドから解析されます。
  - `APPL_NAME` はプロセス名として使用されます。
  - クライアント `APPL_NAME=db2jcc_application` または `APPL_NAME=db2jccThread` を使用したセッションは、標準設定のクライアント名とみなされ、レポートされません。
6. このジョブにより、次の CIT が取得されます。

- **Db2 Schemas**(SYSCAT.SCHEMATA)
- **Db2PartitionGroups**(SYSCAT.DBPARTITIONGROUPS)
- **Db2Partitions**(TABLE(DB\_PARTITIONS()))
- **Db2Partition** から **Db2PartitionGroup** への関係(SYSCAT.DBPARTITIONGROUPDEF)
- **Db2BufferPools**(SYSCAT.BUFFERPOOLS)。データはテーブル SYSCAT.BUFFERPOOLDBPARTITIONS からも取得され、正しくカスタマイズされた **Db2BufferPools** にレポートされます。
- **DB Table**(SYSCAT.TABLES)。Table (untyped) タイプと Typed table タイプのみが検出されます。

注: トポロジを表示するには、[「IBM DB2 トポロジ」\(411 ページ\)](#)を参照してください。

## DB2 Universal Database Connection by SQL ジョブ

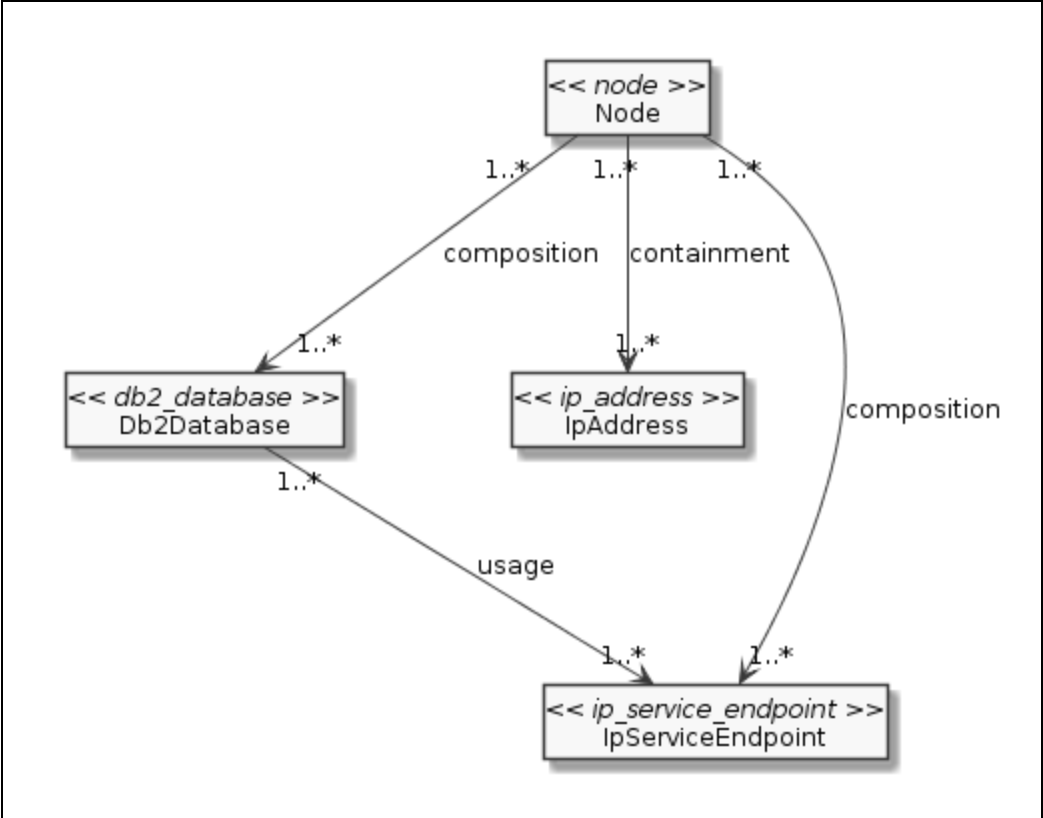
このジョブは、SQL プロトコルを使用して DB2 データベースを検出します。

### アダプタ

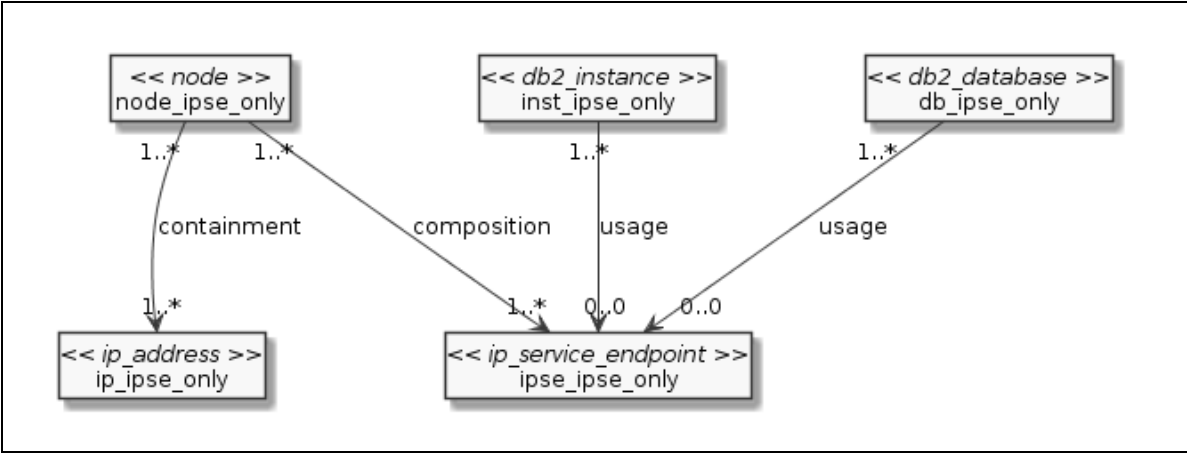
ID : SQL\_NET\_Dis\_Connection\_DB2

トリガ TQL

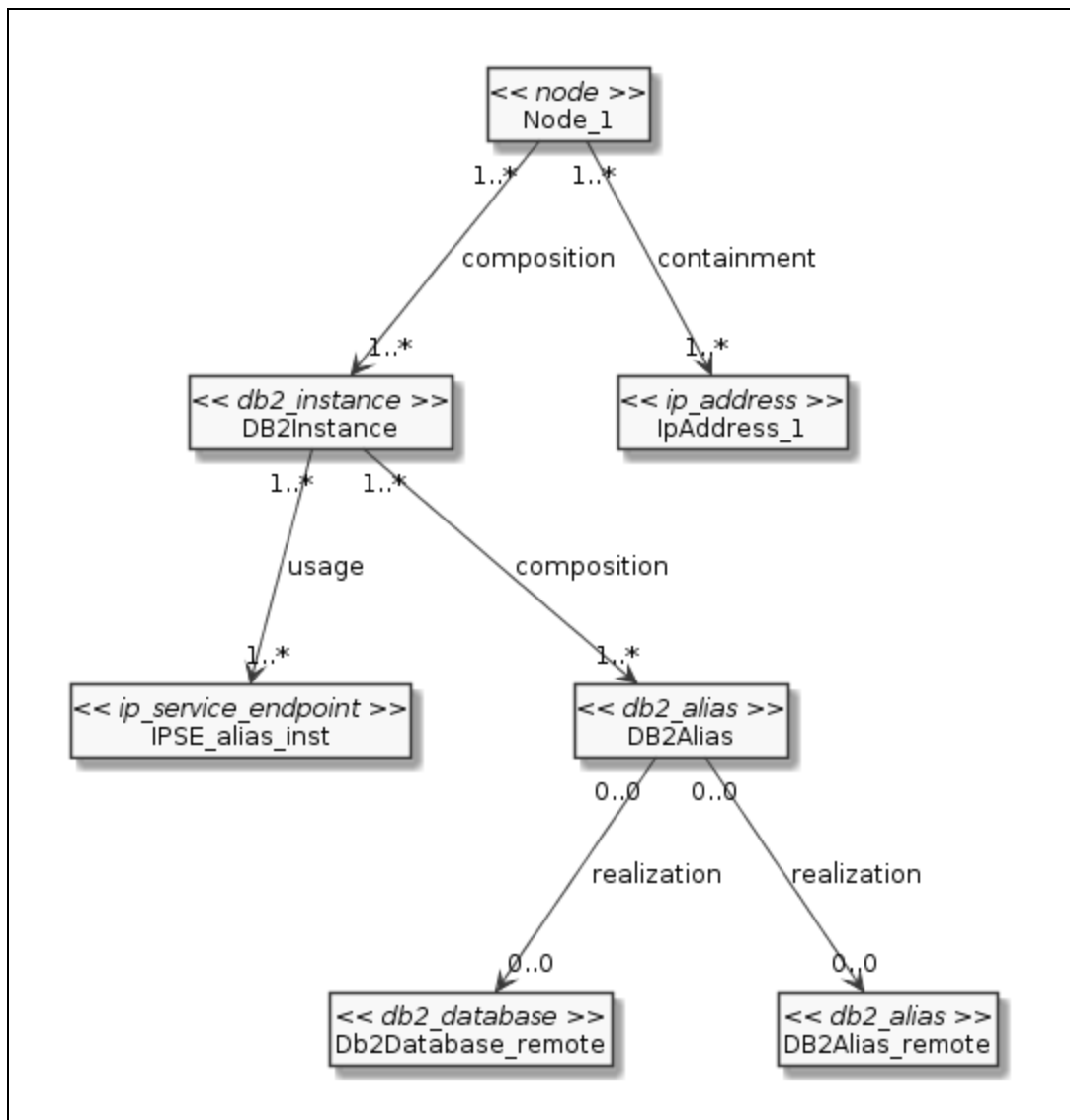
db2\_db\_port\_sa.xml



db2\_ipse\_only.xml



## db2\_alias\_no\_realizaiton.xml



## パラメータ

なし

## 前提条件

- 資格情報を設定します。

このジョブは、DB2 データベース用に定義された SQL 資格情報を使用します。

**注:** DB2 資格情報ではポートとデータベースの名前は省略可能です。これらが設定されない場合は、これらは接続中に使用される候補とみなされます。

## デイスカバリ・フロー

このジョブは、タイプ **db2** の利用可能なすべての汎用 DB プロトコルの資格情報を反復します。ポートまたはデータベース名の情報が含まれない資格情報のエントリは適用可能とみなされます。資格情報のエントリにポートまたはデータベース名情報が含まれる場合、トリガされる CI データのポートおよびデータベース名 (**port** および **db\_name**) とさらに比較されます。

このジョブを実行すると、結果として **Db2Database** または **Db2Alias**(トリガに応じて変わる)が、適切な **credentials\_id** フィールドとともにモデリングされます。

## DB2 Topology by SQL アダプタ

本項ではこのアダプタに関する詳細を説明します。

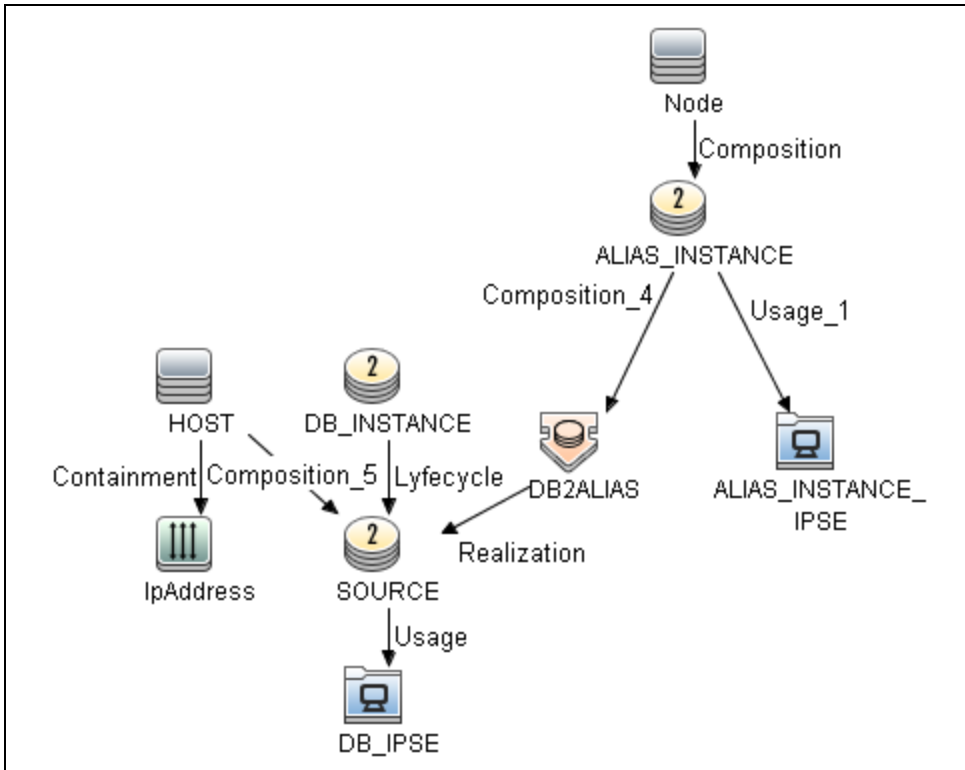
### ID

SQL\_APP\_Dis\_Db2

### 入力 CIT

Db2Database

## 入力 TQL



## トリガ CI データ

名前	値	説明
alias_inst_ip_address	\${ALIAS_INSTANCE.application_ip:!NA!}	対応するエイリアスを持つ DB2 インスタンスの IP アドレス(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Alias</b> の場合)
alias_inst_ipse_ip_address	\${ALIAS_INSTANCE_IPSE.bound_to_ip_address:!NA!}	SQL 接続を確立するために使用される <b>IpServiceEndpoint</b> の IP アドレス(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Alias</b> の場合)
alias_inst_ipse_port	\${ALIAS_INSTANCE_IPSE.network_port_number:!NA!}	SQL 接続を確立するために使用される <b>IpServiceEndpoint</b> のポート(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Alias</b> の場合)
alias_inst_port	\${ALIAS_INSTANCE.application_port:!NA!}	対応するエイリアスを持つ DB2 インスタンスが使用するポート番号(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Alias</b> の場合)

名前	値	説明
alias_cred_id	\${DB2ALIAS.credentials_id:!NA!}	SQL 接続を確立するために使用される <b>credentials_id</b> (トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Alias</b> の場合)
alias_name	\${DB2ALIAS.name:!NA!}	SQL 接続を確立するため、DB 名として使用されるエイリアス名(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Alias</b> の場合)
db_instance_id	\${DB_INSTANCE.root_id:!NA!}	<b>Db2Instance</b> CMDB ID(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
db_ipse_id	\${DB_IPSE.root_id:!NA!}	<b>IpServiceEndpoint</b> CMDB ID(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
db_ipse_ip_address	\${DB_IPSE.bound_to_ip_address:!NA!}	SQL 接続を確立するために使用される <b>IpServiceEndpoint</b> の IP アドレス(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
db_ipse_port	\${DB_IPSE.network_port_number:!NA!}	SQL 接続を確立するために使用される <b>IpServiceEndpoint</b> のポート(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
host_id	\${HOST.root_id}	ターゲットの DB2 データベースを含むノード CMDB ID
db_node_ip_address	\${IpAddress.ip_address:!NA!}	接続の確立に使用される <b>Db2Database</b> のノード IP アドレス(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
db_cred_id	\${SOURCE.credentials_id:!NA!}	SQL 接続を確立するために使用される <b>Db2Database</b> の <b>credentials_id</b> (トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
db_id	\${SOURCE.root_id}	CI を復元するために使用される <b>Db2Database</b> CMDB ID(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
db_name	\${SOURCE.name}	SQL 接続を確立するために使用される <b>Db2Database</b> の名前(トリガが、レポートされた <b>credentials_id</b> を伴う <b>Db2Database</b> の場合)
db_port	\${SOURCE.application_port:!NA!}	接続を確立するために使用される <b>Db2Database</b> のポート(トリガが、レポートされ



名前	値	説明
		た <b>credentials_id</b> を伴う <b>Db2Database</b> の場合 )

使用スクリプト

- `iteratortools.py`
- `db2_pyarg_validator.py`
- `db2_flow.py`
- `db2_host.py`
- `db.py`
- `command.py`
- `db_platform.py`
- `process.py`
- `db_builder.py`
- `db2_discoverer.py`
- `db2_sql_base_discoverer.py`
- `db2_base_parser.py`
- `db2_model.py`
- `db2_sql_v9x_discoverer.py`
- `db2_sql_discoverer.py`
- `db2_topology.py`
- `db2_topology_by_sql.py`

## 検出された CIT

- Composition
- Containment
- DB Data File
- DB Table
- DB Tablespace
- DB2 Schema
- Db2Alias
- Db2BufferPool
- Db2Database
- DB2Instance
- Db2Partition
- Db2PartitionGroup
- IpAddress
- IpServiceEndpoint
- Membership
- Node
- Process
- Resource
- Usage

## パラメータ

名前	タイプ	説明
discoverTables	ブール型	このフラグは、ジョブがテーブルを検出するかどうかを示します。
discoverSystemTables	ブール型	このフラグは、ジョブがシステム・テーブルを検出するかどうかを示します。 <b>discoverTables</b> が false の場合、このフラグも false とみなされます。

## DB2 Connection by SQL アダプタ

本項ではこのアダプタに関する詳細を説明します。

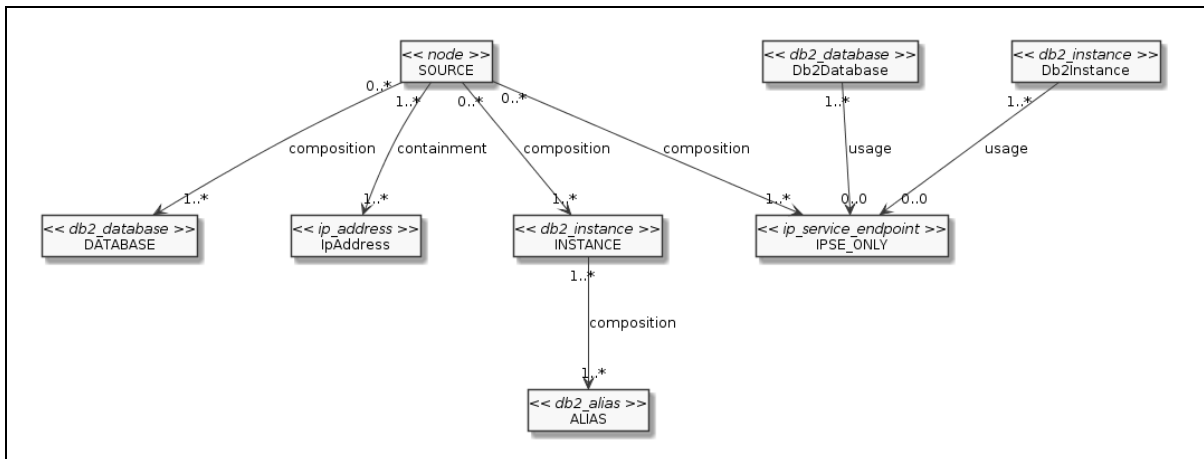
### ID

SQL\_NET\_Dis\_Connection\_DB2

### 入力 CIT

Node

### 入力 TQL



## トリガ CI データ

名前	値	説明
alias_container	\${ALIAS.root_container:!NA!}	<b>Db2Alias</b> コンテナ CMDB ID の一覧 (トリガが、送信 Realization リンクなしの <b>Db2Alias</b> の場合)
alias_id	\${ALIAS.root_id:!NA!}	<b>Db2Alias</b> CMDB ID の一覧 (トリガが、送信 Realization リンクなしの <b>Db2Alias</b> の場合)
alias_name	\${ALIAS.name:!NA!}	接続の確立に使用される <b>Db2Alias</b> 名 の一覧 (トリガが、送信 Realization リンクなしの <b>Db2Alias</b> の場合)
db_application_ip	\${DATABASE.application_ip:!NA!}	接続の確立に使用される <b>Db2Database</b> IP アドレスの一覧 (トリガが、db2 <b>IpServiceEndpoint</b> を伴う <b>Db2Database</b> の場合)
db_application_port	\${DATABASE.application_port:!NA!}	接続の確立に使用される <b>Db2Database</b> ポートの一覧 (トリガが、db2 <b>IpServiceEndpoint</b> を伴う <b>Db2Database</b> の場合)
db_id	\${DATABASE.root_id:!NA!}	CI の復元に使用される <b>Db2Database</b> CMDB ID の一覧 (トリガが、db2 <b>IpServiceEndpoint</b> を伴う <b>Db2Database</b> の場合)
db_name	\${DATABASE.name:!NA!}	接続の確立に使用される <b>Db2Database</b> 名 の一覧 (トリガが、db2 <b>IpServiceEndpoint</b> を伴う <b>Db2Database</b> の場合)
credentials_id	\${DATABASE.credentials_id:!NA!}	接続の確立に使用される <b>Db2Database</b> <b>credentials_ids</b> の一覧 (トリガが、db2 <b>IpServiceEndpoint</b> を伴う <b>Db2Database</b> の場合)
inst_application_port	\${INSTANCE.application_port:!NA!}	接続の確立に使用される <b>Db2Instance</b> ポートの一覧 (トリガが、送信 Realization リンクなしの <b>Db2Alias</b> の場合)
inst_id	\${INSTANCE.root_id:!NA!}	<b>Db2Instance</b> CMDB ID の一覧 (トリガが、送信 Realization リンクなしの <b>Db2Alias</b> の場合)
ipse_only_id	\${IPSE_ONLY.root_id:!NA!}	CI の復元に使用される <b>IpServiceEndpoint</b> CMDB ID の一覧 (トリガが、db2 <b>IpServiceEndpoint</b> の場合)

名前	値	説明
ipse_only_port	\${IPSE_ONLY.network_port_number:!NA!}	接続の確立に使用される <b>IpServiceEndpoint</b> ポートの一覧 (トリガが、 db2 <b>IpServiceEndpoint</b> の場合)
ip_address	\${IpAddress.ip_address}	現在のノード IP アドレス
host_id	\${SOURCE.root_id}	CI の復元に使用される現在のノード CMDB の ID

## 使用スクリプト

- iteratorools.py
- db2\_pyarg\_validator.py
- db2\_flow.py
- db2\_host.py
- db.py
- command.py
- db\_platform.py
- process.py
- db\_builder.py
- db2\_discoverer.py
- db2\_sql\_base\_discoverer.py
- db2\_base\_parser.py
- db2\_model.py
- db2\_sql\_v9x\_discoverer.py
- db2\_sql\_discoverer.py
- db2\_topology.py

- db2\_connection\_by\_sql.py

## 検出された CIT

- Composition
- Db2Alias
- Db2Database
- Db2Instance
- IpServiceEndpoint
- Node
- Usage

## パラメータ

なし

# アプリケーション署名とプラグイン

## アプリケーション署名

Db2 パッケージには次の 2 つの署名があります。

- **IBM DB2 on Unix : db2sysc** プロセスの存在によって Db2Instance を検出します。  
**PortNumberToPortName** の構成に従ってポートが認識されると、そのポートはサービス名 db2 としてレポートされます。標準設定では、マッピングを指定するポートの構成ファイルにポート 50000 とポート 6789 が宣言されます。
- **IBM DB2 on Windows : db2syscs.exe** プロセスの存在によって Db2Instance を検出します。  
**PortNumberToPortName** の構成に従ってポートが認識されると、そのポートはサービス名 db2 としてレポートされます。標準設定では、マッピングを指定するポートの構成ファイルは 50000 のポートを宣言します。

## プラグイン

次のプラグインによって、追加のトポロジがレポートされます。

- **db2\_instances\_on\_windows** : Windows ホストで **Db2Databases**, **Db2Instances**, **Db2Aliases** を検出します。
- **db2\_instances\_on\_unix** : Unix ホストで **Db2Databases**, **Db2Instances**, **Db2Aliases** を検出します。

## PortNumberToPortName 構成

**portNumberToPortName.xml** ファイルには、対象となる検出範囲内にあり、IBM DB2 Database で使用される、すべての候補ポートが含まれています。これらのポートには **"db2"** という名前が付きます。

## トラブルシューティングおよび制限事項

### トラブルシューティング

**問題** : ターゲットの DB2 インスタンスのポートが **portNumberToPortName.xml** ファイルに追加されていない場合 (**db2** として認識されていない場合) に Host Applications by Shell ジョブを実行すると、"複数一致" を通知する警告が UCMDB UI に表示されることがあります(つまり、ターゲット CI がレポートされません)。

**解決策** : ターゲット・ポートを db2 ポート・エントリとして **portNumberToPortName.xml** ファイルに追加します。

### 制限事項

#### シェル・アクセスなしで DB2 ディスカバリを実行する場合の制限事項

DB2 プラットフォームでは、ネットワーク・サービス名をインスタンスのリスニング・ポートとして指定できます。このネットワーク・サービス名は、適切なマッピング・ファイル(Unix の場合は /etc/services、Windows の場合は %SystemRoot%\system32\drivers\etc\services)を使用して解決する必要のあるエイリアスです。このファイルのコンテンツを取得するにはシェル・アクセスが必要です。シェル・アクセスが行えず、SQL ベースのアクセスしか可能でない場合、サービス名の文字列をポート番号に拡張することはできません。これを行うと、接続先データベース向けに **IpServiceEndpoint** がレポートされません。この問題に対する唯一の解決策は、DB2 インスタンスの構成時に、サービス名ではなく実際のポート番号を使用することです。

## 第24章: HP NonStop ディスカバリ

### 本章の内容

概要 .....	437
サポートされるバージョン .....	437
トポロジ .....	438
HP NonStop の検出方法 .....	439
HP NonStop Topology by Shell ジョブ .....	441
HP NonStop ディスカバリ・コマンド .....	443



## 概要

1970年代中頃から、HP NonStop サーバはグローバル・ビジネスの正常で効率的な運営を円滑に進めるうえで重要な役割を担ってきました。現在、NonStop サーバはクレジット・カード、ATM(自動預金支払機)、証券取引の圧倒的多数の処理に使用されています。世界中のトップ企業がNonStop サーバに依存しており、これらには最大手 120 社中 106 社の株式および商品取引所と135社の公共通信会社が含まれています。NonStop プラットフォーム上に構築された革新的ソリューションを導入することで、利用企業は金融、電気通信、医療、小売、公共機関、製造業などの複数の業界で競争力を高めることができます。The Standish Group の調査によれば、NonStop サーバは業界同クラスのサーバの中で最低のTCO(総所有コスト)を実現します。

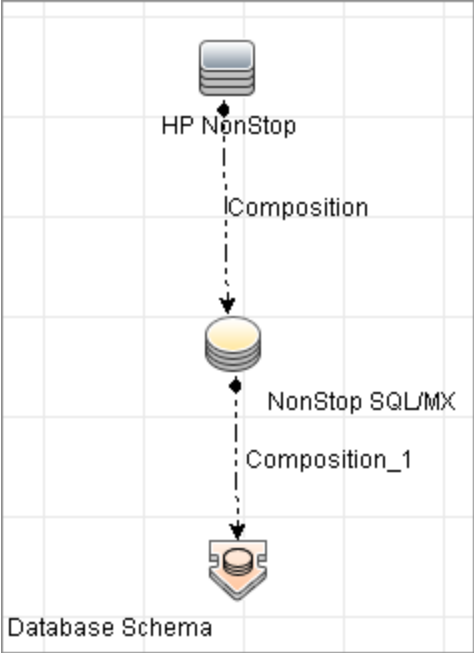
## サポートされるバージョン

このディスカバリ・ソリューションでは、次をサポートします。

- HP NonStop H06.x
- NonStop SQL/MX 2.3
- NonStop SQL/MP H01 シリーズ

**注:** このディスカバリは、HP NonStop の入手可能なすべてのバージョンで動作することが想定されています。

# トポロジ



## HP NonStop の検出方法

次の手順で、HP NonStop ディスカバリの実行方法について説明します。

### 1. 前提条件

ディスカバリを開始する前に、次のコマンドを実行するために必要なすべての権限がディスカバリ・ユーザに付与されていることを確認します。

- `gtacl -p scf info lif '$zzlan.*'`
- `gtacl -p scf info subnet '$*.*'`
- `mxci`
  - `set schema nonstop_sqlmx_<node_name>.system_schema`
  - `select cat_name, cat_uid from catsys`
  - `select schema_name, cat_uid from schemata`
- `gtacl -p sqlci`
  - `fileinfo $system.system.sqlci2, detail`
  - `select catalogname from <catalog_file_name>.catalogs`

### 2. ネットワークおよびプロトコル資格情報の設定

HP NonStop ディスカバリ・ソリューションは、SSH プロトコルに基づいています。このプロトコルを使用するためには、対応する資格情報を指定する必要があります。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 3. ディスカバリの実行

トポロジを検出するには、次の手順を実行します。

- a. **Range IPs by ICMP** ジョブまたは **Range IPs by nmap** ジョブを実行して、HP NonStop システムの IP アドレスを検出します。
- b. **Host Connection by Shell** ジョブを実行して、SSH エージェントとネットワーク・トポロジが接続

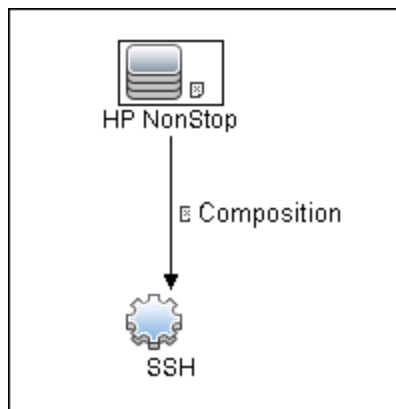
した状態で HP NonStop システムを検出します。

- c. **HP NonStop Topology by Shell** ジョブを実行し、SQL MP / MX トポロジに対してシャロー・ディスカバリを実行します。

## HP NonStop Topology by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

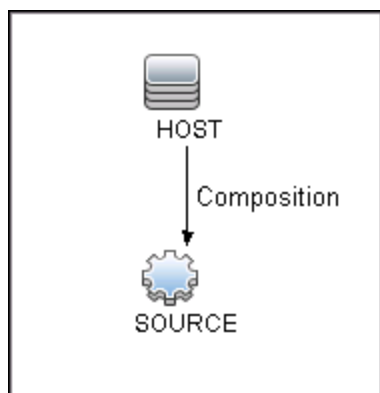
### トリガ・クエリ



### アダプタ

このジョブでは、**hp\_nonstop\_topology\_by\_shell** アダプタを使用します。

- 入力 CIT: **Shell**
- 入力クエリ:



- 使用スクリプト
  - `hpnnonstop_topology_by_shell.py`

**注:** このジョブでは, AutoDiscoveryContent パッケージで提供されるライブラリ・スクリプトも使用される場合があります。

- 作成または変更されたエンティティ:

エンティティ名	エンティティ・タイプ	エンティティの説明
hp_nonstop	CIT	HP NonStop システムを表す新しい CIT
nonstop_sql_mx	CIT	SQL / MX データベースを表す新しい CIT
HP NonStop Topology by Shell	ジョブ	新しいトポロジ・ジョブ
hp_nonstop_ topology_by_shell	アダプタ	ディスカバリ・アダプタ
Host_Connection_ By_Shell	アダプタ	Host Connection by Shell ジョブが使用するアダプタは, HP NonStop サポートが追加されたために変更
hpnonstop_ topology_by_shell.py	スクリプト	ディスカバリ Jython スクリプト
hp_nonstop_ shell.xml	TQL	トリガ TQL
TTY_Connection_ Utils	スクリプト	Host Connection by Shell ジョブで使用するメイン・スクリプトは, HP NonStop システムに対応するように変更
hp_nonstop_ networking.py	スクリプト	HP NonStop ネットワーク情報を検出する Jython スクリプト

## 検出された CIT

- Composition
- Database
- Database Schema
- HP NonStop

- NonStop SQL / MX

## HP NonStop ディスカバリ・コマンド

本項では、HP NonStop ディスカバリで使用する各コマンドについて説明します。

本項の内容

- 「コマンド」: `gtac1 -p scf info lif ';$zzlan.*';`(443ページ)
- 「コマンド」: `gtac1 -p scf info subnet ';$*. *';`(444ページ)
- 「コマンド」: `mxci`(444ページ)
- 「コマンド」: `set schema nonstop_sqlmx_measyos.system_schema;`(445ページ)
- 「コマンド」: `select cat_name, cat_uid from catsys;`(445ページ)
- 「コマンド」: `select schema_name, cat_uid from schemata;`(446ページ)
- 「コマンド」: `exit`(446ページ)
- 「コマンド」: `gtac1 -p sqlci`(446ページ)
- 「コマンド」: `fileinfo $system.system.sqlci2, detail;`(446ページ)
- 「コマンド」: `select catalogname from $QA1.SQL.catalogs;`(447ページ)

**コマンド** : `gtac1 -p scf info lif ';$zzlan.*';`

- サンプル出力

```
SCF - T9082H01 - (16JUL10) (30MAR10) - 11/08/2010 01:32:10 System \NON_STOP_
SYSTEM
(C) 1986 Tandem (C) 2006 Hewlett Packard Development Company, L.P.
SLSA Info LIF
Name                Associated Object    MAC Address          Type
$ZZLAN.LANA         G4SA0.0.A           01:01:01:01:01:01    Ethernet
$ZZLAN.LANB         G4SA0.0.B           02:02:02:02:02:02    Ethernet
$ZZLAN.LANC         G4SA0.0.C           03:03:03:03:03:03    Ethernet
$ZZLAN.LAND         G4SA0.0.D           04:04:04:04:04:04    Ethernet
Total Errors = 0     Total Warnings = 0
```

• モデリングされた CIT : Interface

属性	値	コメント
Name	LANA	
Interface MAC Address	01:01:01:01:01:01	
Interface Description	G4SA0.0.A	

コマンド : `gtac1 -p scf info subnet ';$*.*';`

• サンプル出力 (一部)

```
SCF - T9082H01 - (16JUL10) (30MAR10) - 11/08/2010 04:05:58 System \MEASYOS
(C) 1986 Tandem (C) 2006 Hewlett Packard Development Company, L.P.
TCPIP Info SUBNET \MEASYOS.$ZSM1.*
Name      Devicename      *IPADDRESS      TYPE      *SUBNETMASK      SuName      QIO *R
#SN01     \MEASYOS.LANC      10.10.10.10     ETHERNET   %HFFFFFFC00      ON      N
#LOOP0    127.0.0.1         LOOP-BACK %HFF000000      OFF N
TCPIP Info SUBNET \MEASYOS.$ZTC0.*
Name      Devicename      *IPADDRESS      TYPE      *SUBNETMASK      SuName      QIO *R
#SN01     \MEASYOS.LANC      10.10.10.10     ETHERNET   %HFFFFFFC00      ON      N
#LOOP0    127.0.0.1         LOOP-BACK %HFF000000      OFF N
```

• モデリングされた CIT : IP, Network

属性	値	コメント
IP Address	10.10.10.10	"ETHERNET" タイプのみが該当
IP Network Mask	%HFFFFFFC00	16 進数で示されるネットワーク・マスク
Container	LANC	この IP の接続先 インタフェース名

注: ネットワーク CIT は、このコマンドからも作成されます。

コマンド : `mxci`

• サンプル出力



Hewlett-Packard NonStop(TM) SQL/MX Conversational Interface 2.3.4  
(c) Copyright 2003, 2004-2010 Hewlett-Packard Development Company, LP.

• 取得された値

出力から、SQL / MX のバージョン値が取得されます。この場合、2.3.4 です。

**コマンド : set schema nonstop\_sqlmx\_measyos.system\_schema;**

• サンプル出力

--- SQL operation complete.

• モデリングされた CIT

なし

**コマンド : select cat\_name, cat\_uid from catsys;**

• サンプル出力

```
CAT_NAME                                CAT_UID
-----                                -
C
010101010101010101010101010101010
NONSTOP_SQLMX_MEASYOS
020202020202020202020202020202020
--- 2 row(s) selected.
```

• モデリングされた CIT - NonStop SQL / MX

属性	値	コメント
Name	NonStop SQL / MX	定数の値
Catalog UUID	010101010101010101010101010101010	
The Database instance name	NONSTOP_SQLMX_MEASYOS	

- 出力

- ・ モデリングされた CIT : Database Schema

属性	値	コメント
Name	DEFINITION_SCHEMA_VERSION_1200	スキーマID
Container	0101010101010101010	

- ・ サンプル出力

- ・ サンプル出力

- ・ サンプル出力

```
$SYSTEM.SYSTEM.SQLCI2          8 Nov 2010,  6:22
  ENSCRIBE ( VALID SQL PROGRAM )
  CATALOG $QA1.SQL
  PROGRAM CATALOG VERSION 1
  PROGRAM FORMAT VERSION 350
  TYPE U
  FORMAT 1
  CODE 100
  EXT ( 56 PAGES, 56 PAGES, MAXEXTENTS 978 )
  ODDUNSTR
  NO AUDITCOMPRESS
  OWNER -1
  SECURITY (RWEPR):NUNU
  MODIF:21 Dec 2008, 23:22, OPEN
  CREATION DATE:21 Dec 2008, 23:21
  LAST OPEN:8 Nov 2010,  6:22
  EOF 364544 (0.3% USED)
  EXTENTS ALLOCATED:4
```

- 取得された値

QA1.SQL

**コマンド : select catalogname from \$QA1.SQL.catalogs;**

- サンプル出力

```
CATALOGNAME
-----
\MEASYOS.$QA1.H03SQLMP
\MEASYOS.$QA1.SQL
\MEASYOS.$QA2.PERSNL
\MEASYOS.$SFF04.SALES
\MEASYOS.$SGT01.INVENT
\MEASYOS.$SGT01.PERSNL
\MEASYOS.$SGT02.SALES
\MEASYOS.$SGT03.INVENT
\MEASYOS.$SYSTEM.SRK
\MEASYOS.$SYSTEM.VIMAL
--- 10 row(s) selected.
```

- モデリングされた CIT : Database

属性	値	コメント
Name	NonStop SQL / MX	定数の値
Database instance name	\$QA1.H03SQLMP	

## 第25章: SAP HANA Database ディスカバリ

### 本章の内容

概要 .....	450
サポートされるバージョン .....	450
トポロジ .....	450
ディスカバリ・メカニズム .....	450
SAP HANA データベースの検出方法 .....	452
HanaDb by Shell ジョブ .....	453
HanaDb_by_Shell アダプタ .....	454
ディスカバリ・フロー .....	457

## 概要

SAP HANA(High Performance Analytic Appliance)は, SAP のデータベース・テクノロジーです。このテクノロジーは, SAP が承認したハードウェアの組み合わせであるアプライアンスおよびメモリ内データベース・ソフトウェアとして配布されます。

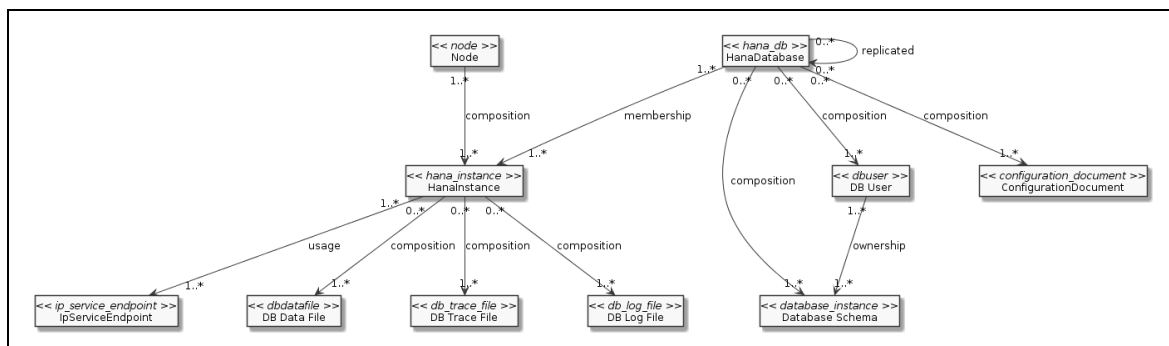
## サポートされるバージョン

このディスカバリは, UNIX 環境で実行される SAP HANA 1.0 をサポートしています。

## トポロジ

次の図に, SAP HANA Database ディスカバリのトポロジを示します。

検出された CIT のリストについては, [「検出された CIT」\(456ページ\)](#)を参照してください。



## ディスカバリ・メカニズム

### 署名

HanaDb パッケージには 1 つの署名があり, これは **SAP HanaDB** です。これにより, **HDB.sap**(HDB デーモン)から始まるプロセスがあるかどうかを検索することによって, Hana インスタンスを検出します。データベース名 およびインスタンス番号が, **hanadb\_instance\_name** 解析ルールを使用して HDB デーモン名から解析されます。

## プラグイン

**hanadb** プラグインで追加のトポロジがレポートされます。このプラグインは、HanaDatabase およびその他の Hana インスタンスと、**IpServiceEndpoints** としてモデル化された使用済みの SQL ポートを検出します。

## SAP HANA データベースの検出方法

本項では、SAP Hana データベースのトポロジの検出方法について説明します。これには次の手順が含まれます。

### 1. 前提条件 - プロトコル資格情報の設定

シェル・プロトコル資格情報の構成

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. hdbsql コマンド・ライン・ツールの実行

シェルベースの HANA のディスカバリでは、SQL クエリの実行に hdbsql コマンド・ライン・ツールが使用されます。次のパスが検索されます。

- `<installation_path>/hdbclient/hdbsql`
- `/usr/sap/hdbclient/hdbsql`
- `/usr/sap/<db_sid>/hdbclient/hdbsql`
- `/usr/sap/<db_sid>/exe/linuxx86_64/hdb/hdbsql`
- `/usr/sap/<db_sid>/SYS/global/hdbclient/hdbsql`
- `/sapmnt/<db_sid>/hdbclient/hdbsql`
- `/sapmnt/<db_sid>/global/hdbclient/hdbsql`

`<db_sid>` は HANA データベースの SID を示します。

これらのいずれかのパスでこのツールが利用可能でない場合、ツールが存在するかどうかを PATH 変数から確認されます。存在しない場合、ディスカバリ・プロセスは **NoHdbsqlException** をスローし、現在の HANA データベースの検出を停止します。

### 3. HDB ユーザ・ストアの構成

シャロー・ディスカバリまたはディープ・ディスカバリを実行するには、検出対象の HDB ユーザ・ストアを適切に構成する必要があります。現在のディスカバリ・メカニズムは `cmdb <db_sid> HDB ユーザ・ストア名` を使用します。ここで `<db_sid>` は HANA データベースの SID を示します。



**注:** hdbsql ツールには、有効な HDB ユーザ・ストア・エントリが -U オプションを使用して指定される必要があります。

#### 4. ディスカバリの実行

次のジョブをアクティブにします。

- **Range IPs by ICMP** ジョブを実行し、IP 範囲内で稼働中のマシンを検出します。

基本トポロジ(HanaDatabase, HanaInstance, IpServiceEndpoint の各 CIT)を検出するシャロー・ディスカバリを実行するには、次の 2 つのジョブを実行するだけでよく、その後の手順に進む必要はありません。

- **Host Connection by Shell** ジョブを実行し、シェル・プロトコルを介したホストから HANA サーバへの接続を検出します。
  - **Host Applications by Shell** ジョブを実行し、HANA データベース、そのインスタンス、そのインスタンスの IP サービス・エンドポイントを検出します。
- ディープ・ディスカバリを実行するには、**HanaDb by Shell** ジョブも実行し、ターゲット HANA データベースのトポロジを検出します。このジョブは、すべての HANA データベース・リソース(スキーマ、データベース・ユーザ、構成ファイル)と HANA インスタンス・リソース(データ、トレース・ファイル、ログ・ファイル)も検出します。

## HanaDb by Shell ジョブ

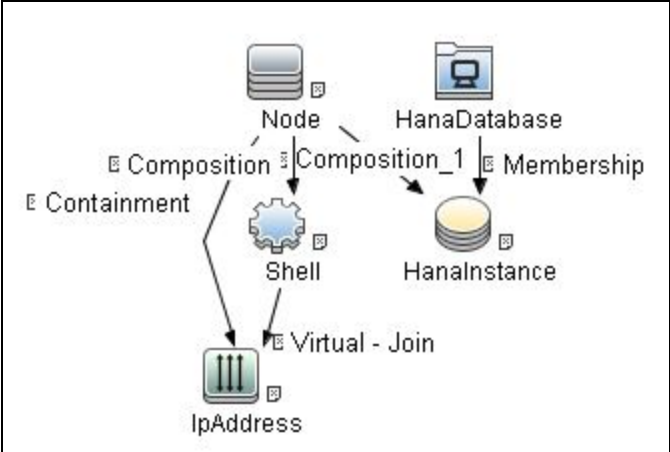
本項の内容

### アダプタ

このジョブでは、**HanaDb\_by\_Shell** アダプタを使用します。

### トリガ・クエリ

**Name :** hanadb



ノード名	条件
Node	なし
HanaDatabase	なし
Shell	NOT Reference to the credentials dictionary entry null
HanaInstance	NOT Application Installed Path null
IPAddress	NOT IP Probe Name null

パラメータ

パラメータは標準設定では上書きされず、アダプタからの値が使用されます。

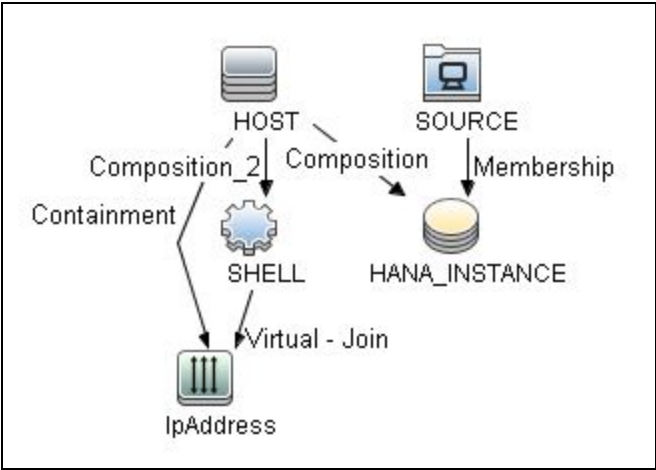
HanaDb\_by\_Shell アダプタ

本項の内容

入力 CIT

Hana Database

入力クエリ



ノード名	条件
HOST	なし
SOURCE	なし
IpAddress	NOT IP Probe Name null
SHELL	NOT Reference to the credentials dictionary entry null
HANA_INSTANCE	NOT Application Installed Path null

トリガ CI データ

名前	値	説明
installpath	\${HANA_INSTANCE.application_path}	現在の HanaDatabase に接続されている各 HanaInstance のインストールパスのリスト
Protocol	\${SHELL.root_class}	既存の各 HanaInstance シェルにつき 1 つのプロトコル名のリスト
credentialsId	\${SHELL.credentials_id}	接続に使用される資格情報辞書のリファレンスのリスト
ip_address	\${SHELL.application_ip}	HanaInstance に接続されている各シェルの IP アドレスのリスト
sid	\${SOURCE.name}	HanaDatabase CI の SID

名前	値	説明
hanadb_cmdbid	\${SOURCE.root_id}	HanaDatabase CI の CMDB ID

検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Database Schema
- DB Data File
- DB User
- DbLogFile
- DbTraceFile
- Dependency
- HanaDatabase
- HanaInstance
- IpAddress
- IpServiceEndpoint
- Node
- Ownership
- Usage

## ディスカバリ・フロー

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、シェル・プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリ・フロー

このアダプタは、インストール・パス属性が空ではなく、各インスタンスのホストに1つ以上の Shell CI が存在するような、利用可能な Hana データベース・インスタンスに対して繰り返し適用されます。

インスタンスに対する接続が正常に確立されている場合、**HanaDb by Shell** ジョブは、そのインスタンスのすべてのリソースの検出を試みます。ディスカバリ・フローを簡略化するため、データベース・リソース(スキーマ、データベース・ユーザ、構成ファイル)も反復のたびに検出されます。

## 第26章: MS-SQL ディスカバリ

### 本章の内容

概要 .....	459
サポートされるバージョン .....	459
トポロジ .....	459
Microsoft SQL Server のデータベース・アプリケーションの検出方法 .....	461
OS 資格情報を使用した MS SQL Server コンポーネントの検出方法 .....	462
Microsoft SQL Server のデータベース・アプリケーションのディスカバリ .....	462
OS 資格情報による SQL サーバ・ディスカバリ .....	463

## 概要

MS SQL ディスカバリは、MS SQL のデータベース・サーバとデータベース・トポロジを検出します。

MS SQL データベース・サーバは、汎用 DB プロトコル(SQL)または OS の資格情報を使用して検出できます。MS SQL データベース・トポロジは、汎用 DB プロトコル(SQL)を使用してのみ検出できます。

## サポートされるバージョン

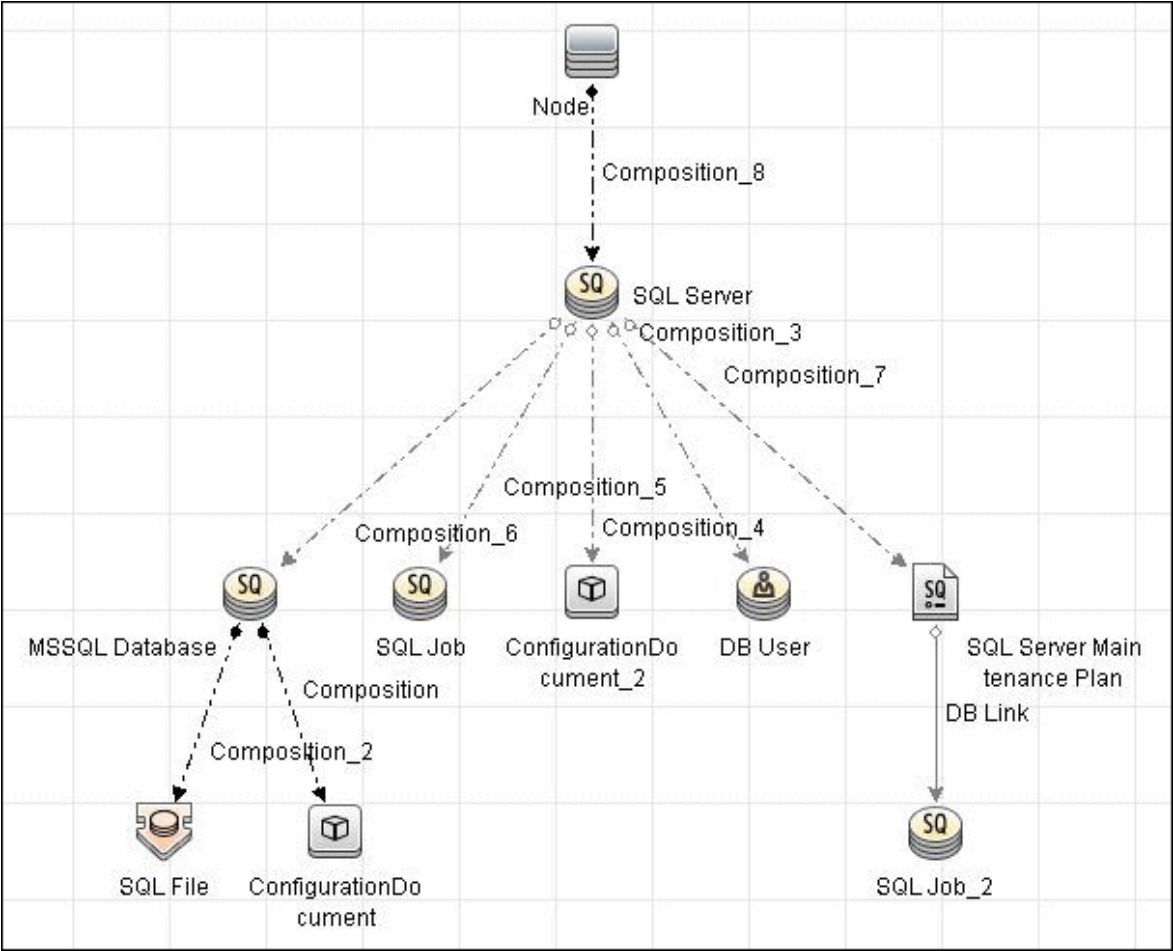
このディスカバリは、Microsoft SQL Server バージョン 2000, 2005, 2008, 2008 R2, 2012, 2014 をサポートしています。

## トポロジ

次の図に、Microsoft SQL Server Database ディスカバリのトポロジを示します。

この図には、Microsoft SQL Server がインストールされているホストが表示されています。Microsoft SQL Server には、データベース、ユーザ、SQL ジョブ、このデータベースの構成ファイル、メンテナンス計画が含まれます。

注: 検出された CIT のリストについては、[「検出された CIT」\(463ページ\)](#)を参照してください。





## Microsoft SQL Server のデータベース・アプリケーションの検出方法

このタスクでは、Microsoft SQL Server のデータベース・アプリケーションを検出する方法を説明します。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

Microsoft SQL Server が汎用 DB プロトコル(SQL)を使用していること。Microsoft SQL Server 向けのこのプロトコルには次が含まれます。

- Microsoft SQL Server プロトコル(認証に使用するデータベース・ログインとパスワード)
- Microsoft SQL Server NTLM プロトコル(認証に使用する OS ログインとパスワード)
- Microsoft SQL Server NTLMv2 プロトコル(認証に使用する OS ログインとパスワードを備えたバージョン 2 のプロトコル)

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - Microsoft SQL Server でのユーザの確認

Microsoft SQL Server で使用するユーザ名、パスワード、ポートを確認します。

### 3. ディスカバリの実行

[Universal Discovery] ウィンドウで、次の順にジョブをアクティブ化します。

- Databases TCP Ports
- MSSQL Server Connection by SQL
- MSSQL Topology by SQL

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## OS 資格情報を使用した MS SQL Server コンポーネントの検出方法

### 1. ディスカバリの実行

次のジョブは、OS 資格情報を使用して MS SQL Server コンポーネントを検出します。

- Host Applications by Shell
- Host Applications by WMI
- DB connections by Shell
- DB connections by WMI

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Microsoft SQL Server のデータベース・アプリケーションのディスカバリ

### アダプタ

#### MSSQL Topology by SQL ジョブのアダプタ・パラメータ

パラメータ	説明
discoverConfigs	<b>True</b> (標準設定)。サーバー構成('mssql database configuration.txt')を取得します。
discoverDbUser	<b>False</b> (標準設定)。MS SQL Server の DB User エンティティは取得されません。
discoverSqlFile	<b>False</b> (標準設定)。MS SQL Server の SQL File エンティティは取得されません。
discoverSqlJob	<b>False</b> (標準設定)。MS SQL Server の SQL Job エンティティは取得されません。

パラメータ	説明
discoverStoredProcedures	<b>False</b> (標準設定)。MS SQL Database Server のストアド・プロシージャは検出されません。
discoverInternalProcedures	<b>False</b> (標準設定)。内部の MS SQL ストアド・プロシージャはディスカバリ中にフィルタ処理され、UCMDB にはレポートされません。 <div> <b>注:</b> 内部の MS SQL ストアド・プロシージャは MSDB と Master データベースから取得され、<b>sp_</b>、<b>xp_</b>、<b>ms_</b>、<b>sysmail_</b> のサフィックスで始まります。 </div>

## 検出された CIT

検出された CIT を表示するには、[リソース] 表示枠で特定のアダプタを選択します。詳細については、『HP Universal CMDB Data Flow Management Guide』の「Discovered CITs Pane」を参照してください。

検出された CI の詳細については、『HP Universal CMDB Data Flow Management Guide』で、ディスカバリの進行状況と結果について解説した項を参照してください。

**注:** トポロジを表示するには、[「トポロジ」\(459ページ\)](#)を参照してください。

## OS 資格情報による SQL サーバ・ディスカバリ

Universal Discovery を使用すると、オペレーティング・システム(OS)の資格情報を使用して MS SQL Server CI を検出できます。Universal Discovery は、一般的な RunningSoftware CI ではなく、識別可能な SQL Server CI を作成します。

以前の SQL Server ディスカバリでは、プロセスの存在を **sqlservr.exe** の名前で判断していました。Universal Discovery がこのプロセスを見つけると、**name** 属性に **MSSQL DB** 値を持つ一般的な実行中ソフトウェアが UCMDB にレポートされていました。

Data Flow Probe では、依存関係リンクで独自の **sqlservr.exe** プロセスにリンクされている SQL Server インスタンスを複数レポートできます。

Universal Discovery は、SQL Server の名前付きインスタンスをサポートしています。

MS SQL Server インスタンスの名前を OS の資格情報で識別する方法は 2 つあります。変更点は **Host\_Resources\_Basic** パッケージに表示されます。

- **プロセス・コマンド・ラインを使用する方法**。SQL Server プロセスでは通常、コマンド・ラインに MS SQL Server インスタンスの名前が含まれます。Universal Discovery はこのインスタンス名を CI に抽出します。

**注:** プロセス・コマンド・ラインを SNMP プロトコルで取得することはできません。したがって、MS SQL Server のインスタンス名を検出するために SNMP を使用することはできず、Universal Discovery は代わりに、一般的な実行中のソフトウェア CI をレポートします。

- **Windows サービスを使用する方法**。Universal Discovery は、コマンド・ラインに **sqlservr.exe** を含む既存サービスを確認し、サービス名からインスタンス名を抽出します(サービス名はインスタンス名を反映しているため)。

# 第27章: SAP MaxDB ディスカバリ

## 本章の内容

- 概要 .....466
- サポートされるバージョン .....466
- トポロジ .....466
- SAP MaxDB の検出方法 .....467
- MaxDb by Shell ジョブ .....468
- MaxDb by Shell アダプタ .....468

## 概要

SAP MaxDB は, SAP AG 製の ANSI SQL-92(エントリ・レベル)準拠のリレーショナル・データベース管理システム(RDBMS)です。MaxDB ディスカバリ・パッケージでは, MaxDB リソースのシャロー・ディスカバリとディープ・ディスカバリを実行できます。

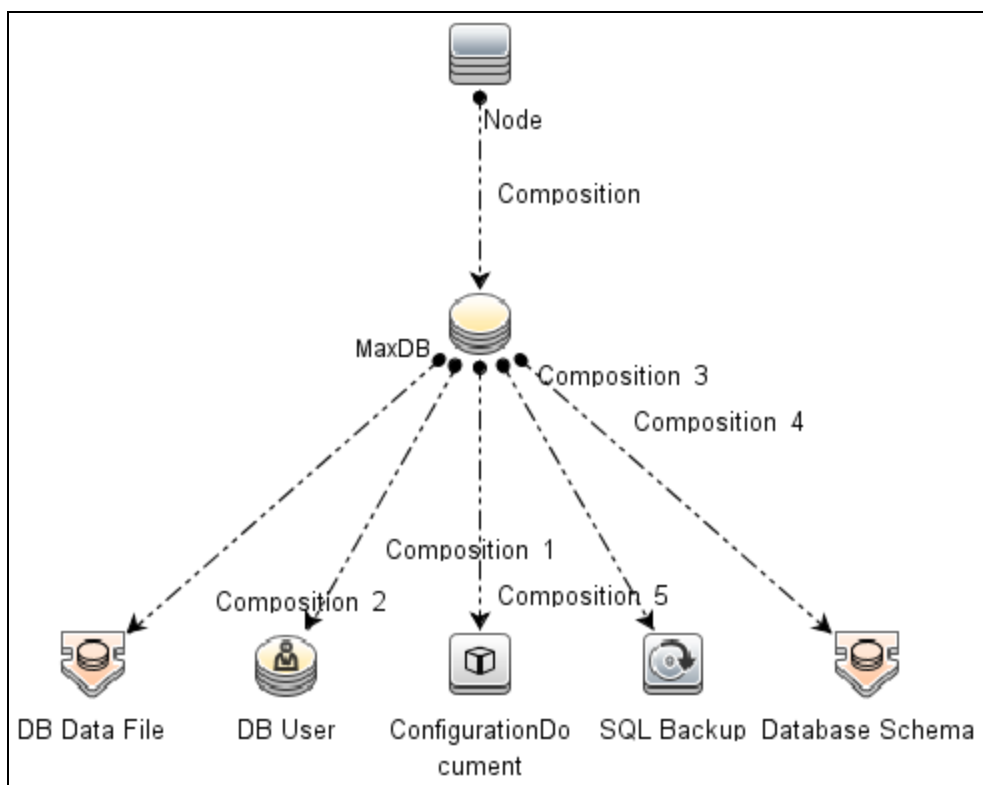
## サポートされるバージョン

このディスカバリは, SAP MaxDB 7.8をサポートしています。

## トポロジ

次の図に, SAP MaxDB Database ディスカバリのトポロジを示します。

検出された CIT のリストについては, [「検出された CIT」\(470ページ\)](#)を参照してください。



## SAP MaxDB の検出方法

本項では、SAP MaxDB のトポロジの検出方法について説明します。

本項の内容

### 1. 前提条件 - 接続およびユーザ・ストア

- a. MaxDB ノード へのシェル接続
- b. 検出される各 MaxDB インスタンスにつき 1 つのキーを持つ適切に構成されたキー・ストア

注: **xuser** コマンドは、**dbmcli** ツールの実行に使用されるため、ツールの呼び出しが適切に認証されるように、キー・ストアは検出先に作成する必要があります。

### 2. 前提条件 - プロトコル資格情報の設定

プラットフォームに応じて、次の資格情報のいずれかを定義します。

- SSH
- Telnet
- NTCMD

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 3. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by Shell** ジョブを実行し、ターゲット・ホストおよびそのホストへのシェル接続を検出します。
- c. **Host Applications by Shell** ジョブを実行し、ターゲット・ホストのリソースを検出します。これには MaxDB ソフトウェアと関連プロセスが含まれます。
- d. **MaxDb by Shell** ジョブを実行し、ターゲット MaxDB データベースのトポロジを検出します。

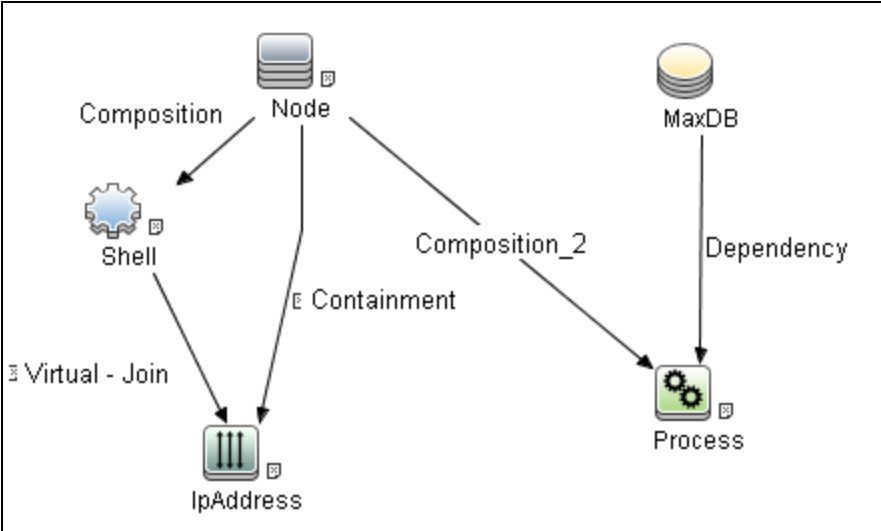
## MaxDb by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、**MaxDb by Shell** アダプタを使用します。

### トリガ・クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
Process	(Name 等しい kernel OR Name 等しい kernel.exe) AND NOT Process Path null
Shell	NOT Reference to the credentials dictionary entry null

### パラメータ

なし。

## MaxDb by Shell アダプタ

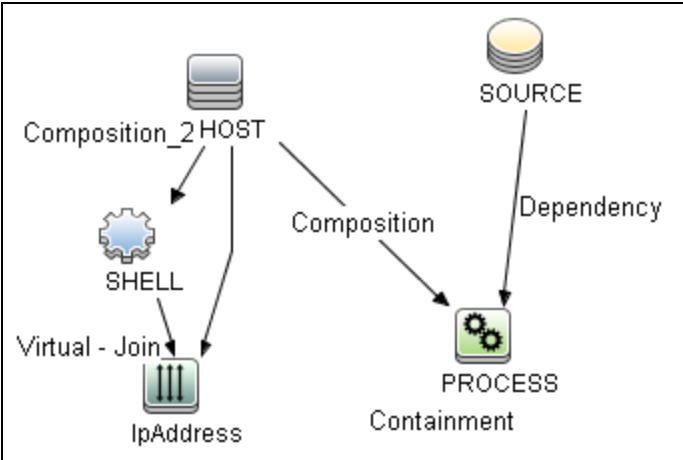
本項ではこのアダプタに関する詳細を説明します。



入力 CIT

MaxDB

入力クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
PROCESS	Name 等しい kernel OR Name 等しい kernel.exe

トリガ CI データ

名前	値
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
dbDataPath	\${SOURCE.data_path}
dbPort	\${SOURCE.application_port:}
dbProgramPath	\${SOURCE.program_path}
dbSid	\${SOURCE.name}
dbVersion	\${SOURCE.application_version}
ip_address	\${SHELL.application_ip}

名前	値
processParams	\${PROCESS.process_parameters:}
processPath	\${PROCESS.process_path:}

使用スクリプト

- db\_platform.py
- db.py
- db\_builder.py
- maxdb.py
- maxdb\_base\_parser.py
- maxdb\_base\_discoverer.py
- maxdb\_discoverer.py
- maxdb\_xuser.py
- maxdb\_by\_shell.py

検出された CIT

- Composition
- ConfigurationDocument
- Containment
- DB Data File
- DB User
- Database Schema
- IpAddress
- IpServiceEndpoint

- MaxDB
- Node
- SQL Backup

## 第28章: データベース間の MySQL レプリケーション・ディスカバリ

### 本章の内容

概要 .....	473
サポートされるバージョン .....	473
トポロジ .....	473
MySQLの構成ジョブとレプリケーション・ジョブの検出方法 .....	474
MySQL by Shell ジョブ .....	475
トラブルシューティングおよび制限事項 .....	480

## 概要

本章では、マスタ / スレーブ関係でデータをレプリケートする MySQL データベース・サーバの検出方法を説明します。

レプリケーションにより、ある MySQL データベース・サーバ(マスタ)のデータを、別の 1 つ以上の MySQL データベース・サーバ(スレーブ)にレプリケートできます。レプリケーションの詳細については、MySQL Web サイトにある MySQL のマニュアル(<http://dev.mysql.com/doc/refman/5.0/en/replication-howto.html>)を参照してください。

現在、データベースに関するすべての情報は、シェル・プロトコルを使用して MySQL 構成ファイルから取得されます。

MySQL デイスカバリを実行するジョブは、**MySQL by Shell** です。

## サポートされるバージョン

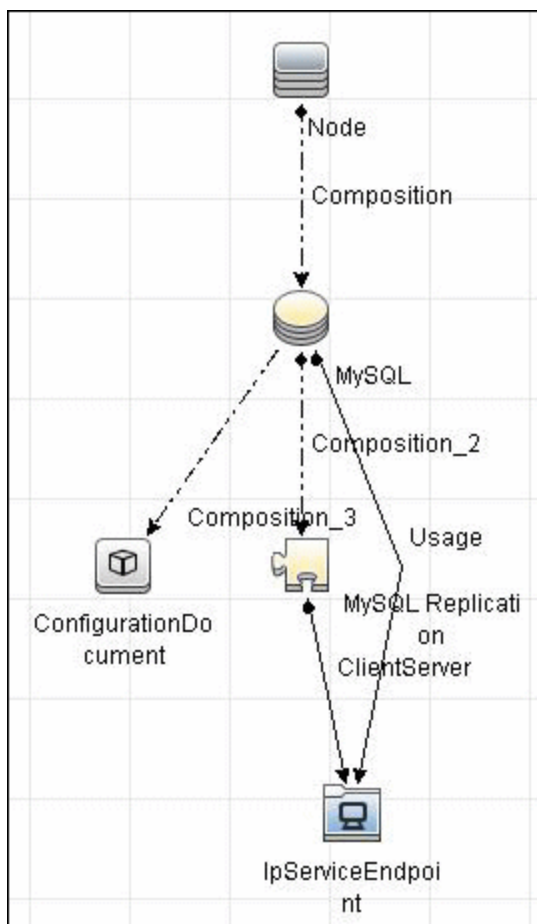
このデイスカバリは、次をサポートしています。

- MySQL バージョン 4.x, 5.x, 6.0
- オペレーティング・システム: Windows, Solaris, Linux

## トポロジ

注: 検出された CIT のリストについては、「[MySQL by Shell ジョブ](#)」(475ページ)を参照してください。

## MySQL レプリケーション・ジョブ



## MySQLの構成ジョブとレプリケーション・ジョブの検出方法

このタスクでは、MySQL の構成とレプリケーションに関するジョブの検出方法を次の手順で説明します。

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、次のプロトコルを使用します。

- SSH プロトコル
- Telnet プロトコル
- NTCMD プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - 情報の取得

関連するすべての情報を取得するには、\$MYSQL\_HOME ディレクトリの読み取り権限と、次のパラメータを指定した **mysqld**(Windows の場合は **mysqld.exe** または **mysqld-nt.exe**)の実行権限が DFM に必要です。

```
mysqld --verbose --help
```

```
mysqld --version
```

**my.cnf(my.ini)**ファイルが \$MYSQL\_HOME ディレクトリ以外の場所にある場合は、そのファイルの読み取り権限を追加する必要があります。

## 3. デイスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行して、IP 範囲内で稼働中のマシンを検出します。
- b. **Host Connection by Shell** ジョブを実行して Shell CIT を作成します。
- c. ホスト・リソースのジョブを実行して、ホスト上で実行中のプロセスに関する情報を収集します。
- d. **MySQL by Shell** ジョブを実行して、MySQL の構成とレプリケーションのジョブに関する情報を取得します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

# MySQL by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

## デイスカバリ・メカニズム

本項では、DFM による MySQL サーバの検出方法について説明します。

1. MySQL by Shell ジョブが、シェルの資格情報を使用したリモート・ホスト接続を実行します。
2. ジョブが、次のコマンドを実行して、MySQL 構成ファイルのパスが存在するかどうかをチェックします。

```
mysqld --verbose --help
```

3. このコマンドを使用しても構成ファイルが見つからない場合、ジョブは構成ファイルが標準設定のパスにあるとみなします。
  - UNIX または Linux の場合 : **/etc/my.cnf**
  - Windows の場合 : **../my.ini**
4. ジョブが、構成ファイルの属性値の取得を試みます。コマンド・ラインから属性値を読み取るか、または構成ファイルを読み取って、コマンド・ラインでは見つからなかった属性値を検索します。

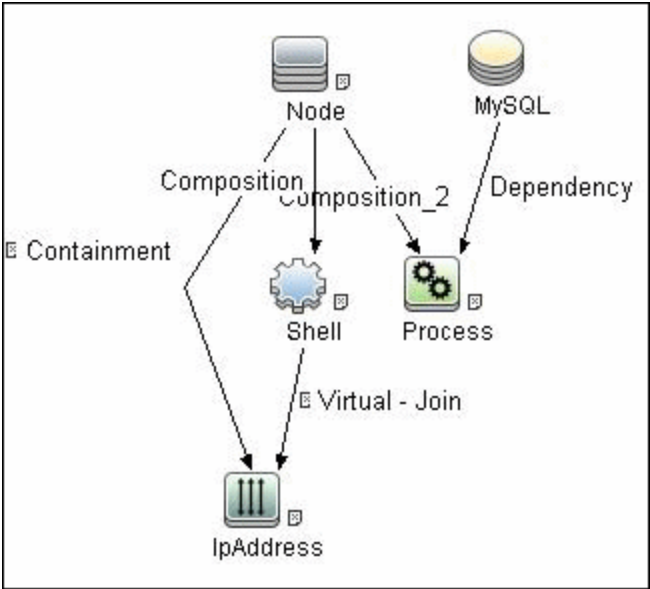
**属性値を伴うコマンド・ラインの例 :**

```
mysqld-nt.exe --defaults-file=C:\hp\UCMDB\DataFlowProbe\MySQL\my.ini DDM_Probe_DB
```

5. 属性値が見つからなかった場合、ジョブは MySQL ドキュメントから標準設定値を取得します。  
MySQL 属性の詳細については、[「MySQL by Shell ジョブ」\(475ページ\)](#)を参照してください。
6. ジョブが、適切な属性値と関係を使用して MySQL CI を作成します。
7. ジョブは次に、この MySQL インスタンスがレプリカかどうかをチェックします。レプリカである場合はマスタ・ホストとマスタ・ユーザを検索します。MySQL エンジンのバージョンは **mysqld --version** コマンド出力から取得されます。
8. ジョブが、適切な属性値と関係を使用して MySQL replication CI を作成します。



トリガ・クエリ



構成アイテム・タイプ

名前	親 CIT	既存の属性を使用する場合	新しい属性を使用する場合	説明
MySQL	Database	database_dbsid	server_id, database_ datadir, database_max_ connections	CIT は MySQL データベースを表す
MySQL Replication	DB Scheduler Job		master_user, master_connect_ retry	CIT は MySQL レプリケーション・ジョブを表す

CIT 属性

- MySQL
  - server\_id。サーバID はレプリケーション・ジョブで使用されるもので、各サーバで一 意 であることが必要。
  - database\_datadir。データベース・ルートへのパス(構成ファイルのdatadir)。

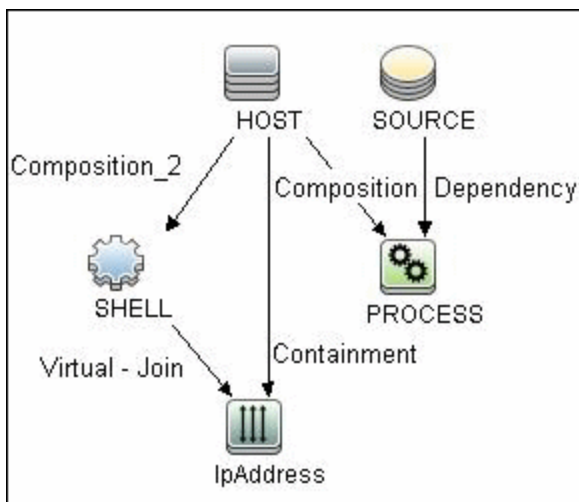
- database\_max\_connections。MySQL サーバが許可する同時実行セッションの最大数(my.ini ファイルの max\_connections)。
- database\_dbsid。MySQL のインスタンス・プロセス・ポートを実行するための一意の識別子。形式は「MySQL on port #####」。
- MySQL Replication
  - master\_user。マスタ・サーバに接続するときに使用されるユーザ名。
  - master\_connect\_retry。マスタが停止した, または接続が失われた場合に, マスタへの再接続を試みるまでにスレーブ・スレッドがスリープする秒数。

## 関係

ソース	ターゲット	関係タイプ	カーディナリティ
mysql	configfile	Composition	1..1
mysql	mysql_replication	Composition	1..1
mysql_replication	IpServiceEndpoint	ClientServer	1..1

## アダプタ

- 入力クエリ

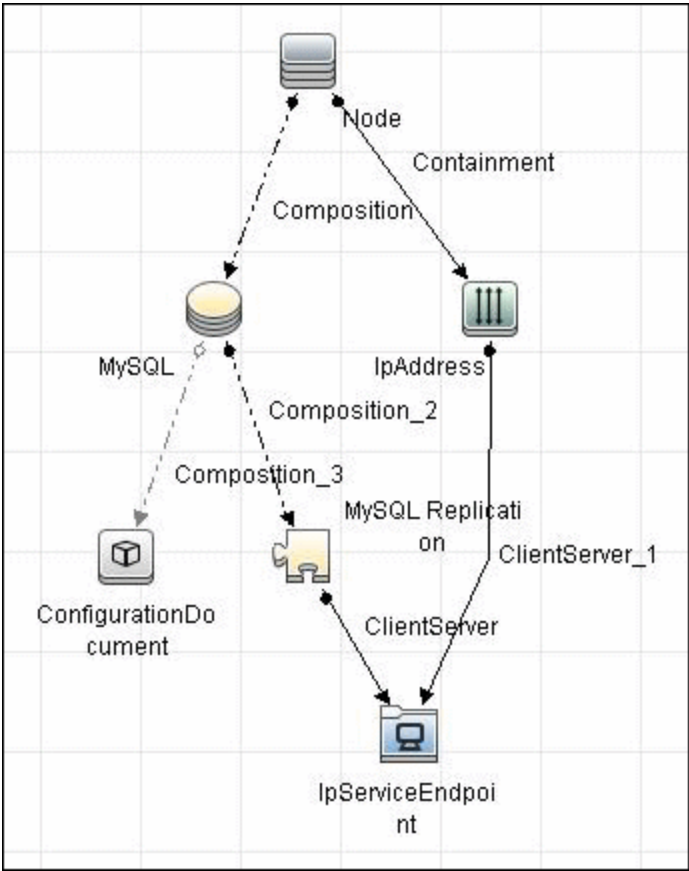


• トリガ CI データ

名前	値
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
dbport	\${SOURCE.database_dbport}
dbsid	\${SOURCE.database_dbsid}
ip_address	\${SHELL.application_ip}
processParams	\${PROCESS.process_parameters}
processPath	\${PROCESS.process_path}

検出された CIT

検出された CIT を表示するには、[リソース]表示 枠で特定のアダプタを選択します。



- ClientServer
- Composition
- ConfigurationDocument
- Containment
- IpAddress
- IpServiceEndpoint
- MySQL
- MySQL Replication
- Node

注: トポロジを表示するには, [「トポロジ」\(473ページ\)](#)を参照してください。

## トラブルシューティングおよび制限事項

本項では, データベース間の MySQL レプリケーション・ディスカバリのトラブルシューティングと制限事項を説明します。

- 1 台のホストで, アクティブな MySQL インスタンスを複数実行するには 2 つの主な方法があります。
  - 2 つの MySQL インスタンスをそれぞれ別のポートで実行する方法 (例: 1 つを 134.44.1.1:3306, もう 1 つを 134.44.1.1:3307 で実行)。
  - ホストに複数の IP があり, 各 MySQL プロセスは独自の IP にバインドする方法 (例: 134.44.1.1:3306 と 134.44.1.2:3306)。

2 番目のケースでは, ある MySQL CI を別の MySQL CI と区別する主な識別子がポート番号 (IP なし) なので, ジョブは 2 つの MySQL インスタンスを区別できず, これらを 1 つの CI に統合します。

## 第29章: Oracle Database サーバ・ディスカバリ

### 本章の内容

概要 .....	482
サポートされるバージョン .....	482
トポロジ .....	483
Oracle Database サーバの検出方法 .....	483
Oracle TNS Names トポロジの検出方法 .....	484
Oracle Database サーバの検出方法 - ライトウェイト .....	485
Oracle Database Connection by SQL- Lightweight ジョブ .....	486
Oracle Connection by SQL- Lightweight アダプタ .....	487
Oracle Database Connection by SQL ジョブ .....	489
Oracle Connection by SQL アダプタ .....	490
Oracle TNS Names by LDAP ジョブ .....	492
Oracle TNS Names by LDAP アダプタ .....	493
Oracle Topology by SQL ジョブ .....	495
Oracle Database Topology by SQL アダプタ .....	496
トラブルシューティングおよび制限事項 .....	499

## 概要

Oracle Database サーバ・ディスカバリには、次の 2 つのタイプがあります。

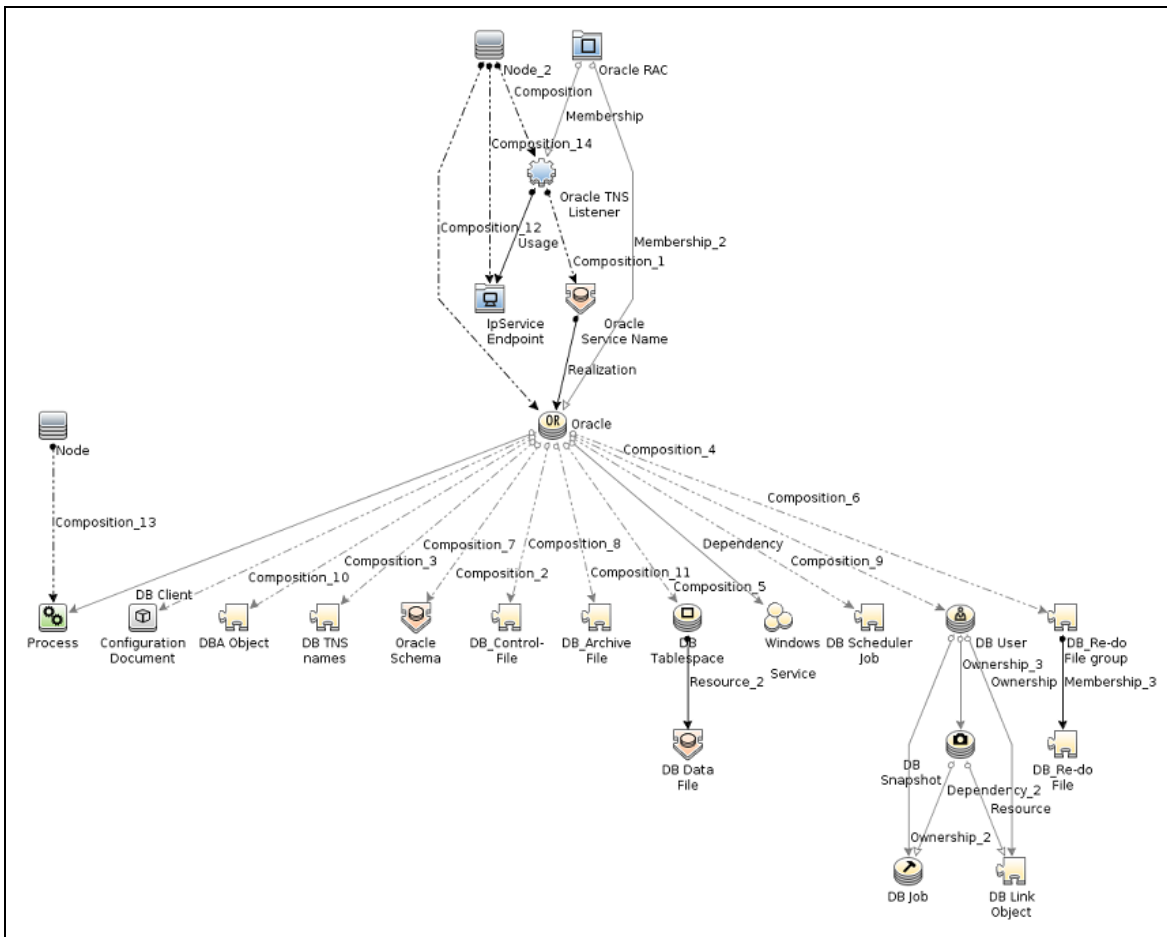
- 一般的な Oracle Database サーバ・ディスカバリ。ネットワーク上のすべての Oracle Database サーバを検出します。次の 2 つの方法を利用できます。
  - **完全 ディスカバリ**。ノード CI を入力トリガ CIT として使用する包括的なディスカバリ方法です。このタイプのディスカバリの詳細については、[「Oracle Database サーバの検出方法」\(483 ページ\)](#)を参照してください。
  - **ライトウェイト・ディスカバリ**。IPServiceEndpoint を入力トリガ CIT として使用する軽量のディスカバリ方法です。このタイプのディスカバリの詳細については、[「Oracle Database サーバの検出方法 - ライトウェイト」\(485 ページ\)](#)を参照してください。
- Oracle TNS Names トポロジ・ディスカバリ。LDAP プロトコルを使用して、Active Directory に格納されたすべての Oracle Database サーバを検出します。このタイプのディスカバリの詳細については、[「Oracle TNS Names トポロジの検出方法」\(484 ページ\)](#)を参照してください。

## サポートされるバージョン

Oracle Database サーバ・ディスカバリおよび Oracle TNS Names トポロジ・ディスカバリは、Oracle 8, 9, 9i, 10, 10g, 11g, 12c をサポートしています。

トポロジ

次の図に、Oracle Database サーバ・ディスクバリのトポロジを示します。



## Oracle Database サーバの検出方法

このタスクでは、Oracle Database の検出方法について説明します。このディスカバリでは、有効な資格情報 ID が CMDDB に追加されます。その後、データベースを完全に検出するためにこの CI を使用できません。

## 本項の内容

## 1. 前提条件 - プロトコル資格情報の設定

Oracle Database サーバ・ディスカバリは、汎用 DB プロトコル(SQL)を使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - Oracle Database サーバのユーザの確認

**Databases TCP Ports** ジョブを実行し、Oracle Database サーバが使用するユーザ名、パスワード、ポートを確認します。

## 3. ディスカバリの実行

次のジョブを次の順序でアクティブにします。

- a. **Databases TCP Ports**
- b. **Oracle Database Connection by SQL**
- c. **Oracle Topology by SQL**

**注:** Oracle Topology by SQL ジョブがレポートするデータが大量なため、トポロジ・データはチャンクで送信されます。チャンク・サイズ(チャンク内のオブジェクト数)は、**discoverReportPageSize** ジョブ・パラメータによって制御されます。標準設定値は、1 つのチャンクにつきオブジェクト 1,000 個です。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

# Oracle TNS Names トポロジの検出方法

本項では、LDAP プロトコルを使用して Active Directory から Oracle TNS Names トポロジを検出する方法について説明します。

## 前提条件

- Oracle TNS 情報が格納されている Active Directory サーバが、適切に構成されていることを確認します。



- LDAP プロトコル資格情報が適切に構成されていることを確認します。

## ディスカバリの実行

1. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
2. **TCP Ports** ジョブを実行し、LDAP ポートを検出します。LDAP ポートが検出されたことを確認します。
3. **Active Directory Connection by LDAP** ジョブを実行し、Active Directory サーバを検出します。
4. **Oracle TNS Names by LDAP** ジョブを実行し、Oracle TNS Names トポロジを検出します。

## Oracle Database サーバの検出方法 - ライトウェイト

このタスクでは、ディスカバリのライトウェイトの方法を使用して Oracle Database を検出する方法について説明します。このディスカバリでは、有効な資格情報 ID が CMDB に追加されます。その後、データベースを完全に検出するためにこの CI を使用できます。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

Oracle Database サーバ・ディスカバリは、汎用 DB プロトコル(SQL)を使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - Oracle Database サーバのユーザの確認

**Databases TCP Ports** ジョブを実行し、Oracle Database サーバが使用するユーザ名、パスワード、ポートを確認します。

### 3. ディスカバリの実行

次のジョブを次の順序でアクティブにします。

- a. **Databases TCP Ports**
- b. **Oracle Database Connection by SQL- Lightweight**
- c. **Oracle Topology by SQL**

**注:** Oracle Topology by SQL ジョブがレポートするデータが大量なため、トポロジ・データはチャンクで送信されます。チャンク・サイズ(チャンク内のオブジェクト数)は、**discoverReportPageSize** ジョブ・パラメータによって制御されます。標準設定値は、1 つのチャンクにつきオブジェクト 1,000 個です。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Oracle Database Connection by SQL- Lightweight ジョブ

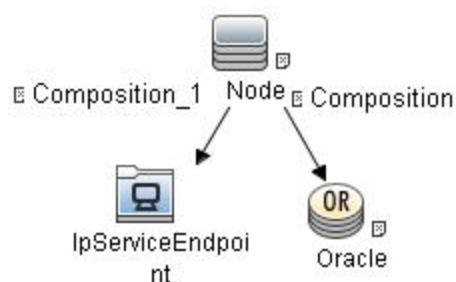
本項ではこのジョブに関する詳細を説明します。

### アダプタ

ID : SQL\_NET\_Dis\_Connection\_Oracle\_Lightweight

### トリガ TQL

- トリガ CI : IpServiceEndpoint
- トリガ・クエリ:



- CI 属性条件 :

CI	属性値
IpServiceEndpoint	IpServiceName 等しい (大文字小文字の区別なし) oracle OR ServiceNames 含む oracle

## デイスカバリ・フロー

このジョブは、次のアクションを実行します。

1. デイスカバリ・プローブ構成内に指定された Oracle 資格情報を取得します。
2. 特定の資格情報を使用して接続を試みます。
3. 接続に成功した場合、接続された資格情報と一緒に Oracle Database をレポートします。

## Oracle Connection by SQL- Lightweight アダプタ

本項ではこのアダプタに関する詳細を説明します。

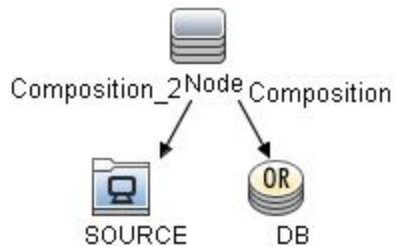
### ID

Oracle Connection By SQL- Lightweight

### 入力 CIT

IpServiceEndpoint

### 入力 TQL



## トリガ CI データ

名前	値
application_ip	\${DB.application_ip:NA}
application_port	\${DB.application_port:NA}
sa_ip	\${SOURCE.bound_to_ip_address:NA}
sa_port	\${SOURCE.network_port_number:NA}
sid	\${DB.name:NA}

## 使用スクリプト

- file\_ver\_lib.py
- SQL\_Connection.py

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node
- Oracle
- Oracle Service Name
- Usage

## パラメータ

- **protocolType**。標準設定値である **oracle** は変更できません。

## Oracle Database Connection by SQL ジョブ

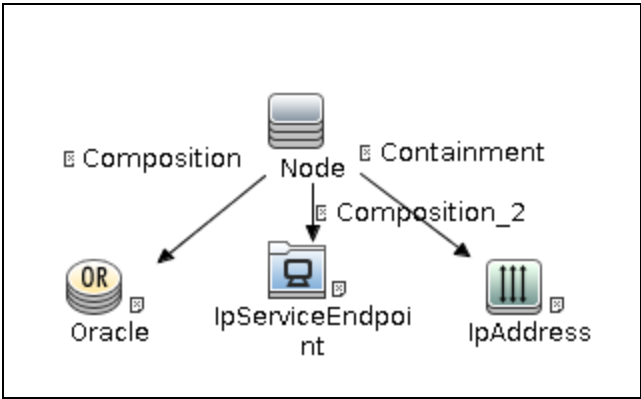
本 項 ではこのジョブに関する詳 細を説 明します。

### アダプタ

ID:Oracle Connection By SQL

### トリガ TQL

- トリガ CI :Node
- トリガ・クエリ:



- CI 属性条件 :

CI	属性値
IpAddress	NOT IP Probe Name null
IpServiceEndpoint	IpServiceName 等しい (大文字小文字の区別なし) oracle OR ServiceNames 含む oracle

### ディスカバリ・フロー

このジョブは、次のアクションを実行します。

1. ディスカバリ・プローブ構成内に指定された Oracle 資格情報を取得します。
2. 特定の資格情報を使用して接続を試みます。
3. 接続に成功した場合、接続された資格情報と一緒に Oracle Database を報告します。
4. 接続に成功した場合、プライマリ IP をデータベースに問い合わせ、その IP が接続に使用された IP と一致するかどうかを確認します。

プライマリ IP が接続に使用されたものと一致しない場合、ジョブはデータベースがクラスタ化モードにあるかどうかを確認します。

5. データベースが報告した IP がプローブからアクセス可能で接続できる場合、その IP は UCMDDB にポピュレートされます。それ以外の場合、初期接続に使用された IP が UCMDDB にポピュレートされます。

## Oracle Connection by SQL アダプタ

本項ではこのアダプタに関する詳細を説明します。

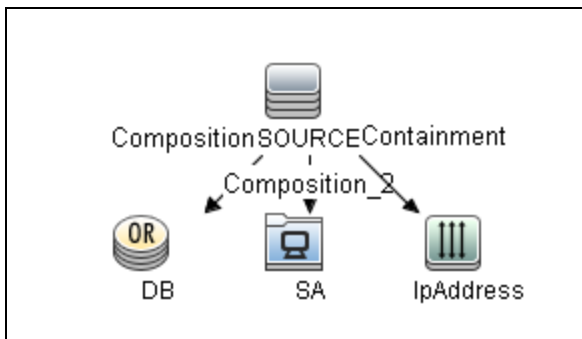
### ID

Oracle Connection By SQL

### 入力 CIT

Node

### 入力 TQL



トリガ CI データ

名前	値
application_ip	\${DB.application_ip:NA}
application_port	\${DB.application_port:NA}
ip_address	\${IpAddress.name}
sa_ip	\${SA.bound_to_ip_address:NA}
sa_port	\${SA.network_port_number:NA}
sid	\${DB.name:NA}

使用スクリプト

- file\_ver\_lib.py
- SQL\_Connection.py

検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node
- Oracle
- Oracle Service Name
- Usage

パラメータ

- **protocolType**。標準設定値である **oracle** は変更できません。

## Oracle TNS Names by LDAP ジョブ

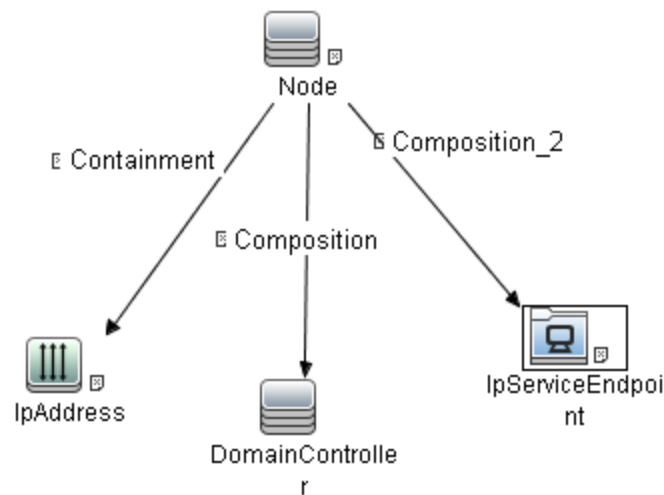
本項ではこのジョブに関する詳細を説明します。

### アダプタ

ID : oracle\_tns\_names\_by\_ldap

### トリガ TQL

- トリガ CI :IpAddress
- トリガ・クエリ :



- CI 属性条件 :

CI	属性値
IpAddress	NOT IP Probe Name null



CI	属性値
<b>IpServiceEndpoint</b>	IpServiceName 等しい (大文字小文字の区別なし) ldap OR ServiceNames 含む ldap
<b>DomainController</b>	NOT Reference to the credentials dictionary entry null AND NOT Application IP null

## パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

## ディスカバリ・フロー

1. **DomainContext** をルートから取得します(**baseDN** パラメータが指定されていない場合)。
2. **orclContext** オブジェクトを前のノードから取得します。
3. **orclNetService** オブジェクトを **orclContext** ノードから取得します。
4. **orclNetDescString** 属性を **orclNetService** オブジェクトから取得します。

# Oracle TNS Names by LDAP アダプタ

本項ではこのアダプタに関する詳細を説明します。

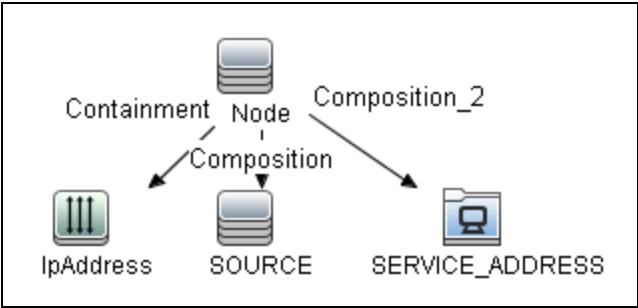
## ID

oracle\_tns\_names\_by\_ldap

## 入力 CIT

DomainController

入力 TQL



トリガ CI データ

名前	値
application_port	\${SOURCE.application_port:NA}
credentials_id	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}
port	\${SERVICE_ADDRESS.network_port_number}

使用スクリプト

- active\_directory\_utils.py
- db.py
- db\_builder.py
- db\_platform.py
- jdbc\_url\_parser.py
- oracle\_by\_ldap.py
- oracle\_ldap\_discoverer.py

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node
- Oracle
- Oracle Service Name
- Oracle TNS Listener
- Realization
- Usage

## パラメータ

- **baseDN orclContext** オブジェクトの検索対象のコンテキストを指定します。

# Oracle Topology by SQL ジョブ

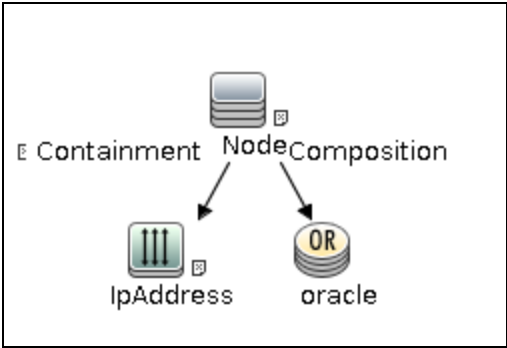
本項ではこのジョブに関する詳細を説明します。

## アダプタ

**ID :** Oracle database topology by SQL

## トリガ TQL

- トリガ CI : Oracle
- トリガ・クエリ :



- CI 属性条件 :

CI	属性値
IpAddress	NOT IP Probe Name null
oracle	NOT Reference to the credentials dictionary entry null

ディスカバリ・フロー

1. Oracle CI に指定された資格情報 ID を使用して、Oracle Database に接続します。
2. アダプタのパラメータに指定されたフラグに従って、特定の SQL クエリを実行します。
3. 検出されたデータを UCMDDB にプッシュします。

Oracle Database Topology by SQL アダプタ

本項ではこのアダプタに関する詳細を説明します。

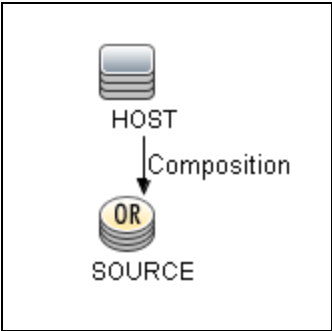
ID

Oracle database topology by SQL

入力 CIT

Oracle

入力 TQL



トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}
port	\${SOURCE.application_port:NA}
sid	\${SOURCE.name:NA}

使用スクリプト

- SQL\_Dis\_Oracle.py

検出された CIT

- Composition
- Containment
- DB Client
- DB Data File
- DB Job
- DB Link Object

- DB Scheduler Job
- DB Snapshot
- DB Tablespace
- DB User
- DB\_Archive File
- DB\_Control-File
- DB\_Re-do File group
- DB\_Re-do File
- DBA Object
- Dependency
- IpAddress
- Membership
- Node
- Oracle RAC
- Oracle Schema
- Oracle
- Ownership
- Process
- Resource

## パラメータ

- **comprehensiveDiscovery**。false の場合、DBA Object、DB Job、DB User は取得されません。標準設定値は false です。パラメータが false に設定されている場合、その他のパラメータを変更しても、ディスカバリ・プロセスに影響はありません(DBA Object は検出されません)。

- **discoverFunctions**。false の場合、ファンクションは取得されません。標準設定値は false です。
- **discoverPackageBody**。false の場合、パッケージ本体は取得されません。標準設定値は false です。
- **discoverPackages**。false の場合、パッケージは取得されません。標準設定値は false です。
- **discoverProcedures**。false の場合、プロシージャは取得されません。標準設定値は false です。
- **discoverReportPageSize**。1 つのバルク内で送信できるクエリ結果オブジェクトの最大量を示します。標準設定値は 1,000 です。
- **discoverTables**。false の場合、テーブルは取得されません。標準設定値は false です。
- **discoveryDBClients**。true の場合、DB クライアントが取得されます。標準設定値は true です。
- **discoveryUsers**。true の場合、データベース・ユーザが取得されます。false の場合、データベース・ユーザの検出は、**comprehensiveDiscovery** パラメータによって制御されます。
- **excludeSysUser**。false の場合、システムのデフォルト・ユーザが含まれます。標準設定値は false です。

## トラブルシューティングおよび制限事項

本項では、Oracle Database サーバ・ディスカバリのトラブルシューティングと制限事項について説明します。

- Oracle 8g を検出する必要がある場合、次の値を **jdbcDrivers** および **jdbcPreUrls** に使用します。

**jdbcDrivers :**

```
<oracle>com.inet.ora.OraDriver</oracle>
<oracleSSL>com.mercury.jdbc.oracle.OracleDriver</oracleSSL>
```

**jdbcPreUrls :**

```
<oracle>jdbc:inetora:%%ipaddress%%:%%protocol_port%%:%%sqlprotocol_
dbsid%%?logging=false&loginTimeout=%%protocol_timeout%%</oracle>
<oracleSSL>jdbc:mercury:oracle://%%ipaddress%%:%%protocol_
port%%;ServiceName=%%sqlprotocol_dbsid%%</oracleSSL>
```

# 第30章: Oracle Real Application Cluster(RAC) ディスカバリ

## 本章の内容

概要 .....	501
サポートされるバージョン .....	501
トポロジ .....	501
Oracle Real Application Cluster(RAC)の検出方法 .....	502
Oracle Listeners by Shell ジョブ .....	503
Oracle RAC Topology by Shell ジョブ .....	507
構成アイテム .....	510
関係 .....	511
トラブルシューティングおよび制限事項 .....	512



## 概要

DFM は、シェル・プロトコルを介して Oracle 構成ファイル **listener.ora** と **tnsnames.ora** から、**lsnrct** ユーティリティを介して Oracle RAC に関する情報を検出します。

## サポートされるバージョン

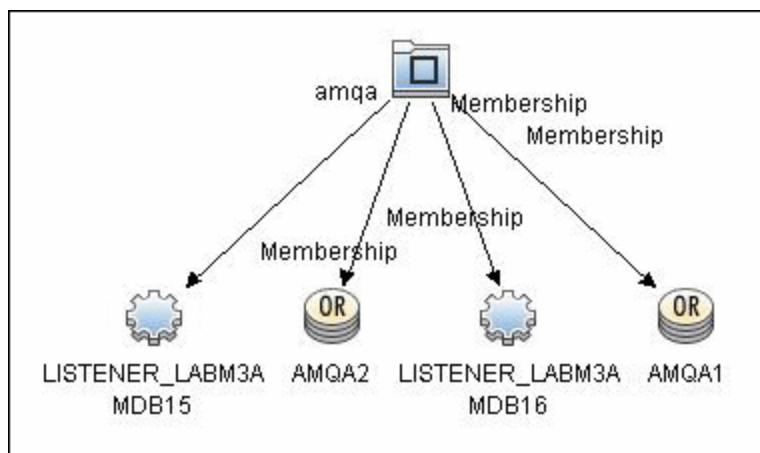
このディスカバリは、Oracle DB 10g, 11g, 12c をサポートしています。

## トポロジ

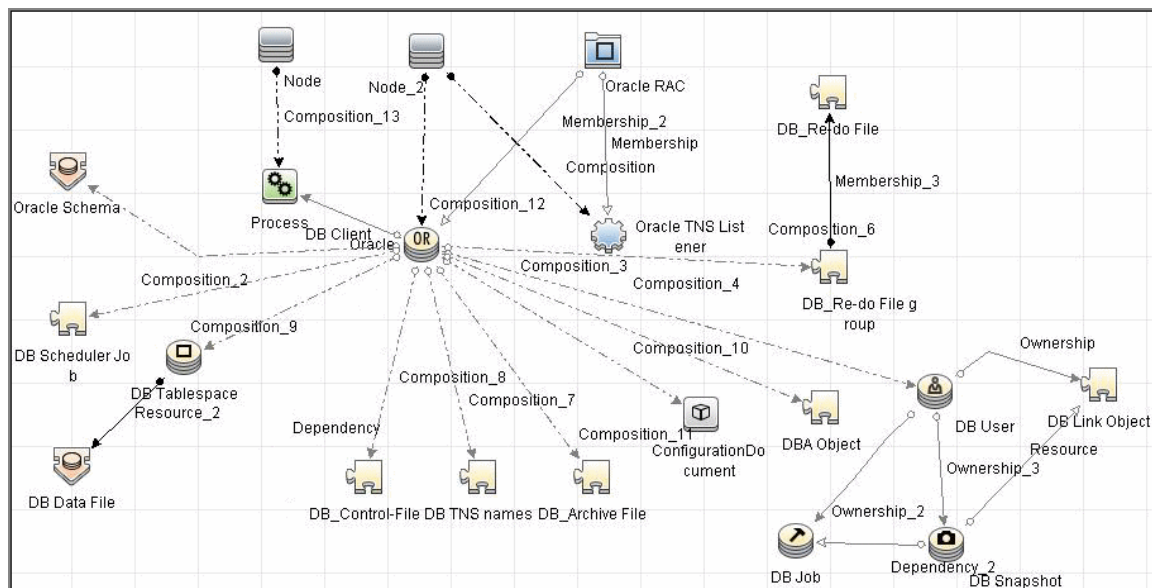
次の図に、Oracle RAC ディスカバリのトポロジのサンプル出力を示します。

注: 検出された CIT の一覧については、[「Oracle Listeners by Shell ジョブ」\(503ページ\)](#)、[「Oracle RAC Topology by Shell ジョブ」\(507ページ\)](#)を参照してください。

### • トポロジ



## • Oracle ビュー



# Oracle Real Application Cluster(RAC)の検出方法

## 本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、NTCMD プロトコル、SSH プロトコル、または Telnet プロトコルが使用されます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - その他

a. すべての関連情報を取得するには、DFM に次の権限があることを確認します。

- \$ORACLE\_HOME\network\admin ディレクトリの読み取り権限
- \$ORACLE\_HOME\bin\lsnrctl および対応するライブラリ・ファイル(lib)とメッセージ・ファイルの適切な実行権限

b. **Oracle Listeners by Shell ジョブ**。RAC 関連プロセスが Oracle Database 上で実行されていることを確認します。ファイル名は ora\_lms, ora\_lmd, ora\_lck, oracm で始まります。

- c. **Oracle RAC Topology by Shell ジョブ**。Listener CIT の **Listened IPs** は **not NULL** である必要があります。
- d. **Host Connection by Shell** ジョブを実行し、Shell CIT をアクティブにします。

### 3. ディスカバリの実行

- a. ホスト上で実行されているプロセスに関する情報を収集する任意のホストリソース・ジョブを実行します。たとえば **Host Applications by Shell** を実行します。

DFM が TNS リスナ・プロセスを検出した場合、このジョブは Oracle TNS Listener CI および Oracle DB CI をその関連プロセスと一緒に作成します。

- b. Oracle TNS Listener CI をフル・データで検出するには、**Oracle Listeners by Shell** ジョブを実行します。このジョブはホストに接続し、Oracle TNS Listener CI のために必要なデータを取得します。詳細については、[「Oracle Listeners by Shell ジョブ」\(503ページ\)](#)を参照してください。
- c. Oracle RAC トポロジを検出するには、**Oracle RAC Topology by Shell** ジョブを実行します。このジョブは、ホストをフル・リスナと接続し、RAC を検出します。詳細については、[「Oracle RAC Topology by Shell ジョブ」\(507ページ\)](#)を参照してください。検出されない要素の詳細については、[「トラブルシューティングおよび制限事項」\(512ページ\)](#)を参照してください。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Oracle Listeners by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

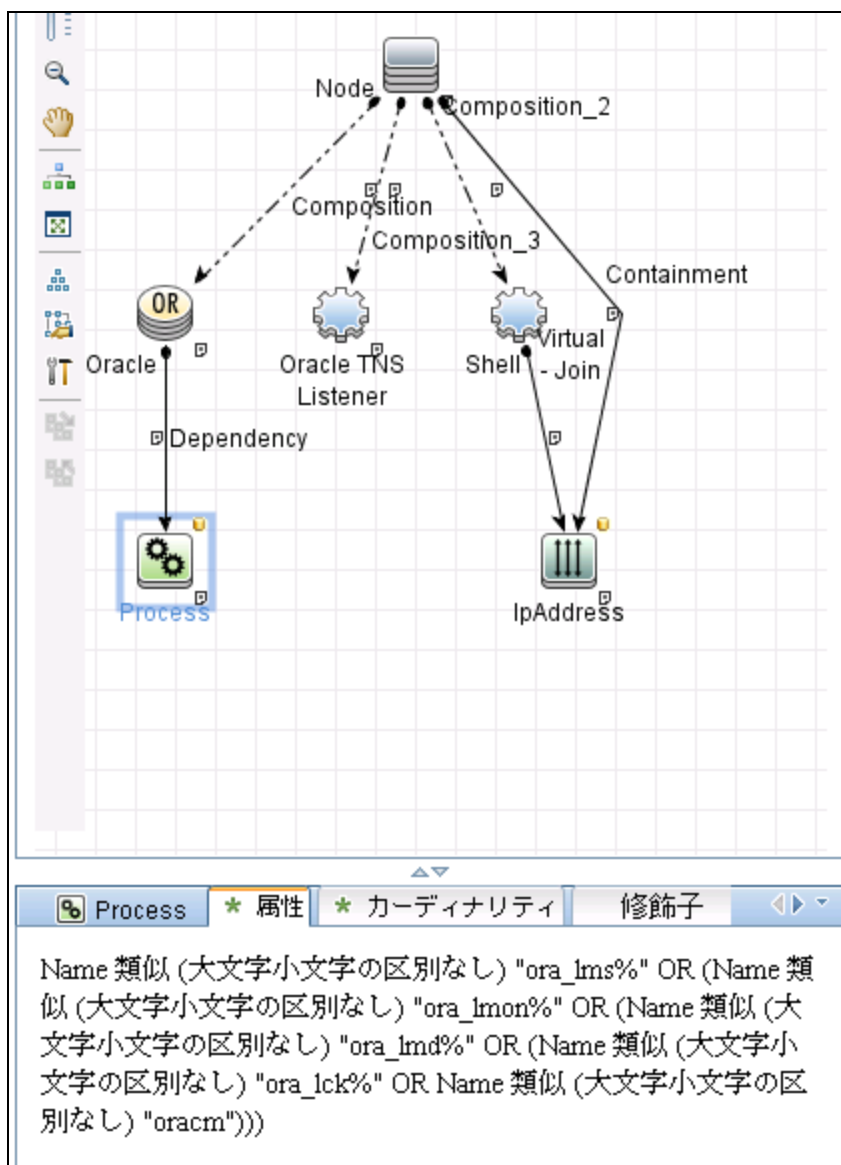
### ディスカバリ・メカニズム

このジョブは、RAC 関連プロセスがある Oracle データベースでトリガされます。このジョブの動作は次のとおりです。

1. シェルを介してリモート・ホストに接続します。
2. **ORACLE\_HOME** 環境変数を確認します。
3. この変数が定義されていない場合、ジョブはジョブ・アダプタから **ORACLE\_HOME** の値(定義されている場合)を取得します。

4. **\$ORACLE\_HOME/network/admin/listener.ora** に格納されている **Oracle TNS リスナ構成ファイル**を読み取り、さらに解析を実行します。
5. このリスナのリスン対象の IP アドレス全体のリストを取得します。
6. **\$ORACLE\_HOME/bin/lsnrctl** ステータスを使用して、リスナのステータスを確認します。
7. 既知のサービスとリスナ・ステータスを出カから取得します。

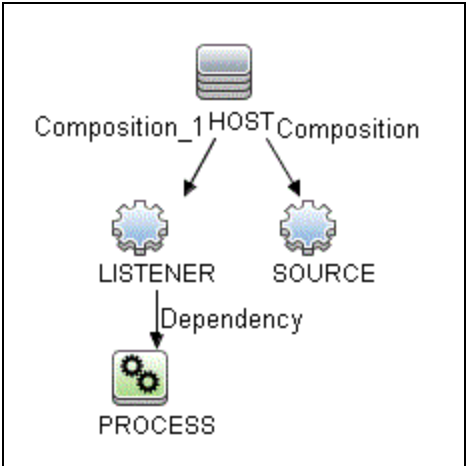
## トリガ・クエリ



アダプタ

このジョブでは、 **Oracle\_Listeners\_by\_Shell** アダプタを使用します。

・ 入力クエリ



・ 使用スクリプト

- oracle\_shell\_utils.py
- oracle\_listeners\_by\_shell.py

・ トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}
listener_names	\${LISTENER.name}
listener_process_path	\${PROCESS.process_path}

・ アダプタ・パラメータ

OracleHomes	ORACLE_HOME 環境変数が定義されていないときに使用されます。
-------------	-------------------------------------

	この値は、Oracle RAC Topology by Shell ジョブのパラメータと同じである必要があります。
--	---

検出された CIT

- Composition
- Containment
- IpAddress
- Node
- Oracle TNS Listener

注: トポロジを表示するには、[「トポロジ」\(501ページ\)](#)を参照してください。

## Oracle RAC Topology by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### ディスカバリ・メカニズム

このジョブは次の処理を行います。

1. シェルを介してリモート・ホストに接続します。
2. **ORACLE\_HOME** 環境変数を確認します。
3. この変数が定義されていない場合、ジョブはジョブ・アダプタからの **OracleHome** の値を使用します。
4. **srvctl config database** コマンドを使用して、構成されたデータベース・サービス名を列挙します。
5. **srvctl status database -d <service\_name>** コマンドを使用して、**sids** およびそのサービスが利用可能なホストを列挙します。
6. Service Name や Nodes などの RAC パラメータを **\$ORACLE\_HOME/network/admin/tnsnames.ora** ファイルから取得します。
7. **lsnrctl status** の出力を解析することで、この RAC インスタンスが実行中かどうかを確認します。

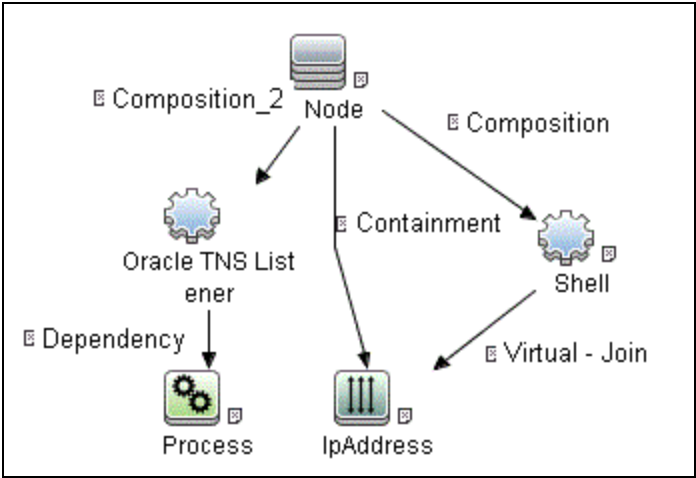
**注:** ノードは内部 IP または内部ドメイン名によって **tnsnames.ora** ファイル内に列挙されています。ドメイン名がある場合、DFM がその名前を解決します。

8. 入力クエリから、クエリと一致するすべてのリスナの Listened IPs 全体のリストを取得します。
9. リッスンされた IP のリストからこの属性の値を解析し、MAC アドレスに対応するホスト・プライマリ・ドメイン名を取得します。この名前が必要なのは、RAC CI の名前キー属性が、コロン記号(:)で区切られたすべてのノード・ドメイン名のリストから構成される必要があるためです。
10. IP アドレスで並べ替えられたビルド・テーブル内の完全なノード名をルック・アップします。結果は各ノードのホスト・プライマリ・ドメイン名です。

この段階で、RAC サービス名、すべての RAC ノードの完全修飾ドメイン名、RAC インスタンス数に関する情報が利用可能になります。

11. RAC CI を作成します。

トリガ・クエリ



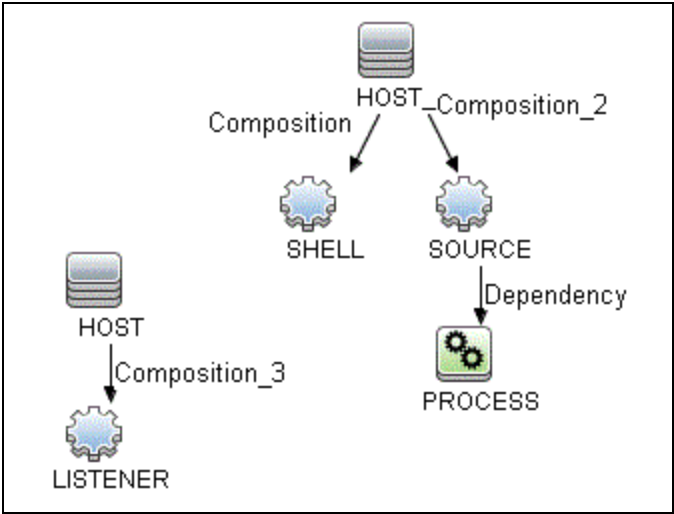
ノード名	条件
Node	なし
Oracle TNS Listener	NOT Listened IPs null
プロセス	なし
IpAddress	NOT IP Probe Name null
Shell	NOT Reference to the credentials dictionary entry null

アダプタ

このジョブでは、Oracle\_RAC\_Topology\_by\_Shell アダプタを使用します。



• 入力クエリ



• トリガ CI データ

名前	値
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
ip_address	\${SHELL.application_ip}
listened_ips	\${LISTENER.listened_ips:NA}
listenerName	\${SOURCE.name}
listener_ip	\${SOURCE.application_ip}
listener_process_path	\${PROCESS.process_path}

• アダプタ・パラメータ

OracleHomes	ORACLE_HOME 環境変数が定義されていないときに使用されます。 この値は、Oracle Listeners by Shell ジョブのパラメータと同じである必要 があります。
-------------	---

検出された CIT

- Composition

- Containment
- IpAddress
- Membership
- Node
- Oracle
- Oracle RAC
- Oracle TNS Listener
- Running Software

注: トポロジを表示するには、[「トポロジ」\(501ページ\)](#)を参照してください。

## 構成アイテム

CI	説明
Oracle TNS Listener	この CIT は Oracle TNS リスナを表します。
CIT 名	oracle_listener
親 CIT 名	application
キー属性	<ul style="list-style-type: none"><li>• <b>name</b>(Name として表示)。TNS Listener 定数です。</li><li>• <b>root_container</b>(Container として表示)。Container CI です。</li><li>• <b>listener_name</b>(Name of the Listener として表示)。TNS Listener の実際の名前です。</li></ul>
追加属性	<p><b>listened_ips</b>(Listened IPs として表示)。リッスン対象の IP アドレスとマシン・ドメイン名です。Listened IPs は、Oracle TNS リスナによりリッスンされる IP アドレスです。</p> <p>形式 :</p> <p>&lt;host_name&gt;:&lt;host_primary_ip&gt;@&lt;listened_ip&gt;:&lt;mac&gt;;...&lt;listened_ip&gt;:&lt;MAC&gt;</p>

CI	説明
	注: MAC アドレスは現在検出されません。このマーカは将来の拡張時のためのプレースホルダとして機能します。

関係

CIT	リンク・タイプ	カーディナリティ
Node	Composition	1.*
RAC	Membership	1.*
Process	Dependency	1.*

# トラブルシューティングおよび制限事項

本項では、Oracle RAC ディスカバリのトラブルシューティングと制限事項について説明します。

エラー・メッセージ	説明
Failed to lookup host name.No RAC CI will be created.	1 つまたは複数のノードについて、ジョブはリスナの <b>listened_ips</b> 属性情報からの FQDN(完全修飾ドメイン名)の取得に失敗しました。 <ul style="list-style-type: none"><li>IP と宛先を取得するため、ログを確認します。</li><li>その IP の FQDN が、DNS またはホスト・ファイルから取得可能であることを確認します。</li></ul>
No RAC CI are retrieved.	すべてのノードが、適切なリスナ情報を使用して検出されたとはかぎりません。
Discovery cannot discover links to the remote machines (database clients)	これは、次の状況で発生する可能性があります。検出されたデータベースが、IP アドレスでなくホスト名でクライアントをレポートし、そのホスト名を IP アドレスとして解決できないことがあります。この場合、リモート・クライアントを作成できません。

## 第31章: Sybase ディスカバリ

### 本章の内容

概要 .....	514
サポートされるバージョン .....	514
トポロジ .....	515
Sybase の検出方法 .....	515
Sybase Database Connection by SQL ジョブ .....	516
Sybase Topology by SQL ジョブ .....	518

## 概要

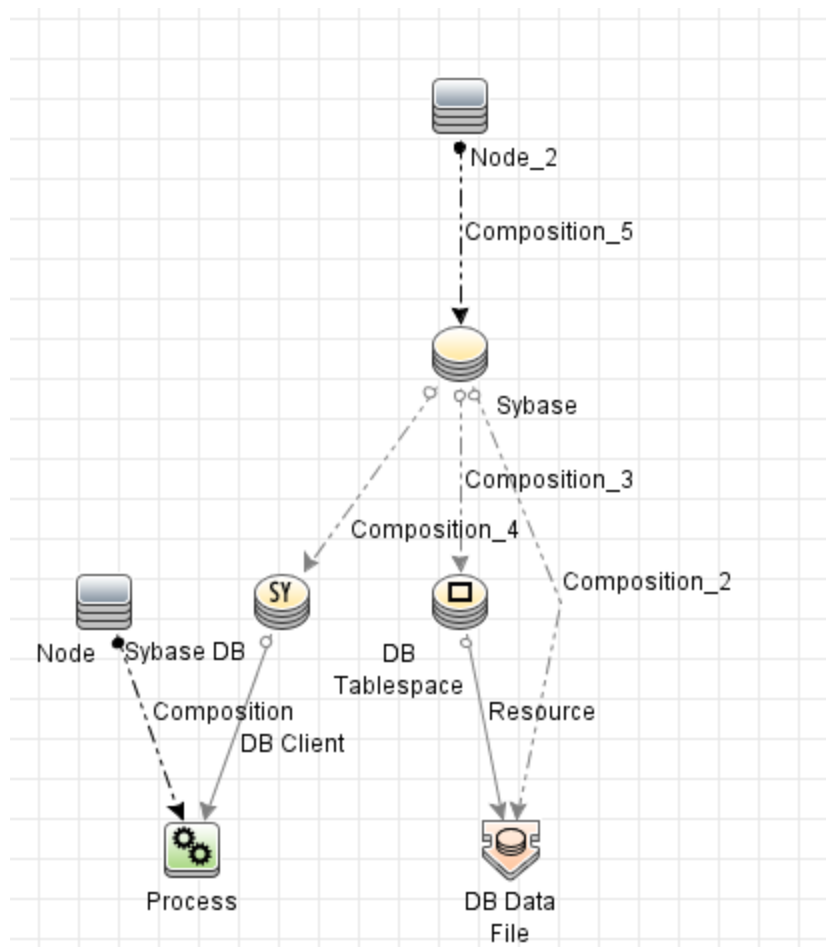
SAP Sybase ASE(Adaptive Server Enterprise)は一般に ASE または単に Sybase DB として知られているリレーショナル・モデルのデータベース・サーバ製品であり, Sybase Corporation(現在は SAP AG が買収)が開発した業務用製品です。ASE は主に UNIX プラットフォームで使用されていますが, Microsoft Windows でも利用できます。

## サポートされるバージョン

このディスカバリは, Sybase ASE の次のバージョンをサポートしています。10.x, 11.x, 12.x, 15.0, 15.5, 16。

## トポロジ

次の図に、Sybase ディスカバリのトポロジを示します。



## Sybase の検出方法

本項では、Sybase が管理するトポロジの検出方法について説明します。

### 前提条件

汎用 DB プロトコル(SQL)プロトコルを設定します。この情報の詳細については、『HP UCMDB Discovery and Integrations Content Guide - Supported Content』で SQL プロトコル資格情報について説明した項を参照してください。

## ディスカバリの実行

次のジョブを実行することによって、Sybase ディスカバリを実行します。

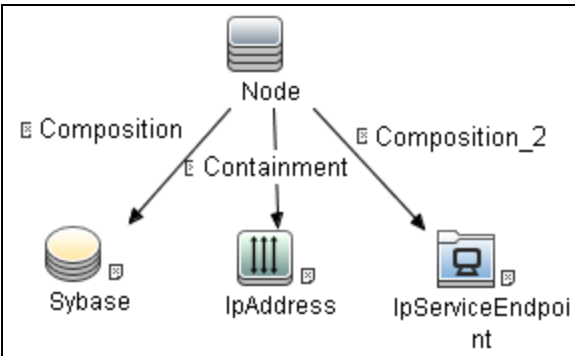
- Databases TCP Ports
- Sybase Database Connection by SQL
- Sybase Topology by SQL

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Sybase Database Connection by SQL ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ TQL



ノード名	条件
Node	なし
Sybase	なし
IPAddress	NOT IP Probe Name null
IpServiceEndpoint	IpServiceName 等しい(大文字小文字の区別なし) sybase OR ServiceNames 含む sybase

### アダプタ

このジョブでは、**Sybase Connection By SQL** アダプタを使用します。



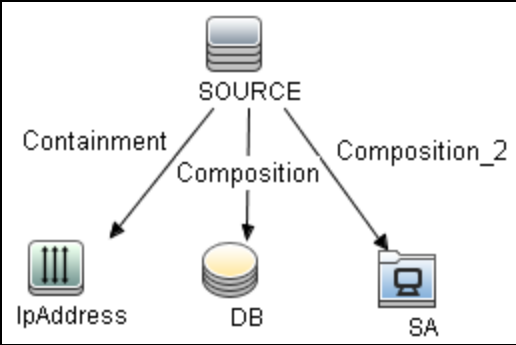
• アダプタ・タイプ

Jython

• 入力 CIT

Node

• 入力 TQL



• トリガ CI データ

名前	値
application_ip	\${DB.application_ip:NA}
application_port	\${DB.application_port:NA}
ip_address	\${IpAddress.name}
sa_ip	\${SA.bound_to_ip_address:NA}
sa_port	\${SA.network_port_number:NA}
sid	\${DB.name:NA}

• 使用スクリプト

- file\_ver\_lib.py
- SQL\_Connection.py

• 必要な権限

権限	操 作	使用状況の詳細	オブジェクトとパラメータ
SQL	select	Check Sybase database version	@@version
SQL	select	sid information	master..sys.servers

• 検出された CIT

- Composition
- Containment
- IPAddress
- IpServiceEndpoint
- Node
- SAP ASE Server
- Usage

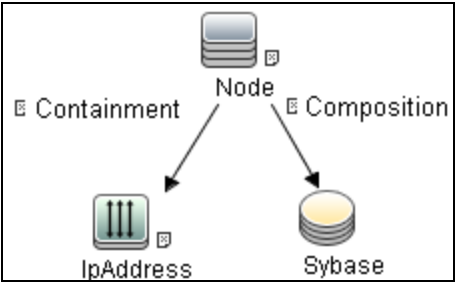
• アダプタ・パラメータ

パラメータ	標準設定	説明
protocolType	Sybase	プロトコル・タイプです。標準設定値は変更しないでください。

## Sybase Topology by SQL ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ TQL



ノード名	条件
Node	なし
IpAddress	NOT IP Probe Name null
Sybase	NOT Application Username null AND NOT Reference to the credentials dictionary entry null

アダプタ

このジョブでは、**Sybase Topology by SQL** アダプタを使用します。

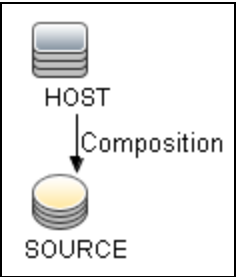
• アダプタ・タイプ

- Jython

• 入力 CIT

Sybase ACE

• 入力 TQL



• トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}
port	\${SOURCE.application_port:NA}

- 使用スクリプト

sybase\_dis\_physical.py

- 必要な権限

権限	操作	使用状況の詳細	オブジェクトとパラメータ
SQL	select	Getting existing schema names	master..spt_values  master..sysdatabases  master.dbo.sysusages
SQL	select	Getting opened db sessions info	master..sysdatabases  master..syslogins  master..sysprocesses
SQL	select	Getting tablespaces info	sybssystemprocs..sp_helpdevice
SQL	select	sid information	master..sys.servers

- 検出された CIT

- Composition
- Containment
- DB Client

- DB Data File
- DB Tablespace
- DB User
- IpAddress
- Node
- Process
- Resource
- SAP ASE Database
- SAP ASE Server

## 第6部: Enterprise Applications

# 第32章: Active Directory ディスカバリ

## 本章の内容

- 概要 ..... 524
- サポートされるバージョン ..... 524
- トポロジ ..... 525
- Active Directory のドメイン・コントローラとトポロジの検出方法 ..... 527
- Active Directory Connection by LDAP ジョブ ..... 528
- Active Directory Topology by LDAP ジョブ ..... 531

## 概要

Active Directory(AD)は、ネットワーク・リソースの効率的な管理を実現する、拡張可能でスケーラブルなディレクトリ・サービスを提供します。

データ・フロー管理は、AD のドメイン・コントローラと通信する LDAP Directory Service Interface を介して Active Directory トポロジを検出します。データ・フロー管理は JNDI を使用して、LDAP Directory Service Interface と対話する API を提供します。

## サポートされるバージョン

このディスカバリ・ソリューションは、次のサーバをサポートしています。

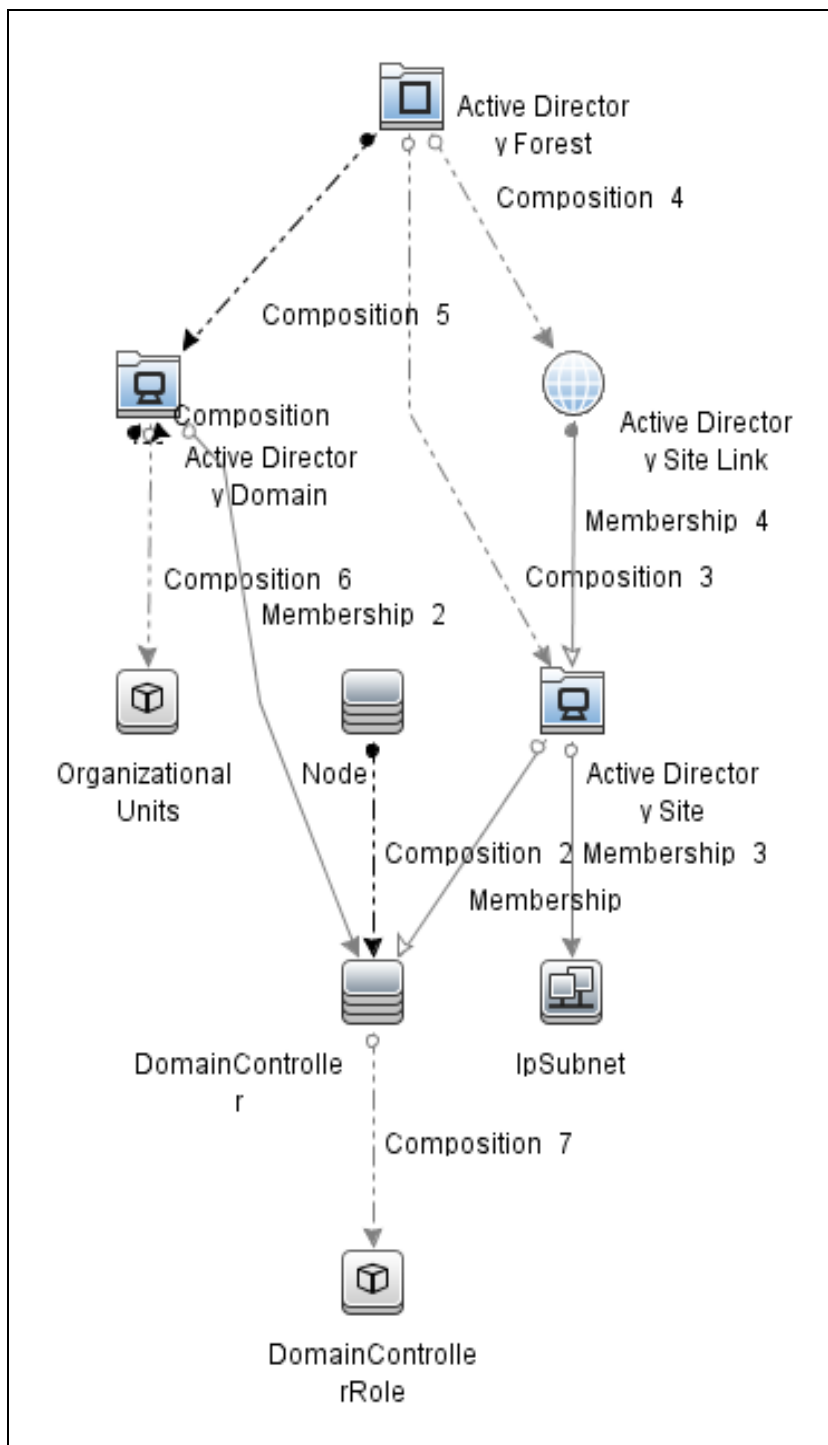
- Windows Server 2000
- Windows Server 2003
- Windows Server 2008
- Windows Server 2012
- Windows Server 2012 R2



## トポロジ

次の図に、AD トポロジを示します。

注：検出された CIT のリストについては、[「検出された CIT」\(533 ページ\)](#)を参照してください。



# Active Directory のドメイン・コントローラとトポロジの検出方法

このタスクでは、Active Directory の検出方法を次の手順で説明します。

## 1. 前提条件 - プロトコル資格情報の設定

- a. ホストを検出するには、SNMP、シェル(NTCMD、SSH、Telnet)、WMI の各プロトコルを設定する必要があります。

- SNMP プロトコル

SNMP プロトコル用の情報として、**コミュニティ名** (v2 プロトコル用)、**ユーザ名** (v3 プロトコル用)、**パスワード** (v3 プロトコル用)を準備します。

- シェル・プロトコル: NTCMD、SSH、Telnet の各プロトコル

シェル・プロトコル用の情報として、**ユーザ名**、**パスワード**、**ドメイン名** (NTCMD では省略可能)を準備します。

- WMI プロトコル

WMI プロトコル用の情報として、**ユーザ名**、**パスワード**、**ドメイン名** (省略可能)を準備します。

- b. すべての AD ジョブを実行するには、LDAP プロトコルを設定する必要があります。利用可能なプロトコルには、**2**と**3**の2つのバージョンがあります。バージョン2の正式な標準化版はないため、データ・フロー管理はバージョン3のプロトコルを使用します。

**注:** ユーザ名 : ドメインが存在する場合は **username@domain** を使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - その他

- a. 各 AD ドメイン・コントローラのホストを検出します。使用しているプロトコルに応じ、次のいずれかのジョブをアクティブ化します。

- **Host Connection by Shell**
  - **Host Connection by SNMP**
  - **Host Connection by WMI**
- b. **portNumberToPortName.xml** 構成ファイルに、可能性のあるすべての AD ポートが含まれていることを確認します。たとえば、AD が LDAP ポート 389 で実行中の場合は、ファイル内の次の行を探します。

```
<portInfo portProtocol="tcp" portNumber="389" portName="ldap" discover="0" />
```

**discover="0"** 属性の値を **discover="1"** に変更します。

詳細については、『HP UCMDB デイスカバリ / インテグレーション・コンテンツ・ガイド - 全般的な参照情報』ドキュメントにある portNumberToPortName.xml ファイルに関する項と新規ポートに関する項を参照してください。

- c. 既知のすべての LDAP ポートを検出するには、TCP Ports ジョブで次の追加構成を実行します。
- **ports** オプションを **ldap** に設定
- d. 各ドメイン・コントローラ・サーバについて、宛先 IP の LDAP ポートを開きます。これは、「**Tools and Samples > Discovery Tools**」のモジュールの次のジョブをアクティブ化することで行います。
- **TCP Ports**。このジョブには **TCP\_NET\_Dis\_Port** アダプタが含まれます。

### 3. デイスカバリの実行

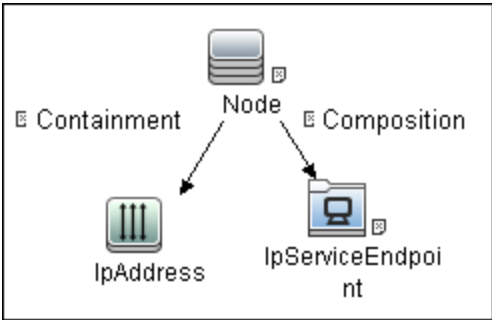
- **Active Directory Connection by LDAP** ジョブをアクティブ化します。このジョブは、LDAP を介して AD ドメイン・コントローラの存在を検出します。クエリとパラメータの詳細については、「[Active Directory Connection by LDAP ジョブ](#)」(528ページ)を参照してください。
- **Active Directory Topology by LDAP** ジョブをアクティブ化します。このジョブは、AD ドメイン・コントローラ・サーバに接続し、そのトポロジを検出します。クエリとパラメータの詳細については、「[Active Directory Topology by LDAP ジョブ](#)」(531ページ)を参照してください。

## Active Directory Connection by LDAP ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ・クエリ

- トリガ CI :IpAddress
- トリガ・クエリ :



- CI 属性条件 :

CI	属性値
Node	なし
IpAddress	NOT IP Probe Name null
IpServiceEndpoint	IpServiceName 等しい (大文字小文字の区別なし) ldap OR ServiceNames 含む ldap

アダプタ

このジョブでは, **LDAP\_Active\_Directory\_Connection** アダプタを使用します。

• トリガ CI データ

名 前	値	説 明
hostId	\${HOST.root_id}	ドメイン・コントローラが配置されているホストの ID
ip_address	\${SOURCE.name}	IpServiceEndpoint から取得した IP アドレス
port_number	\${Service_Address.network_port_number}	IpServiceEndpoint から取得した LDAP ポート番号

• アダプタ・パラメータ

パラメータ	説 明
baseDn	ドメイン・コントローラに関するレコードが保存されているドメインの名 前  標準設定 : OU=Domain Controllers

検出された CIT

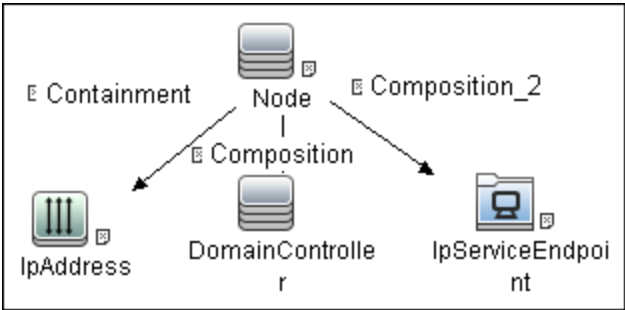
- Containment
- Composition
- DomainController
- Node
- IpAddress

## Active Directory Topology by LDAP ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

- トリガ CI :DomainController
- トリガ・クエリ :



- CI 属性条件 :

CI	属性値
Node	なし
IpAddress	NOT IP Probe Name null
DomainController	NOT Reference to the credentials dictionary entry null AND NOT Application IP null
IpServiceEndpoint	IpServiceName 等しい (大文字小文字の区別なし) ldap OR ServiceNames 含む ldap

### アダプタ

このジョブでは、LDAP\_Active\_Directory\_Topology アダプタを使用します。

• トリガ CI データ

名前	値	説明
<b>application_port</b>	\${SOURCE.application_port:NA}	IpServiceEndpoint から取得されるポート
<b>credentials_id</b>	\${SOURCE.credentials_id}	ドメイン・コントローラの属性に保存されたプロトコルの資格情報 ID
<b>hostId</b>	\${HOST.root_id}	ドメイン・コントローラが配置されているホストの ID
<b>ip_address</b>	\${SOURCE.application_ip}	サーバの IP アドレス
<b>port</b>	\${SERVICE_ADDRESS.network_port_number}	LDAP ポート番号

• アダプタ・パラメータ

パラメータ	説明
<b>baseDn</b>	ドメイン・コントローラに関するレコードが保存されているドメインの名前  標準設定 : OU=Domain Controllers
<b>globalCatalogPort</b>	DFM がグローバル・カタログに指定されているドメイン・コントローラにアクセスするために使用するポート番号です。  標準設定 : 3268  注 : このパラメータは、 <b>tryToDiscoverGlobalCatalog</b> が true に設定されている場合にのみ必要です。
<b>reportOUAsConfigurationDocument</b>	このパラメータが <b>true</b> に設定されている場合、このジョブは組織単位のツリーを構成ドキュメントとしてレポートします。 <b>false</b> の場合、このジョブは組織単位のツリーを CI のツリーとしてレポートします。  標準設定 : false
<b>tryToDiscoverGlobalCatalog</b>	このパラメータが <b>true</b> に設定されている場合、DFM は、グローバル・カタログ・サーバに指定されているドメイン・コントローラに接続することでトポロジ全体の検出を試み



パラメータ	説明
	<p>ます。接続は、<b>globalCatalogPort</b> パラメータで定義されているポートを介して確立されます。</p> <p><b>標準設定</b> : true : ディスカバリにグローバル・カタログを使用</p>

## 検出された CIT

- ActiveDirectoryDomain : AD フォレスト内のドメイン
- Active Directory Forest : 機能レベルと連続名についての情報
- ActiveDirectorySite : AD フォレスト内で設定された使用可能なサイト・オブジェクト
- ActiveDirectorySiteLink
- ActiveDirectorySystem
- Composition
- Containment
- ConfigurationDocument
- DomainController
- DomainControllerRole
- Node
- Membership : サイトとサブネットとの間の関係
- IpSubnet : 利用可能なサブネット・オブジェクト
- IpAddress

注: トポロジを表示するには、[「トポロジ」\(525ページ\)](#)を参照してください。

# 第33章: Cisco UCS

## 本章の内容

概要 .....	535
Cisco UCS Connection ジョブ .....	535
Cisco UCS Topology ジョブ .....	536
Cisco UCS Manual ジョブ .....	538

## 概要

Cisco UCS は、データセンタ内のハードウェアとソフトウェアを管理します。この統合ソリューションは、独自の XML API に基づいて管理対象トポロジを検出します。このソリューションには 3 つのジョブがあります。

## Cisco UCS Connection ジョブ

このジョブは、Cisco UCS Topology ジョブのトリガとなる実行中ソフトウェアとして Cisco UCS を検出するために使用されます。

## バージョン

このパッケージは、Cisco UCS バージョン 2.2(1b)をサポートしています。

## 前提条件

このジョブには、ユーザ名とパスワードを含む UCS 資格情報が必要です。

## アダプタ情報

このジョブでは、**Cisco\_UCS\_Connection** アダプタを使用します。

### 入力 CIT

ip\_address

### トリガ TQL



IpAddress

## 使用スクリプト

- ucs\_connection\_main.py
- ucs\_connection\_data\_manager.py
- ucs\_client.py
- ucs\_base.py
- ucs\_decorators.py

## 検出された CIT

- RunningSoftware

## グローバル構成ファイル

- cisco\_ucs/ucs\_mapping.xml

## ワークフロー

1. トリガ IP の資格情報を取得します。
2. 資格情報を反復し、UCS クライアントを作成してログインします。接続に成功したら、接続された資格情報とともに UCS の実行中ソフトウェアをレポートします。

# Cisco UCS Topology ジョブ

このジョブは、入力トリガとして UCS が実行されている Cisco UCS トポロジを検出するために使用されます。

## バージョン

このパッケージは、Cisco UCS バージョン 2.2(1b)をサポートしています。

## 前提条件

このジョブには、ユーザ名とパスワードを含む UCS 資格情報が必要です。

## アダプタ情報

このジョブでは, **Cisco\_UCS\_Topology** アダプタを使用します。

### 入力 CIT

ip\_address

### トリガ TQL



### 使用スクリプト

- ucs\_topology\_main.py
- ucs\_pull\_base.py
- ucs\_connection\_data\_manager.py
- ucs\_client.py
- ucs\_base.py
- ucs\_mapping\_file\_manager.py
- ucs\_mapping\_implementation.py
- ucs\_mapping\_interfaces.py
- ucs\_validators.py
- ucs\_decorators.py

## 検出された CIT

- Backbone
- Chassis
- Composition
- Containment
- Fibre Channel HBA
- Fibre Channel Port
- Fibre Channel Switch
- HardwareBoard
- Membership
- Node
- PhysicalPort
- Rack

## グローバル構成ファイル

- cisco\_ucs/ucs\_mapping.xml

## ワークフロー

1. UCS の実行中ソフトウェアから資格情報を取得します。
2. 資格情報を使用して UCS システムに接続し、独自の XML API を使用して UCS からデータを取得します。
3. UCS から取得したデータを解析して UCMDB にマッピングし、CI を UCMDB にレポートします。

## Cisco UCS Manual ジョブ

このジョブは、URL 行を含む構成ファイルによって Cisco UCS トポロジを検出するために使用されます。

## バージョン

このパッケージは、Cisco UCS バージョン 2.2(1b)をサポートしています。

## 前提条件

このジョブには、ユーザ名 とパスワードを含む UCS 資格情報が必要です。

## アダプタ情報

このジョブでは **Cisco\_UCS\_Manual** アダプタを使用します。

## 入力 CIT

discoveryprobegateway

## トリガ TQL



## 使用スクリプト

- ucs\_manual\_main.py
- ucs\_pull\_base.py
- ucs\_connection\_data\_manager.py
- ucs\_client.py
- ucs\_base.py

- ucs\_mapping\_file\_manager.py
- ucs\_mapping\_implementation.py
- ucs\_mapping\_interfaces.py
- ucs\_validators.py
- ucs\_decorators.py

## 検出された CIT

- Backbone
- Chassis
- Composition
- Containment
- Fibre Channel HBA
- Fibre Channel Port
- Fibre Channel Switch
- HardwareBoard
- Membership
- Node
- PhysicalPort
- Rack

## グローバル構成ファイル

- cisco\_ucs/ucs\_mapping.xml



## ワークフロー

1. ucs\_url\_list.conf を読み取り, URL のリストを解析します。
2. URL を 1 つ選択し, 接続されるまですべての UCS 資格情報を 1 つずつ試します。その後, UCS の XML API を使用して UCS からデータを取得します。
3. UCS から取得したデータを解析して UCMDB にマッピングし, CI を UCMDB にレポートします。
4. 残りの URL についても手順 2~3 を繰り返します。

## トラブルシューティング

**問題 :** 通信ログに次のエラー・メッセージが表示されます。URLError: <urlopen error (-1, 'SSL handshake exception

**解決策 1 :** Https サーバに有効な SSL 証明書がありません(自己署名された証明書や期限切れの証明書などが使用されています)。このような場合は, 現在使用中の UCS 資格情報にある 'Trust All SSL Certificates' オプションを有効にします。

**解決策 2 :** サーバが Https サーバではないのに, 資格情報で SSL が有効になっています。この場合は, 資格情報の 'Use HTTPS for connection' オプションを無効にします。

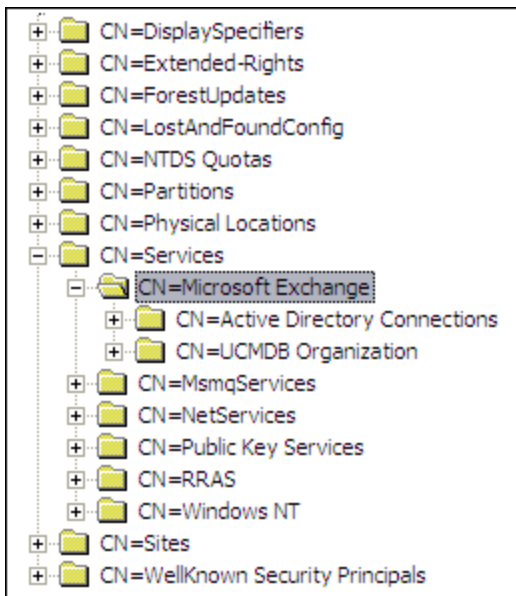
## 第34章: Microsoft Exchange Server with Active Directory ディスカバリ

### 本章の内容

概要 .....	543
サポートされるバージョン .....	544
トポロジ .....	545
Active Directory を使用した Microsoft Exchange Server トポロジの検出方法 .....	549
Microsoft Exchange Topology by LDAP ジョブ .....	550
トラブルシューティングおよび制限事項 .....	553

## 概要

コンテンツ・パック 5 に LDAP プロトコルのサポートが追加されたことで、DFM は Active Directory(AD)を使用して Exchange トポロジを検出できるようになりました。Exchange は AD と緊密に統合されており、その構成内容のほとんどを格納しているため、DFM は AD ドメイン・コントローラに接続してそこから情報を抽出します。Exchange 構成は、次のように Services の下にある特定のノードに格納されています。



このノードの基本識別名は次のとおりです。

**"CN=Microsoft Exchange, CN=Services, CN=Configuration, DC=ucmdb-ex, DC=dot"**

**ucmdb-ex.dot** は、この例に示すドメインの名前です。

このノードが存在する場合、DFM はドリル・ダウンして残りのすべての情報を検出します。たとえば、Exchange 組織、Exchange Server、管理グループ、ルーティング・グループ、コネクタ、ロールなどです。

複数のドメイン・コントローラが同じドメインを扱うことができます。その場合、情報はそれらのドメイン・コントローラ間でレプリケートされます(マルチマスタ・レプリケーション)。これらのコントローラは同じデータを持つため、DFM はそのうちの 1 つに対して実行するだけで十分です。

注: AD デイスカバリのジョブは、検出されたすべてのドメイン・コントローラを対象としてトリガされ、実行されます。ただし、Data Flow Probe の結果処理メカニズムにより、CMDB に送信されるのは更新内容のみであるため、情報は一度しかレポートされません。

ドメイン内の AD マシンは、AD 向けに構成されているように DNS で登録されます。DFM は、各 Exchange デイスカバリから FQDN(完全修飾ドメイン名)を取得します。これは AD での Exchange の名前です。このような Exchange をレポートするため、DFM は次のように、IP アドレスへの FQDN の解決を試みます。

- DFM は、標準設定された Data Flow Probe の DNS を使用して Exchange FQDN を解決します。
- これに失敗すると、DFM はターゲットのドメイン・コントローラを DNS として使用します。なぜなら、多くの場合 DNS サーバはドメイン・コントローラと同じサーバ上で実行されるからです。DFM は Data Flow Probe のローカル・シェルでコマンド **"nslookup <FQDN> <ターゲット・ドメイン・コントローラ>"** を実行します。
- これに失敗すると、DFM はこの Exchange インスタンスをスキップします。

**注:** ローカルの DNS でも、ターゲットのドメイン・コントローラを DNS として使用しても FQDN を解決できない場合、ジョブは次のメッセージを表示します。

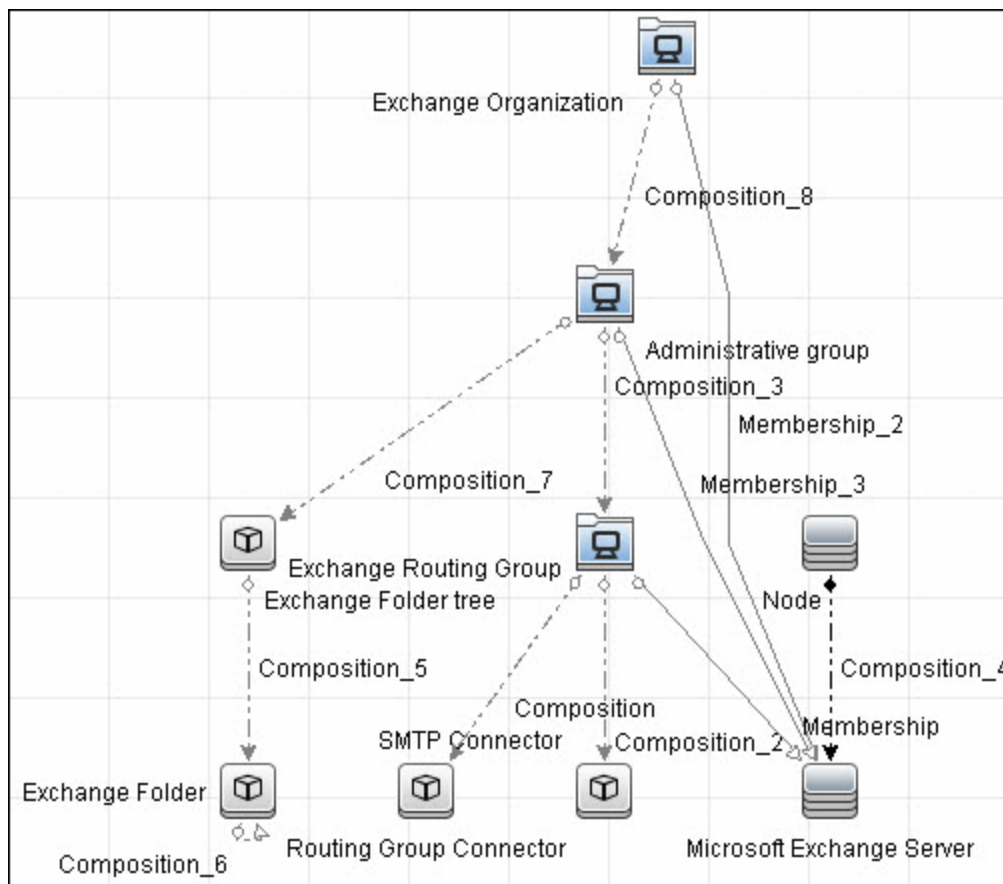
```
Cannot resolve IP address for host '<ホスト>', Exchange Server won't be reported
```

## サポートされるバージョン

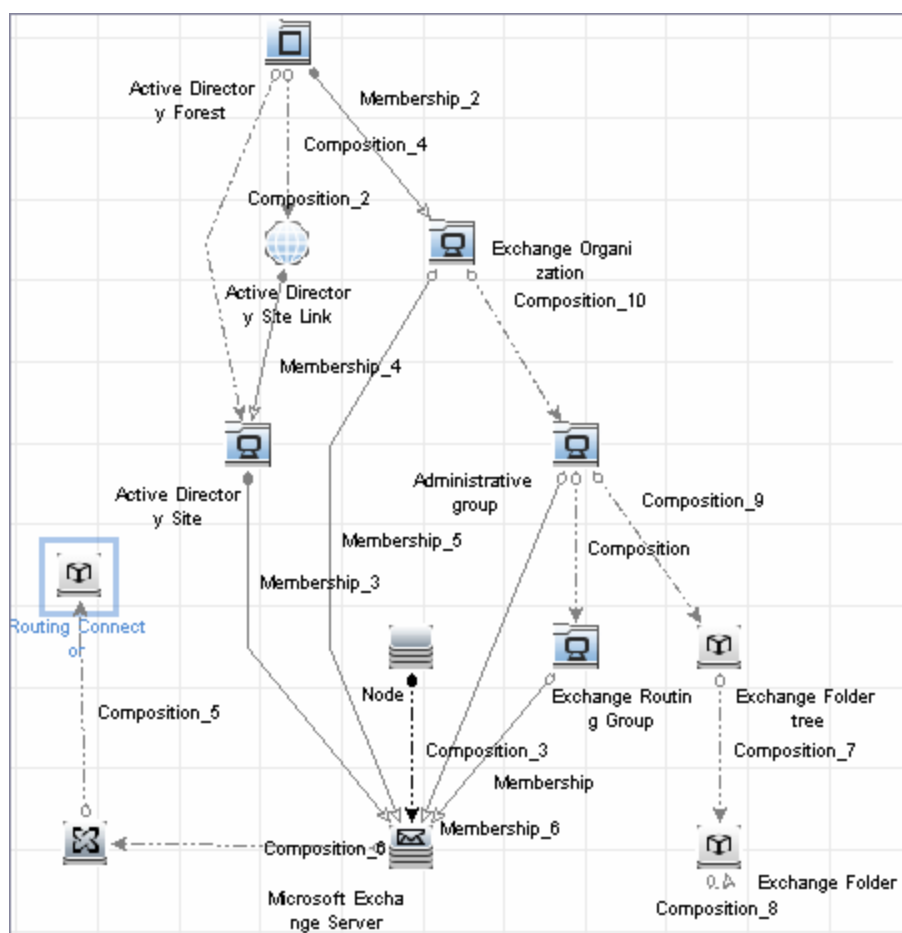
Active Directory を使用した Microsoft Exchange デイスカバリは、MS Exchange 2003, 2007, 2010, 2013 の各バージョンをサポートしています。

## トポロジ

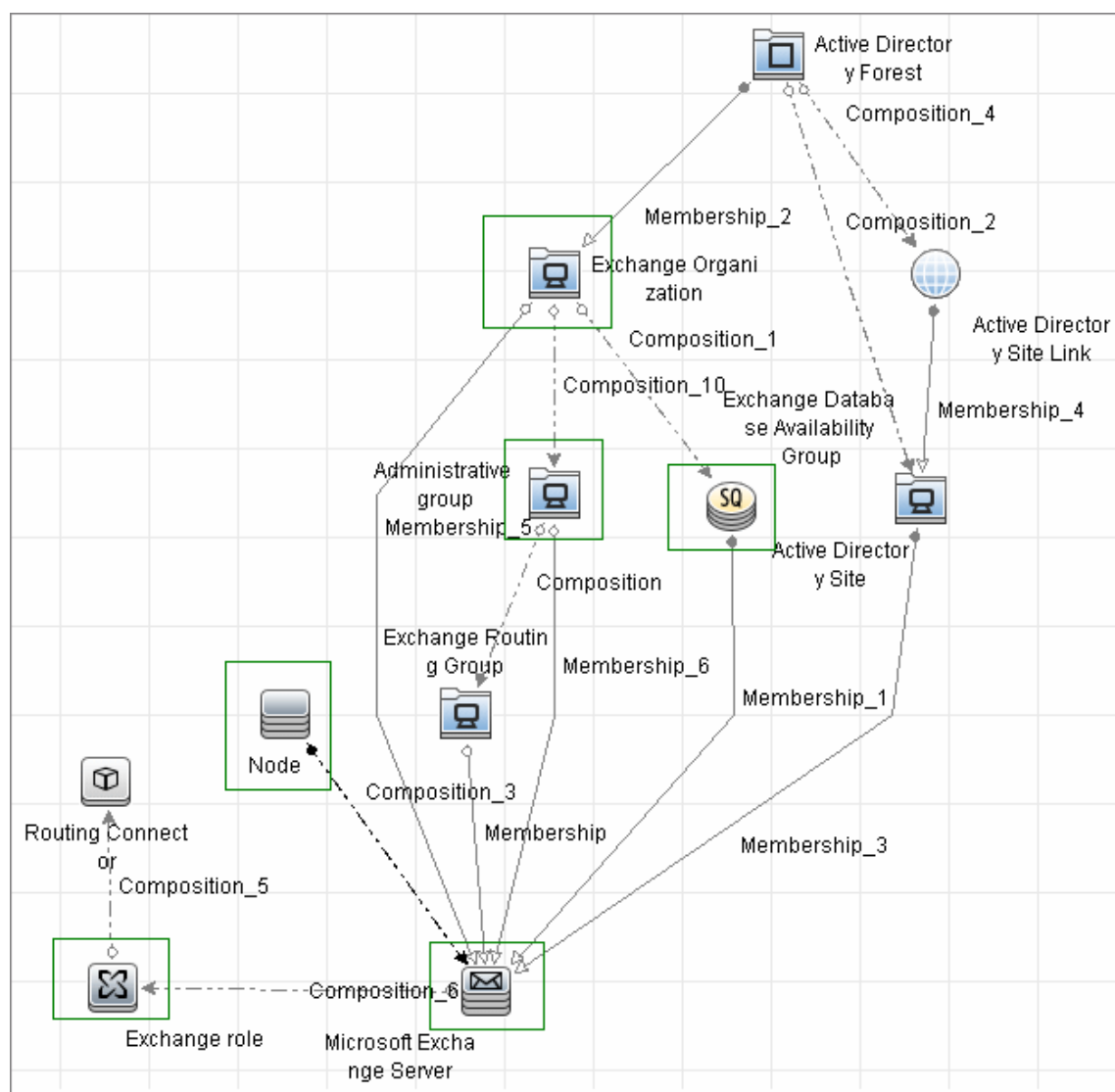
### • Microsoft Exchange Server 2003



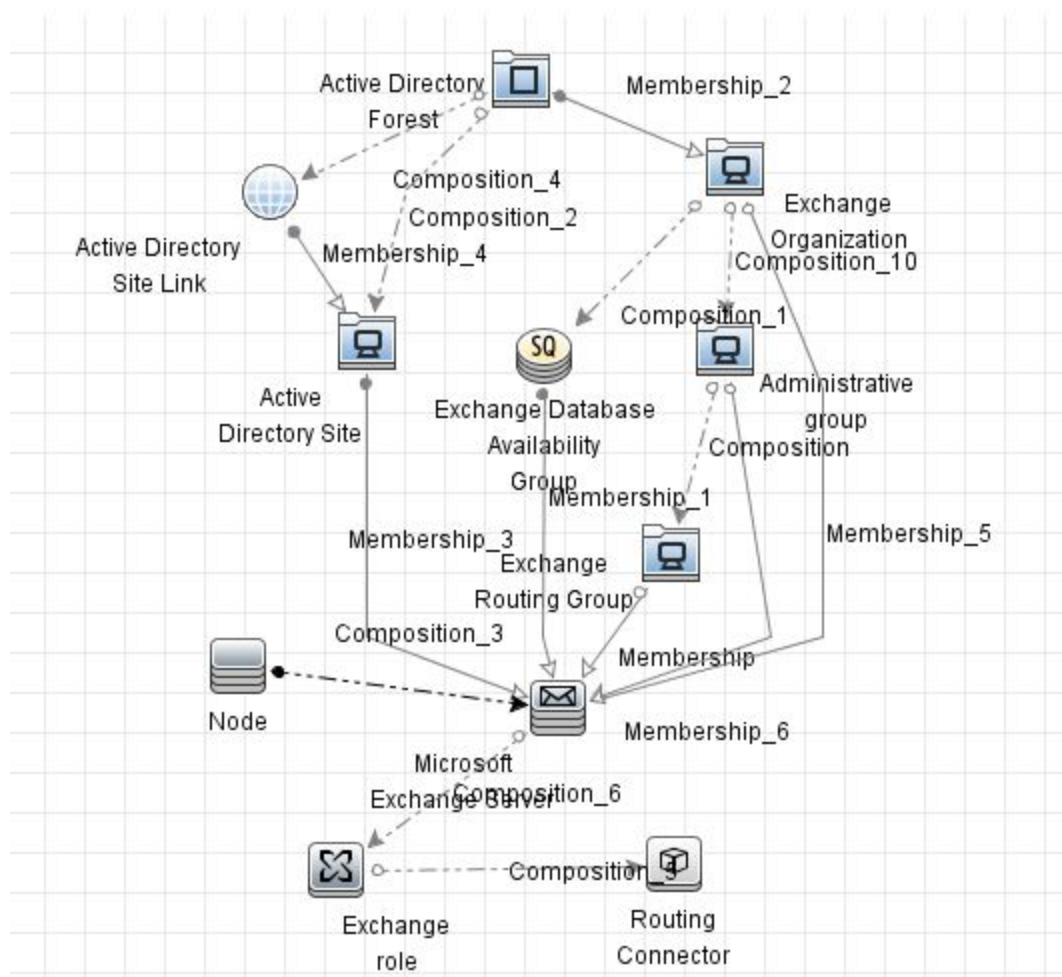
# • Microsoft Exchange Server 2007



- **Microsoft Exchange Server 2010**



- **Microsoft Exchange Server 2013**





# Active Directory を使用した Microsoft Exchange Server トポロジの検出方法

注: この機能は、コンテンツ・パック 5.00 以降で使用できます。

本項では、DFM が Exchange と AD の緊密な統合を利用してどのように Exchange を検出するかを説明します。DFM はジョブを実行することで、AD を介してのみ使用可能なトポロジ内の Exchange 要素を検出します。

本項の内容

## 1. 前提条件 - プロトコル資格情報の設定

1 つ以上の LDAP プロトコル資格情報のセットを定義します。これらの資格情報により、LDAP プロトコルを介したドメイン・コントローラへの接続と、検索の実行が可能になります。DFM は AD 内の情報を変更しません。クエリされたノードは、次のノードの下の構成パーティションにあります。

- **CN=Services,CN=Microsoft Exchange** ノード
- **CN=Sites** ノード

LDAP プロトコル資格情報には次が含まれます。

- **ユーザ名とパスワード**。ターゲット・ドメインからのユーザ・アカウントを使用します。クエリ対象のすべてのノードについて、**コンテンツの一覧表示権限**と、**すべてのプロパティの読み取り権限**を指定します。
- **認証タイプ: 簡易**

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - ドメイン・コントローラの検出

AD を使用した Exchange トポロジを検出するために、DFM はまず、使用可能な LDAP 接続によるドメイン・コントローラの検索を行う必要があります。

- a. **Range IPs by ICMP** ジョブをアクティブ化して、ドメイン・コントローラを実行するターゲット・ホストに対して ping を実行します。

- b. ターゲット・ホストに対して **TCP Ports** ジョブをアクティブ化し、開いている LDAP ポートを検出します。
- c. **Active Directory Connection by LDAP** ジョブをアクティブ化して、ターゲット・ホスト上のドメイン・コントローラを検出します。
- d. DFM による LDAP プロトコルの使用を有効化するには、**[アダプタ管理] > [リソース] 表示枠 > [パッケージ] > [DDMInfra] > [構成ファイル]**にある **portNumberToPortName.xml** ファイルで、次の行を編集します。

変更前の行：

```
<portInfo portProtocol="tcp" portNumber="389" portName="ldap" discover="0" />
```

次のように変更します。

```
<portInfo portProtocol="tcp" portNumber="389" portName="ldap" discover="1" />
```

### 3. ディスカバリの実行

**Microsoft Exchange Topology by LDAP** ジョブをアクティブ化します。

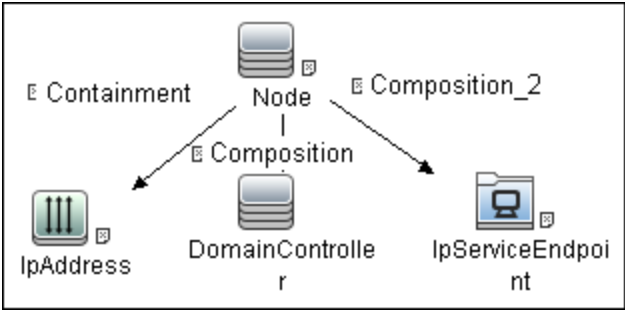
## Microsoft Exchange Topology by LDAP ジョブ

Active Directory を使用して Microsoft Exchange Server を検出するコンポーネントは、Microsoft Exchange Server パッケージ(**Microsft\_exchange\_server.zip**)にバンドルされています。

### トリガ・クエリ

- トリガ CI :DomainController
- トリガ・クエリ：

トリガ・クエリ **trigger\_domainctl\_ldap** は、Active Directory パッケージの一部です。



- CI 属性条件 :

CI	属性値
Node	なし
IpAddress	NOT IP Probe Name null
DomainController	NOT Reference to the credentials entry dictionary null AND NOT Application IP null
IpServiceEndpoint	IpServiceName 等しい (大文字小文字の区別なし) ldap OR ServiceNames 含む ldap

アダプタ

このディスカバリでは、ms\_exchange\_topology\_by\_ldap アダプタを使用します。

- 作成または変更された CIT

追加された CIT	<p>Microsoft Exchange Server パッケージには次の CIT が追加されています。</p> <ul style="list-style-type: none"> <li>◦ Routing Group Connector</li> <li>◦ SMTP Connector</li> <li>◦ Exchange Routing Connector</li> <li>◦ Send Connector</li> <li>◦ Receive Connector</li> <li>◦ Exchange Storage Group</li> <li>◦ Exchange Mailbox Database</li> <li>◦ Exchange Routing group</li> </ul>
廃止された CIT	<p>次の CIT は廃止されました。これらはパッケージ内には残っていますがレポートされません。</p> <ul style="list-style-type: none"> <li>◦ Directory Service Access DC</li> <li>◦ Exchange Message queue</li> <li>◦ Exchange link</li> </ul>
変更された CIT	<p>次の CIT は変更されました。</p> <ul style="list-style-type: none"> <li>◦ <b>Exchange System</b> を <b>Exchange Organization</b> に変更</li> <li>◦ <b>Microsoft Exchange Server</b> に新しい属性を追加 : <b>is_master</b></li> </ul>

## 検出された CIT

- Active Directory Forest
- ActiveDirectorySite
- ActiveDirectorySystem

- Administrative Group
- Composition
- Containment
- Exchange Database Availability Group
- Exchange Folder
- Exchange Folder Tree
- Exchange Mailbox Database
- Exchange Organization
- Exchange Role
- Exchange Routing Connector
- Exchange Routing Group
- ExecutionEnvironment
- IpAddress
- Membership
- MicrosoftExchangeServer
- Node
- Ownership
- Routing Group Connector
- SMTP Connector

## トラブルシューティングおよび制限事項

現在、Exchange フォルダは **Microsoft Exchange Topology by LDAP** ジョブではレポートされません。

## 第35章: NTCMD または UDA による Microsoft Exchange Server ディスカバリ

### 本章の内容

概要 .....	555
サポートされるバージョン .....	555
トポロジ .....	555
NTCMD または UDA による Microsoft Exchange Server の検出方法 .....	559
Microsoft Exchange Connection by NTCMD or UDA ジョブ .....	560
Microsoft Exchange Topology by NTCMD or UDA ジョブ .....	561
作成または変更された CIT .....	563

## 概要

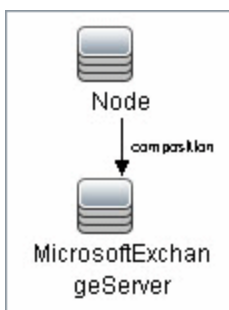
DFM は、Microsoft Exchange Server(Exchange)ソフトウェアのコンポーネントのうち、Microsoft Exchange Server、サーバ・ロール、管理グループ、ルーティング・グループ、組織、クラスタ化されたメール・ボックス、データベース可用性グループ、パブリック・フォルダ、フォルダ・ツリーを検出します。

## サポートされるバージョン

NTCMD または UDA による Microsoft Exchange Server ディスカバリは、MS Exchange Server 2007, 2010, 2013 の各バージョンをサポートしています。

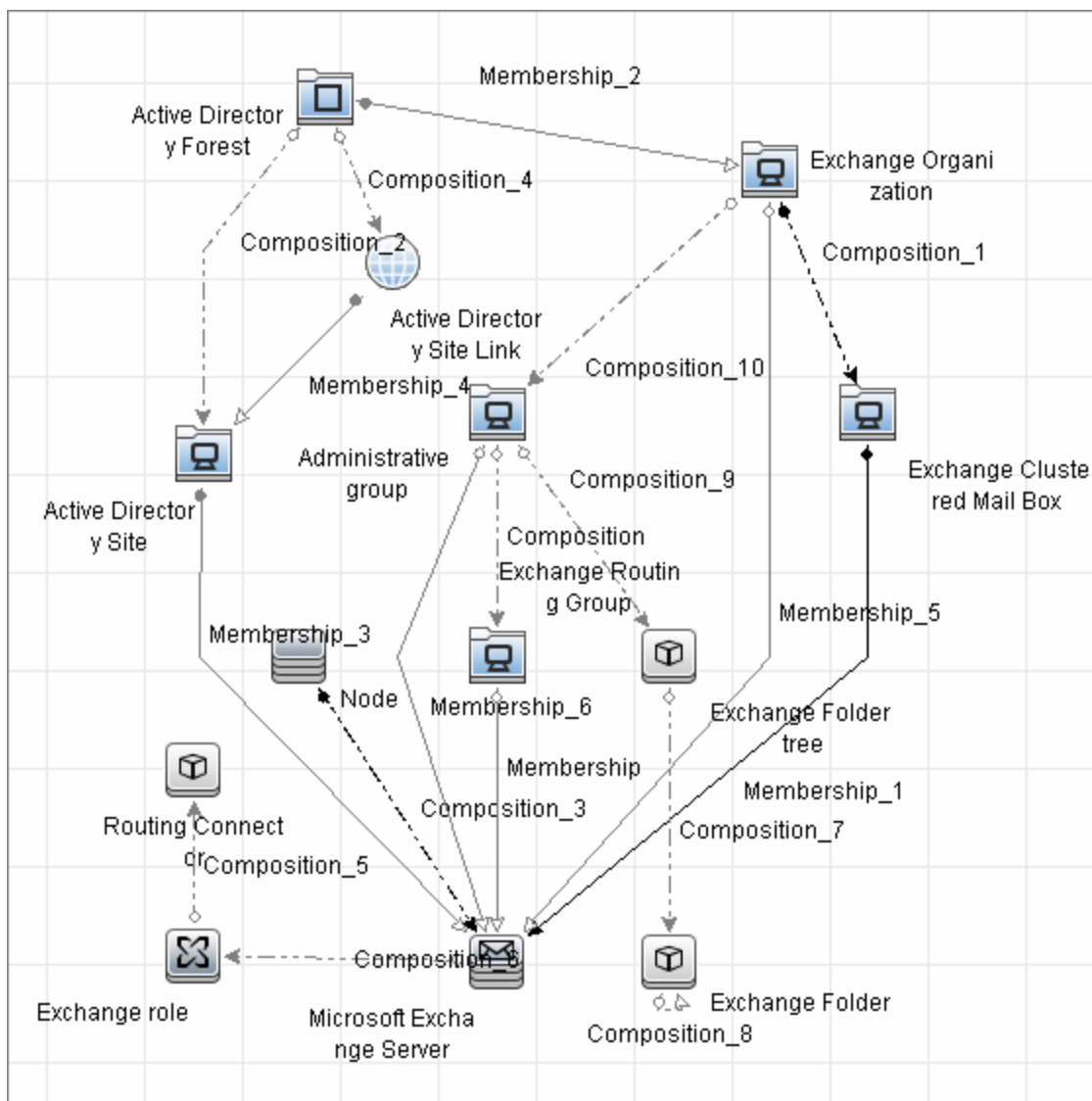
## トポロジ

NTCMD または UDA による MS Exchange 接続:



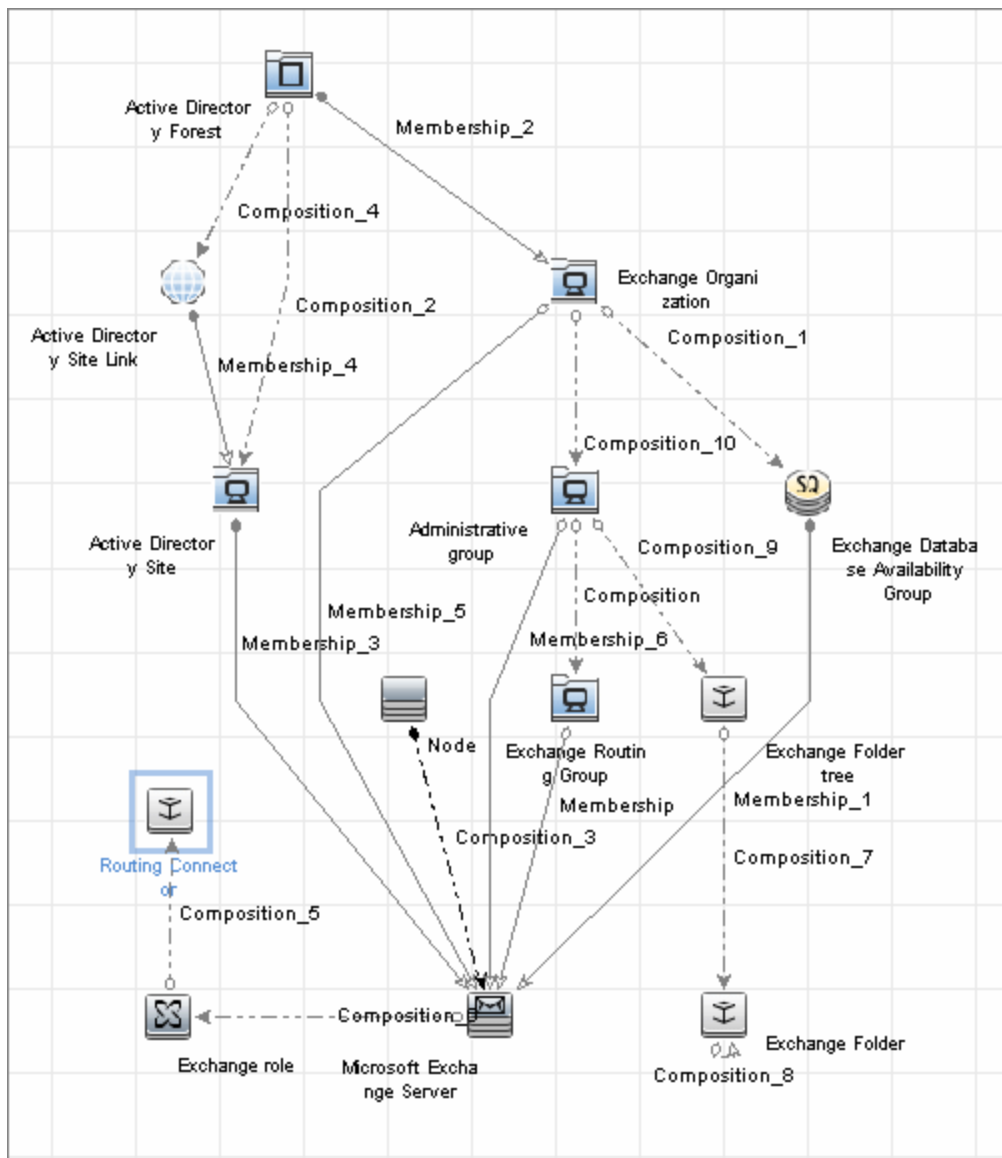
### MS Exchange 2007 トポロジ:

DFM は、NTCMD プロトコルを実行して MS Exchange 2007 のトポロジを取得します。

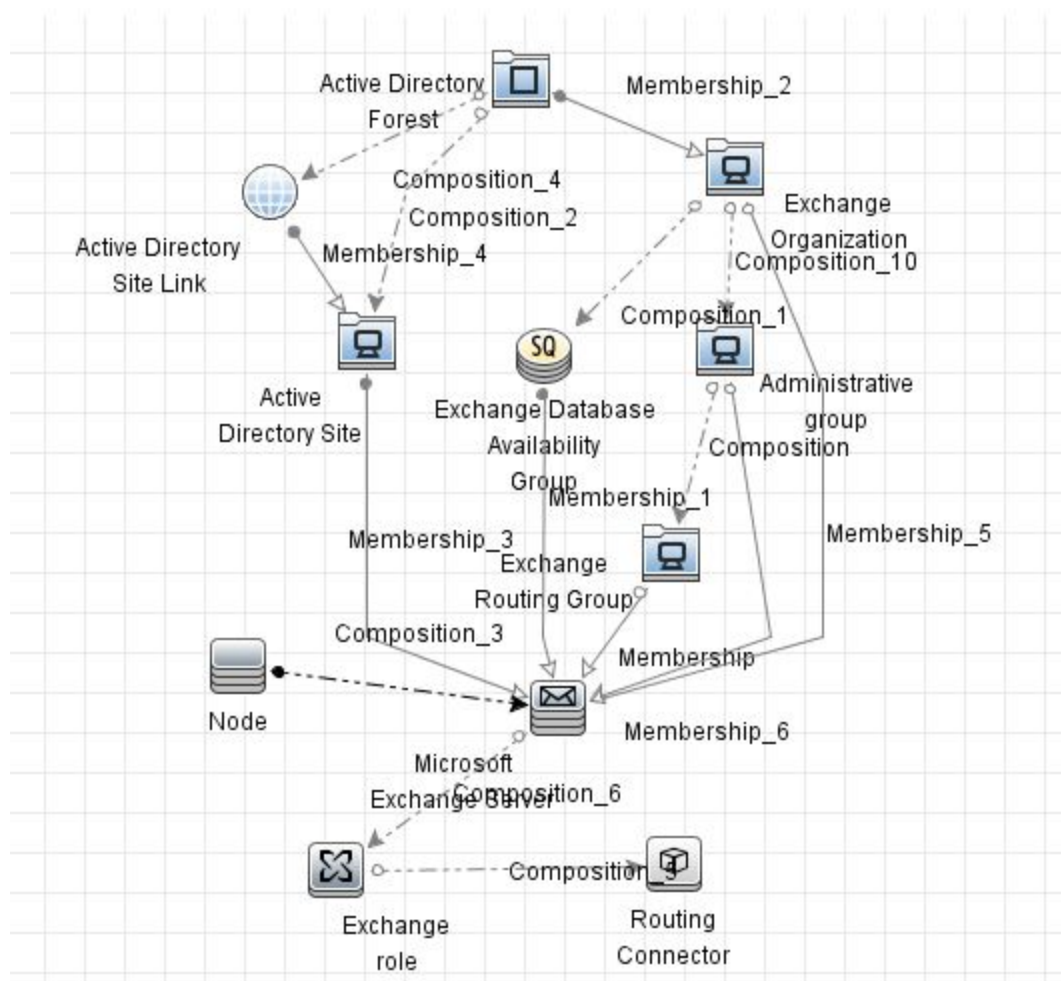




### MS Exchange 2010 トポロジ :



## Microsoft Exchange Server 2013



## NTCMD または UDA による Microsoft Exchange Server の検出方法

DFM は、Exchange がインストールされたリモート・マシンで PowerShell スクリプトを実行することにより、Exchange を検出します。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリは、次のプロトコルに基づいています。

- NTCMD プロトコル、または Universal Discovery プロトコル(UD エージェントが Exchange Server にインストールされている場合)

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - 権限の設定

- スクリプトの実行ポリシーを**無制限**または**リモート署名**に設定します。
- ディスカバリに使用するアカウントに **Exchange の表示専用管理者** ロールの権限があることを確認します。

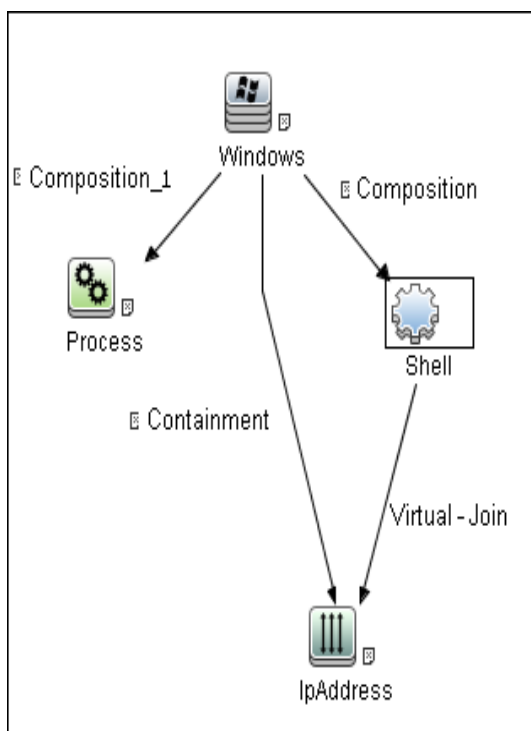
### 3. ディスカバリの実行

- a. **Host Connection by Shell** ジョブを実行します。
- b. **Host Applications by Shell** ジョブを実行して Exchange プロセスを検出します。
- c. **Microsoft Exchange Connection by NTCMD or UDA** ジョブを実行して、Exchange Server CI を検出します。
- d. **Microsoft Exchange Topology by NTCMD or UDA** ジョブを実行して残りのトポロジを検出します。

## Microsoft Exchange Connection by NTCMD or UDA ジョブ

本項ではこのジョブに関する詳細を説明します。

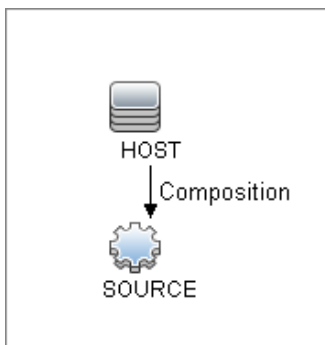
### トリガ・クエリ



### アダプタ

このジョブでは、**ms\_exchange\_connection\_by\_ntcmd** アダプタを使用します。

- 入力クエリ:



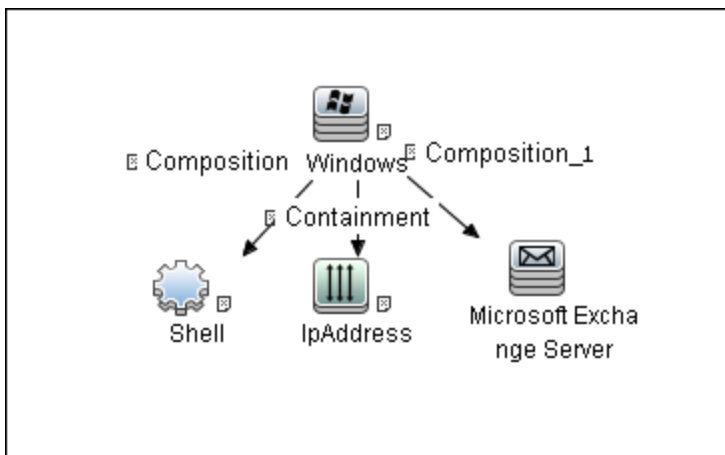
### 検出された CIT

- Composition
- MicrosoftExchangeServer
- Node

## Microsoft Exchange Topology by NTCMD or UDA ジョブ

本項ではこのジョブに関する詳細を説明します。

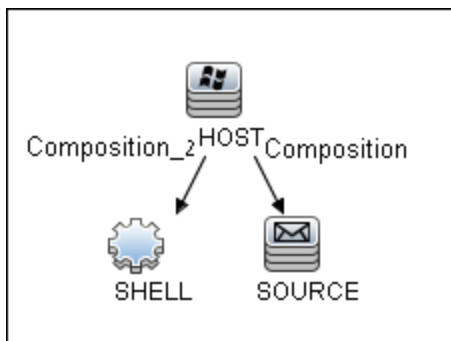
## トリガ・クエリ



## アダプタ

このジョブでは、**ms\_exchange\_topology\_by\_ntcmd** アダプタを使用します。

- 入力クエリ:



## 検出された CIT

- Administrative group
- Composition
- Exchange Client Access Server
- Exchange Clustered Mail Box

- Exchange Database Availability Group
- Exchange Edge Server
- Exchange Hub Server
- Exchange Mail Server
- Exchange Organization
- Exchange Unified Messaging Server
- Membership
- MicrosoftExchangeServer
- Node

## 作成または変更された CIT

Exchange コンポーネント用の CI を作成するため、次の CIT が使用されます。

<b>Exchange Organization</b>	この CIT は、最上位の Exchange システムを表します。たとえば、ある組織が Exchange ソリューションを使用している場合、すべての Exchange コンポーネントは 1 つの Exchange Organization CI にリンクされます。
<b>Microsoft Exchange Server</b>	この CIT は RunningSoftware CIT から継承されます。この CIT は、ホストにインストールされた Exchange ソフトウェアを表します。
<b>Exchange Folder</b>	この CIT は、Exchange システムで使用可能なパブリック・フォルダを表します。パブリック・フォルダは階層構造に編成できます。つまり、1 つのパブリック・フォルダに別のパブリック・フォルダを含めることができます。
<b>Exchange Role</b>	<p>この CIT は <b>Application Resource &gt; Microsoft Exchange Resource</b> フォルダにあります。これは抽象 CIT で、次の CIT の親です。</p> <ul style="list-style-type: none"><li>• <b>Exchange Client Access Server</b> : クライアント・アクセス・サーバのロールを表します。</li><li>• <b>Exchange Mail Server</b> : メール・サーバのロールを表します。</li><li>• <b>Exchange Edge Server</b> : エッジ・サーバのロールを表します。</li><li>• <b>Exchange Hub Server</b> : ハブ・サーバのロールを表します。</li></ul>

	<ul style="list-style-type: none"><li>• <b>Exchange Unified Messaging server</b> : ユニファイド・メッセージング・サーバのロールを表します。</li></ul>
--	---



## 第36章: Microsoft Exchange Server by PowerShell ディスカバリ

### 本章の内容

概要 .....	566
サポートされるバージョン .....	566
トポロジ .....	566
PowerShell を使用した Microsoft Exchange の検出方法 .....	570
PowerShell のリモート処理の構成方法 .....	571
Active Directory 側の構成方法 .....	572
Microsoft Exchange Topology by PowerShell ジョブ .....	574
トラブルシューティングおよび制限事項 .....	580

## 概要

Microsoft Exchange Server は、クライアント / サーバ関係におけるサーバ側に相当する、Microsoft が開発したコラボレーション・アプリケーション製品です。これは Microsoft Server 系のサーバ製品であり、Microsoft のインフラストラクチャ製品を使用する企業によって使用されています。Exchange の主な機能には、電子メール、カレンダー、連絡先、タスクのほか、モバイルや Web ベースでの情報アクセスのサポート、データ・ストレージのサポートがあります。

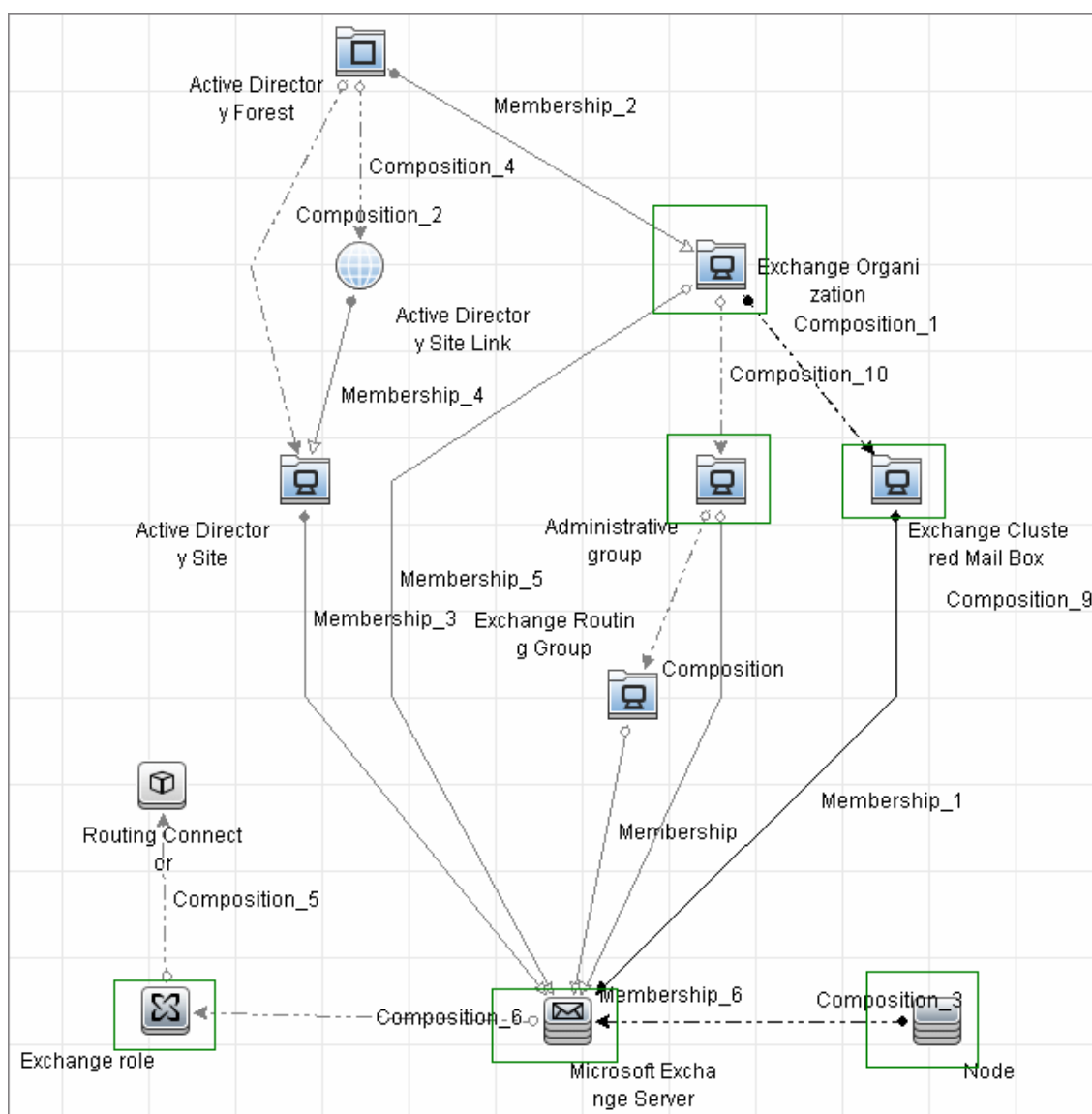
## サポートされるバージョン

Microsoft Exchange by PowerShell ディスカバリは、MS Exchange Server 2007, 2010, 2013 の各バージョンをサポートしています。

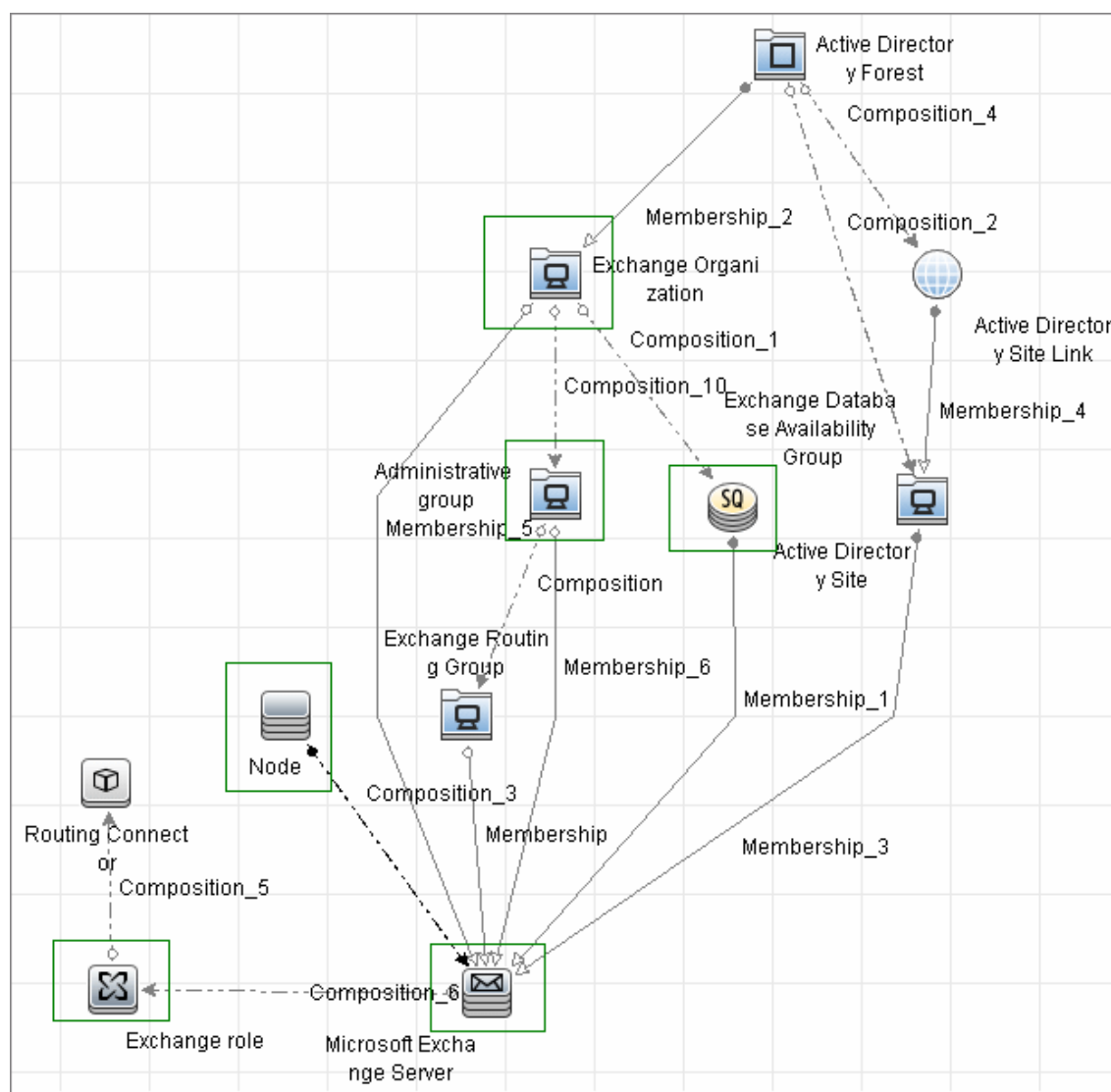
## トポロジ

次の図に、Microsoft Exchange by PowerShell トポロジを示します。枠で囲んだ CIT は、**Microsoft Exchange Topology by PowerShell** ジョブで検出できます。

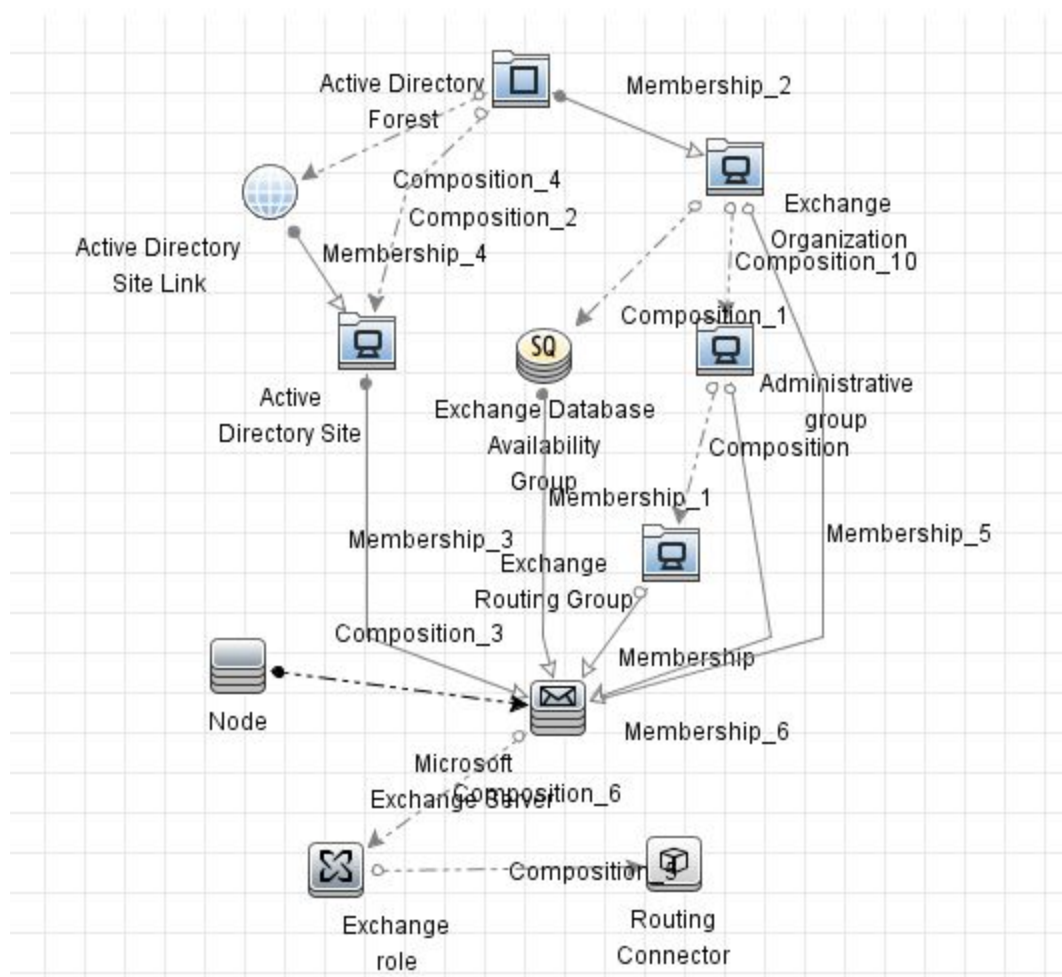
• Microsoft Exchange Server 2007 by PowerShell



- **Microsoft Exchange Server 2010 by PowerShell**



- **Microsoft Exchange Server 2013 by PowerShell**



## PowerShell を使用した Microsoft Exchange の検出方法

次の手順では、PowerShell を使用した Microsoft Exchange の検出方法を説明します。

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリ・ソリューションは、PowerShell プロトコルをベースにしています。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

検出を開始する前に、Data Flow Probe マシンに PowerShell v2.0 がインストールされていることを確認します。

### 2. 前提条件 - PowerShell のリモート処理と AD の構成

- a. PowerShell のリモート・アクセスを有効にします。詳細については、[「PowerShell のリモート処理の構成方法」\(571ページ\)](#)を参照してください。
- b. Active Directory 側を構成します。詳細については、[「Active Directory 側の構成方法」\(572ページ\)](#)を参照してください。

### 3. 前提条件 - 権限の設定

ディスカバリを開始する前に、次のコマンドを実行するために必要なすべての権限がディスカバリ・ユーザに付与されていることを確認します。

- **スナップイン :**
  - Microsoft.Exchange.Management.PowerShell.Admin(Exchange 2007)
  - Microsoft.Exchange.Management.PowerShell.E2010(Exchange 2010)
- **Get-ClusteredMailboxServerStatus**
- **Get-ExchangeServer**
- **Get-DatabaseAvailabilityGroup**
- **hostname**

#### 4. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行して、Windows システムの IP アドレスを検出します。
- b. **Host Connection by PowerShell** ジョブを実行して、PowerShell エージェントとネットワーク・トポロジを伴う Windows 接続を検出します。
- c. **Host Applications by PowerShell** ジョブを実行して、ホスト・アプリケーションを検出します。
- d. **Microsoft Exchange Topology by PowerShell** ジョブを実行します。

## PowerShell のリモート処理の構成方法

このタスクでは、PowerShell リモート・アクセスを有効にする方法について説明します。

本項の内容

#### 1. PowerShell 構成の起動

PowerShell コマンド・プロンプトで **winrm quickconfig** を実行します。

注: PowerShell 構成の起動後は、サーバ側の構成とクライアント側の構成を間違えないように注意してください。

#### 2. サーバ側マシンの構成

使用する認証方法に応じて、次の手順をサーバで実行します。

- a. **cd WSMAN:\localhost\Service\Auth** を実行します。
- b. **dir** を実行し、必要な認証タイプが有効であること(**State = True**)を確認します。必要な認証タイプが無効な場合は、"**Set-Item <認証タイプ名> True**" を実行します。標準設定では、**Kerberos** と **Negotiate** が有効になっています。
- c. **cd WSMAN:\localhost\Service** を実行し、**IPv4Filter** または **IPv6Filter** が "\*" または使用中の環境で有効な値に設定されていることを確認します。
- d. **cd WSMAN:\localhost\Listener** を実行してから **dir** を実行します。リスナが必要な IP を実際にリスンしていることを確認します。標準設定では、値 "\*" が使用されている場合、リスナはすべての IP をリスンしています。

- e. 何らかの変更を加えた場合は、**restart-service winrm** コマンドを実行して **winrm サービス** を再起動します。

### 3. クライアント側マシンの構成

クライアント・マシンで次の手順を実行します。

- a. **cd WSMAN:\localhost\Client\Auth** を実行します。
- b. **dir** を実行し、必要な認証タイプが有効であること(**State = True**)を確認します。必要な認証タイプが無効な場合は、**Set-Item <認証タイプ名> True** を実行します。

**注:** 許可されるプロトコルは、サーバ側で構成されるプロトコルと同一である必要があります。

- c. **cd WSMAN:\localhost\Client** を実行します。
- d. **dir** を実行し、**TrustedHosts** の値を確認します。標準設定では値は空なので、外部接続は不可能です。**TrustedHosts** は ACL フィールドで、許可される値はドメイン名またはドメイン名リストと、IP アドレスまたは IP アドレス・リストです。値には特殊記号 **""** が含まれる場合があります。これは、指定した宛先リストに任意の宛先または記号が表示されることを意味します。値が **""** のみの場合、クライアントはどのホストにも接続できます。これが推奨される値です。

**TrustedHosts** の値を変更するには、**Set-Item TrustedHosts <値>** を使用します。

**注:** ACL の検証中、FQDN から IP への変換は行われません。つまり、IP から接続が実行され、**TrustedHosts** フィールドにリストされているのが FQDN のみの場合 (またはその逆の場合)、接続は許可されません。

- e. 何らかの変更を加えた場合は、**restart-service winrm** コマンドを実行して **winrm サービス** を再起動します。

## Active Directory 側の構成方法

一部の Exchange PowerShell コマンドレットでは、AD のルックアップを実行する必要があります。AD サーバ(Windows 2003 以降)では、偽装が適用されたままの場合、**Anonymous** でのルックアップを許可しません。その結果、Exchange / AD 関連のコマンドレットをリモートで実行しようとするときさまざまなエラーが発生します。

本項の内容



## 1. Active Directory 側での委任の構成

このようなコマンドレットのリモート呼び出しを有効にするには、Active Directory 側で **Delegation** を構成する必要があります。

- a. 管理者アカウントを使用して、ドメイン・コントローラにログオンします。
- b. **[Start] > [Programs] > [Administrative Tools] > [Active Directory Users and Computers]**を選択します。
- c. 自分のドメインで**[Users]**フォルダを選択します。
- d. 委任するユーザ・アカウントを右クリックし、**[Properties]**をクリックします。
- e. **[Account]**タブの**[Account options]**で、**[Account is sensitive and cannot be delegated]**オプションが選択解除されていることを確認します。
- f. **[OK]**をクリックします。

## 2. 必須サーバに対する委任された要求の実行の許可

Windows ユーザ・アカウントでサーバ・プロセスが実行されている場合は、サーバ・プロセス・アカウントが委任に必要な信頼を受けていることを確認します。

- a. **[Active Directory Users and Computers] > [Users]**フォルダで、クライアントを偽装するサーバ・プロセスの実行に使用するユーザ・アカウントを右クリックし、**[Properties]**をクリックします。
- b. **[Account]**タブの**[Account options]**で、**[Account is trusted for delegation]**オプションを選択します。

## 3. サーバ・プロセス・アカウントが、サーバ・プロセスの委任に必要な信頼を受けていることを確認します。

- a. **[Active Directory Users and Computers]**で**[Computers]**を右クリックし、**[Properties]**をクリックします。
- b. サーバ・コンピュータ(クライアントを偽装するプロセスが実行される場所)を右クリックし、**[Properties]**をクリックします。
- c. **[General]**ページで**[Trust computer for delegation]**を選択します。
- d. **[Use any authentication protocol]**を選択します。

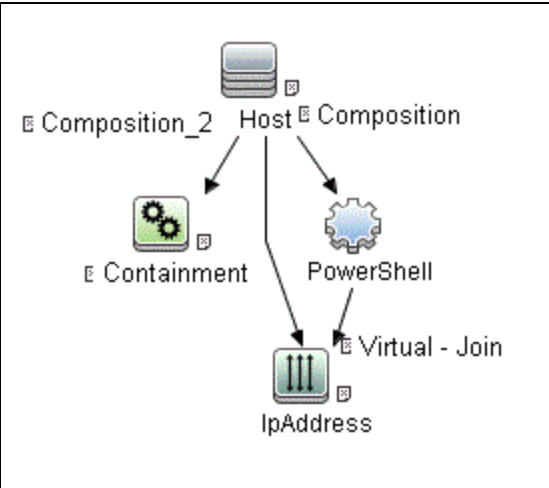
- e. [Add]をクリックし、必要なプロセスを選択します。
- f. Kerberos プロトコルのみを使用している場合は、[Trust this computer for delegation to any service]または[Use Kerberos only]を選択します。

注: Kerberos 認証を使用しており、接続が宛先ドメインの外から実行される場合は、ターゲット AD でTrust Domainを構成する必要があります。

## Microsoft Exchange Topology by PowerShell ジョブ

PowerShell を使用した Microsoft Exchange Server の検出を担当するコンポーネントは、Microsoft Exchange Server パッケージ(Microsoft\_exchange\_server.zip)にバンドルされています。

### トリガ・クエリ

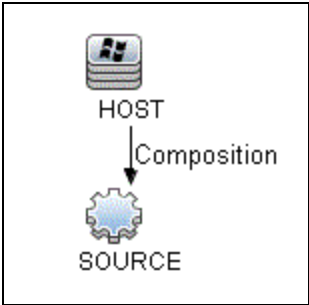


ノード名	条件
Host	なし
Containment	Name 類似 Microsoft.Exchange.%.exe
IpAddress	NOT IP Probe Name null
PowerShell	NOT Reference to the credentials dictionary entry null

アダプタ

このジョブでは、**MS\_Exchange\_Topology\_by\_Powershell** アダプタを使用します。

- 入力クエリ



- トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

- 使用スクリプト

Microsoft Exchange by PowerShell ディスカバリでは、次のスクリプトが使用されます。

- host\_win.py
- host\_win\_shell.py
- msexchange.py
- msexchange\_win\_shell.py
- ms\_exchange\_topology\_by\_powershell.py
- networking.py
- networking\_win\_shell.py

## 作成または変更されたエンティティ

エンティティ名	エンティティ・タイプ	エンティティの説明
Microsoft Exchange Topology by PowerShell.xml	ジョブ	メイン・ジョブ
MS_Exchange_Topology_by_PowerShell.xml	アダプタ	ディスカバリ・アダプタ
ms_exchange_topology_by_powershell.py	スクリプト	ディスカバリ・スクリプト
ms_exchange_process_and_powershell.xml	TQL	トリガ・クエリ
ms_exchange_clustered_mailbox.xml	クラス	CI タイプ
ms_exchange_dag.xml	クラス	CI タイプ
ms_exchange_win_shell.py	スクリプト	ディスカバリ・スクリプト
ms_exchange.py	スクリプト	ディスカバリ・スクリプト

## コマンド

Microsoft Exchange by PowerShell ディスカバリでは、次のコマンドが使用されます。

### • Get-ExchangeServer コマンド

```
Get-ExchangeServer | Where-Object {$_.Fqdn.ToLower().StartsWith
((hostname).ToLower())} | Format-List Name, Guid, Fqdn, ServerRole, DataPath,
WhenCreated, ExchangeVersion, AdminDisplayVersion, OrganizationalUnit, Site,
ExchangeLegacyDN
```

#### ◦ 出力

```
Name :SAM-RND-DC01
Guid : e8f5c340-6cf1-4fc6-aa34-226ab99282dd
Fqdn :SAM-RND-DC01.ddm-rnd.ua
ServerRole :Mailbox, ClientAccess, UnifiedMessaging, HubTransport
DataPath :C:\Program Files\Microsoft\Exchange Server\V14\Mailbox
WhenCreated :8/6/2010 5:24:05 PM
ExchangeVersion :0.1 (8.0.535.0)
AdminDisplayVersion :Version 14.0 (Build 639.21)
OrganizationalUnit : ddm-rnd.ua/SAM-RND-DC01
Site : ddm-rnd.ua/Configuration/Sites/Default-First-Site-Name
ExchangeLegacyDN :/o=SiteScope Rnd Lab/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=SAM-RND-DC01
```

。 マッピング

このコマンドの出力は、CI の次の属性の入力に使用されます。

コマンド出力属性	CI タイプ	CI 属性
Name	Exchange Server	Name
Guid	Exchange Server	Guid
Fqdn	Exchange Server	Fqdn
ServerRole	対応する Server Role CI が作成される	対応する Server Role CI が作成される
WhenCreated	Exchange Server	Creation Date
ExchangeLegacyDN	Exchange Server	Organization
AdminDisplayVersion	Exchange Server	Version
AdminDisplayVersion	Exchange Server	Application Version
AdminDisplayVersion	Exchange Server	Application Version Description

● **Get-ClusteredMailboxServerStatus コマンド**

Get-ClusteredMailboxServerStatus

。 出力

```
Identity : ddm-ex2k7ccr
ClusteredMailboxServerName :DDM-EX2K7CCR.ddm01.local
State :Online
OperationalMachines :{DDM-EX2K7CCR-N1 <Active, Quorum Owner>,
DDM-EX2K7CCR-N2}
FailedResources :{}
OperationalReplicationHostNames :{ddm-ex2k7ccr-n1, ddm-ex2k7ccr-n2}
FailedReplicationHostNames :{}
InUseReplicationHostNames :{ddm-ex2k7ccr-n1, ddm-ex2k7ccr-n2}
IsValid :True
ObjectState :Unchanged
```

。 マッピング

このコマンドの出力は、CI の次の属性の入力に使用されます。

コマンド出力属性	CI タイプ	CI 属性
Identity	Exchange Clustered Mailbox	Name
ClusteredMailboxServerName	クラスタ名の決定に使用	クラスタ名の決定に使用

## • Get-DatabaseAvailabilityGroupCommand

Get-DatabaseAvailabilityGroup | format-list

### ◦ 出力

```
Name :DDMDAG
Servers :{DDM-EXCLN2, DDM-EXCLN1}
WitnessServer :DDM-EXCLDC.DDM.LOCAL
WitnessDirectory : c:\EXCLFSW
AlternateWitnessDirectory :
NetworkCompression :InterSubnetOnly
NetworkEncryption :InterSubnetOnly
DatacenterActivationMode :Off
StoppedMailboxServers :{}
StartedMailboxServers :{}
DatabaseAvailabilityGroupIpv4Addresses :{172.24.10.129}
OperationalServers :
PrimaryActiveManager :
ThirdPartyReplication :Disabled
ReplicationPort :0
NetworkNames :{}
AdminDisplayName :
ExchangeVersion :0.10 (14.0.100.0)
DistinguishedName :CN=DDMDAG,CN=Database Availability
Groups,CN=Exchange Administrative Group
(FYDIBOHF23SPDLT),CN=Administrative Groups,CN=Discovery,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=ddm, DC=local
Identity :DDMDAG
Guid :51799b4d-9c0d-4842-990a-f9862be3e7a4
ObjectCategory : ddm.local/Configuration/Schema/ms-Exch-MDBAvailability-
Group
ObjectClass :{top, msExchMDBAvailabilityGroup}
WhenChanged :1/31/2011 4:24:34 PM
WhenCreated :1/31/2011 3:45:06 PM
WhenChangedUTC :1/31/2011 2:24:34 PM
WhenCreatedUTC :1/31/2011 1:45:06 PM
OrganizationId :
OriginatingServer : ddm-excldc.ddm.local
IsValid :True
```

。 マッピング

このコマンドの出力は、CI の次の属性の入力に使用されます。

コマンド出力 属性	CI タイプ	CI 属性
Name	Exchange Database Availability Group	Name
Distinguished name	Exchange 組織と関連付けるた めに使用	Exchange 組織と関連付けるた めに使用

## 検出された CIT

- Administrative group
- Composition
- Containment
- Exchange Client Access Server
- Exchange Clustered Mail Box
- Exchange Database Availability Group
- Exchange Edge Server
- Exchange Hub Server
- Exchange Mail Server
- Exchange Mailbox Database
- Exchange Mail Server
- Exchange Organization
- Exchange Unified Messaging Server
- ExecutionEnvironment

- IpAddress
- Membership
- MicrosoftExchangeServer
- Node
- Ownership

## トラブルシューティングおよび制限事項

本項では、Microsoft Exchange Server by PowerShell ディスカバリのトラブルシューティングと制限事項について説明します。

- **問題** : 結果が取得されず、コマンドレット呼び出しが次のエラーで終了します。

ドメイン <ドメイン名> のドメイン コントローラーを検索中に Active Directory エラー 0x80072020 が発生しました。操作エラーが発生しました。

+CategoryInfo :

+FullyQualifiedErrorId :7D2B0C9D

**理由** : "委任" が正しく構成されていません。

**解決策** : 「[Active Directory 側の構成方法](#)」(572ページ)の説明に従って、Active Directory の "委任" を構成します。

- **問題** : 結果が取得されず、コマンドレット呼び出しが次のエラーで終了します。

値を null にすることはできません。

パラメータ名 : parameters

+ CategoryInfo :

+ FullyQualifiedErrorId

:System.ArgumentNullException,Microsoft.Exchange.Management.  
SystemConfigurationTasks.GetExchangeServer



**理由 :** "委任" が正しく構成されていない, 接続が信頼されていないドメインから実行されている, または必要なすべてのパッチがサーバにインストールされていません。詳細については, Microsoft の公式サイトを参照してください。

**解決策 :** 「[Active Directory 側の構成方法](#)」(572ページ)の説明に従って Active Directory の "委任" を構成し, パッチレベルを確認します。詳細については, Microsoft の公式サイトを参照してください。

- **問題 :** Exchange のコマンドレットへの呼び出しがタイムアウトかセッション切断, またはその両方によって失敗します。

**アプリケーションはユーザを偽装できず, Windows PowerShell コマンドを Exchange Server 2007 環境で実行します。**

**理由 :** これは Exchange 2007 に関する既知のバグです。

**解決策 :** この問題を解決するには, Microsoft 修正プログラム KB943937(MS Exchange 2007 SP1 の一部)を実行します。詳細については, Microsoft 修正プログラムの説明 (<http://support.microsoft.com/kb/943937>)を参照してください。

## 第37章: Microsoft Exchange Server by WMI ディスカバリ

### 本章の内容

概要 .....	583
サポートされるバージョン .....	583
トポロジ .....	584
WMI を使用した Microsoft Exchange Server 2003 の検出方法 .....	585
Microsoft Exchange Connection by WMI ジョブ .....	586
Microsoft Exchange Topology by WMI ジョブ .....	588
作成または変更された CIT .....	590
トラブルシューティングおよび制限事項 .....	591

## 概要

DFM は、Microsoft Exchange Server(Exchange)ソフトウェア、バージョン 2003 のコンポーネントのうち、Microsoft Exchange Server、管理グループ、ルーティンググループ、組織、パブリックフォルダ、フォルダツリーを検出します。

Exchange に関するすべての情報は、WMI プロトコルを使用して **root\MicrosoftExchangeV2** 名前空間から取得されます。

Exchange ディスカバリを扱うジョブは次の 2 つです。

- Microsoft Exchange Connection by WMI
- Microsoft Exchange Topology by WMI

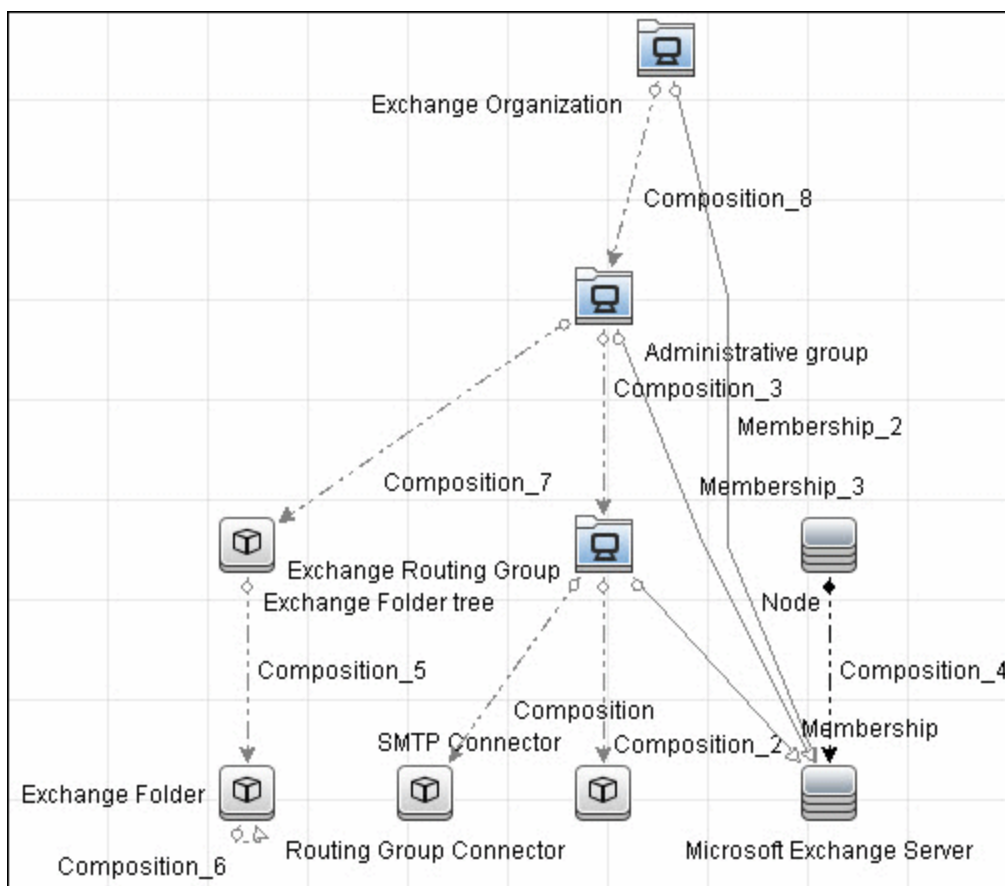
## サポートされるバージョン

Microsoft Exchange Server by WMI ディスカバリは、Microsoft Exchange Server 2003 をサポートしています。

## トポロジ

### Microsoft Exchange Topology by WMI ジョブ

DFM は、リモート・ホストに接続し、MS Exchange 2003 のトポロジを取得します。



# WMI を使用した Microsoft Exchange Server 2003 の 検出方法

このタスクでは、WMI プロトコルを使用した MS Exchange Server 2003 の検出方法を説明します。

## 1. 前提条件 - プロトコル資格情報の設定

このディスカバリは WMI に基づいています。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

Exchange に関する情報は、**root\MicrosoftExchangeV2** 名前空間から取得されます。

## 2. 前提条件 - 権限の設定

**root\MicrosoftExchangeV2** WMI 名前空間の読み取り権限を有効にする必要があります。場合によっては、**root\cimv2** 名前空間(読み取り専用権限付き)も必要になることがあります。詳細については、[「トラブルシューティングおよび制限事項」\(591ページ\)](#)を参照してください。

## 3. ディスカバリの実行

次のジョブをアクティブにします。

### ◦ Network Infrastructure ディスカバリ :

- i. **[Host Connection] > [Host Connection by WMI]**を実行して WMI CIT を検出します。
- ii. ホスト・リソースのジョブおよびホスト・アプリケーションのジョブのいずれかを実行して、ホストで実行中のプロセスに関する情報を収集します。 **emsmta.exe** または **exmgmt.exe** という名前のプロセスがホスト上で検出された場合は、**Microsoft Exchange Connection by WMI** ジョブがトリガされます。

### ◦ Enterprise Applications > Microsoft Exchange

- i. **Microsoft Exchange Connection by WMI** を実行します。このジョブは、このホストを実際に行っているサーバをレポートします。ほかの Exchange Server を検出するには、Exchange を実行している各ホストでこのジョブを実行する必要があります。このジョブは Exchange CIT を作成します。

このジョブは、WMI を使用してリモート・ホストに接続し、**root\MicrosoftExchangeV2** 名前空間にアクセスします。

次の WMI クエリが実行されます。

```
SELECT AdministrativeNote, CreationTime, ExchangeVersion, FQDN, GUID,  
MTADDataPath, MessageTrackingEnabled, MessageTrackingLogFileLifetime,  
MessageTrackingLogFilePath, MonitoringEnabled, Type FROM Exchange_Server
```

このクエリは、Exchange 組織にあるすべての Exchange Server を返します。

- ii. **Microsoft Exchange Connection by WMI** ジョブで作成される Exchange CI は、**Microsoft Exchange Topology by WMI** ジョブのトリガとして動作します。トリガ CI は、Exchange を実行中のホストに接続し、完全なトポロジを取得します。エラー・メッセージのトラブルシューティングの詳細については、[「トラブルシューティングおよび制限事項」\(591ページ\)](#)を参照してください。

このジョブは、WMI を使用してリモート・ホストに接続し、**root\MicrosoftExchangeV2** 名前空間にアクセスします。次の WMI クエリが実行されます(順序はこのまま)。

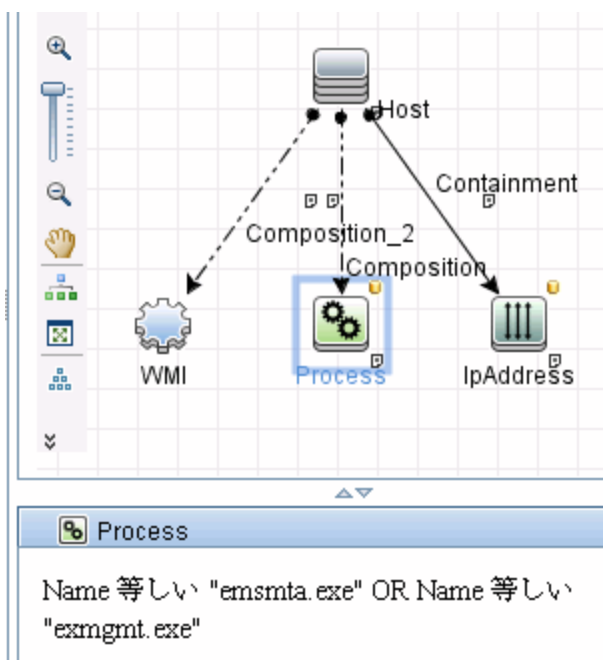
```
SELECT AdministrativeGroup, DN, FQDN, Name, RoutingGroup FROM Exchange_  
Server  
SELECT AdministrativeGroup, AdministrativeNote, CreationTime,  
Description, GUID, Name, RootFolderURL FROM Exchange_FolderTree  
SELECT AddressBookName, AdministrativeNote, Comment, ContactCount,  
FolderTree, FriendlyUrl, IsMailEnabled, Path, Url FROM Exchange_  
PublicFolder
```

## Microsoft Exchange Connection by WMI ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

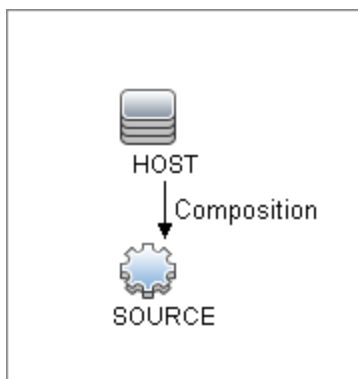
- トリガ CI : ms\_exchange\_process\_and\_wmi
- トリガ・クエリ:



## アダプタ

このジョブでは、**MS\_Exchange\_Connection\_by\_WMI** アダプタを使用します。

- 入力クエリ:



検出された CIT

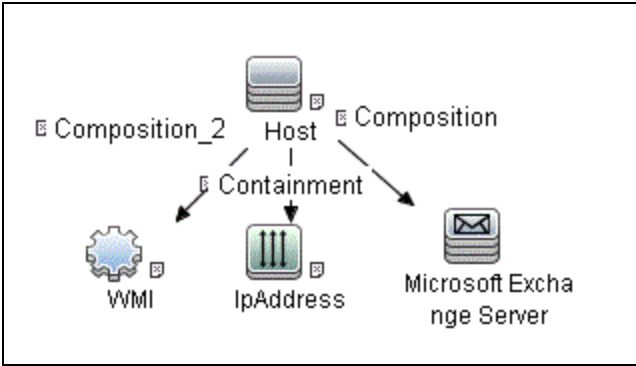
- Composition
- Computer
- MicrosoftExchangeServer

Microsoft Exchange Topology by WMI ジョブ

本 項 ではこのジョブに関する詳細を説明します。

トリガ・クエリ

- トリガ CI :ms\_exchange\_server\_and\_host\_and\_wmi
- ビュー : Microsoft Exchange Topology
- トリガ・クエリ:



ノード名	条件
Host	なし
WMI	なし
IpAddress	NOT IP Probe Name null

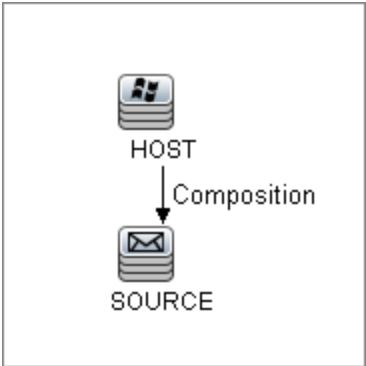


ノード名	条件
Microsoft Exchange Server	NOT Reference to the credentials dictionary entry null  AND Version 類似 %2003%

アダプタ

このジョブでは、**MS\_Exchange\_Topology\_by\_WMI** アダプタを使用します。

- 入力クエリ:



検出された CIT

- Administrative group
- Composition
- Containment
- Exchange Folder
- Exchange Folder tree
- Exchange Organization
- Exchange Routing Group

- IPAddress
- Membership
- MicrosoftExchangeServer
- Node

## 作成または変更された CIT

次の CIT が Exchange コンポーネント用に作成されます。

CIT	説明
<b>Exchange</b>	<p>この CIT は Application System フォルダにあります。これは抽象 CIT で、次の CIT の親です。</p> <ul style="list-style-type: none"> <li>• <b>Administrative group</b> : この CIT は、Exchange 組織内の管理グループを表します。</li> <li>• <b>Exchange Organization</b> : この CIT は、Exchange 組織の最上位レベルを表します。たとえば、ある組織が Exchange ソリューションを使用している場合、すべての Exchange コンポーネントは 1 つの Exchange Organization CI にリンクされます。</li> <li>• <b>Exchange Routing Group</b> : この CIT は、Exchange 組織内にあるルーティング・グループを表します。ルーティング・グループは、サーバ間にさまざまなネットワーク接続を提供するほか、特定の範囲内でユーザ・アクセスを制限します。ルーティング・グループは Exchange 2007 で廃止されました。その代わりに Exchange 2007 では、Active Directory サイトの構成を活用して Exchange Server 間の接続を実現しています。</li> </ul>
<b>Microsoft Exchange Server</b>	<p>この CIT は RunningSoftware CIT から継承されます。この CIT は、ホストにインストールされた Exchange ソフトウェアを表します。</p>
<b>Microsoft Exchange Resource</b>	<p>この CIT は Application Resource フォルダにあります。これは抽象 CIT で、次の CIT の親です。</p> <ul style="list-style-type: none"> <li>• <b>Exchange folder</b> : この CIT は、Exchange 組織で使用可能なパブリック・フォルダを表します。パブリック・フォルダは階層構造に編成できます。つまり、1 つのパブリック・フォルダに別のパブリック・フォルダを含めることができます。</li> <li>• <b>Exchange folder tree</b> : この CIT は、Exchange Server 上のパブリック・フォルダとプライベート・フォルダのツリーに関する情報を提供します。</li> </ul>

## トラブルシューティングおよび制限事項

本項では、Microsoft Exchange by WMI ディスカバリのトラブルシューティングと制限事項について説明します。

### • 管理グループの制限事項

管理グループに Exchange Server やフォルダ・ツリーが含まれない場合、管理グループは検出されません。

### • エラー・メッセージ

エラー・メッセージ	理由	解決策
Failed to obtain host name	<p>Exchange トポロジを正しくモデリングするため、<b>Microsoft Exchange Connection by WMI</b> ジョブは接続先のホスト名を認識する必要があります。</p> <p>DFM は、入力クエリと一致する、ホストの <b>host_hostname</b> 属性の取得を試みます。この属性が設定されていない場合、DFM は次の WMI クエリを実行してホストのドメイン名を取得します。</p> <pre>SELECT Name FROM Win32_ComputerSystem</pre> <p>このクエリが何らかの理由で失敗した場合も、このエラー・メッセージが表示されてこのジョブは失敗します。</p>	<ul style="list-style-type: none"><li>正しいホスト名を取得するジョブを実行します。</li><li>ホスト名を手動で設定します。</li><li>ログ・ファイルで、ホスト名の WMI クエリが失敗した理由に関する情報を確認します。</li></ul>
Failed to discover folder trees and public folders		接続に使用している資格情報が、 <a href="#">「前提条件 - プロトコル資格情報の設定」</a> (585ページ)で説明されているものと一致しているかどうかを確認します。

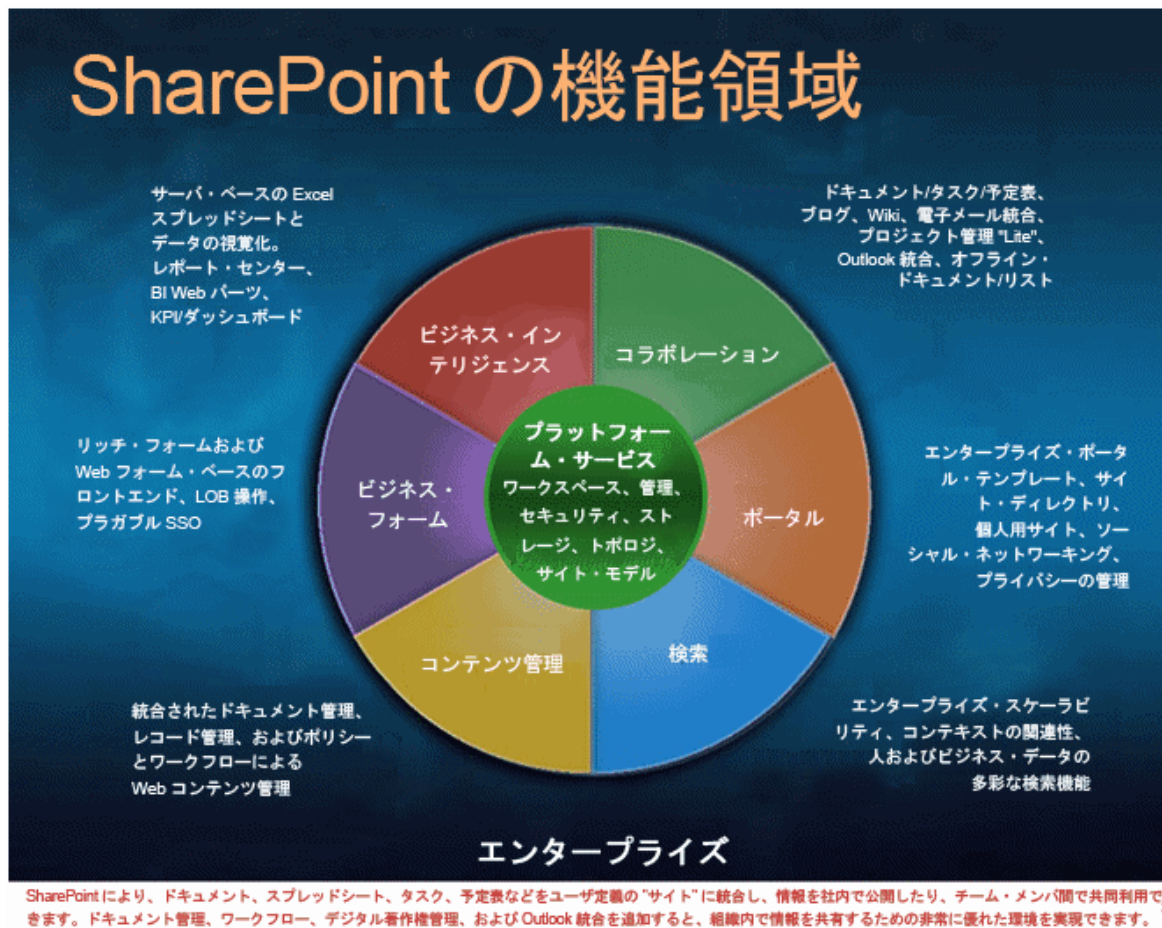
## 第38章: Microsoft SharePoint ディスカバリ

### 本章の内容

概要 .....	593
サポートされるバージョン .....	593
トポロジ .....	594
Host Connection by Shell ジョブ .....	594
Host Applications by Shell ジョブ .....	594
Microsoft SharePoint Topology ジョブ .....	596
Microsoft SharePoint の検出方法 .....	596
Microsoft SharePoint Topology ジョブ .....	597
Miscrosoft SharePoint ディスカバリ・コマンド .....	602
トラブルシューティングおよび制限事項 .....	606

## 概要

Microsoft SharePoint は、Microsoft がコラボレーション、ファイル共有、Web パブリッシングのために開発したソフトウェア製品ファミリです。この製品ファミリには、Microsoft SharePoint Server、Microsoft SharePoint Foundation、Microsoft Search Server、Microsoft SharePoint Designer、Microsoft SharePoint Workspace が含まれます。



CMDB クラス・モデルの観点からは、Microsoft SharePoint は、IIS ベースの Web 層と MS SQL Server ベースのストレージ層を持つサービス・セット (アプリケーション・サーバ、検索 サーバ、インデックス付けサーバなど) であると言えます。

## サポートされるバージョン

Microsoft SharePoint ディスカバリは、次のバージョンをサポートしています。

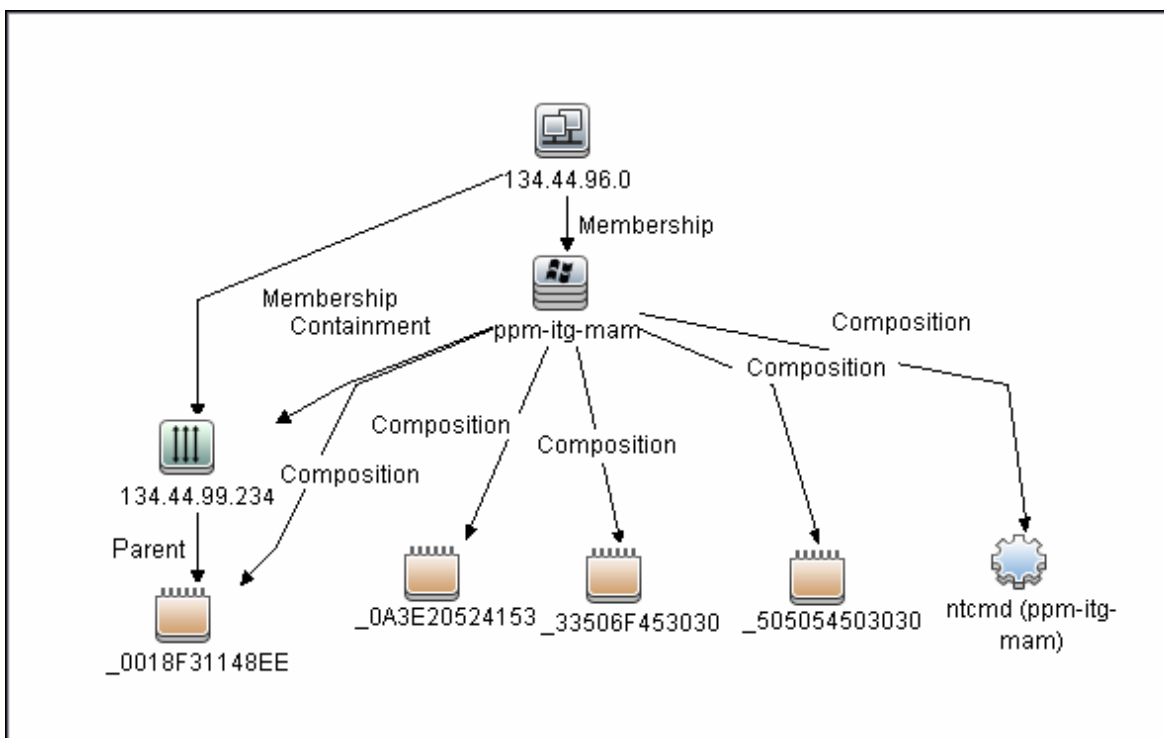
- Microsoft SharePoint 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013

注: このディスカバリは、Microsoft SharePoint の入手可能なすべてのバージョンで動作することが想定されています。

## トポロジ

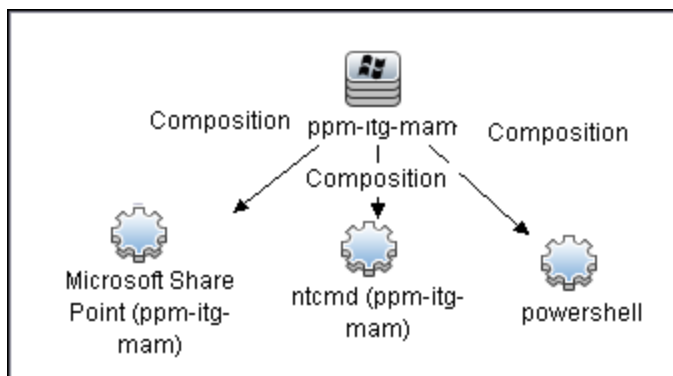
次の図に、SharePoint ディスカバリ・ジョブのサンプル出力を示します。

### Host Connection by Shell ジョブ



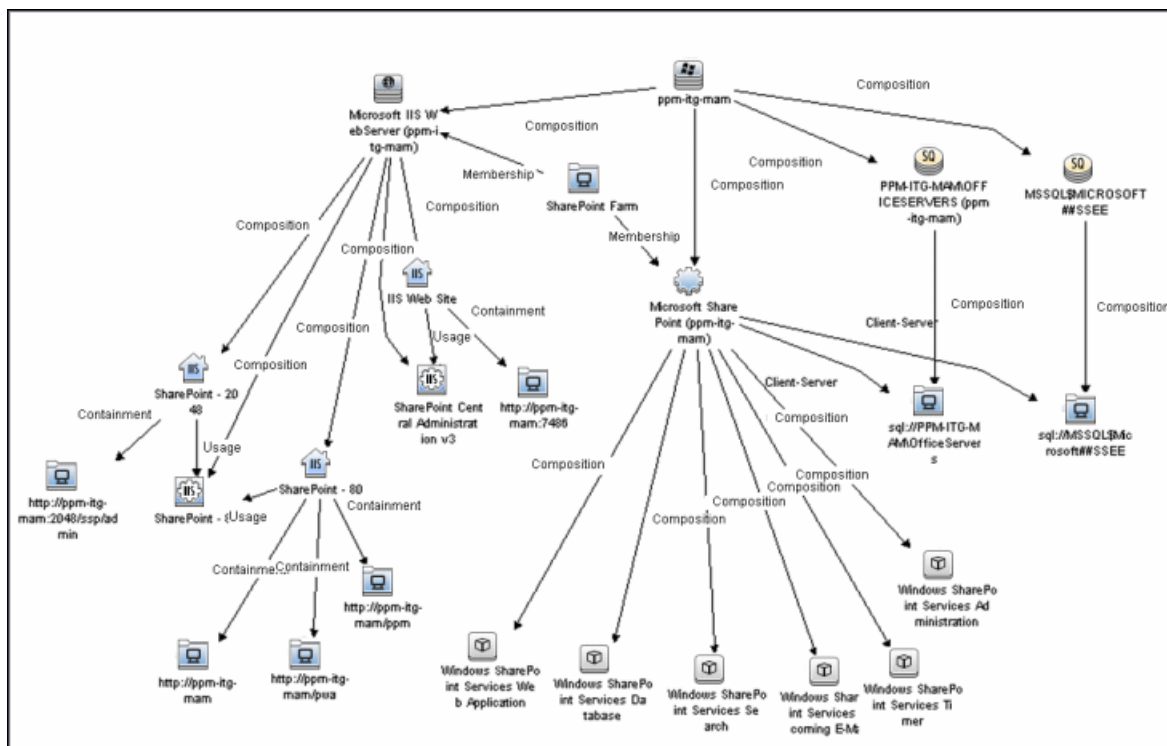
### Host Applications by Shell ジョブ

注: 継続フローに必要なデータのみが示されています。



## Microsoft SharePoint Topology ジョブ

注: 検出された CIT のリストについては、「[検出された CIT](#)」(601ページ)を参照してください。



## Microsoft SharePoint の検出方法

次の手順では、Microsoft SharePoint の検出方法について説明します。

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリ・ソリューションは、PowerShell プロトコルに基づいています。これには、スクリプトの実行レベルで NTCMD, SSH, Telnet の各プロトコルを介してアクセスすることもできます。対応する資格情報が用意されていることを確認します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - ユーザ権限の設定



ログインしたユーザに、SharePoint 構成 データベースへの読み取り権限がある必要があります。

### 3. ディスカバリの実行

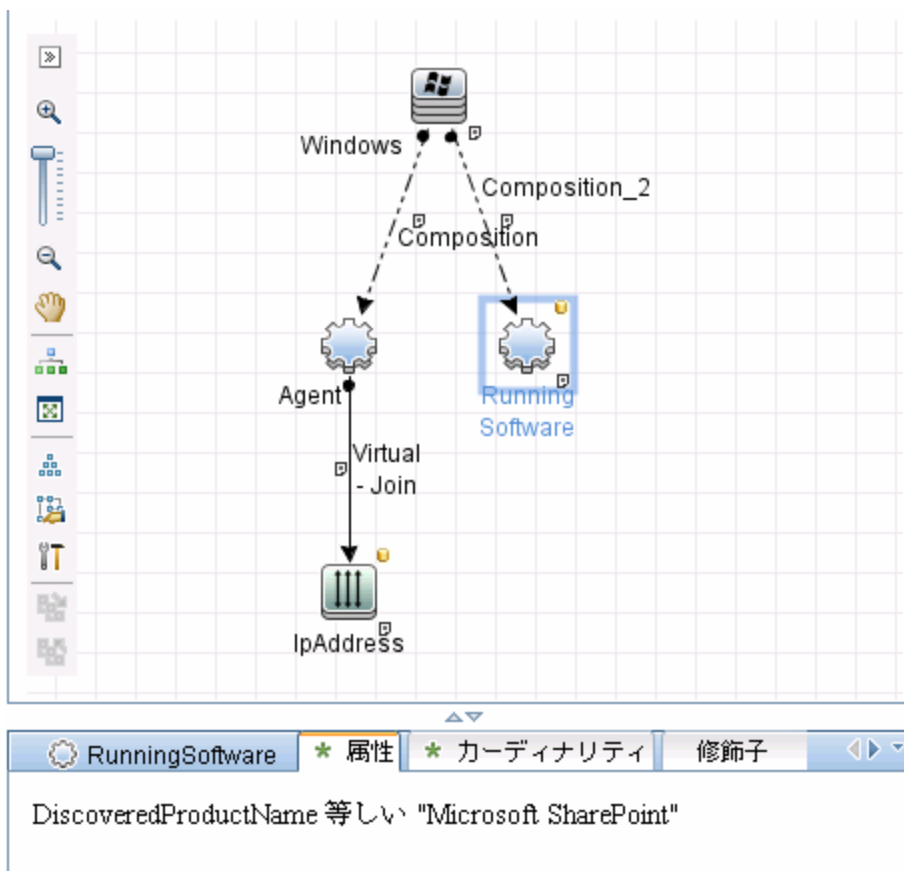
- a. **Range IPs by ICMP** ジョブまたは **Range IPs by nmap** ジョブを実行して、SharePoint システムの IP アドレスを検出します。
- b. **Host Connection by Shell** ジョブまたは **Host Connection by Powershell** ジョブを実行し、SharePoint およびシェルまたは PowerShell エージェントの間の接続と、ネットワークトポロジを検出します。
- c. **Host Applications by Shell** ジョブまたは **Host Applications by PowerShell** ジョブを実行し、SharePoint システムと SharePoint ソフトウェア要素の間の接続と、ホストトポロジの詳細を検出します。
- d. **Microsoft SharePoint Topology** ジョブを実行し、Microsoft SharePoint Server トポロジを検出します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Microsoft SharePoint Topology ジョブ

本項ではこのジョブに関する詳細を説明します。

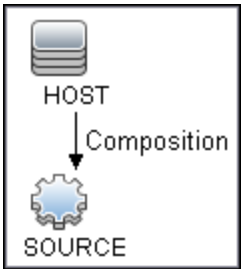
## トリガ・クエリ



注: IPAddress に, **IP Probe name not null** 属性が設定されます。

## アダプタ

- 入力 CIT :Agent
- 入力クエリ



- 使用スクリプト
  - sharepoint\_win\_shell.py
  - sharepoint.py
  - SharePointMain.py

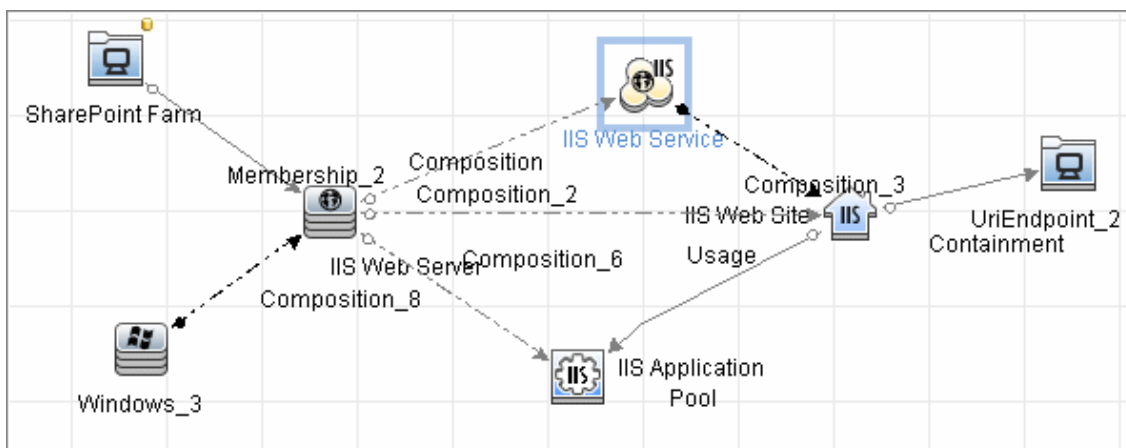
注: このジョブでは, **AutoDiscoveryContent** パッケージで提供されるライブラリ・スクリプトも使用される場合があります。

ジョブ・パラメータ

パラメータ	説明
discoverSharePointUrls	SharePoint サイトの URL を検出するかどうかを示します。
relativeCommandTimeoutMultiplier	標準設定のコマンド実行時間に替えて結果を待機する時間を指定します。
reportIntermediateWebService	IIS Web サーバと IIS Web サイトとの間の IIS WebService をレポートするかどうかを示します。このパラメータは, IIS Applications by NTCMD or UDA ジョブの <b>report_legacy_topology</b> パラメータに応じて設定する必要があります。

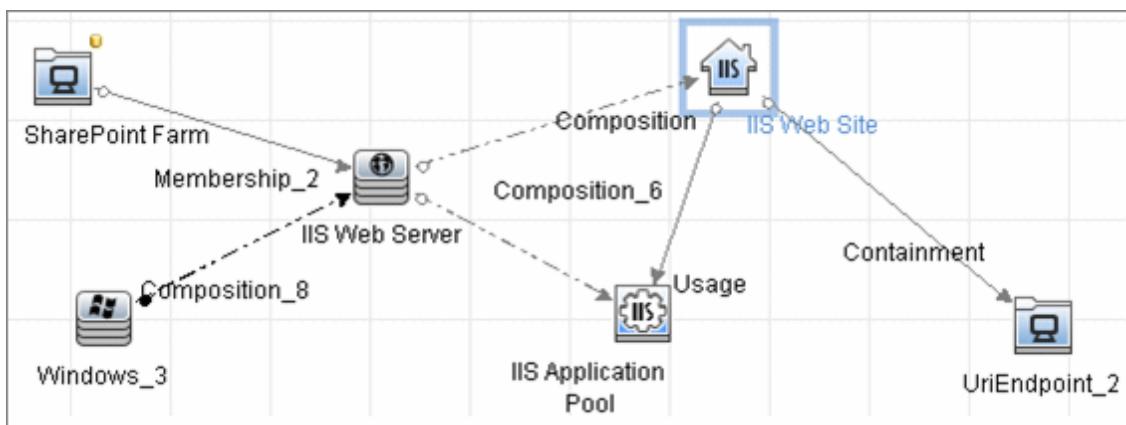
**reportIntermediateWebService** パラメータの設定に応じて, このジョブは次の IIS トポロジのいずれかをレポートします。

- **reportIntermediateWebService = true :**  
  
IIS Web Server -> IIS Web Service -> IIS Web Site



- **reportIntermediateWebService = false :**

IIS Web Server -> IIS Web Service -> IIS Web Site



## 作成または変更されたエンティティ

エンティティ名	エンティティ・タイプ	エンティティの説明
SharePoint Farm	CIT	SharePoint ファームに関する新しい CIT 情報
SharePoint Service	CIT	新しい CIT - SharePoint サービス構成に関するデータが格納されたテキスト・ファイル
Microsoft SharePoint Topology	ジョブ	新しいトポロジ・ジョブ

エンティティ名	エンティティ・タイプ	エンティティの説明
<b>Enterprise Applications - Microsoft SharePoint</b>	モジュール	ディスカバリ・モジュール
<b>ms_sharepoint_by_shell</b>	アダプタ	ディスカバリ・アダプタ
<b>sharepoint_application_agents.xml</b>	TQL クエリ	トリガ TQL クエリ
<b>sharepoint.py</b>	スクリプト	SharePoint トポロジ・スクリプト
<b>sharepoint_win_shell.py</b>	スクリプト	SharePoint discovery by Shell および PowerShell のメカニズムが格納されたスクリプト
<b>SharePointMain.py</b>	スクリプト	ジョブのエントリ・ポイントであるメイン・スクリプト
<b>Sharepoint_xml.ps1</b>	リソース	SharePoint 構成を XML 形式で表す PowerShell スクリプト

## 検出された CIT

- ClientServer
- Composition
- Containment
- IIS Application Pool
- IIS Web Server
- IIS Web Service
- IIS Web Site
- IpAddress
- Membership
- RunningSoftware
- SharePoint Farm

- SharePoint Service
- SQL Server
- UriEndPoint
- Usage
- Windows

注: トポロジを表示するには, [「トポロジ」\(594ページ\)](#)を参照してください。

## Microsoft SharePoint ディスカバリ・コマンド

SharePoint トポロジは, **Sharepoint\_xml.ps1** スクリプトを実行することで検出されます。これには, 関連情報を XML 形式で提供する次の機能が含まれます。

本項の内容

- [「ShowSharePointConfig」\(602ページ\)](#)
- [「ShowSharePointHostConfig」\(603ページ\)](#)
- [「ShowSharePointWebConfig」\(604ページ\)](#)
- [「SharePoint ライブラリ・コマンド・フロー」\(606ページ\)](#)

### ShowSharePointConfig

- サンプル出力

```
<farm id="4ddfb9c7-754a-4a66-8ee6-7d86613b873c" version="12.0.0.6421">  
<hosts> As described for ShowSharePointHostConfig section </hosts>  
<webServices> As described for ShowSharePointWebConfig section </webServices>  
</farm>
```

- モデリングされた CIT : SharePoint Farm

属性	値
ID	4ddfb9c7-754a-4a66-8ee6-7d86613b873c

## ShowSharePointHostConfig

### • サンプル出力

```
<hosts>
  <host name="ucmdb-11">
    <db type="SharedDatabase">Server=ucmdb-11;Database=SharedServices1_DB;Trusted_
Connection=yes;App=Windows SharePoint Services;Timeout=15</db>
    <db type="SPConfigurationDatabase">Server=ucmdb-11;Database=SharePoint_
Config;Trusted_Connection=yes;App=Windows SharePoint Services;Timeout=15</db>
    <service name="Windows SharePoint Services Database">
      Databases          :
      NormalizedDataSource : ucmdb-11
    ...
  </service>
</host>
</hosts>
```

### • モデリングされた CIT : IPAddress

属性	値
IP Address	ucmdb-11 の解決された IP

### • モデリングされた CIT : Windows

属性	値
Host key	'ucmdb-11 の解決された IP' 'IP ドメイン'

### • モデリングされた CIT : Software Element

属性	値	コメント
Container	前述の Windows	
Name	Microsoft SharePoint	
Vendor	microsoft_corp	
Application version	12.0.0.6421	SharePoint ファームのバージョン属性から取得

• モデリングされた CIT : SQL Server

属性	値
Container	前述の Windows
Database Name	ucmdb-11
Vendor	microsoft_corp

• モデリングされた CIT : SharePoint Service

属性	値
Container	前述の Software Element
名前	Windows SharePoint Services データベース
Document Data	Databases : NormalizedDataSource : ucmdb-11 ...

## ShowSharePointWebConfig

• サンプル出力

```
<webServices>
  <webService id="c8e64134-0daa-4614-9ed8-257aa653fe9c">
    <applicationPool name="SharePoint - 80">
      <webApplication name="SharePoint - 80">
        <url>http://ddvm-shrpnt/</url>
        <site>http://ddvm-shrpnt</site>
```



```
<site>http://ddvm-shrpnt/personal/administrator</site>
<site>http://ddvm-shrpnt/ssp/admin</site>
</webApplication>
</webService>
</webServices>
```

• モデリングされた CIT : Windows

属性	値
Host key	'ddvm-shrpnt の解決された IP' 'IP ドメイン'

• モデリングされた CIT : IIS

属性	値	コメント
Container	前述の Windows	
Name	Microsoft IIS WebServer	
Vendor	microsoft_corp	

• モデリングされた CIT : IIS Application Pool

属性	値
Container	前述の IIS
Name	SharePoint - 80
Vendor	microsoft_corp

• モデリングされた CIT : IIS Web Site

属性	値
Container	前述の IIS
Name	SharePoint - 80

• モデリングされた CIT : URL

属性	値
Container	IIS Host(Windows)
Name	http://ddvm-shrpnt

## SharePoint ライブラリ・コマンド・フロー

SharePoint ライブラリは、次のコマンド・フローを使用してロードされます。

- `[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SharePoint");`
- `$spFarm = [Microsoft.SharePoint.Administration.SPFarm]::Local;`
- `if(!$spFarm){echo("—CANNOT EXECUTE DISCOVERY—"); exit(1)}`

最後のコマンドの実行後、ローカルの SharePoint ファームが初期化されるか、**—CANNOT EXECUTE DISCOVERY—** というメッセージが表示されます。

SharePoint が PowerShell によって検出された場合、**ShowSharePointHostConfig** コマンドと **ShowSharePointWebConfig** コマンドが呼び出されます(上記の「[Microsoft SharePoint ディスカバリ・コマンド](#)」(602ページ)を参照)。SharePoint Farm CI は、次のコマンドの実行から構築されます。

- `Echo($spFarm.Id.Guid)` – ファーム ID を検出
- `Echo($spFarm.BuildVersion.ToString())` – ファーム・バージョンを検出

## トラブルシューティングおよび制限事項

本項では、Microsoft SharePoint ディスカバリのトラブルシューティングと制限事項について説明します。

1. ジョブが SharePoint ホストに接続するために使用する資格情報によって、SharePoint 構成データベースへの信頼される接続が提供される必要があります。データベース・ホストが3番目のホスト(検出されたホスト)であり、信頼される接続が SharePoint 構成データベースに使用される場合、このような構成は検出されません。この問題を回避するため、SQL 資格情報が SharePoint 構成に使用される必要があります。

ディスカバリ・メカニズムは次の場合に機能します。

- SharePoint 構成データベースは、名前付きパイプ(単一のホスト上のファーム)を介して接続される

- 構成データベースに SQL 接続が使用される
  - 構成データベースに信頼される接続が使用され、このデータベースはその他の SharePoint コンポーネントを使用してホストされる
2. 各 SharePoint サービスについて、すべての構成詳細は、SharePoint Service CIT の **service configuration** 属性の 1 つの文字列にマージされます。
  3. 「**No SharePoint library found**」という警告が表示される場合、SharePoint データベース・マシンでイベント・ビューアを参照し、検出対象の SharePoint インスタンスから成功しなかった接続が試みられたかどうかを確認することをお勧めします。成功しなかった接続の試行があった場合、MS SQL Server マネージャ(データベースにアクセスできなかったもの)に新しいログインを追加し、この新しいログインに対して **SharePoint\_Config** の **db\_owner** 権限を付与します。

## 第39章: SAP ディスカバリの概要

SAP ディスカバリは、主に SAP アーキテクチャ・レイヤを対象とし、さまざまな SAP システム構成に従って検出を行います。サポートされている SAP システム構成は次のとおりです。

- ABAP
- JAVA
- DS(二重スタック)

各 SAP システムは、次の要素から構成されます。

- 次のタイプを含む NetWeaver アプリケーション・サーバ
  - セントラル・インスタンス
  - SCS(セントラル・サービス)
  - ダイアログ・インスタンス
- メッセージ・サーバまたはエンキュー・サーバ
- データベース
- 次を含むコンポーネント
  - ソフトウェア・コンポーネント
  - 開発コンポーネント

注: SAP ディスカバリでは、JAVA 構成および ABAP 構成の場合、1つのアプリケーション・サーバが構成の1つのインスタンスとみなされますが、DS 構成の場合、同じホスト上の同じ名前のインスタンスを持つ2つのアプリケーション・サーバが、DS 構成の1つのインスタンスとみなされます。

## ディスカバリ・メカニズム

SAP ディスカバリは、ディスカバリを実行するために次の2種類のメカニズムを使用します。

- **シャロー** : SAP ディスカバリはアプリケーション署名とプラグインを使用します。
- **ディープ** : SAP ディスカバリは専用ジョブを使用します。

ほとんどの専用ジョブは、ABAP または JAVA のアプリケーション・サーバ・タイプで分離されます。サーバが DS 構成の一部とみなされた場合、システム全体が DS としてマークされます。

**注:** SAP Solution Manager への接続を検出することによって、SAP システム全体を検出できます。この方法で、単一の資格情報セットを作成するので SAP システムごとに 1 つの資格情報セットを作成する必要がありません。

## アプリケーション署名とプラグイン

各種アプリケーション・サーバでの JAVA 構成および ABAP 構成用に、次のアプリケーション署名を使用できます。

- SAP ABAP SCS
- SAP ABAP アプリケーション・サーバ(ダイアログ)
- SAP ABAP アプリケーション・サーバ(セントラル)
- SAP JAVA SCS
- SAP JAVA アプリケーション・サーバ(ダイアログ)
- SAP メッセージ・サーバと **plugins\_sap\_message\_server** プラグイン(ABAP, JAVA, DS を含むすべての SAP 構成に関係)
- SAP エンキュー・サーバと **plugins\_sap\_enqueue\_server** プラグイン(ABAP, JAVA, DS を含むすべての SAP 構成に関係)

すべての ABAP 関連署名は、次のプラグインを使用して処理されます。

- **plugins\_sap\_abap\_instance\_to\_system\_linkage**
- **sap\_cs\_and\_ap\_version\_by\_shell**
- **sap\_cs\_and\_ap\_version\_by\_wmi**

すべての JAVA 関連署名は、次のプラグインを使用して処理されます。

- **plugins\_sap\_java\_instance\_to\_system\_linkage**
- **sap\_cs\_and\_ap\_version\_by\_shell**
- **sap\_cs\_and\_ap\_version\_by\_wmi**

注: 次のプラグインは, ABAP 関連署名と JAVA 関連署名の両方で共通です。

- **sap\_cs\_and\_ap\_version\_by\_shell**
- **sap\_cs\_and\_ap\_version\_by\_wmi**

## 第40章: SAP ABAP ディスカバリ

### 本章の内容

概要 .....	612
サポートされるバージョン .....	612
トポロジ .....	613
SAP ABAP の検出方法 .....	613
SAP Solution Manager Topology by SAP JCO ジョブ .....	618
SAP Solution Manager by SAP JCO ジョブ .....	620
SAP Applications by SAP JCO ジョブ .....	622
SAP ABAP Topology by SAP JCO ジョブ .....	624
SAP ABAP Connection by SAP JCO ジョブ .....	627
SAP ITS by NTCMD or UDA ジョブ .....	629
SAP System by Shell ジョブ .....	631
SAP TCP Ports ジョブ .....	632
アダプタ情報 .....	635
トラブルシューティングおよび制限事項 .....	636

## 概要

UCMDB は、SAP アプリケーション・サーバ ABAP を検出します。これは、ABAP アプリケーションを実行するための包括的テクノロジーおよびインフラストラクチャを提供する言語です。

注: SAP Solution Manager への接続を検出することによって、SAP システム全体を検出できます。この方法で、単一の資格情報セットを作成しますので、SAP システムごとに 1 つの資格情報セットを作成する必要がありません。DFM はこの 1 つのセットですべてのシステム(およびトポロジ)を検出します。詳細については、「[SAP Solution Manager ディスカバリ](#)」(656 ページ)を参照してください。

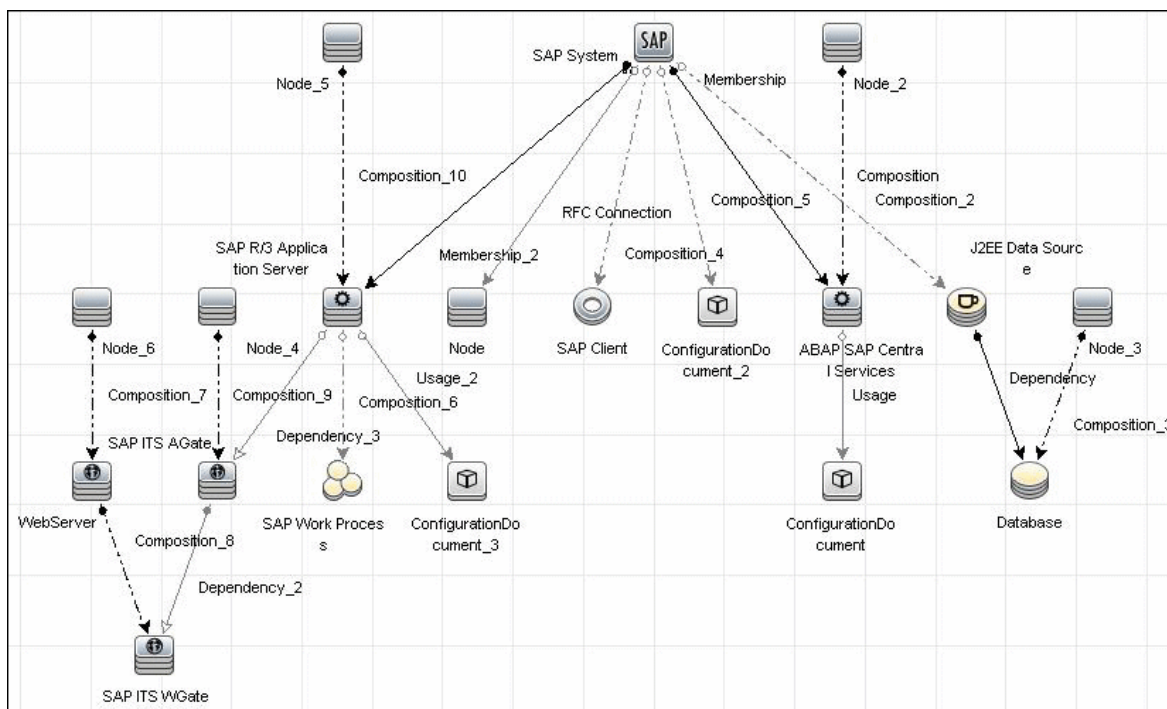
## サポートされるバージョン

SAP BASIS および SAP AS(アーキテクチャ・レイヤ)	バージョン 3.x ~ 6.x
SAP JCo	2.x および 3.x(3.0.7 以降) バージョン 3.0.7 以降を推奨
SAP Solution Manager	バージョン 6.x, 7.x



## トポロジ

次の図に、SAP ABAP ディスカバリのトポロジを示します。



## SAP ABAP の検出方法

このタスクでは、SAP ABAP アーキテクチャ、SAP アプリケーション・コンポーネント、SAP トランザクション、SAP Solution Manager ビジネス・プロセス定義を検出します。これには次の手順が含まれます。

### 1. 前提条件 - プロトコル資格情報の設定

次のプロトコルで、SAP システムがマシン上にインストールされているかどうかを確認するためのマシン接続が可能です。

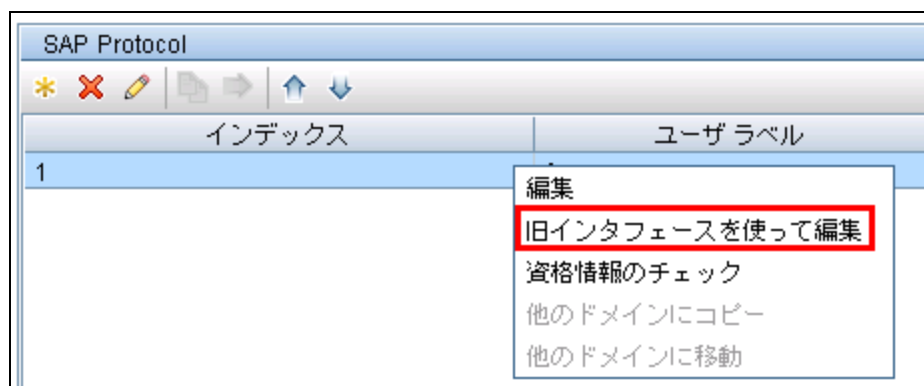
- NTCMD プロトコル
- SSH プロトコル

- Telnet プロトコル
- SAP プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

適切な JCo バージョンを選択するには、次の手順を実行します。

- SAP プロトコル用の資格情報を作成するか、既存の資格情報を選択します。
- 資格情報を右クリックして、下に示すように、[旧インターフェースを使って編集]を選択します。



- ダイアログ・ボックスが開いたら、JCo バージョンが適切であることを確認するか、適切なバージョンに変更します。標準設定のバージョンは 2.x です。

## 2. 前提条件 – Java コネクタのインストール

注: このパートのすべての操作は、Data Flow Probe がインストールされているマシン上で実行する必要があります。

### JCo バージョン 3.x(3.0.7 以降)

- SAP JCo パッケージをダウンロードします。SAP JCo パッケージは、**SAP Service Marketplace > SAP JCo > Tools & Services** ウィンドウ(<http://service.sap.com/connectors>)からアクセスできます。
- JCo インストール ZIP の中身を一時ディレクトリ(C:\temp など)に展開します。
- 一時ディレクトリから **sapjco3.jar** を <DataFlowProbe\_root>\content\lib\ ディレクトリにコピー

します。

c の代わりに, **sapjco3.jar** を <DataFlowProbe\_root>\content\lib\sap\ ディレクトリにコピーできます。その場合, <DataFlowProbe\_root>\bin\WrapperEnv.conf ファイルから次の文字列を見つけて変更する必要があります。

```
set.SAP_CLASSES=%sap%/sapjco.jar;%sap%/com_sap_pj_jmx.jar;
%sap%/exception.jar;%sap%/logging.jar;%sap%/sapj2eeclient.jar;
%sap%/sapxmltoolkit.jar
```

次のように, 行末に ;%sap%/sapjco3.jar を追加します。

```
set.SAP_CLASSES=%sap%/sapjco.jar;%sap%/com_sap_pj_jmx.jar;
%sap%/exception.jar;%sap%/logging.jar;%sap%/sapj2eeclient.jar;
%sap%/sapxmltoolkit.jar;%sap%/sapjco3.jar
```

- d. 一時ディレクトリから, sapjco3.jar ファイルが格納されている DataFlowProbe ディレクトリに **sapjco3.dll** をコピーします。
- e. プローブの **WrapperEnv.conf** ファイルで, %sap%/sapjco.jar を %sap%/sapjco3.jar に置き換えて, SAP CLASSES の環境ディスカバリ・パスを変更します。

## JCo バージョン 2.x

- a. SAP JCo パッケージをダウンロードします。SAP JCo パッケージは, **SAP Service Marketplace** > **SAP JCo** > **Tools & Services** ウィンドウ(<http://service.sap.com/connectors>) からアクセスできます。
- b. JCo インストール ZIP の中身を一時ディレクトリ(C:\temp など)に展開します。
- c. 一時ディレクトリから **sapjco.jar** を <DataFlowProbe\_root>\content\lib\ ディレクトリにコピーします。

c の代わりに, **sapjco.jar** を <DataFlowProbe\_root>\content\lib\sap\ ディレクトリにコピーできます。その場合, 次の文字列に示すように, **sapjco.jar** が <DataFlowProbe\_root>\bin\WrapperEnv.conf ファイルの SAP\_CLASSES パラメータで定義されていることを確認する必要があります。

```
set.SAP_CLASSES=%sap%/sapjco.jar;%sap%/com_sap_pj_jmx.jar;
%sap%/exception.jar;%sap%/logging.jar;%sap%/sapj2eeclient.jar;
%sap%/sapxmltoolkit.jar
```

- d. 一時ディレクトリから, sapjco.jar ファイルが格納されている DataFlowProbe ディレクトリに

**sapjcorfc.dll** をコピーします。

- e. 一時ディレクトリから、リンク機能によってロードされる共有ライブラリ用のディレクトリに **librfc32.dll** をコピーします。この場所は通常 **%winnt%** ディレクトリまたは **%winnt%\System32\** ディレクトリです。詳細については、JCo の README を参照してください。

### 3. アダプタ・パラメータの構成

検出する CI を明確に指定する、または不要な CI を除外するには、アダプタ・パラメータを次のように構成できます。

ディスカバリ	構成
すべての SAP トランザクションを検出	<b>getAllTransactions</b> を <b>true</b> に設定
アクティブな SAP トランザクションを検出	<b>getActiveTransactions</b> を <b>true</b> に設定
検出されたトランスポートによって変更された SAP トランザクションを検出	<ul style="list-style-type: none"> <li>◦ <b>getTransChanges</b> を <b>true</b> に設定</li> <li>◦ <b>from</b> の日付 (<b>transChangesFromDate</b>) と <b>to</b> の日付 (<b>transChangesToDate</b>) の設定。日付形式は <b>MM/DD/YYYY</b> または <b>YYYYMMDD</b>。</li> <li>◦ <b>from</b> の時刻 (<b>transChangesFromTime</b>) と <b>to</b> の時刻 (<b>transChangesToTime</b>) の設定。時刻形式は <b>HH:MM:SS</b> または <b>HHMMSS</b>。</li> </ul>

アダプタ・パラメータの構成の詳細については、『HP Universal CMDB Data Flow Management Guide』でアダプタ管理について説明した項を参照してください。

### 4. ディスカバリの実行

- a. [Universal Discovery] ウィンドウで、次の順にジョブをアクティブ化します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の Module/Job-Based Discovery について説明した項を参照してください。

- **Range IPs by ICMP** または **Range IPs by nmap**, **Host Connection By Shell**。
- **Host Resources by Shell** および **Host Applications by Shell**。SAP を実行中のソフトウェアおよびプロセスを検出します。
- **SAP TCP Ports**。

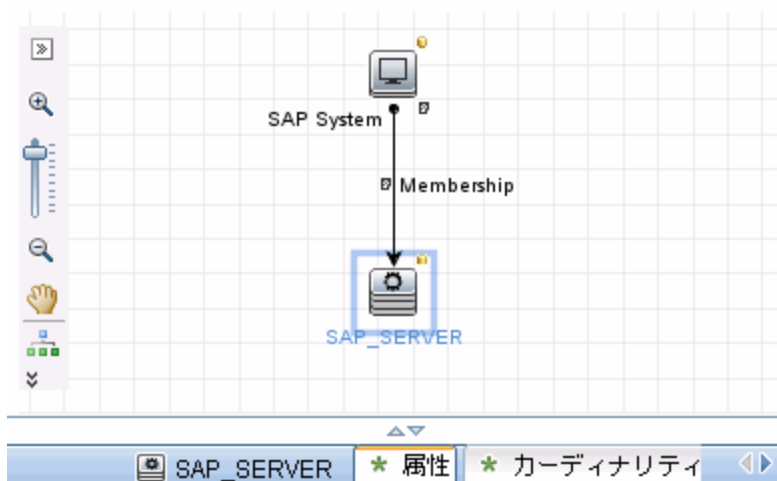
- **SAP System by Shell。**ファイル・システムおよびプロセス・リストを参照することによって、SAP システムを検索します。作成された SAP CI は、**SAP ABAP Connection by SAP JCO** ジョブのトリガとして使用されます。このジョブは、SAP の資格情報でなく、シェルの資格情報を必要とします。
  - **SAP ABAP Connection by SAP JCO。**SAP システムに接続し、資格情報 ID を属性として持つ SAP System CI を作成します。その後、これらの資格情報は、他の ABAP ジョブが SAP に接続するために使用します。
  - **SAP ABAP Topology by SAP JCO。**SAP システム内のインフラストラクチャ・エンティティを検出します。これにはホスト、アプリケーション・サーバ、ワーク・プロセス、データベース、SAP クライアント、構成ファイル、ソフトウェア・コンポーネント(構成ファイルとして検出)、サポート・パッケージ(構成ファイルとして検出)が含まれます。
  - **SAP Applications by SAP JCO。**このシステムのアプリケーション・コンポーネントを検出します。このジョブの結果、多数の CI が検出される可能性があります。不要な CI を除外するには、アダプタ・パラメータを構成できます。詳細については、[「アダプタ・パラメータの構成」\(616ページ\)](#)を参照してください。
  - **SAP ITS by NTCMD or UDA。**Internet Transaction Server(ITS)エンティティ(アプリケーション・ゲートウェイおよび Web ゲートウェイ)を検出します。
  - **SAP Solution Manager by SAP JCO。**SAP Solution Manager コンポーネントを検出します。SAP Solution Manager ディスカバリを使用すると、ビジネス・プロセス階層を検出できます。詳細については、[「SAP Solution Manager ディスカバリ」\(656ページ\)](#)を参照してください。
- b. 検出された CI の詳細については、『HP Universal CMDB Data Flow Management Guide』の[ディスカバリ ジョブの詳細]表示枠について記述された項を参照してください。
- c. DFM が適切なコンポーネントを検出したことを確認します。モデリング・スタジオの **SAP\_ABAP\_Topology** ビューにアクセスし、マップにすべてのコンポーネントが表示されることを確認します。
- d. SAP APAB ディスカバリで検出された CI を表示するには、『HP Universal CMDB Data Flow Management Guide』の[検出 CI]ウィンドウについて記述された項を参照してください。

## SAP Solution Manager Topology by SAP JCO ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

- トリガ CI :SapAbapApplicationServer
- トリガ・クエリ:



NOT Reference to the credentials dictionary entry null AND NOT instance number null

### 使用スクリプト

- db.py
- db\_builder.py
- db\_platform.py
- iteratortools.py
- jdbc.py
- jmx.py

- jdbc\_url\_parser.py
- jee.py
- sap.py
- sap\_abap.py
- sap\_abap\_discoverer.py
- sap\_discoverer.py
- sap\_jee.py
- sap\_solman\_discoverer.py
- sap\_solman\_topology.py
- saputils.py

## 検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Database
- Dependency
- IpAddress
- J2eeCluster
- Membership
- Node
- SapAbapApplicationServer
- SapClient

- SapJ2eeApplicationServer
- SapJ2eeCentralServices
- SapSystem
- Usage

### アダプタ・パラメータ

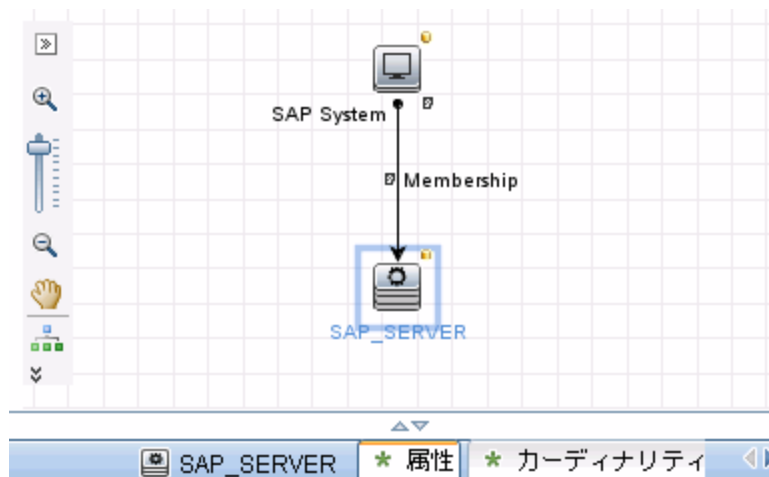
- **remoteJVMArgs**。リモート・プロセスに渡される必要がある JVM パラメータを示します。
- **runInSeparateProcess**。true の場合、パターンは個別のスレッドで実行されます。

## SAP Solution Manager by SAP JCO ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

- トリガ CI :SapAbapApplicationServer
- トリガ・クエリ:



NOT Reference to the credentials dictionary entry null AND NOT instance number null



## 使用スクリプト

- cmdlineutils.py
- command.py
- iteratortools.py
- sap.py
- sap\_abap.py
- sap\_abap\_discoverer.py
- sap\_discoverer.py
- sap\_solman\_discoverer.py
- sap\_solution\_manager.py
- saputils.py

## 検出された CIT

- Composition
- Containment
- IPAddress
- Membership
- Node
- SapAbapApplicationServer
- SapBusinessProcess
- SapBusinessScenario
- SapProcessStep

- SapProject
- SapSystem
- SapTransaction

### アダプタ・パラメータ

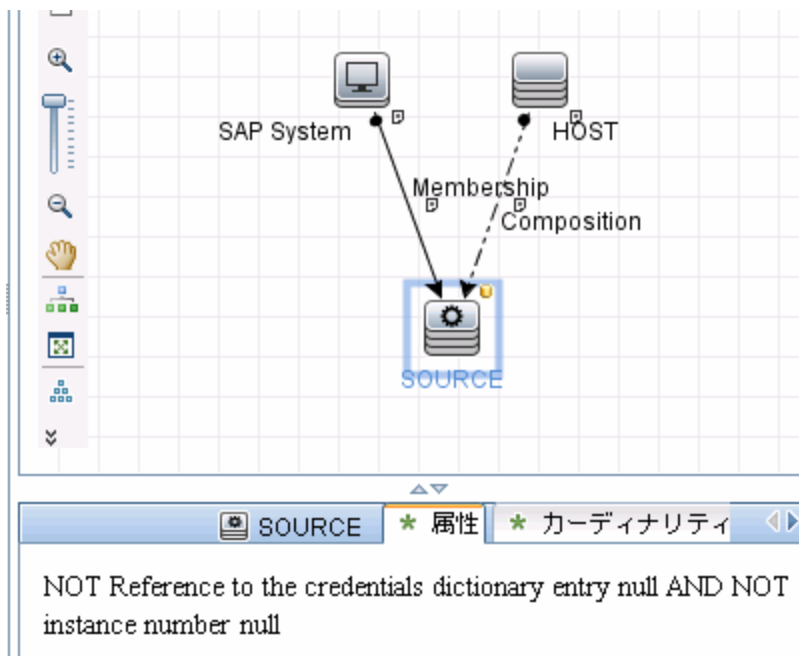
名 前	標準設定値	説明
discoverScenarioProcesses	false	シナリオ・プロセスを検出するかどうかを示します。
getProcessSteps	true	プロセス・ステップを取得するかどうかを示します。
remoteJVMArgs	-Xms256m -Xmx1024m - XX:PermSize=256m - XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータを示します。
reportCIsInChunks	false	CI をチャンクでレポートするかどうかを示します。
runInSeparateProcess	true	パターンを個別のスレッドで実行するかどうかを示します。

## SAP Applications by SAP JCO ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

- トリガ CI :SapAbapApplicationServer
- トリガ・クエリ:



## 使用スクリプト

- sapapputils.py
- saputils.py
- sap\_applications.py

## 検出された CIT

- Composition
- Containment
- SapApplicationComponent
- SapSystem
- SapTransaction
- SapTransport
- SapTransportChange

- Usage

## アダプタ・パラメータ

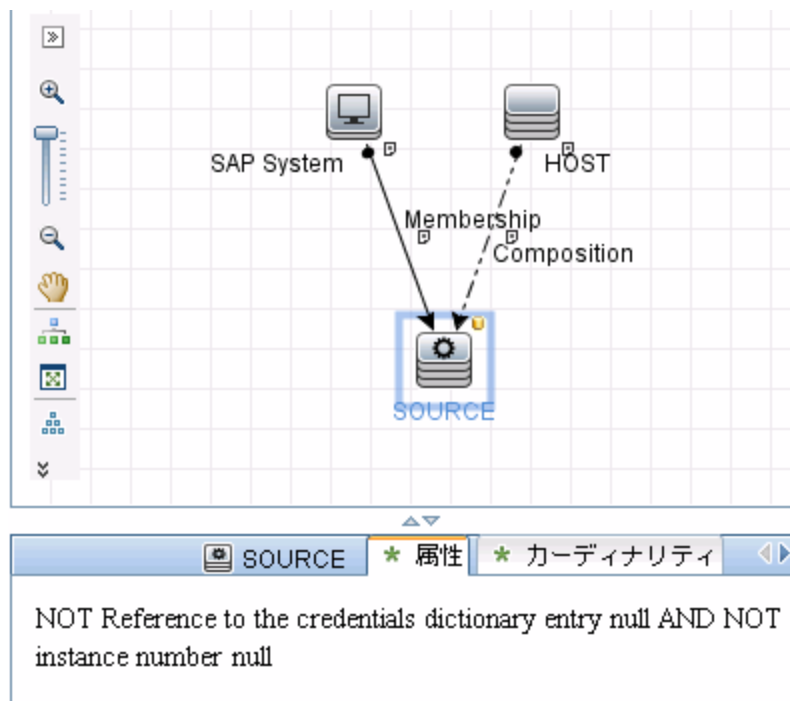
名 前	標準設定値	説明
getActiveTransactions	false	アクティブなトランザクションおよびアグリゲーションされた統計情報を取得するかどうかを示します。
getAllTransactions	false	すべてのトランザクションを取得するかどうかを示します。
getAppComponents	true	SAP アプリケーション・コンポーネントを取得するかどうかを示します。
getTransChanges	false	変更および転送システムを使用して、トランザクションの変更を取得するかどうかを示します。
remoteJVMArgs	-Xms256m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータを示します。
runInSeparateProcess	true	パターンを個別のスレッドで実行するかどうかを示します。
transChangesDaysInterval	0	今日から以前に遡る指定日数で、トランザクションの変更を取得します。
transChangesFromDate	01/25/2010	この日付以降のトランザクションの変更を取得します。
transChangesFromTime	00:00:00	この時刻以降のトランザクションの変更を取得します。
transChangesToDate	01/25/2010	この日付までのトランザクションの変更を取得します。
transChangesToTime	23:00:00	この時刻までのトランザクションの変更を取得します。

## SAP ABAP Topology by SAP JCO ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ

- トリガ CI :SapAbapApplicationServer
- トリガ・クエリ:



## 使用スクリプト

- iteratorutils.py
- cmdlineutils.py
- entity.py
- db\_platform.py
- db.py
- db\_builder.py
- sap\_db.py

- jdbc\_url\_parser.py
- jdbc.py
- saputils.py
- command.py
- sap.py
- flow.py
- sap\_flow.py
- sap\_abap.py
- sap\_discoverer.py
- sap\_abap\_discoverer.py
- sap\_site\_by\_jco.py

## 検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Database
- Dependency
- IpAddress
- JdbcDataSource
- Membership
- Node
- RfcConnection

- RunningSoftware
- SapAbapApplicationServer
- SapClient
- SapGateway
- SapSystem
- SapWorkProcess
- Usage

### アダプタ・パラメータ

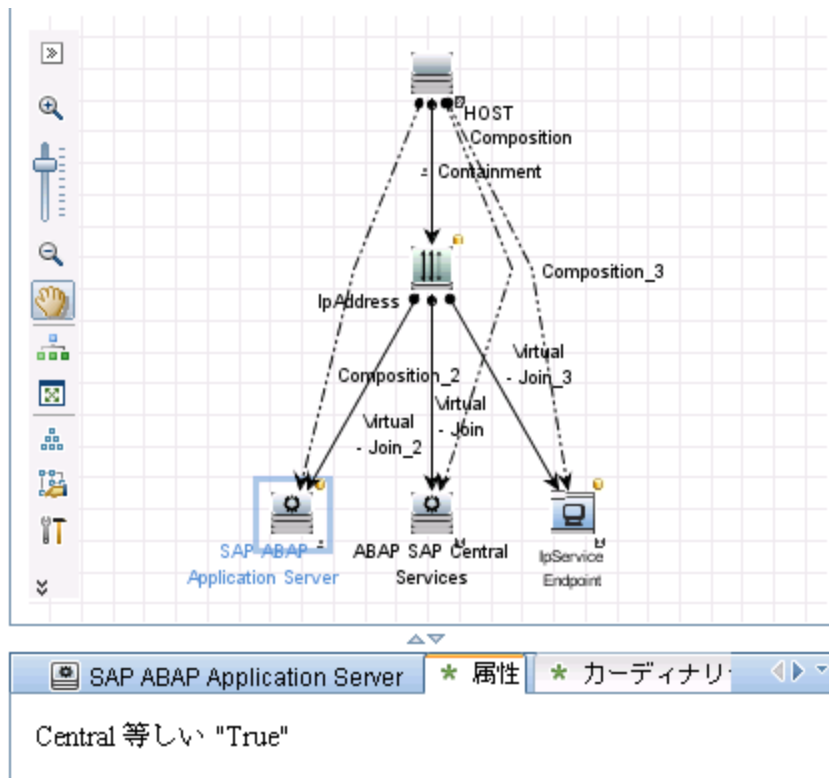
- **discoverRFCConnections**。 **true** の場合、このジョブは SAP システムの RFC 接続を取得します。
- **discoverSAPProfiles**。 **true** の場合、このジョブは SAP データベースに対してクエリを実行することで SAP プロファイルを検出します。
- **remoteJVMArgs**。リモート・プロセスに渡される必要がある JVM パラメータを示します。
- **reportComponentsAsConfigFile**。 **true** の場合、このジョブはソフトウェア・コンポーネントをレジストリ (構成ファイル) としてレポートします。 **false** の場合、このジョブはコンポーネントごとに個別の CI をレポートします。標準設定値は **true** です。
- **runInSeparateProcess**。 **true** の場合、パターンは個別のスレッドで実行されます。

## SAP ABAP Connection by SAP JCO ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

- トリガ CI : IpAddress
- トリガ・クエリ:



## 使用スクリプト

- iteratortools.py
- flow.py
- sap\_flow.py
- entity.py
- cmdlineutils.py
- saputils.py
- command.py
- sap.py
- sap\_abap.py



- sap\_discoverer.py
- sap\_abap\_discoverer.py
- sap\_system\_dis.py

## 検出された CIT

- Composition
- Containment
- IpAddress
- Membership
- Node
- SapAbapApplicationServer
- SapSystem

## アダプタ・パラメータ

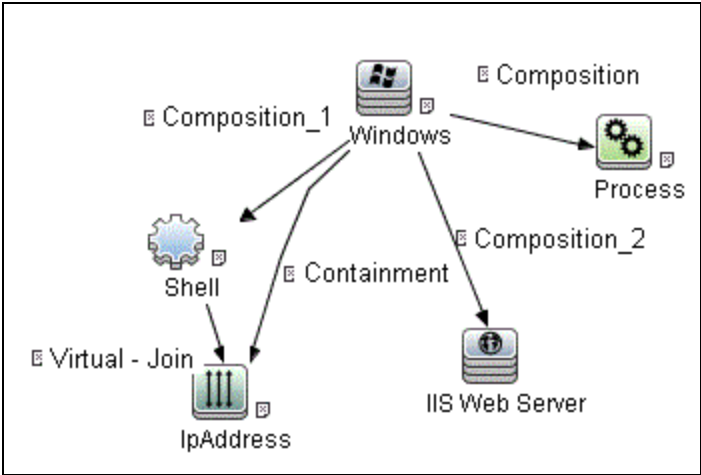
- **remoteJVMArgs**。リモート・プロセスに渡される必要がある JVM パラメータを示します。
- **runInSeparateProcess**。true の場合、パターンは個別のスレッドで実行されます。

# SAP ITS by NTCMD or UDA ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ

- トリガ CI :IIS Web Server
- トリガ・クエリ:



ノード名	条件
Process	Name 類似 %mmanager% AND NOT Process Path null
Shell	NOT Reference to the credentials dictionary entry null AND(CI Type 等しい ntcmd OR CI Type 等しい uda)
IpAddress	NOT IP Probe Name null

使用スクリプト

- cmdlineutils.py
- command.py
- entity.py
- iteratortools.py
- sap.py
- sap\_abap.py
- sap\_its.py

- saputils.py

## 検出された CIT

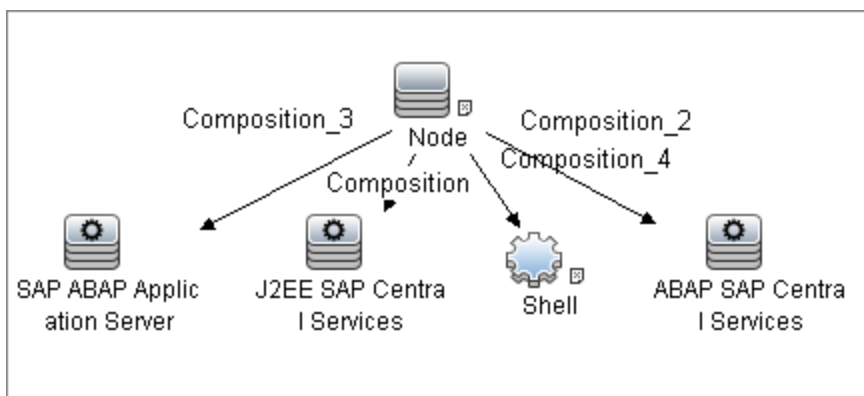
- Composition
- Containment
- Dependency
- IpAddress
- Node
- SapAbapApplicationServer
- SapItsAGate
- SapItsWGate
- WebServer

## SAP System by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。このジョブの使用は推奨されないため、[ディスカバリ モジュール] > [Tools and Samples] > [Deprecated Jobs] に移動されています。

### トリガ・クエリ

- トリガ CI :SapApplicationServer
- トリガ・クエリ:



## 使用スクリプト

- cmdlineutils.py
- entity.py
- sap.py
- sap\_discoverer.py
- sap\_system\_by\_shell.py

## 検出された CIT

- Composition
- ConfigurationDocument
- Membership
- SapApplicationServer
- SAP System
- Usage

## SAP TCP Ports ジョブ

本項ではこのジョブに関する詳細を説明します。

## アダプタ

このジョブでは、**TCP Ports Discovery** アダプタを使用します。

## トリガ・クエリ



## ノード条件

ノード名	条件
IpAddress	NOT IP Probe Name null

## ジョブ・パラメータ

名前	標準設定値	説明
checkIfIpsIsReachable	true	このフラグは、ホストのポートが利用可能かどうかの確認をジョブが開始する前に、検出された IP に到達可能であることを確認する必要があるかどうかを示します。
checkOnlyKnownPorts	true	このフラグは、既知のポートのみをこのジョブで検出するかどうかを示します。このフラグによって、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータがキャンセルされることはありません。このフラグの <b>false</b> への設定は、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータ内の実際のポート範囲と合わせて使用する場合にのみ適用されます。
connectTimeOut	5000	IP およびポートに接続するときのタイムアウトをミリ秒単位で示します。
nmapPath		nmap 実行可能ファイルのフルパスを示します (例 : <b>C:\Program Files (x86)\Nmap\nmap.exe</b> )。
pingTimeOut	2000	ICMP ping のタイムアウトをミリ秒単位で示します。

名前	標準設定値	説明
<b>ports</b>	<p><b>JEE TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>weblogic, weblogicSSL, websphere_jmx, rmi</li> </ul> <p><b>Databases TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>oracle, db2, sybase, sql, mysql</li> </ul> <p><b>SAP TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>sap, sap_jmx, sap_http, sap_https</li> </ul> <p><b>SAP TCP Ports ジョブの場合 : 標準設定値なし</b></p>	<p>このパラメータには、ディスカバリの実行に使用される TCP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の TCP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、<b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。</p>
<b>scanUDP</b>	false	<p>このフラグは、UDP ポートをスキャンするかどうかを示します。</p> <p><b>注:</b> UDP スキャンは、<b>useNMap=true</b> に設定されている場合にのみサポートされます(下記を参照)。</p>
<b>UDPports</b>		<p>このパラメータには、ディスカバリの実行に使用される UDP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の UDP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、<b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。</p>

名前	標準設定値	説明
<b>useNMap</b>	<b>Databases TCP Ports ジョブおよび JEE TCP Ports ジョブの場合:</b> false  <b>SAP TCP Ports ジョブおよび TCP Ports ジョブの場合:</b> true	このフラグは、ポートのスキャン中に nmap を使用するかどうかを示します。  <div> <b>注:</b> <code>nmapPath</code> にパスが指定されないと(上記を参照), システム・パスからの nmap が使用されます。         </div>

**注:** `ports` パラメータまたは `UDPports` パラメータ内でポート名が割り当てられており, `portNumberToPortName.xml` 構成ファイル内で検出可能(`isDiscovered=1`)としてマークされているポートのみが検出対象になります。

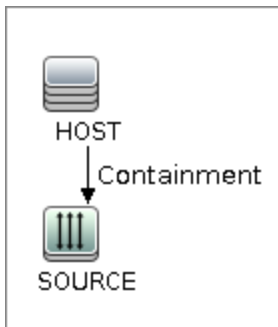
## アダプタ情報

このアダプタでは、TCP ポートを検出します。

### 入力 CIT

IpAddress

### 入力クエリ



### トリガ CI データ

名前	値
<b>ip_address</b>	<code>\${SOURCE.name}</code>

名前	値
ip_domain	\${SOURCE.routing_domain}

## 使用スクリプト

- TcpPortScanner.py
- nmap.py

## グローバル構成ファイル

portNumberToPortName.xml

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node

## トラブルシューティングおよび制限事項

- **問題** : SAP ディスカバリが失敗し、次の Java メッセージが表示される

This application has failed to start because MSVCR71.dll was not found.

**解決策** : 2 つの .dll ファイルが見つかりません。これを解決するには、ノート #684106 を参照してください([https://websmp205.sap-ag.de/~form/sapnet?\\_FRAME=CONTAINER&\\_OBJECT=012003146900000245872003](https://websmp205.sap-ag.de/~form/sapnet?_FRAME=CONTAINER&_OBJECT=012003146900000245872003))。

- **問題** : SAP ABAP ディスカバリ・ジョブで、SAP Java コネクタ・ドライバがインストールされているにもかかわらず、SAP ドライバが見つからないことを示すエラーが発生して失敗する



**解決策 1:** 標準設定では、Discovery Probe は JCo 3 ドライバを使用して接続を試みますが、これらのドライバはインストールされていません。そのため、JCo 3.x ドライバをインストールしてください。

**解決策 2:** 標準設定では、Discovery Probe は JCo 3 ドライバを使用して接続を試みますが、SAP システムは JCo 3 をサポートしていません。これを解決するには、[**Data Flow Probe 設定**]に移動し、**SAP Protocol** の必要な権限を右クリックします。[**旧 インタフェースを使って編集**]を選択し、**JCo version** を **2.x** に変更します。これがすでに選択されている場合でもこの操作を行い、権限を保存します。

- **問題:** SAP JCo dll ファイルの読み込みがスムーズに行われない

**解決策:** Microsoft Visual Studio C/C++ ランタイム再頒布可能パッケージの最新バージョンがインストールされていることを確認します。

## 第41章: SAP Java ディスカバリ

### 本章の内容

概要 .....	639
サポートされるバージョン .....	639
トポロジ .....	640
SAP Java 関連トポロジ全体の検出方法 .....	640
SAP Java Topology by HTTP ジョブ .....	642
SAP Java Topology by SAP JMX ジョブ .....	644
SAP Java Topology by WebServices ジョブ .....	647
SAP J2EE Topology by HTTP アダプタ .....	649
SAP J2EE Topology by JMX アダプタ .....	651
トラブルシューティングおよび制限事項 .....	654

## 概要

UCMDB は SAP アプリケーション Java サーバを検出します。このサーバによって、Java Enterprise Edition プログラムを開発および実行するための Java 2 Enterprise Edition(J2EE)環境が提供されます。

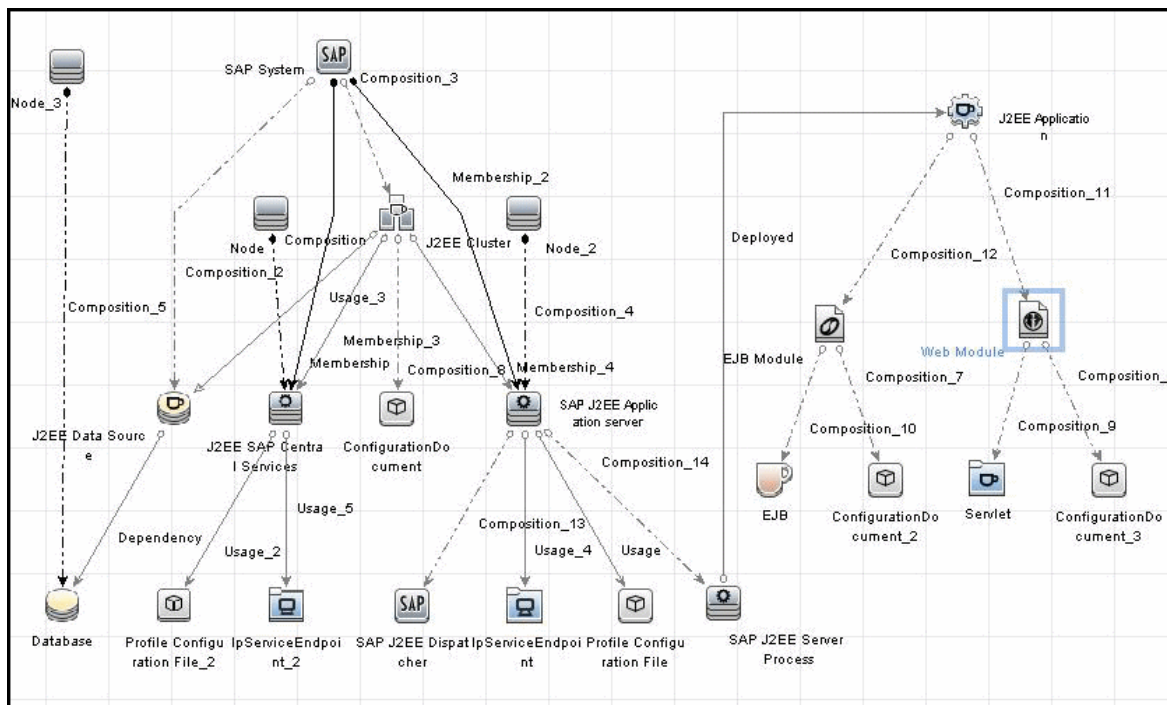
注: SAP Solution Manager への接続を検出することによって、SAP システム全体を検出できます。この方法で、単一の資格情報セットを作成しますので、SAP システムごとに1つの資格情報セットを作成する必要がありません。Universal Discovery はこの1つのセットですべてのシステム(およびトポロジ)を検出します。詳細については、[「SAP Solution Manager ディスカバリ」\(656ページ\)](#)を参照してください。

## サポートされるバージョン

SAP BASIS および SAP AS(アーキテクチャ・レイヤ)	バージョン 3.x ~ 6.x
SAP J2EE クライアント	このバージョンは、関連 SAP システムのバージョンと一致している必要があります。
SAP Solution Manager	バージョン 6.x, 7.x

## トポロジ

次の図に、SAP Java トポロジの要素を示します。

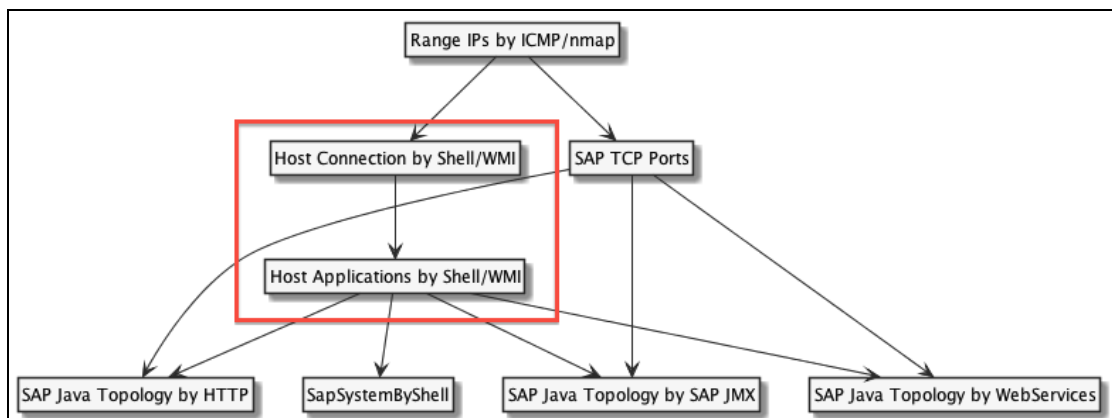


## SAP Java 関連トポロジ全体の検出方法

SAP Java ディスカバリを使用すると、SAP Java 関連トポロジ全体と、SAP JAVA サーバ上の J2EE アプリケーションを検出できます。

このディスカバリの実行のために実行する必要があるジョブは、シェル・アクセスがSAPの宛先に提供されているかどうかによって判断されます。シェル・アクセスが提供されている場合、[「SAPの宛先にシェル・アクセスが提供されているSAP Java ディスカバリ」\(641ページ\)](#)に記載されているようにディスカバリを実行します。シェル・アクセスがSAPの宛先に提供されていない場合、[「SAPの宛先にシェル・アクセスが提供されていないSAP Java ディスカバリ」\(642ページ\)](#)に記載されているようにディスカバリを実行します。

次の図に、SAP Java ディスカバリを実行するための各種シナリオを示します。



## SAP の宛先にシェル・アクセスが提供されている SAP Java ディスカバリ

**注:** SAP の宛先にシェル・アクセスが提供されている場合、SAP Java ディスカバリはシャロー・モードまたはディープ・モードで実行できます。シャロー・ディスカバリを実行するには、次の手順 1~3 を実行します。ディープ・ディスカバリを実行するには、次のすべての手順を実行します。

SAP Java ディスカバリを実行するには、次の手順を実行します。

1. [Universal Discovery] ウィンドウで、**Range IPs by ICMP** ジョブを実行します。このジョブは、到達可能なすべての IP を検出します。
2. **Host Connection By Shell** ジョブを実行します。このジョブは、アプリケーション・サーバへのシェル・プロトコルによって、ホストの接続を検出します。
3. **Host Applications by Shell** ジョブを実行します。このジョブは、アプリケーション・サーバ、メッセージ・サーバおよびエンキュー・サーバ、それらの SAP システムへの関係を検出します。

**注:** シャロー・ディスカバリのみを実行する場合、ここで終了し、次の手順に進む必要はありません。

4. 次のジョブの 1 つまたは複数を実行します(検出する Java トポロジのタイプに応じて異なる)。
  - **SAP Java Topology by HTTP**。このジョブは、使用されているデータベースを含む SAP Java トポロジ全体を検出します。システム構成は、スタンドアロン Java または二重スタックのいずれかに特定されます。
  - **SAP Java Topology by SAP JMX**。このジョブは、インスタンスに関する詳細を含む SAP Java トポロジ全体を検出します。データベースは、新しい MBean モデルが利用可能な場合 (Java アプリケーション・サーバ・バージョン 7.1 以降) にのみ検出されます。システム構成のタイプを特定す

ることはできません。

- **SAP Java Topology by WebServices。**このジョブは、JMX ジョブと同じですが、異なる転送を使用します。
- **SAP System by Shell**

## SAP の宛先にシェル・アクセスが提供されていない SAP Java ディスカバリ

SAP Java ディスカバリを実行するには、次の手順を実行します。

1. [Universal Discovery] ウィンドウで、**Range IPs by ICMP** ジョブを実行します。このジョブは、到達可能なすべての IP を検出します。
2. **SAP TCP Ports** ジョブを実行します。
3. 次のジョブの 1 つまたは複数を実行します(検出する Java トポロジのタイプに応じて異なる)。
  - **SAP Java Topology by HTTP。**このジョブは、使用されているデータベースを含む SAP Java トポロジ全体を検出します。システム構成は、スタンドアロン Java または二重スタックのいずれかに特定されます。
  - **SAP Java Topology by SAP JMX。**このジョブは、インスタンスに関する詳細を含む SAP Java トポロジ全体を検出します。データベースは、新しい MBean モデルが利用可能な場合 (Java アプリケーション・サーババージョン 7.1 以降) にのみ検出されます。システム構成のタイプを特定することはできません。
  - **SAP Java Topology by WebServices。**このジョブは、**SAP Java Topology by SAP JMX** ジョブと同じですが、異なる転送を使用します。

## SAP Java Topology by HTTP ジョブ

このジョブは、HTTP URL クエリによる XML から解析されたデータに基づきます。URL は、アプリケーション・サーバ(インスタンス)がある場所のシステム構成に応じて若干異なります。このジョブが使用する構成の主な 2 種類は次のとおりです。

1. **ピュアな、またはスタンドアロンの Java システム構成** : 次の 2 種類のクエリがあります。
  - **http(s)://<アドレス>:<ポート>/sap/monitoring/SystemInfoServlet**(システム全体に関する情報を照会するために使用)
2. **二重スタック(DS)構成** : 次の 2 種類のクエリがあります。

- **http(s)://<アドレス>:<ポート>/monitoring/SystemInfoServlet**(システム全体に関する情報を照会するために使用)

## アダプタ

ID : sap\_java\_topology\_by\_http

## トリガ TQL

このジョブは、エンドポイントが sap\_http または sap\_https のホストに属する IP でトリガされます。

## パラメータ

なし

## 前提条件

1. **資格情報の設定。**HTTP クライアントは、**SAP Java Topology by SAP JMX** ジョブに使用されるのと同じ資格情報を使用します。
2. **プローブ上へのライブラリのインストール。**外部ライブラリをインストールする必要はありません。必要なすべての依存関係は、プローブのインストールに付属しています。

## ディスカバリ・フロー

SAP Java Topology by HTTP ジョブのディスカバリ・フローは次のとおりです。

1. この宛先に利用できる SAP JMX 資格情報を取得します。

**注:** 資格情報に指定されたポートが宛先ポートと異なっても、そのような資格情報も使用されます。

2. このジョブは、各資格情報について、上述の **SystemInfoServlet** URL を使用して HTTP クエリの実行を試みます。正常な出力結果は XML であり、次の SAP システム詳細を取得するために解析されます。
  - システム ID(XML : /**SAP\_J2EE/SID/@value** から)
  - システム・インストール・タイプ(XML : /**SAP\_J2EE/INSTALLATION\_TYPE/@value** から)
  - 次から構成されるバージョン :

- `/SAP_J2EE/REL/@value`
- `/SAP_J2EE/PATCH_LEVEL/@value`
- データベース情報 (XML: `/SAP_J2EE/DB_INSTANCE` から)
- SCS (XML: `/SAP_J2EE/SCS_INSTANCE` から)

**注:** SCS データには、ポートとホスト情報のみが含まれ、インスタンス番号がないために完全ではありません。そのため、このジョブはインスタンス番号をポートから計算します。3xNN, 81NN(http), 444NN(https)などのいくつかのポート・パターンが考慮されます。ここで NN はインスタンス番号を意味します。

- ダイアログ・インスタンス (XML: `/SAP_J2EE/DIALOG_INSTANCE` から)

**注:** `/SAP_J2EE/DIALOG_INSTANCE/INSTANCE_DIR` および `/SAP_J2EE/DIALOG_INSTANCE/NAME` には無効なデータが含まれている可能性があるため、これらは検出アルゴリズムから意図的に除外されます。

- セントラル・インスタンス (XML: `/SAP_J2EE/CENTRAL_INSTANCE` から)

レポートダイアログとセントラル・インスタンスには、インスタンス名の解決のためのその他いくつかのロジックが含まれます。インスタンス名を取得するため、このジョブはすべてのワーカ(サーバ)の情報と、`application.home`, `com.sap.jvmdir`, `java.home`, `rdbms.driverLocation`, `user.dir`などのランタイム・プロパティを取得するためのデイスパッチ機能を収集します。このジョブは、対応するSIDと、それまでに検出されたパスからの有効なインスタンス名を使用して、パスの検出を試みます。通常、`application.home`にそのような情報が含まれています。

- ソフトウェア・コンポーネント (XML: `/SAP_J2EE/SOFTWARE_COMPONENTS/COMPONENT` から)

## SAP Java Topology by SAP JMX ジョブ

このジョブは、アプリケーション・サーバの MBean サーバに P4 ポート (5xx04) を介してアクセスするために SAP Java クライアントを使用します。

**注:** P4 ポートは通常閉じられており、Java アプリケーション・サーバ・バージョン 7.x 以降 **SAP Java Topology by WebServices** ジョブまたは **SAP Java Topology by HTTP** ジョブ、またはこれらの両



方を使用することが推奨されるようになりました。

## アダプタ

ID : SAP\_Dis\_J2EE\_Site

## トリガ TQL

このジョブは、エンドポイントが sap\_jmx のホストに属する IP でトリガされます。

## 前提条件

### 1. 資格情報の設定。

Java クライアント・ディスカバリが使用する資格情報を設定するには、**SAP JMX** という独自の資格情報を提供する必要があります。**SAP JMX** 資格情報によって、マシンへの接続を有効にし、SAP システムがマシン上にインストールされているかどうかを確認できます。

### 2. DataFlowProbe マシンに Java クライアント(jar ファイル)を追加します。

**注:** DataFlowProbe マシンの `j2ee\sap` ディレクトリの下にバージョンのフォルダを作成すると、各フォルダに `.jar` ファイルを追加することで、SAP の複数のバージョンに接続できます。たとえば、バージョン SAP 6.4 および 7.0 に接続するには、SAP フォルダに 6.x と 7.x の2つのサブフォルダを作成し、これらのフォルダに適切な `.jar` ファイルを配置します。

a. 次の `.jar` ファイルを Data Flow Probe マシンの `<DataFlowProbe_root>\runtime\probeManager\discoveryResources\j2ee\sap` ディレクトリに追加します。

- sapj2eeclient.jar
- logging.jar
- exception.jar
- sapxmltoolkit.jar

これらのファイルは、いずれかの SAP インスタンス・マシン上の `\usr\sap\<SID>\<インスタンス名>\j2ee\j2eeclient` ディレクトリにあります。

b. `com_sap_pj_jmx.jar` ファイルを Data Flow Probe マシンの `<DataFlowProbe_`

**root>\runtime\probeManager\discoveryResources\j2ee\sap** ディレクトリに追加します。

このファイルは、いずれかの SAP インスタンス・マシン上の **\usr\sap\<SID>\<インスタンス名>\j2ee\admin\lib** ディレクトリにあります。

## ディスカバリ・フロー

1. この宛先用に利用可能な資格情報を取得し、ポートを持たない、または宛先で開かれているポートのリストに属するポートを使用する資格情報のみが使用されるようにフィルタで除外します。
2. 各資格情報に対し、このジョブは宛先への接続を確立しようと試みます。
3. 接続の確立後は、このジョブは **SAP\_J2EECluster** クエリを送信し、クラスタの詳細を取得します。このクエリは、次の情報を返します。

- **Name** : この属性には、クラスタまたはシステム名が含まれます。
- **InstanceNames** : この属性には、このシステムで利用可能な Java アプリケーション・サーバの名前が含まれます。

このクエリ後、初期トポロジがレポートされます。

4. 各インスタンスについての詳細を取得するため、このジョブは同じ MBean に対して **SAP\_J2EECluster** クエリを繰り返しますが、**AllInstanceInfos** 属性を追加します。ホスト名およびインスタンス番号は、**Name** 値に基づいて特定されます。
5. **reportComponentsAsConfigFile** パラメータが **true** に設定されている場合、このジョブは開発コンポーネントを検出します。インタフェース、ライブラリ、サービスに対してクエリを実行するため、次の各 MBean が使用されます。
  - **SAP\_J2EEInterfacePerNode**
  - **SAP\_J2EELibraryPerNode**
  - **SAP\_J2EEServicePerNode**
6. このジョブは、**SAP\_J2EEInstance** の **name** が **SCS** に設定されたクエリを使用して、**SCS**(セントラル・サービス)アプリケーション・サーバを検出します。
7. このジョブは、**SAP\_J2EEClusterNode** クエリを使用して、1 つ以上のワークに関する情報とディスパッチ機能を検出します。

8. このジョブは、SAP\_ITSAMJ2eeCluster クエリを使用してデータベースの検出を試みます。ただし、これは新しい MBean モデル(バージョン 7.1 以降)にしか機能しません。

## SAP Java Topology by WebServices ジョブ

このジョブは、**SAP Java Topology by SAP JMX** ジョブと同じ MBean モデルに基づいていますが、SAP WebServices 転送を使用します。このジョブは、**SAP Java Topology by SAP JMX** ジョブと同じコードベースも共有し、逆シリアル化の制限があるために、ディスカバリ・フローは若干異なります。

### アダプタ

ID : SAP\_Dis\_J2EE\_Site

### トリガ TQL

このジョブは、エンドポイントが **sap\_http** のホストに属する IP でトリガされます。

### パラメータ

- **remoteJVMArgs**。リモート・プロセスに渡される必要がある JVM パラメータを示します。
- **remoteJVMClasspath**。このパラメータは、リモート・プロセスの **classpath** の追加のライブラリを指定します。このプロセスでは **wsconnector** ライブラリが必要です。標準設定値は **%minimal\_classpath%;../lib/wsconnector.jar** です。
- **reportComponentsAsConfigFile**。true の場合、このジョブは Java システム・コンポーネントをレジストリ(構成ファイル)としてレポートします。false の場合、このジョブはコンポーネントごとに個別の CI をレポートします。標準設定値は **true** です。
- **runInSeparateProcess**。このパラメータは、パターンを個別のスレッドで実行するかどうかを示します。標準設定値は **true** です。

### 前提条件

#### 1. 資格情報の設定

WebServices クライアントは、**SAP Java Topology by SAP JMX** ジョブに使用されるのと同じ資格情報を使用します。

## 2. プロブ上へのライブラリのインストール

外部ライブラリをインストールする必要はありません。必要なすべての依存関係は、プロブのインストールに付属しています。

## ディスカバリ・フロー

このジョブのフローは、**SAP Java Topology by SAP JMX** ジョブのフローとよく似ていますが、システム情報全体を検出するときに一部が異なります。

**注:** 逆シリアル化の制限があるため、検出されたインスタンスの一部は、**instance\_name** 情報が利用可能でないためにこの情報なしでレポートされます。

SAP Java Topology by WebServices ジョブのディスカバリ・フローは次のとおりです。

1. この宛先用に利用可能な資格情報を取得し、ポートを持たない、または宛先で開かれているポートのリストに属するポートを使用する資格情報のみを使用されるようにフィルタで除外します。
2. 各資格情報に対し、このジョブは WebServices を使用する宛先への接続を確立しようと試みます。
3. 接続の確立後、このジョブは、クラスタの詳細を取得するために **SAP\_J2EECluster** クエリを送信することによって、アプリケーション・サーバの名前ですべてのアプリケーション・サーバを検出します。このクエリは、次の情報を返します。
  - **Name** : この属性には、クラスタまたはシステム名が含まれます。
  - **InstanceNames** : この属性には、このシステムで利用可能な Java アプリケーション・サーバの名前が含まれます。

このクエリ後、初期トポロジがレポートされます。

**注:** この時点で、WebServices クライアントに逆シリアル化の制限があるために、このジョブはインスタンスの詳細を検出できません。

4. **reportComponentsAsConfigFile** パラメータが **true** に設定されている場合、このジョブは開発コンポーネントを検出します。インタフェース、ライブラリ、サービスに対してクエリを実行するため、次の各 MBean が使用されます。

- **SAP\_J2EEInterfacePerNode**
  - **SAP\_J2EELibraryPerNode**
  - **SAP\_J2EEServicePerNode**
5. このジョブは、**SAP\_J2EEInstance** の **name** が **SCS** に設定されたクエリを使用して、SCS(セントラル・サービス)アプリケーション・サーバを検出します。
  6. このジョブは、**SAP\_J2EEClusterNode** クエリを使用して、1 つ以上のワーカに関する情報とディスパッチ機能を検出します。
  7. このジョブは、**SAP\_ITSAMJ2eeCluster** クエリを使用してデータベースの検出を試みます。ただし、これは新しい MBean モデル(バージョン 7.1 以降)にしか機能しません。

## SAP J2EE Topology by HTTP アダプタ

本項ではこのアダプタに関する詳細を説明します。

### ID

sap\_java\_topology\_by\_http

### 入力 CIT

IpServiceEndpoint

### 入力 TQL

SAP Java Topology by HTTP ジョブは、IpServiceEndpoint に sap\_http または sap\_https という名前があるときにトリガされます。

### トリガ CI データ

- **ip\_service\_name**。IP サービス・エンドポイントの名前
- **ip\_port\_pair**。エンドポイントを表す文字列であり、**address:port** と類似

### 使用スクリプト

エントリ・ポイント・モジュールは sap\_jee\_topology\_by\_http.py です。

- jmx.py
- sap.py
- db\_platform.py
- db.py
- db\_builder.py
- sap\_db.py
- sap\_discoverer.py
- jdbc\_url\_parser.py
- jdbc.py
- jee.py
- sap\_jee.py
- sap\_jee\_discoverer.py
- sap\_jee\_topology\_by\_http.py

## 検出された CIT

- Database
- IpAddress
- IpServiceEndpoint
- J2eeCluster
- SapJ2eeApplicationServer
- SapJ2eeCentralServices
- SapJ2eeDispatcher
- SapJ2eeServerProcess

- SapJavaSoftwareComponent
- SapSystem

## パラメータ

なし

# SAP J2EE Topology by JMX アダプタ

本項ではこのアダプタに関する詳細を説明します。

## ID

SAP\_Dis\_J2EE\_Site

## 入力 CIT

IpAddress

## 入力 TQL

SAP Java Topology by SAP JMX ジョブは、IpAddress がそのホストに属しており、IpServiceEndpoint が同じアドレスにある場合にトリガされます。IpServiceEndpoint の名前は、sap\_jmx または sap\_http のいずれかである必要があります。

## トリガ CI データ

- **hostId**。UCMDB ID の宛先
- **ip\_address**。IP アドレスの宛先
- **ip\_domain**。ドメインの宛先
- **sap\_jmx\_port**。エンドポイントの省略可能なポート

## 使用スクリプト

エントリ・ポイント・モジュールは sap\_jee\_topology\_by\_jmx.py です。

- `iteratortools.py`
- `entity.py`
- `cmdlineutils.py`
- `command.py`
- `sap.py`
- `jee.py`
- `jmx.py`
- `sap_jee.py`
- `db_platform.py`
- `db.py`
- `db_builder.py`
- `jdbc.py`
- `jdbc_url_parser.py`
- `sap_db.py`
- `sap_discoverer.py`
- `sap_jee_discoverer.py`
- `flow.py`
- `sap_flow.py`
- `sap_jee_topology_by_jmx.py`

## 検出された CIT

- `Composition`
- `ConfigurationDocument`



- Containment
- Database
- Dependency
- Deployed
- Ejb
- EjbModule
- EntityBean
- IpAddress
- IpServiceEndpoint
- J2EE Domain
- J2eeApplication
- J2eeCluster
- JdbcDataSource
- Membership
- MessageDrivenBean
- Node
- RunningSoftware
- SapJ2eeApplicationServer
- SapJ2eeCentralServices
- SapJ2eeDispatcher
- SapJ2eeServerProcess
- SapJavaSystemComponents
- SapSystem

- Servlet
- StatefulSessionBean
- StatelessSessionBean
- Usage
- Web Module

## パラメータ

次の2つのパラメータは、このアダプタの最上位で定義されるジョブの実行方法を宣言します。

- **runInSeparateProcess**。このパラメータが `true` に設定されている場合、このジョブは個別のプロセス内で実行されます。標準設定値は `true` です。
- **remoteJVMArgs**。リモート・プロセスに渡される必要がある JVM パラメータが格納されます。

次のパラメータは、トポロジ・レポート・フローに影響します。

- **reportComponentsAsConfigFile**。 `true` に設定された場合、ジョブは Java 開発コンポーネントを構成ドキュメントとしてレポートします。 `false` に設定された場合、ジョブはコンポーネントごとに個別の CI をレポートします。標準設定値は `true` です。

## トラブルシューティングおよび制限事項

すべての前提条件を満たしたにもかかわらず、ディスカバリが「接続エラー」メッセージを返す場合、**RemoteProcesses.log** を DDM Flow Probe のログ・フォルダ (`C:\hp\UCMDB\DataFlowProbe\runtime\log`) から確認します。「`NoClassDefFoundError`」がここに記録されている場合、次の回避策に従います。

1. 次の SAP jar ファイルを `C:\hp\UCMDB\DataFlowProbe\content\lib\sap` フォルダにコピーします。
  - `sapj2eeclient.jar`
  - `logging.jar`
  - `exception.jar`

- sapxmltoolkit.jar
- com\_sap\_pj\_jmx.jar

sap フォルダが存在しない場合は、作成します。

2. Data Flow Probe を再起動します。

この回避策を実行する場合、1つのバージョンの SAP jar ファイルしか使用できません。

# 第42章: SAP Solution Manager ディスカバリ

## 本章の内容

- 概要 ..... 657
- サポートされるバージョン ..... 657
- トポロジ ..... 657
- SAP Solution Manager の検出方法 ..... 657
- トラブルシューティングおよび制限事項 ..... 659

## 概要

複数の SAP システムが組み込まれている環境の場合、それぞれが異なる資格情報セット(ユーザ名、パスワード、システム番号、クライアント番号など)を使用することがよくあります。

慣例として、すべての SAP システムは、SAP システムの管理を一元化するために SAP Solution Manager に登録されます。DFM では、SAP Solution Manager に対するこの接続を検出することによって、すべての SAP システムの検出が可能になります。この方法で、単一の資格情報セットを作成しますので、SAP システムごとに 1 つの資格情報セットを作成する必要がありません。DFM はこの 1 つのセットですべてのシステム(およびトポロジ)を検出します。

## サポートされるバージョン

SAP BASIS および SAP AS(アーキテクチャ・レイヤ)	バージョン 3.x ~ 6.x
SAP JCo	2.x および 3.x(3.0.7 以降) バージョン 3.0.7 以降を推奨
SAP Solution Manager	バージョン 6.x, 7.x

## トポロジ

SAP Solution Manager Topology by SAP JCo トポロジを表示するには、次の手順を実行します。

[Universal Discovery] > [Enterprise Applications] > [SAP] > [SAP Solution Manager Topology by SAP JCo] > [詳細] 表示 枠を選択します。[マップに CI を表示] ボタンをクリックします。

## SAP Solution Manager の検出方法

DFM は、SAP ビジネス・レイヤと、登録された SAP システムのトポロジ全体を検出します。これには次の手順が含まれます。

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリ・ソリューションは、SAP プロトコルをベースにしています。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - 権限の設定

SAP Solution Manager を実行するには、所定のプロファイルについて次のオブジェクトに対する権限を付与してもらうように、SAP Solution Manager 管理者に依頼します。

- **S\_RFC** オブジェクトに対し、次の権限を取得します。RFC1, SALX, SBDC, SDIF, SDIFRUNTIME, SDTX, SLST, SRFC, STUB, STUD, SUTL, SXMB, SXMI, SYST, SYSU, SEU\_COMPONENT

- **S\_XMI\_PROD** オブジェクトに対し、次の権限を取得します。

EXTCOMPANY=MERCURY;EXTPRODUCT=DARM;INTERFACE=XAL

- **S\_TABU\_DIS** オブジェクトに対し、次の権限を取得します。

DICBERCLS=SS; DICBERCLS=SC; DICBERCLS=&NC& ACTVT=03

## 3. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

### 方法 1:

- **SAP TCP Ports** ジョブを実行し、SAP ポートを検出します。
- **SAP ABAP Connection by SAP JCO** ジョブを実行します。
- **SAP Solution Manager Topology by SAP JCO** ジョブを実行し、登録された SAP システムのトポロジ全体を検出します。
- **SAP Solution Manager by SAP JCO** ジョブを実行し、SAP ビジネス・レイヤを検出します。

### 方法 2:

- **Host Resources by ...** ジョブを実行し、SAP(ABAP または J2EE)アプリケーション・サーバまたは SAP(ABAP または J2EE)セントラル・サービス、またはそれらの両方を検出します。
- **SAP System by Shell** ジョブを実行し、SAP system CI を検出します(ただし、その CI が SAP Solution Manager かどうかは定義しません)。

- **SAP ABAP Connection by SAP JCO** ジョブを実行します。
- **SAP Solution Manager Topology by SAP JCO** ジョブを実行し、登録された SAP システムのトポロジ全体を検出します。
- **SAP Solution Manager by SAP JCO** ジョブを実行し、SAP ビジネス・レイヤを検出します。

**SAP ABAP Connection by SAP JCO** ジョブの実行中、SAP Solution Manager として定義された SAP システムが、**SAP Solution Manager Topology by SAP JCO** ジョブおよび **SAP Solution Manager by SAP JCO** ジョブの2つのジョブでトリガされます。

## トラブルシューティングおよび制限事項

**問題 :** SAP ディスカバリが失敗し、次の Java メッセージが表示される

This application has failed to start because MSVCR71.dll was not found.

**解決策 :** 2つの .dll ファイルが見つかりません。これを解決するには、ノート #684106 を参照してください ([https://websmp205.sap-ag.de/~form/sapnet?\\_FRAME=CONTAINER&\\_OBJECT=012003146900000245872003](https://websmp205.sap-ag.de/~form/sapnet?_FRAME=CONTAINER&_OBJECT=012003146900000245872003))。

## 第43章: Siebel ディスカバリ

### 本章の内容

概要 .....	661
サポートされるバージョン .....	661
トポロジ .....	662
Siebel トポロジ・ビュー .....	662
Siebel Web トポロジ・ビュー .....	663
Siebel トポロジの検出方法 .....	664
Siebel Application Server Configuration ジョブ .....	666
Siebel Application Servers ジョブ .....	668
Siebel Gateway Connection ジョブ .....	669
Siebel Web Applications by NTCMD or UDA ジョブ .....	671
Siebel Web Applications by TTY ジョブ .....	672
Siebel DB by NTCMD or UDA ジョブ .....	675
Siebel DB by TTY ジョブ .....	676
トラブルシューティングおよび制限事項 .....	678



## 概要

Siebel アダプタを使用することで、HP Universal CMDB 内部に Siebel コンポーネントを備えた Siebel 環境を構築するための Siebel ディスカバリを自動的に実行できます。ディスカバリ中に次のことが実行されます。

- 組織内のすべての Siebel 関連 IT エンティティが検出され、構成アイテム(CI)が CMDB に書き込まれます。
- 要素間の関係が作成され、CMDB に保存されます。
- 新しく生成された CI は、Siebel Enterprises root CI の下のビュー・エクスプローラで Siebel エンタープライズ・ビューが選択されると表示されます。

**注:** すべての Siebel サーバ IP アドレスが、この範囲に含まれていることを確認します。すべてのサーバを1つの IP 範囲に含められない場合、範囲をいくつかの範囲に分割できます。

## サポートされるバージョン

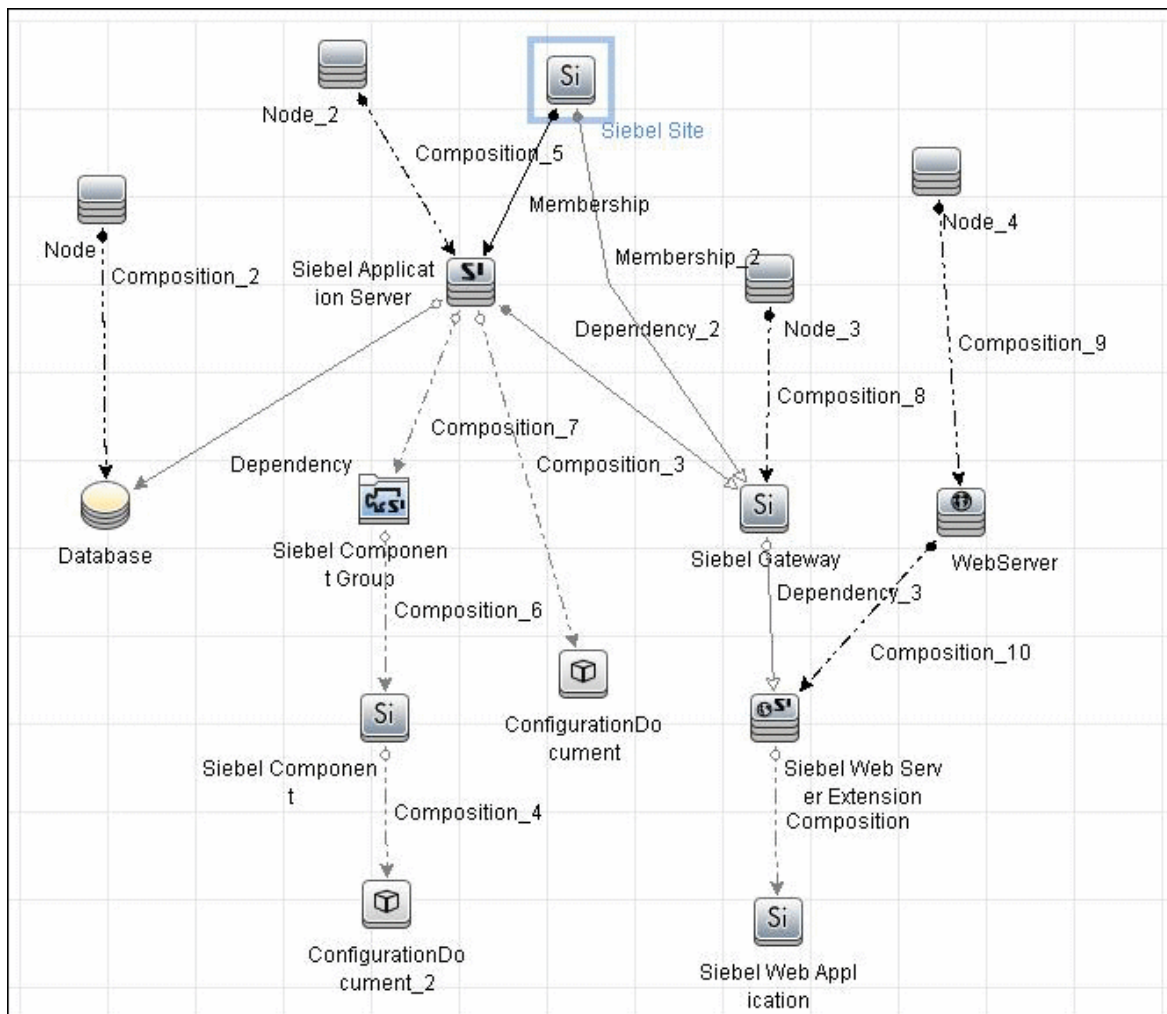
このディスカバリ・ソリューションは、次のサーバをサポートしています。

- Siebel 7.5
- Siebel 7.7
- Siebel 8.0
- Siebel 8.1

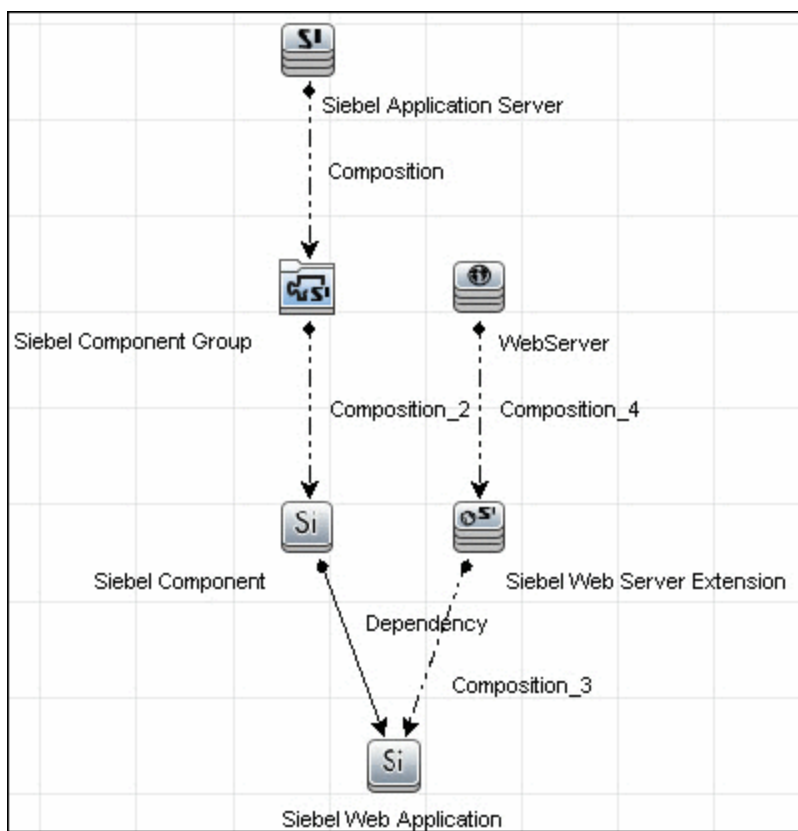
## トポロジ

次の図に、Siebel のトポロジを示します。

### Siebel トポロジ・ビュー



## Siebel Web トポロジ・ビュー



## Siebel トポロジの検出方法

このタスクでは、Siebel トポロジの検出方法について説明します。これには次の手順が含まれます。

- [「前提条件 - プロトコル資格情報の設定」\(664ページ\)](#)
- [「前提条件 - その他」\(664ページ\)](#)
- [「ディスカバリの実行」\(665ページ\)](#)

### 前提条件 - プロトコル資格情報の設定

次のプロトコルを設定します。

プラットフォーム	プロトコル
Windows	<ul style="list-style-type: none"><li>• WMI プロトコル</li><li>• NTCMD プロトコル</li><li>• Siebel Gateway プロトコル</li></ul>
UNIX	<ul style="list-style-type: none"><li>• SSH プロトコル</li><li>• Telnet プロトコル</li><li>• Siebel Gateway プロトコル</li></ul>

**注:** Siebel Gateway プロトコルを使用すると、ユーザはゲートウェイとの接続中に使用するポートを指定できます。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 前提条件 - その他

ドライバ・ツールが、Siebel からエンタープライズ構成に関するデータを抽出するために使用されます。

**注:**

- 組織内で複数のバージョンの Siebel を使用している場合、その Siebel サーバに適したバージョンのドライバ・ツールを使用していることを確認してください。

- Data Flow Probe が Windows プラットフォームの 64 ビット・マシンにインストールされている場合、**ntdll.dll**, **MSVCR70.DLL**, **msvcrt70.dll** の各ドライバを Siebel ドライバと一緒に Probe マシンの Siebel ドライバ用フォルダに配置します。このフォルダの詳細を Siebel の資格情報セット (**[Siebel クライアントへのパス]**)に入力します。これらのドライバは通常 32 ビット・マシン上にあり、64 ビット・マシンにコピーできます。

詳細については、『HP Universal CMDB Data Flow Management Guide』の Siebel Gateway プロトコルに関する項を参照してください。

ドライバ・ツールを Data Flow Probe にコピーするには、次の手順を実行します。

1. ドライバ・コマンド・ライン・インタフェース(CLI)ツールを Siebel サーバから Data Flow Probe マシンの任意のフォルダにコピーします。
2. (推奨)ドライバのインストール状態を検証するため、Siebel 接続テストを実行します。接続テストを実行するには、Data Flow Probe マシンのコマンド・ラインを開き、ディレクトリを **driver.exe** ファイルの場所に変更します。
3. コマンド・ラインで次のコマンドを実行します。

```
>driver /e [site_name] /g [gateway_host] /u [ユーザ名] /p [パスワード]
```

接続が正常に確立されたら、コマンド・プロンプト・ウィンドウに driver プロンプトが表示され、接続されたサーバ数に関するステータス・メッセージが表示されます。

## ディスカバリの実行

1. Siebel ネットワーク機能のディスカバリをトリガするには、Network CI を CMDB に追加します。詳細については、『HP Universal CMDB モデリング・ガイド』の「[新規 CI / 新規関連 CI]ダイアログ・ボックス」を参照してください。
2. [Universal Discovery] ウィンドウで、次の順にジョブをアクティブ化します。
  - a. Range IPs by ICMP または Range IPs by nmap
  - b. Host Connection by Shell
  - c. Host Connection by WMI
  - d. Host Resources by Shell
  - e. Host Resources by WMI

- f. Host Applications by Shell
- g. Host Applications by WMI
- h. Siebel Web Applications by NTCMD or UDA または Siebel Web Applications by TTY(またはこれら両方)
- i. Siebel Gateway Connection
- j. Siebel Application Servers
- k. Siebel Application Server Configuration
- l. Siebel DB by NTCMD or UDA または Siebel DB by TTY(またはこれら両方)

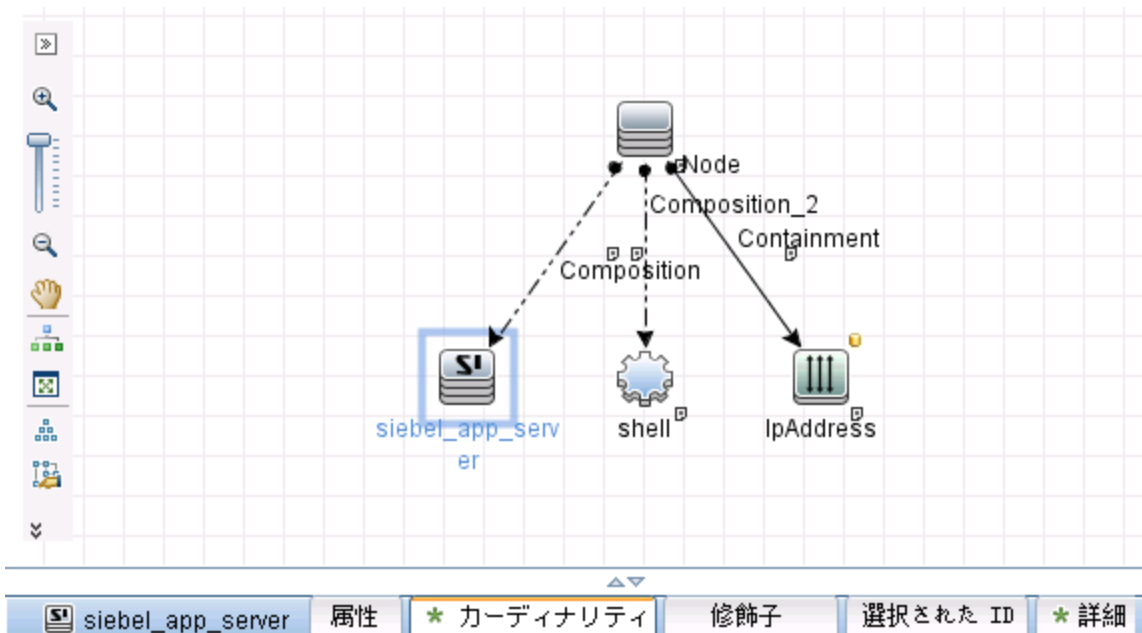
**注:** 次のエンリッチメント・アダプタは、ディスカバリ中にバックグラウンドで自動的に実行されます。  
**Siebel\_Route\_WebApp\_To\_Component。** Siebel Web Application CI と Siebel Component CI との間のルートを構築します。  
**Siebel\_Web\_To\_Middle\_Tier。** Siebel エンタープライズが負荷分散のために Resonate サーバを使用するときに、Web 層と中間層との間のルートを構築します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Siebel Application Server Configuration ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ



Composition (Node, siebel\_app\_server) : 1..\*

## アダプタ

このジョブでは、**SIEBEL\_DIS\_APP\_SERVER\_CONFIG** アダプタを使用します。

## 使用スクリプト

- file\_ver\_lib.py
- siebel\_discover\_appserver\_config.py

## 検出された CIT

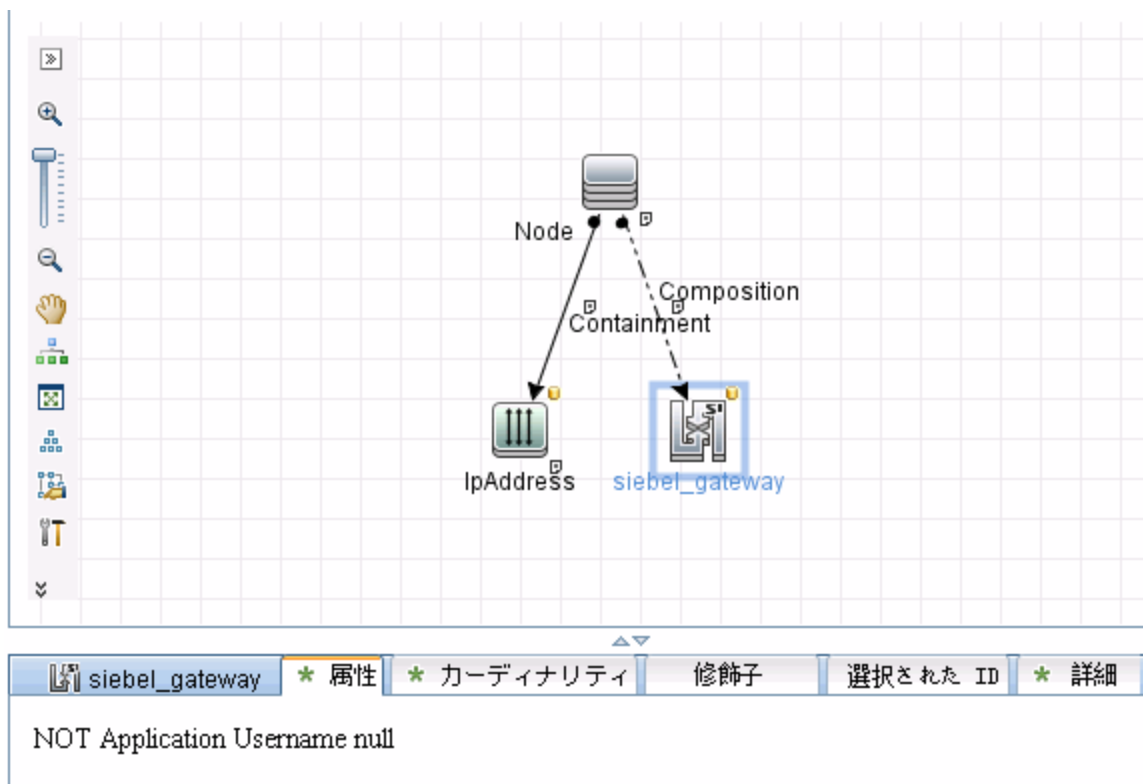
- Composition
- ConfigurationDocument
- Siebel Application Server

注: トポロジを表示するには、「[Siebel トポロジ・ビュー](#)」(662ページ)を参照してください。

## Siebel Application Servers ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



### アダプタ

このジョブでは、**SIEBEL\_DIS\_APP\_SERVERS** アダプタを使用します。

### 使用スクリプト

- siebel\_common.py
- siebel\_discover\_enterprise.py



## 検出された CIT

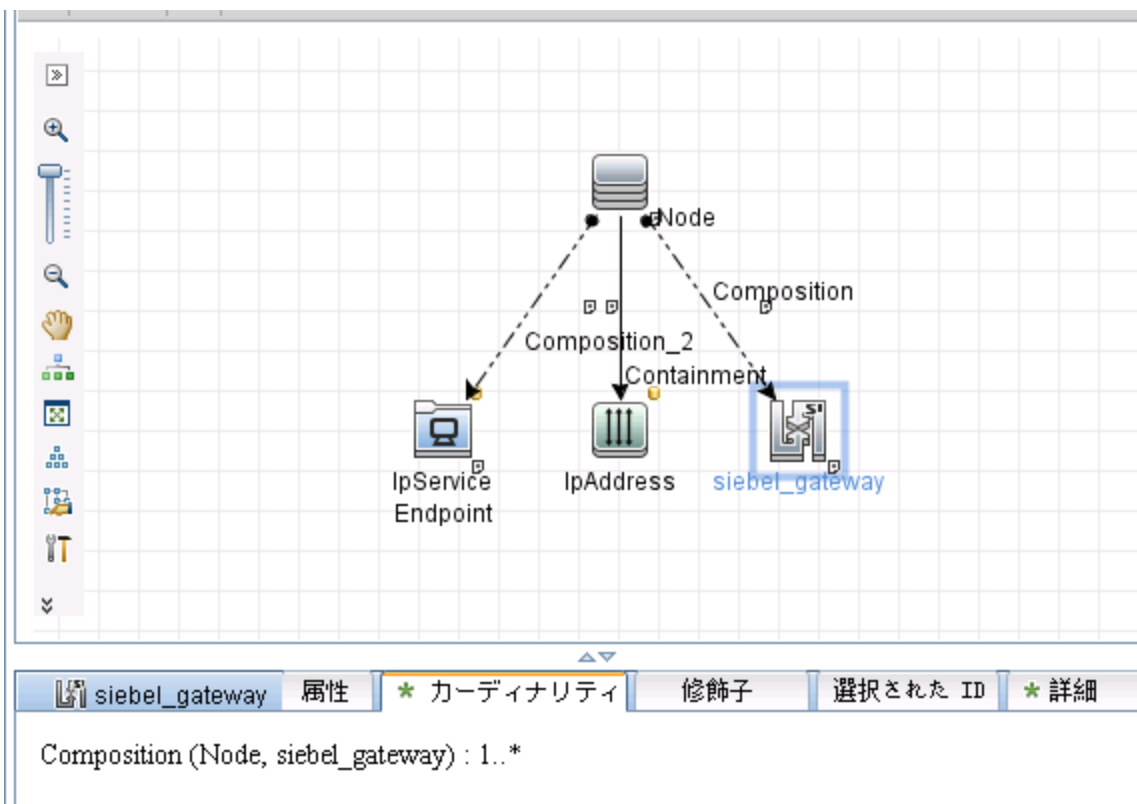
- Composition
- ConfigurationDocument
- Containment
- Dependency
- IpAddress
- Membership
- Node
- Siebel Application
- Siebel Application Server
- Siebel Component
- Siebel Component Group

注: トポロジを表示するには, 「[Siebel トポロジ・ビュー](#)」([662ページ](#))を参照してください。

## Siebel Gateway Connection ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ



## アダプタ

このジョブでは、**SIEBEL\_DIS\_GATEWAY\_CONNECTION\_(GTWY)** アダプタを使用します。

## 使用スクリプト

- siebel\_common.py
- siebel\_discover\_gateway.py

## 検出された CIT

検出された CI の詳細については、**[詳細]** タブの統計テーブルを参照してください。

- Composition
- Membership

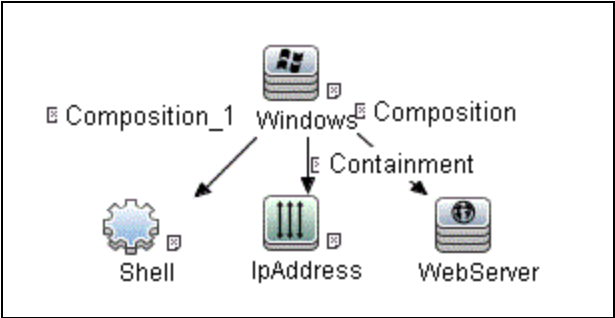
- Siebel Enterprise
- Siebel Gateway

注: トポロジを表示するには, 「[Siebel トポロジ・ビュー](#)」(662ページ)を参照してください。

## Siebel Web Applications by NTCMD or UDA ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



ノード名	条件
Windows	なし
Shell	NOT Reference to the credentials dictionary entry null AND(CI Type 等しい ntcmd OR CI Type 等しい uda)
IpAddress	NOT IP Probe Name null
WebServer	なし

### アダプタ

このジョブでは, **SIEBEL\_DIS\_WEBAPPS\_NT** アダプタを使用します。

### 使用スクリプト

- NTCMD\_HR\_REG\_Software\_Lib.py

- siebel\_discover\_wse.py

## 検出された CIT

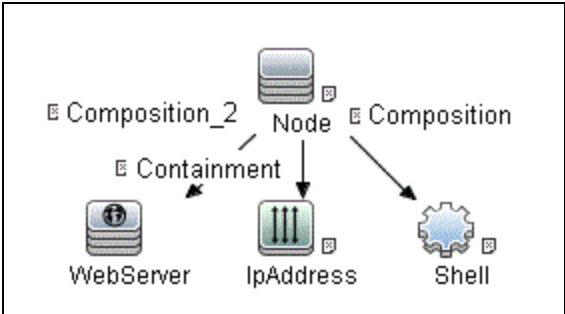
- Composition
- ConfigurationDocument
- Containment
- Dependency
- IpAddress
- Node
- Route
- Siebel Enterprise
- Siebel Gateway
- Siebel Web Application
- Siebel Web Server Extension
- WebServer

注: トポロジを表示するには, 「[Siebel Web トポロジ・ビュー](#)」(663ページ)を参照してください。

## Siebel Web Applications by TTY ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ・クエリ



ノード名	条件
Node	NOT CI Type 等しい nt
WebServer	なし
IpAddress	NOT IP Probe Name null
Shell	NOT Reference to the credentials dictionary entry null

アダプタ

このジョブでは、**SIEBEL\_DIS\_WEBAPPS\_UNIX** アダプタを使用します。

使用スクリプト

- siebel\_discover\_wse.py
- NTCMD\_HR\_REG\_Software\_Lib.py

検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Dependency
- IpAddress

- Node
- Route
- Siebel Enterprise
- Siebel Gateway
- Siebel Web Application
- Siebel Web Server Extension
- WebServer

注: トポロジを表示するには, [「Siebel Web トポロジ・ビュー」\(663ページ\)](#)を参照してください。

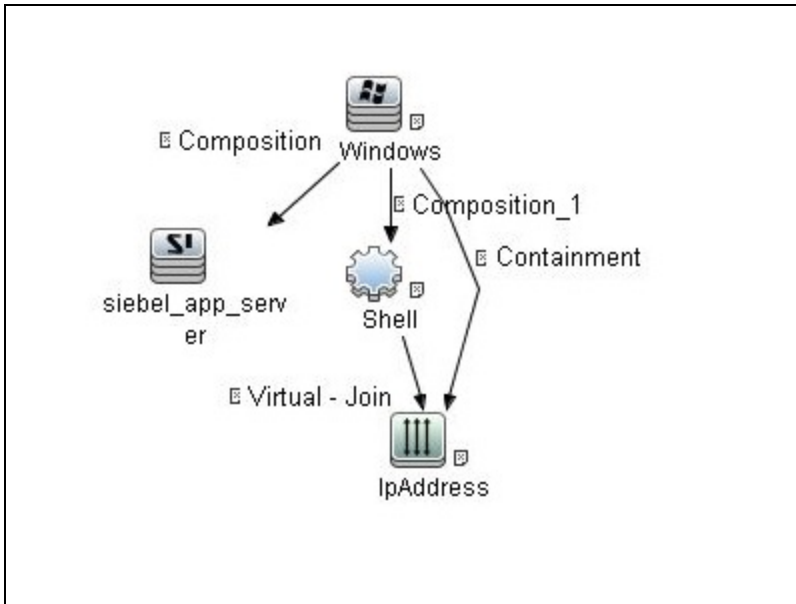
## パラメータ

- **eappsCfgPath**。Siebel Webserver Extension 構成ファイル(eapps.cfg)に対するパスを示します。

## Siebel DB by NTCMD or UDA ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ・クエリ



アダプタ

このジョブでは、**SIEBEL\_DIS\_DB\_NT** アダプタを使用します。

## 使用スクリプト

- db\_platform.py
- db.py
- db\_builder.py
- siebel\_discover\_odbc.py

## 検出された CIT

- Composition
- Containment
- Database
- Dependency
- IPAddress
- Node

注: トポロジを表示するには, 「[Siebel トポロジ・ビュー](#)」(662ページ)を参照してください。

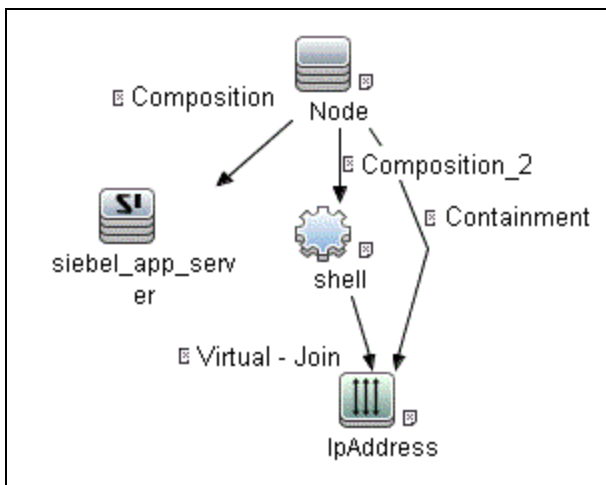
## パラメータ

- **oracle\_name**. 異なるマシンに対する複数の ORACLE\_NAME パスを含めることができます(カンマ区切り)。空の場合, スクリプト内にハードコードされたレジストリが使用されます。

## Siebel DB by TTY ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ





ノード名	条件
Node	NOT CI Type 等しい nt
siebel_app_server	なし
IpAddress	NOT IP Probe Name null
Shell	NOT Reference to the credentials dictionary entry null

## アダプタ

このジョブでは、**SIEBEL\_DIS\_DB\_UNIX** アダプタを使用します。

## 使用スクリプト

- db\_platform.py
- db.py
- db\_builder.py
- siebel\_discover\_odbc.py

## 検出された CIT

- Composition
- Containment
- Database
- Dependency
- IpAddress
- Node

注: トポロジを表示するには、[「Siebel トポロジ・ビュー」\(662 ページ\)](#)を参照してください。

## トラブルシューティングおよび制限事項

Siebel DB by TTY ジョブは、UNIX マシン上で実行されている仮想 Siebel アプリケーション・サーバ(実際の Siebel アプリケーション・サーバと異なる名前と構成)を検出できません。

## 第7部: Hosts and Resources

## 第44章: アプリケーション署名

### 本章の内容

概要 .....	681
メソッドの機能 .....	681
署名リポジトリ .....	682
入力データ .....	682
トポロジ .....	683
アプリケーション署名のメカニズム .....	685
ジョブおよびアダプタ .....	686
ジョブおよび標準設定の動作 .....	686
Host Resources ジョブ・フロー .....	687
アプリケーション署名のフロー .....	688
アプリケーション署名の構成ファイル .....	692
構成ファイルの構造 .....	692
要素および属性 .....	693
プラグイン .....	698

## 概要

アプリケーション署名とは、以前に検出されたホスト・リソース・データに基づいて、ターゲット・ホスト上で稼働するアプリケーションを特定する方法です。具体的なデータは次のとおりです。

- プロセス
- 開かれているポート(オプション)

ただし、場合によって、アプリケーション署名は次を示すことがあります。

- 一般的な識別方法
- メソッドを実装するディスカバリ・モジュール
- 署名が記述された構成ファイル

## メソッドの機能

アプリケーション署名メソッドの機能は次のとおりです。

- プロセス名の完全一致または部分一致に基づいてアプリケーションを識別します。
- 特定の開かれたポートが存在するかどうかに基づいてアプリケーションを識別します。
- プロセスのコマンド・ラインに特定の部分文字列が存在するかどうかに基づいてアプリケーションを識別します。
- 同じホスト上でいくつかのアプリケーション・インスタンスが実行されている場合に、インスタンスを区別します(制限あり)。
- 利用可能な識別情報が十分にある場合、RunningSoftware CIT の特定のサブクラスをレポートします。
- RunningSoftware CI に追加の属性を付加します。データは、解析ルールを使用することによってプロセスから取得します。
- プラグインをサポートします。アプリケーションが検出されると、追加のコマンドを実行できるようになり、データを追加してアプリケーションのトポロジを強化できるようになります。

詳細については、「[アプリケーション署名のメカニズム](#)」(685ページ)を参照してください。

## 署名リポジトリ

アプリケーション署名では、**applicationsSignature.xml** という構成ファイル内に保存された署名のリポジトリが使用されます。このXML ファイルには、多くの Application-Component 要素が含まれており、それぞれの要素に、固有のアプリケーションまたはアプリケーションのフレーバーの署名が記述されています。たとえば、Microsoft IIS Web サーバの署名は次のようになります。

### Microsoft IIS Web サーバの署名

```
<Application-Component name="Microsoft IIS WebServer" ci_type="iis"
category="Web Server"
  vendor="microsoft_corp" installedSoftwareName="\s*IIS\s+.*" supported_
versions="5.1, 6.0, 7.0">
  <process name="inetinfo.exe" ports="all,None" cmdline=""
description="Microsoft Internet Information Service process."/>
  <process name="w3wp.exe" ports="None" cmdline=""
description="Microsoft Internet Information Service worker process."/>
</Application-Component>
```

詳細については、「[アプリケーション署名の構成ファイル](#)」(692ページ)を参照してください。

## 入力データ

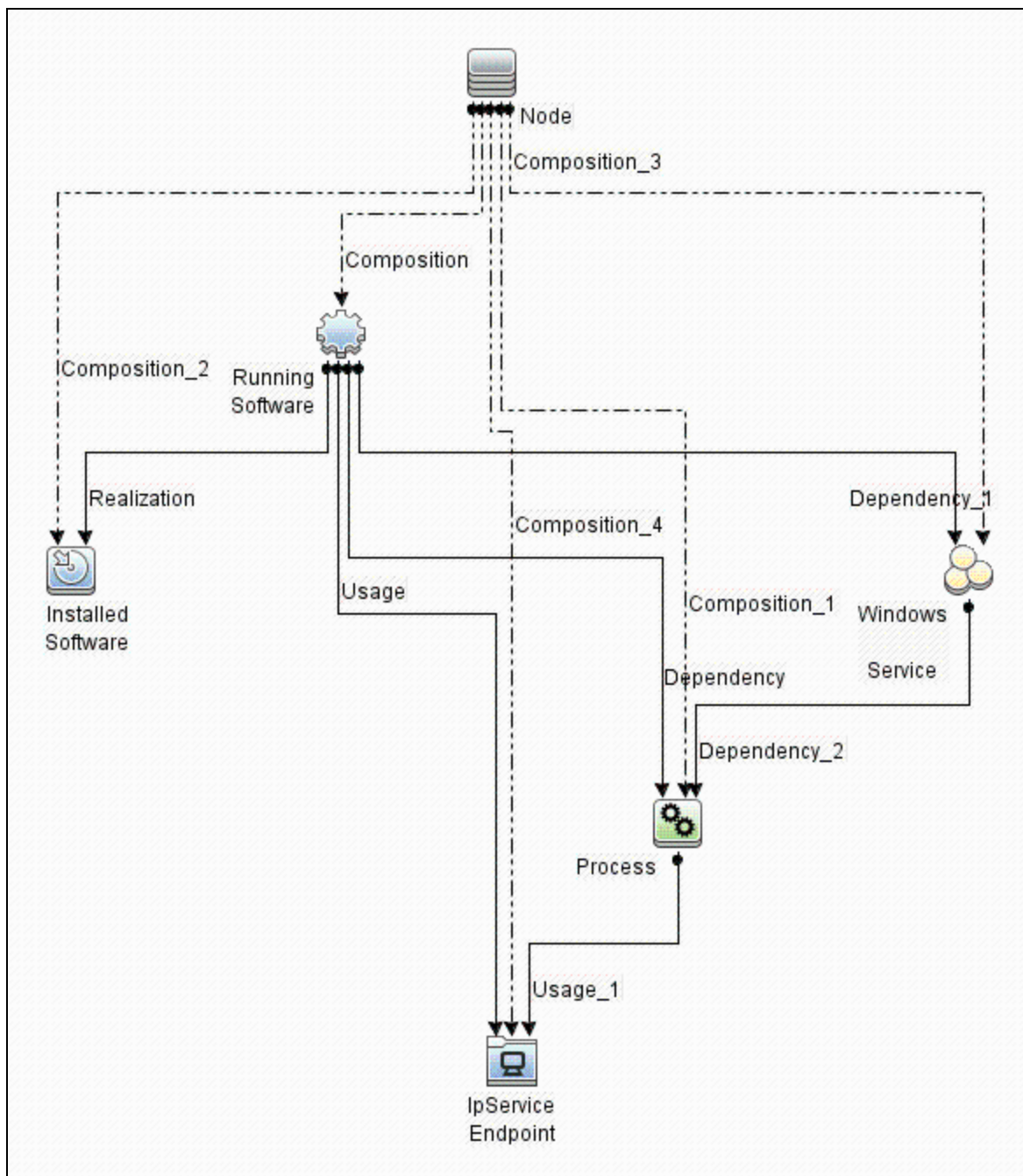
アプリケーション署名では次の入力データが使用されます。

- ホスト上で実行されている次のようなプロセスの情報
  - 名前
  - PID
  - コマンド・ライン、実行可能ファイルのパス、引数
  - 所有者
- プロセスのポート(オプション)
  - 標準の開かれたポートおよびリスナ・ポート
  - ポートを開くために使用されるインタフェース
- インストール済みソフトウェア(オプション)
- サービス(オプション)

プロセスおよびポートの情報は、アプリケーションの識別に使用されます。サービスおよびインストール済みソフトウェアに関する情報は識別には使用されませんが、対応するトポロジとの関係を構築するために使用されます。

## トポロジ

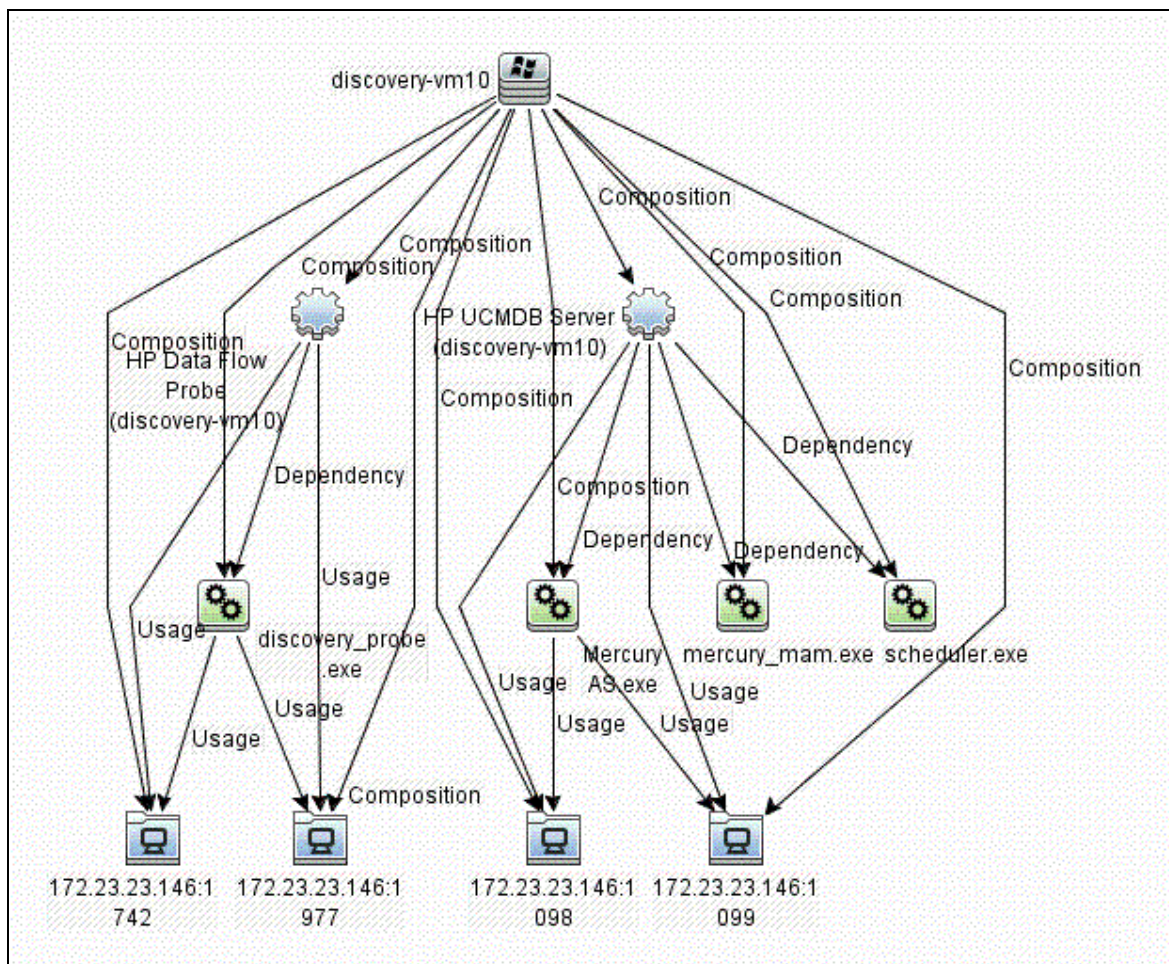
Application Signatures トポロジの主な要素は、実行中のアプリケーションを表す RunningSoftware CIT です。次の図に、アプリケーション署名によってレポートされるトポロジのタイプを示します。





## トポロジの例

次の図は、検出されるトポロジの例です。このトポロジには、同じホスト上で実行されている HP UCMDB サーバと Data Flow Probe が含まれています。



## アプリケーション署名のメカニズム

本部分の内容

- 「ジョブおよびアダプタ」(686ページ)
- 「ジョブおよび標準設定の動作」(686ページ)
- 「Host Resources ジョブ・フロー」(687ページ)
- 「アプリケーション署名のフロー」(688ページ)

## ジョブおよびアダプタ

アプリケーション署名は、ホスト・リソース・ジョブの一部として実行されます。ホスト・リソース・ディスカバリを実装するアダプタはいくつかあり、それぞれに次の2つのジョブがあります。

- **アダプタ : Host Resources by TTY(TTY\_HR\_All.xml)**
  - ジョブ :Host Applications by Shell
  - ジョブ :Host Resources by Shell
- **アダプタ : Host Resources by WMI(WMI\_HR\_ALL.xml)**
  - ジョブ :Host Applications by WMI
  - ジョブ :Host Resources by WMI
- **アダプタ : Host Resources by SNMP(SNMP\_HR\_ALL.xml)**
  - ジョブ :Host Applications by SNMP
  - ジョブ :Host Resources by SNMP
- **アダプタ : Host Resources by PowerShell(PowerShell\_HR\_All.xml)**
  - ジョブ :Host Applications by PowerShell
  - ジョブ :Host Resources by PowerShell

## ジョブおよび標準設定の動作

Host Resources by X ジョブの動作は、標準設定では次のようになります。

- CPU の検出 およびレポート
- ディスクの検出 およびレポート
- メモリの検出 およびレポート
- ユーザの検出 およびレポート
- ネットワーク共有の検出 およびレポート
- プロセスの検出 およびプローブ・データベースへのデータの保存

- プロセス・ポートの検出 およびプローブ・データベースへのデータの保存
- プロセス間トポロジの検出 およびレポート
- アプリケーション署名 ディスカバリは実行しない

**Host Applications by X** ジョブの動作は、標準設定では次のようになります。

- プロセスの検出 およびプローブ・データベースへのデータの保存
- プロセス・ポートの検出 およびプローブ・データベースへのデータの保存
- インストール済みソフトウェアの検出
- プロセス間トポロジの検出 およびレポート
- アプリケーション署名 ディスカバリの実行

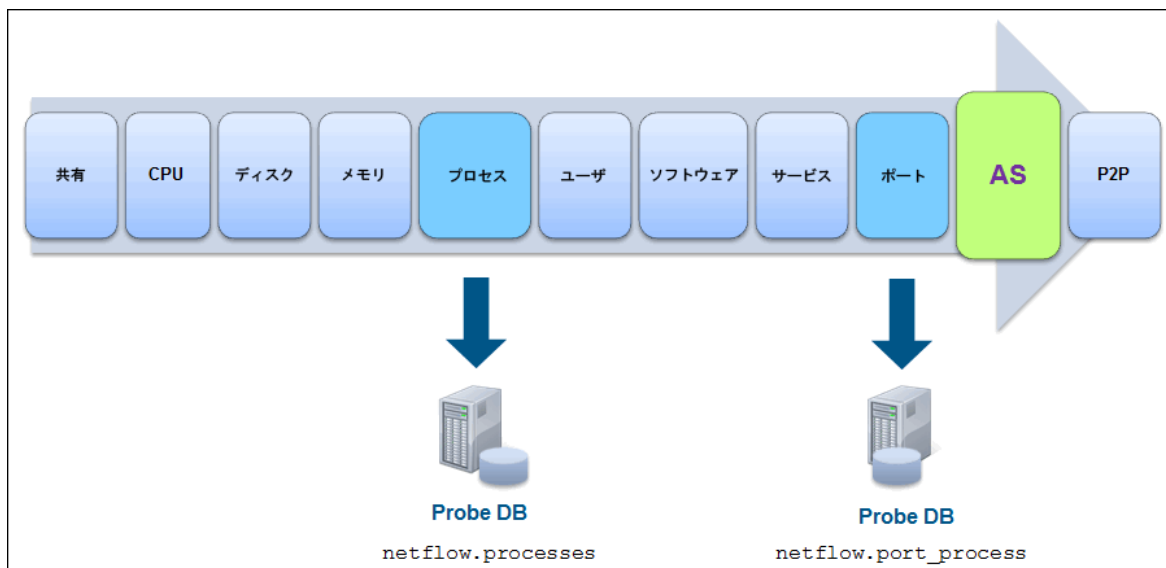
**注:** ここに示されたジョブに、標準設定でサービスを検出するものはありません。

ジョブは、ホスト・リソース・トポロジのどの部分をレポートする必要があるかを制御するパラメータを提供します。これらの同じアダプタに基づいて新しいカスタム・ジョブを作成し、提供されるパラメータが目的の値になるように変更できます。[「ホストのリソースおよびアプリケーションのディスカバリ」\(711ページ\)](#)を参照してください。

アプリケーション署名 ディスカバリは、管理ゾーン・ディスカバリの一部としても実行できます。アプリケーション署名を使用したアプリケーション・ディスカバリを呼び出すには、基本ソフトウェア構成 ディスカバリを構成する必要があります。「Universal Discovery Activities」を『HP UCMDB Discovery and Integrations Content Guide - Discovery Activities』ドキュメントから参照してください。

## Host Resources ジョブ・フロー

次の図に、Host Resources ジョブの一般的なフローを示します。ただし、フローの一部はパラメータを介して有効または無効にされる場合があります。



アプリケーション署名に関する Host Resources ジョブのフローは次のとおりです。

1. プロセス・ディスカバリは、Host Resources ジョブの一部として実行されます。検出されたプロセスは、プローブのデータベース(netflow.processes テーブル内)に保存されます。ジョブの **discoverProcesses** パラメータが true に設定されている場合、これらのプロセスは、この時点で結果のベクトルに追加されます。それ以外の場合、プロセスは即時にはレポートされません。
2. TCP ディスカバリが実行され、これにより、リスナ・ポートと通常のクライアント・ポートの両方の開かれたすべてのポートが検出されます。このデータは、プローブのデータベース(netflow.port\_process テーブル内)に保存されます。
3. アプリケーション署名エンジンが構成され、検出されたプロセスと開かれたポートは、入力データとしてエンジンに渡されます。サービス・オブジェクトとインストール済みソフトウェア・オブジェクトが検出されると、これらもアプリケーション署名エンジンに渡されます。
4. アプリケーション署名が実行され、アプリケーションの識別が行われ、対応するトポロジがレポートされます。
5. プロセス間ディスカバリが実行され、プロセス間のクライアント・サーバ・リンクがレポートされます。

## アプリケーション署名のフロー

アプリケーション署名は、アプリケーションと関連トポロジを特定してレポートするために次のアクティビティを実行します。

1. エンジン は applicationsSignature.xml ファイルから構成を読み取ります。
2. エンジン は検出されたプロセスを各署名と関連付けようとします。
3. エンジン は署名を検証し、一致しない署名および満たされない要件を持つ署名を破棄します。
4. エンジン は、各署名のアプリケーション・インスタンス数について決定します。
5. エンジン は、アプリケーションに対応するOSH オブジェクトを作成します。
6. 解析ルールがある場合、エンジンはそのルールを使用して式を評価することによって、アプリケーション属性の更新を実行します。属性の更新は、任意の場合と必須の場合があります。必須の属性の式に使用された解析ルールに一致しなかった場合、そのアプリケーション・インスタンスはスキップされます。属性の更新が任意の場合には、影響ありません。
7. エンジン は、特定のアプリケーションに適切なすべてのプラグインを検出し、それらを実行します。プラグインは、同じクライアントで追加のコマンドを実行する、属性を更新する、トポロジを強化する、アプリケーションをスキップするかどうかの判断を行うなどの処理を行うことができます。
8. エンジン は、すべてのアプリケーション・オブジェクトを含む結果のベクトルを形成します。また、この段階で、エンジンはアプリケーションとプロセスをインストール済みソフトウェアとサービスにリンクします。

## アプリケーション・インスタンスの識別

多くのアプリケーションは、インスタンス化をサポートしています。これは、複数の独立したサブアプリケーションが、同じホスト上で実行されることです。このようなインスタンスは、個別に識別するための名前やIDなどを持つのが一般的です。アプリケーション署名でサポートされるインスタンス化は、各インスタンスにつき、対応する1つの指定の実行中プロセスがあるという考えに基づいています。つまり、このようなプロセスの数と、アプリケーション・インスタンスの数との間には、1対1の関係があります。アプリケーション署名構成ファイルでこれらのプロセスを特定してマークすることによって、インスタンス化されたアプリケーションをレポートできるようになります。これを実装する目的で、**main-process** ブール属性が、署名の **process** 要素に導入されました。

**main-process** 属性は、アプリケーションのトポロジに次のように作用します。

- **application-component** に **main-process** 属性が設定されたプロセスがない場合、またはこの属性が **false**(標準設定)に設定されている場合、単一の **RunningSoftware** CI が作成され、すべてのプロセスがこのCIにリンクされます。
- 特定のプロセスに対して **main-process** 属性が **true** に設定されている場合、そのプロセスが、アプリケーション・インスタンスを識別するプロセスとみなされます。このようなプロセスの数によって、アプリケーション・インスタンスの数が定義され、**RunningSoftware** CI が作成される数が決まります。メイン・プロ

セスの各インスタンスは、対応する1つの **RunningSoftware** CI とのみリンクされます。その他のすべてのプロセスは共有プロセスとみなされ、作成されたすべての **RunningSoftware** CI とリンクされます。

#### インスタンス化されたアプリケーションの署名の例

```
<Application-Component name="A" >
  <process name="P1" main-process="true" ports="None" cmdline=""
    required="true" />
  <process name="P2" ports="None" cmdline="" />
</Application-Component>
```

- アプリケーション A の署名は、**main-process** 属性が **true** に設定されたプロセス **P1** およびプロセス **P2** から構成されます。
- 検出されたデータにはプロセス **P1**(P11, P12)の2つのインスタンスとプロセス **P2**(P21, P22)の2つのインスタンスが含まれます。
- 結果のトポロジは次のようになります。
- メインとしてマークされた2つのプロセスがあるため、**A1**と**A2**の2つの **RunningSoftware** CI があります。
- **A1** はプロセス **P11**, **P21**, **P22** とリンクされます。
- **A2** はプロセス **P12**, **P21**, **P22** とリンクされます。

注: プロセスを指定せずに、インスタンスごとに1対1の関係でアプリケーションをインスタンス化する方法はサポートされません。

## 属性の更新および式の評価

アプリケーション署名では、属性の更新がサポートされます。この場合、レポートされた **RunningSoftware** CI の特定の属性に、指定した値が入力されることがアプリケーション署名で宣言されます。次の値を使用できます。

- 定数
- 文字列および評価された解析ルールへの参照を含む式

解析ルールとは署名で宣言されるパターンであり、コマンド・ラインやプロセスのパスなどのアプリケーションデータに対して照合されます。解析ルールのパターンは、一致するか一致しないかのいずれかです。解析ルールのパターンは、正規表現のパターンです。解析ルールが一致すると、属性の更新の式に、対応す

る一致文字列またはグループを使用できます。解析ルールが、プロセスのプロパティと照合されるように宣言された場合、解析ルールの照合は、すべてのプロセスに対し、最初の一一致が見つかるまで1つずつ実行されます。

次に **Active Directory Application Mode** アプリケーションの署名の例を挙げます。ここでは、解析ルールによる属性の更新が使用されています。

#### 属性の更新がある Active Directory Application Mode

```
<Application-Component name="Active Directory Application Mode" ci_type="adam"
category="Enterprise App" vendor="microsoft_corp">
  <parse-rule id="instance_name" name="cmdline">.+\\s+-sn:([\\w-]
  +).*</parse-rule>
  <process name="dsamain.exe" main-process="true" ports="ldap,636,
  None" required="true" cmdline="" description="Main AD
  Application Mode process" />
  <attribute name="name" value="{instance_name(1)}" type="string" />
</Application-Component>
```

この署名のフローは次のとおりです。

- ターゲット・ホストで見つかった **dsamain.exe** プロセスによって、この署名の照合が可能になり、**adam** タイプの **RunningSoftware** が生成されます。
- エンジンが属性の更新を開始すると、次のように、すべての解析ルールがこのアプリケーションに対して照合されます。
  - 解析ルール **instance\_name** のパターンが、見つかったプロセスのコマンド・ラインと照合されます。
  - このパターンはキャプチャするグループを使用します。一致したパターンによって、一致した文字列全体と、キャプチャしたグループ 1 の両方が提示されます。このグループには、ADAM インスタンスの値が含まれます。
- 属性の更新の式 **{instance\_name(1)}** が評価され、コマンド・ラインから抽出された ADAM のインスタンス名を含む文字列に変換されます。
- この値は、アプリケーション CI の **name** 属性にレポートされます。

#### ポートの照合およびレポート

署名の **process** 要素に、この **process** 要素が一致するかどうかに影響するポートの値と、レポートするポートを指定できます。ポートは **ports** 属性を使用して、カンマ区切りの値リストの形式で指定されます。

次のルールが適用されます。



- **ports** 属性の値に 'None' 文字列がある場合、このプロセスに対して検出されたポートに関係なく、**process** 要素は常に一致します。
- それ以外の場合、**process** 要素が一致するためには、検出されたプロセスで、属性で指定されたすべての対応するポートが開かれている必要があります。
- **ports** 属性で指定されたポートのみがレポートされます。
- アプリケーション署名では、アプリケーションがリッスンするすべてのポートをレポートすることもできます。これを実現するには、次の2つの方法があります。
  - 特定のアプリケーションと特定のプロセスのみに対し、**ports** 属性の値に 'all' キーワードを含めます。
  - **globalSettings.xml** 構成ファイルで **discoverAllListenPorts** ブール・パラメータを使用して、グローバルに実行します。この例は次のとおりです。

#### globalSettings.xml

```
<property name="discoverAllListenPorts">false</property>
```

## アプリケーション署名の構成ファイル

**applicationsSignature.xml** 構成ファイルは、アプリケーション署名を検出するための署名のリポジトリです。各署名には、アプリケーションと関連付けられた一連のプロセスが含まれます。署名の例は次のとおりです。

#### PostgreSQL データベース・アプリケーション署名

```
<Application-Component name="PostgreSQL" category="Database" vendor="PostgreSQL"
supported_versions="7.x, 8.x" installedSoftwareName="PostgreSQL.*">
  <process name="pg_ctl.exe" ports="postgresql,None" cmdline=""
description="PostgreSQL Server."/>
  <process name="postmaster" ports="postgresql,None" cmdline=""
description="PostgreSQL Server."/>
</Application-Component>
```

## 構成ファイルの構造

**Applications** 要素は、構成ファイルのルートです。ここには、**Application-Component** 要素として表される署名のリストが含まれます。追加の要素 **Default-Application-Type** は、アプリケーションのレポート時に使用される CIT を指定します。



**PostgreSQL データベース・アプリケーション署名**

```

<Applications parserClassName="com.hp.ucmdb.discovery.library.
communication.downloader.cfgfiles.ApplicationSignatureConfigFile">

    <Default-Application-Type>running_software</Default-Application-Type>

    <Application-Component name= ...>
        <process name= .../>
    </Application-Component>

    ...

    <Application-Component name= ...>
        <process name= .../>
    </Application-Component>

</Applications>

```

**要素および属性****Application-Component 要素**

**Application-Component** 要素は、特定のアプリケーションまたはアプリケーション・タイプの署名を定義します。この要素には、次の表に示す属性を含めることができます。

属性	任意指定	説明
<b>name</b>	いいえ	アプリケーションの名前。この値は、アプリケーション CI の <b>DiscoveredProductName</b> 属性にレポートされます。
<b>app_id</b>	はい	署名の ID。この属性が設定されないと、 <b>name</b> 属性の値が ID に使用されます。  <b>注:</b> すべての ID の値は、構成ファイル内で一意である必要があります。
<b>category</b>	はい	アプリケーションのカテゴリ。この値は、アプリケーションの <b>application_category</b> 属性に渡されます。  例 : Database, Cluster
<b>vendor</b>	はい	アプリケーション・ベンダ。この値は、アプリケーション

属性	任意指定	説明
		<p>ンの <b>vendor</b> 属性に渡されます。</p> <p>例: hewlett_packard_co</p>
<b>supported_versions</b>	はい	署名が適用されるバージョンのリスト。この情報は、レポートされるトポロジには影響を与えず、情報提供のみを目的としています。
<b>installedSoftwareName</b>	はい	インストール済みソフトウェアとの関係をレポートするために使用される正規表現パターン。インストール済みソフトウェアの名前がパターンに一致すると、対応する関係がレポートされます。
<b>ci_type</b>	はい	<p>アプリケーションのレポート時に使用される必要がある CIT 名。指定されないと、次の標準設定の CIT が使用されます。RunningSoftware。</p> <p>例: &lt;Application-Component name="Oracle DB" ci_type="oracle"&gt; はアプリケーションが CIT 「oracle」としてレポートされる必要があることを示します。</p>
<b>discover</b>	はい	<p>署名を有効化 (<b>true</b>) または無効化 (<b>false</b>) するブール属性。</p> <p>標準設定: True</p>

**Application-Component** には、1 つ以上の **process** 要素が含まれる必要があります。この要素は、このアプリケーションがレポート対象とされるために検出する必要があるプロセスを定義します。process 要素には、次の表に示す属性を含めることができます。

属性	任意指定	説明
<b>name</b>	いいえ	プロセスの名前。
<b>ports</b>	いいえ	<p>プロセスが一致するためにリッスンする必要がある、または開かれている必要があるポートのカンマ区切りリスト。各ポートは、次の値のいずれかを持つ場合があります。</p> <ul style="list-style-type: none"> <li>None: プロセスがリッスンするポートがない場合でも、プロセスは一致します。</li> <li>数値、または数値に解決される名前付きポート。プロセスが一致するためには、このポートをリッスンする必要があるか、このポートが開かれている必要があります。例: 1521, mysql。ポート名は、</li> </ul>

属性	任意指定	説明
		<p>portNumberToPortName.xml 構成ファイルと照合して解決されます。</p> <ul style="list-style-type: none"> <li>All: リッスンするすべてのポートがレポートされる必要があることを指定します。このパターンを介して一致するすべてのポートは、IpServiceEndpoint CI としてレポートされます。</li> </ul>
<b>cmdline</b>	はい	プロセスが一致するために、プロセスのコマンド・ラインに存在する必要がある部分文字列。
<b>description</b>	はい	プロセスの説明。この属性の値は、このプロセス要素が一致するプロセス CI の <b>description</b> 属性に渡されます。
<b>startswith</b>	はい	プロセス名の部分一致を有効にするブール属性。この属性が <b>true</b> に設定されていると、 <b>name</b> 属性に指定された文字列で始まる名前のすべてのプロセスが一致します。
<b>required</b>	はい	このブール属性が <b>true</b> に設定されていると、アプリケーションがレポート対象となるためには、そのプロセスが必須であることを示します。このプロセス要素がどのプロセスとも一致しない場合、アプリケーションはレポートされません。
<b>main-process</b>	はい	<b>true</b> に設定した場合に、このプロセスがアプリケーションのメイン・プロセスであることを示すブール属性。ホスト上で検出される各メイン・プロセスについて、個別のアプリケーション・トポロジがレポートされます。
<b>ignore-case</b>	はい	<p>プロセス名の照合で大文字と小文字を区別するか(<b>true</b>)、区別しないか(<b>false</b>)を示すブール属性。</p> <p><b>標準設定</b> : False</p>

## parse-rule 要素

parse-rule 要素では、何らかの値を抽出する目的で、検出されたプロセスの特定のプロパティに適用される解析ルールを定義します。解析ルールは基本的に正規表現であり、プロセスのプロパティと照合されます。この正規表現に、キャプチャするグループが含まれている場合、これらのグループの値を抽出し、それらを属性更新の式に使用できる可能性があります。この正規表現は、この要素のテキスト部分に記述します。parse-rule 要素には、次の表に示す属性を含めることができます。

属性	任意指定	説明
<b>id</b>	いいえ	解析ルール ID。この解析ルールを特定する一意の文字列です。
<b>name</b>	いいえ	このルールと照合するプロセス・プロパティの名前。サポートされるプロパティは次のとおりです。 <ul style="list-style-type: none"> <li>• <b>cmdline</b></li> <li>• <b>ip</b></li> <li>• <b>name</b></li> <li>• <b>owner</b></li> <li>• <b>path</b></li> <li>• <b>port</b></li> </ul>
<b>method</b>	はい	パターンの照合時に使用するメソッド。次のいずれかを使用します。 <ul style="list-style-type: none"> <li>• <b>match</b>。プロパティ全体がパターンと照合されます。正規表現の 'match' 演算と同様に動作します。</li> <li>• <b>search</b>。パターンはこの文字列内で検索されます。正規表現の 'search' 演算と同様に動作します。</li> </ul> <b>標準設定</b> : match

## Attribute 要素

**Attribute** 要素が **Application-Component** 要素の下にある場合、レポートされるアプリケーションCIの特定の属性に、指定された値が設定されることを意味します。この値として定数、文字列、解析ルール参照を指定した式を使用できます。attribute 要素には、次の表に示す属性を含めることができます。

属性	任意指定	説明
<b>name</b>	いいえ	設定する属性の名前。
<b>value</b>	いいえ	属性の値を定義する式。この式には、変更なしで使用される通常のテキストと、解析ルールへの参照を含めることができます。解析ルール参照は次のように定義されます。

属性	任意指定	説明
		<ul style="list-style-type: none"> <li>• <math>\{rule\_id\}</math> - rule_id という ID を持つ解析 ルールの全体一致が、式の評価時に結果の文字列に挿入されます。</li> <li>• <math>\{rule\_id(group\_number)\}</math> - rule_id という ID の解析 ルールが、キャプチャするグループ番号の値と一致した場合、group_number が結果の文字列に挿入されます。</li> </ul>
<b>required</b>	はい	<p>属性が必須か(<b>true</b>)必須でないか(<b>false</b>)を指定するブール属性。属性が必須の場合、属性の式の評価に失敗すると、アプリケーションはスキップされます。属性が必須でなく、式の評価に失敗した場合、アプリケーションはこの属性が設定されずにレポートされます。</p> <p><b>標準設定 : True</b></p>

## clustered-application 要素

Application-Component 要素には、**clustered-application** という追加の要素が含まれる場合があります。この要素は、アプリケーションをクラスタ化されたアプリケーションとしてマークします。アプリケーションをクラスタ化としてマークすると、この特定のアプリケーションの親ノード CI が、アプリケーション ID ごとに弱いノードとしてレポートされます。クラスタ化としてマークされないアプリケーションは、CMDB ID から復元された親ノード CI を使用します(これは標準設定の動作です)。

### clustered-application 要素の使用例

```
<Application-Component name="MSSQL DB" category="Database" ci_type="sqlserver"
vendor="microsoft_corp">

    <clustered-application/>

</Application-Component>
```

## application-ip-source 要素

process 要素には、**application-ip-source** という追加の要素が含まれる場合があります。この要素は、プロセスをアプリケーション IP のソースとしてマークし、アプリケーション IP が特定のアプリケーションに対して選択される方法に影響を与えます。プロセスのいずれかが、アプリケーション IP のソースとしてマークされると、このプロセスがリスンする IP アドレスのみが、アプリケーション IP の対象として考慮されます。マークされたプロセスがない場合、すべてのプロセスのすべての IP がアプリケーション IP の対象とみなされます(これは標準設定の動作です)。

**application-ip-source 要素の使用例**

```
<Application-Component name="MSSQL DB" category="Database" ci_type="sqlserver"
vendor="microsoft_corp">

    <process name="sqlservr.exe" main-process="true" ports="sql,None"
cmdline="" required="true" description="Provides storage, processing and
controlled access of data and rapid transaction processing.">

        <application-ip-source />

    </process>

</Application-Component>
```

**service-endpoint-name 要素**

process 要素には、追加の要素「**service-endpoint-name**」が含まれる場合があります。Process 要素にこの追加の要素がある場合、このプロセスのすべての IpServiceEndpoint CI は、指定した名前ですポートされます。この要素が使用されていない場合、IpServiceEndpoint CI の名前は、portNumberToPortName.xml ファイルを使用して解決されます(これは標準設定の動作です)。

**service-endpoint-name 要素の使用例**

```
<Application-Component app_id="Weblogic on Windows" name="WebLogic AS"
category="J2EE Server" ci_type="weblogicas" vendor="bea_systems_ltd">

    <process name="java.exe" main-process="true"
ports="weblogic,weblogicSSL,None" cmdline="-Dweblogic.Name=" >

        <service-endpoint-name>weblogic</service-endpoint-name>

    </process>

</Application-Component>
```

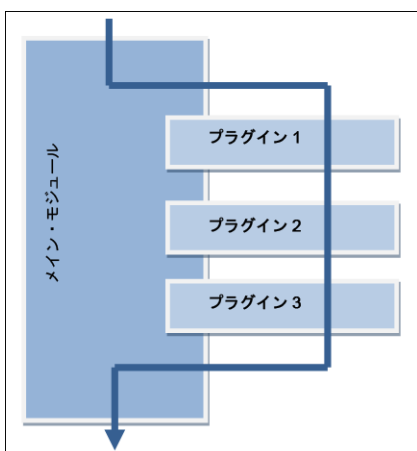
## プラグイン

アプリケーション署名 プラグインは、機能を動的に追加または削除する手段です。通常、プラグインはそれぞれ独立して動作し、単一のタスクを実行します。

Universal Discovery では、実行フローは常に次のフローと類似したものになります。

- ディスカバリ・ジョブの開始
- ジョブはいくつかのアクティビティを実行(つまりコマンドを実行)
- ジョブは結果を Object State Holder のベクトルとして形成
- ジョブは結果を返し、ジョブの実行は終了

プラグインをこのジョブに追加するには、ジョブはいずれかの時点でプラグインに制御を渡し、プラグインが結果に影響を及ぼせるようにする必要があります。これを次の図に示します。ここで、矢印は実行フローを示しています。



一般に、このメカニズムが有効になるためには、メイン・モジュールがプラグインに制御を渡す時点で、以下について確認できている必要があります。

- システムに存在するプラグイン
- プラグイン・セット全体のうち、このモジュールで実行するプラグイン
- プラグインの順序
- プラグインのコードの場所と、プラグインのインスタンスの作成方法
- プラグインが結果をメイン・モジュールに渡す方法

## プラグイン・フレームワークの概要

AutoDiscoveryContent パッケージには、**plugins.py** という Python ライブラリが含まれています。このライブラリには、プラグイン機能で使用する次のクラスが含まれます。

## • Plugin

プラグインの基本クラスです。新しいプラグインを作成する場合は、これを拡張する必要があります。このクラスには次のメソッドが含まれます。

- `isApplicable(context)` – プラグインのフレームワークから呼び出されるメソッドであり、ここでプラグインは、要求された条件を満たしていることを確認するため、渡されたデータに対してランタイム・チェックを実行できます。たとえば、アプリケーション署名では、見つかったアプリケーションが特定の名称のプロセスに関する情報を持っており、そのプロセスに空でないコマンド・ラインがあることを確認できます。要件が満たされると、メソッドは `True(1)` を返します。このメソッドの実装は任意です。
- `process(context)` – プラグインのメイン・メソッドであり、ここに主な機能が含まれています。

## • PluginContext

このクラスは、メイン・モジュールとプラグインの間、またはプラグイン間でのデータ共有を可能にするオブジェクトを表します。メイン・モジュールは、関連するデータとサポート・オブジェクト(ディスカバリ・フレームワーク・オブジェクトや初期化されたクライアントなど)をプラグインに渡すことができます。次に、プラグインはそれらのオブジェクトをプラグインの動作のために使用でき、結果を同じコンテキストに格納できます。Plugin クラスの `isApplicable()` と `process()` のどちらのメソッドも、単一のパラメータ・コンテキストを受け入れます。

**注:** `PluginContext` 基本クラスには、作成せずにそのまま使用できるメソッドはありません。そのようなメソッドは主にプラグインが使用するデータによって定義されるものであり、モジュールごとに異なる可能性があるためです。プラグインからモジュールへのサポートを追加する場合、適切なコンテキスト・クラスを作成する必要があります。

## • PluginEngine

プラグインとのすべての連携動作をカプセル化するメイン・クラスです。この目的は、適切なプラグインを見つけ、それらのインスタンスを作成し、クライアント・コードから制御が渡されたときにそれらを実行することです。クライアント・モジュールは、プラグインのメカニズムの実装方法を考慮せずに、このオブジェクトを作成し、使用できます。このクラスには、次の1つのパブリック・メソッドがあります。

- `process(context, filter)` – プラグイン・エンジンのメイン・メソッドです。このメソッドでは、エンジンは、提供されたフィルタ・オブジェクトを使用してプラグインのチェーンを形成します(次の「`PluginFilter`」を参照)。これらの各プラグインに対し、コンテキストを渡すときに、`isApplicable()` メソッドを呼び出します。プラグインが `true` を返すと、プラグインの `process()` メソッドが同じコンテキストで呼び出されます。この方法で、コンテキストはプラグイン間で受け渡しされます。



## • PluginFilter

これは、プラグインをフィルタするロジックをカプセル化するクラスを表します。このロジックは、使用可能なすべてのプラグインのプールから必要なプラグインのみを選択するものです。PluginFilter は次の単一のメソッドを定義します。

filterPlugins(pluginDescriptors) – プラグイン記述子(後述)のリストを受け入れ、これらの記述子のうち、基準を満たす記述子のみのリストを返すメソッドです。このメソッドは PluginEngine から呼び出されます。

## • QualifyingFilter

これは **PluginFilter** の実装であり、ここでは、プラグインのいくつかがチェーンに組み込まれる必要があるかどうかの判断は、プラグイン記述子内に定義された修飾子と、フィルタ自体の修飾子に基づきます([「qualifiers」\(705ページ\)](#)を参照)。このクラスには、次の追加のメソッドがあります。**addQualifier (type, value)** はこのフィルタに特定のタイプと値で修飾子を追加します。

## プラグイン・パッケージの概要

プラグインは、プラグイン・パッケージに編成されます。各パッケージは、通常は共通のテーマを持つプラグインの集合です。たとえば、すべてのシェルによるSQLデータベース・バージョンの検出です。通常は、プラグインのコードは1つのPython モジュール(1つの.py ファイル)に含まれていますが、必要があれば複数のスクリプトを使用することもできます。

各プラグイン・パッケージに、次のファイルが含まれます。

- パッケージ構成ファイル
- プラグインのコードを含む1つまたは複数のPython スクリプト(1つのプラグインにつき Plugin クラスの1つのサブクラス)

## プラグイン・パッケージ構成ファイル

プラグイン・パッケージ構成ファイルは、プラグインについて記述したメタ情報を含むXML ファイルです。例：

### db\_versions.package.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<package parserClassName="com.hp.ucmdb.discovery.
library.communication.downloader.cfgfiles.PluginsPackageConfigFile">
  <plugins>
    <plugin id="mysql_version_by_shell">
      <name>MySQL version by shell</name>
```

```

        <description>Sets MySQL version attribute for discovered
        MySQL Server CI</description>
        <module>plugins_appsignature_dbversion_by_shell</module>
        <class>MySQLVersionShellPlugin</class>
        <qualifiers>
            <qualifier type="application">MySQL DB</qualifier>
            <qualifier type="protocol">ntadmin</qualifier>
            <qualifier type="protocol">ssh</qualifier>
            <qualifier type="protocol">uda</qualifier>
            <qualifier type="protocol">telnet</qualifier>
        </qualifiers>
        <dependencies>
            <module>file_ver_lib</module>
            <module>mysql_version_by_shell</module>
        </dependencies>
    </plugin>
    ...
</plugins>
</package>

<?xml version="1.0" encoding="UTF-8" ?>
<package parserClassName="com.hp.ucmdb.discovery.
library.communication.downloader.cfgfiles.PluginsPackageConfigFile">
    <plugins>
        <plugin id="mysql_version_by_shell">
            <name>MySql version by shell</name>
            <description>Sets MySQL version attribute for discovered
            MySQL Server CI</description>
            <module>plugins_appsignature_dbversion_by_shell</module>
            <class>MySQLVersionShellPlugin</class>
            <qualifiers>
                <qualifier type="application">MySQL DB</qualifier>
                <qualifier type="protocol">ntadmin</qualifier>
                <qualifier type="protocol">ssh</qualifier>
                <qualifier type="protocol">uda</qualifier>
                <qualifier type="protocol">telnet</qualifier>
            </qualifiers>
            <dependencies>
                <module>file_ver_lib</module>
                <module>mysql_version_by_shell</module>
            </dependencies>
        </plugin>
        ...
    </plugins>
</package>

```

この構成ファイルの形式は次のとおりです。

- **plugin** 要素の **id** 属性は、このプラグインの一意のID文字列を定義します。デプロイ済みパッケージ全体で、このスコープを一意に管理するにはid属性が必要です。
- **name** 要素には、ユーザが識別できるプラグイン名が含まれます。この名前はUIに表示されます。
- **description** 要素には、プラグインについてのユーザ向けのわかりやすい説明が含まれます。
- **module** 要素は、プラグインのコードを持つPythonモジュール(.pyファイル)の名前を定義します。
- **class** 要素は、**Plugin** クラスを拡張するクラスの名前を定義します。プラグイン・エンジンは、このクラスのインスタンスの作成を試みます。
- **qualifier** 要素は、プラグインのすべての修飾子を定義します。これらの各要素に、修飾子のタイプを示す **type** 属性があります。
- **dependencies** 要素に包含される **module** 要素は、プラグインのモジュールをロードする前にロードする必要があるモジュールを定義します。この要素はこのプラグイン・モジュールで使用されます。

## プラグイン Python モジュール

プラグイン Python モジュールは通常の Python スクリプトであり、プラグインのコードが含まれます。各プラグインは、**plugins.py** の **Plugin** クラスを拡張する必要があります。1つのPythonスクリプトの中に複数のプラグインを含めることもできますし、1つのプラグイン・パッケージに複数のスクリプトを含めることもできます。また、独自のメソッドやクラスを制限なく追加できます。

「[プラグイン・パッケージ構成ファイル](#)」の場合、対応するスクリプトは次のとおりです。

```
plugins_appsignature_dbversion_by_shell.py

#coding=utf-8
import re
import sys
import logger
import mysql_version_by_shell
from file_ver_lib import getLinuxFileVer

from plugins import Plugin

class MySQLVersionShellPlugin(Plugin):
    """
        Plugin set MySQL version by shell, depends on OS type.
    """
```

```

def __init__(self):
    Plugin.__init__(self)
    self.__client = None
    self.__process = None
    self.__isWinOs = None
    self.__cmd = None

def isApplicable(self, context):
    self.__client = context.client
    try:
        if self.__client.isWinOs():
            self.__isWinOs = 1
            self.__process = context.application.getProcess(
                'mysqld-nt.exe')
            if not self.__process:
                self.__process = context.application.getProcess(
                    'mysqld.exe')
        else:
            self.__process = context.application.getProcess(
                'mysqld')
        if self.__process:
            return 1
    except:
        logger.errorException(sys.exc_info()[1])

def process(self, context):
    applicationOsh = context.application.getOsh()
    mysql_version_by_shell.setVersion(applicationOsh, self.__
    process.executablePath, self.__client)
...

class MySQLVersionShellPlugin(Plugin):
    """
        Plugin set MySQL version by shell, depends on OS type.
    """

    def __init__(self):
        Plugin.__init__(self)
        self.__client = None
        self.__process = None
        self.__isWinOs = None
        self.__cmd = None

    def isApplicable(self, context):
        self.__client = context.client
        try:
            if self.__client.isWinOs():
                self.__isWinOs = 1
                self.__process = context.application.getProcess

```

```

        ('mysqld-nt.exe')
        if not self.__process:
            self.__process = context.application.getProcess(
                'mysqld.exe')
    else:
        self.__process = context.application.getProcess(
            'mysqld')
    if self.__process:
        return 1
except:
    logger.errorException(sys.exc_info()[1])

def process(self, context):
    applicationOsh = context.application.getOsh()
    mysql_version_by_shell.setVersion(applicationOsh, self.__
        process.executablePath, self.__client)
...

```

スクリプト名とクラス名は、構成ファイル内のデータと対応しています。

## qualifiers

修飾子は、クエリを実行して使用できるプラグインのメタ情報を指定する方法です。たとえば、あるプラグインがWindows上でのみ実行可能であると示し、シェル・プロトコルを使用していると示すことができます。修飾子はタグと似ていますが、修飾子には **qualifier type** および **qualifier value** の2つの情報が含まれます。修飾子にはどのような制約もないため、任意のテキスト値を使用できます。

修飾子メカニズムは、特定のタイミングで実行するプラグインを選択するために使用されます。

QualifyingFilter インスタンスを作成し、そのフィルタで修飾子を指定します。このフィルタは、同じタイプおよび同じ値の修飾子を指定するプラグインのみ、またはそのタイプの修飾子を持たないプラグインのみを選択します。

たとえば、[「プラグイン・パッケージ構成ファイル」\(701ページ\)](#)には、次の2つの修飾子があります。

- **application** 修飾子は、プラグインが、「MySQL DB」という名前を持つアプリケーションに対してのみ実行する必要があることを指定します。
- **protocol** 修飾子は、プラグインが動作するプロトコルが、NTCMD、SSH、UDA、Telnetであることを指定します。

修飾子情報には、プラグインのインスタンスが作成される前であっても、プラグイン構成ファイルからアクセスできます。これにより、プラグインを静的にフィルタできます。

## プラグインの作成方法

1. プラグインで扱われるデータを理解するため、**PluginContext** クラスの特定の実装に精通する必要があります。アプリケーション署名の場合、**applications.py** を参照します。ここに **ApplicationSignatureContext** クラスがあります。
2. 新しい Python モジュールを作成し、そこに **Plugin** クラスの実装を追加します。具体的には、コンテキストのインスタンスを受け入れる **isApplicable()** メソッドと **process()** メソッドのコードを記述する必要があります。プラグインの動作の結果は、コンテキストに保存される必要があります。
3. プラグインに関するメタ情報を含む新しいプラグイン・パッケージ構成ファイルを作成します。ここに Python モジュール名、クラス名などの情報を指定する必要があります。「[アプリケーション署名の構成ファイル](#)」(692ページ)を参照してください。

注: 具体的には、必要に応じて、このプラグインが実行されるかどうかの決定に使用される修飾子を、プラグインに対して指定する必要がありますので、どの修飾子が実際に使用されるかを理解しておく必要があります。たとえば、アプリケーション署名内では、'application' 修飾子と 'protocol' 修飾子のみが指定されます。

4. 新しく作成されたファイルを新規または既存のデイスカバリ・パッケージに追加します。

## プラグインの制限

修飾子とプラグインの使用において不都合な点としては、プラグインがいくつ存在しているか、および検出された特定の RunningSoftware に対してそれらのどれを実行するかを特定するのが難しいことです。

# 第45章: Host Resources and Applications by PowerShell ディスカバリ

## 本章の内容

概要 .....	708
PowerShellによるホストのリソースおよびアプリケーションの検出方法 .....	708
Host Resources by PowerShell ジョブと Host Applications by PowerShell ジョブ .....	709

## 概要

Windows PowerShell は、Microsoft 社が提供するタスク自動化フレームワークであり、.NET Framework を基本構造としてこのフレームワークに統合されたコマンドライン・シェルおよび関連スクリプト言語から構成されます。PowerShell を使用することで、COM および WMI へのフル・アクセスが可能になり、管理者は、ローカルとリモートの両方の Windows システムで管理タスクを実行できます。

## PowerShellによるホストのリソースおよびアプリケーションの検出方法

次の手順では、PowerShellによるホストのリソースおよびアプリケーションの検出方法を説明します。

### 1. 前提条件 - プロトコル資格情報の設定

このデイスカバリ・ソリューションは、PowerShell プロトコルをベースにしています。このプロトコルを使用するためには、対応する資格情報を指定する必要があります。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

検出を開始する前に、Data Flow Probe マシンに PowerShell v2.0 がインストールされていることを確認します。

### 2. デイスカバリの実行

トポロジを検出するには、次の手順を実行します。

- a. **Range IPs by ICMP** ジョブまたは **Range IPs by nmap** ジョブを実行して、Windows システムの IP アドレスを検出します。
- b. **Host Connection by PowerShell** ジョブを実行して、Windows が PowerShell エージェントやネットワーク・トポロジとどのような関係を持つかを検出します。
- c. **Host Resources by PowerShell** ジョブと **Host Applications by PowerShell** ジョブを実行して、ホスト・リソース・トポロジを検出します。

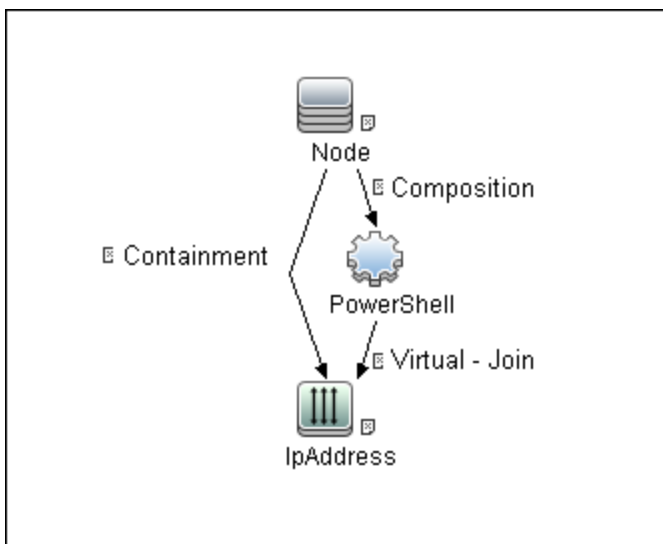
ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。



## Host Resources by PowerShell ジョブと Host Applications by PowerShell ジョブ

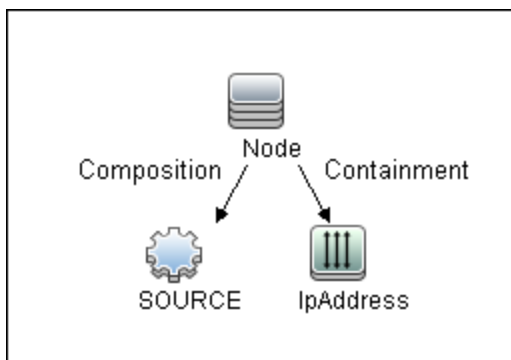
本項の内容

### トリガ・クエリ



### アダプタ

- 入力 CIT : PowerShell
- 入力 TQL クエリ:



## 検出された CIT

- ClientServer
- Composition
- Containment
- Cpu
- Dependency
- FileSystem
- FileSystemExport
- IIS Application Pool
- InstalledSoftware
- IpAddress
- IpServiceEndpoint
- Node
- OS User
- Process
- Realization
- RunningSoftware
- Usage
- Windows Device Driver
- WindowsService

## 第46章: ホストのリソースおよびアプリケーションのディスカバリ

### 本章の内容

概要 .....	712
トポロジ .....	712
ホストのリソースおよびアプリケーションの検出方法 .....	713
インストール済みソフトウェアの検出方法を以前のメカニズムに戻す方法 .....	714
ホストのリソースおよびアプリケーションのディスカバリ .....	715
トラブルシューティングおよび制限事項 .....	721

## 概要

**Hosts and Resources** モジュールは、ホスト上に存在するリソース(ディスク、CPU、ユーザなど)と、そのホストで実行されるアプリケーションを検出します。また、このモジュールは、アプリケーションと、関連するプロセス、該当するサービス、関連する IP サービス・エンドポイント(ポート)との間の関係も検出します。

**Host Resources by Shell/SNMP/WMI** ジョブおよび **Host Applications by Shell/SNMP/WMI** ジョブの役割は次のとおりです。

- 検出されたマシンの TCP 接続を、シェルまたは SNMP を使用して検出します。
- Data Flow Probe 専用の netflow データベースに情報を保存します。
- Data Flow Probe データベースに TCP 情報をクエリします。

また、**Host Resources by Shell** ジョブと **Host Applications by Shell** ジョブは、接続情報も収集します(netstat コマンドまたは **lsof** コマンドを実行)。

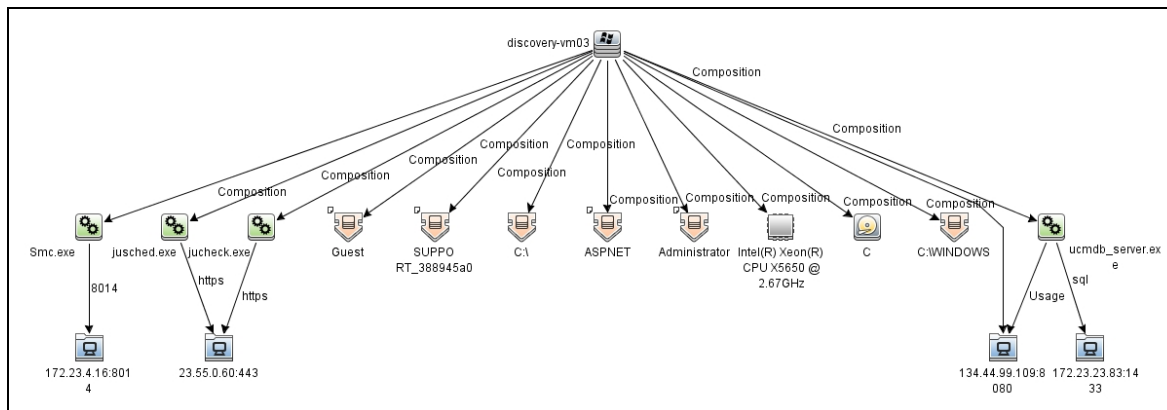
プロセスと、関連する IP サービス・エンドポイント(サーバ・ポート)との関係は、Windows 2003, Windows XP, SunOS, Hewlett-Packard UniX(HP-UX), AIX, Linux の各オペレーティング・システムで検出できます。

HP-UX と AIX マシンの場合は、<http://www.netadmintools.com/html/lsof.man.html> などからダウンロードできる lsof ソフトウェアをインストールする必要があります。**lsof** ソフトウェアは SunOS にインストールすることもできます。このソフトウェアをインストールしない場合は、SunOS にインストール済みの **pfiles** ソフトウェアが使用されます。

注: プロセス間 (P2P) ディスカバリとは、環境内のホストで実行中のプロセスを検出するディスカバリを指します。

## トポロジ

注: 検出された CIT のリストについては、「[検出された CIT](#)」(721 ページ)を参照してください。



## ホストのリソースおよびアプリケーションの検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このモジュールを実行するには、次のプロトコルを定義します。

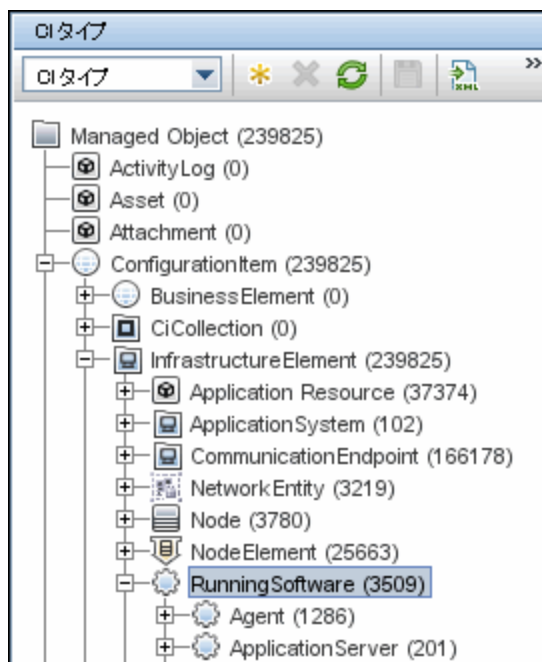
- NTCMD プロトコル
- SNMP プロトコル
- SSH プロトコル
- Telnet プロトコル
- WMI プロトコル

ユーザには、ルート権限は必要ありませんが、リモート・マシンに接続でき、該当するコマンドを実行できる適切な資格情報が必要です。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - その他

CMDB に Agent と Shell の CIT が含まれていることを確認します([モデリング] > [CI タイプ マネージャ])。RunningSoftware を検索して、Agent と Shell があることを確認します。



### 3. Host Resources by Shell/SNMP/WMI および Host Applications by Shell/SNMP/WMI デイスカバリの実行

[Universal Discovery] ウィンドウで目的の **Host Resources by Shell/SNMP/WMI** ジョブおよび **Host Applications by Shell/SNMP/WMI** ジョブをアクティブ化します。

前者のジョブは、ノード上に存在するリソース(ディスク, CPU, ユーザなど)を検出し、後者のジョブはそのホストで実行されるアプリケーションを検出します([「アプリケーション署名」\(680ページ\)](#)を参照)。ジョブは毎日実行するようにスケジュール設定されます。

## インストール済みソフトウェアの検出方法を以前のメカニズムに戻す方法

Host Applications by WMI ジョブは、WMI Windows インストーラ・プロバイダを使用してインストールされたソフトウェアを検出します。

ソフトウェアのインストールに Windows インストーラを使用しなかった場合は、以前のメカニズムを使用してそのソフトウェアを検出する必要があります。

**このジョブを以前のデイスカバリ・メカニズムに戻す方法:**

1. Host Resources by WMI および Host Applications by WMI アダプタにアクセスします([**アダプタ管理**] > [**Host\_Resources\_By\_WMI**] > [**アダプタ**] > [**WMI\_HR\_All**])。
2. [**アダプタ定義**] タブで, [**アダプタ パラメータ**] 表示 枠を見つけます。
3. **discoverInstalledSoftwareByOldMechanism** パラメータをダブルクリックして, 標準設定値の **false** を **true** に変更します。
4. 変更を保存します。

通信ログに警告メッセージが追加されます。

## ホストのリソースおよびアプリケーションのディスカバリ

本項の内容

- [「ジョブ・スレッド」](#)(715ページ)
- [「ロケールベース・プロセス」](#)(716ページ)
- [「Host Resources by Shell ジョブおよび Host Applications by Shell ジョブのアダプタ・パラメータ」](#)(717ページ)
- [「Host Resources by SNMP ジョブおよび Host Applications by SNMP ジョブのアダプタ・パラメータ」](#)(720ページ)
- [「Host Resources by WMI ジョブおよび Host Applications by WMI ジョブのアダプタ・パラメータ」](#)(720ページ)
- [「TCP ディスカバリ」](#)(720ページ)
- [「検出された CIT」](#)(721ページ)
- [「使用スクリプト」](#)(721ページ)

### ジョブ・スレッド

各ジョブは複数のスレッドを使って実行されます。ジョブを実行するときに同時に使用できるスレッドの最大数を定義できます。このボックスを空のままにすると, Data Flow Probe の標準設定のスレッド数(8)が使用されます。

標準設定の値は, **DataFlowProbe.properties** の **defaultMaxJobThreads** パラメータで定義します。

- **regularPoolThreads** : マルチスレッド・アクティビティに割り当てられているワーカ・スレッドの最大数です(標準設定値は 50)。
- **priorityPoolThreads** : 優先 ワーカ・スレッドの最大数です(標準設定値は 20)。

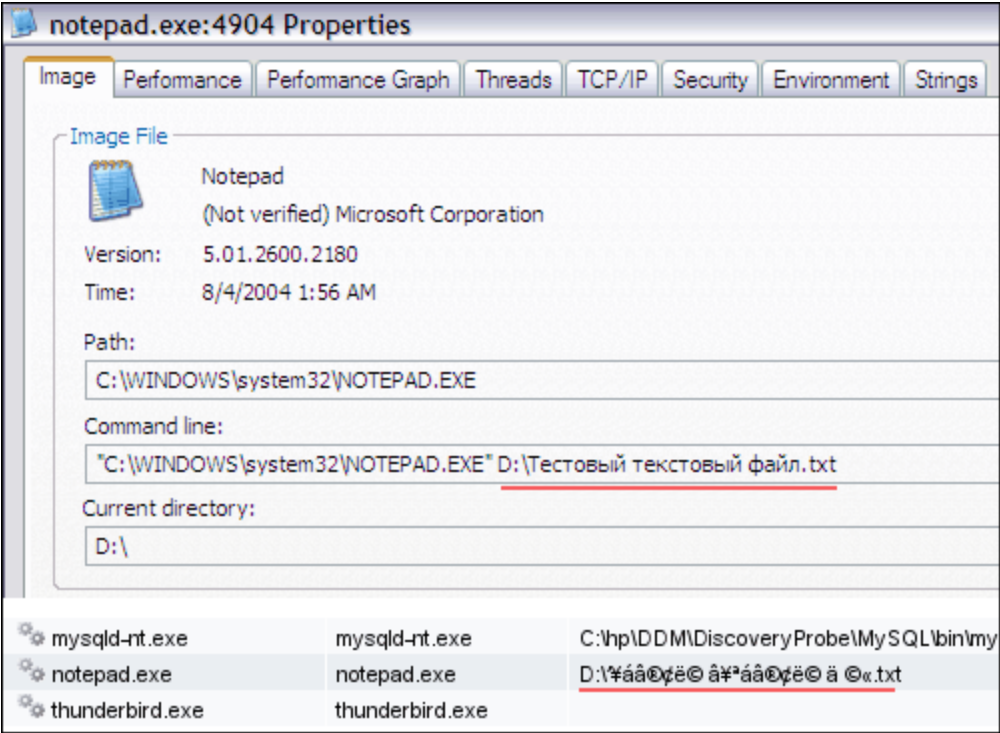
**注:**

- 実際のスレッド数は、常に regularPoolThreads + priorityPoolThreads 以下にします。
- このモジュールのジョブでは、Data Flow Probe の内部データベースに永続的に接続する必要があります。そのため、これらのジョブの最大同時実行スレッド数は 20(内部データベースに対して許可される最大同時接続数)に制限されています。
- 最大スレッド数のフィールドの詳細については、『HP Universal CMDB Data Flow Management Guide』の「Execution Options Pane」を参照してください。

**ロケールベース・プロセス**

このデイスカバリでは、リモート・マシンで使用されるロケールの検出を、既知のキーワードの検索、エンコーディングの調整、適切な正規表現と文字列の使用などの方法で行います。ただし、出力に複数言語の文字が含まれることがあり、そのような場合には文字化けが発生する可能性があります。たとえば、次の画像では、英語版の Windows マシンのコマンド・ラインに、ロシア語のファイル名を持つテキスト・ファイルが使用されています。





文字化けを防ぐために、このディスカバリでは **wmic** コマンドが使用されます。このコマンドは、ファイルを UTF-16 エンコーディングで保存します。これは、**globalSettings.xml** ファイルの **useIntermediateFileForWmic** パラメータで制御します([アダプタ管理]>[AutoDiscoveryContent]>[構成ファイル])。true に設定すると、このパラメータが有効化されます。標準設定値は false です。

### Host Resources by Shell ジョブおよび Host Applications by Shell ジョブのアダプタ・パラメータ

詳細については、『HP Universal CMDB Data Flow Management Guide』の「Adapter Parameters Pane」を参照してください。

パラメータ	説明
P2PServerPorts	これらのポートに(クライアントまたはサーバとして)接続されているプロセスのみが、ポートと一緒に検出されます。このパラメータには数値または既知の名前を含められます。カンマを使用してエントリを区切ります。アスタリスク(*)はすべてのポートを示します。標準設定値は * です。
discoverCPUs	CPU の検出を有効にします。  標準設定 :True

パラメータ	説明
<b>discoverDisks</b>	<p>ディスクの検出を有効にします。</p> <p><b>標準設定</b> :True</p>
<b>discoverFcHBAs</b>	<p>ファイバ・チャネルの HBA の検出を有効にします。</p> <p><b>標準設定</b> :False</p> <p>サポートされるプラットフォームおよびプロトコル:</p> <ul style="list-style-type: none"> <li>• Windows, シェルベースのプロトコルおよび WMI プロトコル</li> <li>• SunOS, シェルベースのプロトコル</li> <li>• HP-UX, シェルベースのプロトコル</li> </ul>
<b>discoverInstalledSoftware</b>	<p>ソフトウェアのディスカバリを実行します。true に設定すると、インストールされているすべてのソフトウェアが、署名の有無を問わずにレポートされます。</p> <p><b>標準設定</b> :False</p> <div> <p><b>注:</b> このパラメータを true に設定すると、対象マシンのパフォーマンスに悪影響が出る可能性があります。</p> </div>
<b>discoverMemory</b>	<p>メモリの検出を有効にします。</p> <p><b>標準設定</b> :True</p>
<b>discoverP2P</b>	<p>実行中のプロセスの TCP 接続をレポートするかどうかを指定します。</p> <ul style="list-style-type: none"> <li>• True: 実行中のプロセスの TCP 接続がレポートされます。</li> <li>• False: 実行中のプロセスの TCP 接続はレポートされません。</li> </ul> <p><b>標準設定</b> :True</p>
<b>discoverProcesses</b>	<ul style="list-style-type: none"> <li>• <b>False</b>: 次の 2 つのシナリオに該当する場合に、プロセスがレポートされます。 <ul style="list-style-type: none"> <li>◦ <b>discoverP2P</b> パラメータを true に設定している場合は、<b>discoverProcesses</b> パラメータの値に関係なく、TCP 接続に関連するプロセスがレポートされます。</li> <li>◦ <b>discoverRunningSW</b> パラメータを true に設定している場合は、<b>discoverProcesses</b> パラメータの値に関係なく、実行中のソフトウェアに関連するプロセスがレポートされます(実行中のソフトウェアは、<b>applicationsSignature.xml</b> ファイルで指</li> </ul> </li> </ul>

パラメータ	説明
	<p>定します)。</p> <ul style="list-style-type: none"> <li>• <b>True</b> : すべてのプロセスがレポートされます。</li> </ul> <p>標準設定 :False</p>
<b>discoverRunningSW</b>	<p>実行中のソフトウェアを検出するかどうかを指定します。</p> <p>標準設定 :False</p>
<b>discoverServices</b>	<ul style="list-style-type: none"> <li>• <b>False</b> : サービスはレポートされません。</li> <li>• <b>True</b> : すべてのサービスが検出されます。</li> </ul> <p>標準設定 :False</p>
<b>discoverShares</b>	<p><b>True</b> : 共有リソースが検出されて、<b>FileSystemExport</b> CIT が作成されます。</p> <p>標準設定 :True</p>
<b>discoverUsers</b>	<p>ユーザの検出を有効にします。</p> <div> <p>注: Windows マシンの場合は、ローカル・ユーザのみが検出されます。</p> </div> <p>標準設定 :True</p>
<b>filterP2PProcessesByName</b> (旧 filterProcessesByName)	<p>レポートされないプロセスの名前を示します。</p> <div> <p>注: このパラメータは、P2P ディスカバリにのみ影響します。</p> </div> <p>標準設定 :system,svchost.exe,lsass.exe,System Idle Process</p> <p>P2P が実行されないようにするには、値にアスタリスク(*)を入力します。</p>
<b>ignoreP2PLocalConnections</b>	<p><b>False</b> : P2P ディスカバリでローカル接続が考慮されます。つまり、クライアントとサーバが同じホストにインストールされていて、それらの間がクライアントとサーバの関係で接続されている場合は、P2P ディスカバリでこの関係がレポートされます。</p>
<b>ignoreUnnamedPorts</b>	<p>サービス名を含まないでポートをレポートするかどうかを指定します。</p> <p>標準設定 :False</p>
<b>knownListeningPorts</b>	<p>(*すべてをレポートする)IP サービス・エンドポイントとしてレポートされる必要があるポート番号またはポート名です。</p>

パラメータ	説明
<b>IssofPath</b>	UNIX マシンのプロセス通信の検出を可能にする lsof コマンドのパスです。標準設定値は <code>/usr/local/bin/lsof,lsof,/bin/lsof</code> です。
<b>useLSOF</b>	<b>True</b> : UNIX マシン上のポートからプロセスへのマッピングを検出するため、lsof ユーティリティの使用が試行されます。 <b>標準設定</b> : True
<b>useNetstatOnly</b>	追加コマンド (LSOF) を実行するか、または、netstat コマンドのみを使用するかを指定します。 <b>標準設定</b> : False
<b>wmicPath</b>	Windows の wmic.exe ユーティリティの場所を指定します。 <b>標準設定</b> : %SystemRoot%\system32\wbem\
<b>workInTopDown</b>	Top-Down ディスカバリかどうかを指定します。 <b>標準設定</b> : False

## Host Resources by SNMP ジョブおよび Host Applications by SNMP ジョブのアダプタ・パラメータ

パラメータの定義については、「[Host Resources by Shell ジョブおよび Host Applications by Shell ジョブのアダプタ・パラメータ](#)」(717 ページ)を参照してください。

## Host Resources by WMI ジョブおよび Host Applications by WMI ジョブのアダプタ・パラメータ

パラメータの定義については、「[Host Resources by Shell ジョブおよび Host Applications by Shell ジョブのアダプタ・パラメータ](#)」(717 ページ)を参照してください。

## TCP ディスカバリ

**クライアント / サーバの関係** : 2 つの宛先間の接続 (IP とポートの組合せ) を確認する際に、DFM では、次のロジックを使用してどちら側がサーバでどちら側がクライアントかを判断します (重要度で降順)。

- ポートのいずれかがリスナ・ポートの場合 (つまり、port\_process テーブル内でリスナとしてマークされている)、そのポートはサーバ・ポートです。
- サーバ・プロセスであることがわかっているプロセスが、ポートのいずれかを使用している場合、そのポートはサーバ・ポートです。

- ローカル・ポートがリッスンしておらず、リモート側がまだ処理されていない(リモート側で TCP ディスカバリが未実行)場合、リモート・ポートがサーバ・ポートであるとみなされます。
- どちらのポートもリッスンしておらず、サーバ・プロセスであることがわかっているプロセスがない場合、DFM では P2P 接続がレポートされません。

## 検出された CIT

- hardware\_board
- environmental\_sensor
- fan
- power\_supply
- printer\_toner
- printer\_tray

検出されたすべての CIT を表示するには、[リソース] 表示 枠で特定のアダプタを選択します。

詳細については、『HP Universal CMDB Data Flow Management Guide』の「Discovered CITs Pane」を参照してください。

注: トポロジを表示するには、[「トポロジ」\(712ページ\)](#)を参照してください。

## 使用スクリプト

- snmp\_model\_finder.py
- snmp\_model\_discovery.py

注: Host Resource by SNMP ジョブにのみ適用されます。

## トラブルシューティングおよび制限事項

本項では、ホストのリソースおよびアプリケーションのディスカバリのトラブルシューティングと制限事項について説明します。

- Solaris マシンで実行中のプロセスおよびソフトウェアを検出するには、Solaris マシンに **/usr/ucb/ps** ユーティリティがインストールされていることを確認します。
- 名前にスペースが含まれるプロセスの検出は、UNIX マシンではサポートされません。
- UNIX マシンから ssh クライアントおよび telnet クライアントによって取り込まれた英語以外のコンテンツの検出はサポートされません。
- 英語以外のロケールのユーザによってインストールされたソフトウェアは、インストール日がレポートされません。
- WMI によってインストールされたソフトウェアを DFM が検出し、そのソフトウェアに名前が定義されていない場合、DFM はそのソフトウェア・エンティティを CMDB にレポートしません。
- ホスト・リソース(プロセス、Windows サービス、ユーザ、インストール済みソフトウェアなど)の名前または説明にマルチバイト文字が使用されている場合、**Host Resource by SNMP** ジョブおよび **Host Applications By SNMP** ジョブの出力結果には、文字化けしたマルチバイト文字が含まれます。
- VMware ESX サーバではプロセスの検出はサポートされません。
- Mac OS X では、CPU、ディスク、メモリの検出はサポートされません。

## 第47章: IBM i by Eview ディスカバリ

### 本章の内容

概要 .....	724
ディスカバリの領域 .....	724
サポートされるバージョン .....	725
トポロジ .....	725
IBM i Resources .....	725
IBM i Objects .....	726
ディスカバリ・メカニズム .....	727
iSeries の検出方法 .....	727
IBM i Connection ジョブ .....	727
IBM i Objects ジョブ .....	728
IBM i Resources ジョブ .....	731

## 概要

IBM i by Eview ディスカバリは、iSeries(AS400)サーバ用の、iSeries エージェントを全面的にベースとした ディスカバリです。EView Technology の iSeries クライアントおよびエージェントを使用して、iSeries システム上で ディスカバリを実行します。ディスカバリを実行するには、EView エージェントを iSeries ノードにインストールします。

**注:** インストールの説明については、EView 400 iSeries のドキュメントを参照してください。

iSeries EView クライアントは、IBM i by Eview ディスカバリ・ジョブの実行に使用する各プロンプトにインストールします。

## ディスカバリの領域

### • IBM i Resources

- ASP を持つローカル・ストレージ
- メモリ
- LPAR
- CPU
- ネットワーク接続
- インストール済みソフトウェア
- 選択済みシステム値
- サブシステム
- アクティブ・ジョブ

### • IBM i Objects

- ジョブ・キュー
- 出力キュー



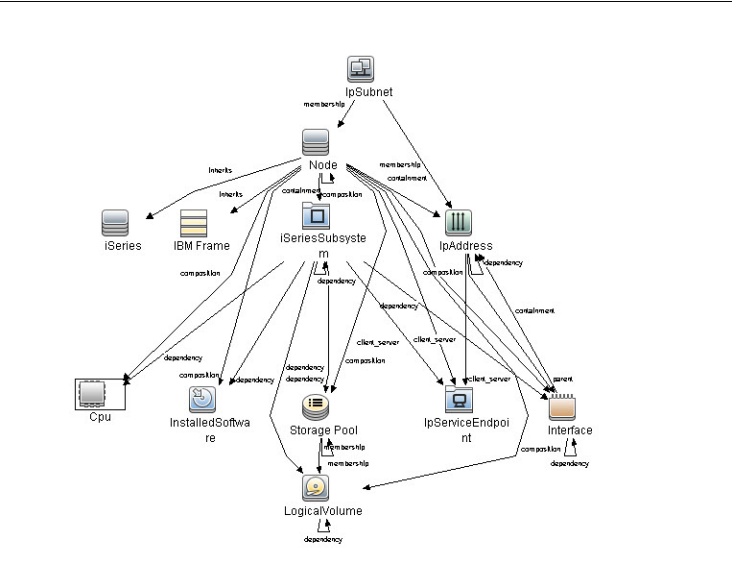
- 。 ライブラリ
- 。 プログラム・オブジェクト

サポートされるバージョン

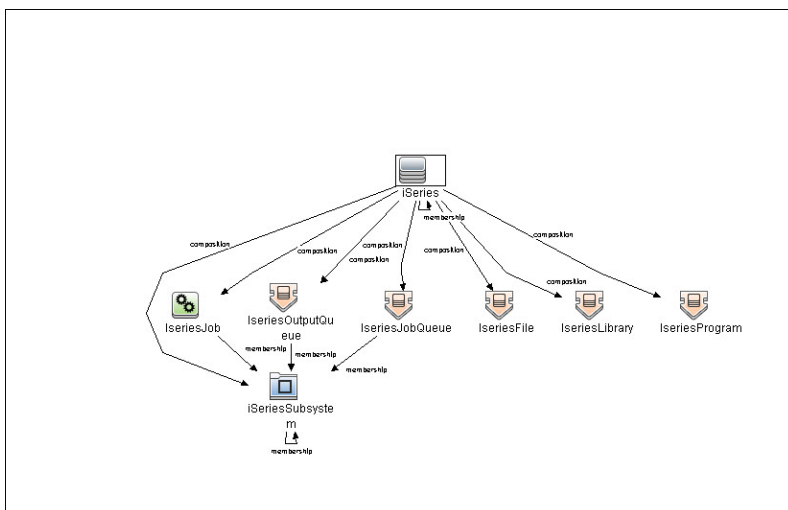
UCMDB バージョン	iSeries バージョン
9.x	V5R1M0 以降の OS/400 リリース

トポロジ

IBM i Resources



## IBM i Objects



## ディスカバリ・メカニズム

このディスカバリ・ジョブでは、EView 400 クライアントおよびエージェントが使用されます。ジョブがアクティブ化されると、ディスカバリ・スクリプトは、プローブにインストールされている EView 400 クライアントを使用します。EView 400 クライアントは、ローカル・シェルとしてアクセスされます。

EView 400 クライアントは、スクリプトから発行されたコマンドを、iSeries ノードで実行中の EView 400 エージェントに送ります。これらのコマンドは、OS/400 コマンドおよび EView エージェント・コマンドです。コマンド実行の結果はクライアントに返され、その後呼び出しスクリプトに渡されます。

## iSeries の検出方法

このタスクでは、EView クライアントおよびエージェントを使用して iSeries CI を検出する方法について説明します。

### 1. 前提条件

**EView エージェント**を iSeries 側にインストールし、**EView クライアント**を DFM プローブ側にインストールします。その方法については、『EView 400 Discovery Installation Guide』を参照してください。

### 2. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **IBM i Connection** ジョブを実行してターゲット iSeries ホストを検出します。
- c. **IBM i Resources** ジョブを実行して、iSeries LPAR のリソース情報を検出します(CPU、メモリ、補助ストレージ・プールおよびディスク、サブシステム、ネットワーク接続性など)。
- d. **IBM i Objects** ジョブを実行して iSeries LPAR のオブジェクト情報を検出します(キュー、ジョブ、プログラム・オブジェクト、ライブラリなど)。

## IBM i Connection ジョブ

本項ではこのジョブに関する詳細を説明します。

### 入力 CIT

- Discovery Probe Gateway

使用スクリプト

- eview400\_connection.py
- eview400\_lib.py
- file\_mon\_utils.py
- file\_ver\_lib.py

検出された CIT

- composition
- containment
- EView
- IpAddress
- iSeries

アダプタ・パラメータ

パラメータ	説明
EViewInstallationFolder	プローブ・サーバ上の EView クライアントのインストール・ルート・ディレクトリ
debugMode	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、プローブのデバッグ・ログで詳細ログ記録が有効化されます。  標準設定 : false

IBM i Objects ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ TQL

入力 CIT :EView

## トリガ CI データ

名前	値
ApplicationPath	\${SOURCE.application_path:NA}
LparName	\${HOST.name}
NodeName	\${SOURCE.discovered_product_name}

## 使用スクリプト

- eview400\_objects.py
- eview400\_lib.py

## 検出された CIT

- Composition
- IseriesFile
- IseriesJob
- IseriesJobQueue
- IseriesLibrary
- IseriesOutputQueue
- IseriesProgram
- Membership
- iSeries
- iSeriesSubsystem

## アダプタ・パラメータ

パラメータ	説明
<b>commandTimeout</b>	<p>EView エージェントに対して発行されたコマンドがタイムアウトするタイムアウト値(秒単位)を示します。</p> <p><b>標準設定</b> : 60</p>
<b>debugMode</b>	<p>この値は <b>true</b> または <b>false</b> に設定できます。<b>true</b> に設定すると、プローブのデバッグ・ログで詳細ログ記録が有効化されます。</p> <p><b>標準設定</b> : false</p>
<b>discover_Jobs</b>	<p>この値は <b>true</b> または <b>false</b> に設定できます。<b>true</b> に設定すると、iSeries LPAR 上のアクティブ・ジョブが検出されます。</p> <p><b>標準設定</b> : false</p>
<b>discover_Library</b>	<p>この値は <b>true</b> または <b>false</b> に設定できます。<b>true</b> に設定すると、iSeries ライブラリ・オブジェクトが検出されます。</p> <p><b>標準設定</b> : true</p>
<b>discover_Program</b>	<p>この値は <b>true</b> または <b>false</b> に設定できます。<b>true</b> に設定すると、iSeries プログラム・オブジェクトが検出されます。</p> <p><b>標準設定</b> : false</p> <div> <p><b>注:</b> プログラム・オブジェクトの検出には時間がかかります。このため、<b>true</b> に設定する場合は、<b>commandTimeout</b> パラメータの値を増やすことをお勧めします。</p> </div>
<b>discover_Queue</b>	<p>この値は <b>true</b> または <b>false</b> に設定できます。<b>true</b> に設定すると、キュー(ジョブ、出力)が検出されます。</p> <p><b>標準設定</b> : true</p>

# IBM i Resources ジョブ

本 項 ではこのジョブに関する詳 細を説 明します。

## トリガ TQL クエリ

入 力 CIT :EView

## トリガ CI データ

名 前	値
ApplicationPath	\${SOURCE.application_path:NA}
LparName	\${HOST.name}
NodeName	\${SOURCE.discovered_product_name}

## 使用スクリプト

- eview400\_resources.py
- eview400\_lib.py

## 検出された CIT

- ClientServer
- Composition
- Containment
- Cpu
- Dependency
- IBM Frame
- InstalledSoftware
- Interface

- IpAddress
- IpServiceEndpoint
- IpSubnet
- LogicalVolume
- Membership
- Node
- Parent
- Storage Pool
- iSeries
- iSeries Subsystem

## アダプタ・パラメータ

パラメータ	説明
<b>commandTimeout</b>	EView エージェントに対して発行されたコマンドがタイムアウトするタイムアウト値(秒単位)を示します。  標準設定 : 60
<b>debugMode</b>	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、プローブのデバッグ・ログで詳細ログ記録が有効化されます。  標準設定 : false
<b>discover_ASP</b>	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、補助ストレージ・プールおよびディスク・ユニットが検出されます。  標準設定 : false
<b>discover_CPUs</b>	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、iSeries LPAR CPU CI が検出されます。  標準設定 : true
<b>discover_Network</b>	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、iSeries インタフェース CI が検出されます。  標準設定 : true



パラメータ	説明
discover_Software	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、iSeries インストール済みソフトウェア CI が検出されます。  標準設定 : false
discover_Subsystems	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、iSeries サブシステム CI が検出されます。  標準設定 : true
discover_TCP_UDP	この値は <b>true</b> または <b>false</b> に設定できます。 <b>true</b> に設定すると、iSeries LPAR TCP ポートおよび接続性と、UDP ポートが検出されます。  標準設定 : false

## 第48章: インベントリ・ディスカバリ

インフラストラクチャ・ディスカバリを含むインベントリ・ディスカバリでは、どのデバイスがネットワーク内にあるか決定し、それぞれについての情報を収集します。また、その他のディスカバリ・モジュールの基盤の役割も果たします。

通常、インベントリ・ディスカバリは、ウィザードによるアクティビティを使用して、管理ゾーンに基づき管理および実行します。インベントリ・ディスカバリ・アクティビティの詳細については、『HP UCMDB Discovery and Integrations Content Guide - Discovery Activities』ドキュメントの「Inventory Discovery Activity」を参照してください。その他の情報については、『HP Universal CMDB Data Flow Management Guide』のインベントリ・ディスカバリに関する項を参照してください。

# 第49章: NetApp Filer ディスカバリ

## 本章の内容

- 概要 ..... 736
- サポートされるバージョン ..... 736
- トポロジ ..... 736
- NetApp Filer の検出方法 ..... 738
- NetApp Filer by WebServices ジョブ ..... 741

## 概要

HP Universal CMDB では、NetAppのNAS情報を NetApp Filer から直接取得できます。検出には、UCMDB データベース(CMDB)内のデバイス、トポロジ、ストレージ・インフラストラクチャ階層の同期が含まれます。この検出の結果を活用して、UCMDB にマッピングされたすべてのビジネス・サービスにわたる変更管理および影響分析を、ストレージの観点から行うことができます。

この検出では、NetApp Filer WebService APIによるUCMDB主導の検出を伴います。また、さまざまなハードウェア間の物理的な関係や、論理ボリュームとハードウェア・デバイス間の論理的关系が同期され、ストレージ・インフラストラクチャのエンド・ツー・エンドのマッピングが可能になります。

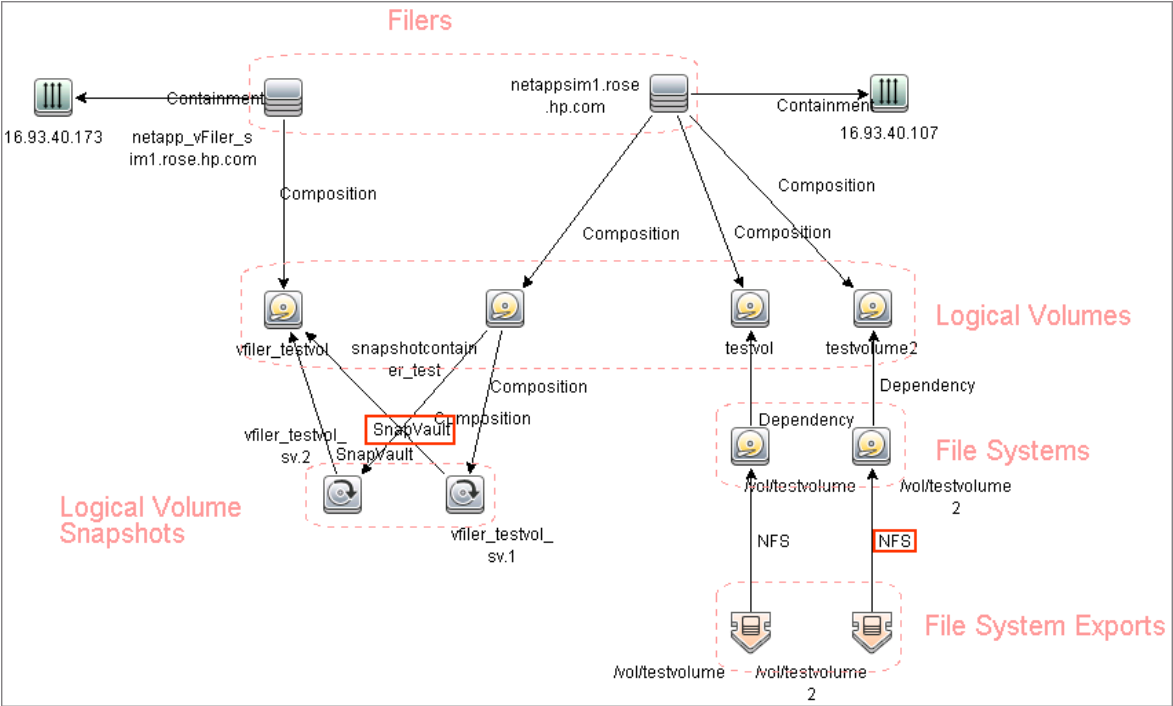
## サポートされるバージョン

この検出では、NetApp Data ONTAP 7.2.x, 7.3.x, ONTAP SDK 3.5.1 がインストールされた 8.x, ONTAP SDK 5.3.1 がインストールされた 8.x がサポートされます。

## トポロジ

次の図は、NetApp Filer デイスカバリのトポロジとサンプル出力を示します。

注: 検出された CIT のリストについては、[「検出された CIT」\(742ページ\)](#)を参照してください。



# NetApp Filer の検出方法

このタスクでは、NetApp Filer の検出方法について説明します。

## 1. 前提条件 - プロトコル資格情報の設定

この検出には、NetApp WebService 用の NetApp プロトコルが含まれます。NetApp プロトコルを使用するには、NetApp WebService API への適切な資格情報とポートを構成します。NetApp ONTAP SDK を使用して NetApp Filer から情報を取得します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - 権限

注: ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

Netapp Filer システムで次の検出コマンドを実行するための十分な権限をユーザが持っていることを確認します。

コマンド	説明
login-http-admin	検出に必要な権限です。この権限がないと、Netapp WebServices API を使用して NetApp Filer への認証またはログインを行えません。
api-system-get-info	CPU やバックプレーンの情報などの、アプライアンスの詳細を取得します(sysconfig -a コマンドのヘッダ情報)。I/O 情報は含まれません。
api-system-get-ontapi-version	ONTAPI の最新のメジャー・バージョンとマイナー・バージョンを取得するために必要です。
api-ipspace-list-info	IP アドレスと関連する IP の詳細などの ipspace に関する情報を取得します(vfiler ライセンスが必要)。
options-get	オプション・パラメータの値を取得します。
api-volume-list-info-iter-start	アプライアンスのボリュームに関する詳細を取得します。

コマンド	説明
api-volume-list-info-iter-next api-volume-list-info-iter-end	
api-snapshot-list-info	特定のボリュームのスナップショットに関する詳細を取得します。
api-snapvault-<SnapvaultLevel> -relationship-status-list-iter-start api-snapvault-<SnapvaultLevel> -relationship-status-list-iter-next	アプライアンスの snapvault に関する詳細を取得します。<SnapvaultLevel> の値には、 <b>primary</b> または <b>secondary</b> 、あるいはその両方を指定できます。
api-cifs-share-list-iter-start api-cifs-share-list-iter-next api-cifs-share-list-iter-end	このアプライアンスの CIFS 共有に関する詳細を取得します(cifs ライセンスが必要)。
api-cifs-session-list-iter-start api-cifs-session-list-iter-next api-cifs-session-list-iter-end	このアプライアンスの CIFS セッションに関する詳細を取得します(cifs ライセンスが必要)。
api-nfs-exportfs-list-rules	このアプライアンスの NFS 共有に関する詳細を取得します。
api-security-api-vfiler api-nfs-exportfs-list-rules-2	vFiler に関する詳細を取得します。
api-system-cli api-cli-ifconfig	ネットワーク・インタフェースに関する詳細を取得します。

### 3. ディスカバリの実行

**注:** ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次のジョブを次の順序で実行します。

- a. **Range IPs by ICMP** ジョブを実行します。
- b. **Host Connection by SNMP** ジョブを実行して NetApp Filer を識別します。

- c. **NetApp Filer by WebServices** ジョブを実行します。ジョブの詳細については、[「NetApp Filer by WebServices ジョブ」](#)(741ページ)を参照してください。



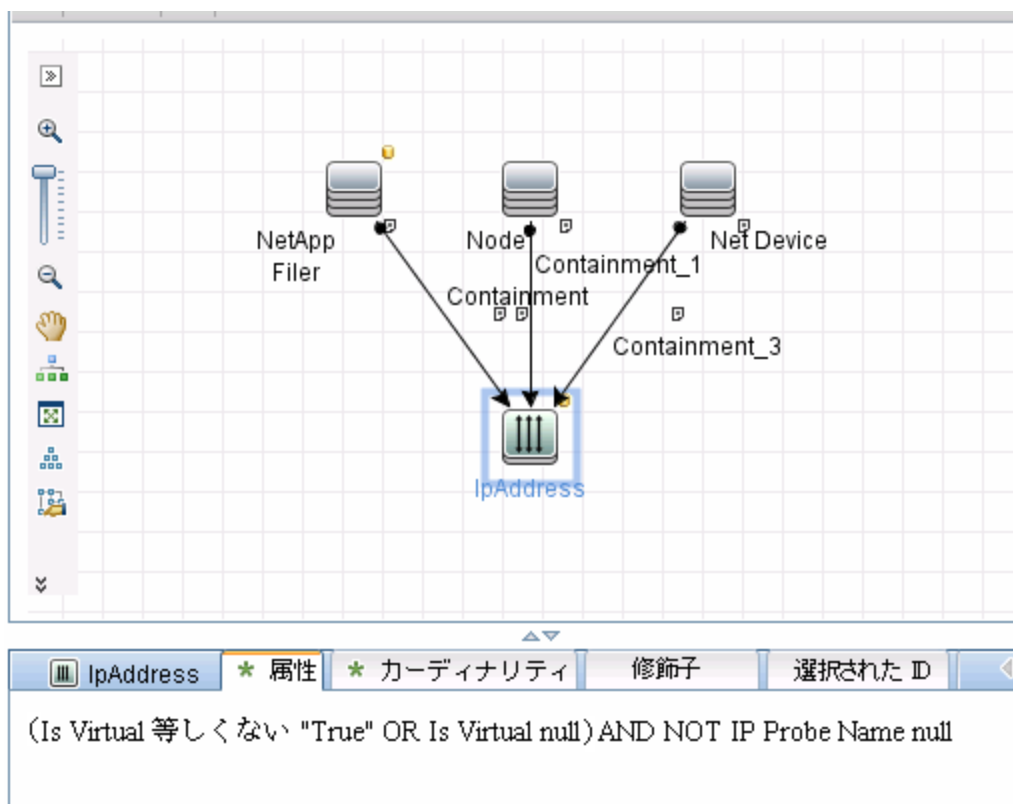
## NetApp Filer by WebServices ジョブ

NetApp Filer 検出パッケージは **NetAppFiler.zip** に含まれています。

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

このトリガ TQL では、**Net Device** と **Node** で **[サブタイプを含める]** オプションが選択されていません。これにより、NetApp Filer ではない (Windows, UNIX などの) CI に関連付けられている IP が除外されます。



### アダプタ

このジョブでは、**NetApp Filers by WebServices** アダプタを使用します。

- 入力クエリ: なし
- アダプタ・パラメータ

パラメータ	説明
<b>getNetworkShareInfo</b>	<p><b>True</b> : ネットワーク共有の検出が実行されます。</p> <p><b>False</b> : ネットワーク共有の検出は実行されません。</p>
<b>getSnapshotInfo</b>	<p><b>True</b> : 論理ボリューム・スナップショットの検出が実行されます。</p> <p><b>False</b> : 論理ボリューム・スナップショットの検出は実行されません。</p>
<b>getSnapVaultInfo</b>	<p><b>True</b> : SnapVault の検出が実行されます。</p> <p><b>False</b> : SnapVault の検出は実行されません。</p>
<b>chunksize</b>	<p>SOAP 呼び出しごとに NetApp Operations Manager から取得される最大オブジェクト数です。</p> <p>NetApp Filer の負荷を減らすために、このパラメータを 1000(標準設定値)よりも低い値に設定します。</p>
<b>filerOptions</b>	<p>NetApp Filer の[Options]フィールドで定義されている NetApp Filer の追加パラメータおよび設定を検出します。</p> <p>このパラメータには、検出する追加 vFiler オプションの名前をカンマで区切って指定できます。これらのオプションの値は、UCMDB の NetApp Filer クラスの[Options]属性に保存されます。</p> <p><b>例</b> : nfs.tcp.recvwindowsize,nfs.tcp.xfersize,nfs.tcp.enable</p>

## 検出された CIT

- Composition
- Containment
- Cpu
- Dependency
- FileSystem
- FileSystemExport
- Interface

- IpAddress
- Logical Volume Snapshot
- LogicalVolume
- Membership
- Netapp Filer
- Node
- Realization

注: トポロジを表示するには, [「トポロジ」\(736ページ\)](#)を参照してください。

## 第50章: SMI-S ディスカバリ

### 本章の内容

概要 .....	745
サポートされるバージョン .....	745
ディスカバリ・メカニズム .....	745
SMI-S の検出方法 .....	745
Storage Devices Connection by CIM ジョブ .....	746
Storage Devices Connection by CIM アダプタ .....	747
Storage Devices Topology by CIM ジョブ .....	749
Storage Devices Topology by CIM アダプタ .....	751

## 概要

Storage Networking Industry Association(SNIA)は、SMI-S(Storage Management Initiative Specification)を、ストレージ・エリア・ネットワーク(SAN)管理の標準的な方法として進化および発展させました。ISO 規格であるこの仕様には、WEBM クライアントを介して CIM サーバからアクセス可能なデータが含まれます。

このパッケージは、CIM プロトコルを使用したストレージ関連データを検出します。

## サポートされるバージョン

このディスカバリは、SMI-S バージョン 1.50 をサポートしており、TPD, CIMV2, EVA, LsiArray13, EMC の名前空間をサポートしています。さらに、CIM 名前空間への汎用的なサポートを提供します。

## ディスカバリ・メカニズム

このディスカバリでは、CIM プロトコルを使用して、ストレージ管理と CIM エージェントを実行しているターゲット・マシンに接続します。接続後は、CIM クラスをカウントし、属性を解析し、UCMDB クラス・モデルへのマッピングを行います。

## SMI-S の検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、CIM プロトコルを使用します。CIM プロトコル・エントリを、ポート、名前空間、http / https トランスポートと一緒に定義します。

**注:** CIM プロトコルは、http と https のサブ・トランスポートをサポートします。サポートされる名前空間は、次のとおりです。

- **root/tpd**
- **root/cimv2**(限定サポート)

- **root/eva**
- **root/LsiArray13**
- **root/emc**

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. ジョブの実行

- a. **Range IPs by ICMP** ジョブを実行して、SMI-S で使用されるサーバの IP アドレスを検出します。
- b. **Storage Devices Connection by CIM** ジョブを実行して、適切な資格情報を検出し、CIM CI を作成します。
- c. **Storage Devices Topology by CIM** ジョブを実行してストレージ・トポロジを検出します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

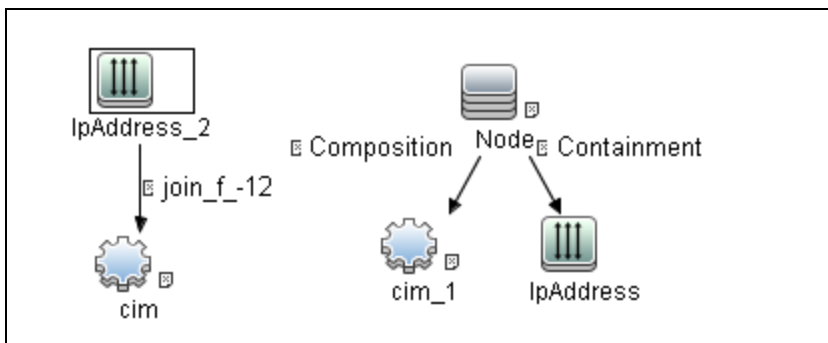
# Storage Devices Connection by CIM ジョブ

本項ではこのジョブに関する詳細を説明します。

## アダプタ

このジョブでは、**Storage Devices Connection by CIM** アダプタを使用します。

## トリガ TQL



## ディスカバリ・フロー

この検出では、名前空間に応じて CIM クラスのいずれかのインスタンスを取得することが可能かどうかを確認するためのチェックが行われます。

- 名前空間 : "root/cimv2" -> CIM\_OrganizationalEntity
- 名前空間 : "root/tpd" -> TPD\_StorageSystem
- 名前空間 : "root/eva" -> HPEVA\_StorageSystem
- 名前空間 : "root/LsiArray13" -> LSISSI\_StorageSystem
- 名前空間 : "root/emc" -> EMC\_ComputerSystem

## Storage Devices Connection by CIM アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

IP Address

入力 TQL クエリ



トリガ CI データ

名前	値
ip_domain	\${SOURCE.routing_domain}
ip_address	\${SOURCE.name}

使用スクリプト

- cim.py
- entity.py
- smis\_discoverer.py
- smis\_connection.py
- cim\_discover.py
- smis.py

検出された CIT

- CIM
- Composition
- Containment
- IpAddress
- Node



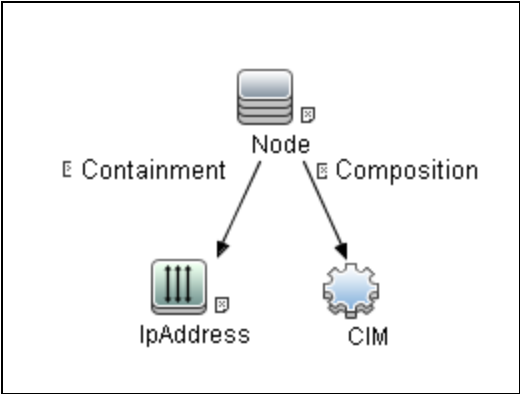
# Storage Devices Topology by CIM ジョブ

本 項 ではこのジョブに関する詳 細を説 明します。

## アダプタ

このジョブでは、 **Storage Devices Topology by CIM** アダプタを使用します。

## トリガ TQL クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
IpAddress	NOT UcmdbRoutingDomain null
CIM	NOT Reference to the credentials dictionary entry null
CIM	CimCategory 含む Storage

## ディスカバリ・フロー

この検出 では、 取得および解析されたインスタンスのクラスが、 名 前 空間ごとに一 覧 表 示 され ます。

- 名 前 空間 "root/tpd"
  - TPD\_AllocatedFromStoragePool
  - TPD\_DynamicStoragePool
  - TPD\_FCPort

- TPD\_NodeSystem
- TPD\_SCSIController
- TPD\_StoragePool
- TPD\_StorageSystem
- TPD\_StorageVolume
- **名前空間 "root/cimv2"**
  - CIM\_FCPort
  - CIM\_NodeSystem
  - CIM\_StoragePool
  - CIM\_StorageSystem
  - CIM\_StorageVolume
- **名前空間 "root/eva"**
  - HPEVA\_StoragePool
  - HPEVA\_StorageSystem
  - HPEVA\_DiskFCPort
  - HPEVA\_StorageVolume
  - HPEVA\_ProtocolControllerForVolume
  - HPEVA\_ViewProtocolController
  - HPEVA\_StorageProcessorSystem
  - HPEVA\_DiskExtent
- **名前空間 "root/LsiArray13"**
  - LSISSI\_StoragePool
  - LSISSI\_StorageSystem

- LSISSI\_FCPort
- LSISSI\_StorageVolume
- LSISSI\_ControllerCanister
- LSISSI\_ControllerFirmwareIdentity
- LSISSI\_StorageProcessorSystem
- LSISSI\_DiskExtent
- 名前空間 "root/emc"
  - EMC\_StoragePool
  - EMC\_StorageSystem
  - EMC\_FCPort
  - EMC\_StorageVolume
  - EMC\_StorageSystemSoftwareIdentity
  - EMC\_ArrayChassis
  - EMC\_SCSIProtocolController
  - EMC\_StorageProcessorSystem
  - EMC\_DiskExtent

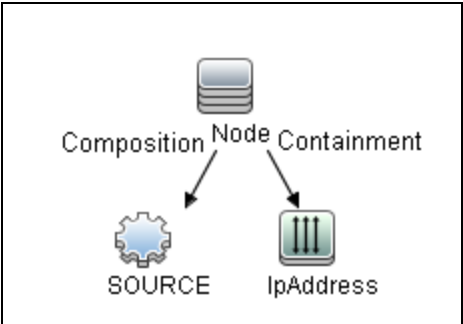
## Storage Devices Topology by CIM アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

CIM

入力 TQL クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null

使用スクリプト

- cim.py
- entity.py
- smis.py
- smis\_discoverer.py
- smis\_topology.py
- cim\_discover.py

検出された CIT

- Composition
- Fibre Channel Connect
- Fibre Channel Port
- IpAddress
- LogicalVolume

- Node
- Storage Array
- Storage Pool

## 第8部: Mainframe

## 第51章: EView Agent ディスカバリ

### 本章の内容

概要 .....	756
サポートされるバージョン .....	756
トポロジ .....	757
EView Connection .....	757
LPAR Resources by EView .....	758
CICS by EView .....	758
DB2 by EView .....	759
IMS by EView .....	760
MQ by EView .....	761
EView によるメインフレームの検出 方法 .....	762
ディスカバリ・メカニズム .....	763
LPAR Resources by EView ジョブ .....	764
EView Connection ジョブ .....	765
CICS by EView ジョブ .....	766
DB2 by EView ジョブ .....	767
IMS by EView ジョブ .....	768
MQ by EView ジョブ .....	769
トラブルシューティングおよび制限事項 .....	769

## 概要

多くのエンタープライズ・アプリケーションが、メインフレームと分散型環境(Linux, UNIX, Windows)の両方で使用されています。メインフレームが関与するレベルが低い場合(基盤データベース・ソリューション用など)もあれば、たとえば、実行待ち、中間層アプリケーション、複数のメインフレーム・サブシステムをサポートするホストとして機能する環境のように、メインフレームが分散環境側よりも多くのアプリケーションの基盤となっている場合もあります。

HP データ・フロー管理(DFM)の目的は、アプリケーションがどこに配置されているかにかかわらず、インフラストラクチャ全体でアプリケーションを適切にマッピングすることにあります。インフラストラクチャ全体でのアプリケーションのマッピングは、通常次の3つの部分から構成されます。

1. インフラストラクチャの検出
2. アプリケーションの検出
3. アプリケーション依存関係のマッピング

現在メインフレーム上で動作する検出ソリューションは、z/OS ホストおよびネットワーク・リソースと、DB2, IMS, CICS, MQ などのアプリケーションを検出することによって、上記の最初の2つの部分に対応します。

Mainframe by EView ディスカバリは、エージェント・ベースのディスカバリ・ソリューションです。このディスカバリでは、メインフレーム・トポロジの検出のために、**EView/390z Discovery for z/OS** というアプリケーションが使用されます。

ディスカバリ・メカニズムの詳細については、[「ディスカバリ・メカニズム」\(763ページ\)](#)を参照してください。

ディスカバリを実行するには、[「EView によるメインフレームの検出方法」\(762ページ\)](#)を参照してください。

## サポートされるバージョン

ターゲット・プラットフォーム	バージョン
z/OS	1.8, 1.9, 1.10, 1.11, 1.12
DB2 for z/OS	8, 9
CICS	3.x, 4.x
z/OS での WebSphere MQ	6.0, 7.0
IMS	9 以降

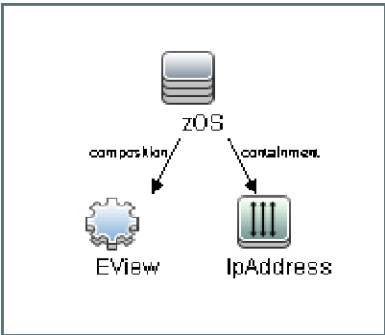


# トポロジ

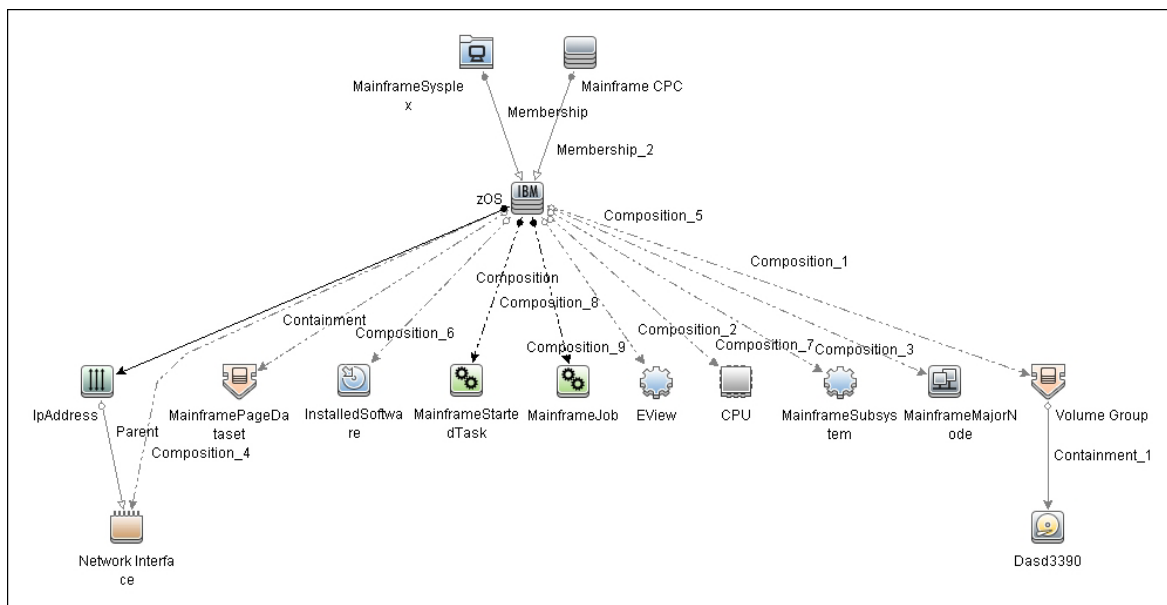
本項では、次のジョブのトポロジ・マップを示します。

EView Connection .....	757
LPAR Resources by EView .....	758
CICS by EView .....	758
DB2 by EView .....	759
IMS by EView .....	760
MQ by EView .....	761

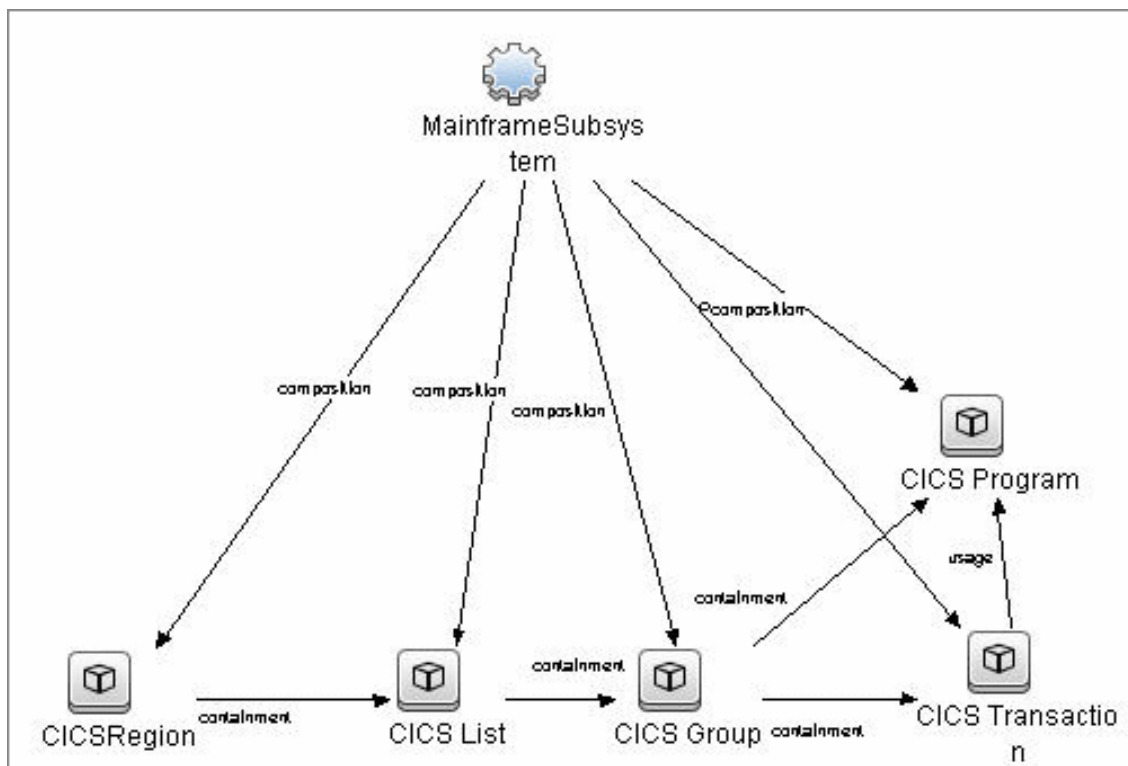
## EView Connection



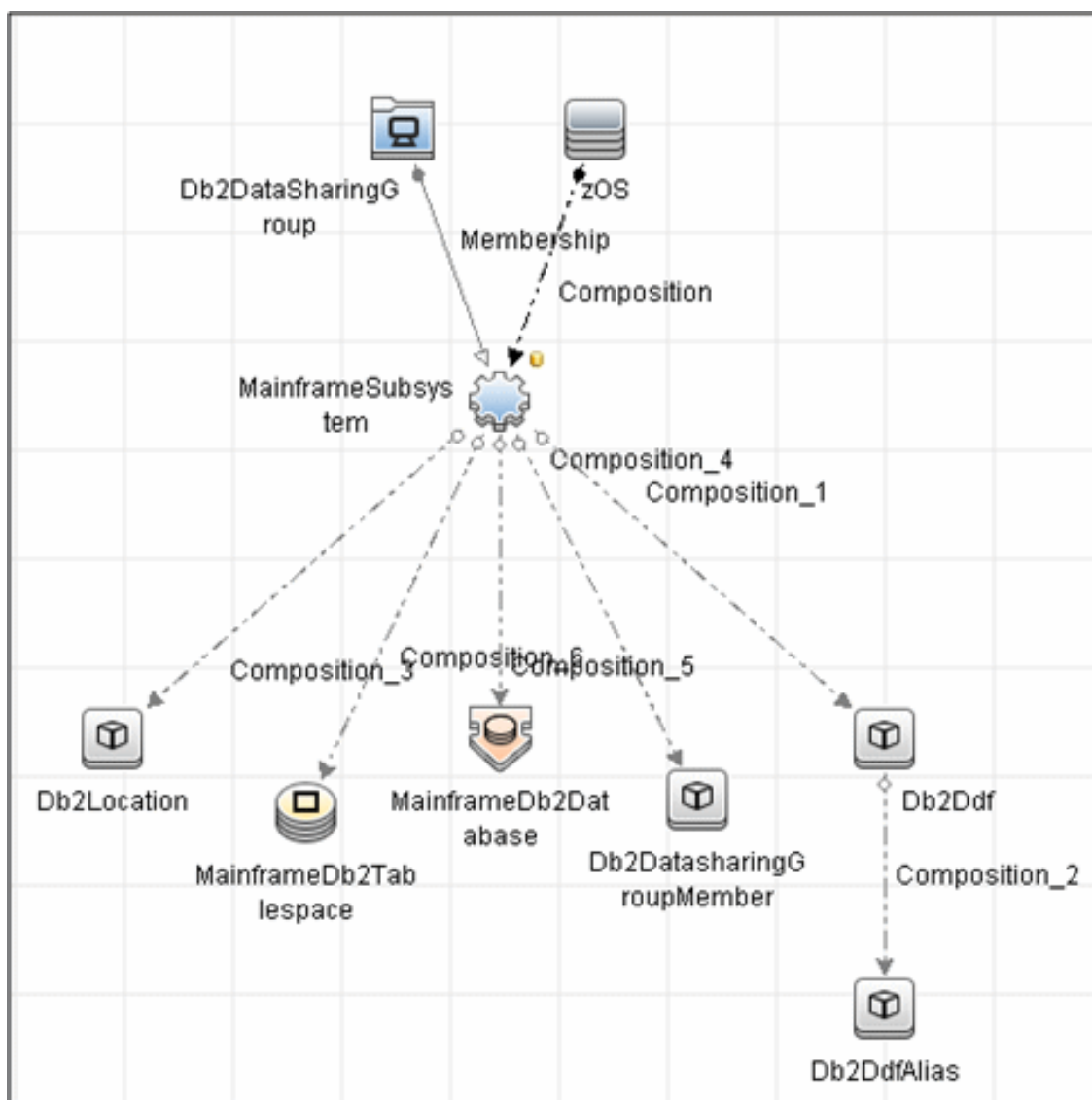
## LPAR Resources by EView



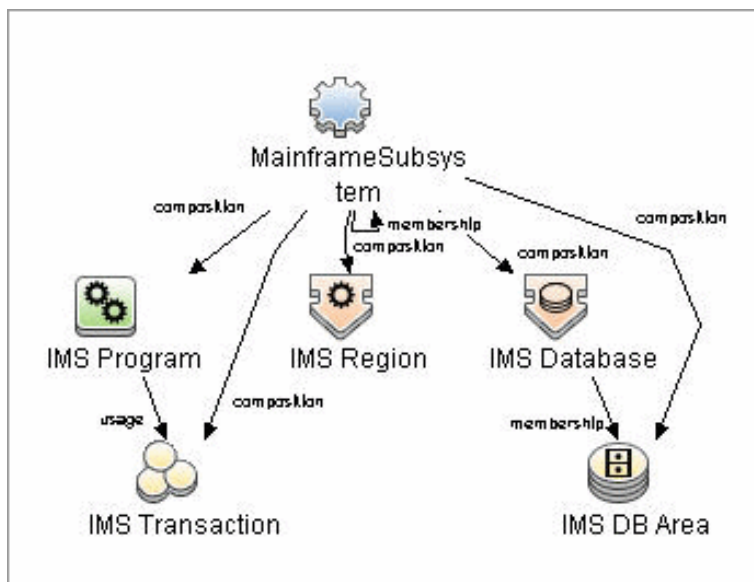
## CICS by EView



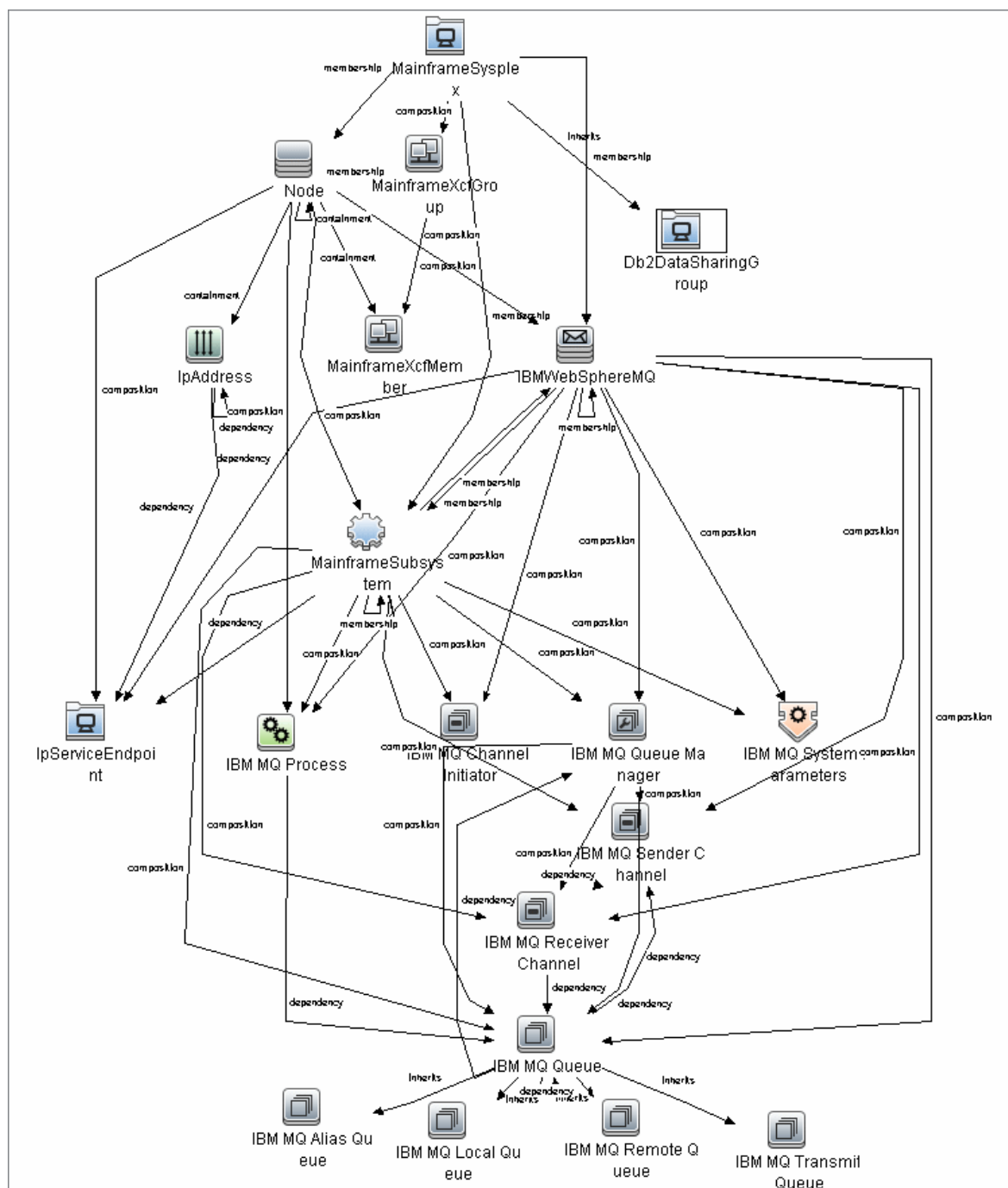
## DB2 by EView



## IMS by EView



## MQ by EView



## EView によるメインフレームの検出方法

次の手順では、Mainframe by EView ディスカバリについて説明します。

### 1. 前提条件

- EView/390z エージェント(バージョン 6.3 以降)が、リソースおよびアプリケーションを検出するすべての LPAR にインストールされていることを確認します。
- EView/390z ディスカバリ・クライアント(バージョン 6.3 以降)が、メインフレーム・インフラストラクチャの検出に使用される Data Flow Probe と同じマシンにインストールされていることを確認します。
- EView/390z ディスカバリ・クライアント内の LPAR が適切に構成されていることを確認します。
- このディスカバリのためのすべてのセキュリティ要件が設定されていることを確認します。

これらの前提条件の詳細については、次の EView/390z Discovery for z/OS のドキュメントを参照してください。

<http://www.eview-tech.com/e390ddisc.php>

### 2. EView Connection ジョブの実行

**注:** このジョブは、その他の Mainframe by EView ディスカバリ・ジョブの実行前に実行する必要があります。

- a. Data Flow Probe マシン上で EView/390z ディスカバリ・クライアントがインストールされている絶対パスを指定することによって、EView Connection ディスカバリ・ジョブの **EViewInstallationFolder** パラメータを構成します。

例:

C:\EviewTechnology\EView390

- b. Data Flow Probe マシンの EView/390z ディスカバリ・クライアント構成にあるすべてのノードに対して構成された EView/390z エージェント・オブジェクトを検出するため、ディスカバリ・ジョブをアクティブにします。

### 3. ディスカバリ・ジョブの実行

メインフレーム・トポロジを検出するため、次のジョブをアクティブにします。

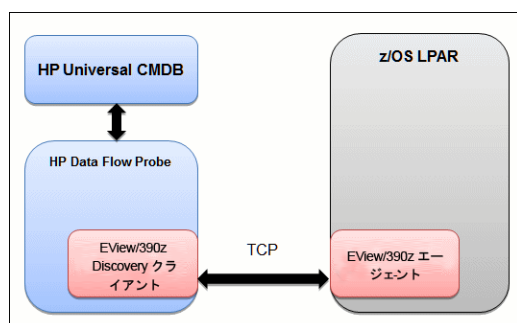
- z/OS LPAR ホストおよびネットワーク・リソースを検出 するため, **LPAR Resources by EView** ジョブをアクティブにします。このジョブの詳細については, 「[LPAR Resources by EView ジョブ](#)」(764ページ)を参照してください。
- CICS サブシステムおよびそのリソースを検出 するため, **CICS by EView** ジョブをアクティブにします。このジョブの詳細については, 「[CICS by EView ジョブ](#)」(766ページ)を参照してください。
- DB2 サブシステムおよびそのリソースを検出 するため, **DB2 by EView** ジョブをアクティブにします。このジョブの詳細については, 「[DB2 by EView ジョブ](#)」(767ページ)を参照してください。
- IMS サブシステムおよびそのリソースを検出 するため, **IMS by EView** ジョブをアクティブにします。このジョブの詳細については, 「[IMS by EView ジョブ](#)」(768ページ)を参照してください。
- MQ サブシステムおよびそのリソースを検出 するため, **MQ by EView** ジョブをアクティブにします。このジョブの詳細については, 「[MQ by EView ジョブ](#)」(769ページ)を参照してください。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## ディスカバリ・メカニズム

Mainframe by EView ディスカバリは, エージェント・ベースのディスカバリ・ソリューションです。z/OS LPAR 上のインフラストラクチャ・リソースとアプリケーションを検出 するには, エージェント・コンポーネントが検出 対象のすべての LPAR にデプロイされている必要があります。

このディスカバリ・ソリューションのアーキテクチャ概要図を次の図に示します。



ディスカバリ・プロセスは次のように機能します。

1. 接続ジョブは次のように機能します。
  - a. **EView Connection** ジョブは、このディスカバリのための CI を検出する最初のジョブです。このジョブは、UCMDB 内に構成されたすべての Probe Gateway CI に対してトリガされます。
  - b. Data Flow Probe で、まず **evview\_connection.py** ディスカバリ・スクリプトは、このディスカバリ・ジョブ内に事前に構成された EView/390z ディスカバリ・クライアントのインストール・パスに、EView/390z ディスカバリ・クライアントがあるかどうかを検索します。次に、EView/390z ディスカバリ・クライアント内に構成された z/OS LPAR ノードを検索します。
  - c. ディスカバリ・ジョブは、EView/390z ディスカバリ・クライアント内に構成されたすべての z/OS LPAR ノードに対し、zOS CI に接続された evview エージェント CI とそのプライマリ IP アドレス用の CI を作成します。
2. リソースおよびアプリケーションのディスカバリ・ジョブは次のように機能します。
  - a. 残りのすべてのジョブが、**evview\_agent** TQL クエリでアクティブになります。このクエリによって、検出されたすべての evview エージェント CI に対してジョブが呼び出されます。
  - b. ディスカバリ・スクリプトは、EView/390z エージェントを使用することによって、z/OS LPAR に対する各種 MVS コマンドを実行し、返された出力を解析し、関連 CI タイプを作成します。

ディスカバリの実行の詳細については、「[EView によるメインフレームの検出方法](#)」(762ページ)を参照してください。

## LPAR Resources by EView ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

トリガ・クエリ名: **evview\_agent**

### ディスカバリ・パラメータ

パラメータ	説明
<b>commandTimeout</b>	EView/390z エージェントに対して発行されたコマンドがタイムアウトするタイムアウト値(秒単位)



パラメータ	説明
<b>maxCommandSize</b>	z/OS LPAR でのコマンド出力用に割り当てられた最大サイズ(バイト単位)
<b>debugMode</b>	True に設定すると、プローブのデバッグ・ログで詳細ログ記録が有効化
<b>discover_CPUs</b>	zOS LPAR CPU CI を検索
<b>discover_Jobs</b>	アドレス領域(ジョブ、開始タスク)を検出するかどうかを示す True か False のフラグ  標準設定 : False
<b>discover_MajorNodes</b>	zOS Major Node CI を検索
<b>discover_PageDatasets</b>	zOS Page Dataset CI を検索
<b>discover_Software</b>	zOS Installed Software CI を検索
<b>discover_Subsystems</b>	zOS Subsystem CI を検索
<b>discover_TCP_UDP</b>	z/OS LPAR TCP ポートおよび接続とUDP ポートを検索
<b>discover_DASD</b>	z/OS DASD ストレージ・デバイスおよびストレージ・グループを検索  標準設定 : False  <b>注:</b> True に設定した場合、EView/390 クライアントのコマンド・タイムアウト・パラメータの値を増やす必要があります。
<b>job_Regex</b>	検出対象のジョブを決定する UNIX 形式の正規表現の値を格納するパラメータ  標準設定 : *  <b>注:</b> 標準設定値に設定された場合、 <b>discover_Jobs</b> が true に設定されていると、すべてのジョブが検出されます。

**注:** このディスカバリのトポロジ・マップを参照するには、[「LPAR Resources by EView」\(758ページ\)](#)を参照してください。

## EView Connection ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ

トリガ・クエリ名 : **probe**

## ディスカバリ・パラメータ

パラメータ	説明
<b>EViewInstallationFolder</b>	Data Flow Probe マシン上の EView/390z ディスカバリ・クライアントのインストール・ルート・ディレクトリ
<b>EViewStartedTask</b>	EView エージェントの開始タスク名 (VP390 など)

注: このディスカバリのトポロジ・マップを参照するには、[「EView Connection」\(757ページ\)](#)を参照してください。

# CICS by EView ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ

トリガ・クエリ名 : **eview\_agent**

## ディスカバリ・パラメータ

パラメータ	説明
<b>commandTimeout</b>	EView/390z エージェントに対して発行されたコマンドがタイムアウトするタイムアウト値(秒単位)を示します。
<b>maxCommandSize</b>	z/OS LPAR でのコマンド出力用に割り当てられた最大サイズ(バイト単位)を示します。
<b>debugMode</b>	True に設定すると、プローブのデバッグ・ログで詳細ログ記録が有効化されます。
<b>discover_CICS_Regions</b>	CICS 領域およびそれらの詳細プロパティを検索します。

パラメータ	説明
<b>discover_CICS_programs</b>	<p>CICS プログラムおよびトランザクションを検出 するかどうかを示す True か False のフラグを示します。</p> <p><b>標準設定</b> : False</p> <p><b>注</b>: True に設定した場合, EView/390 クライアントのコマンド・タイムアウト・パラメータの値を増やす必要があります。</p>
<b>exclude_restricted_programs</b>	<p>'RESTRICTED' というラベルが付いた IBM が提供する要素を検出 するかどうかを表す True か False のフラグを示します。これらの要素は, ベンダ・ソフトウェア・パッケージの標準オペレーティング・コンポーネントです。</p> <p><b>標準設定</b> : True</p>

**注**: このディスカバリのトポロジ・マップを参照するには, 「[CICS by EView](#)」(758ページ)を参照してください。

## DB2 by EView ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

トリガ・クエリ名: **eview\_agent**

### ディスカバリ・パラメータ

パラメータ	説明
<b>commandTimeout</b>	EView/390z エージェントに対して発行されたコマンドがタイムアウトするタイムアウト値(秒単位)
<b>maxCommandSize</b>	z/OS LPAR でのコマンド出力用に割り当てられた最大サイズ(バイト単位)
<b>debugMode</b>	True に設定すると, プローブのデバッグ・ログで詳細ログ記録が有効化
<b>discover_DDF</b>	z/OS DB2 分散データ機能を検索
<b>discover_</b>	z/OS DB2 分散データ共有グループを検索

パラメータ	説明
<b>DataSharingGroups</b>	
<b>discover_Databases</b>	z/OS DB2 データベースを検索
<b>discover_Locations</b>	z/OS DB2 の場所を検索
<b>discover_Tablespaces</b>	z/OS DB2 テーブルスペースを検索

注: このディスカバリのトポロジ・マップを参照するには、[「DB2 by EView」\(759ページ\)](#)を参照してください。

## IMS by EView ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

トリガ・クエリ名 : **eview\_agent**

### ディスカバリ・パラメータ

パラメータ	説明
<b>commandTimeout</b>	EView/390z エージェントに対して発行されたコマンドがタイムアウトするタイムアウト値(秒単位)を示します。
<b>debugMode</b>	True/False フラグを示します。True に設定すると、プローブのデバッグ・ログで詳細ログ記録が有効化されます。
<b>maxCommandSize</b>	z/OS LPAR でのコマンド出力用に割り当てられた最大サイズ(バイト単位)を示します。
<b>DiscoverIMSDB</b>	IMS データベースの検出を試みるかどうかを示す True か False のフラグを示します。 <b>標準設定 : False</b>
<b>discover_ims_programs</b>	IMS プログラムおよびトランザクションを検出するかどうかを示す True か False のフラグを示します。 <b>標準設定 : False</b>

注: このディスカバリのトポロジ・マップを参照するには、[「IMS by EView」\(760ページ\)](#)を参照してください。

## MQ by EView ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

トリガ・クエリ名: **eview\_agent**

### ディスカバリ・パラメータ

パラメータ	説明
<b>commandTimeout</b>	EView/390z エージェントに対して発行されたコマンドがタイムアウトするタイムアウト値(秒単位)を示します。
<b>debugMode</b>	True/False フラグを示します。True に設定すると、プローブのデバッグ・ログで詳細ログ記録が有効化されます。
<b>maxCommandSize</b>	z/OS LPAR でのコマンド出力用に割り当てられた最大サイズ(バイト単位)を示します。
<b>discover_remote_hosts</b>	接続されたリモート・ホスト上でホストおよびキューを検出するかどうかを示す True か False のフラグです。  標準設定: False

注: このディスカバリのトポロジ・マップを参照するには、[「MQ by EView」\(761ページ\)](#)を参照してください。

## トラブルシューティングおよび制限事項

Mainframe by EView ディスカバリのトラブルシューティングは、次の 2 つのおおまかなカテゴリに分類されます。

- UCMDB または DFM メインフレームのディスカバリ・プロセスのトラブルシューティング
  - ディスカバリ・ジョブの適切なトリガ検証, ディスカバリ・ジョブの呼び出し確認, プローブ・ログからのトラブルシューティング情報の検索などを実行します。
  - z/OS LPAR に対してコマンドを手動で呼び出します。これには EView/390z ディスカバリ・クライアントを使用します。
  - EView/390z ディスカバリ・クライアントと EView/390z エージェントとの間の接続を検証します。
  - コマンドが正常に発行できることを確認し, z/OS LPAR から有効な応答が返されることを確認します。

#### • EView/390z エージェントのトラブルシューティング

ディスカバリ・プロセスが CI および関係を正常に検出できなかった場合には, ほぼ必ずディスカバリのトラブルシューティング・プロセスが開始されます。このときに, 問題の原因が UCMDB または DFM のディスカバリ・プロセス(ジョブ, トリガ, アダプタ, スクリプトなど)にあるのか, EView/390z Discovery for z/OS にあるのかを判断するのが重要です。このトラブルシューティング・プロセスで役立つ手順のいくつかを次に挙げます。

- UCMDB または DFM プロセスまたはサービスが, 正常に実行されていることを確認します。
- すべてのメインフレーム・ディスカバリ・パッケージが適切にデプロイされており, ディスカバリ・ジョブが適切に構成されていることを確認します。
- EView/390z ディスカバリ・クライアント(バージョン 6.3 以降)および EView/390z エージェント(バージョン 6.3 以降)がインストールされていることを確認します。以前のバージョンがインストールされている場合, ディスカバリに失敗することがあります。
- EView/390z ディスカバリ・クライアントが Data Flow Probe マシンに適切にインストールされており, このクライアントのサービスが適切にインストールおよび実行されていることを確認します。
- EView/390z ディスカバリ・クライアントで, 検出対象の LPAR が適切に構成されていることを確認します。
- 問題が発生しているディスカバリ・ジョブを実行し, ディスカバリ・ログからジョブの呼び出しおよびコマンドの実行に関係するメッセージを探します。
  - ディスカバリ・ジョブの呼び出しに問題がある, ディスカバリ・スクリプト構文のエラー, または CI 調整エラーと考えられる場合, UCMDB のディスカバリ・プロセスで行うのと同様にそれらをトラブルシューティングします。

- 。 コマンドが EView/390z エージェントに対して発行されないためにディスカバリにエラーが発生したことがログに示されている場合、エラーになったコマンドをプローブ・デバッグ・ログ・ファイルから特定し、EView/390z ディスカバリ・クライアントを使用して関連コマンドの呼び出しを手動で試します。詳細については、EView Technology Inc. のカスタマ・サポートにお問い合わせください。

## 第9部: Middleware > Java EE Application Servers



# 第52章: Apache Tomcat ディスカバリ

## 本章の内容

- 概要 ..... 774
- サポートされるバージョン ..... 775
- トポロジ ..... 776
- Apache Tomcat の検出方法 ..... 777
- Bugzilla, Wordpress, MediaWiki の検出方法 ..... 778
- Apache Tomcat by Shell ジョブ ..... 779
  - アダプタ情報 ..... 779

## 概要

Apache Tomcat を検出するために、DFM は次の構成ファイルを解析します。

- **server.xml** : Apache Tomcat のメインの構成ファイル。Tomcat インストール済み環境のコンポーネント、そのアーキテクチャ、トポロジが記述されています。このファイルには、グローバル・リソースの構成も含まれています。

**server.xml** ファイルには、次に示すスクリプトの断片が含まれています。これは、CI 作成用の情報を取得するために **Apache Tomcat by Shell** ジョブで使用される部分です。

```
<Server port="8505" shutdown="SHUTDOWN">
  <GlobalNamingResources>
    <Resource name="jdbc/GlobalDS"
      type="javax.sql.DataSource"
      driverClassName="com.inet.ora.OraDriver"
      url="jdbc:inetora:labm3mam13:1521:UCMDB"
      maxActive="20" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8580" protocol="HTTP/1.1"/>
    <Connector port="8509" protocol="AJP/1.3" />
    <Engine name="Catalina">
      <Host name="localhost" appBase="webapps">
        <Cluster">
          <Membership mcastAddr="228.0.0.4" mcastPort="45564"/>
        </Cluster>
      </Host>
      <Host name="grabinovic01" appBase="genadiwebapps">
        <Membership mcastAddr="228.0.0.4" mcastPort="45564"/>
      </Cluster>
    </Host>
  </Engine>
</Service>
</Server>
```

- **context.xml** : このファイルは、アプリケーション・コンテキスト構成を定義します。インストールされている各アプリケーションには、一意の URL プレフィックスがあります。このファイルには、ファイルの場所に応じて、さまざまな範囲のリソース構成が含まれています。
- **web.xml** : このファイルでは、アプリケーションの表示名や HTTP 要求の処理に使用するサーブレットなどのアプリケーション構成を定義します。現在、DFM は、このファイルを使用してアプリケーションの表示名を取得します。

## サポートされるバージョン

このディスカバリでは、次のバージョンの Apache Tomcat がサポートされています。

- 5
- 5.5
- 6.0

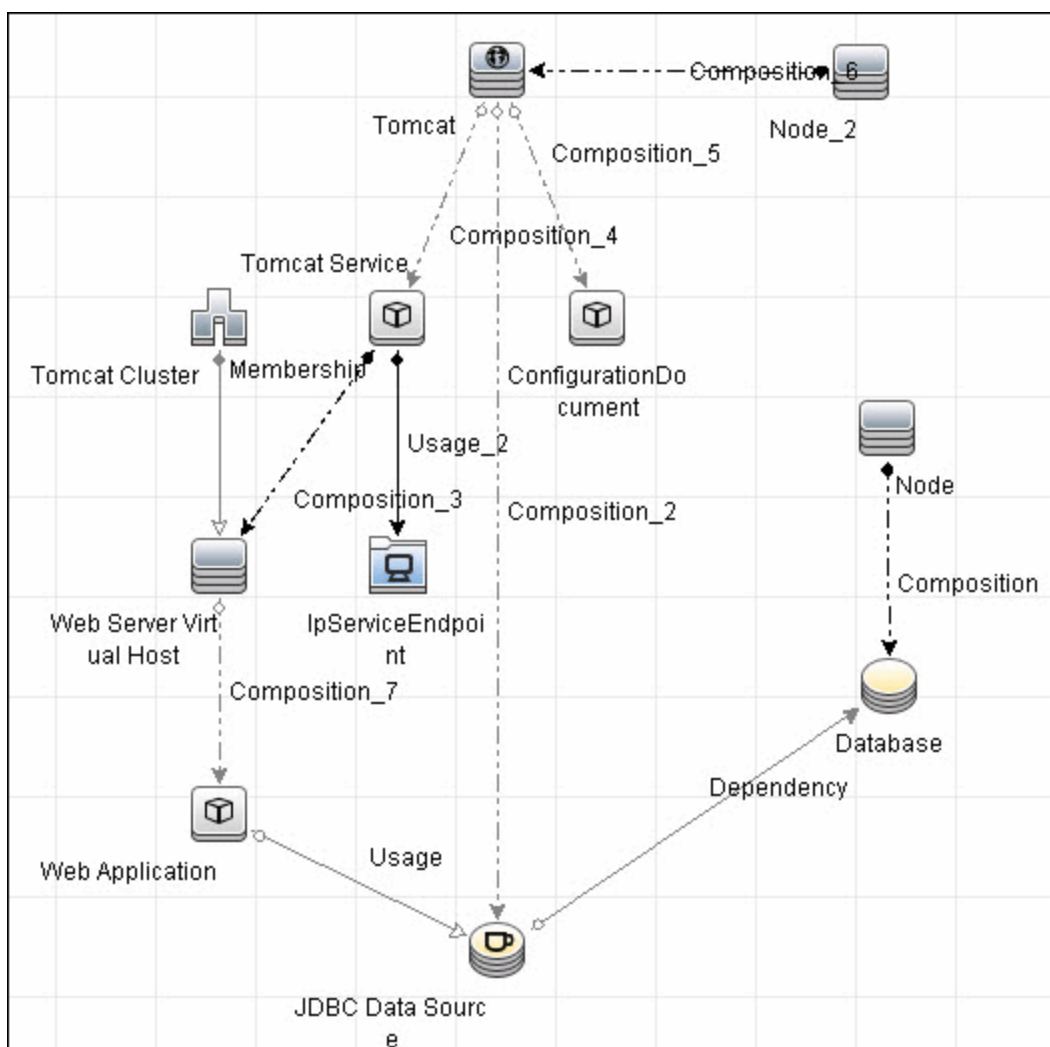
DFM は、次のオペレーティング・システム上で稼働している Tomcat を検出します。

- Windows
- UNIX
- Linux

## トポロジ

次の図は、Apache Tomcat ディスカバリのトポロジを示しています。

注: 検出された CIT のリストについては、「[Apache Tomcat by Shell ジョブ](#)」(779ページ)を参照してください。



## Apache Tomcat の検出方法

このタスクでは、Apache Tomcat アプリケーションの検出方法を次の手順で説明します。

### 1. 前提条件 - ネットワーク, プロトコル資格情報の設定

このディスカバリでは、次のプロトコルを使用します。

- NTCMD プロトコル
- SSH プロトコル
- Telnet プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行し、Tomcat が稼働している範囲の IP を検出します。
- b. **Host Connection by Shell** ジョブを実行し、シェル・エージェントを検出します。
- c. **Host Applications by Shell** ジョブを実行し、Apache Tomcat がシステムで稼働していることを確認して、Tomcat 固有のプロセスを検出します。これらのプロセスが検出された場合は、Tomcat CI が作成されます。

このジョブでは、**java.exe** (または **java**) プロセス名が検索され、次にコマンド・ラインで部分文字列 **-Dcatalina.home=** または **-Dcatalina.base=** が検索されます。この部分文字列には、Tomcat ホーム・ディレクトリのパスが含まれています。この部分文字列が見つからない場合は、**tomcat** で始まるプロセス名が検索され、その名前からホーム・ディレクトリのパスが取得されます。

次に、Tomcat 構成ファイルの絶対パスが検索され、このパスが属性 (**webserver\_configfile**) として Tomcat CI に追加されます。

- d. **Apache Tomcat by Shell** ジョブを実行します。このジョブでは、Tomcat CI 属性を使用して、**Host Applications by Shell** ジョブで検出された構成ファイルが検索されます。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Bugzilla, Wordpress, MediaWiki の検出方法

次の Web ベース・アプリケーションは、Apache および IIS ディスカバリ・ジョブの一部として検出されます。サポートされるバージョンは、次のとおりです。

アプリケーション	サポートされるバージョン
Bugzilla	3.x
Helpzilla	0.x
MediaWiki	1.15.x
Wordpress	2.5.x

ディスカバリをアクティブ化するには、次の操作を実行します。

1. **Host Connection by Shell** ジョブを実行して Shell CIT を作成します。
2. **Host Resources** および **Host Applications** ジョブのいずれかを実行して、ホストで実行中のプロセスに関する情報を収集します。
3. **Web Server by Shell** ジョブを実行して、Apache に関する情報と、Apache サーバにデプロイ済みの使用可能な Web アプリケーションに関する情報を取得します。

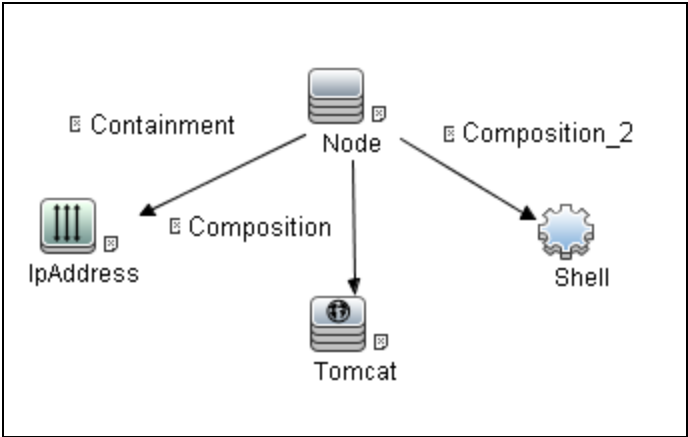
### Web Application CIT :

- **ID** : webapplication
- **親 CIT** : application
- **既存の属性の使用** : name
- **新規属性** : type (ブログ・エンジン, Wiki などのアプリケーションのタイプ)

## Apache Tomcat by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
Shell	なし
Node	なし
Tomcat	なし

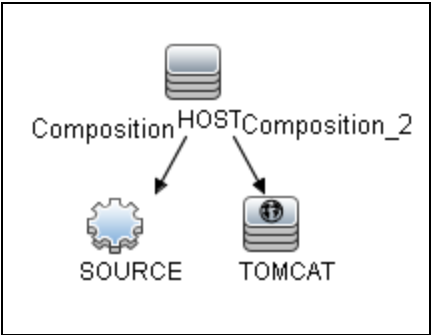
### アダプタ情報

このジョブでは、**ApacheTomcat\_Topology** アダプタを使用します。

### 入力 CIT

Shell

入力 TQL クエリ



ノード名	条件
SOURCE	なし
TOMCAT	なし
HOST	なし

トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
configfile	\${TOMCAT.name}
credentialsId	\${SOURCE.credentials_id}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}

使用スクリプト

- tomcat\_by\_shell.py
- jdbc.py
- jdbc\_url\_parser.py
- db\_builder.py



- db\_platform.py
- db.py
- file\_mon\_utils.py
- file\_ver\_lib.py
- iteratorutils.py
- entity.py

## 検出された CIT

- Apache Tomcat
- Apache Tomcat Cluster
- Apache Tomcat Service
- Composition
- ConfigurationDocument
- Containment
- Database
- Dependency
- IPAddress
- IpServiceEndpoint
- JdbcDataSource
- Membership
- Node
- Usage
- Web Application

- Web Server Virtual Host

**グローバル構成ファイル**

globalSettings.xml

**アダプタ・パラメータ**

パラメータ	説明
configFiles	Tomcat <b>server.xml</b> へのフルパス(カンマ区 切り)。

注: トポロジを表示するには, [「トポロジ」\(776ページ\)](#)を参照してください。

## 第53章: GlassFish ディスカバリ

### 本章の内容

概要 .....	784
サポートされるバージョン .....	784
GlassFish Topology by Shell の検出方法 .....	784
JEE Glassfish by Shell ジョブ .....	785
Glassfish_By_Shell アダプタ .....	786
トラブルシューティングおよび制限事項 .....	789

## 概要

GlassFish は、Sun Java System Application Server Platform Edition 9 (Sun Microsystems) のソース・コードと TopLink (Oracle) のソース・コードをベースにしたオープン・ソースのアプリケーション・サーバです。GlassFish では、JDBC、RMI、電子メール、JMS、Web サービス、XML などの Java Platform Enterprise Edition API 仕様がすべてサポートされており、これらの仕様を連携する方法が詳細に説明されています。

GlassFish のディスカバリ・プロセスでは、J2EE アプリケーション、JDBC リソース、JMS リソースを含むトポロジ全体を検出できます。

## サポートされるバージョン

バージョン	サポートされている	J2EE のバージョン	JVM のバージョン
GlassFish 2.1	はい	J2EE 1.5	JVM 1.5
GlassFish 3.1	はい	J2EE 1.6	JVM 1.6

## GlassFish Topology by Shell の検出方法

このタスクでは、シェル・プロトコルを使用して GlassFish を検出する方法について説明します。GlassFish のディスカバリ・プロセスでは、J2EE アプリケーション、JDBC リソース、JMS リソースを含む GlassFish トポロジ全体を検出できます。DFM は最初にシェル・プロトコルまたはエンドポイント (TCP ポート) に基づいてアプリケーション・サーバを検索し、次にシェルにより GlassFish J2EE 環境およびコンポーネントを検出します。

### 1. 前提条件 - プロトコル資格情報の設定

ディスカバリは、シェル・プロトコルを使用して実行されます。次のいずれかの資格情報を定義します。

- SSH
- Telnet
- NTCMD

### 2. ディスカバリの実行

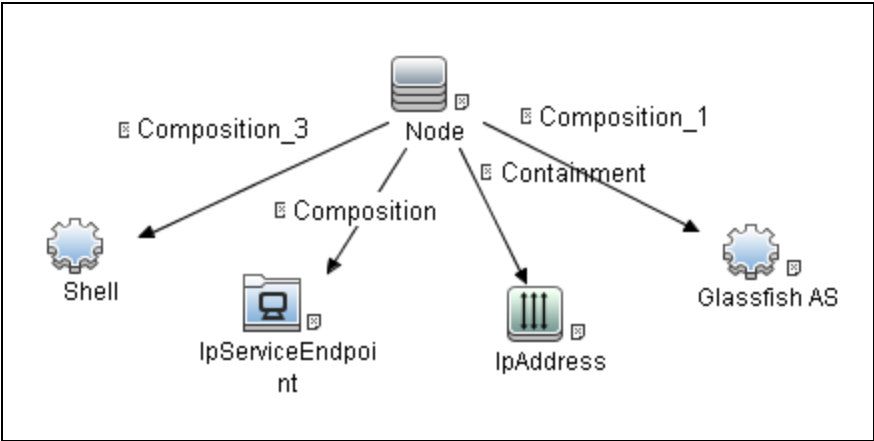
- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by Shell** ジョブを実行して、ターゲット・ホストとそれに対するシェル接続を検出します。
- c. 次の 2 つのジョブのいずれかを実行します。
  - **Host Applications by Shell** : 実行中のプロセスを含め、ターゲット・ホストのアプリケーションを検出するために実行します。
  - **JEE TCP Ports** : サービス・エンドポイントの情報を検出するために実行します。

3. **JEE Glassfish by Shell** ジョブを実行します。

## JEE Glassfish by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



ノード名	条件
Glassfish AS	なし
IpAddress	NOT IP Probe Name null
IpServiceEndPoint	IpServiceName 等しい glassfish OR ServiceNames 含む glassfish

ノード名	条件
Node	なし
Shell	NOT Reference to the credentials dictionary entry null

## アダプタ

このジョブでは、**Glassfish\_By\_Shell** アダプタを使用します。

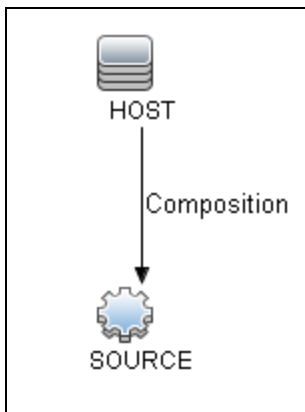
## Glassfish\_By\_Shell アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Shell

### 入力クエリ



### トリガ CI データ

名前	値
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}

名前	値
ip_address	\${SOURCE.application_ip}
protocol	\${SOURCE.root_class}

## 使用スクリプト

- connection.py
- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- file\_ver\_lib.py
- glassfish.py
- glassfish\_by\_shell.py
- glassfish\_discoverer.py
- iteratortools.py
- jdbc.py
- jdbc\_url\_parser.py
- jdbcutils.py
- jee.py
- jee\_connection.py
- jee\_discoverer.py
- jms.py
- jmx.py

- process.py
- process\_discoverer.py
- protocol.py

## 検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Database
- Dependency
- Deployed
- Glassfish AS
- IpAddress
- IpServiceEndPoint
- J2eeCluster
- JdbcDataSource
- J2EE Domain
- J2EE Managed Object
- Membership
- Node
- Usage
- Web Service



## グローバル構成ファイル

globalSettings.xml

## パラメータ

- **reportAdminApps** : 値が 'true' の場合は、管理アプリケーションのレポートが有効になり、値が 'false' の場合は、無効になります

## トラブルシューティングおよび制限事項

本項では GlassFish ディスカバリのトラブルシューティングと制限事項について説明します。

- DFM が J2EE アプリケーションを検出できるのは、その .ear ファイルがフォルダに解凍されている場合に限られます。
- HRA ディスカバリの実行中、GlassFish プロセスのコマンド・ラインが大きすぎて、プローブ・データベースの該当フィールドに収まらないことがあります。そのような場合は、次の操作を実行します。
  - プローブを停止する
  - **%DataFlowProbeHome%/tools/dbscripts/create\_netlinks\_db\_tables.sql** を開く
  - プロセス・テーブルの **cmdline** のサイズを 4000 から 8000 に変更する(または、必要に応じて 8000 より大きくする)
  - アプリケーション・テーブルの **cmdline** のサイズを 512 から 8000 に変更する(または、必要に応じて 8000 より大きくする)
  - ファイルを保存する
  - **clearProbeData.bat** スクリプトを実行する
  - プローブを起動する

## 第54章: JBoss ディスカバリ

### 本章の内容

概要 .....	791
サポートされるバージョン .....	791
JEE JBoss by JMX の検出方法 .....	791
JEE JBoss by Shell の検出方法 .....	796
JEE TCP Ports ジョブ .....	797
アダプタ情報 .....	799
JEE JBoss Connections by JMX ジョブ .....	802
アダプタ情報 .....	803
JEE JBoss by JMX ジョブ .....	806
JEE JBoss by Shell ジョブ .....	809
アダプタ情報 .....	810
トラブルシューティングおよび制限事項 .....	813

## 概要

JBoss アプリケーション・サーバ(JBoss AS)は、JBoss (現在は Red Hat の一部門)が開発したフリーソフトでありオープンソースの Java EE ベース・アプリケーション・サーバです。

このクラスのソフトウェアは、Java で稼働するサーバを実装しているだけではなく、Java の Java EE 部分も実際に実装していることが重要な特徴です。JBoss アプリケーション・サーバは、Java ベースであるため、クロスプラットフォームで作動します。つまり、Java をサポートしていれば、どのオペレーティング・システムでも使用できます。

JBoss のディスカバリ・プロセスでは、J2EE アプリケーション、JDBC リソース、JMS リソースを含む JBoss トポロジ全体を検出できます。DFM は最初に JMX プロトコルに基づいて JBoss サーバを検索し、次に JBoss J2EE 環境およびコンポーネントを検出します。

## サポートされるバージョン

- JBoss by JMX ディスカバリ: JBoss バージョン 4.x, 5.x, 6.x, 7.x
- JBoss by Shell ディスカバリ: JBoss バージョン 4.x, 5.x, 6.x, 7.x

## JEE JBoss by JMX の検出方法

本項の内容

- [「前提条件 - プロトコル資格情報の設定」\(791ページ\)](#)
- [「前提条件 - ドライバの設定」\(791ページ\)](#)
- [「ディスカバリの実行」\(795ページ\)](#)

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、JBoss プロトコルを使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - ドライバの設定

標準設定では、プローブのインストールに標準設定の JBoss ドライバが含まれています。必要な \*.jar ファイルの詳細については、『HP Universal CMDB Data Flow Management Guide』の「JBoss」を参照してください。プローブのインストールには、バージョン 3.x と 4.x の JBoss ドライバが含まれていますが、必要に応じてご自身のドライバを使用することもできます。

.jar ファイルを更新するには、次の操作を実行します。

- a. 次の場所にある適切なバージョンのフォルダにドライバをコピーします。

```
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager  
\discoveryResources\j2ee\jboss\<version_folder>
```

**注:** 商用版の JBoss 5.x クライアント API (EAP) には、エラーがあります。認証が有効になっている EAP 5.x を検出するには、非商用版の 5.x からクライアント・ドライバを取得する必要があります。

- b. DFM ジョブを実行する前にプローブを再起動します。

**例 :**

JBoss 5.x バージョンを検出するには、次のドライバ・フォルダ

```
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager  
\discoveryResources\j2ee\jboss\5.x
```

を **jbossall-client.jar** ファイルで更新し、その中で宣言されているすべての依存関係を含める必要があります。

必要な jar は、<JBOSS\_5\_BASE\_DIR>/client/ フォルダにあります。

**jbossall-client.jar** ファイルには、JBoss クライアント・アプリケーションで使用される各種のクライアント .jar ファイルへのクラスパス参照が含まれています。次のリストの各 .jar ファイルは、**jbossall-client.jar** と同じディレクトリで使用できる必要があります。そうしないと、クラスローダで検出されません。

クラスパスには、次のファイルが含まれています。

- commons-logging.jar
- concurrent.jar
- ejb3-persistence.jar

- hibernate-annotations.jar
- jboss-aop-client.jar
- jboss-appclient.jar
- jboss-aspect-jdk50-client.jar
- jboss-client.jar
- jboss-common-core.jar
- jboss-deployers-client-spi.jar
- jboss-deployers-client.jar
- jboss-deployers-core-spi.jar
- jboss-deployers-core.jar
- jboss-deployment.jar
- jboss-ejb3-common-client.jar
- jboss-ejb3-core-client.jar
- jboss-ejb3-ext-api.jar
- jboss-ejb3-proxy-client.jar
- jboss-ejb3-proxy-clustered-client.jar
- jboss-ejb3-security-client.jar
- jboss-ha-client.jar
- jboss-ha-legacy-client.jar
- jboss-iiop-client.jar
- jboss-integration.jar
- jboss-j2se.jar

- jboss-javaee.jar
- jboss-jsr77-client.jar
- jboss-logging-jdk.jar
- jboss-logging-log4j.jar
- jboss-logging-spi.jar
- jboss-main-client.jar
- jboss-mdr.jar
- jboss-messaging-client.jar
- jboss-remoting.jar
- jboss-security-spi.jar
- jboss-serialization.jar
- jboss-srp-client.jar
- jboss-system-client.jar
- jboss-system-jmx-client.jar
- jbosscx-client.jar
- jbosssx-as-client.jar
- jbosssx-client.jar
- jmx-client.jar
- jmx-invoker-adaptor-client.jar
- jnp-client.jar
- slf4j-api.jar
- slf4j-jboss-logging.jar

- xmlsec.jar

### 3. ディスカバリの実行

次のジョブを次の順序で実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

- **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- **JEE TCP Ports** ジョブを実行して、サービス・エンドポイントの情報を検出します。ジョブの詳細については、「[JEE TCP Ports ジョブ](#)」(797ページ)を参照してください。
- **JEE JBoss Connections by JMX** ジョブを実行し、アプリケーション・サーバに対してシャロー・ディスカバリを実行します。ジョブの詳細については、「[JEE JBoss Connections by JMX ジョブ](#)」(802ページ)を参照してください。
- **JEE JBoss by JMX** ジョブを実行し、JBoss アプリケーション・サーバトポロジに対してディープ・ディスカバリを実行します。ジョブの詳細については、「[JEE JBoss by JMX ジョブ](#)」(806ページ)を参照してください。

## JEE JBoss by Shell の検出方法

注: この機能は、コンテンツ・パック 2.00 以降の一部として使用できます。

JBoss のディープ・ディスカバリを実行するときに、各サーバの JMX 資格情報を入力する必要がなく、追加のライブラリ(\*.jar ファイル)を定義する必要もありません。その代わりとして、通常のシェル資格情報を使用します。

ディープ・ディスカバリでは、J2EE アプリケーション・システムのトポロジを検出できます。つまり、アプリケーション自体だけではなく、アプリケーションのコンポーネントも検出できます。

本項の内容

- [「前提条件 - プロトコル資格情報の設定」\(796ページ\)](#)
- [「ディスカバリの実行」\(796ページ\)](#)

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、シェル・プロトコルを使用します。次のいずれかのプロトコルの資格情報を定義します。

- NTCMD プロトコル
- SSH プロトコル
- Telnet プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

ユーザには、ルート権限は必要ありません。ただし、リモート・マシンへの接続と、関連するコマンド (`dir` や `type` など)の実行が許可される適切な資格情報が必要です。

### 2. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

- Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- Host Connection by Shell** ジョブを実行して、ターゲット・ホストおよびそのホストへのシェル接



続を検出します。

- c. 次の 2 つのジョブのいずれかを実行します。
  - **Host Applications by Shell**。実行中のプロセスを含む、ターゲット・ホストのアプリケーションを検出する場合に実行します。
  - **JEE TCP Ports**。サービス・エンドポイントの情報を検出する場合に実行します。ジョブの詳細については、[「JEE TCP Ports ジョブ」\(797ページ\)](#)を参照してください。
- d. **JEE JBoss by Shell** ジョブを実行します。ジョブの詳細については、[「JEE JBoss by Shell ジョブ」\(809ページ\)](#)を参照してください。

## JEE TCP Ports ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、**TCP Ports Discovery** アダプタを使用します。

### トリガ・クエリ



### ノード条件

ノード名	条件
IpAddress	NOT IP Probe Name null

## ジョブ・パラメータ

名前	標準設定値	説明
<b>checkIfPlsReachable</b>	true	このフラグは、ホストのポートが利用可能かどうかの確認をジョブが開始する前に、検出された IP に到達可能であることを確認する必要があるかどうかを示します。
<b>checkOnlyKnownPorts</b>	true	このフラグは、既知のポートのみをこのジョブで検出するかどうかを示します。このフラグによって、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータがキャンセルされることはありません。このフラグの <b>false</b> への設定は、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータ内の実際のポート範囲と合わせて使用する場合にのみ適用されます。
<b>connectTimeOut</b>	5000	IP およびポートに接続するときのタイムアウトをミリ秒単位で示します。
<b>nmapPath</b>		nmap 実行可能ファイルのフルパスを示します (例: <b>C:\Program Files (x86)\Nmap\nmap.exe</b> )。
<b>pingTimeOut</b>	2000	ICMP ping のタイムアウトをミリ秒単位で示します。
<b>ports</b>	<p><b>JEE TCP Ports ジョブの場合:</b></p> <ul style="list-style-type: none"> <li>weblogic, weblogicSSL, websphere_jmx, rmi</li> </ul> <p><b>Databases TCP Ports ジョブの場合:</b></p> <ul style="list-style-type: none"> <li>oracle, db2, sybase, sql, mysql</li> </ul> <p><b>SAP TCP Ports ジョブの場合:</b></p> <ul style="list-style-type: none"> <li>sap, sap_jmx, sap_http, sap_https</li> </ul>	このパラメータには、ディスカバリの実行に使用される TCP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の TCP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、 <b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。

名前	標準設定値	説明
	<b>SAP TCP Ports</b> ジョブの場合: 標準設定値なし	
<b>scanUDP</b>	false	このフラグは、UDP ポートをスキャンするかどうかを示します。  <b>注:</b> UDP スキャンは、 <b>useNMap=true</b> に設定されている場合にのみサポートされます(下記を参照)。
<b>UDPports</b>		このパラメータには、デイスカバリの実行に使用される UDP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、デイスカバリはすべての既知の UDP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、 <b>checkOnlyKnownPorts=true</b> が指定されていれば、デイスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。
<b>useNMap</b>	<b>Databases TCP Ports</b> ジョブおよび <b>JEE TCP Ports</b> ジョブの場合: false  <b>SAP TCP Ports</b> ジョブおよび <b>TCP Ports</b> ジョブの場合: true	このフラグは、ポートのスキャン中に nmap を使用するかどうかを示します。  <b>注:</b> <b>nmapPath</b> にパスが指定されないと(上記を参照)、システム・パスからの nmap が使用されます。

**注:** **ports** パラメータまたは **UDPports** パラメータ内でポート名が割り当てられており、**portNumberToPortName.xml** 構成ファイル内で検出可能 (**isDiscovered=1**) としてマークされているポートのみが検出対象になります。

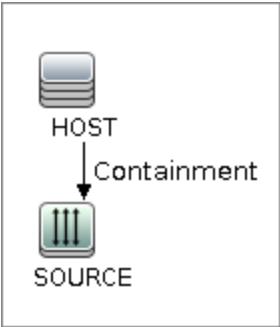
## アダプタ情報

このアダプタでは、TCP ポートを検出します。

入力 CIT

IpAddress

入力クエリ



トリガ CI データ

名前	値
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}

使用スクリプト

- TcpPortScanner.py
- nmap.py

グローバル構成ファイル

portNumberToPortName.xml

検出された CIT

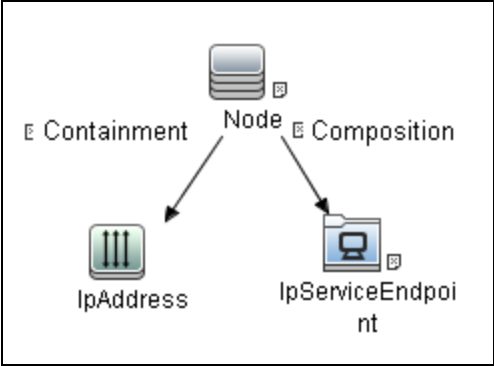
- Composition
- Containment
- IpAddress

- IpServiceEndpoint
- Node

## JEE JBoss Connections by JMX ジョブ

本 項 ではこのジョブに関する詳 細を説 明します。

### トリガ・クエリ



ノード名	条件
Node	なし
IpServiceEndPoint	IpServiceName 等しい rmi OR IpServiceName 等しい jboss-port OR ServiceNames 含む rmi OR ServiceNames 含む jboss-port
IpAddress	NOT IP Probe Name null

### パラメータ

名前	値	説明
disableUDPDDiscovery	false	JMX ポートの UDP ディスカバリを無効にするかどうかを示します。
remoteJVMArgs	-Xms64m -Xmx256m -XX:PermSize=256m -XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータを示します。
runInSeparateProcess	true	パターンを個別のスレッドで実行するかどうかを示します。

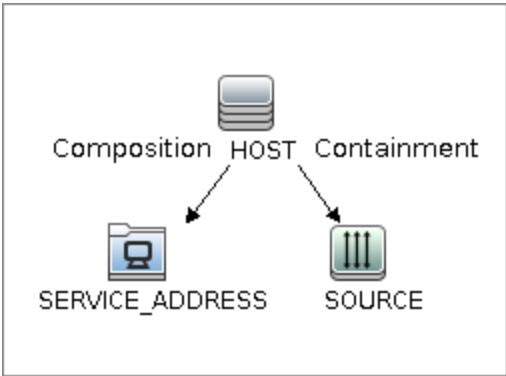
## アダプタ情報

このジョブでは, **JMX\_J2EE\_JBoss\_Connection** アダプタを使用します。このアダプタは, JMX プロトコルに基づいて JBoss サーバ・インスタンスを検出します。

### 入力 CIT

IpAddress

### 入力クエリ



### トリガ CI データ

名前	値
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}
ports	\${SERVICE_ADDRESS.network_port_number:NA}

### 使用スクリプト

- connection.py
- db.py
- db\_builder.py
- db\_platform.py

- entity.py
- iteratortools.py
- jboss.py
- jboss\_discoverer.py
- jdbc.py
- jdbc\_url\_parser.py
- jee.py
- jee\_connection.py
- jee\_discoverer.py
- jmx.py
- JMX\_J2EE\_JBoss\_Connection.py
- protocol.py

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndPoint
- J2EE Domain
- JBoss AS
- JEE Node
- Membership
- Node

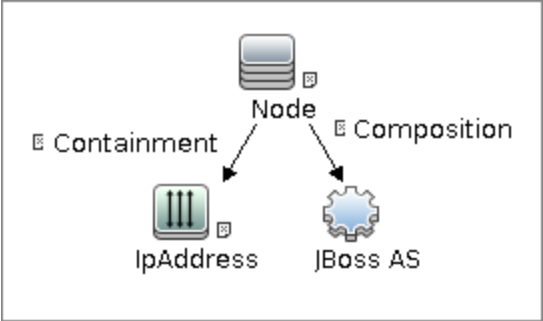


- Usage

# JEE JBoss by JMX ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ



- ノード条件

ノード名	条件
Node	なし
JBoss AS	NOT Reference to the credentials dictionary entry null
IpAddress	NOT IP Probe Name null

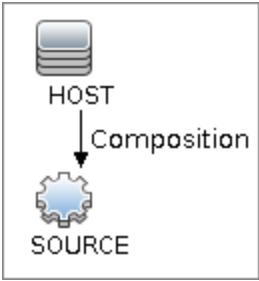
## ジョブ・パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

## アダプタ - JMX\_J2EE\_JBoss

このアダプタは、JMX プロトコルに基づいて JBoss サーバ・インスタンスを検出します。

- 入力 CIT : JBoss AS
- 入力クエリ



• トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip:}
port	\${SOURCE.application_port:}
servername	\${SOURCE.name}
userName	\${SOURCE.application_username:}
version	\${SOURCE.application_version:}

• 使用スクリプト

- connection.py
- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- iteratortools.py
- jboss.py
- jboss\_discoverer.py
- jdbc.py

- jdbc\_url\_parser.py
- jee.py
- jee\_connection.py
- jee\_discoverer.py
- jms.py
- jmx.py
- JMX\_J2EE\_JBoss.py
- protocol.py
- グローバル構成ファイル : globalSettings.xml
- パラメータ :

名前	値	説明
<b>discoverAppResources</b>	true	true に設定した場合、モジュール、EJB、サーブレットが検出される。
<b>discoverJMSResources</b>	true	true に設定した場合、JMS プロバイダと JMS サーバが検出される。
<b>remoteJVMArgs</b>	-Xms64m -Xmx256m - XX:PermSize=256m - XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータ。
<b>runInSeparateProcess</b>	true	パターンを個別のスレッドで実行するかどうかを設定する。

## 検出された CIT

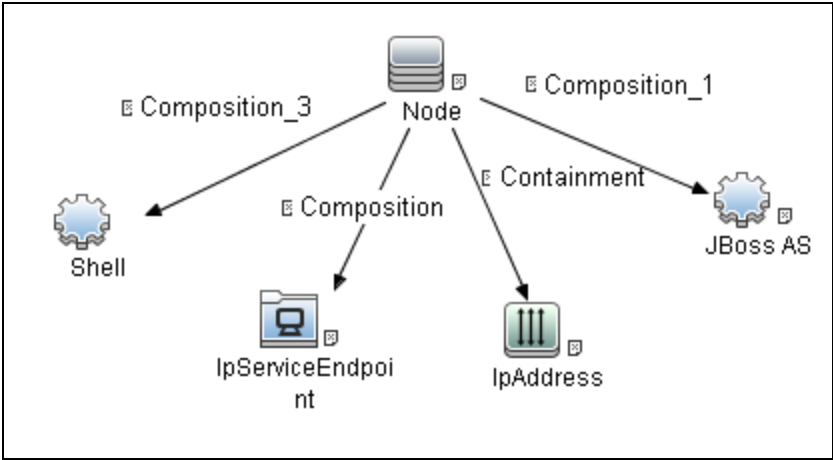
- Composition
- ConfigurationDocument

- Containment
- Database
- Dependency
- Deployed
- IpAddress
- IpServiceEndpoint
- J2eeCluster
- J2EE Domain
- J2EE Managed Object
- JBoss AS
- JdbcDataSource
- JEE Node
- Membership
- Node
- Usage
- Web Service

## JEE JBoss by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

トリガ・クエリ



ノード名	条件
Node	なし
Shell	NOT Reference to the credentials dictionary entry null
JBoss AS	なし
IpAddress	NOT IP Probe Name null
IpServiceEndPoint	IpServiceName 等しい rmi OR IpServiceName 等しい jboss-port OR ServiceNames 含む rmi OR ServiceNames 含む jboss-port

パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

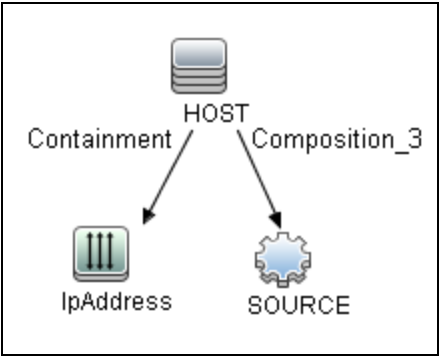
アダプタ情報

このジョブでは、JBoss\_By\_Shell アダプタを使用します。

入力 CIT

Shell

入力クエリ



トリガ CI データ

名前	値
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}
ip_address_list	\${IpAddress.name}
Protocol	\${SOURCE.root_class}

使用スクリプト

- connection.py
- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- file\_ver\_lib.py
- iteratortools.py

- jboss.py
- jboss\_by\_shell.py
- jboss\_discoverer.py
- jdbc.py
- jdbc\_url\_parser.py
- jee.py
- jee\_connection.py
- jee\_discoverer.py
- jms.py
- jmx.py
- process.py
- process\_discoverer.py
- protocol.py

## グローバル構成ファイル

- globalSettings.xml

## パラメータ

なし

## 検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Database



- Dependency
- Deployed
- IpAddress
- IpServiceEndpoint
- J2eeCluster
- J2EE Domain
- J2EE Managed Object
- JBoss AS
- JdbcDataSource
- JEE Node
- Membership
- Node
- Usage
- Web Service

## トラブルシューティングおよび制限事項

本項では JBoss ディスカバリのトラブルシューティングと制限事項について説明します。

- **制限事項** : DFM が J2EE アプリケーションを検出できるのは、その .ear ファイルがフォルダに解凍されている場合に限られます。
- **制限事項** : JBoss 7.x を使用している場合、このディスカバリでは、ローカル・ホスト・コントローラ構成のみがサポートされます。このような管理対象 JBoss サーバの JMX MBean には、リモート・ドメイン・コントローラに関する情報がないためです。

## 第55章: WebLogic ディスカバリ

### 本章の内容

概要 .....	815
サポートされるバージョン .....	815
WebLogic Topology by JMX の検出方法 .....	815
WebLogic Topology by Shell の検出方法 .....	817
JEE TCP Ports ジョブ .....	818
アダプタ情報 .....	820
JEE Weblogic Connections by JMX ジョブ .....	822
JEE Weblogic by JMX ジョブ .....	825
JEE Weblogic by Shell ジョブ .....	830
アダプタ情報 .....	830
トラブルシューティングおよび制限事項 .....	833

## 概要

WebLogic ディスカバリでは、J2EE アプリケーション、JDBC リソース、JMS リソースを含むトポロジ全体を検出できます。

## サポートされるバージョン

サポートされるバージョンは、次のとおりです。

WebLogic 9.x, 10.x, 11g, 11gR1 PS1, 11gR1 PS2, 11gR1 PS3, 12c。

## WebLogic Topology by JMX の検出方法

このタスクでは WebLogic の検出方法について説明します。WebLogic ディスカバリ・プロセスでは、J2EE アプリケーション、JDBC、および JMS リソースを含む WebLogic トポロジー式を検出できます。

DFM は JMX プロトコルに基づいて最初に WebLogic サーバを検出し、次に WebLogic J2EE 環境とコンポーネントを検出します。

本項の内容

- [「前提条件 - プロトコル資格情報の設定」\(815ページ\)](#)
- [「前提条件 - ドライバの設定」\(815ページ\)](#)
- [「ディスカバリの実行」\(817ページ\)](#)

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリは、Weblogic プロトコルからの資格情報を使用した JMX プロトコルに基づきます。Weblogic プロトコル資格情報を定義する必要があります。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - ドライバの設定

WebLogic を検出するために必要なドライバを設定します。標準設定の WebLogic ドライバは含まれていないので、プローブにコピーする必要があります。

- a. SSL で WebLogic を検出するには、次のドライバを取得します。

ドライバ	説明
<b>wlcipher.jar</b>	WebLogic を SSL で実行している場合。  <b>注:</b> サポートされているすべての WebLogic バージョン
<b>クライアント・トラスト・ストア JKS ファイル</b>	WebLogic を SSL で実行している場合。 たとえば, <b>DemoTrust.jks</b>
<b>jsafeFIPS.jar</b>	WebLogic を SSL で実行している場合。  <b>注:</b> WebLogic 8.1 SP5 以降
<b>wlfullclient.jar</b>	WebLogic を SSL で実行している場合。  <b>wlfullclient.jar</b> は JarBuilder ツールを使用して最初に生成する  i. ディレクトリを %weblogic.home%/server/lib2 に変更  ii. java -jar wljarbuilder.jar を実行  <b>注:</b> WebLogic 9.x および 10.x
<b>wlclient.jar</b>	WebLogic 9.x および 10.x のみ
<b>wljsxclient.jar</b>	WebLogic 9.x および 10.x のみ

- b. ドライバを次の場所にある正しいバージョン・フォルダに配置します。

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager  
\discoveryResources\j2ee\weblogic\<version\_folder>

例 :

C:\hp\UCMDB\DataFlowProbe\runtime\probeManager  
\discoveryResources\j2ee\weblogic\10.x

- c. DFM ジョブを実行する前にプローブを再起動します。

### 3. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **JEE TCP Ports** ジョブを実行して、サービス・エンドポイントの情報を検出します。ジョブの詳細については、[「JEE TCP Ports ジョブ」\(818ページ\)](#)を参照してください。
- c. **JEE Weblogic Connections by JMX** ジョブを実行して、アプリケーション・サーバのシャロー・ディスカバリを実行します。ジョブの詳細については、[「JEE Weblogic Connections by JMX ジョブ」\(822ページ\)](#)を参照してください。
- d. **JEE Weblogic by JMX** ジョブを実行して、アプリケーション・サーバトポロジのディープ・ディスカバリを実行します。ジョブの詳細については、[「JEE Weblogic by JMX ジョブ」\(825ページ\)](#)を参照してください。

## WebLogic Topology by Shell の検出方法

WebLogic ディスカバリ・プロセスでは、J2EE アプリケーション、JDBC、および JMS リソースを含む WebLogic トポロジー 式を検出できます。DFM は Shell プロトコルまたはエンドポイント (TCP ポート) に基づいて最初にアプリケーション・サーバを検出し、次にシェルによって WebLogic J2EE 環境とコンポーネントを検出します。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、シェル・プロトコルを使用します。次のいずれかのプロトコルの資格情報を定義します。

- NTCMD プロトコル
- SSH プロトコル
- Telnet プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリ・ワークフロー

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by Shell** ジョブを実行して、ターゲット・ホストおよびそのホストへのシェル接続を検出します。
- c. 次の 2 つのジョブのいずれかを実行します。
  - **Host Applications by Shell** ジョブを実行して、実行中のプロセスを含めてターゲット・ホストのリソースを検出します。
  - **JEE TCP Ports**。サービス・エンドポイントの情報を検出する場合に実行します。ジョブの詳細については、「[JEE TCP Ports ジョブ](#)」(818ページ)を参照してください。
- d. **JEE Weblogic by Shell** ジョブを実行します。ジョブの詳細については、「[JEE Weblogic by Shell ジョブ](#)」(830ページ)を参照してください。

## JEE TCP Ports ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、**TCP Ports Discovery** アダプタを使用します。

### トリガ・クエリ



### ノード条件

ノード名	条件
IpAddress	NOT IP Probe Name null

## ジョブ・パラメータ

名前	標準設定値	説明
<b>checkIfIpsReachable</b>	true	このフラグは、ホストのポートが利用可能かどうかの確認をジョブが開始する前に、検出された IP に到達可能であることを確認する必要があるかどうかを示します。
<b>checkOnlyKnownPorts</b>	true	このフラグは、既知のポートのみをこのジョブで検出するかどうかを示します。このフラグによって、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータがキャンセルされることはありません。このフラグの <b>false</b> への設定は、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータ内の実際のポート範囲と合わせて使用する場合にのみ適用されます。
<b>connectTimeOut</b>	5000	IP およびポートに接続するときのタイムアウトをミリ秒単位で示します。
<b>nmapPath</b>		nmap 実行可能ファイルのフルパスを示します (例 : <b>C:\Program Files (x86)\Nmap\nmap.exe</b> )。
<b>pingTimeOut</b>	2000	ICMP ping のタイムアウトをミリ秒単位で示します。
<b>ports</b>	<p><b>JEE TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>weblogic, weblogicSSL, websphere_jmx, rmi</li> </ul> <p><b>Databases TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>oracle, db2, sybase, sql, mysql</li> </ul> <p><b>SAP TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>sap, sap_jmx, sap_http, sap_https</li> </ul>	このパラメータには、ディスカバリの実行に使用される TCP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の TCP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、 <b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。

名前	標準設定値	説明
	<b>SAP TCP Ports</b> ジョブの場合 : 標準設定値なし	
<b>scanUDP</b>	false	このフラグは、UDP ポートをスキャンするかどうかを示します。  <b>注:</b> UDP スキャンは、 <b>useNMap=true</b> に設定されている場合にのみサポートされます(下記を参照)。
<b>UDPports</b>		このパラメータには、ディスカバリの実行に使用される UDP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の UDP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、 <b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。
<b>useNMap</b>	<b>Databases TCP Ports</b> ジョブおよび <b>JEE TCP Ports</b> ジョブの場合 : false  <b>SAP TCP Ports</b> ジョブおよび <b>TCP Ports</b> ジョブの場合 : true	このフラグは、ポートのスキャン中に nmap を使用するかどうかを示します。  <b>注:</b> <b>nmapPath</b> にパスが指定されないと(上記を参照)、システム・パスからの nmap が使用されます。

**注:** **ports** パラメータまたは **UDPports** パラメータ内でポート名が割り当てられており、**portNumberToPortName.xml** 構成ファイル内で検出可能 (**isDiscovered=1**) としてマークされているポートのみが検出対象になります。

## アダプタ情報

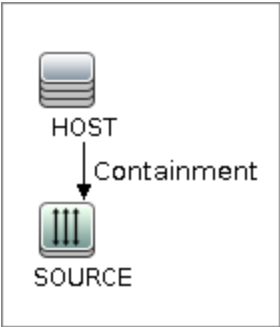
このアダプタでは、TCP ポートを検出します。



入力 CIT

IpAddress

入力クエリ



トリガ CI データ

名前	値
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}

使用スクリプト

- TcpPortScanner.py
- nmap.py

グローバル構成ファイル

portNumberToPortName.xml

検出された CIT

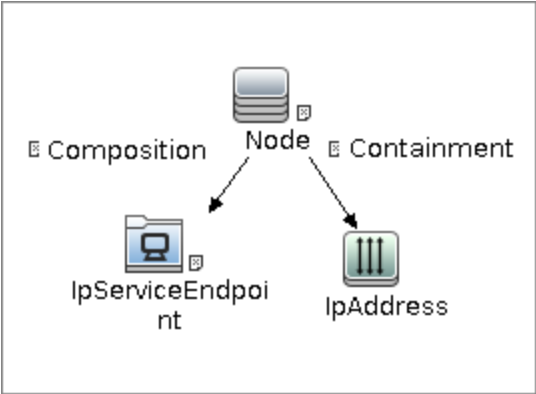
- Composition
- Containment
- IpAddress

- IpServiceEndpoint
- Node

## JEE Weblogic Connections by JMX ジョブ

本 項 ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



- ノード条件

ノード名	条件
Node	なし
IpServiceEndPoint	IpServiceName 等しい weblogic  OR IpServiceName 等しい weblogicSSL  OR ServiceNames 含む weblogic  OR ServiceNames 含む weblogicSSL
IpAddress	NOT IP Probe Name null

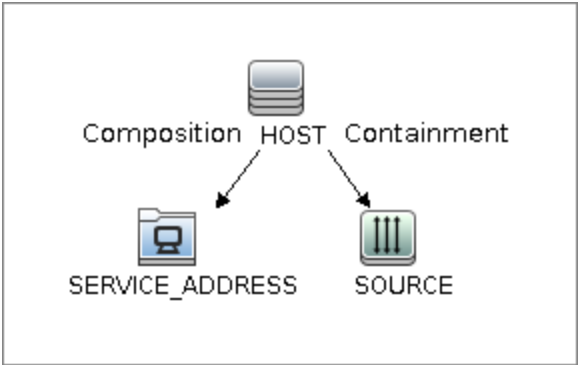
### ジョブ・パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

**アダプタ - JMX\_J2EE\_WebLogic\_Connection**

このアダプタは、Weblogic サーバ・ディスカバリで使 用します。

- 入 力 CIT : IpAddress
- 入 力クエリ:



- トリガ CI データ

名 前	値
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}
ports	\${SERVICE_ADDRESS.network_port_number:NA}
hostId`	\${HOST.root_id}

- 使用スクリプト
  - connection.py
  - db.py
  - db\_builder.py
  - db\_platform.py
  - entity.py
  - iteratorortools.py

- jdbc.py
- jdbc\_ulr\_parser.py
- jee.py
- jee\_connection.py
- jee\_discoverer.py
- jms.py
- jmx.py
- JMX\_J2EE\_WebLogic\_Connection.py
- protocol.py
- weblogic.py
- weblogic\_discoverer.py
- グローバル構成ファイル: なし
- アダプタ・パラメータ

名前	値	説明
remoteJVMArgs	-Xms64m -Xmx256m - XX:PermSize=256m - XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータ。
runInSeparateProcess	true	個別のスレッドでパターン実行される。

検出された CIT

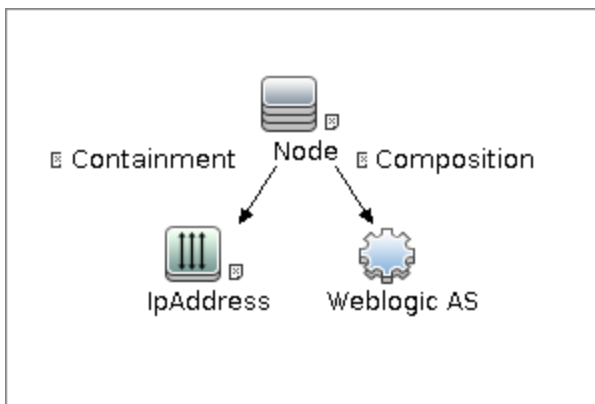
- Composition
- Containment
- IpAddress

- IpServiceEndPoint
- J2EE Domain
- JEE Node
- JVM
- Membership
- Node
- Usage
- WebLogic AS

## JEE Weblogic by JMX ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



- ノード条件

ノード名	条件
<b>Node</b>	なし
<b>IpAddress</b>	NOT IP Probe Name null
<b>Weblogic AS</b>	NOT Reference to the credentials dictionary entry null

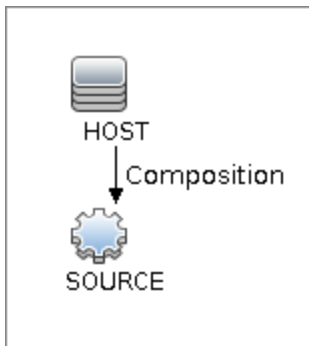
## ジョブ・パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

## アダプタ - JMX\_J2EE\_WebLogic

このアダプタは Weblogic J2EE Topology Discovery by JMX に使用されます。

- 入力 CIT :Weblogic AS
- 入力クエリ:



- トリガ CI データ

名前	値
<b>credentialsId</b>	\${SOURCE.credentials_id}
<b>ip_address</b>	\${SOURCE.application_ip}
<b>port</b>	\${SOURCE.application_port}

名前	値
<b>servername</b>	\${SOURCE.name}
<b>version</b>	\${SOURCE.application_version}
<b>protocol</b>	\${SOURCE.j2eeserver_protocol}

- 使用スクリプト
  - connection.py
  - db.py
  - db\_builder.py
  - db\_platform.py
  - entity.py
  - iteratortools.py
  - jdbc.py
  - jdbc\_url\_parser.py
  - jee.py
  - jee\_connection.py
  - jee\_discoverer.py
  - jms.py
  - jmx.py
  - JMX\_J2EE\_WebLogic.py
  - protocol.py
  - weblogic.py
  - weblogic\_discoverer.py

- グローバル構成ファイル : globalSettings.xml
- アダプタ・パラメータ

名前	値	説明
<b>deploymentDescriptors</b>	true	J2EE アプリケーション, EJB モジュール, Web モジュールのデプロイ記述子を取得するには <b>true</b> に設定する(値 : true/false)。
<b>discoverAppResources</b>	true	true に設定した場合, モジュール, EJB, サブレットが検出される。
<b>discoverJMSResources</b>	true	true に設定した場合, jms プロバイダと jms サーバが検出される。
<b>remoteJVMArgs</b>	-Xms64m -Xmx256m -XX:PermSize=256m -XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータ。
<b>runInSeparateProcess</b>	true	個別のスレッドでパターン実行される。
<b>discoverDeployedOnly Applications</b>	true	デプロイされていて実行ステータスにあるアプリケーションが検出される

## 検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Database
- Dependency
- Deployed
- IpAddress
- IpServiceEndpoint



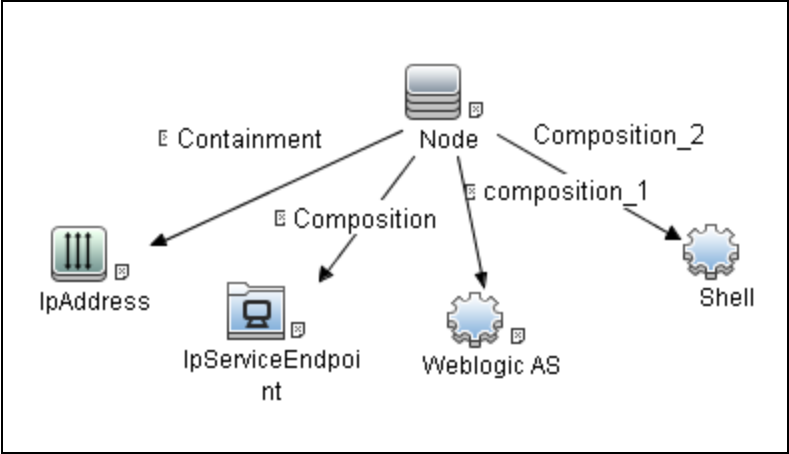
- J2eeCluster
- J2EE Domain
- J2EE Managed Object
- JdbcDataSource
- JEE Node
- Membership
- Node
- Usage
- Web Service
- Weblogic AS

**注:** JDBC Datasource を検出するには、事前に Weblogic 管理コンソールでアクティブ化しておく必要があります。

## JEE Weblogic by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



ノード名	条件
Node	なし
Shell	なし
Weblogic AS	なし
IpAddress	NOT IP Probe Name null
IpServiceEndPoint	IpServiceName 等しい weblogic OR IpServiceName 等しい weblogicSSL OR ServiceNames 含む weblogic OR ServiceNames 含む weblogicSSL

### パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

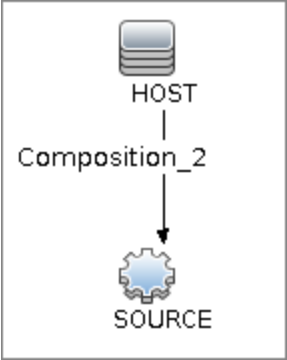
### アダプタ情報

このジョブでは **WebLogic\_By\_Shell** アダプタを使用します。

### 入力 CIT

Shell

入力クエリ



トリガ CI データ

名前	値
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}
Protocol	\${SOURCE.root_class}

使用スクリプト

- connection.py
- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- file\_ver\_lib.py
- fptools.py

- iteratortools.py
- jdbc.py
- jdbc\_url\_parser.py
- jee.py
- jee\_connection.py
- jee\_discoverer.py
- jms.py
- jmx.py
- process.py
- process\_discoverer.py
- protocol.py
- weblogic.py
- weblogic\_by\_shell.py
- weblogic\_discoverer.py
- asm\_weblogic\_discoverer.py

## グローバル構成ファイル

globalSettings.xml

## アダプタ・パラメータ

なし

## 検出された CIT

- Composition
- ConfigurationDocument

- Containment
- Database
- Database Schema
- Dependency
- Deployed
- IpAddress
- IpServiceEndpoint
- J2eeCluster
- J2EE Domain
- J2EE Managed Object
- JdbcDataSource
- JEE Node
- Membership
- Node
- Usage
- Web Service
- Weblogic AS

## トラブルシューティングおよび制限事項

### トラブルシューティング

- **問題** : SSL プロトコルを使用して WebLogic by JMX ジョブを実行する場合、UCMDB サーバと Data Flow Probe が SSL プロトコルを使用して接続されていると、ジョブはターゲット・ノードに接続できません。

次の代替解決策があります。

**解決策 1 :** UCMDB サーバと Data Flow Probe 間の HTTP 接続を設定します。

**解決策 2 :** WebLogic サーバへの非 SSL 接続を許可し、UCMDB JMX 資格情報を設定します。  
SSL 接続は使用しません。

**解決策 3 :** 次の引数を追加して、ジョブ(JEE WebLogic Connections by JMX ジョブおよび JEE WebLogic by JMX ジョブ)の **remoteJVMArgs** パラメータを更新します。

```
Djavax.net.ssl.trustStore=..\runtime\probeManager\discoveryResources
\j2ee\websphere\UCMDB_store.jks
```

### 制限事項

- Weblogic バージョン 8.x 以前では、DFM は WebLogic の設定 ウィザードで作成されたドメインのみを検出します。
- WebLogic 9 より前のバージョンでは、JEE WebLogic by Shell ジョブは管理サーバ・ホストでのみ実行できます。WebLogic バージョン 9 以降では、このジョブは管理ノードのみが含まれるホストでも実行できます。
- DFM が J2EE アプリケーションを検出できるのは、その .ear ファイルがフォルダに解凍されている場合に限られます。
- WebLogic のインストールには、標準設定でフィルタリングによって除外される例が含まれます。このフィルタは、**weblogic\_by\_shell.py** Jython スクリプトで削除できます。**WL\_EXAMPLE\_DOMAINS = 'medrec'**を検索してください。

## 第56章: WebSphere ディスカバリ

### 本章の内容

概要 .....	836
サポートされるバージョン .....	836
WebSphere Topology by JMX の検出方法 .....	837
WebSphere Topology by Shell の検出方法 .....	840
JEE TCP Ports ジョブ .....	842
アダプタ情報 .....	844
JEE WebSphere Connections by JMX ジョブ .....	846
JEE WebSphere by Shell or JMX ジョブ .....	850
アダプタ情報 .....	850
JEE WebSphere by Shell ジョブ .....	855
アダプタ情報 .....	855
トラブルシューティングおよび制限事項 .....	861

## 概要

本項では、WebSphere アプリケーション・センタを検出する方法について説明します。WebSphere ディスカバリ・プロセスでは、J2EE アプリケーション、JDBC、および JMS リソースを含む WebSphere トポロジースを検出できます。

## サポートされるバージョン

WAS のバージョン	J2EE のバージョン	JVM のバージョン
5.0	J2EE 1.3	JVM 1.3
5.1	J2EE 1.3	JVM 1.4
6.0	J2EE 1.4	JVM 1.4
6.1	J2EE 1.4	JVM 1.5
7.0	Java EE 5	JVM 1.6



## WebSphere Topology by JMX の検出方法

DFM は SOAP または RMI 認証に基づいて最初に WebSphere サーバを検出し、次に WebSphere J2EE 環境とコンポーネントを検出します。

このタスクでは JMX ごとの WebSphere 接続の検出方法を次の手順で説明します。

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリは、WebSphere プロトコルからの資格情報を使用した JMX プロトコルに基づきます。WebSphere プロトコル資格情報を定義する必要があります。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - ドライバの設定

WebSphere の検出に必要なドライバを設定します。標準設定では、プローブのインストールに標準設定の WebSphere ドライバが含まれています。

プローブのインストールにはバージョン 5 および 6 の WebSphere ドライバが含まれていますが、必要に応じてご自身のドライバを使用することもできます。ただし、サポートされるバージョンで機能するドライバしか使用できません。サポートされるバージョンの詳細については、「検出されるアプリケーション」を参照してください。

**.jar ファイルを更新するには、次の手順で行います。**

- a. 次の場所にある適切なバージョンのフォルダにドライバをコピーします。

```
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager  
\discoveryResources\j2ee\websphere\<version_folder>
```

例 :

```
C:\hp\UCMDB\DataFlowProbe\runtime\probeManager  
\discoveryResources\j2ee\websphere\5.x
```

- b. DFM ジョブを実行する前にプローブを再起動します。

### 3. .jar ファイルの更新

- a. 次のファイルを WebSphere 8.x アプリケーション・サーバからコピーします。

- **<WebSphere root folder>** \AppServer\runtimes\com.ibm.ws.orb\_8.x.0.jar
- **<WebSphere root folder>** \AppServer\runtimes\com.ibm.ws.ejb.thinclient\_8.x.0.jar
- **<WebSphere root folder>** \AppServer\runtimes\com.ibm.ws.admin.client\_8.x.0.jar
- **<WebSphere root folder>** \AppServer\deploytool\itp\plugins\  
<com.ibm.websphere.v8\_....>\wasJars\ibmpkcs.jar
- **<WebSphere root folder>** \AppServer\deploytool\itp\plugins\  
<com.ibm.websphere.v8\_....>\wasJars\ibmkeycert.jar
- **<WebSphere root folder>** \AppServer\deploytool\itp\plugins\  
<com.ibm.websphere.v8\_....>\wasJars\ibmjceprovider.jar

ここで、**<WebSphere root folder>** は WebSphere をインストールしたフォルダを指します。

- b. Data Flow Probe を停止します。
- c. **UCMDB\_store.jks** を除き **<DataFlowProbe root folder>** \runtime\probeManager\discoveryResources\j2ee\websphere\ のすべてのファイルをバックアップします。

ここで、**<DataFlowProbe root folder>** は Data Flow Probe をインストールしたフォルダを指します。

- d. **<DataFlowProbe root folder>** \runtime\probeManager\discoveryResources\j2ee\websphere\ から **UCMDB\_store.jks** を除くすべてのファイルを削除します。
- e. [手順 a](#) でコピーしたすべてのファイルを Data Flow Probe の次の場所に配置します：  
**<DataFlowProbe root folder>** \runtime\probeManager\discoveryResources\j2ee\websphere
- f. Data Flow Probe を再起動します。

#### 4. ディスカバリの実行

次のジョブを次の順序で実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **JEE TCP Ports** ジョブを実行して、サービス・エンドポイントの情報を検出します。ジョブの詳細については、[「JEE TCP Ports ジョブ」\(842ページ\)](#)を参照してください。
- c. **JEE WebSphere Connections by JMX** ジョブを実行して、アプリケーション・サーバのシャロー・ディスカバリを実行します。ジョブの詳細については、[「JEE WebSphere Connections by JMX ジョブ」\(846ページ\)](#)を参照してください。
- d. **JEE WebSphere by Shell or JMX** ジョブを実行して、アプリケーション・サーバトポロジのディープ・ディスカバリを実行します。ジョブの詳細については、[「JEE WebSphere by Shell or JMX ジョブ」\(850ページ\)](#)を参照してください。

## WebSphere Topology by Shell の検出方法

このタスクでは、Shell プロトコルを使用して WebSphere トポロジー式を検出する方法について説明します。WebSphere ディスカバリ・プロセスでは、IBM WebSphere サーバにデプロイされている Web サービスを検出します。検出される Web サービスは、CMDB の webservice CIT によって表されます。

DFM は、Shell プロトコルまたはエンドポイント(TCP ポート)に基づいて最初にアプリケーション・サーバを検出し、次に Shell ごとの WebSphere J2EE 環境とコンポーネントを検出します。

本項の内容

- [「前提条件 - プロトコル資格情報の設定」\(840ページ\)](#)
- [「前提条件 - キー・ストアの設定」\(840ページ\)](#)
- [「ディスカバリの実行」\(841ページ\)](#)

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、シェル・プロトコルを使用します。次のいずれかのプロトコルを定義する必要があります。

- SSH プロトコル
- Telnet プロトコル
- NTCMD プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - キー・ストアの設定

次の手順は、それぞれが特定の WebSphere サーバでの識別に必要な 2 つのキー・ストアが含まれるクライアント・マシンを実行している場合に関係します。クライアントが間違ったキー・ストアを使用していずれかの WebSphere サーバに接続しようとすると、試行は失敗します。次にクライアントが 2 番目の正しいキー・ストアを使用して WebSphere サーバに接続すると、その試行も失敗します。

- **解決策 1:** すべての WebSphere サーバについてクライアントで 1 つのキー・ストアを設定します。
- **解決策 2:** 同じユーザ名 およびパスワードを使用するすべての WebSphere サーバについて IP ア

ドレス範囲ごとに 1 つのキー・ストアを設定します。異なるユーザ名 およびパスワードを使用するサーバについては、別の IP 範囲でキー・ストアを設定します。

### 3. ディスカバリの実行

次のジョブを次の順序で実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by Shell** ジョブを実行して、ターゲット・ホストおよびそのホストへのシェル接続を検出します。
- c. 次のいずれかのジョブを実行します。
  - **Host Applications by Shell** ジョブを実行して、実行中のプロセスを含めてターゲット・ホストのアプリケーションを検出します。
  - **JEE TCP Ports** ジョブを実行して、サービス・エンドポイントの情報を検出します。ジョブの詳細については、[「JEE TCP Ports ジョブ」\(842ページ\)](#)を参照してください。
- d. **JEE WebSphere by Shell** ジョブを実行します。ジョブの詳細については、[「JEE WebSphere by Shell ジョブ」\(855ページ\)](#)を参照してください。

# JEE TCP Ports ジョブ

本項ではこのジョブに関する詳細を説明します。

## アダプタ

このジョブでは、**TCP Ports Discovery** アダプタを使用します。

## トリガ・クエリ



## ノード条件

ノード名	条件
IpAddress	NOT IP Probe Name null

## ジョブ・パラメータ

名前	標準設定値	説明
checkIfIPsReachable	true	このフラグは、ホストのポートが利用可能かどうかの確認をジョブが開始する前に、検出された IP に到達可能であることを確認する必要があるかどうかを示します。
checkOnlyKnownPorts	true	このフラグは、既知のポートのみをこのジョブで検出するかどうかを示します。このフラグによって、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータがキャンセルされることはありません。このフラグの <b>false</b> への設定は、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータ内の実際のポート範囲と合わせて使用する場合にのみ適用されます。
connectTimeout	5000	IP およびポートに接続するときのタイムアウトをミリ秒単位で示します。

名前	標準設定値	説明
<b>nmapPath</b>		nmap 実行可能ファイルのフルパスを示します (例 : <b>C:\Program Files (x86)\Nmap\nmap.exe</b> )。
<b>pingTimeOut</b>	2000	ICMP ping のタイムアウトをミリ秒単位で示します。
<b>ports</b>	<p><b>JEE TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>weblogic, weblogicSSL, websphere_jmx, rmi</li> </ul> <p><b>Databases TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>oracle, db2, sybase, sql, mysql</li> </ul> <p><b>SAP TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>sap, sap_jmx, sap_http, sap_https</li> </ul> <p><b>SAP TCP Ports ジョブの場合 : 標準設定値なし</b></p>	<p>このパラメータには、ディスカバリの実行に使用される TCP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の TCP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、<b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。</p>
<b>scanUDP</b>	false	<p>このフラグは、UDP ポートをスキャンするかどうかを示します。</p> <div> <p><b>注:</b> UDP スキャンは、<b>useNMap=true</b> に設定されている場合にのみサポートされます(下記を参照)。</p> </div>
<b>UDPports</b>		このパラメータには、ディスカバリの実行に使用される UDP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての

名前	標準設定値	説明
		既知の UDP ポートに対してのみ実行されます。 1000-1100 などのポート範囲が入力された場合、 <b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。
<b>useNMap</b>	<b>Databases TCP Ports ジョブおよび JEE TCP Ports ジョブの場合 :</b> false  <b>SAP TCP Ports ジョブおよび TCP Ports ジョブの場合 :</b> true	このフラグは、ポートのスキャン中に nmap を使用するかどうかを示します。  <div> <b>注 :</b> nmapPath にパスが指定されないと(上記を参照), システム・パスからの nmap が使用されます。         </div>

**注 :** ports パラメータまたは UDPports パラメータ内でポート名が割り当てられており, portNumberToPortName.xml 構成ファイル内で検出可能(isDiscovered=1)としてマークされているポートのみが検出対象になります。

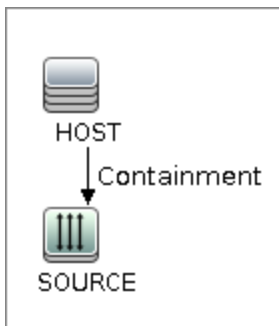
## アダプタ情報

このアダプタでは、TCP ポートを検出します。

### 入力 CIT

IpAddress

### 入力クエリ





トリガ CI データ

名前	値
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}

使用スクリプト

- TcpPortScanner.py
- nmap.py

グローバル構成ファイル

portNumberToPortName.xml

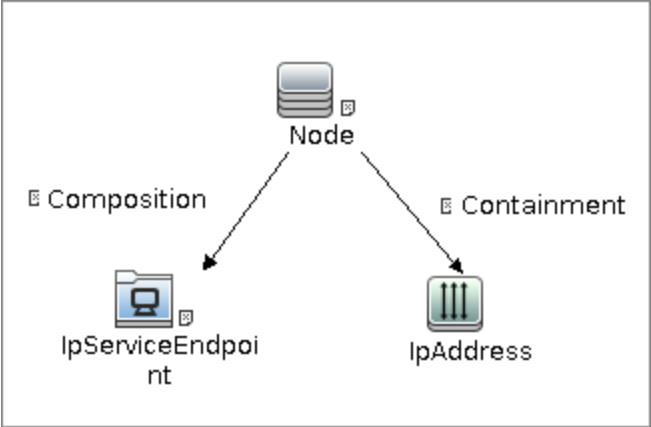
検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node

## JEE WebSphere Connections by JMX ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



### ノード条件

ノード名	条件
Node	なし
IpServiceEndpoint	IpServiceName 等しい websphere_jmx OR ServiceNames 含む websphere_jmx
IpAddress	NOT IP Probe Name null

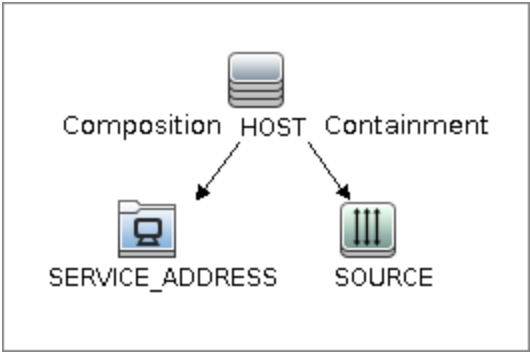
### ジョブ・パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

### アダプタ - JMX\_J2EE\_WebSphere\_Connection

このアダプタは WebSphere サーバ・ディスカバリに使用されます。

- 入力 CIT : IpAddress
- 入力クエリ:



• トリガ CI データ

名前	値
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}
ports	\${SERVICE_ADDRESS.network_port_number:NA}
hostId	\${HOST.root_id}
ip_dnsname	\${SOURCE.authoritative_dns_name:NA}

• 使用スクリプト

- connection.py
- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- iteratortools.py
- jdbc.py
- jdbc\_url\_parser.py
- jee.py

- jee\_connection.py
- jee\_discoverer.py
- jms.py
- jmx.py
- JMX\_J2EE\_WebSphere\_Connection.py
- protocol.py
- websphere.py
- グローバル構成ファイル: なし
- パラメータ

名前	値	説明
remoteJVMArgs	-Xms64m -Xmx256m - XX:PermSize=256m - XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータ。
runInSeparateProcess	true	個別のスレッドでパターン実行される。

検出された CIT

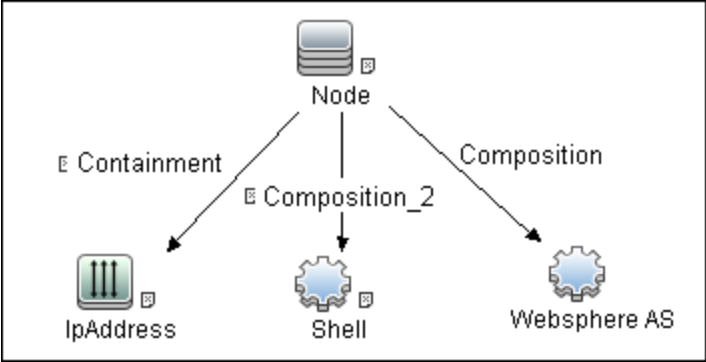
- Composition
- IpAddress
- IpServiceEndpoint
- J2EE Domain
- JEE Node
- Node
- Usage

- Websphere AS

## JEE WebSphere by Shell or JMX ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



ノード名	条件
Node	なし
IpAddress	NOT IP Probe Name null
Websphere AS	NOT Reference to the credentials dictionary entry null
Shell	NOT Reference to the credentials dictionary entry null

### ジョブ・パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

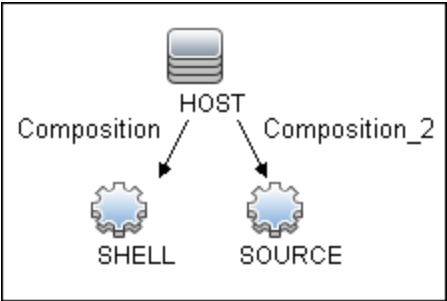
## アダプタ情報

このジョブでは **JMX\_J2EE\_WebSphere** アダプタを使用します。

### 入力 CIT

WebSphere AS

入力クエリ



トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip	\${SHELL.application_ip:NA}
ip_address	\${SHELL.application_ip}
port	\${SHELL.application_port:NA}
protocol	\${SHELL.root_class:NA}
shellCredentialsId	\${SHELL.credentials_id:NA}
version	\${SOURCE.application_version}

使用スクリプト

- connection.py
- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- iteratortools.py

- jdbc.py
- jdbc\_url\_parser.py
- jee.py
- jee\_connection.py
- jee\_discoverer.py
- jms.py
- jmx.py
- JMX\_J2EE\_WebSphere.py
- protocol.py
- websphere.py
- websphere\_discoverer.py

グローバル構成ファイル

globalSettings.xml

アダプタ・パラメータ

名前	値	説明
applications	なし	検出するアプリケーションのリスト(カンマ区切り)。
discoverAppResources	true	true に設定した場合、モジュール、EJB、サーブレットが検出される。
discoverConfigFile	true	true に設定した場合、セル、サーバ、およびアプリケーションの追加の構成ファイルを検出する。
discoverEAR	true	true に設定した場合、J2EE アプリケーションの EAR ファイルを検出する。
discoverJDBCResources	true	true に設定した場合、jdbc プロバイダおよびデータソースを検出する。



名前	値	説明
discoverJMSResources	true	true に設定した場合、jms プロバイダと jms サーバが検出される。
remoteJVMArgs	-Xms64m -Xmx256m - XX:PermSize=256m - XX:MaxPermSize=256m	リモート・プロセスに渡される必要がある JVM パラメータ。
runInSeparateProcess	true	パターンを個別のスレッドで実行する必要があるかどうかを決定する。
servers	なし	検出するサーバのリスト (カンマ区切り)。

## 検出された CIT

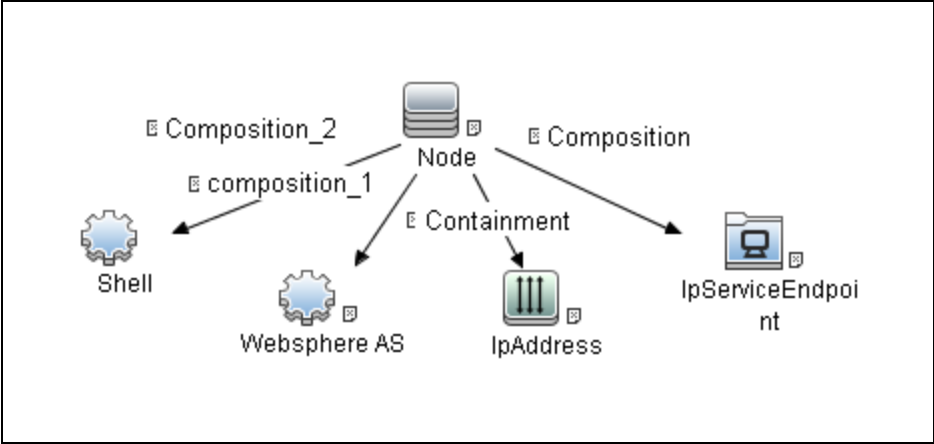
- Composition
- ConfigurationDocument
- Database
- Dependency
- Deployed
- IpAddress
- IpServiceEndpoint
- J2eeCluster
- J2EE Domain
- J2EE Managed Object
- JdbcDataSource
- JEE Node
- Membership
- Node

- Usage
- Web Service
- Websphere AS

## JEE WebSphere by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



ノード名	条件
Node	なし
IpAddress	NOT IP Probe Name null
WebSphere AS	なし
Shell	NOT Reference to the credentials dictionary entry null
IpServiceEndPoint	IpServiceName 等しい websphere_jmx OR ServiceNames 含む websphere_jmx

### ジョブ・パラメータ

標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

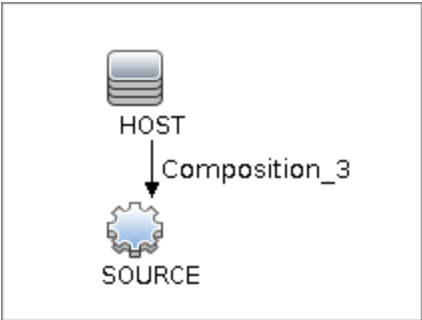
### アダプタ情報

このジョブでは **WebSphere\_By\_Shell** アダプタを使用します。

### 入力 CIT

Shell

入力クエリ



トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}

使用スクリプト

- asm\_websphere\_discoverer.py
- connection.py
- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- file\_ver\_lib.py
- fptools.py
- iteratortools.py

- jdbc.py
- jdbc\_url\_parser.py
- jee.py
- jee\_connection.py
- jee\_constants.py
- jee\_discoverer.py
- jms.py
- jmx.py
- process.py
- process\_discoverer.py
- protocol.py
- websphere.py
- websphere\_by\_shell.py
- websphere\_discoverer.py

## グローバル構成ファイル

globalSettings.xml

## アダプタ・パラメータ

なし

## 検出要素

DFM は次の要素を検出します。

- バージョン番号

DFM は、**<WebSphere の基本ディレクトリ>\properties\version** フォルダの **WAS.product** または **BASE.product** ファイル(WebSphere バージョンによって異なる)から WebSphere Application Server のバージョン番号を検出します。

- サーバのリスナ・ポートとアドレス

DFM は、**<WebSphere の基本ディレクトリ>\profiles\<プロファイル>\config\cells\<セル>\nodes\<ノード>** フォルダまたは **<WebSphere の基本ディレクトリ>\config\cells\<セル>\nodes\<ノード>** フォルダのいずれかにある **serverindex.xml** ファイルを検索して、WebSphere サーバに関する情報を取得します。

- J2EE アプリケーション

DFM は、各 **<WebSphere の基本ディレクトリ>\profiles\<プロファイル>\config\cells\<セル>\applications** フォルダ(または **<WebSphere の基本ディレクトリ>\config\cells\<セル>\nodes\<ノード>\applications** フォルダ)で **deployment.xml** ファイルを検索します。**deployment.xml** ファイルはすべてのインストール済みアプリケーション・フォルダにあり、アプリケーション・ターゲットに関する情報が含まれています。

- 構成ファイル

DFM は、**resources.xml** リソース構成ファイルの CI を作成します。CI は、各セル、ノード、およびサーバについて作成され(関連するプレフィックス付き)、各 CI は WebSphere サーバ CI にアタッチされます。

- JMS リソース

WebSphere JMS リソースは JMS プロバイダとして設定されます。リソースには主に**接続ファクトリ**と**宛先**(トピック、キュー)の 2 つの種類があります。これらはさらに次のようにカテゴリ分けされる場合があります。

- 接続ファクトリ

- resources.jms.mqseries:MQConnectionFactory

- キュー接続ファクトリ

- resources.jms.mqseries:MQQueueConnectionFactory
    - resources.jms.internalmessaging:WASQueueConnectionFactory

- トピック接続ファクトリ

- resources.jms.mqseries:MQTopicConnectionFactory
- resources.jms.internalmessaging:WASTopicConnectionFactory
- キューまたはトピック
  - resources.jms.GenericJMSDestination
  - resources.jms.mqseries:MQTopic
  - resources.jms.mqseries:MQQueue
  - resources.jms.internalmessaging:WASTopic
  - resources.jms.internalmessaging:WASQueue

DFM は、上記すべてのタイプを使用して、使用されるリソースに関する情報を取得しようとします。ディスカバリでは、異なるデプロイ範囲で構成ファイル **resources.xml** が探索されます。次の表は、デプロイ範囲と構成ファイルへの相対パスを示します。

範囲	相対ファイル・パス
<b>Cell</b>	<プロファイル>\config\cells\<セル>\resources.xml
<b>Cluster</b>	<プロファイル>\config\cells\<セル>\clusters\<クラスタ>\resources.xml
<b>Node</b>	<プロファイル>\config\cells\<セル>\nodes\<ノード>\resources.xml
<b>Server</b>	<プロファイル>\config\cells\<セル>\nodes\<ノード>\servers\<サーバ>\resources.xml

注: ファイル・パスは、<プロファイル> ホーム・ディレクトリに対して相対です。

## 検出された CIT

- Composition
- ConfigurationDocument
- Containment
- Database

- Database Schema
- Dependency
- Deployed
- IpAddress
- IpServiceEndpoint
- J2eeCluster
- J2EE Domain
- J2EE Managed Object
- JdbcDataSource
- JEE Node
- Membership
- Node
- Usage
- Web Service
- Websphere AS



## トラブルシューティングおよび制限事項

本項では、WebSphere ディスカバリのトラブルシューティングと制限事項について説明します。

### トラブルシューティング

- **問題** : SSL プロトコルを使用して Websphere by JMX ジョブを実行する場合、UCMDB サーバと Data Flow Probe が SSL プロトコルを使用して接続されていると、ジョブはターゲット・ノードに接続できません。

次の代替解決策があります。

**解決策 1** : UCMDB サーバと Data Flow Probe 間の HTTP 接続を設定します。

**解決策 2** : Websphere サーバへの非 SSL 接続を許可し、UCMDB JMX 資格情報を設定します。  
SSL 接続は使用しません。

**解決策 3** : 次の引数を追加して、ジョブ(JEE WebSphere Connections by JMX ジョブおよび JEE WebSphere by Shell or JMX ジョブ)の **remoteJVMArgs** パラメータを更新する。

```
Djavax.net.ssl.trustStore=..\runtime\probeManager\discoveryResources  
\j2ee\websphere\UCMDB_store.jks
```

### 制限事項

- DFM が同じホスト上で同じ名前のセルを 2 つ見つけた場合、1 つのセル構成 (**j2eedomain** トポロジ) のみがレポートされます。
- EJB および Web Service CI は検出されません。
- DFM は、その **.ear** ファイルをフォルダに解凍している場合のみ J2EE アプリケーションを検出できます。
- ジョブ(スクリプト)は jks\* キー形式の証明書でのみ機能します。

## 第10部: Middleware > Messaging Servers

## 第57章: Microsoft MQ (メッセージ・キュー)ディスカバリ

### 本章の内容

サポートされるバージョン .....	864
Microsoft MQ の検出方法 .....	864
Microsoft Message Queue Topology by NTCMD or UDA ジョブ .....	866
Microsoft Message Queue Topology by LDAP ジョブ .....	869
Microsoft MQ ディスカバリ・スクリプト .....	870
Microsoft MQ トポロジのディスカバリ方法 .....	871
Host Applications by Shell ジョブ .....	871
Microsoft Message Queue Topology by NTCMD or UDA ジョブ .....	874
Microsoft Message Queue Topology by LDAP ジョブ .....	880

## サポートされるバージョン

MS-MQ デイスカバリは、MS MQ バージョン 3.0 以降をサポートしています。

## Microsoft MQ の検出方法

Microsoft メッセージ・キュー(MS MQ)デイスカバリ・プロセスでは、Active Directory とともに実行されている MS MQ トポロジと、すべての MS MQ サーバの末端構成を検出できます。

デイスカバリ・フローは 2 つあります。各 デイスカバリ・フローの詳細を次に示します。

### 1. LDAP でデイスカバリを実行する

- a. **Range IPs by ICMP** ジョブまたは **Range IPs by nmap** ジョブを実行して、MS MQ システムの IP アドレスを検出します。
- b. **TCP Ports** ジョブを実行して、MS MQ システムの LDAP ポートを検出します。
- c. **Active Directory Connection by LDAP** ジョブを実行して、**Microsoft Message Queue Topology by LDAP** ジョブのデイスカバリに必要な LDAP 資格情報を検出します。
- d. **Microsoft Message Queue Topology by LDAP** ジョブを実行して、Active Directory トポロジ (フォレスト、サイト・リンク)を検出します。

### 2. NTCMD または UDA でデイスカバリを実行する

- a. **Range IPs by ICMP** ジョブまたは **Range IPs by nmap** ジョブを実行して、MS MQ システムの IP アドレスを検出します。
- b. **Host Connection by Shell** ジョブを実行して、**Host Applications by Shell** ジョブのデイスカバリに必要なシェル資格情報を検出します。
- c. **Host Applications by Shell** ジョブを実行します。この段階で、サーバ(MS MQ がインストールされている物理マシン)がドメインのメンバーである場合は、MS MQ マネージャと、ドメイン・コントローラを備えたマシンに関する情報が UCMDB に格納されます。
- d. **Microsoft Message Queue Topology by NTCMD or UDA** ジョブを実行して、サーバ側トポロジ(キュー、トリガ、ルール)を検出します。

**注:** 情報は、3 つの短いレジストリ・ブランチの構成ファイルからのみ取得され、各 ファイルは 2 KB 未

満であるため、システム・パフォーマンスへの影響はありません。

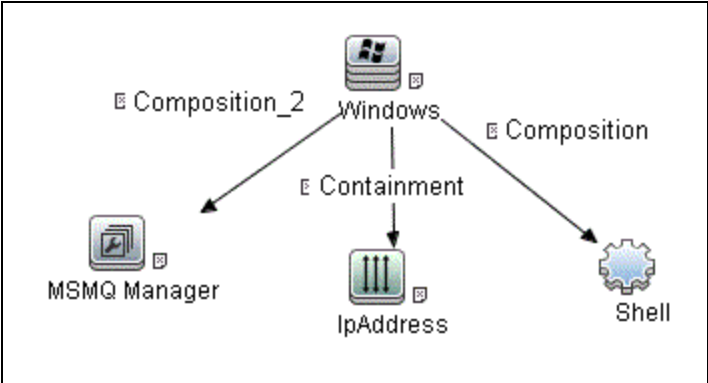
DFM が MQ トポロジを検出する方法の詳細については、[「Microsoft MQ トポロジのディスカバリ方法」\(871 ページ\)](#)を参照してください。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

# Microsoft Message Queue Topology by NTCMD or UDA ジョブ

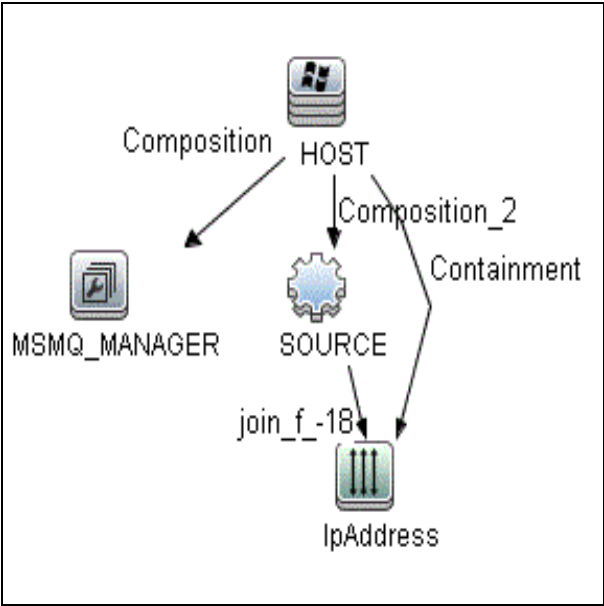
本 項 ではこのジョブに関する詳細を説明します。

## トリガ・クエリ



ノード名	条件
Windows	なし
MSMQ Manager	なし
IPAddress	NOT IP Probe Name Null
Shell	NOT Reference to the credentials dictionary entry null AND NOT Application IP null AND(CI Type 等しい ntcmd OR CI Type 等しい uda)

入力クエリ



ノード名	条件
SOURCE	CI Type 等しい ntcmd OR CI Type 等しい uda
HOST	なし
MSMQ_MANAGER	なし
IpAddress	NOT IP Probe Name Null

検出された CIT

- Composition
- Containment
- IpAddress
- MSMQ Manager
- MSMQ Queue
- MSMQ Rule

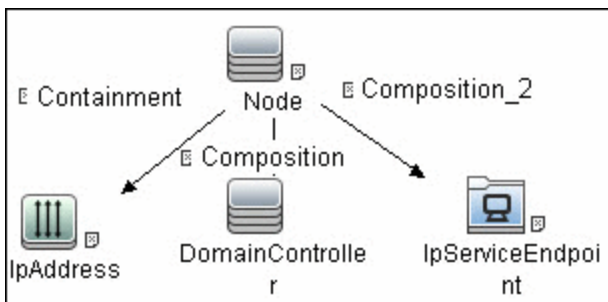
- MSMQ Trigger
- Node
- Usage



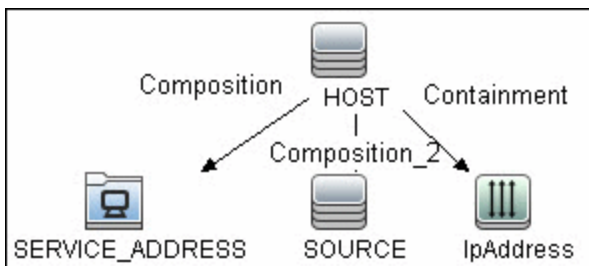
## Microsoft Message Queue Topology by LDAP ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



### 入力クエリ



### 検出された CIT

- Active Directory Forest
- ActiveDirectorySite
- ActiveDirectorySystem
- Composition
- Containment
- IPAddress
- MSMQ Manager

- Membership
- Node
- Usage

アダプタ・パラメータ

- **baseDn** この値により、ドメイン・コントローラ・サーバに関するレコードを保存する DN が決定されます。標準設定値は、**OU=Domain Controllers** です。

Microsoft MQ ディスカバリ・スクリプト

スクリプトを表示するには、[アダプタ管理]>[リソース]表示 枠に移動し、[Microsoft\_MQ]>[スクリプト]をクリックします。

スクリプト	説明
ntcmd_msmq.py	Microsoft Message Queue Topology by NTCMD or UDA ジョブのメイン・スクリプト
ldap_msmq.py	Microsoft Message Queue Topology by LDAP ジョブのメイン・スクリプト
plugin_microsoft_mq.py	MS MQ マネージャ・ディスカバリのシャロー・プラグイン ([アダプタ管理]>[リソース]表示 枠 > [Host_Resources_Basic]>[スクリプト])
host_resolve_utils.py	DNS 解決ユーティリティ ([アダプタ管理]>[リソース]表示 枠 > [Host_Resources_Basic]>[スクリプト])

## Microsoft MQ トポロジのディスカバリ方法

本項では DFM が MS MQ トポロジを検出する方法について説明します。

本項の内容

- [「Host Applications by Shell ジョブ」\(871ページ\)](#)
- [「Microsoft Message Queue Topology by NTCMD or UDA ジョブ」\(874ページ\)](#)
- [「Microsoft Message Queue Topology by LDAP ジョブ」\(880ページ\)](#)

### Host Applications by Shell ジョブ

このジョブでは、`plugin_microsoft_mq.py` スクリプトを使用します。

次のレジストリ・ブランチから情報が解析されます。

#### レジストリ・ブランチ(1)

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache\

##### • コマンド出力

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache
EnterpriseId  REG_BINARY    C209A2FE9203F64CB543441CC92A40DC
SiteId       REG_BINARY    FB7BA54DFF5F40429ECA64752D0130A0
MQS_DepClients  REG_DWORD     0x0
MQS           REG_DWORD     0x1
MQS_DsServer   REG_DWORD     0x0
MQS_Routing    REG_DWORD     0x1
QMId          REG_BINARY    1D19B008D7BF654B84050FC7353F993C
MachineQuota   REG_DWORD     0x100000
MachineJournalQuota REG_DWORD     0xffffffff
LongLiveTime   REG_DWORD     0x54600
```

##### • 正規表現のパターン

メッセージ・ルーティングの有効化:

```
"\s*MQS_Routing\s+REG_DWORD\s+0x[0]*(\d)\s*"
```

メッセージ・ストレージの制限:

```
"\s*MachineQuota\s+REG_DWORD\s+(\w+)\s"
```

メッセージ・ジャーナルの制限:

```
"\s*MachineJournalQuota\s+REG_DWORD\s+(\w+)\s"
```

## レジストリ・ブランチ(2)

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\setup\

### ・ コマンド出力

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\setup
MachineDomain    REG_SZ    UCMDB-EX
MachineDomainFQDN REG_SZ    ucmdb-ex.dot
OSType           REG_DWORD 0x500
CreateMsmqObj    REG_DWORD 0x0
UserSid          REG_BINARY 10500000000000515000000576A62162631895
C45612C98F4010000
MachineDN        REG_SZ    CN=MSMQ-VM01,CN=Computers,DC=ucmdb-ex,DC=dot
JoinStatus       REG_DWORD 0x2
MSMQAddedToICFExceptionList REG_DWORD 0x1
MQDSSvcInstalled REG_DWORD 0x1
InetpubWebDir    REG_DWORD 0x1
```

### ・ 正規表現のパターン

マシンのドメイン名:

```
"\s*MachineDomainFQDN\s+REG_SZ\s+(\[w\-.]+\)\s"
```

## レジストリ・ブランチ(3)

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSMQ\Setup\

### ・ コマンド出力

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Setup
msmq_Core        REG_DWORD 0x1
msmq_LocalStorage REG_DWORD 0x1
msmq_ADIntegrated REG_DWORD 0x1
InstalledComponents REG_DWORD 0xf8000000
msmq_MQDSService REG_DWORD 0x1
msmq_TriggersService REG_DWORD 0x1
msmq_HTTPSupport REG_DWORD 0x1
msmq_RoutingSupport REG_DWORD 0x1
```

- **正規表現のパターン**

MsMQ はドメイン番号 :

```
"\s*msmq_ADIntegrated\s+REG_DWORD\s+0x[0]*(\d)\s"
```

トリガの有効化 :

```
"\s*msmq_TriggersService\s+REG_DWORD\s+0x[0]*(\d)\s"
```

## Microsoft Message Queue Topology by NTCMD or UDA ジョブ

このジョブは、トリガ、ルール、キューの設定と関係を検出します。

### MS MQ キュー・ディスカバリ

- レジストリ・ブランチ

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters /v StoreReliablePath
```

- コマンド出力

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters  
StoreReliablePath REG_SZ C:\WINDOWS\system32\msmq\storage
```

- 正規表現のパターン

メッセージ・ストレージの基本親フォルダ

```
"\s*StoreReliablePath\s+REG_SZ\s+(.+)"
```

- コマンド

```
dir /B /A:-D <ms mq queue settings folder>
```

- コマンド出力

```
dir /B /A:-D C:\WINDOWS\system32\msmq\storage\lqs  
00000002.990736e8  
00000003.6ab7c4b8  
00000004.4c1eb11b  
00000006.e2f46f06  
00000010.d1c14377  
00000012.e6d243aa  
9b0b035bf61b429d845bbd61740403b7.0d0d6ec1
```

- 結果

MS MQ キュー構成のファイル名が取得されます。次に DFM は、このファイル・リストを反復処理し、各ファイルを読み取り、キュー設定を解析します。

- コマンド

```
type <full_path_to_the_file>
```

## • コマンド出力

```
type C:\WINDOWS\system32\msmq\storage\lqs\00000002.990736e8
[Properties]
Label=private$\admin_queue$
Type=00000000-0000-0000-0000-000000000000
QueueName=\private$\admin_queue$
Journal=00
Quota=4294967295
Security=010007805c000000680000000000000014000000200
480003000000000018003f000e0001020000000000052000000020
0200000000140024000200010100000000000100000000000140
0040000000101000000000005070000000101000000000051200
0000010100000000000512000000
JournalQuota=4294967295
CreateTime=1259681363
BasePriority=32767
ModifyTime=1259681363
Authenticate=00
PrivLevel=1
Transaction=00
SystemQueue=01
Signature=DoronJ
```

## • 解析ルール

キュー名:

```
".*QueueName\s*=\s*(.+\?)\n.*"
```

トランザクションかどうか:

```
".*Transaction\s*=\s*(\d+).*"
```

キュー・タイプ(public/private):

```
"^[\\]*(private).*$" against Queue name
```

メッセージの制限:

```
".*\s+Quota\s*=\s*(\d+).*"
```

ジャーナルが有効かどうか:

```
".*Journal\s*=\s*(\d+).*"
```

ジャーナルの制限 :

".\*JournalQuota\s\*=\s\*(\d+).\*"

## MS MQ トリガ・ディスカバリ

### • レジストリ・ブランチ

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Triggers\

### • コマンド出力

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers
\Data\Triggers\31b8e2c4-f412-431e-9b2c-517f7e5031d7
  Name      REG_SZ      Test Trigger
  Queue     REG_SZ      msmq-vm2\Test Queue
  Enabled   REG_DWORD    0x1
  Serialized REG_DWORD    0x0
  MsgProcessingType REG_DWORD    0x1
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\31b8e2c4-f412-431e-9b2c-517f7e5031d7\AttachedRules
  Rule0     REG_SZ      9c172d69-c832-453e-826b-4415b7d0dfef
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\728b0d45-531d-4887-9762-3191b0069bb1
  Name      REG_SZ      remote Trigger
  Queue     REG_SZ      msmq-vm01\Test Queue
  Enabled   REG_DWORD    0x1
  Serialized REG_DWORD    0x0
  MsgProcessingType REG_DWORD    0x0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\728b0d45-531d-4887-9762-3191b0069bb1\AttachedRules
  Rule0     REG_SZ      9c172d69-c832-453e-826b-4415b7d0dfef
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\b900d598-e3c2-4958-bf21-c8c99ed264e2
  Name      REG_SZ      qqqqqqqq
  Queue     REG_SZ      msmq-vm2\private$\Private Test Queue
  Enabled   REG_DWORD    0x1
  Serialized REG_DWORD    0x0
  MsgProcessingType REG_DWORD    0x1
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\b900d598-e3c2-4958-bf21-c8c99ed264e2\AttachedRules
  Rule0     REG_SZ      9c172d69-c832-453e-826b-4415b7d0dfef
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\dc4302f0-d28c-40e4-a19a-492dcee231fe
  Name      REG_SZ      Test2
  Queue     REG_SZ      msmq-vm2\private$\Test Transactional
```



```

    Enabled    REG_DWORD    0x1
    Serialized REG_DWORD    0x1
    MsgProcessingType REG_DWORD    0x2
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\dc4302f0-d28c-40e4-a19a-492dcee231fe\AttachedRules
    Rule0     REG_SZ      9c172d69-c832-453e-826b-4415b7d0dfef
    Rule1     REG_SZ      2874c4c1-57f1-4672-bbdd-0c16f17788cf

```

## MS MQ ルール・ディスカバリ

### • 正規表現のパターン

出力バッファは次の正規表現で分割されます。

```
"(HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\
Triggers\Data\Triggers\[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\
-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12})\s*\n"
```

文字列バッファが分割されるたびに、次のパターンが適用されます。

トリガ名 :

```
".*Name\s+REG_SZ\s+(.*?)\n.*"
```

トリガ GUID :

```
" HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\
Data\Triggers\[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\
-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12})\s*\n"
```

割り当てられたキュー :

```
".*Queue\s+REG_SZ\s+(.*?)\n.*"
```

トリガのシリアル化 :

```
".*Serialized\s+REG_DWORD\s+(0x\d+).*"

```

トリガの有効化 :

```
".*Enabled\s+REG_DWORD\s+(0x\d+).*"

```

トリガ・メッセージの処理タイプ :

```
".*MsgProcessingType\s+REG_DWORD\s+(0x\d+).*"

```

トリガによって割り当てられたルール GUID :

```
".*Rule\d+\s+REG_SZ\s+([0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}).*"

```

## • レジストリ・プランチ

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\

## • コマンド出力

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\
2874c4c1-57f1-4672-bbdd-0c16f17788cf
Name      REG_SZ      Test Rule2
Description  REG_SZ      bla bla
ImplementationProgID  REG_SZ      MSQMTriggerObjects.MSMQRuleHandler
Condition   REG_SZ      $MSG_PRIORITY_EQUALS=1
            $MSG_LABEL_DOES_NOT_CONTAIN=bla
Action      REG_SZ      EXE      C:\WINDOWS\system32\calc.exe
ShowWindow  REG_DWORD    0x1
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\
9c172d69-c832-453e-826b-4415b7d0dfe
Name      REG_SZ      Test Rule
Description  REG_SZ
ImplementationProgID  REG_SZ      MSQMTriggerObjects.MSMQRuleHandler
Condition   REG_SZ      $MSG_LABEL_CONTAINS=Test
Action      REG_SZ      EXE      C:\WINDOWS\notepad.exe
ShowWindow  REG_DWORD    0x1

```

## • 正規表現のパターン

出力バッファは次の定数で分割されます。

```
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Triggers\Data\Rules\"

```

文字列バッファが分割されるたびに、次のパターンが適用されます。

ルール名 :

```
".*Name\s+REG_SZ\s+(.*?)\n.*"

```

ルール条件 :

```
".*Condition\s+REG_SZ\s+(.*?)\n.*"

```

ルール操作 :

```
".*Action\s+REG_SZ\s+(.*?)\n.*"

```

## ルール GUID

```
"\s*([0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}).*"
```

## Microsoft Message Queue Topology by LDAP ジョブ

このジョブは、MS MQ デプロイメントの Active Directory に関する部分 (AD フォレスト、AD サイト、MS MQ マネージャ、MS MQ ルーティング・リンク) をレポートします。

スキーマのパラメータ:

CN=Configuration,DC=<domain\_name>,DC=<domain\_suffix>

サイト・ディスカバリ (AD ディスカバリから派生):

CN=Sites,CN=Configuration,<domain\_name>,DC=<domain\_suffix>

### MS MQ マネージャを使用したサーバ・ディスカバリ

- ブランチ

CN=Servers,CN=<site\_name>,CN=Sites,CN=Configuration,DC=<domain\_name>,DC=<domain\_suffix>

- 値

サーバ名のプロパティ:

'name'

サーバの完全識別名:

'distinguishedName'

基本ブランチが存在する場合 (objectClass=mSMQSettings), サーバは MS MQ マネージャを含むものとみなされます。

## 第58章: TIBCO BusinessWorks および EMS の ディスカバリ

### 本章の内容

概要 .....	882
ディスカバリ・メカニズム .....	882
サポートされるバージョン .....	882
トポロジ .....	883
TIBCO BusinessWorks および EMS の検出方法 .....	884
TIBCO BusinessWorks by Shell ジョブ .....	886
TIBCO EMS by Shell ジョブ .....	888

## 概要

**TIBCO Enterprise Message Service (EMS)**は、共通のエンタープライズ・バックボーン上にある各種の IT リソースを集約し、情報フローをリアルタイムで管理するメッセージング・プラットフォームです。

**TIBCO ActiveMatrix BusinessWorks (BusinessWorks)**は、サービスを作成、調整、統合するための製品で、オープン・スタンダードを全面的に使用して構築されています。

TIBCO ディスカバリ・プロセスでは、トポロジ全体を検出できます。

## ディスカバリ・メカニズム

TIBCO には、アプリケーションに関するシステム構成ファイルがありません。そのため、TIBCO ディスカバリ・メカニズムは、最初に TIBCO の **AppManage** ユーティリティを使用して、xml ファイルのリストを BusinessWorks サーバの一時フォルダにエクスポートします。次に、TIBCO の **TibcoEmsAdmin** ユーティリティを使用して、EMS および JMS トポロジに関する情報を取得します。

ディスカバリ・メカニズムは引き続き、**TIBCO BusinessWorks by Shell** ジョブと **TIBCO EMS by Shell** ジョブを実行します。

## サポートされるバージョン

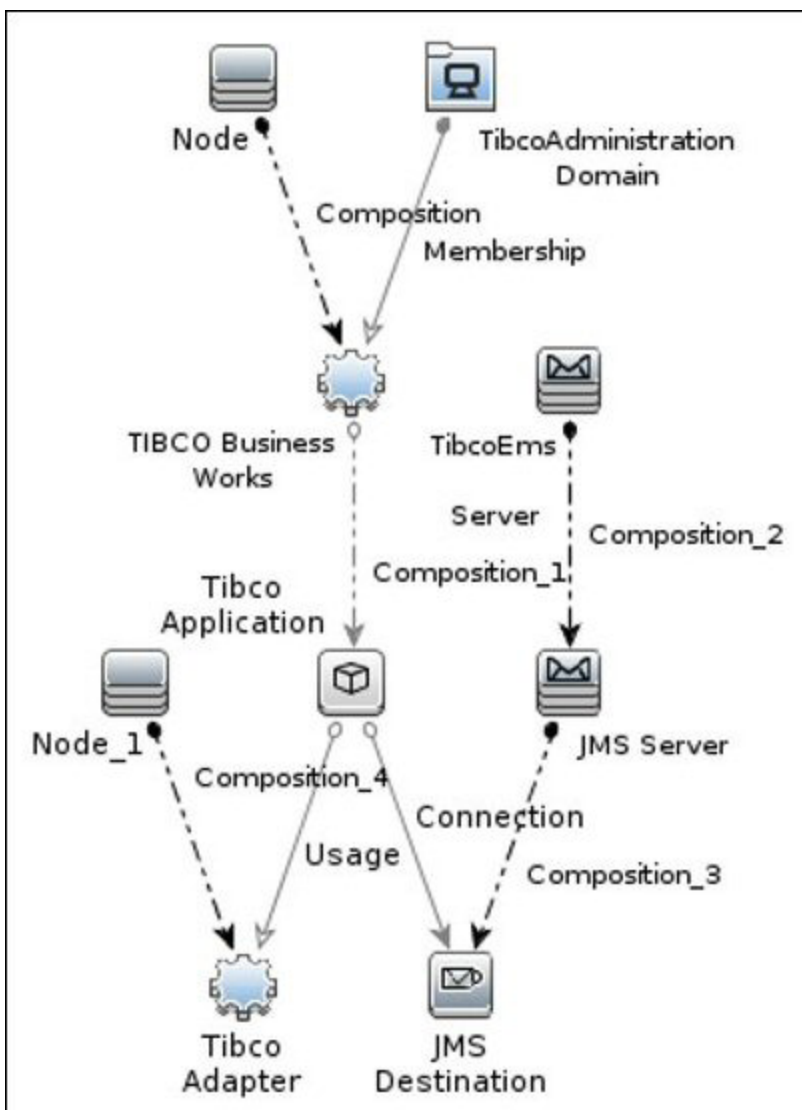
TIBCO ディスカバリでは、UNIX 環境で稼働している次のバージョンのソフトウェアがサポートされています。

- EMS バージョン 6.0
- BusinessWorks バージョン 5.7 および 5.8

## トポロジ

次の図は、BusinessWorks のトポロジを示しています。

注: 検出された CIT のリストについては、[「TIBCO BusinessWorks by Shell ジョブ」\(886ページ\)](#)を参照してください。



## TIBCO BusinessWorks および EMS の検出方法

### 本項の内容

#### 1. 前提条件 - プロトコル資格情報の設定

シェル・プロトコル(SSH または Telnet)と TIBCO プロトコルを設定する必要があります。

- シェル・プロトコル: SSH, Telnet.

ユーザ名, パスワード, ドメイン名を準備します。

- TIBCO プロトコル

ユーザ名とパスワードを準備します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

#### 2. 前提条件 - その他

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by Shell** ジョブを実行して、ターゲット・ホストとそれに対するシェル接続を検出します。
- c. TIBCO BusinessWorks ソフトウェアとエージェント・プロセスを含む、ターゲット・ホストのアプリケーションを検出するには、**Host Applications by Shell** ジョブを実行します。

**注:** **discoverProcesses** 属性を有効にする必要があります。この属性により、**TIBCO EMS by Shell** ジョブがトリガする **Process CI** が検出されます。

- d. 次の両方を所有していることを確認します。
  - i. TIBCO BusinessWorks サーバの一時フォルダに対する読み取りアクセス権と書き込みアクセス権。標準設定のフォルダは **/tmp** です。
  - ii. **TIBCO ランタイム・アシスタント (TRA) AppManage** ユーティリティを実行するためのアクセス権。

#### 3. ディスカバリの実行



- a. ターゲットの BusinessWorks サーバのトポロジを検出するには、**TIBCO BusinessWorks by Shell** ジョブを実行します。
- b. ターゲットの EMS サーバのトポロジを検出するには、**TIBCO EMS by Shell** ジョブを実行します。

## TIBCO BusinessWorks by Shell ジョブ

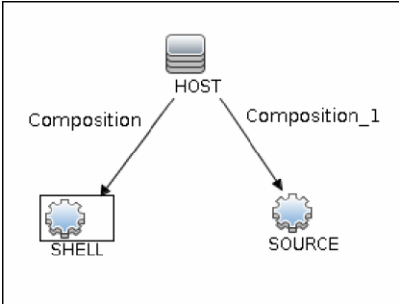
本項ではこのジョブに関する詳細を説明します。

### 入力 CIT

TibcoBusinessWorks

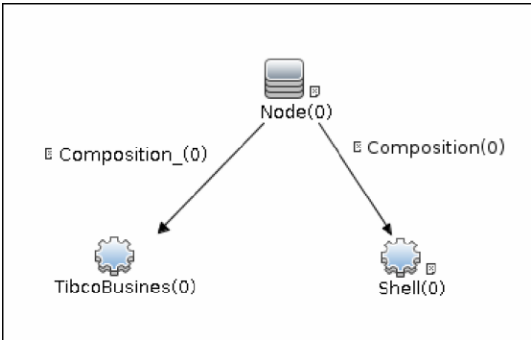
### 入力 TQL クエリ

次の図は、このジョブの入力 TQL クエリを示しています。



### トリガ TQL クエリ

次の図は、このジョブのトリガ TQL クエリを示しています。



### トリガ CI データ

名前	値
Protocol	\${SHELL.root_class}

名 前	値
<b>bwId</b>	\${SOURCE.root_id}
<b>bwPath</b>	\${SOURCE.application_path}
<b>credentialsId</b>	\${SHELL.credentials_id}
<b>hostId</b>	\${HOST.root_id}
<b>ip_address</b>	\${SHELL.application_ip}

## 使用スクリプト

- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- iteratortools.py
- jdb\_url\_parser.py
- jdbc.py
- jee.py
- jms.py
- jmx.py
- tibco.py
- tibco\_businessworks\_by\_shell.py
- tibco\_discoverer.py

## 検出された CIT

- Composition

- Connection
- Containment
- IpAddress
- IpServiceEndpoint
- JMS Desination
- JMS Server
- Membership
- Node
- TibcoAdapter
- TibcoAdministrationDomain
- TibcoApplication
- TibcoBusinessWorks
- TibcoEmsServer
- Usage

### アダプタ・パラメータ

パラメータ	説明
<b>temp_directory</b>	ディスカバリ・プロセスで作成されたファイルが保存される TIBCO BusinessWorks サーバの一時ディレクトリ。
<b>discover_jms_topology</b>	JMS トポロジを検出するかどうかを設定する。 標準設定 : false。

## TIBCO EMS by Shell ジョブ

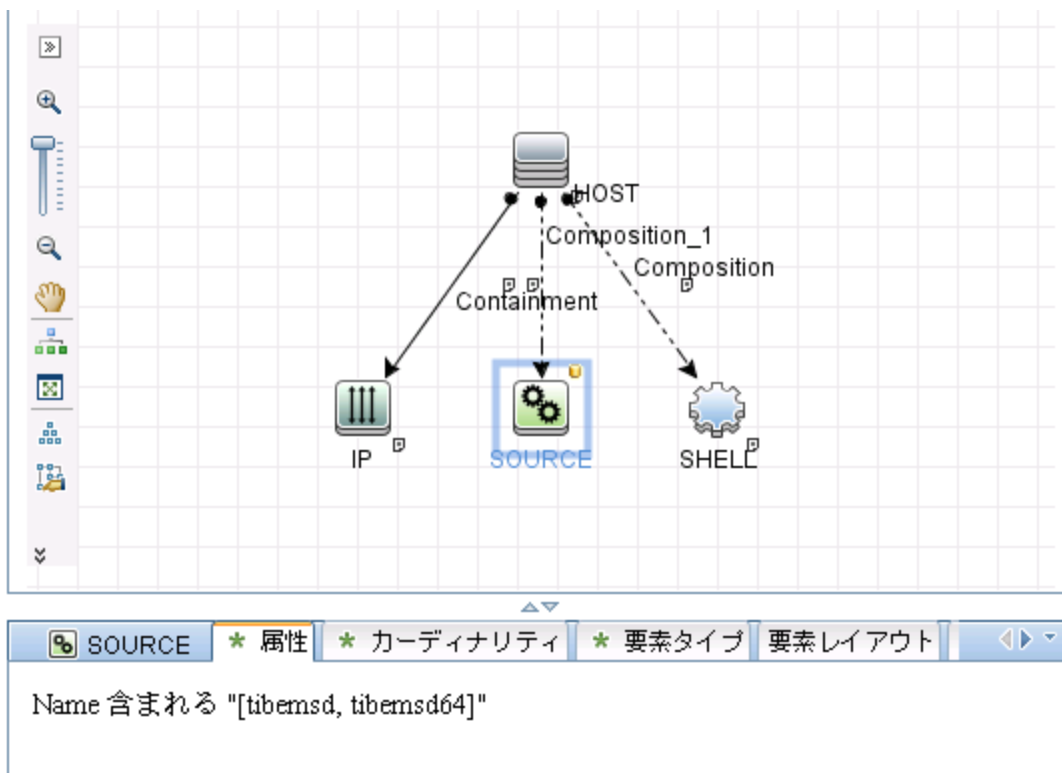
本項ではこのジョブに関する詳細を説明します。

## 入力 CIT

Process

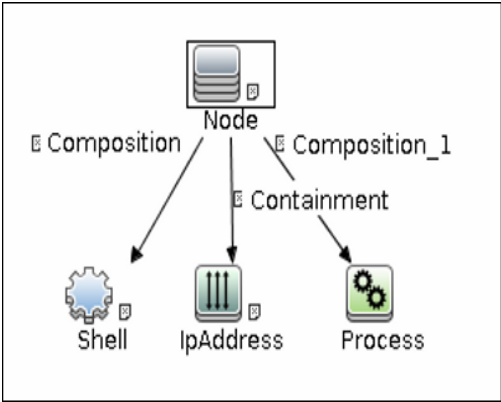
## 入力 TQL クエリ

次の図は、このジョブの入力 TQL クエリを示しています。



## トリガ TQL クエリ

次の図は、このジョブのトリガ TQL クエリを示しています。



トリガ CI データ

名前	値
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SHELL.application_ip}
processCmdLine	\${SOURCE.process_cmdline}
processPath	\${SOURCE.process_path}
processRootId	\${SOURCE.root_id}

使用スクリプト

- db.py
- db\_builder.py
- db\_platform.py
- entity.py
- iteratortools.py
- jdbc\_url\_parser.py
- jdbc.py

- jee.py
- jms.py
- jmx.py
- tibco.py
- tibco\_discoverer.py
- tibco\_ems\_by\_shell.py

検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- JMS Destination
- JMS Server
- Node
- Process
- TibcoEmsServer
- Usage

アダプタ・パラメータ

パラメータ	説明
discover_queues	キューに関する情報を検出するかどうかを設定する。  標準設定 : false。
discover_topics	トピックに関する情報を検出するかどうかを設定す

パラメータ	説明
	る。 標準設定 : false。
emsadmin_tool_absolute_paths	プロセスから取得したパスが相対パスである場合の、emasadmin ツールの絶対パスのリスト(パスはカンマ区切り)。



## 第59章: WebSphere MQ ディスカバリ

### 本章の内容

概要 .....	894
サポートされるバージョン .....	894
トポロジ .....	894
MQ Queue Dependency .....	895
MQ Q Manager Resources on Non-Local Cluster .....	896
MQ Namelist Membership .....	897
MQ Cluster Membership .....	897
MQ Channel Communication .....	898
MQ Alias Queue Managers .....	898
MQ Topology .....	899
WebSphere MQ の検出方法 .....	901
ディスカバリ・メカニズム .....	902
アダプタ .....	903
アダプタ・パラメータ .....	903
検出された CIT .....	904
関係 .....	906
トラブルシューティングおよび制限事項 .....	910

## 概要

WebSphere MQ パッケージにより、組織内の WebSphere MQ インフラストラクチャのさまざまなコンポーネントをマッピングできます。最終目的は、組織内のほかのアプリケーションまたはサービスとの相互依存関係をモデル化し、メッセージング・サイロ全体のエンド・ツー・エンドの影響分析を可能にすることです。

メッセージ・キューイングとは、異なるソフトウェア・サービスがターゲット・サービスの知識を必要とせずに通信できるようにするミドルウェア技術です。ターゲット・システムの現在の可用性や2つのシステムを接続するインフラストラクチャの複雑さに関係なく信頼できる通信を実現できます。

メッセージには単純な文字データ、数値データ、複雑なバイナリ・データ、情報に対する要求、コマンド、またはこれらすべての組み合わせが含まれる場合があります。メッセージング・インフラストラクチャは、ソースからターゲットへのメッセージの信頼できる透過的な転送を管理し、そのコンテンツを理解または認識する必要がありません。

## サポートされるバージョン

- **ターゲット・プラットフォーム** : IBM WebSphere MQ
- **ターゲット・プラットフォーム・バージョン** : 5.x, 6.x, 7.1
- **ターゲット・プラットフォーム OS** : Microsoft Windows, Solaris, Linux, AIX

## トポロジ

WebSphere MQ パッケージには、MQ インフラストラクチャの詳細をモデル化する次のビューが含まれています。各ビューには、同じクエリ設定の対応するレポートがあります。

### 注:

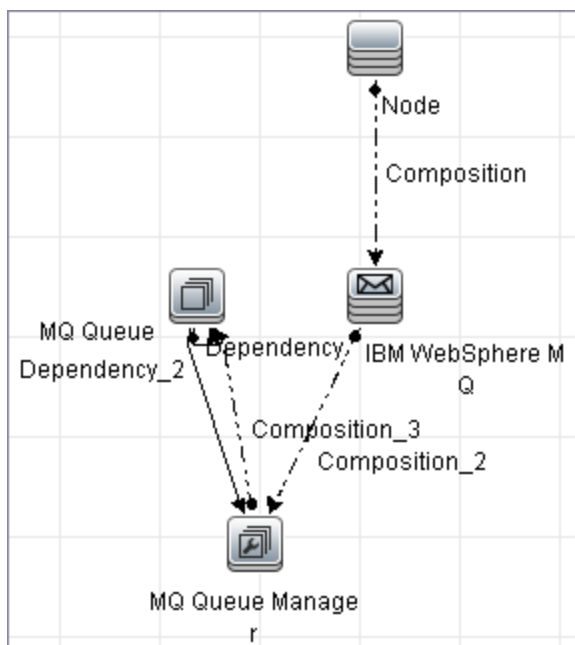
- これらの定義済みのビューは、例としてのみ提供されます。できれば独自のビューを定義してください。
- 検出された CIT のリストについては、[「検出された CIT」\(904ページ\)](#)を参照してください。

本項では、次のビューについて説明します。

- 「MQ Queue Dependency」(895ページ)
- 「MQ Q Manager Resources on Non-Local Cluster」(896ページ)
- 「MQ Namelist Membership」(897ページ)
- 「MQ Cluster Membership」(897ページ)
- 「MQ Channel Communication」(898ページ)
- 「MQ Alias Queue Managers」(898ページ)
- 「MQ Topology」(899ページ)

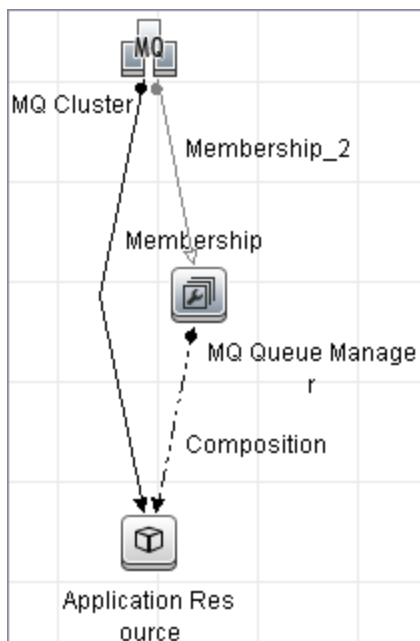
## MQ Queue Dependency

このビューにはほかの MQ オブジェクトに依存するキューが表示され、通常は Remote Queue, Alias Queue, および Remote Queue Manager が含まれています。



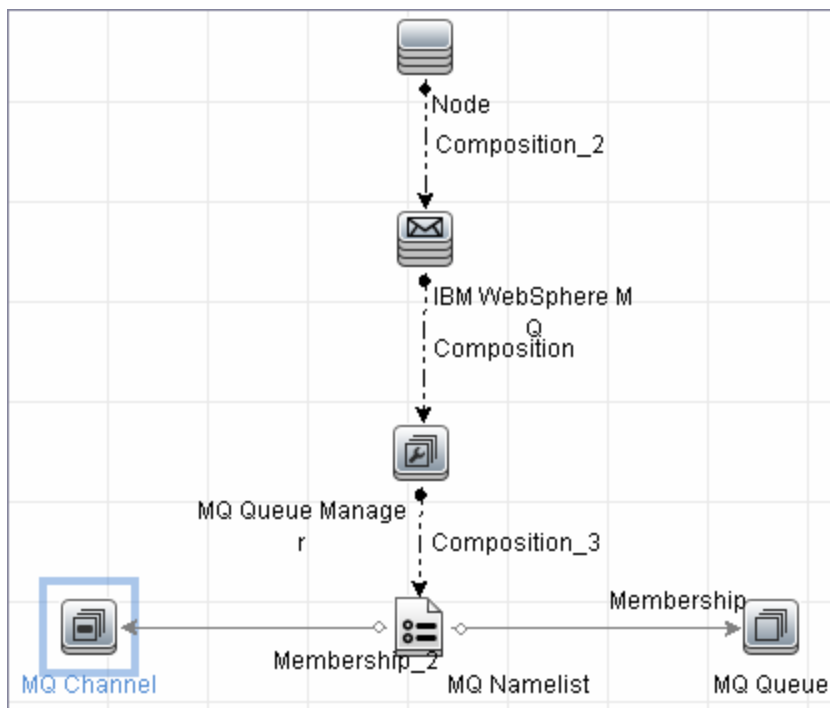
## MQ Q Manager Resources on Non-Local Cluster

このビューには、Queue Manager によって管理され、Queue Manager がメンバでない MQ Cluster に属している MQ オブジェクトが表示されます。このビューの MQ オブジェクトには設定ミスが含まれている可能性があり、このビューの目的は、このような設定ミスのあるオブジェクトを識別することです。



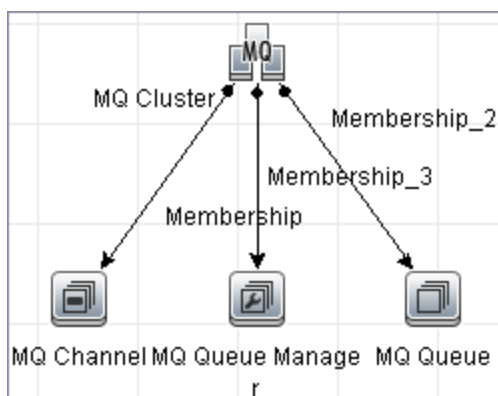
## MQ Namelist Membership

このビューには、名前リストとそのメンバが表示されます。



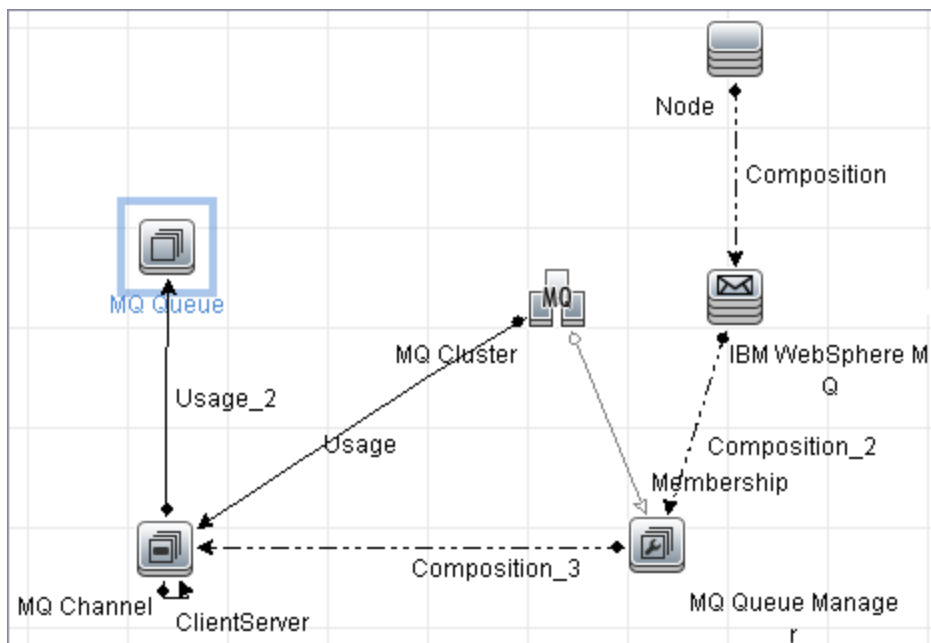
## MQ Cluster Membership

このビューには、クラスターとそのメンバが表示されます。



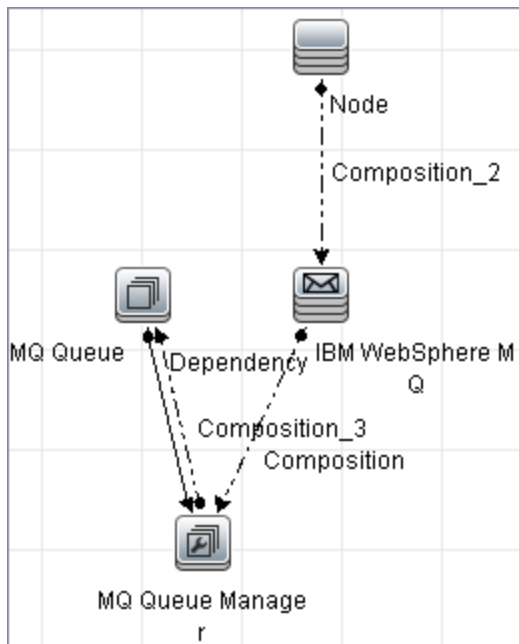
## MQ Channel Communication

このビューには、MQ Channel とチャネルによって使用されるキュー間のクライアント-サーバ通信が表示されます。



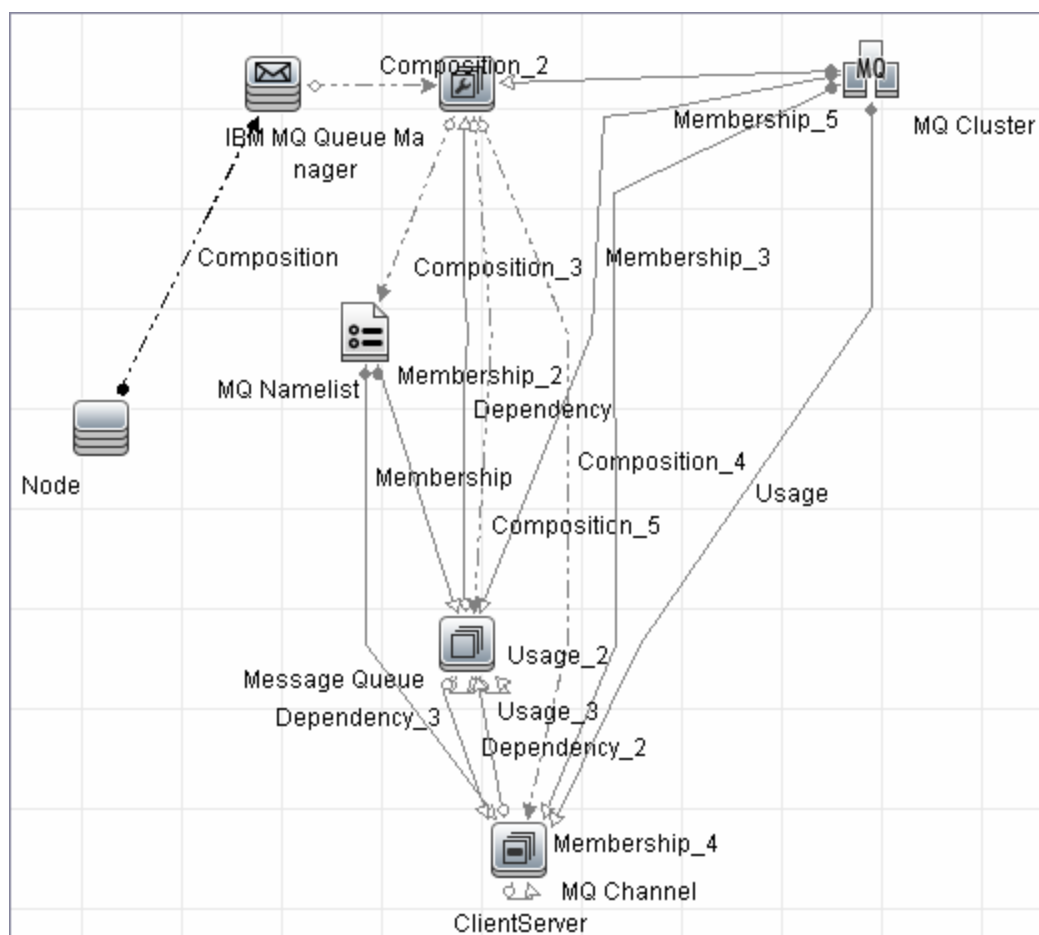
## MQ Alias Queue Managers

このビューには、リモート Queue Manager となるキューが表示されます。



## MQ Topology

このビューには、関係および相互依存関係を含めた MQ インフラストラクチャ内のすべての MQ オブジェクトが表示されます。





## WebSphere MQ の検出方法

このタスクでは、WebSphere MQ ジョブによる WebSphere MQ コンポーネントの検出方法を次の手順で説明します。

### 1. 前提条件 - プロトコル資格情報の設定

このデイスカバリでは、SSH、Telnet、または NTCMD プロトコルを使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

Shell コマンドは次のようになります(**sudo** は任意です)。

- **dspmqr** または **mqr**
- **dsmpr**
- **runmqsc** または **runmqadm -r**

### 2. 前提条件 - IP アドレス

すべての WebSphere MQ サーバの IP アドレスが Data Flow Probe の範囲内にあることを確認します。詳細については、『HP Universal CMDB Data Flow Management Guide』の、Probe 範囲の追加方法を説明している項を参照してください。

### 3. デイスカバリの実行

- a. 必要に応じて **MQ by Shell** ジョブのパラメータを設定します。詳細については、『HP Universal CMDB Data Flow Management Guide』の「Data Flow Probe Setup」を参照してください。
- b. WebSphere MQ デイスカバリをトリガするために必要な情報を収集するには、次のジョブを実行します。
  - **Range IPs by ICMP** : WebSphere MQ サーバの IP アドレスを検出します。
  - **Host Connection by Shell** : WebSphere MQ サーバ上のオペレーティング・システム情報を検出します。
  - **Host Applications by Shell** : サーバ上の WebSphere MQ のインスタンスを検出します。
  - **MQ by Shell** : WebSphere MQ インフラストラクチャを検出します。

## ディスカバリ・メカニズム

WebSphere MQ は複数の UNIX プラットフォームおよび Microsoft Windows にインストールでき、プラットフォーム間で標準化されたコマンド・ライン・インタフェースを使用して管理されます。コマンド・ライン・インタフェースは、WebSphere MQ インストールに含まれている **runqsc** または **runmqadm** のプログラムを使用してアクセスできます。

**MQ by Shell** ジョブはトリガとしてサーバに関連付けられた **Shell CI** を使用します。CMDB のすべてのサーバには **Shell CI** が関連付けられているため、トリガ・クエリの結果には WebSphere MQ ソフトウェアがインストールされているサーバの **Shell CI** のみが含まれます。

**MQ by Shell** ジョブは、WebSphere MQ コマンド・ライン・インタフェースを使用して MQ オブジェクトとその詳細をクエリします。**runmqsc** コマンドには管理者またはルート権限が必要であり、**runmqadm** コマンドは常に使用できるわけではないため、このジョブは最初に **runmqadm -r** コマンドを試みます。

**runmqadm** が失敗した場合、このジョブは **runmqsc** コマンドを試みます。

**Shell CI(Host Connection by Shell** ジョブによって作成)を使用して MQ サーバにログインした後で、DFM は次のことを行います。

1. サーバにインストールされた WebSphere MQ のバージョンを識別します。これは、**dspmqrver** コマンドを使用して実行されます(**dspmqrver** が失敗した場合、**mqrver** コマンドが試みられます)。
2. **dspmqr** コマンドを使用して WebSphere MQ Queue Manager のリストを取得します。
3. MQ CLI(コマンド・ライン・インタフェース)コマンドを使用して各 Queue Manager の詳細を取得します。

```
DISPLAY QMGR DESCR DEADQ DEFXMITQ REPOS CCSID
```

4. MQ CLI コマンドを使用して各 Queue Manager のキューのリストを取得します。

```
DISPLAY QUEUE(*) TYPE DESCR CLUSTER CLUSNL USAGE RNAME RQMNAME XMITQ TARGQ  
DEFTYPE
```

キューと、ほかのキュー、Queue Manager などのほかの MQ オブジェクト間の関係がオンザフライで構築されます。

5. 転送キューに関連付けられた送信チャネルを使用して、(見つかった各 TRANSMIT Queue について) リモート・サーバ名 および IP およびポートを取得します。これは、MQ CLI コマンドを使用して実行されます。

```
DISPLAY CHANNEL(*) WHERE(xmitq EQ <transmitQueueName>) TYPE(SDR) CONNAME
```

6. MQ CLI コマンドを使用して各 Queue Manager のチャンネルのリストを取得します。

```
DISPLAY CHANNEL(*) CHLTYPE TRPTYPE DESCR CLUSTER CLUSNL CONNAME XMITQ
```

キューと、ほかのキュー、チャンネルなどのほかの MQ オブジェクト間の関係がオンザフライで構築されます。

7. MQ CLI コマンドを使用して各 Queue Manager がメンバである、または把握しているクラスタのリストを取得します。

```
DISPLAY CLUSQMGR(*) CONNAME QMTYPE
```

クラスタとほかのクラスタ間の関係がオンザフライで構築されます。

8. MQ CLI コマンドを使用して各 Queue Manager がメンバである、または把握している namelists を取得します。

```
DISPLAY NAMELIST(*) NAMES NAMCOUNT DESCR
```

## アダプタ

このディスカバリでは **WebSphere MQ Topology by shell** アダプタを使用します。

## アダプタ・パラメータ

パラメータ	説明
<b>discover_dynamic_queues</b>	動的キュー(アプリケーションによってオンザフライで作成および破棄されるキュー)のディスカバリを有効にする。
<b>discover_remote_hosts</b>	検出している MQ サーバによって参照されるリモート・サーバおよび MQ オブジェクトの解決およびディスカバリを有効にする。 <b>false</b> に設定すると、異なるサーバ上の MQ オブジェクト間の関係が検出されない。
<b>mq_cmd_timeout</b>	MQ CLI コマンドのコマンド・タイムアウトを設定する。
<b>mqver_path</b>	<b>mqver</b> または <b>dspmqver</b> 実行可能ファイルへのパス。複数のエントリはカンマ(;)で区切る。
<b>sudo_command</b>	<b>use_sudo</b> パラメータを <b>true</b> に設定した場合は設定する必要がある。このパラメータのエントリは MQ コマンド・ライン・インタフェース・プログラムの先頭に付けられる。このパラメータは通常、MQ ユーザ名を設定するために使用される。たとえば、このパラメータを <b>sudo -u mqm</b> に設定すると、 <b>runmqsc</b> コマンドが <b>sudo -u mqm runmqsc</b> として呼び出される。

パラメータ	説明
use_sudo	sudo の使用を可能にするには <b>true</b> に設定する。

## 検出された CIT

WebSphere MQ ディスカバリは次の CI タイプを検出します。

注: トポロジを表示するには、[「トポロジ」\(894ページ\)](#)を参照してください。

CI タイプ	キー属性	説明
Composition		
Containment		
Dependency		
IBM MQ Channel (mqchannel)  親 : Message Queue Resource	<ul style="list-style-type: none"> <li>名前</li> <li>コンテナ : IBM MQ Queue Manager</li> </ul>	この抽象 CI タイプは MQ Channel を表す。MQ Channel は、Queue Manager がほかの Queue Manager と通信するために必要。チャネルには単方向および双方向の通信 (要求応答システムなど) があり、データを返すためにはほかのチャネルが必要。チャネルは TCP/IP ネットワークの特定のポートでデータを送受信する。
IBM MQ Cluster (mqcluster)  親 : Failover Cluster	名前	MQ Queue Manager Cluster を表す。MQ Cluster により、最小限の設定で複数の Queue Manager に柔軟に参加できる。これにより、複数の Queue Manager を使用して同じサービスの複数のインスタンスをホストすることができ、結果としてパフォーマンス、容量、柔軟性が高くなる。キュー・マネージャは動的にクラスタに参加または離脱できる。
IBM MQ Namelist (mqnamelist)  親 : Message Queue Resource	<ul style="list-style-type: none"> <li>名前</li> <li>コンテナ : IBM MQ Queue Manager</li> </ul>	MQ Namelist を表す。MQ 名前リストには名前のリストが含まれ、通常は MQ Queue Manager Cluster のリストを含むために使用される。次に、これらの名前リストはクラスタ名前リスト・プロパティで指定され、そのクラスタ内のすべての Queue Manager によって検索のために使用される場合がある。
IBM MQ Queue (mqqueue)	<ul style="list-style-type: none"> <li>名前</li> <li>コンテナ : IBM MQ</li> </ul>	キューとは MQ インフラストラクチャ内のメッセージのコンテナであり、メッセージが MQ インフラストラク

CI タイプ	キー属性	説明
親 : MQ Queue	Queue Manager	チャ内の Queue Manager 間でルーティングされる方法を制御する。キューは、メッセージの順序および配信 (F/LIFO, メッセージの優先度, 連続配信, 保証された配信など)を制御するために複数の構成で設定でき、少量の情報を伝えるために最適化される。
IBM MQ Queue Manager(mqqueue) 親 : Message Queue Resource	<ul style="list-style-type: none"> <li>名前</li> <li>コンテナ : IBM WebSphere MQ CI</li> </ul>	MQ Queue Managerを表す。WebSphere MQ インスタンスには 1 つ以上の Queue Manager を含むことができる。Queue Manager は、保存、タイミング、トリガなど、データの移動に直接は関係しない関数を管理する。キュー・マネージャは、管理する MQ オブジェクトおよびネットワークを介してリモート・クライアントと通信するために <b>バインディング</b> 接続として知られる特許取得の IBM 技術を使用する。
IBM MQ Receiver Channel (mqreceiverchannel) 親 : IBM MQ Channel	<ul style="list-style-type: none"> <li>名前</li> <li>コンテナ : IBM MQ Queue Manager</li> </ul>	受信チャネルは、同じ名前の送信チャネルを通じてリモート Queue Manager からメッセージを受信する。
IBM MQ Sender Channel (mqreceiverchannel) 親 : IBM MQ Channel	<ul style="list-style-type: none"> <li>名前</li> <li>コンテナ : IBM MQ Queue Manager</li> </ul>	送信チャネルは同じ親 Queue Manager 内の特定の転送キューに関連付けられていて、十分に定義された宛先を持つ。
IBMWebSphereMQ (webspheremq) 親 : メッセージ・キューイング・ソフトウェア	<ul style="list-style-type: none"> <li>名前 : 常に <b>IBM WebSphere MQ</b></li> <li>コンテナ : Node</li> </ul>	サーバにインストールされた WebSphere MQ ソフトウェアのインスタンスを表す。
IpAddress		
IpServiceEndpoint		
Membership		
Node		
Usage		

## 関係

WebSphere MQ ディスカバリには、次の関係が含まれています。

リンク	End1	End2	カーディナリティ	説明
Client Server	IBM MQ Send Channel	IBM MQ Receive Channel	1..*	MQ Channel 間のメッセージ・フローの方向を表す
Realization	IBM MQ Remote Queue	IBM MQ Queue	1..*	MQ Remote Queue とそれがリモートとなるほかのキュー間の強い依存関係を示す。これは、キューのタイプが不明な状況で使用される。
Realization	IBM MQ Remote Queue	IBM MQ Local Queue	1..*	MQ Remote Queue とそれがリモートとなる Local Queue 間の強い依存関係を示す。
Realization	IBM MQ Remote Queue	IBM MQ Alias Queue	1..*	MQ Remote Queue とそれがリモートとなる Alias Queue 間の強い依存関係を示す。
Realization	IBM MQ Remote Queue	IBM MQ Remote Queue	1..*	MQ Remote Queue とそれがリモートとなる Remote Queue 間の強い依存関係を示す。
Realization	IBM MQ Alias Queue	IBM MQ Queue	1..*	MQ Alias Queue とそれがエイリアスとなるほかのキュー間の強い依存関係を示す。これは、キューのタイプ

リンク	End1	End2	カーディナリティ	説明
				が不明な状況で使用される。
Realization	IBM MQ Alias Queue	IBM MQ Local Queue	1..*	MQ Alias Queue とそれがエイリアスとなる Local Queue 間の強い依存関係を示す。
Realization	IBM MQ Alias Queue	IBM MQ Remote Queue	1..*	MQ Alias Queue とそれがエイリアスとなる Remote Queue 間の強い依存関係を示す。
Realization	IBM MQ Alias Queue	IBM MQ Alias Queue	1..*	MQ Alias Queue とそれがエイリアスとなる Alias Queue 間の強い依存関係を示す。
Realization	IBM MQ Remote Queue	IBM MQ Queue Manager	1..*	タイプがリモート・キューのキュー (Remote Queue Manager) とそれが表す Queue Manager を関連付ける。これは、(リモート・キューではなく) Queue Manager のリモートとなる専用の Remote Queue。Queue Manager の QM1 および QM2 では、QM2 のリモートである RQM2 という名前の QM1 で Remote Queue を設定できる。RQM2 に発行される MQ コマンドは実行のために

リンク	End1	End2	カーディナリティ	説明
				QM2 に渡される。
Membership	IBM MQ Cluster	IBM MQ Queue Manager	1..*	MQ Queue Manager が MQ Queue Manager Cluster のメンバであることを示す。MQ Queue Manager がクラスタの full repository である場合、この関係の名前は <b>Repository</b> に設定される。
Membership	IBM MQ Cluster	IBM MQ Channel	1..*	MQ Channel が MQ Queue Manager Cluster のメンバであることを示す。キューまたはチャネルが Queue Manager で定義されている場合、このキューがどの MQ クラスタのメンバであるかを指定できる(ただし必須ではない)。これは、たとえば、キューがクラスタのメンバであるが Queue Manager がそのクラスタのメンバでない場合のように、非常に具体的な構成が必要な場合に役立つ。このリンクは、こうした特別な構成を識別するために使用される。
Membership	IBM MQ Cluster	IBM MQ Queue	1..*	MQ Queue が MQ Queue Manager Cluster のメンバで



リンク	End1	End2	カーディナリティ	説明
				あることを示す。このリンクは前の行と同じ理由で追加されている。
Membership	IBM MQ Namelist	IBM MQ Channel	1..*	MQ Channel でその CLUSNL パラメータに MQ Namelist の名前が含まれていることを示す。
Membership	IBM MQ Namelist	IBM MQ Queue	1..*	MQ Queue でその CLUSNL パラメータに MQ Namelist の名前が含まれていることを示す。
Usage	IBM MQ Cluster	IBM MQ Channel	1..*	MQ Queue Manager Cluster がほかのクラスタと通信するために使用する MQ Channel(タイプは Cluster Sender Channel または Cluster Receiver Channel)を示す。この関係は、タイプが Cluster Sender Channel および Cluster Receiver Channel の MQ Channel に固有。これらのチャネルはクラスタ間通信専用であり、キューまたはほかの MQ オブジェクトによって使用されない。
Usage	IBM MQ Remote Queue	IBM MQ Transmit Queue	1..*	通信に転送キューを使用するリモート・キューを示す。
Usage	IBM MQ Transmit	IBM MQ Sender	1..*	通信に Sender

リンク	End1	End2	カーディナリティ	説明
	Queue	Channel		チャネルを使用する送信 Transmission Queue を示す。

## トラブルシューティングおよび制限事項

- ログ・ファイルに DNS 解決エラーがあり、デイスカバリの完了までに異常なほど長い時間がかかる場合、**discovery\_remote\_hosts** パラメータを **false** に設定してみてください。詳細については、「[アダプタ・パラメータ](#)」(903ページ)を参照してください。
- デイスカバリの結果表示が不完全な場合、**mq\_cmd\_timeout** パラメータの値を増やしてみてください。詳細については、「[アダプタ・パラメータ](#)」(903ページ)を参照してください。

## 第11部: Middleware > Web Servers

## 第60章: 基本 Web サーバ・ディスカバリ

### 本章の内容

概要 .....	913
サポートされるバージョン .....	913
基本 Web サーバの検出方法 .....	913
Web Server by Shell ジョブ .....	914
はじめに .....	914
トポロジ・マップ .....	915
サポートされているポリシー .....	915
アダプタ情報 .....	915
Web Server Detection using TCP Ports ジョブ .....	918
はじめに .....	918
トポロジ・マップ .....	919
サポートされているポリシー .....	919
アダプタ情報 .....	920
WebSphere to Web Server Dependency ジョブ .....	921
はじめに .....	921
トポロジ・マップ .....	923
サポートされているポリシー .....	923
アダプタ情報 .....	923

## 概要

基本 Web サーバ・ディスカバリでは、Apache Tomcat を除く Web サーバを検出できるとともに、Web サーバと WebSphere Application Server 間の依存関係を検出できます。

## サポートされるバージョン

このディスカバリは、次のバージョンをサポートしています。

ベンダ	製品	サポートされるバージョン
IBM	IBM HTTP Server	5, 6.1, 7, 8.0, 8.5, 8.5.5
Apache	Apache HTTP Server	1.3, 2.0, 2.2, 2.4
Microsoft	IIS	5, 6, 7, 7.5, 8, 8.5
IBM	WebSphere Application Server	5, 6.1, 7, 8.0, 8.5, 8.5.5

## 基本 Web サーバの検出方法

本項では基本 Web サーバの検出方法について説明します。

### 前提条件

- 次のプロトコルを設定します。
  - **SSH プロトコル**
  - **NTCMD プロトコル**
  - **Telnet プロトコル**

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### ディスカバリの実行

基本 Web サーバのディスカバリを実行するには、次のジョブを実行します。

- 1. Web Server by Shell
- 2. Web Server Detection using TCP Ports
- 3. WebSphere to Web Server Dependency

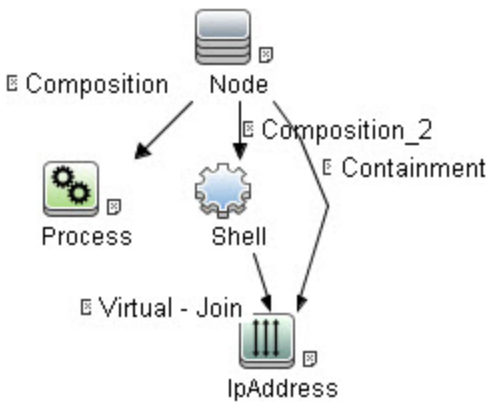
## Web Server by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

### はじめに

このジョブは Shell プロトコルを使用して Apache Web サーバを検出します。

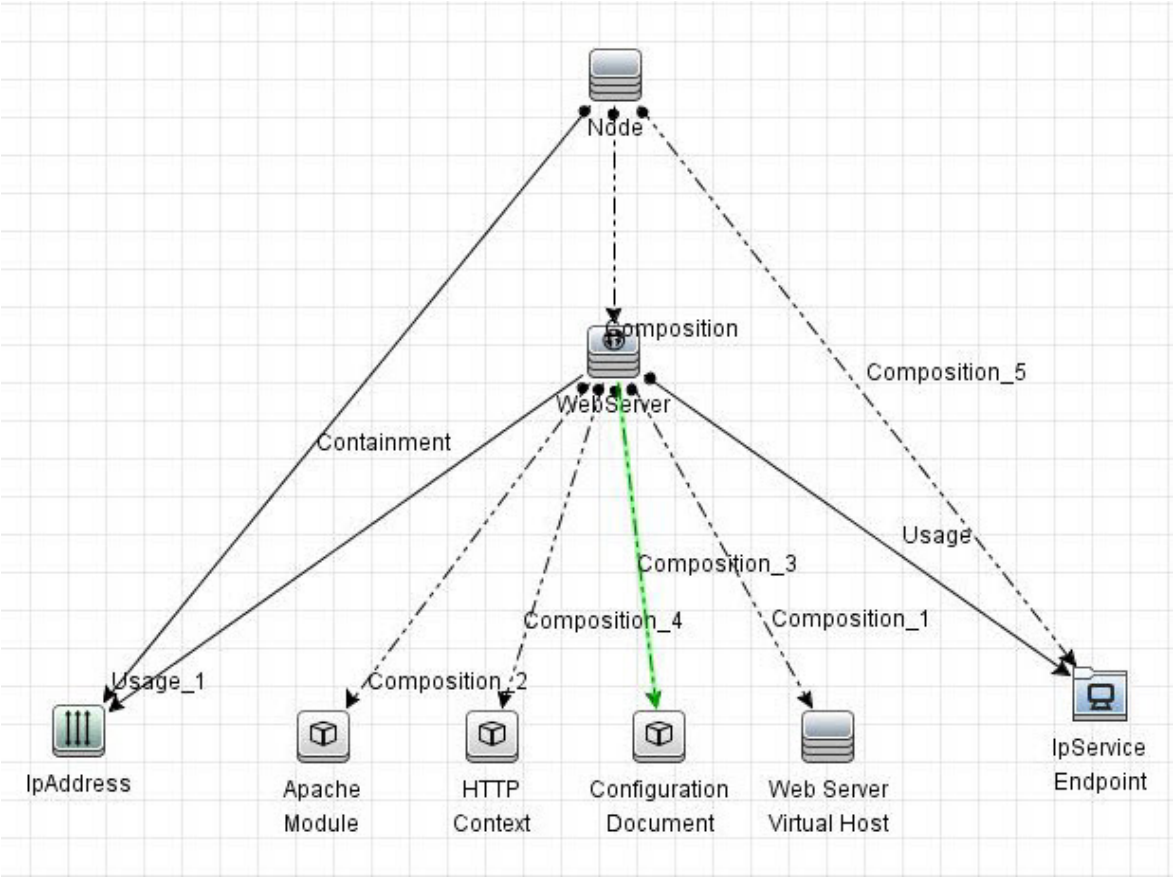
### トリガ TQL



ノード名	条件
Node	なし
Process	(Process Path 類似 (大文字小文字の区別なし) %httpd% OR Process Path 類似 (大文字小文字の区別なし) %apache%) AND NOT Process Path 類似 (大文字小文字の区別なし) %tomcat%
Shell	NOT Reference to the credentials dictionary entry null
IpAddress	NOT IP Probe Name null

## トポロジ・マップ

Web Servers by Shell トポロジは次に示すとおりです。



## サポートされているポリシー

このジョブは次のバージョンをサポートしています。

ベンダ	製品	サポートされるバージョン
IBM	IBM HTTP Server	5, 6.1, 7, 8.0, 8.5, 8.5.5
Apache	Apache HTTP Server	1.3, 2.0, 2.2, 2.4

## アダプタ情報

このジョブでは **Apache Web Servers by Shell** アダプタを使用します。

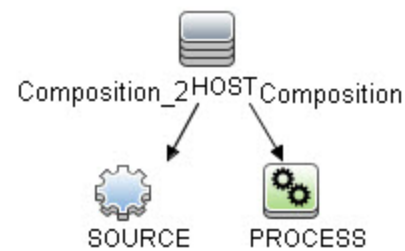
アダプタ・タイプ

Jython

入力 CIT

Shell

入力 TQL



トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}
serverProcCmdLine	\${PROCESS.process_cmdline:NA}
serverProcPath	\${PROCESS.process_path:NA}

使用スクリプト

- file\_ver\_lib.py
- websphere\_plugin\_config.py
- websphere\_plugin\_config\_reporter.py
- tcp\_discovery\_oam.py



- Apache.py

## 検出された CIT

- Apache
- Apache Module
- Composition
- ConfigurationDocument
- Containment
- Dependency
- Deployed
- HTTP Context
- IBM HTTP Server
- IpAddress
- IpServiceEndpoint
- Node
- RunningSoftware
- Usage
- Web Server Virtual Host
- WebServer

## グローバル構成ファイル

globalSettings.xml

アダプタ・パラメータ

パラメータ	説明
configFiles	ファイル名を含む Apache 構成ファイルへの完全パス。 <div>注 :複数のファイルはセミコロン(;)で区切る必要がある。</div>

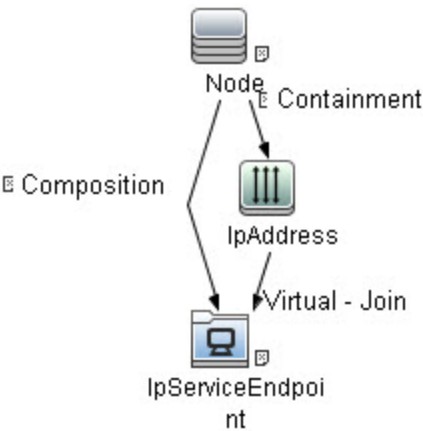
Web Server Detection using TCP Ports ジョブ

本 項 ではこのジョブに関する詳細を説明します。

はじめに

このジョブは TCP ポートを使用して Web サーバを検出します。

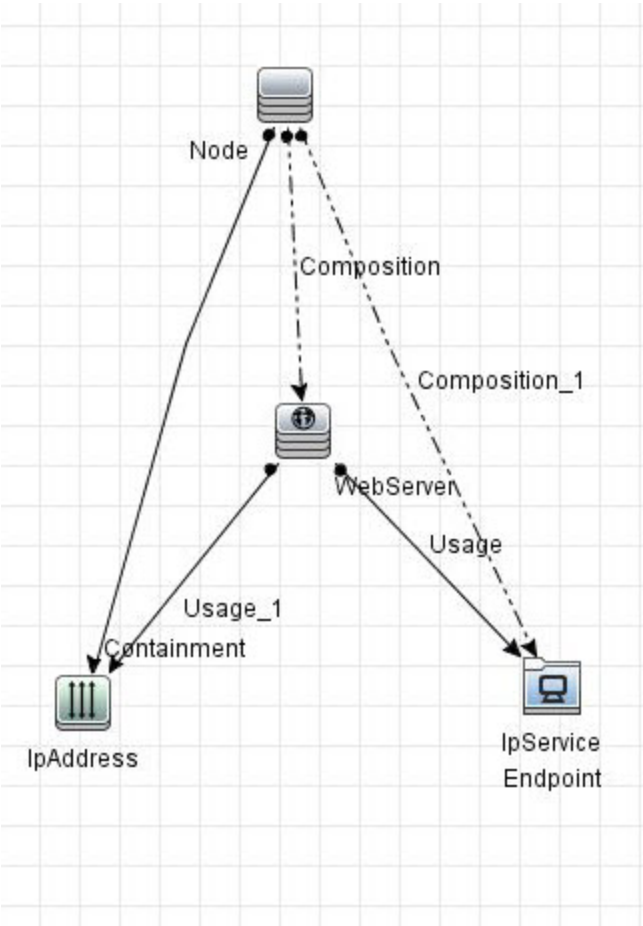
トリガ TQL



ノード名	条件
Node	なし
IpAddress	NOT IP Probe Name null
IpServiceEndpoint	IpServiceName 等しい http OR ServiceNames 含む http

## トポロジ・マップ

**Web Server Detection using TCP Ports** トポロジは次に示すとおりです。



## サポートされているポリシー

このジョブは次のバージョンをサポートしています。

ベンダ	製品	サポートされるバージョン
IBM	IBM HTTP Server	5, 6.1, 7, 8.0, 8.5, 8.5.5
Apache	Apache HTTP Server	1.3, 2.0, 2.2, 2.4
Microsoft	IIS	5, 6, 7, 7.5, 8, 8.5

## アダプタ情報

このジョブでは **TCP Web Server Detection** アダプタを使用します。

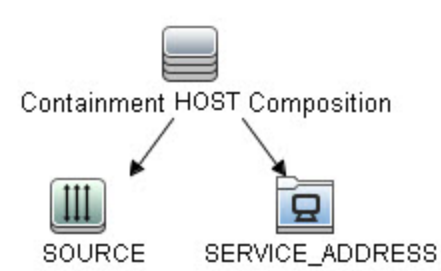
### アダプタ・タイプ

Jython

### 入力 CIT

IpAddress

### 入力 TQL



### トリガ CI データ

名前	値
http_port	\${SERVICE_ADDRESS.network_port_number}
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}

### 使用スクリプト

WebserverConnection.py

### 検出された CIT

- Composition
- Containment

- IIS Web Server
- IPAddress
- IpServiceEndpoint
- Node
- RunningSoftware
- SunOne
- Usage
- WebServer

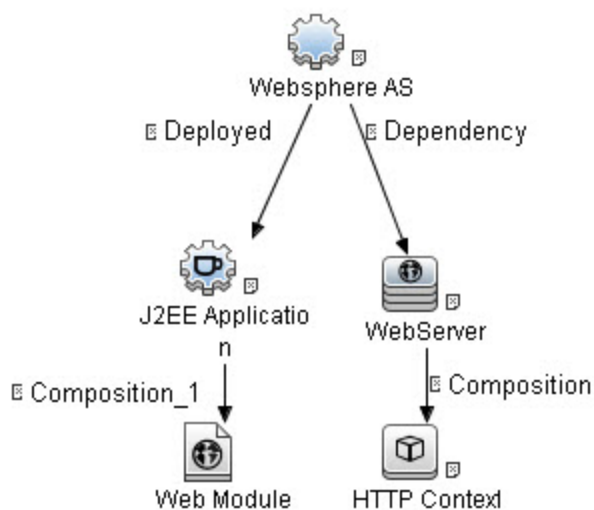
## WebSphere to Web Server Dependency ジョブ

本項ではこのジョブに関する詳細を説明します。

### はじめに

このジョブは Web サーバと WebSphere Application Server 間の依存関係を検出します。

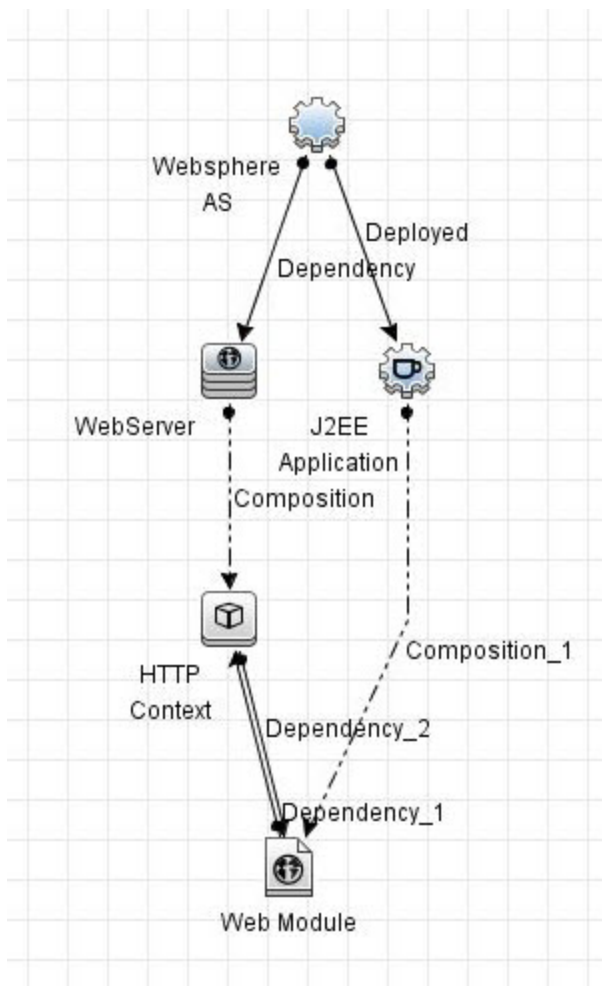
### トリガ TQL



ノード名	条件
Websphere AS	NOT J2EE Server Full Name null
J2EE Application	なし
Web Module	NOT J2EE Managed Object Context Root null
WebServer	なし
HTTP Context	NOT Web Application Context null AND NOT Web Application Server Name null

## トポロジ・マップ

**WebSphere to Web Server Dependency** トポロジは次に示すとおりです。



## サポートされているポリシー

このジョブは次のバージョンをサポートしています。

IBM WebSphere Application Server 5, 6.1, 7, 8.0, 8.5, および 8.5.5。

## アダプタ情報

このジョブでは **Web server to WebSphere Dependency** アダプタを使用します。

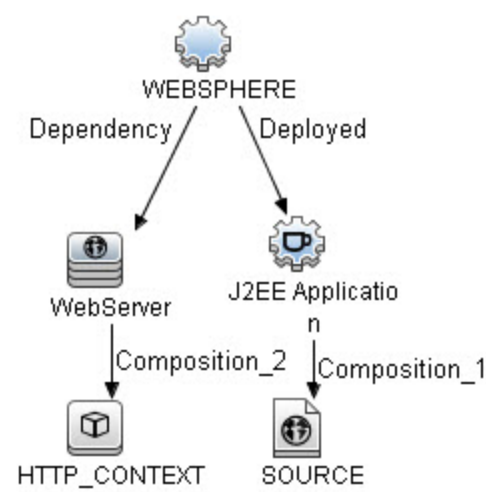
アダプタ・タイプ

Jython

入力 CIT

Web Module

入力 TQL



トリガ CI データ

名前	値
http_context_context_root	\${HTTP_CONTEXT.httpcontext_webapplicationcontext}
http_context_id	\${HTTP_CONTEXT.root_id}
http_context_root_container	\${HTTP_CONTEXT.root_container}
http_context_server	\${HTTP_CONTEXT.httpcontext_webapplicationserver}
web_module_context_root	\${SOURCE.j2eemanagedobject_contextroot}
web_module_id	\${SOURCE.root_id}
websphere_full_name	\${WEBSPHERE.j2eeserver_fullname}



## 使用スクリプト

websphere\_webserver\_dependency.py

## 検出された CIT

- Dependency
- HTTP Context
- J2EE Deployed Object
- RunningSoftware

# 第61章: Microsoft Internet Information Services (IIS)ディスカバリ

## 本章の内容

サポートされるバージョン .....	927
Microsoft Internet Information Services (IIS)ディスカバリのトポロジ .....	927
Microsoft Internet Information Services (IIS)のトポロジの検出方法 .....	928
IIS Applications by NTCMD or UDA ジョブ .....	930
Bugzilla, Wordpress, MediaWiki のディスカバリ .....	933
トラブルシューティングおよび制限事項 .....	934

## サポートされるバージョン

このディスカバリでは、Microsoft Internet Information Services (IIS) バージョン 5, 6, 7, 7.5, 8, 8.5 がサポートされています。

### 注:

- IIS 7 のディスカバリは、IIS 6 管理互換ツールを通じてサポートされます。
- IIS 7 以降、IIS では、新しいコマンド・ライン・ツール **AppCmd.exe** が利用できるようになりました。このツールでは、Web サーバのオブジェクトを構成、クエリし、出力をテキストまたは XML 形式で返すことができます。したがって、**AppCmd.exe** を使用することにより、IIS 7 以降のバージョンを検出することが可能です。

## Microsoft Internet Information Services (IIS) ディスカバリのトポロジ

注: 検出された CIT のリストについては、[「IIS Applications by NTCMD or UDA ジョブ」](#)(930ページ)を参照してください。



ページ 928 / 1163

## 1. 前提条件 - プロトコル資格情報の設定

このデイスカバリでは、**NTCMD** プロトコルを使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

## 2. 前提条件 - その他

- 関連する情報をすべて取得するには、DFM が Visual Basic スクリプトを実行可能で、かつ DFM に **%SystemRoot%\system32\drivers\etc** フォルダへの書き込み権限がある必要があります。
- IIS を実行中のターゲット・マシンが Data Flow Probe の範囲内にあることを確認します。

## 3. デイスカバリの実行

[Universal Discovery] ウィンドウで、次の順にジョブをアクティブ化します。

- Host Connection by Shell** ジョブを実行して Shell CIT を作成します。
- Host Applications by Shell** ジョブを実行して、IIS Web Server CI と IIS Application Pool CI、および管理プロセスへの対応する**依存関係**リンクを検出します。
- IIS Applications by NTCMD or UDA** ジョブを実行して、IIS の詳細なトポロジを検出します。

接続が確立されると、DFM は、リモート・マシンの **adsutil.vbs** スクリプトをコピーします。DFM は、このツールの出力から IIS のトポロジ情報を取得します。

Microsoft IIS バージョン 7.0 では、IIS アプリケーションを(以前のバージョンと同様に)仮想ディレクトリから作成できるだけでなく、Web ディレクトリからも作成できます。したがって、Web ディレクトリから作成されたアプリケーションを検出した場合、DFM は IIS Web Directory CI を作成します。

必要な権限を表示するには、[Universal Discovery] > [デイスカバリ モジュール / ジョブ] > [Middleware] > [Web Servers] > [IIS] > [IIS Applications by NTCMD or UDA] ジョブを選択し、[詳細] タブ > [デイスカバリ ジョブの詳細] 表示枠を選択して、[権限を表示] ボタンをクリックします。詳細については、[「IIS Applications by NTCMD or UDA ジョブ」\(930ページ\)](#)を参照してください。

**注:** IIS Web Dir CI が作成されるのは、トポロジの下に IIS Virtual Dir CI または

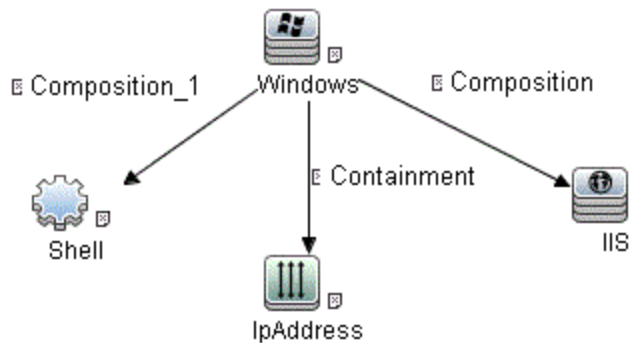
web.config ファイルが存在する場合に限られます。それ以外の場合、IIS Web Dir CI はレポートされません。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## IIS Applications by NTCMD or UDA ジョブ

本項の内容

### トリガ・クエリ



ノード名	条件
Windows	なし
Shell	NOT Reference to the credentials dictionary entry null AND(CI Type 等しい ntcmd OR CI Type 等しい uda)
IpAddress	NOT IP Probe Name null
IIS	NOT Version null

### アダプタ

このジョブでは、NTCMD\_APP\_Dis\_IIS アダプタを使用します。

• トリガ CI データ

名前	値
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
host_ips	\${IpAddress.name}
iis_name	\${SOURCE.discovered_product_name}
iis_version	\${SOURCE.version}
ip_address	\${SHELL.application_ip}

• 権限

権限	操作	使用状況の詳細	オブジェクトとパラメータ
Shell	exec	Basic login	uname ver wmic OS Get CodeSet wmic OS Get OSLanguage
Shell	copy	Copy file to remote machine	adsutil.vbs - Visual Basic script for IIS discovery
Shell	exec	Discover IIS Topology	cscript.exe adsutil.vbs ENUM "MSFTPSVC/{SITENUM}/root" cscript.exe adsutil.vbs ENUM "W3SVC" cscript.exe adsutil.vbs ENUM "W3SVC/AppPools" cscript.exe adsutil.vbs ENUM "W3SVC/AppPools/{POOLNAME}" cscript.exe adsutil.vbs ENUM "W3SVC/{SITENUM}" cscript.exe adsutil.vbs ENUM "W3SVC/{SITENUM}/root" cscript.exe adsutil.vbs ENUM /p "W3SVC/{SITENUM}/Root" cscript.exe adsutil.vbs ENUM /p "W3SVC/{SITENUM}/Root/{IIS_...}" cscript.exe adsutil.vbs ENUM /p MSFTPSVC cscript.exe adsutil.vbs ENUM /p MSFTPSVC/{SITENUM}/Root cscript.exe adsutil.vbs ENUM /p W3SVC cscript.exe adsutil.vbs ENUM /p W3SVC/AppPools cscript.exe adsutil.vbs ENUM MSFTPSVC cscript.exe adsutil.vbs ENUM MSFTPSVC/{SITENUM} cscript.exe adsutil.vbs ENUM SMTPSVC cscript.exe adsutil.vbs ENUM W3SVC/{SITENUM}/Root/{IIS_...} cscript.exe adsutil.vbs GET "{PATH}/KeyType" cscript.exe adsutil.vbs GET KeyType cscript.exe adsutil.vbs GET MSFTPSVC/{SITENUM}/Root/{P...} cscript.exe adsutil.vbs GET MaxBandwidth dir /B hostname nslookup <hostname> type <file_path>

• アダプタ・パラメータ

パラメータ	説明
<b>acceptedStatusCodes</b>	URL の検証時に <b>OK</b> として処理するステータス・コードが含まれる。
<b>adsutil_path</b>	<b>adsutil.vbs</b> スクリプトのパスと名前を入力する。 <b>adsutil.vbs</b> スクリプトは、Microsoft for IIS 管理タスクで提供される無料のスクリプト。
<b>checkConnectionToUrl</b>	<b>true</b> に設定すると、プローブ・マシンからの HTTP(s) HEAD メソッドにより、レポートされた URL が利用可能であるかどうかを検証される。接続できなかった場合、その URL はスキップされる。
<b>do_web_service</b>	<p><b>True</b> : IIS Web Service CI がレポートされる。</p> <p>注: DFM でこの CI がレポートされるようにするには、<b>report_legacy_topology</b> も <b>true</b> に設定する必要がある。</p>
<b>prefer_appcmd</b>	<b>AppCmd.exe</b> を有効にすることで IIS を検出する。標準設定では、このパラメータは <b>true</b> に設定されている。
<b>report_legacy_topology</b>	後方互換性を維持するため、標準設定では DFM はレガシー IIS トポロジを引き続きレポートする。
<b>web_service_file_extensions</b>	Web サービスとして検出するファイル拡張子のリスト。
	注: ワイルドカードはサポートされていない。

## 検出された CIT

- ClientServer
- Composition
- ConfigurationDocument
- Containment
- Database
- Database Resource
- Dependency



- Deployed
- IIS FTP Server
- IIS Resource
- IIS SMTP Server
- IIS Web Server
- IpAddress
- IpServiceEndpoint
- Node
- UriEndpoint
- Usage
- Web Server Virtual Host

## Bugzilla, Wordpress, MediaWiki のディスカバリ

詳細については、[「Bugzilla, Wordpress, MediaWiki の検出方法」](#)(778ページ)を参照してください。

## トラブルシューティングおよび制限事項

本項では Microsoft Internet Information Services (IIS) ディスカバリのトラブルシューティングと制限事項について説明します。

- IIS Web Server CI は、マシン上で Web サービスが実行されていなくても、IIS FTP サービスと IIS SMTP サービスが存在すれば作成されます。
- 検出された web.config ファイルの ConnectionStrings プロパティにパスワードが含まれている場合は、構成ファイル CI の作成時、パスワードがアスタリスク文字で置換されます。

## 第12部: Middleware > Web Services

# 第 62 章: UDDI レジストリ・ディスカバリ

## 本章の内容

概要 .....	937
サポートされるバージョン .....	937
トポロジ .....	937
UDDI プロセスの検出方法 .....	938

## 概要

UDDI ディスカバリ・プロセスでは、UDDI レジストリから Web サービスを検出できます。

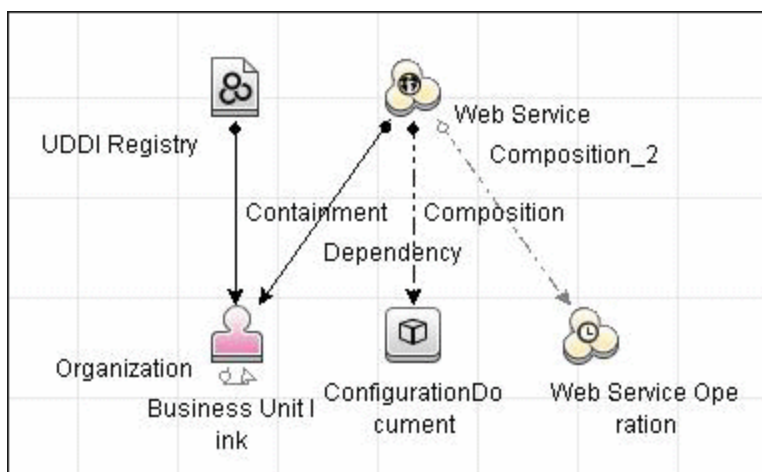
DFM は UDDI レジストリに対してクエリを実行し、その Web サービス(SOAP 以外のサービスを含む)や特定の発行者 サービス(UDDI レジストリ・プロトコルで定義されている場合)がないか確認します。UDDI レジストリで見つかった Web サービスは、CMDB の **WebService Resource** CI によって表され、**UDDI Registry** CI としてレジストリが作成されます。

## サポートされるバージョン

このディスカバリでは、UDDI バージョン 2 および 3 がサポートされています。

## トポロジ

次の図は、**SOA\_UDDI\_View** のトポロジを示しています。



## UDDI プロセスの検出方法

本項の内容

### 1. 前提条件 - ruddi.jar ファイルのインストール

- a. ruddi-1.0-bin.jar ファイルを次の場所からダウンロードします。

[http://sourceforge.net/projects/s-feng/files/S-FENG/lib/ruddi1.0/ruddi-1.0-bin.jar/download?use\\_mirror=iweb](http://sourceforge.net/projects/s-feng/files/S-FENG/lib/ruddi1.0/ruddi-1.0-bin.jar/download?use_mirror=iweb)

- b. **ruddi.jar** ファイルを名前変更し、プローブ・マシンの次のディレクトリにコピーします。

<hp>\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources\ruddi\

### 2. 前提条件 - プロトコル資格情報の設定

UDDI プロトコルを設定します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 3. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次のジョブをアクティブにします。

- **Web Services by URL**
- **Web Service Connections by UDDI Registry**
- **Web Services by UDDI Registry**

### 4. サービス発行者の詳細の指定(任意指定)

UDDI レジストリ・アダプタの **organization** パラメータをサービス発行者の名前と組織の説明で更新します。

アダプタ・パラメータの編集の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Adapter Definition Tab」を参照してください。

## 第13部: Middleware > Proxy Servers

# 第 63 章 : IBM Security Access Manager ディスカバリ

本章の内容

概要 ..... 941

サポートされるバージョン ..... 941

WebSeal Connection by Web Services ジョブ ..... 941

WebSeal Connection By Shell ジョブ ..... 944

WebSeal Topology by Web Services ジョブ ..... 946

WebSeal Topology by Shell ジョブ ..... 949

HTTP を使用して IBM Security Access Manager を検出する方法 ..... 951

シェルを使用して IBM Security Access Manager を検出する方法 ..... 952



## 概要

このパッケージは、IBM Security Access Manager (旧称 IBM Tivoli Access Manager)を検出します。IBM Security Access Manager は、Web アクセスのセキュリティを保護する統合アクセス・アプライアンスです。このパッケージに含まれているアダプタは、IBM Security Access Managerを検出する際に次のプロトコルを使用します。

- HTTP/Web
- Shell

## サポートされるバージョン

サポートされるバージョンは、次のとおりです。

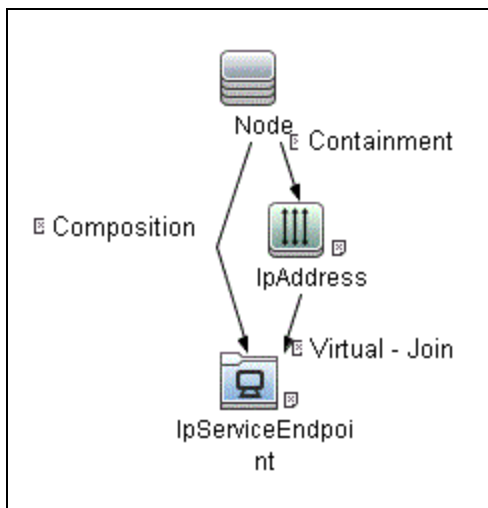
ジョブ	サポートされるバージョン
IBM Security Access Manager for Web by HTTP	8.x
IBM Security Access Manager for Web by Shell	6 ~ 8.x

## WebSeal Connection by Web Services ジョブ

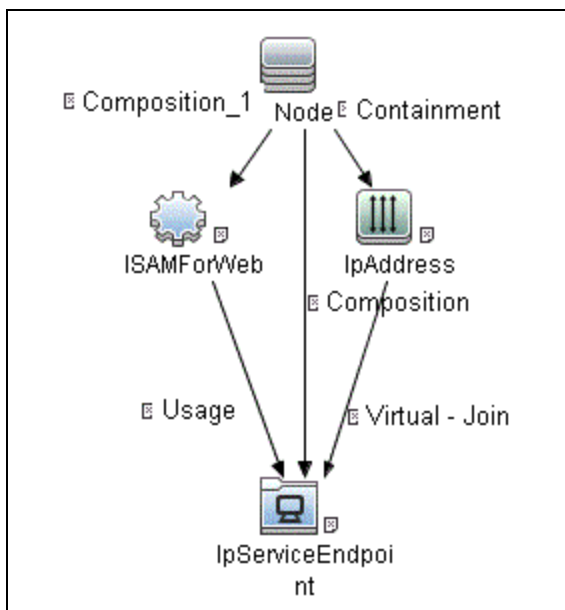
本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

- `https_ports`



- isam\_with\_ipse



## アダプタ

このジョブでは、**webseal\_connection\_by\_webservices** アダプタを使用します。

## 使用スクリプト

- webseal\_connection\_by\_webservices.py

検出された CIT

- Composition
- Containment
- ISAMForWeb
- IpAddress
- IpServiceEndpoint
- Node
- Usage

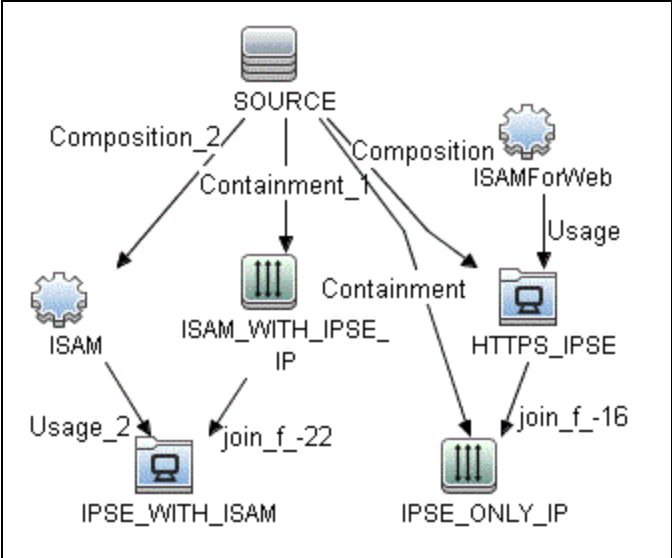
パラメータ

名前	タイプ	説明
autoAcceptCerts	string	受信した証明書を信頼済みとして一時的に受け入れる
firmware_settings_api_query	string	ファームウェア設定 Web サービス API に対する Http クエリ
management_authentication_api_query	string	管理認証 Web サービス API に対する Http クエリ
pdadmin_api_query	string	pdadmin Web サービス API に対する Http クエリ
reverseproxy_api_query	string	reverseproxy Web サービス API に対する Http クエリ

入力 CIT

- Node

入力 TQL



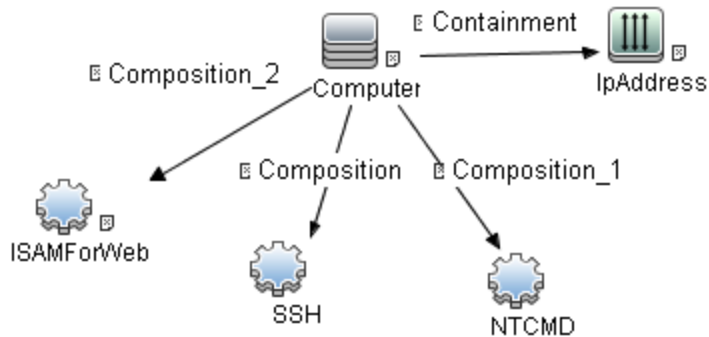
トリガ CI データ

名前	値	説明
https_ipse_ids	\${HTTPS_IPSE.root_id:NA}	トリガが ISAM インスタンスを含まない https IpServiceEndpoint の場合、IpServiceEndpoint のルート ID のリスト
ipse_only_ips	\${IPSE_ONLY_IP.name:NA}	トリガが ISAM インスタンスを含まない https IpServiceEndpoint の場合、IP アドレスのリスト
isam_credential_ids	\${ISAM.credentials_id:NA}	トリガが ISAM インスタンスの場合、資格情報 ID のリスト
isam_ids	\${ISAM.root_id:NA}	トリガが ISAM インスタンスの場合、ISAM ルート ID のリスト
isam_with_ipse_ips	\${ISAM_WITH_IPSE_IP.name:NA}	トリガが ISAM インスタンスの場合、IP アドレスのリスト

WebSeal Connection By Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ



## アダプタ

このジョブでは、**webseal\_connection\_by\_shell** アダプタを使用します。

## 使用スクリプト

- webseal\_topology.py
- pdadmin\_shell\_webseal\_discoverer.py
- webseal\_connection\_by\_shell.py

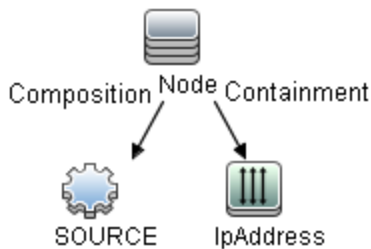
## 検出された CIT

- Composition
- Containment
- ISAMForWeb
- IpAddress
- Node

入力 CIT

- Shell

入力 TQL



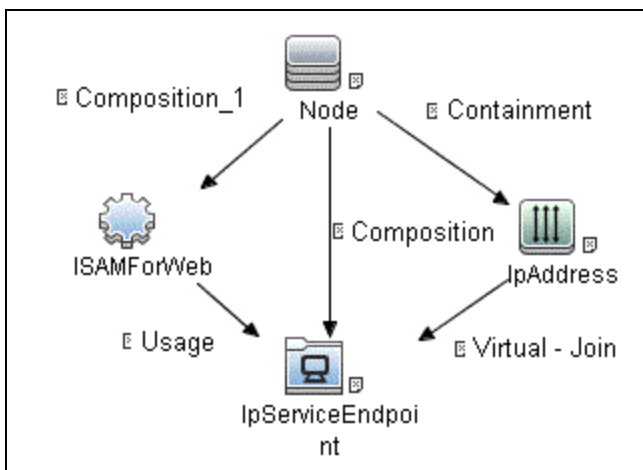
トリガ CI データ

名前	値	説明
Protocol	\${SOURCE.root_class}	シェル・タイプ(SSH/NTCMD)
credentialsId	\${SOURCE.credentials_id}	有効なシェルの資格情報 ID
hostId	\${SOURCE.root_container}	Webseal が属しているノードの UCMDB ID
ip_address	\${SOURCE.application_ip}	宛先 IP アドレス

## WebSeal Topology by Web Services ジョブ

本項ではこのジョブに関する詳細を説明します。

## トリガ・クエリ



## アダプタ

このジョブでは、**webseal\_topology\_by\_webservices** アダプタを使用します。

## 使用スクリプト

- webseal\_topology\_by\_webservices.py

## 検出された CIT

- Composition
- Containment
- DirectoryServer
- ISAMForWeb
- ISAMJunction
- ISAMPolicyServer
- IpAddress
- IpServiceEndpoint
- Node
- RunningSoftware

- Usage

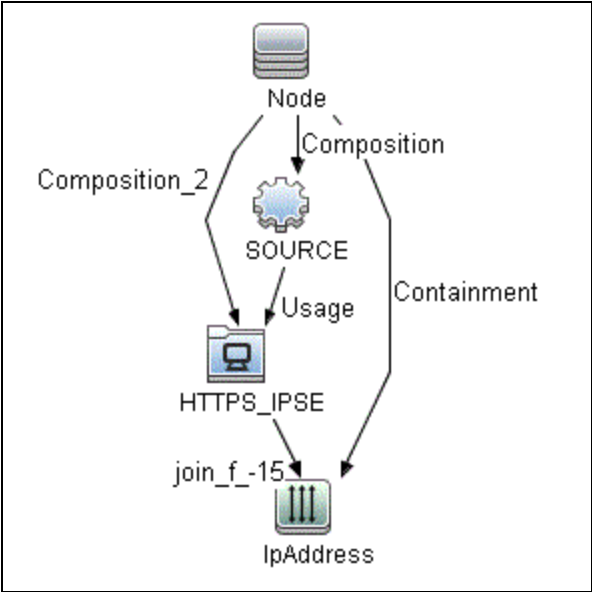
パラメータ

名前	タイプ	説明
autoAcceptCerts	string	受信した証明書を信頼済みとして一時的に受け入れる
firmware_settings_api_query	string	ファームウェア設定 Web サービス API に対する Http クエリ
management_authentication_api_query	string	管理認証 Web サービス API に対する Http クエリ
pdadmin_api_query	string	pdadmin Web サービス API に対する Http クエリ
reverseproxy_api_query	string	reverseproxy Web サービス API に対する Http クエリ

入力 CIT

- ISAMForWeb

入力 TQL





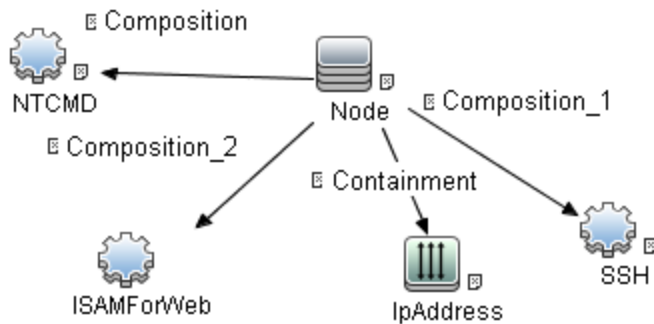
#### トリガ CI データ

名前	値	説明
ip_address	\${IpAddress.name}	トリガ ISAM インスタンスの IP アドレスのリスト
container_cmdbid	\${Node.root_id}	コンテナのルート ID
cmdbid	\${SOURCE.root_id}	ISAM インスタンスのルート ID
credential_id	\${SOURCE.credentials_id}	資格情報ディクショナリの参照
name	\${SOURCE.name}	ISAM インスタンスの名前

## WebSeal Topology by Shell ジョブ

本項ではこのジョブに関する詳細を説明します。

#### トリガ・クエリ



#### アダプタ

このジョブでは、**webseal\_topology\_by\_shell** アダプタを使用します。

### 使用スクリプト

- webseal\_topology\_by\_shell.py
- pdadmin\_shell\_webseal\_discoverer.py
- webseal\_topology.py

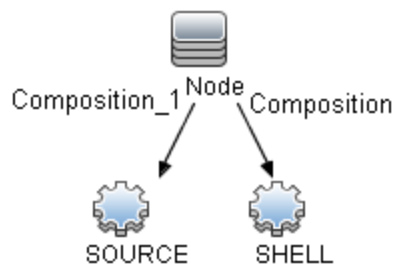
## 検出された CIT

- Composition
- Containment
- DirectoryServer
- ISAMForWeb
- ISAMJunction
- ISAMPolicyServer
- IpAddress
- IpServiceEndpoint
- Node
- RunningSoftware
- Usage

## 入力 CIT

- ISAMForWeb

## 入力 TQL



## トリガ CI データ

名前	値	説明
Protocol	\${SHELL.root_class}	シェル・タイプ(SSH/NTCMD)
credentialsId	\${SHELL.credentials_id}	有効なシェルの資格情報 ID
hostId	\${SHELL.root_container}	Webseal が属しているノードの UCMDB ID
ip_address	\${SHELL.application_ip}	宛先 IP アドレス
websealId	\${SOURCE.root_id}	Webseal のルート ID
webseal_credentials_id	\${SOURCE.isam_credentials_id}	資格情報ディクショナリの参照

## HTTP を使用して IBM Security Access Manager を検出する方法

このタスクでは、IBM Security Access Manager の検出方法について説明します。これには次の手順が含まれます。

- [「前提条件 - プロトコル資格情報の設定」\(664ページ\)](#)
- [「前提条件 - その他」\(664ページ\)](#)
- [「ディスカバリの実行」\(665ページ\)](#)

### 前提条件 - プロトコル資格情報の設定

次のプロトコルを設定します。

ジョブ	プロトコル
IBM Security Access Manager by HTTP	HTTP

### 前提条件 - その他

ポリシー・サーバ・アカウントの入手

HTTP ベースのディスカバリは、WebSeal プラットフォームが備えている Webservices API に基づきます。このディスカバリでは、pdadmin webservices API が使用されます。この API を使用するには、ポリシー・サーバの資格情報を設定する必要があります。ディスカバリでは、基本認証で使用されたものと同じユーザ名とパスワードが使用されます。つまり、WebSeal 管理コンソールへのアクセス用に構成されたものと同じユーザ名とパスワードの LDAP ユーザアカウントが使用される必要があります。

## ディスカバリの実行

次のジョブをアクティブにします。

- **Range IPs by ICMP**
- **Databases TCP Ports**
- **Webseal Connection By Web Services**
- **Webseal Topology By Web Services**

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## シェルを使用して IBM Security Access Manager を検出する方法

このタスクでは、IBM Security Access Manager の検出方法について説明します。これには次の手順が含まれます。

- [「前提条件 - プロトコル資格情報の設定」\(664ページ\)](#)
- [「前提条件 - その他」\(664ページ\)](#)
- [「ディスカバリの実行」\(665ページ\)](#)

### 前提条件 - プロトコル資格情報の設定

次のプロトコルを設定します。

ジョブ	プロトコル
IBM Security Access Manager by Shell	SSH または NTCMD

ジョブ	プロトコル
	任意指定 : トポロジに対してディープ・ディスカバリを実行する場合は, LDAP。

## 前提条件 - その他

ポリシー・サーバ・アカウントの入手

シェル・ベースのディスカバリでは, リモート・ノードに接続し, pdadmin ユーティリティを使用します。このユーティリティでは, WebSeal アカウントと同じ資格情報を備えた LDAP アカウントが必要です。

## ディスカバリの実行

次のジョブをアクティブにします。

- Range IPs by ICMP
- Host Connection by Shell
- Host Applications by Shell
- Webseal Connection By Shell
- Webseal Topology By Shell

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## 第14部: Middleware > Security Servers

# 第64章: Oracle Access Management ディスカバリ

## 本章の内容

Oracle Access Management Connection by Web Services ジョブ .....	956
はじめに .....	956
トポロジ・マップ .....	956
サポート・ポリシー .....	956
アダプタ情報 .....	956
Oracle Access Management Policies by Web Services ジョブ .....	958
はじめに .....	958
トポロジ・マップ .....	958
サポート・ポリシー .....	959
アダプタ情報 .....	959
Oracle Access Management Dependencies via URL ジョブ .....	960
はじめに .....	960
トポロジ・マップ .....	961
サポート・ポリシー .....	962
アダプタ情報 .....	962

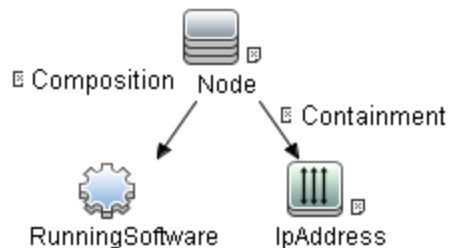
# Oracle Access Management Connection by Web Services ジョブ

本項ではこのジョブに関する詳細を説明します。

## はじめに

このジョブでは、Oracle Access Management サーバのバージョンと、Oracle Access Management サーバが使用している HTTP 資格情報を検出します。

## トリガ TQL



ノード名	条件
IpAddress	NOT IP Probe Name null
RunningSoftware	DiscoveredProductName 等しい Oracle Access Management

## トポロジ・マップ

なし

## サポート・ポリシー

Oracle Access Management 11.1.2.0.0 をサポートします。

## アダプタ情報

このジョブでは、**Oracle Access Management Connection by Web Services** アダプタを使用します。



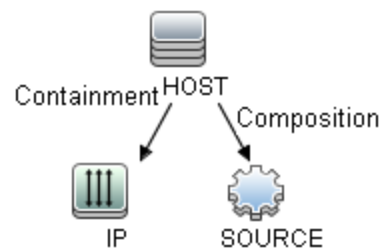
アダプタ・タイプ

Jython

入力 CIT

RunningSoftware

入力 TQL



トリガ CI データ

名前	値
cmdb_id	\${SOURCE.root_id}
ip_address	\${IP.name}
ip_domain	\${IP.routing_domain}

ワークフローのステップ

なし

使用スクリプト

oam\_connection\_by\_webservices.py

検出された CIT

RunningSoftware

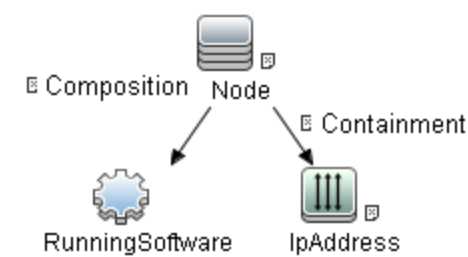
# Oracle Access Management Policies by Web Services ジョブ

本 項 ではこのジョブに関する詳細を説明します。

## はじめに

このジョブでは, **Oracle Access Management Connection by Web Services** ジョブで検出された HTTP 資格情報を使用して Oracle Access Management ポリシーを検出します。

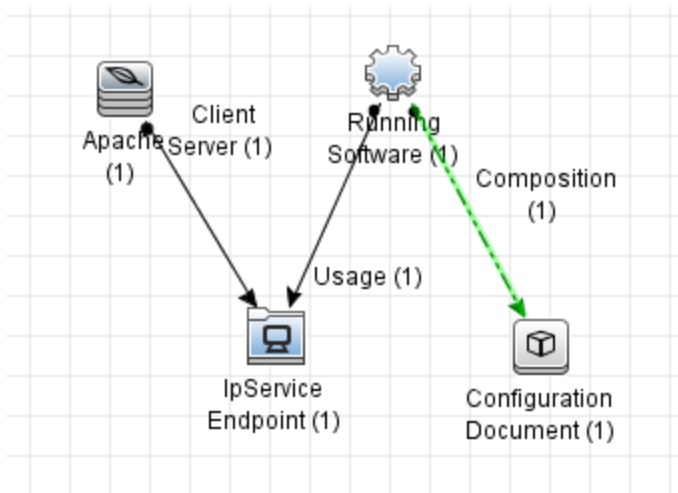
## トリガ TQL



ノード名	条 件
IpAddress	NOT IP Probe Name null
RunningSoftware	DiscoveredProductName 等しい Oracle Access Management AND NOT Version null

## トポロジ・マップ

**Oracle Access Management Policies by Web Services** トポロジを次に示します。



## サポート・ポリシー

Oracle Access Management 11.1.2.0.0 をサポートします。

## アダプタ情報

このジョブでは、**Oracle Access Management Policies by Web Services** アダプタを使用します。

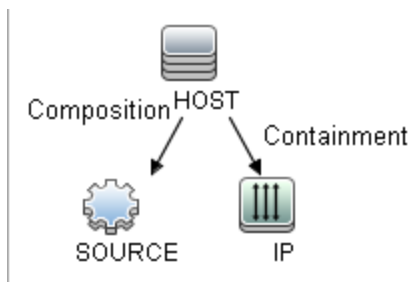
### アダプタ・タイプ

Jython

### 入力 CIT

RunningSoftware

### 入力 TQL



## トリガ CI データ

名前	値
cmdb_id	\${SOURCE.root_id}
credential_id	\${SOURCE.credentials_id}
ip_address	\${IP.name}
version	\${SOURCE.version}

## ワークフローのステップ

なし

## 使用スクリプト

oam\_policies\_by\_webservices.py

## 検出された CIT

Composition

RunningSoftware

ConfigurationDocument

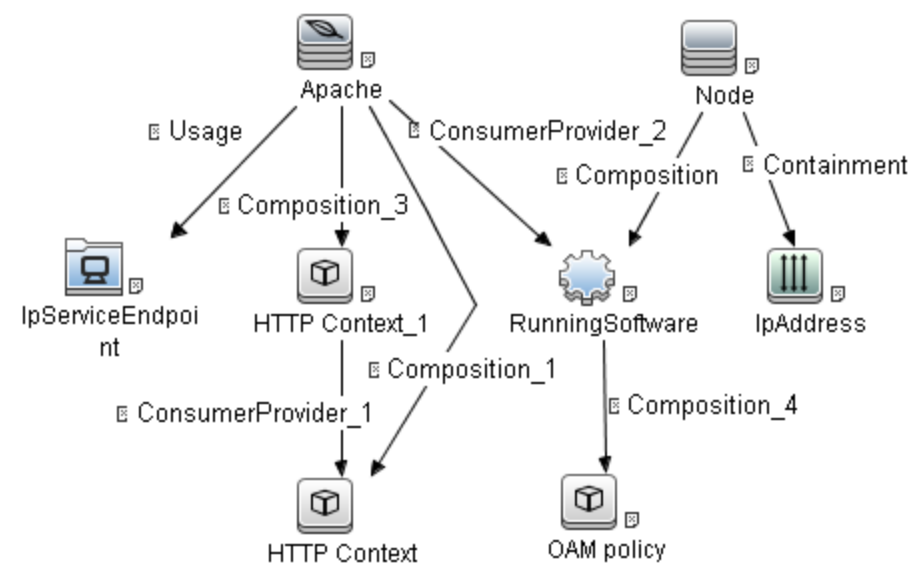
# Oracle Access Management Dependencies via URL ジョブ

本項ではこのジョブに関する詳細を説明します。

## はじめに

このジョブでは、Oracle Access Management (OAM)ポリシーを分析した後、OAM サーバのリダイレクト URL を検出し、OAM Webgate URL とリダイレクト URL 間の依存関係を作成します。

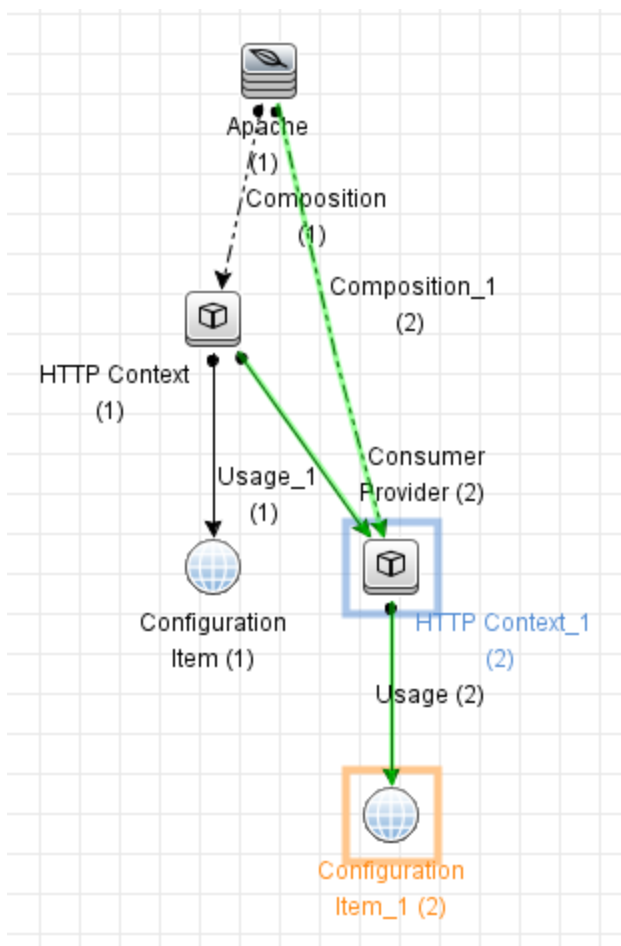
トリガ TQL



ノード名	条件
Node	なし
IpAddress	NOT IP Probe Name null
RunningSoftware	なし
OAM policy	Name 等しい policy.xml
Apache	なし
HTTP Context	なし
HTTP Context_1	なし
IpServiceEndpoint	なし

トポロジ・マップ

Oracle Access Management Dependencies via URL トポロジを次に示します。



## サポート・ポリシー

Oracle Access Management 11.1.2.0.0 をサポートします。

## アダプタ情報

このジョブでは、**OAM url dependencies via WebGate url** アダプタを使用します。

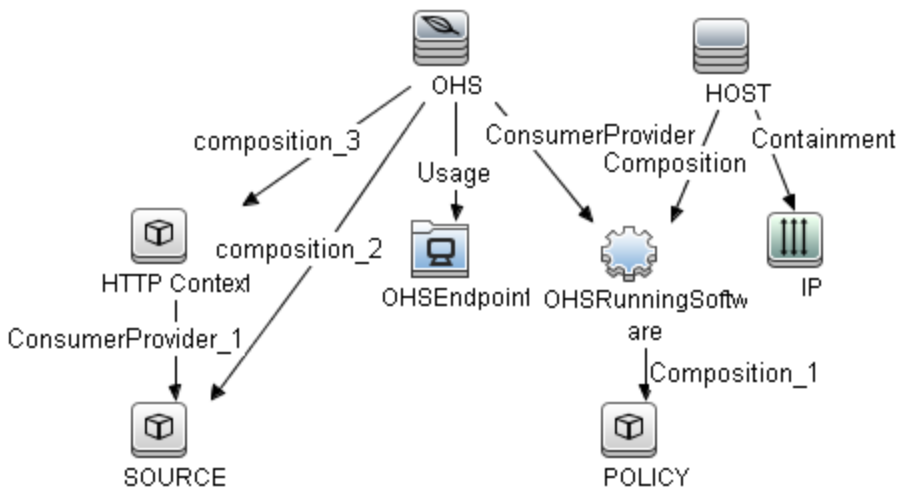
### アダプタ・タイプ

Jython

### 入力 CIT

HTTP Context

入力 TQL



トリガ CI データ

名前	値
OHS_ADDRESS	\${OHSEndpoint.ipserver_address}
OHS_CONTEXT	\${SOURCE.httpcontext_webapplicationcontext}
OHS_CONTEXT_ID	\${SOURCE.root_id}
OHS_ID	\${OHS.root_id}
POLICY_FILE	\${POLICY.document_content}

使用スクリプト

OAMPolicy.py  
OAMURLDependency.py

検出された CIT

ConfigurationDocument  
ConsumerProvider  
HTTP Context

## 第15部: Network Infrastructure



## 第65章: ネットワーク - 基本ディスカバリ

### 本章の内容

概要 .....	967
Host Connection by Shell の検出方法 .....	969
Host Connection by SNMP の検出方法 .....	970
Host Connection by WMI の検出方法 .....	972
Client Connection by SNMP の検出方法 .....	972
Host Connection by Shell ジョブ .....	973
ディスカバリ・メカニズム .....	973
トリガ・クエリ .....	981
ジョブ・パラメータ .....	982
アダプタ .....	983
検出された CIT .....	983
トラブルシューティングおよび制限事項 .....	985
Host Connection by SNMP ジョブ .....	987
ディスカバリ・メカニズム .....	987
トリガ・クエリ .....	988
ジョブ・パラメータ .....	989
アダプタ .....	990
検出された CIT .....	990
トラブルシューティングおよび制限事項 .....	991
Host Connection by WMI ジョブ .....	993
ディスカバリ・メカニズム .....	993
トリガ・クエリ .....	995
ジョブ・パラメータ .....	996
アダプタ .....	996
検出された CIT .....	997
トラブルシューティングおよび制限事項 .....	997
Client Connection by SNMP ジョブ .....	999

ディスカバリ・メカニズム .....	999
トリガ CI .....	1000
ジョブ・パラメータ .....	1000
トリガ CI データ .....	1001
検出された CIT .....	1001

## 概要

ネットワーク・モジュールのジョブをアクティブ化すると、ホスト・マシンへのシェル接続が確立されます。ディスカバリは、最初の有効な接続が検出されるまで、SSH, Telnet, NTCMD の各プロトコルを使用してリモート・マシンへの接続を試みます。

モジュールには、次のジョブが含まれています。

- **Host Connection by Shell** : SSH, Telnet, NTCMD, Universal Discovery の各プロトコルを使用して、リモート・マシンへの接続を確立します。このジョブは、ホスト・タイプ、OS 情報、ネットワーク接続情報を検出します。詳細については、[「Host Connection by Shell の検出方法」\(969ページ\)](#)を参照してください。
- **Host Connection by SNMP** : SNMP プロトコルを使用してデータ・センタ・マシンへの接続を試みることで、SNMP エージェントを検出します(データ・センタ・マシンの IP アドレスは、前もって発見され、IpAddress CI として UCMDB に存在するものです)。SNMP エージェントの検出後、関連する OID に従って、目的のホスト・クラス(Windows, UNIX, ルータなど)を更新します。詳細については、[「Host Connection by SNMP の検出方法」\(970ページ\)](#)を参照してください。
- **Host Connection by WMI** : WMI プロトコルを使用してリモート・マシンへの接続を確立し、ホスト・タイプ、OS 情報、ネットワーク接続情報を検出します。詳細については、[「Host Connection by WMI の検出方法」\(972ページ\)](#)を参照してください。
- **Client Connection by SNMP** : Data Flow Probe (または管理ゾーン)で構成されたすべてのクライアント範囲に対して ping スweep を実行することにより、SNMP エージェントを検出します。これが成功した場合は、SNMP プロトコルを使用して IP アドレスに接続し、関連する OID に従って目的のホスト・クラス(Windows, UNIX, ルータなど)を更新します。詳細については、[「Client Connection by SNMP の検出方法」\(972ページ\)](#)を参照してください。

ウィザードを使用してネットワークを検出する方法の詳細については、『HP UCMDB Discovery and Integrations Content Guide - Discovery Activities』ドキュメントの「Infrastructure Discovery Wizard」を参照してください。

各ジョブのディスカバリ・メカニズムについては、次を参照してください。

- **Host Connection by Shell** : [「ディスカバリ・メカニズム」\(973ページ\)](#)
- **Host Connection by SNMP** : [「ディスカバリ・メカニズム」\(987ページ\)](#)

- **Host Connection by WMI** : [「ディスカバリ・メカニズム」\(993ページ\)](#)
- **Client Connection by SNMP** : [「ディスカバリ・メカニズム」\(999ページ\)](#)

## Host Connection by Shell の検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、次のプロトコルを使用します。

- NTCMD プロトコル
- SSH プロトコル
- Telnet プロトコル
- Universal Discovery プロトコル

**注:** SSH サーバを実行している Windows マシンを検出するには、プロトコルの **Shell コマンドセパレータ**を自動検出に設定します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - Host Connection by Shell ジョブ

F-Secure アプリケーションを実行中の SSH サーバがインストールされている Windows マシンを検出するには **Host Connection by Shell** ジョブを実行しますが、その場合、F-Secure に対して次の変更を行う必要があります。

- F-Secure サービスを完全に停止する。
- F-Secure の残ったプロセス(**fssh\*** プロセス)が実行中でないことを確認する。
- **sshd2\_config** ファイル(F-Secure インストール・ディレクトリにある F-Secure 構成ファイル)で次の行を変更する。
  - **DoubleBackspace** 設定値は **no**(DoubleBackspace no)とする。
  - **EmulationType** 設定値は **raw**(EmulationType raw)とする。
  - **EmulationTypeForCommands** 設定値は **raw**(EmulationTypeForCommands raw)とする。

- 変更した **sshd2\_config** ファイルを保存する。
- F-Secure サービスを再起動する。

**注:** Data Flow Probe では、リモート SSH サーバ・プロバイダが **Open-SSH** または **F-Secure** である場合のみ、リモート Windows マシンへの SSH ベースの接続が有効になります。

**Open-SSH**(Windows, UNIX, および Linux オペレーティング・システム用の SSH サーバとなる)の場合、DFM はいずれのオペレーティング・システムにおいても Open-SSH バージョンが 3.7.1 以降の場合のみ Open-SSH への接続をサポートします。

### 3. ディスカバリの実行

**Host Connection by Shell** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

**注:** Data Flow Probe では、リモート SSH サーバ・プロバイダが **Open-SSH** または **F-Secure** である場合のみ、リモート Windows マシンへの SSH ベースの接続が有効になります。

**Open-SSH**(Windows, UNIX, および Linux オペレーティング・システム用の SSH サーバとなる)の場合、DFM はいずれのオペレーティング・システムにおいても Open-SSH バージョンが 3.7.1 以降の場合のみ Open-SSH への接続をサポートします。

## Host Connection by SNMP の検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは SNMP プロトコルを使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

**Host Connection by SNMP** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Host Connection by WMI の検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは WMI プロトコルを使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

**Host Connection by WMI** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Client Connection by SNMP の検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは SNMP プロトコルを使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

**Client Connection by SNMP** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。



## Host Connection by Shell ジョブ

本項では次の項について説明します。

ディスカバリ・メカニズム .....	973
トリガ・クエリ .....	981
ジョブ・パラメータ .....	982
アダプタ .....	983
検出された CIT .....	983
トラブルシューティングおよび制限事項 .....	985

## ディスカバリ・メカニズム

ディスカバリのメカニズムは、ディスカバリ対象のコンポーネントが Windows マシン、UNIX ベース・マシン、または Nexus マシンのうち、どのマシンにインストールされているかによって異なります。DFM プロセスの詳細については、以下を参照してください。

- [「Windows のプロセス」\(974ページ\)](#)
- [「UNIX ベースのプロセス」\(976ページ\)](#)
- [「Nexus のプロセス」\(981ページ\)](#)

### 注:

- DFM は、プロトコルに定義された資格情報すべてを試行して、いずれかの資格情報を通じて正常な接続を試みます。接続が成功した場合、ディスカバリは最後に成功した資格情報を記憶してキャッシュに入れ、次のディスカバリ実行時に再利用します。
- 資格情報(この宛先で最後に使用したもの)が存在しない場合、DFM は設定されたすべてのシェル資格情報のリストを反復処理します。

この反復処理で、DFM は SSH > Telnet > NTCMD の順に各プロトコルの資格情報を使用して、検出した宛先への接続を試みます。

検出した宛先の IP アドレスが資格情報の IP 範囲に含まれない場合、DFM はその資格情報のエントリをスキップします。

次の場合、DFM は、検出した宛先のプロトコルの使用を直ちに停止します。

- リモート・マシン上にエージェントが存在しない
- 接続が拒否される
- 接続がタイムアウトする
- ソケットを開くときに IO 例外が発生する

次の場合、DFM は設定された資格情報のリストの反復処理を停止します。

- 接続が正常に確立された
- 設定されたすべての資格情報を使用してみたが、検出した宛先に接続できない

## Windows のプロセス

本項では、Windows マシンに存在するコンポーネントを検出するために DFM が実行するワークフローの一部について説明します。

1. DFM はホスト属性 (OS の名前、バージョン、ビルド番号、サービス・パック、インストール・タイプ) を検出します。まず、DFM は次の最初の手順を使用してホスト属性を検出します。最初の手順が失敗した場合、DFM は引き続き次の手順を実行します。

- a. WMIC "OS" オブジェクト:

### フル・コマンド

```
'wmic os get caption, otherTypeDescription, version, buildnumber,
csdversion /format:list < %SystemRoot%\win.ini'
```

- b. Windows レジストリ:

### フル・クエリ

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion VER
command; %SYSTEMROOT%\system32\prodspec.ini processing
```

2. BIOS UUID(wmic)を取得します。

### フル・コマンド

```
'wmic path win32_ComputerSystemProduct get uuid /format:list < %SystemRoot%\win.ini'
```

3. デフォルト・ゲートウェイ(**netstat**)を取得します。

#### フル・コマンド

```
'netstat -r -n'
```

4. DNS サーバの IP(**ipconfig**)を取得します。

5. 起動日を定義します。

#### フル・コマンド

```
'wmic OS Get LastBootUpTime /format:list < %SystemRoot%\win.ini'
```

6. ネットワーク・インタフェースを取得します。最初に **wmic** コマンドを使用して、インタフェースに関するより詳細な情報を取得します。これに失敗した場合、**ipconfig** コマンドの出力を使用します。
  - a. NICCONFIG オブジェクトのクエリを実行し、MAC アドレス、IP アドレス、インタフェースの説明、サブネット IP、動的または静的フラグに関する情報を取得します。

#### フル・コマンド

```
'wmic nicconfig where "MACAddress <> NULL" get  
IPAddress,MACAddress,IPSubnet,Description,DhcpEnabled /format:list <  
%SystemRoot%\win.ini'
```

- b. IP フィルタリング: 不正な IP とローカルの IP は無視されます。
7. DFM は、宛先の IP がローカルかどうかをチェックします。ローカルの場合、DFM はそのホストと IP のみをレポートします。ローカルでない場合、次の情報をレポートします。
  - a. DFM はネットワーク・インタフェースをレポートします。ただし、次のインタフェースを除きます。
    - MAC アドレスを持たないインタフェース。
    - ループバック、ワイヤレス、仮想、WAN ミニポート、RAS ASYNC、Bluetooth、FireWire、VPN、IPv6 トンネリングのいずれかのタイプに属するインタフェース。
    - VMware インタフェース(**globalSettings.xml** 構成ファイルで **ignoreVmwareInterfaces** が **true** に設定されている場合)。
  - b. DFM はネットワーク、IP、および対応するリンクをレポートします。

## UNIX ベースのプロセス

本項では、UNIX ベース・マシンに存在するコンポーネントを検出するために DFM が実行するワークフローの一部について説明します。DFM はオペレーティング・システムを取得します。詳細については、次の各オペレーティング・システムで DFM が何を検出するかに関する説明を参照してください。

- [「AIX」\(976ページ\)](#)
- [「FreeBSD」\(977ページ\)](#)
- [「HPUX」\(978ページ\)](#)
- [「LINUX」\(978ページ\)](#)
- [「OpenBSD」\(979ページ\)](#)
- [「SunOs」\(980ページ\)](#)
- [「VMKernel」\(980ページ\)](#)

フル・コマンド : 'uname -a'

### 注:

ディスカバリをレポートする前に、DFM は以下を確認します。

- 宛先 IP が仮想アドレスの場合、IP とホストのみがレポートされます。
- OS が ZLinux で、ホスト・モデルが **s390x** の場合、ホストは IP とドメイン名によって定義されます。
- インタフェースの MAC アドレスが無効な場合、DFM はそのインタフェースをレポートしません。

## AIX

DFM は以下を検出します。

1. DHCP が有効になっているネットワーク・インタフェース(**ps**)。

フル・コマンド : 'ps -aef | grep dhcpcd | grep -v grep'

2. ネットワーク・インタフェース(MAC アドレス, 名前, 説明)(**lsdev, entstat**)。

**フル・コマンド** : 'lsdev -Cc adapter -S | egrep ^ent'

3. IP(**ifconfig**)。

**フル・コマンド** : 'ifconfig -a inet'

4. DFM は, FreeBSD の場合と同じ方法で, 起動日, ドメイン名, およびデフォルト・ゲートウェイを定義します。

5. モデルおよびベンダ(**uname**)。

**フル・コマンド** : 'uname -M'

6. シリアル番号(**lsattr**)。

7. OS バージョン(**oslevel**)。

## FreeBSD

DFM は以下を検出します。

1. DHCP が有効になっているインタフェース(**ps**)。

**フル・コマンド** : 'ps aux | grep dhclient | grep -v grep'

2. 起動日 (**uptime**)。

3. ネットワーク・インタフェース(**name**, **MAC**, **IP**, **network mask**, **DHCPEnabled flag**)および IP (**ifconfig**)。

**フル・コマンド** : 'ifconfig -a'

ホストは, ネットワーク・インタフェースの中で最も低い MAC アドレスによって定義されます。

4. OS バージョンおよびホスト・モデル(**uname**)。

**フル・コマンド** :

'uname -r' : バージョンの場合

'uname -m' : モデルの場合

5. ドメイン名 (**domainname**)。

フィルタリングされた名 前のみをレポート: '(none)', 'localdomain'

6. BIOS UUID(**dmidecode**)。

フル・コマンド: 'dmidecode | grep UUID'

7. デフォルト・ゲートウェイ(**netstat**)。

フル・コマンド: 'netstat -r -n'

## HPUX

1. DFM は、次 のどちらかの方法を使用してネットワーク・インタフェースを検出します。

- a. **nwmgr**

- b. **lanscan**(**nwmgr** が失敗した場合)

2. DFM は、検出されたインタフェースのエイリアス(**netstat**)を取得します。

フル・コマンド: 'netstat -I'

3. DFM はインタフェースごとに IP(**ifconfig**)を取得します。

4. DFM はホスト・モデル, 起動日, OS バージョン, シリアル番号, およびデフォルト・ゲートウェイを検出します。

5. DFM は OS フレーバー(**swlist**)を検出します。

フル・コマンド: 'swlist | grep -E "HPUX.\*?OE"'

## LINUX

DFM は以下を検出します。

1. DHCP が有効になっているネットワーク・インタフェース(**ps**)。

フル・コマンド: 'ps aux | grep dhclient | grep -v grep'

2. IP およびネットワーク・インタフェース(MAC アドレス, 名前, 説明)(**ifconfig**)。

フル・コマンド: 'ifconfig -a'

3. 起動日, シリアル番号 (**dmidecode**), OS バージョン, ホスト・モデル, ドメイン名, およびデフォルト・ゲートウェイ。

4. HMC(ハードウェア管理コンソール)に関する情報およびその IP(**lshmc**)。

フル・コマンド : 'lshmc -V'

5. BIOS UUID(**dmidecode**)。

フル・コマンド : 'dmidecode | grep UUID'

6. OS フレーバー(**redhat-release**)。

フル・コマンド : 'cat /etc/redhat-release'

## OpenBSD

DFM は以下を検出します。

1. DHCP が有効になっているインタフェース(**ps**)。

フル・コマンド : 'ps aux | grep dhclient | grep -v grep'

2. 起動日 (**uptime**)。

3. ネットワーク・インタフェース(**name**, **MAC**, **IP**, **network mask**, **DHCP enabled flag**)および IP (**ifconfig**)。

フル・コマンド : 'ifconfig -a'

ホストは, ネットワーク・インタフェースの中で最も低い MAC アドレスによって定義されます。

4. OS バージョンおよびホスト・モデル(**uname**)。

フル・コマンド

'uname -r' : バージョンの場合

'uname -m' : モデルの場合

5. ドメイン名 (**domainname**)。

フィルタリングされた名前のみをレポート : '(none)', 'localdomain'

6. BIOS UUID(**dmidecode**)。

フル・コマンド : 'dmidecode | grep UUID'

7. デフォルト・ゲートウェイ(**netstat**)。

フル・コマンド : 'netstat -r -n'

## SunOs

DFM は以下を検出します。

1. ネットワーク・インタフェース(**netstat**)。

フル・コマンド : 'netstat -np'

2. IP アドレス。

フル・コマンド : 'ifconfig -a'

3. 起動日, ドメイン名, BIOS UUID, およびデフォルト・ゲートウェイ。

4. OS のバージョンおよびリリース(**uname**)。

フル・コマンド : 'uname -rv'

5. ホスト・モデル(**prtdiag**)。

6. 製造元(**showrev**)。

7. シリアル番号(**dmidecode**)。

フル・コマンド : 'dmidecode | grep UUID'

## VMKernel

DFM は以下を検出します。

1. ネットワーク・インタフェース(MAC アドレス, 名前)および IP(**esxcfg-vmknics**)。

フル・コマンド : 'esxcfg-vmknics -l'

2. 起動日, OS バージョン, およびホスト・モデル。



3. ドメイン名 (**esxcfg-info**)。

フル・コマンド : 'esxcfg-info | grep Domain'

4. BIOS UUID(**esxcfg-info**)。

フル・コマンド : 'esxcfg-info | grep \'BIOS UUID\'

5. シリアル番号 (**esxcfg-info**)。

フル・コマンド : 'esxcfg-info -w | grep \'Serial Number\''

6. デフォルト・ゲートウェイ(**esxcfg-route**)。

7. OS フレーバー(**vmware**)。

フル・コマンド : 'vmware -v'

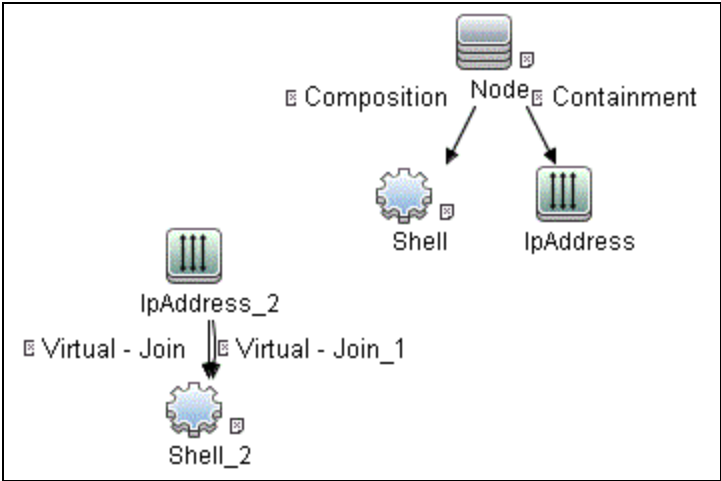
## Nexus のプロセス

本項では、Nexus マシンに存在するコンポーネントを検出するために DFM が実行するワークフローの一部について説明します。

1. DFM はコマンド **sh hostname** を使用してホスト名を取得します。
2. DFM はコマンド **sh ver** を使用してスイッチのバージョン、ビルド、および機能の情報を取得します。
3. DFM はコマンド **sh hosts** を使用して dns サーバおよびローカル・ホストのデータを取得します。
4. DFM はコマンド **sh int** を使用してインタフェースおよび設定された IP の情報を取得します。

## トリガ・クエリ

- **トリガ CI** : IP アドレス。
- **トリガ TQL** : DFM は、このクエリを使用して、シェルを持たない IP、または同じ IP のシェルを使用して再接続する IP を取得します。



・ ノード条件 :

- IP ノード

Probe Name NOT null (IP Is Broadcast 等しい false OR IP Is Broadcast NOT null)

ジョブ・パラメータ

パラメータ	説明
codepage	検出されたマシン・コード・ページ。 標準設定 : NA。
enableStamping	管理対象コンピュータで <b>ud_unique_id</b> をスタンプするかどうかを決定する。 標準設定 : false。
language	検出されたマシンの言語。
onlyStampingClient	クライアント・マシンのみをスタンプするかどうかを決定する。false に設定した場合、データ・センタとクライアント・マシンの両方がスタンプされる。 標準設定 : true。
udaConnectionOrder	プロトコル接続順序での UD の位置。指定できる値は、 <b>first</b> , <b>last</b> , および <b>none</b> 。 標準設定 : last。
useAIXhwid	IBM AIX マシンをハードウェア ID で識別するかどうかを指定する。 <b>true</b> に設定して、SNMP ディスカバリとともに使用すると、ホストが重

パラメータ	説明
	<p>複して作成される場合がある。<b>false</b> に設定した場合、AIX LPAR は検出されない。</p> <p><b>標準設定</b> : false。</p>
<b>useLastSuccessConnection</b>	<p>最後に接続に成功したプロトコルを使用するかどうかを指定する。<b>true</b> に設定すると、このジョブは <b>udaConnectionOrder</b> の値にかかわらず最後に成功した接続を使用する。</p> <p><b>標準設定</b> : true。</p>

## アダプタ

トリガ CI データ:

名前	値
connected_os_credentials_id	\${SHELL.connected_os_credentials_id:NA}
host_cmdbid	\${NA}
host_key	\${HOST.host_key:NA}
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}
ip_mac_address	\${SOURCE.arp_mac:NA}
mac_addrs	\${NA}

## 検出された CIT

- ATM Switch
- Composition
- Containment
- DnsServer
- HP NonStop
- IPMP Group

- Interface
- IpAddress
- IpSubnet
- Membership
- NTCMD
- Node
- OpenVMS
- Parent
- Realization
- Remote Access Service
- Router
- RunningSoftware
- SEA Adapter
- SNMP
- SSH
- Switch
- Telnet
- Terminal Server
- UDA
- Unix
- Usage
- Windows

## トラブルシューティングおよび制限事項

### トラブルシューティング

- **問題 :** **Host Connection by Shell** ジョブの実行中に次のエラーが表示される場合があります。

Error : Multiple connections to a server or shared resource by the same user, using more than one user name, are not allowed.

**解決策 :** 原因として、次に示す NetBIOS プロトコル制限のいずれかが考えられます。

- ネットワーク共有が実際には使用中ではないにもかかわらず、使用中であると認識されています。つまり、セッションがフリーズしています。この場合は、次のコマンドを試みます。

```
net use * /delete
```

- ユーザ名がローカル・マシンのユーザ名にバインドされている別のユーザによってネットワーク共有が使用されています。この場合は、リモート・マシンのセキュリティ・ポリシーを再構成するか、別のユーザの作業が完了するのを待ちます。
- **問題 :** **HPCmd Commands Execution Context** に **User** が設定されている場合、ユーザのアカウントに[サービスとしてログオン]の権限がないと、**Host Connection by Shell** ジョブで NTCMD ディスカバリが失敗します。

**解決策 :** ユーザのアカウントに[サービスとしてログオン]の権限が必要です。[サービスとしてログオン]の権限を持つようにユーザを構成する方法の詳細については、[http://technet.microsoft.com/ja-jp/library/cc739424\(v=ws.10\).aspx](http://technet.microsoft.com/ja-jp/library/cc739424(v=ws.10).aspx) を参照してください。

- **問題 :** **アクセス拒否のエラー・メッセージ**が表示されます。

NTCMD プロトコルで Windows 2008 SP2 の宛先を検出中に、次のエラー・メッセージが表示される場合があります。

```
Connecting to remote server failed with the following error message: Access is denied. For more information, see the about_Remote_Troubleshooting Help topic.
```

このメッセージは、宛先ホストの検出を試行しているユーザがローカルの Administrator ユーザでない場合に 표시됩니다(ユーザが Administrator グループのメンバであるかどうかは関係ありません)。

**解決策 :** この問題を解決するには、セキュリティ・ポリシーを追加で構成する必要があります。

**LocalAccountTokenPolicy** キーを変更し、Administrator グループのユーザが管理者権限を使用してリモート接続できるようにする必要があります。検出するホストの PowerShell で、次のコマンドを実行します。

```
Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\  
Policies\System -Name LocalAccountTokenFilterPolicy -Value 1 -Type DWord
```

この特殊なケースの詳細については、「他のドメインの管理者のリモート処理機能を有効にする方法」(<http://technet.microsoft.com/ja-jp/library/dd347642.aspx>)を参照してください。

### 制限事項

- **制限事項** : インタフェースの MAC アドレスが 0 の場合、そのインタフェースまたはそのインタフェースに割り当てられた IP アドレスはジョブでレポートされません。
- **制限事項** : このディスカバリで PAE 状態のレポートがサポートされるのは、Windows オペレーティング・システムの場合に限られます。

## Host Connection by SNMP ジョブ

本項では次の項について説明します。

ディスカバリ・メカニズム .....	987
トリガ・クエリ .....	988
ジョブ・パラメータ .....	989
アダプタ .....	990
検出された CIT .....	990
トラブルシューティングおよび制限事項 .....	991

### ディスカバリ・メカニズム

1. DFM は、プロトコルに定義された資格情報すべてを試行して、いずれかの資格情報を通じて正常な接続を試みます。接続が成功した場合、ディスカバリは最後に成功した資格情報を記憶してキャッシュに入れ、次のディスカバリ実行時に再利用します。
2. DFM は、SNMP クエリを実行し、クラス名、ベンダ名、ホスト OS 名、ホスト・モデル、ホストのバージョン、ホストのリリース日を取得します。

使用する OID :

SNMP MIB-2 System 1.3.6.1.2.1.1

SNMP MIB-2 Interfaces 1.3.6.1.2.1.20

3.

x3x.x3.x.xxxxxxxxxxxx x

The vendor's authoritative identification of the network management subsystem obtained from the system table.

3. DFM は、ホスト IP とマスクを取得します。

使用する OID :

ipAdEntNetMask (1.3.6.1.2.1.4.20.1.3) for subnet mask

ipAdEntBcastAddr (1.3.6.1.2.1.4.20.1.4) for the least-significant bit in the IP broadcast address

ipAdEntIfIndex (1.3.6.1.2.1.4.20.1.2) for the index value which uniquely identifies the interface

4. DFM は、ネットワーク・インタフェース情報を取得します。

OID (1.3.6.1.2.1.2.2.1) - an interface entry containing objects at the subnetwork layer and below for a particular interface.

5. DFM は、デフォルト・ゲートウェイを取得します。

Used OIDs:

```
ipRouteDest (1.3.6.1.2.1.4.21.1.1) -
for the destination IP address of this route
ipRouteMask (1.3.6.1.2.1.4.21.1.11) -
for the mask
ipRouteDest (1.3.6.1.2.1.4.21.1.1) -
for the destination IP address of this route
ipRouteMetric1 (1.3.6.1.2.1.4.21.1.3) -
for the primary routing metric for this route
ipRouteNextHop (1.3.6.1.2.1.4.21.1.7) -
for the IP address of the next hop of this route
```

6. DFM は、ホストのシリアル番号を取得します。シリアル番号は、パブリック MIB から取得します(パブリック MIB から取得できない場合は、プライベート MIB から取得します)。いずれの場合も、OID を取得します。このジョブは、幅広いデバイスをサポートしています。ただし、シリアル番号がわかっている場合でも DFM がシリアル番号を抽出できない場合は、HP が今後のコンテンツ・パックまたは更新にサポートを追加できるように、サポート・ケースを登録してください(MIB ウォークから必ずすべての詳細を入力してください)。
7. 可能な場合、DFM は、ホストのリモート管理カードを取得します。たとえば、HP iLO カードや Dell DRAC カードを取得します。

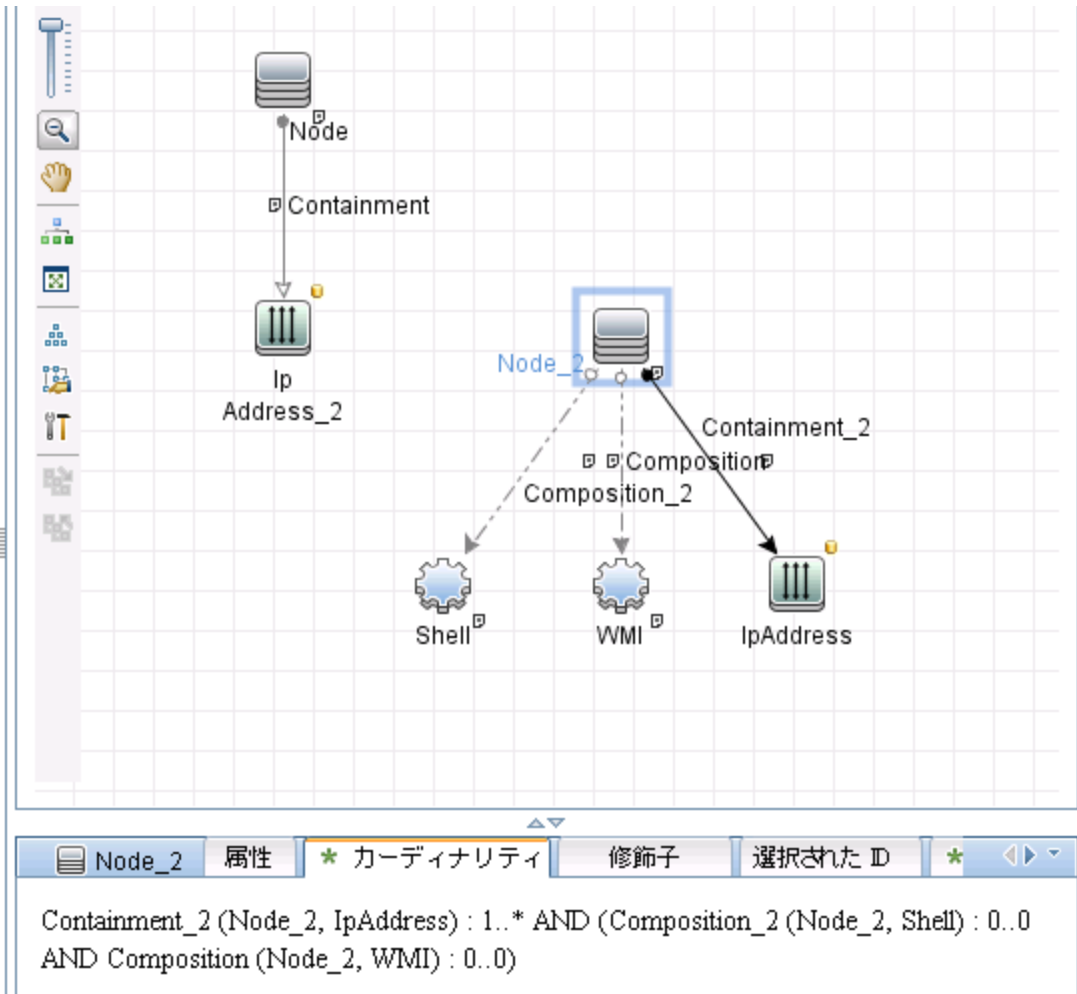
Get iLO cards from OID table 1.3.6.1.4.1.232.9.2.5.1

Get DRAC cards from OID table 1.3.6.1.4.1.674.10892.1.1900.10

## トリガ・クエリ

- **トリガ CI**: IP アドレス。
- **トリガ TQL**: このクエリでは、(a) Containment リンクでノードに接続されていない IP、または(b) シェルも WMI エージェントも持たないノードに接続されている IP を取得できます。





・ ノード条件。

◦ IP ノード :

NOT IP Lease Time 等しい Short  
AND NOT IP Probe Name null  
AND (IP Is Broadcast 等しい false  
OR IP Is Broadcast null)

ジョブ・パラメータ

パラメータ	説明
reportInterfaceName	ジョブにノードのインタフェース名をレポートさせるかどうかを指定できる。 標準設定 : True。

## アダプタ

- トリガ CI データ:

名前	値
host_cmdbid	\${NA}
host_key	\${HOST.host_key:NA}
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}
ip_mac	\${SOURCE.arp_mac:NA}
mac_addrs	\${NA}

## 検出された CIT

- ATM Switch
- Composition
- Containment
- Fibre Channel Switch
- Firewall
- Interface
- IpAddress
- IpSubnet
- LDOM Virtual Switch
- Load Balancer
- Mainframe CPC
- Marconi ATM Switch

- Membership
- Net Device
- Net Printer
- Node
- OpenVMS
- Parent
- Remote Access Service
- Router
- SNMP
- Switch
- Terminal Server
- Unix
- VMware Virtual Switch
- Windows

## トラブルシューティングおよび制限事項

- **問題 : Host Connection by SNMP** ジョブまたは **Host Networking by SNMP** ジョブの実行後に、多くの警告メッセージが表示される。

Detected multiple updates in bulk - found attribute:  
'interface\_description' on current CIT:'interface'


これらのメッセージは無視しても問題ありません。メッセージが表示されないようにするには、**globalSettings.xml** ファイルの **multipleUpdateIgnoreTypes** パラメータを変更します。

```
<!--multipleUpdateIgnoreTypes
- don't check multiple updates for the following types-->
<property name="multipleUpdateIgnoreTypes">
process,clientserver,node</property>
```

**interface** CIT を CIT のリストに追加して、無視されるようにします。

- **問題** : ホスト接続 ディスカバリは、**Host Connection by Shell**, **Host Connection by WMI**, **Host Connection by SNMP** という順にワークフローを使用します。そのため、**Host Connection by Shell** が成功した場合、以降のジョブはいずれも完了しません。また、**Host Connection by WMI** が成功した場合、**Host Connection by SNMP** も完了しません。

**解決策** : この制限を排除するには、ジョブのトリガ・クエリを変更します。

- **[Host Connection by SNMP]**を選択します。
- **[プロパティ]**タブを選択します。
- トリガ・クエリ **ip\_with\_snmp\_or\_without\_host** を削除します。
- **[トリガ クエリ]**セクションの  ボタンをクリックし、新しいトリガ・クエリを作成します。**[ディスカバリ クエリの選択]**ダイアログ・ボックスが表示されます。
- リストから **[ip]**を選択し、**[OK]**をクリックします。

# Host Connection by WMI ジョブ

本 項 では次の項について説明します。

ディスカバリ・メカニズム .....	993
トリガ・クエリ .....	995
ジョブ・パラメータ .....	996
アダプタ .....	996
検出された CIT .....	997
トラブルシューティングおよび制 限 事 項 .....	997

## ディスカバリ・メカニズム

1. DFM は WMI プロトコルについて定 義された資 格 情 報を確認し、そのいずれかを介して正 常な接 続を試みます。
2. DFM は Win32\_ComputerSystem の WMI クエリを実行して、マシン名を取得します。

**WMI クエリ:**

```
select Name from Win32_ComputerSystem
```

DFM は Win32\_NetworkAdapterConfiguration の WMI クエリを実行して、次のインタフェース情 報を取得します: IP アドレス, MAC アドレス, サブネット IP, 説明, および DHCP enabled 属 性。 DFM はインタフェースのローカル IP を無視します。

**WMI クエリ:**

```
'SELECT DnsHostName,IPAddress,MACAddress,IPSubnet,Description,  
DhcpEnabled FROM Win32_NetworkAdapterConfiguration  
WHERE MACAddress <> NULL'
```

3. DFM は宛 先 IP アドレスがローカル IP アドレスであるかどうかを確認します。ローカル IP アドレスの場 合, DFM は IP とホストのみをレポートします。

DFM がこの方法でホストを検出できない場合, DFM は検出されたネットワーク・インタフェースのう ち最小の MAC アドレスによって定 義されたホストを作成しようとします。有効な MAC アドレスを提 供するインタフェースが存在しない場合, DFM は宛 先 IP アドレスによってホストを定 義します。

MAC アドレスは、次のルールに従うインタフェースでのみ使用されます。

- インタフェースに有効な MAC アドレスがある。
  - インタフェースが次のいずれかのタイプに属していない: ループバック, ワイヤレス, 仮想, WAN ミニポート, RAS ASYNC, Bluetooth, FireWire, VPN, または IPv6 トンネリング。
  - コンポーネントが VMware インタフェースではなく, **globalSettings.xml** 構成ファイルで **ignoreVmwareInterfaces** オプションが 1 に設定されていない。
4. DFM は Win32\_OperatingSystem をクエリして、ホスト・ベンダ、OS 名、バージョン、ブート時刻、およびインストール・タイプを取得します。

#### WMI クエリ:

```
select Caption,Version,
ServicePackMajorVersion,ServicePackMinorVersion,
BuildNumber,Organization,RegisteredUser,TotalVisibleMemorySize,
LastBootUpTime,OtherTypeDescription from Win32_OperatingSystem
```

5. DFM は Win32\_IP4RouteTable をクエリしてデフォルト・ゲートウェイを取得します。

#### WMI クエリ:

```
select NextHop, Metric1 from Win32_IP4RouteTable Where destination
= '0.0.0.0' and mask = '0.0.0.0'
```

6. DFM は Win32\_ComputerSystem をクエリしてホスト製造元、プロセッサ数、ホスト・モデル、および OS ドメインを取得します。

#### WMI クエリ:

```
select Manufacturer,NumberOfProcessors,Model,Domain from
Win32_ComputerSystem
```

7. DFM は次の方法でシリアル番号を取得します。

- Win32\_BaseBoard をクエリする。

#### WMI クエリ:

```
SELECT SerialNumber FROM Win32_BaseBoard
```

- Win32\_SystemEnclosure をクエリする。

**WMI クエリ:**

```
SELECT SerialNumber,SMBIOSAssetTag FROM Win32_SystemEnclosure
```

8. DFM は Win32\_SystemEnclosure をクエリしてシステム・アセット・タグを取得します。

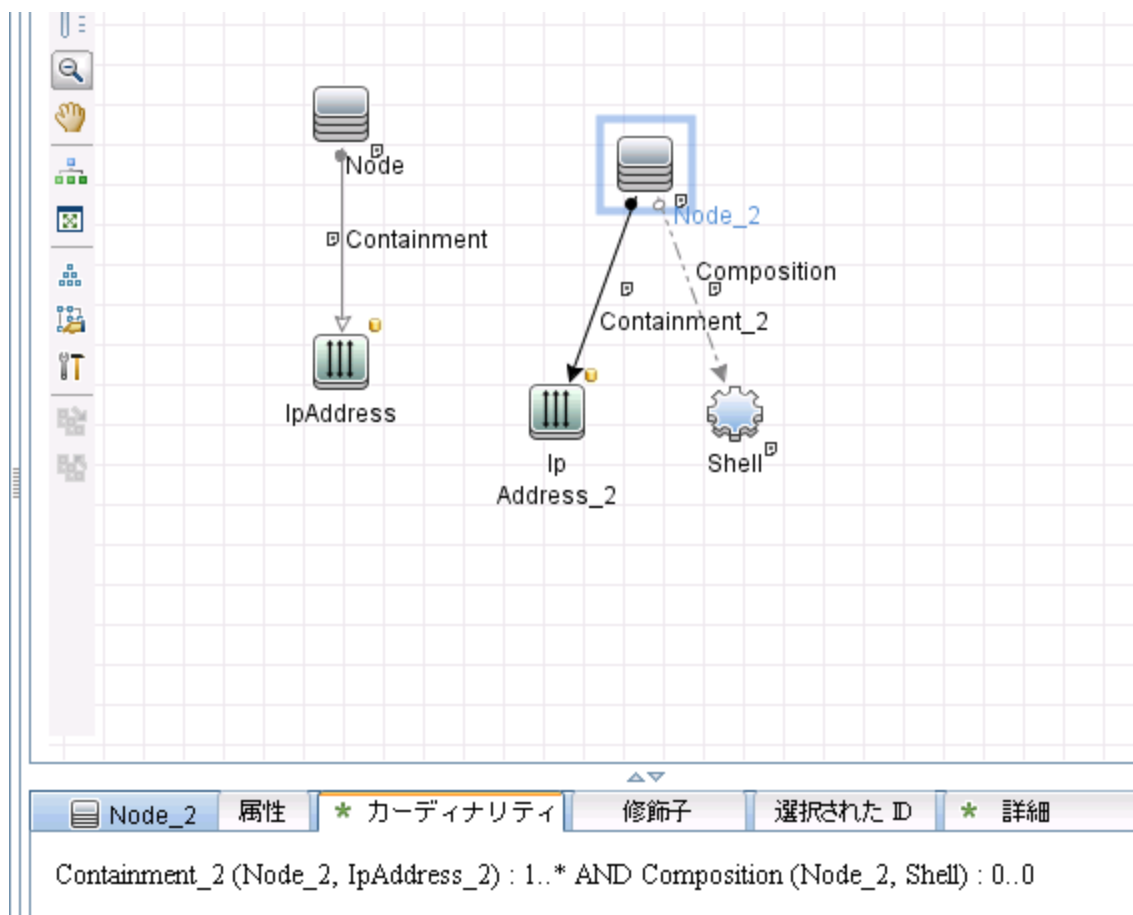
**WMI クエリ:**

```
SELECT SerialNumber,SMBIOSAssetTag FROM Win32_SystemEnclosure
```

9. 接続に成功した場合、DFM は前回の接続試行で生成された可能性のあるすべてのエラーおよび警告をクリアして、結果を返します。
10. 接続に失敗した場合、DFM はすべての試行が終わるまで次の WMI 資格情報で続行します。

## トリガ・クエリ

- **トリガ CI**: IP アドレス。
- **トリガ TQL**: このクエリにより、(a)Containment(包含)リンクによってノードに接続されていない、または (b)Shell Agent を持たないノードに接続されている IP を取得できます。



## ・ ノード条件。

### 。 IP ノード :

Probe Name NOT null  
 (IP Is Broadcast 等しい false OR IP Is Broadcast NOT null)

## ジョブ・パラメータ

なし。

## アダプタ

トリガ CI データ:



名前	値
host_cmdbid	\${HOST.root_id:NA}
host_key	\${HOST.host_key:NA}
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}
ip_mac_address	\${SOURCE.arp_mac:NA}
mac_addr	\${NA}

## 検出された CIT


- Composition
- Containment
- Interface
- IpAddress
- IpSubnet
- Membership
- Node
- Parent
- WMI

## トラブルシューティングおよび制限事項

- **問題** : ホスト接続 ディスカバリは, **Host Connection by Shell**, **Host Connection by WMI**, **Host Connection by SNMP** という順にワークフローを使用します。そのため, **Host Connection by Shell** が成功した場合, 以降のジョブはいずれも完了しません。また, **Host Connection by WMI** が成功した場合, **Host Connection by SNMP** も完了しません。

**解決策** : この制限を排除するには, ジョブのトリガ・クエリを変更します。

- **[Host Connection by WMI]**を選択します。
- **[プロパティ]**タブを選択します。

- トリガ・クエリ `ip_with_wmi_or_without_host` を削除します。
- [トリガ クエリ] セクションの  ボタンをクリックし、新しいトリガ・クエリを作成します。[ディスカバリ クエリの選択] ダイアログ・ボックスが表示されます。
- リストから[`ip`]を選択し、[OK]をクリックします。

## Client Connection by SNMP ジョブ

本項では次の項について説明します。

ディスカバリ・メカニズム .....	999
トリガ CI .....	1000
ジョブ・パラメータ .....	1000
トリガ CI データ .....	1001
検出された CIT .....	1001

### ディスカバリ・メカニズム

1. DFM は、クライアント・タイプの IP 範囲に対して ping スweep を実行し、アクティブなクライアント・マシンの詳細を取得します。
2. DFM は、プロトコルに定義された資格情報すべてを試行して、いずれかの資格情報を通じて正常な接続を試みます。接続が成功した場合、ディスカバリは最後に成功した資格情報を記憶してキャッシュに入れ、次のディスカバリ実行時に再利用します。
3. DFM は、SNMP クエリを実行し、クラス名、ベンダ名、ホスト OS 名、ホスト・モデル、ホストのバージョン、ホストのリリース日を取得します。

使用する OID :

SNMP MIB-2 System 1.3.6.1.2.1.1

SNMP MIB-2 Interfaces 1.3.6.1.2.1.20

3.

x3x.x3.x.xxxxxxxxxxxx x

The vendor's authoritative identification of the network management subsystem obtained from the system table.

4. DFM は、ホスト IP とマスクを取得します。

使用する OID :

ipAdEntNetMask (1.3.6.1.2.1.4.20.1.3) for subnet mask

ipAdEntBcastAddr (1.3.6.1.2.1.4.20.1.4) for the least-significant bit in the IP broadcast address

ipAdEntIfIndex (1.3.6.1.2.1.4.20.1.2) for the index value which uniquely identifies the interface

5. DFM は、ネットワーク・インタフェース情報を取得します。

OID (1.3.6.1.2.1.2.2.1) - an interface entry containing objects at the subnetwork layer and below for a particular interface.

6. DFM は、デフォルト・ゲートウェイを取得します。

Used OIDs:

```
ipRouteDest (1.3.6.1.2.1.4.21.1.1) -
for the destination IP address of this route
ipRouteMask (1.3.6.1.2.1.4.21.1.11) -
for the mask
ipRouteDest (1.3.6.1.2.1.4.21.1.1) -
for the destination IP address of this route
ipRouteMetric1 (1.3.6.1.2.1.4.21.1.3) -
for the primary routing metric for this route
ipRouteNextHop (1.3.6.1.2.1.4.21.1.7) -
for the IP address of the next hop of this route
```

7. DFM は、ホストのシリアル番号を取得します。シリアル番号は、パブリック MIB から取得します(パブリック MIB から取得できない場合は、プライベート MIB から取得します)。いずれの場合も、OID を取得します。このジョブは、幅広いデバイスをサポートしています。ただし、シリアル番号がわかっている場合でも DFM がシリアル番号を抽出できない場合は、HP が今後のコンテンツ・パックまたは更新にサポートを追加できるように、サポート・ケースを登録してください(MIB ウォークから必ずすべての詳細を入力してください)。

8. 可能な場合、DFM は、ホストのリモート管理カードを取得します。たとえば、HP iLO カードや Dell DRAC カードを取得します。

Get iLO cards from OID table 1.3.6.1.4.1.232.9.2.5.1

Get DRAC cards from OID table 1.3.6.1.4.1.674.10892.1.1900.10

## トリガ CI

Discovery Probe Gateway

## ジョブ・パラメータ

パラメータ	説明
excludePatternsList	セミコロンで区切られたワイルドカード・パターンのリスト。いずれかのパターンに一致した IP アドレスは、スキップされます。パターンには、数字、ドット、ワイルドカード * (0 個以上の文字に一致)、? (正確に 1 文字に一致)を含めることができます。
pingProtocol	選択した ping プロトコルを表す番号。1 は ICMP, 2 はエコー・ポート, 3 はそ

パラメータ	説明
	の両方を表します。 標準設定 : 1
range	ping を実行する IP アドレスの範囲。IP アドレスはセミコロンで区切ります。
retryDiscover	ジョブが ping を試みる回数。 標準設定 : 2
threadPoolSize	ポート 7 のエコーを実行するプール内のスレッド数。 標準設定 : 10
timeoutDiscover	ping のタイムアウト(ミリ秒) 標準設定 : 3,000

## トリガ CI データ

名前	値
probeName	\${SOURCE.name}

## 検出された CIT

- ATM Switch
- Composition
- Containment
- Fibre Channel Switch
- Firewall
- Interface
- IpAddress
- IpSubnet
- LDOM Virtual Switch

- Load Balancer
- Mainframe CPC
- Marconi ATM Switch
- Membership
- Net Device
- Net Printer
- Node
- OpenVMS
- Parent
- Remote Access Service
- Router
- SNMP
- Switch
- Terminal Server
- Unix
- VMware Virtual Switch
- Windows

## 第66章: DNS ゾーン・ディスカバリ

### 本章の内容

概要 .....	1004
サポートされるバージョン .....	1005
nslookup による DNS ゾーンの検出方法 .....	1006
DNS による DNS ゾーンの検出方法 .....	1007
DNS サーバで nslookup を使用してシェルでホストを検出する方法 .....	1007
DNS Zone by nslookup ジョブ .....	1008
アダプタ情報 .....	1009
DNS Zone by DNS ジョブ .....	1011
アダプタ情報 .....	1011
Hosts by Shell using nslookup on DNS Server ジョブ .....	1014
NSLOOKUP on DNS Server アダプタ .....	1015
ディスカバリ・メカニズム - Windows .....	1018
ディスカバリ・メカニズム - UNIX 系 .....	1020
用語集 .....	1021

## 概要

DNS ゾーン・ディスカバリでは、DNS ゾーンのとポロジとゾーンに属するレコードを取得します。ゾーンを転送するには、クエリを実行するマシンが、ネーム・サーバで設定されているホワイト・リストに含まれる必要があります。この方法では、プローブ・ゾーンの転送を許可するための特別な DNS サーバ設定が必要です。

レポートする必要があるゾーンを記録している特定のネーム・サーバ上でディスカバリ・メカニズムが次のようにトリガします。

1. **zoneList** パラメータをチェックして、エイリアス・レコードを転送するゾーンのリストを確認します。
2. **arpa**, **localhost**, または **'.'**(ルート)を含む名前のゾーンを無視します。
3. ゾーンごとに、**CNAME** タイプと **A** タイプ(2 番目の手順)のすべてのレコードを転送します。転送が失敗した場合、ゾーンはレポートされません。
4. Realization リンクを作成します。

詳細については、[「DNS Zone by nslookup ジョブ」\(1008ページ\)](#)を参照してください。

次の方法で DNS ゾーン・ディスカバリが実装されています。

- **DNS Zone by nslookup** ジョブは、DNS サーバに対してクエリを実行し、サーバ自体のゾーン・レコードを照会します。この方法では、シェル・アクセスが必要です。詳細については、[「nslookup による DNS ゾーンの検出方法」\(1006ページ\)](#)を参照してください。
- **DNS Zone by DNS** ジョブは、DNS サーバに対してクエリを実行し、Data Flow Probe マシンのゾーン・レコードを照会します。この方法では、プローブ・ゾーンの転送を許可するための特別な DNS サーバ設定が必要です。詳細については、[「DNS による DNS ゾーンの検出方法」\(1007ページ\)](#)を参照してください。

DNS サーバにシェル・アクセスを追加することや、構成ファイルに読み取りアクセスを追加することを管理者が望まない場合は、必須の **zoneList** アダプタ・パラメータで指定したゾーンを転送できます。詳細については、[「DNS Zone by nslookup ジョブ」\(1008ページ\)](#)を参照してください。

これらの実装は、同じとポロジを取得します。これらの実装には、クライアント・タイプ(サーバまたはプローブ)のみが異なる共通のディスカバリ・メカニズムがあります。

**注:** 取得するとポロジ・データの量は、特定のジョブのパラメータ・セットの影響を受ける場合があります。



## サポートされるバージョン

- Microsoft Windows 2000 Advanced Server 以降
- UNIX 系 OS の BIND 9 ネーム・サーバ

## nslookup による DNS ゾーンの検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、次のプロトコルを使用します。

- SSH プロトコル
- NTCMD プロトコル
- Telnet プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - プロトコル・パラメータ

- ターゲット・ホストで **sudo** を使用して実行するようにコマンドを設定した場合、[**プロトコルパラメータ**] ダイアログ・ボックスで次のフィールドを入力します。
  - **sudo パス** : **sudo** 実行可能ファイルのフル・パスと、実行可能ファイルの名前を入力します。実行可能ファイルをターゲット・オペレーティング・システムのいくつかの場所に配置する場合は、複数のエントリを追加できます。

例 : `sudo,/usr/bin/sudo,/bin/sudo`

- **コマンドリスト** : **sudo** (プレフィックス) を付けるコマンドのリストを入力します。

例 : `lspath,ifconfig`

- ディスカバ리를 アクティブ化する前に、次のコマンドを実行するためのすべての必要な権限をディスカバリ・ユーザが持っていることを確認します。

```
cat <path to named config file and its include files>
```

詳細については、『HP Universal CMDB Data Flow Management Guide』の「Protocol Parameter Dialog Box」を参照してください。

### 3. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行します。

- b. **Host Connection by Shell** ジョブを実行します。
- c. **Host Applications by Shell** ジョブを実行します。
- d. **DNS Zone by nslookup** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## DNS による DNS ゾーンの検出方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

ディスカバリは DNS プロトコルによって実行されます。ディスカバリを実行するには、次のようにセットアップを行います。

- すべての要求はプローブ・マシンから実行されます。したがって、指定されたゾーン・レコードの転送ができるサーバのリストに、このマシンを含める必要があります。ネーム・サーバの管理者は、プローブ・マシンからゾーンを転送するための権限を付与できます。
- 転送する必要があるゾーンのリストを提供します。詳細については、[「DNS Zone by nslookup ジョブ」\(1008ページ\)](#)を参照してください。

### 2. ディスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行します。
- b. **TCP ports** ジョブを実行します。
- c. **DNS Zone by DNS** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## DNS サーバで nslookup を使用してシェルでホストを検出する方法

本項の内容

## 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、次のプロトコルを使用します。

- NTCMD
- SSH
- Telnet
- UDA

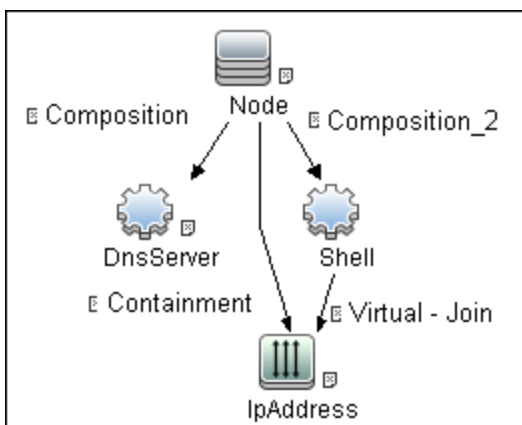
## 2. ディスカバリの実行

- Range IPs by ICMP** ジョブを実行します。
- Host Connection by Shell** ジョブを実行します。
- Hosts by Shell using nslookup on DNS Server** ジョブを実行します。

# DNS Zone by nslookup ジョブ

このジョブは、リモート・シェルを使用してサーバの名前をクエリすることで、DNS ゾーンのDNS リソース・レコード・トポロジを検出します。

## トリガ・クエリ



## CI 属性条件

CI	属性値
Shell	NOT Reference to the credentials dictionary entry null
IpAddress	NOT IP Probe Name null

## アダプタ情報

### ID

DNS\_Zone

### 入力 CIT

Shell

### 入力クエリ



### トリガ CI データ

名前	値
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
ip_address	\${SOURCE.application_ip}

### パラメータ

アダプタには次のパラメータが含まれます。

パラメータ	説明
reportBrokenAliases	<b>true</b> の場合、正規リソースを含まないエイリアスがレポートされる。エイリアスがアドレス・レコードまたは別のエイリアス・レコードを指していて、転送さ

パラメータ	説明
	れたデータ内でそのレコードが見つからない場合は、このパラメータが必要。標準設定値は <b>false</b> 。
zoneList	転送するゾーンのカンマ区切りのリストが含まれる。この属性は省略可能。標準設定では、ゾーン・リストが指定されていないため、ゾーンは自動的に決定される。
includeOutscopeIPs	パラメータが false に設定されている場合、プローブの範囲外にある IP アドレスはレポートされない。標準設定値は "false"。

使用スクリプト

- entity.py
- dns.py
- dns\_discoverer.py
- dns\_zone\_by\_remote\_shell.py

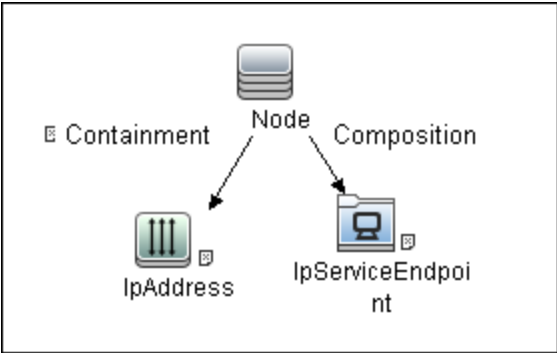
検出された CIT

- Composition
- DnsRecord
- DnsZone
- IpAddress
- Realization

## DNS Zone by DNS ジョブ

このジョブは、DNS プロトコルを使用してローカル・シェル(プローブ)からサーバ名 をクエリすることで、DNS ゾーンの DNS リソース・レコード・トポロジを検出します。

### トリガ・クエリ



### CI 属性条件

CI	属性値
IpServiceEndpoint	(IpServiceName 等しい dns OR ServiceNames 含む Idap) AND NOT BoundToIpAddress null

## アダプタ情報

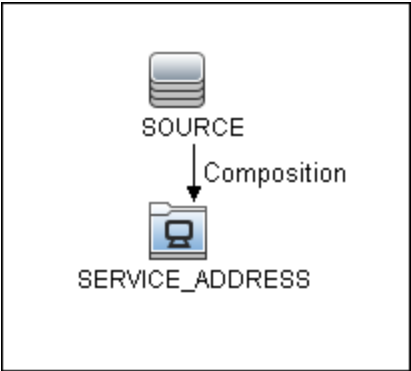
### ID

DNS\_Zone\_by\_DNS

### 入力 CIT

Node

入カクエリ



トリガ CI データ

名前	値
ip_address	\${SERVICE_ADDRESS.bound_to_ip_address}

使用スクリプト

- entity.py
- dns.py
- dns\_discoverer.py
- dns\_zone\_by\_probe\_shell.py

検出された CIT

- Composition
- DnsRecord
- DnsZone
- IpAddress
- Realization



パラメータ

名前	説明
includeOutscopeIPs	このパラメータが <b>false</b> に設定されている場合、プローブの範囲外にある IP はレポートされない。標準設定値は <b>false</b> 。
reportBrokenAliases	このパラメータが <b>true</b> に設定されている場合、正規リソースが存在しないエイリアスがレポートされる。
zoneList	転送するゾーンのカンマ区切りのリストが含まれる。これは必須属性。

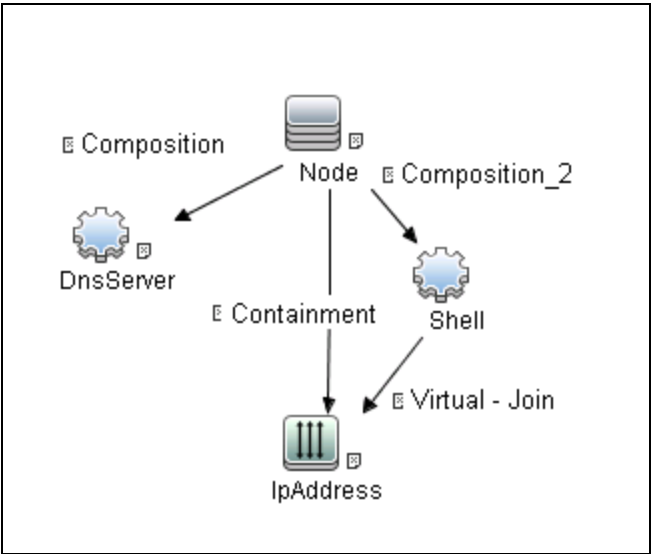
# Hosts by Shell using nslookup on DNS Server ジョブ

このジョブは、利用可能なすべての DNS サーバに対してクエリを実行することでホストを検出します。

## アダプタ

ID : NSLOOKUP\_on\_DNS\_Server

## トリガ TQL



ノード名	条件
IpAddress	NOT IP Probe Name null
Shell	NOT Reference to the credentials dictionary entry null

## パラメータ

名前	説明
DNSServerDomain	DNS サーバのドメイン。
DNSServerName	DNS サーバの名前。

名前	説明
discoverUnknownIPs	このフラグは、プローブの範囲外にある IP を検出するかどうかを決定する。

ディスカバリ・フロー

- 1. DNS サーバとの接続を確立します。
- 2. 入力パラメータ **DNSServerName** と **DNSServerDomain** を確認し、**nslookup** コマンドを実行して標準設定のサーバ名とそのドメインを要求することで、転送先の DNS サーバとドメインを決定します。
- 3. すべての IPv4(A)DNS レコードのリストを表示して、各レコードを処理します。
- 4. 表示された IP アドレスに基づいてホストをレポートします。

NSLOOKUP on DNS Server アダプタ

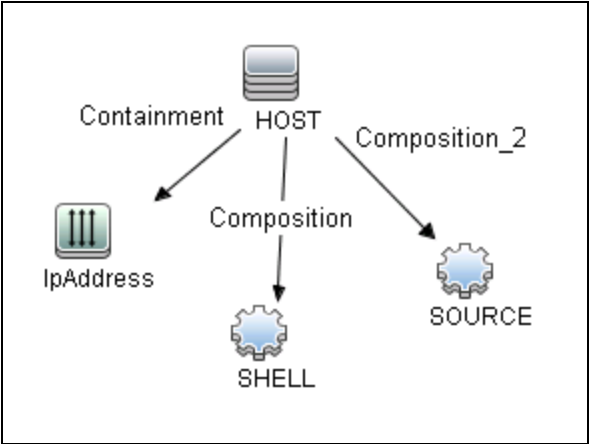
ID

NSLOOKUP\_on\_DNS\_Server

入力 CIT

Shell

入力 TQL



ノード名	条件
IpAddress	IpAddressType 等しい IPv4
Shell	CI Type 等しい ssh OR CI Type 等しい uda

## トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
codepage	\${SOURCE.codepage:NA}
credentialsId	\${SOURCE.credentials_id}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
ip_address	\${SOURCE.application_ip}
language	\${SOURCE.language:NA}
sshKeyPath	\${SHELL.ssh_keypath:NA}

## 使用スクリプト

- flow.py
- dns.py
- dns\_discoverer.py
- dns\_flow.py
- NSLOOKUP.py

## 検出された CIT

- Windows
- Node
- IpAddress
- Net Printer

- Unix
- Containment

グローバル構成ファイル

- globalSettings.xml

パラメータ

名 前	説 明
DNSServerDomain	DNS サーバのドメイン。
DNSServerName	DNS サーバの名 前。
discoverUnknownIPs	<b>true</b> の場 合，ジョブはプローブの範 囲 外 にある IP も検 出。

## ディスカバリ・メカニズム - Windows

本項では以下のコマンドについて説明します。

### Windows レジストリに対してクエリを実行し、ゾーン情報を照会する

#### コマンド

```
Reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS  
Server\Zones"
```

#### 出力

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS  
Server\Zones\104.24.172.in-addr.arpa  
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS  
Server\Zones\foo.bar.net  
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS  
Server\Zones\od5.lohika.com  
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS  
Server\Zones\ucmdb-ex.dot
```

#### マッピング

CMD 出力属性	CI 名	CI 属性
キー名	DNS Zone	Name

### ルート・ドメインのリストを表示してリソース・レコードを転送する

**nslookup** コマンドでゾーンのルート・ドメインを表示することで、**CNAME** タイプおよび **A** タイプのゾーン・リソース・レコードを転送します。

#### コマンド

```
echo ls -d <domain> | nslookup - <name server>
```

#### 出力

```
Ns-2.od5.lohika.com. CNAME dc05-2.od5.lohika.com  
  
od5.lohika.com. A 134.44.98.22  
ftp.od5.lohika.com. CNAME od5.lohika.com.
```

#### マッピング

CMD 出力属性	CI 名	CI 属性
1 番目の列	DNS Alias	Name
3 番目の列	DNS Alias	Canonical name

## ディスカバリ・メカニズム - UNIX 系

本項では以下のコマンドについて説明します。

### ネーム・サーバ構成ファイルを解析してゾーン情報を取得する

1. 対応するプロセスなど、コマンド内のネーム・サーバ構成ファイルに関する情報を見つけます。

#### コマンド

```
ps -ef | grep named | awk '{for(i=11; i < NF; i++) {printf("%s ", $i)}}printf("\n")}'
```

#### 出力

```
/usr/sbin/named -t /var/lib/named -u
```

#### マッピング

**-t** オプションが指定されたパスは、構成ファイルのパスです。

2. パスが認識された場合、ジョブはゾーンに関する情報を取得し、処理するファイルを含めます。標準設定のパスは **/etc/named.conf** および **/etc/namedb/named.conf** です。

#### コマンド

```
cat <configuration file path> | awk '/zone|include/ {print}'
```

#### 出力

```
zone "." in {  
zone "localhost" in {  
zone "od5.lohika.com" in {
```

#### マッピング

CMD 出力属性	CI 名	CI 属性
キー名	DNS Zone	Name

### ルート・ドメインのリストを表示してリソース・レコードを転送する

**CNAME** タイプおよび **A** タイプのゾーン・リソース・レコードを、**dig** コマンドと **axfr** 転送タイプを使用して転



送します。

コマンド

```
dig @<server> <domain> axfr | awk '/(CNAME|A)/{print $1, "\t", $4, "\t", $5}'
```

出力

```
Ns-2.od5.lohika.com. CNAME      dc05-2.od5.lohika.com
od5.lohika.com.      A          134.44.98.22
ftp.od5.lohika.com.  CNAME      od5.lohika.com.
```

マッピング

CMD 出力属性	CI 名	CI 属性
1 番目の列	DNS Alias	Name
3 番目の列	DNS Alias	Canonical name

用語集

• CNAME レコードまたは正規名レコード

ドメイン名が別の正規ドメイン名のエイリアスであることを指定するドメイン・ネーム・システム(DNS)内のリソース・レコードのタイプ。

• ゾーン転送

ゾーンに含まれるレコードのリストを表示すること。

# 第67章: ファイアウォール

## 本章の内容

Firewall Topology by SNMP ジョブ ..... 1022

はじめに ..... 1022

トポロジ・マップ ..... 1023

アダプタ情報 ..... 1023

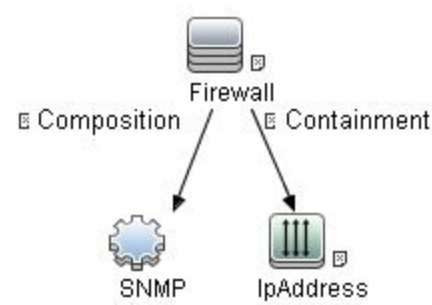
## Firewall Topology by SNMP ジョブ

本項ではこのジョブに関する詳細を説明します。

### はじめに

このジョブは、SNMP によって Juniper および Fortinet のファイアウォールを検出します。

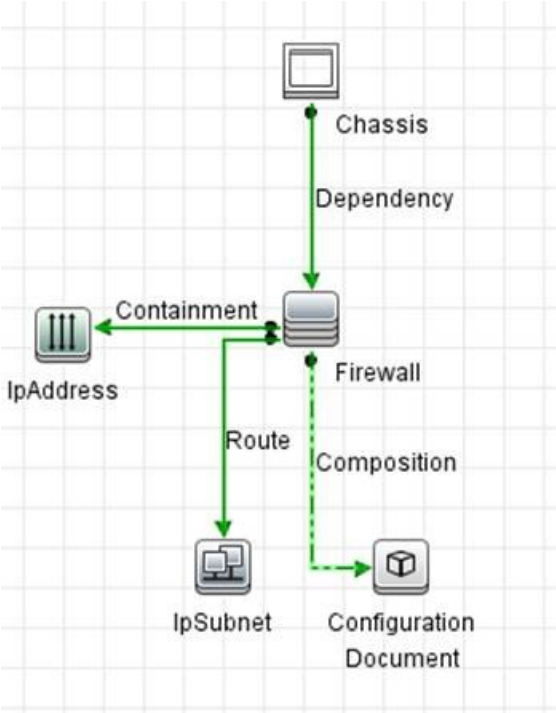
### トリガ TQL



ノード名	条件
Firewall	なし
SNMP	NOT Reference to the credentials dictionary entry null
IpAddress	NOT IP Probe Name null

## トポロジ・マップ

**Firewall Topology by SNMP** トポロジは次に示すとおりです。



## アダプタ情報

このジョブでは, **Firewall Topology by SNMP** アダプタを使用します。

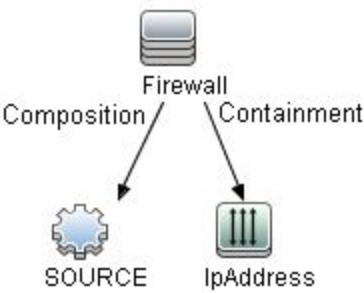
### アダプタ・タイプ

Jython

### 入力 CIT

SNMP

入力 TQL



トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
codepage	\${SOURCE.codepage:NA}
credentialsId	\${SOURCE.credentials_id}
discovered_vendor	\${Firewall.discovered_os_vendor:NA}
hostId	\${SOURCE.root_container}
ip_address	\${SOURCE.application_ip}
language	\${SOURCE.language:NA}

使用スクリプト

- firewall.py
- firewall\_discoverer.py
- firewall\_topology\_by\_snmp.py

必要な権限

権限	操作	使用状況の詳細	オブジェクトとパラメータ
SNMP	read	Juniper related	1.3.6.1.4.1.2636.3.39.1.4.1.1.2.1.1

権限	操作	使用状況の詳細	オブジェクトとパラメータ
			1.3.6.1.4.1.2636.3.39.1.4.1.1.2 .1.3 1.3.6.1.4.1.2636.3.39.1.4.1.1.2 .1.5 1.3.6.1.4.1.2636.3.39.1.4.1.1.2 .1.7 1.3.6.1.4.1.2636.3.39.1.7.1.1.2 .1.1 1.3.6.1.4.1.2636.3.39.1.7.1.1.2 .1.2 1.3.6.1.4.1.2636.3.39.1.7.1.1.2 .1.4 1.3.6.1.4.1.2636.3.39.1.7.1.1.2 .1.5 1.3.6.1.4.1.2636.3.39.1.7.1.1.2 .1.6 1.3.6.1.4.1.2636.3.5.1.1.1 1.3.6.1.4.1.2636.3.5.1.1.2 1.3.6.1.4.1.2636.3.5.1.1.3
SNMP	read	Fortinet related	1.3.6.1.4.1.12356.101.10.112. 5.1.1 1.3.6.1.4.1.12356.101.10.112. 5.1.2 1.3.6.1.4.1.12356.101.10.112. 5.1.3 1.3.6.1.4.1.12356.101.10.112. 5.1.4 1.3.6.1.4.1.12356.101.10.112. 5.1.5 1.3.6.1.4.1.12356.101.10.112. 5.1.6 1.3.6.1.4.1.12356.101.10.112. 5.1.7 1.3.6.1.4.1.12356.101.10.113. 1.1

権限	操作	使用状況の詳細	オブジェクトとパラメータ
			1.3.6.1.4.1.12356.101.10.113.1.2
			1.3.6.1.4.1.12356.101.10.113.1.3
			1.3.6.1.4.1.12356.101.10.113.1.4
			1.3.6.1.4.1.12356.101.10.113.1.5
			1.3.6.1.4.1.12356.101.10.113.1.6
			1.3.6.1.4.1.12356.101.12.2.4.1.1
			1.3.6.1.4.1.12356.101.12.2.4.1.2
			1.3.6.1.4.1.12356.101.12.2.4.1.4
			1.3.6.1.4.1.12356.101.12.2.4.1.5
			1.3.6.1.4.1.12356.101.5.1.2.1.1.1
			1.3.6.1.4.1.12356.101.5.1.2.1.1.2
			1.3.6.1.4.1.12356.101.5.1.2.1.1.3
			1.3.6.1.4.1.12356.101.8.2.1.1.1
			1.3.6.1.4.1.12356.101.8.2.1.1.10
			1.3.6.1.4.1.12356.101.8.2.1.1.2
			1.3.6.1.4.1.12356.101.8.2.1.1.3
			1.3.6.1.4.1.12356.101.8.2.1.1.4
			1.3.6.1.4.1.12356.101.8.2.1.1.

権 限	操 作	使 用 状 況 の 詳 細	オブジェクトとパラメータ
			5 1.3.6.1.4.1.12356.101.8.2.1.1. 6 1.3.6.1.4.1.12356.101.8.2.1.1. 7 1.3.6.1.4.1.12356.101.8.2.1.1. 8 1.3.6.1.4.1.12356.101.8.2.1.1. 9

必要なディスカバリ・プロトコル

SNMP Protocol

検出された CIT

- Chassis
- Composition
- ConfigurationDocument
- Dependency
- Firewall
- IpAddress
- IpServiceEndpoint
- Usage

# 第 68 章: AS400 Host ディスカバリ

## 本章の内容

概要 .....	1029
サポートされるバージョン .....	1029
トポロジ .....	1030
AS400 ホストの検出方法 .....	1031
Host Connection to AS400 ジョブ .....	1032



AS400 Host ディスカバリは、AS400 コンピュータ向けの簡単なホスト接続ディスクバリです。UCMDB Data Flow Probe は、IBM(R) jt400 ライブラリによって作成された AS/400 オブジェクトを使用して、AS400 システムにアクセスし、ホスト情報を取得します。

The diagram illustrates the data flow for AS400 objects. A central circle labeled "AS400 オブジェクト" is connected to four surrounding rectangular boxes. The top-left box is labeled "データ・キュー・オブジェクト" (Data Queue Object). The top-right box is labeled "データ・キュー・サーバ・ジョブ" (Data Queue Server Job). The bottom-left box is labeled "IFS オブジェクト" (IFS Object). The bottom-right box is labeled "ファイル・サーバ・ジョブ" (File Server Job). The left side of the diagram is labeled "UCMDB Data Flow Probe" and the right side is labeled "AS400 システム". A vertical label "JT400 ライブラリ" (JT400 Library) is positioned between the probe and the AS400 system.

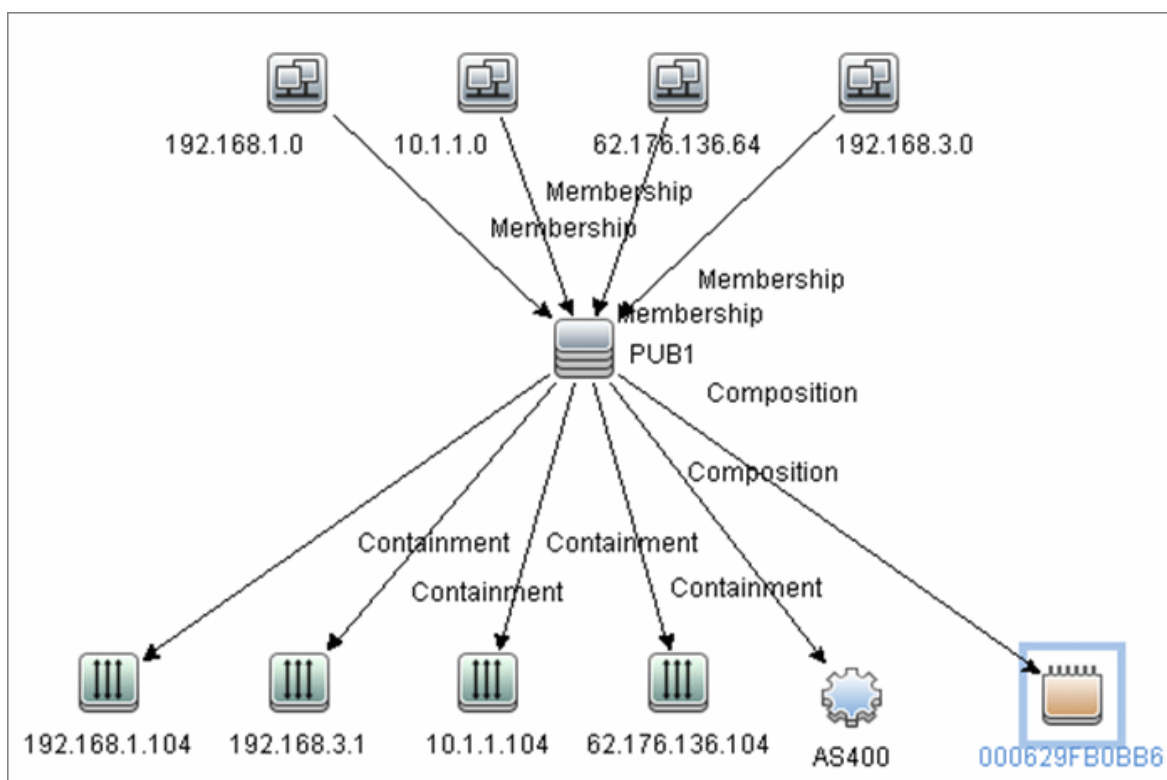
このディスカバリは AS400 の次のバージョンをサポートします。

- V4R2M0
- V3R2M1
- V3R2M0
- V4R5M0
- V5R3
- V5R4M0
- V6R1

## トポロジ

次の図は、AS400 Host ディスカバリのトポロジとサンプル出力を示しています。

注: 検出された CIT のリストについては、[「Host Connection to AS400 ジョブ」](#)(1032ページ)を参照してください。



## AS400 ホストの検出方法

このタスクは、AS400 ホストの検出方法を説明するものであり、次の手順を含みます。

### 1. 前提条件 - プロトコル資格情報の設定

このディスカバリでは、AS400 プロトコルを使用します。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - IP アドレスおよび権限

- AS400 Host ディスカバリの対象範囲に対して IP ping スweepが完了していることを確認します。
- AS400 システムでディスカバリを実行するための適切な権限をユーザが持っていることを確認します。
  - \*OBJMGT
  - \*OBJEXIST
  - \*ADD
  - \*READ
  - \*EXCLUDE
  - \*EXECUTE
  - \*CHANGE
  - \*USE
  - \*SHRNUP

### 3. ディスカバリの実行

**Host Connection to AS400** ディスカバリ・ジョブをアクティブにします。

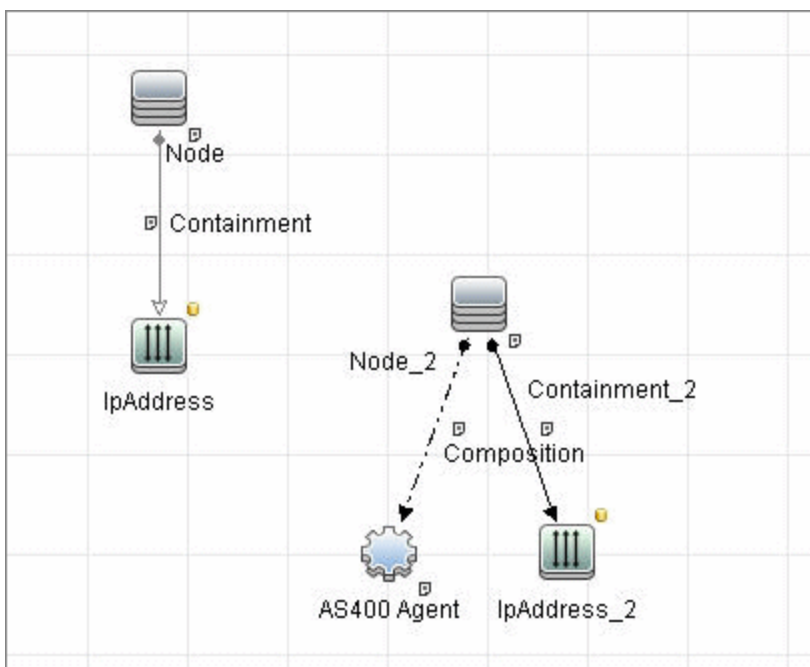
ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Host Connection to AS400 ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ

トリガ CI : IP アドレス



### 検出された CIT

- AS400Agent
- Composition
- Containment
- Interface
- IPAddress
- IpSubnet

- Membership
- Node
- Parent

注: トポロジを表示するには, [「トポロジ」\(1030ページ\)](#)を参照してください。

## 第69章: Host Connection by PowerShell ディスカバリ

### 本章の内容

概要 .....	1035
サポートされるバージョン .....	1035
PowerShell によるホスト接続の検出方法 .....	1036
Host Connection by PowerShell ジョブ .....	1037
トラブルシューティングおよび制限事項 .....	1047

## 概要

Windows PowerShell は、Microsoft 社が提供するタスク自動化フレームワークであり、.NET Framework を基本構造としてこのフレームワークに統合されたコマンドライン・シェルおよび関連スクリプト言語から構成されます。PowerShell を使用することで、COM および WMI へのフル・アクセスが可能になり、管理者は、ローカルとリモートの両方の Windows システムで管理タスクを実行できます。

通常、PowerShell では、**cmdlet**(コマンドレットと呼ぶ)を使用して管理タスクを実行します。これは、特定の操作を実行する特別な .NET クラスです。一連のコマンドレットをスクリプトや実行可能ファイル(スタンドアロン・アプリケーション)内で組み合わせることができます。また、通常の .NET クラス(または WMI/COM オブジェクト)のインスタンスを作成することで、一連のコマンドレットを組み合わせることができます。コマンドレットは、さまざまなデータ・リポジトリ(ファイル・システム、レジストリなど)のデータにアクセスすることで機能します。これにより、PowerShell は Windows PowerShell プロバイダを介して各データ・リポジトリにアクセスできます。

## サポートされるバージョン

このディスカバリは PowerShell 2.0 をサポートします。

## PowerShell によるホスト 接続の検出方法

本項では、Host Connection by PowerShell デイスカバリについて説明します。

### 1. 前提条件 - プロトコル資格情報の設定

Host Connection by PowerShell デイスカバリ・ソリューションは、PowerShell プロトコルをベースにしています。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - PowerShell の設定

デイスカバリを開始する前に、Data Flow Probe マシンに PowerShell v2.0 がインストールされ、設定されていることを確認します。インストール・ファイルにアクセスするには、<http://support.microsoft.com/kb/968929> を参照してください。

a. PowerShell のリモート処理を有効にします。

- 管理者として PowerShell v 2.0 を起動します。
- **Enable-PSRemoting** コマンドレットを実行します。これにより、WinRM サービスが開始され、スタートアップの種類が自動的に設定されます。また、WS-Management 通信のファイアウォール例外が有効になり、任意の IP アドレスで要求を受け付けるためのリスナーが作成されます。

注: ドメイン内のすべてのコンピュータで PowerShell のリモート処理を有効にするには、[ドメイングループポリシー]で、[コンピューターの構成]>[ポリシー]>[管理用テンプレート]>[Windows コンポーネント]>[Windows リモート管理 (WinRM)]>[WinRM サービス]の順に移動し、[リスナーの自動構成を許可する]を選択します。

b. すべてのホストを信頼するには、コマンドラインから次のコマンドを実行します。

```
Set-Item WSMan:\localhost\Client\TrustedHosts *
```

制限された IP アドレスのみを信頼するには、アスタリスク(\*)の代わりにそのアドレスを指定します。

c. コマンドラインから次のコマンドを実行して、WinRM を再起動します。



```
restart-Service winrm
```

注: 標準設定では, WinRM は Kerberos を使用して認証を行います。https に対応するように WinRM を設定するには, <http://support.microsoft.com/kb/2019527> を参照してください。

### 3. デイスカバリの実行

- a. **Range IPs by ICMP** ジョブを実行します。
- b. **Host Connection by PowerShell** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Host Connection by PowerShell ジョブ

本項ではこのジョブに関する詳細を説明します。

### コマンド

本項では, Host Connection by PowerShell デイスカバリで使用する各コマンドについて説明します。

### コマンド

```
Get-WmiObject -Query "SELECT BuildNumber, Caption, Version, csdversion, lastBootUpTime, otherTypeDescription FROM Win32_OperatingSystem " | Format-List BuildNumber, Caption, Version, csdversion, lastBootUpTime, otherTypeDescription
```

#### • 出力

```
BuildNumber : 2600
Caption : Microsoft Windows XP Professional
Version : 5.1.2600
csdversion : Service Pack 3
lastBootUpTime : 20101108094626.357090+120
otherTypeDescription :
```

#### • マッピング

このコマンドの出力をもとに CI の属性が入力されます。

コマンド出力属性	CI タイプ	CI 属性
BuildNumber	Windows	Host Operating System Release
Caption(1)	Windows	Host Operating System
Version	Windows	Host Operating System Version
csdversion	Windows	Windows Service Pack
lastBootUpTime	Windows	Host Boot Time
Caption(2)	Windows	Host Operating System Installation Type

## コマンド

```
Get-WmiObject -Query "SELECT Domain, Manufacturer, Model, Name FROM Win32_
ComputerSystem " | Format-List Domain, Manufacturer, Model, Name
```

### • 出力

```
Domain : od5.lohika.com
Manufacturer : INTEL_
Model : D946GZIS
Name : DDM-RND-SV
```

### • マッピング

このコマンドの出力をもとに CI の属性が入力されます。

コマンド出力属性	CI タイプ	CI 属性
Domain	Windows	OS domain name
Manufacturer	Windows	PC manufacturer
Model	Windows	Host model
Name	Windows	Host name

## コマンド

```
Get-WmiObject -Query "SELECT name, uuid FROM win32_ComputerSystemProduct " |
Format-List name, uuid
```

- 出力

```
name :  
uuid : EAB9B406-CE4F-DB11-9150-0013D4D0773D
```

- マッピング

このコマンドの出力をもとに CI の属性が入力されます。

コマンド出力属性	CI タイプ	CI 属性
Uuid	Windows	Host BIOS UUID
Name	Windows	Host model

## コマンド

```
Get-WmiObject -Query "SELECT serialNumber FROM Win32_BIOS " | Format-List  
serialNumber
```

- 出力

```
serialNumber : BQJ0749007TY
```

- マッピング

このコマンドの出力をもとに CI の属性が入力されます。

コマンド出力属性	CI タイプ	CI 属性
serialNumber	Windows	Host serial number

## コマンド

```
Get-WmiObject -Query "SELECT serialNumber FROM Win32_SystemEnclosure " | Format-  
List serialNumber
```

- 出力

```
serialNumber : BQJ0749007TY
```

- マッピング

このコマンドの出力をもとに CI の属性が入力されます。

コマンド出力属性	CI タイプ	CI 属性
serialNumber	Windows	Host serial number

## コマンド

```
Get-WmiObject -Query "SELECT metric1, nextHop FROM Win32_IP4RouteTable WHERE destination = '0.0.0.0' and mask = '0.0.0.0'" | Format-List metric1, nextHop
```

### • 出力

```
metric1 : 20
nextHop : 134.44.98.7
```

### • マッピング

このコマンドの出力をもとに CI の属性が入力されます。

コマンド出力属性	CI タイプ	CI 属性
メトリック値が最小の nextHop	Windows	Default gateway

## コマンド

```
Get-WmiObject -Query "SELECT dnsServerSearchOrder FROM Win32_NetworkAdapterConfiguration WHERE domainDnsRegistrationEnabled <> NULL" | Format-List dnsServerSearchOrder
```

### • 出力

```
dnsServerSearchOrder : {16.110.135.51, 16.110.135.52}
dnsServerSearchOrder : {134.44.98.21, 134.44.98.22}
```

### • マッピング

このコマンドの出力をもとに CI の属性が入力されます。IP アドレスに基づいて、DNS サーバ・アプリケーション CI が関連付けられた不完全なホストが作成されます。

## コマンド

```
Get-WmiObject -Query "SELECT WinsPrimaryServer, WinsSecondaryServer FROM Win32_NetworkAdapterConfiguration WHERE WinsPrimaryServer <> NULL or WinsSecondaryServer <> NULL" | Format-List WinsPrimaryServer, WinsSecondaryServer
```

## • 出力

```
WinsPrimaryServer : 16.232.7.246
WinsSecondaryServer : 16.236.105.246
```

## • マッピング

このコマンドの出力をもとに CI の属性が入力されます。IP アドレスに基づいて、WINS サーバ・アプリケーション CI が関連付けられた不完全なホストが作成されます。

## コマンド

```
Get-WmiObject -Query "SELECT dhcpServer FROM Win32_NetworkAdapterConfiguration
WHERE dhcpServer <> NULL" | Format-List dhcpServer
```

## • 出力

```
dhcpServer : 134.44.98.22
```

## • マッピング

このコマンドの出力をもとに CI の属性が入力されます。IP アドレスに基づいて、DHCP サーバ・アプリケーション CI が関連付けられた不完全なホストが作成されます。

## コマンド

```
Get-WmiObject -Query "SELECT Caption, Description, DhcpEnabled, IPAddress,
IPSubnet, MACAddress FROM Win32_NetworkAdapterConfiguration WHERE MACAddress <>
NULL" | Format-List Caption, Description, DhcpEnabled, IPAddress, IPSubnet,
MACAddress
```

## • 出力

```
Caption : [00000003] WAN Miniport (PPTP)
Description :WAN Miniport (PPTP)
DhcpEnabled :False
IPAddress :
IPSubnet :
MACAddress : 50:50:54:50:30:30
Caption :[00000004] WAN Miniport (PPPOE)
Description :WAN Miniport (PPPOE)
DhcpEnabled :False
IPAddress :
IPSubnet :
MACAddress : 33:50:6F:45:30:30
Caption :[00393219] WAN Miniport (IP)
Description :WAN (PPP/SLIP) Interface
```

```

DhcpEnabled :False
IPAddress :{16.213.65.117}
IPSubnet :{255.255.255.255}
MACAddress :0:53:45:0:00:00
Caption :[00000007] Packet Scheduler Miniport
Description :Packet Scheduler Miniport
DhcpEnabled :False
IPAddress :
IPSubnet :
MACAddress : 4A:6F:20:52:41:53
Caption :[00000008] Intel(R) PRO/100 VE Network Connection
Description :Intel(R) PRO/100 VE Network Connection - Teefer2 Miniport
DhcpEnabled :True
IPAddress :{134.44.99.108}
IPSubnet :{255.255.252.0}
MACAddress :00:16:76:BE:7E:DD
Caption :[00000009] Packet Scheduler Miniport
Description :Packet Scheduler Miniport
DhcpEnabled :False
IPAddress :
IPSubnet :
MACAddress : 00:16:76:BE:7E:DD
Caption :[00000013] Teefer2 Miniport
Description :Teefer2 Miniport
DhcpEnabled :False
IPAddress :
IPSubnet :
MACAddress : 00:16:76:BE:7E:DD
Caption :[00000014] Teefer2 Miniport
Description :Teefer2 Miniport
DhcpEnabled :False
IPAddress :
IPSubnet :
MACAddress : 4A:6F:20:52:41:53
    
```

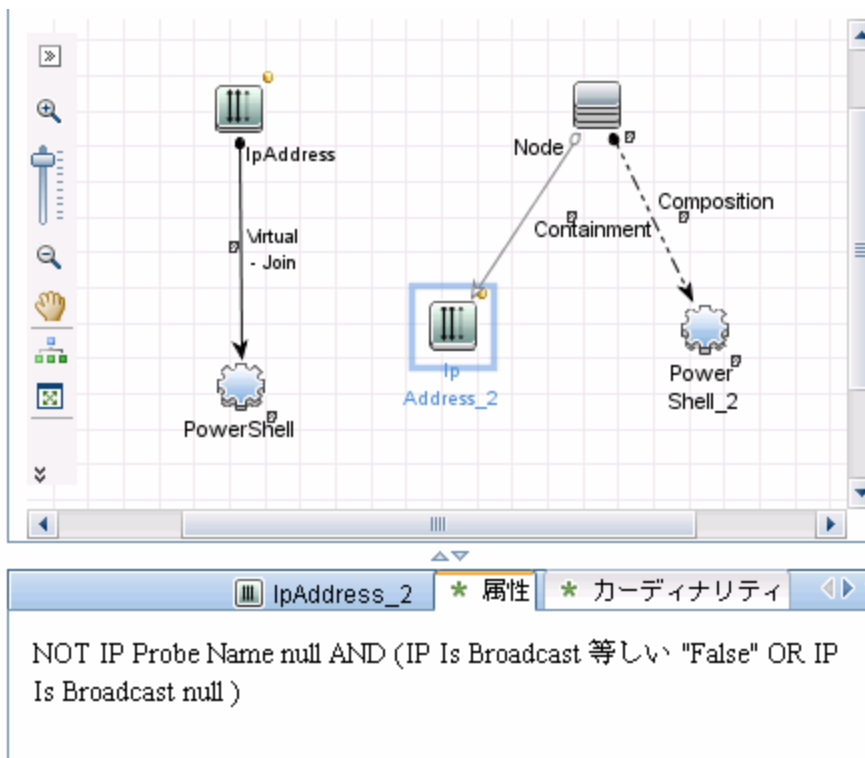
## • マッピング

このコマンドの出力をもとに CI の属性が入力されます。

コマンド出力属性	CI タイプ	CI 属性
Description	Interface	Interface description
DhcpEnabled	Interface	DHCP Enabled

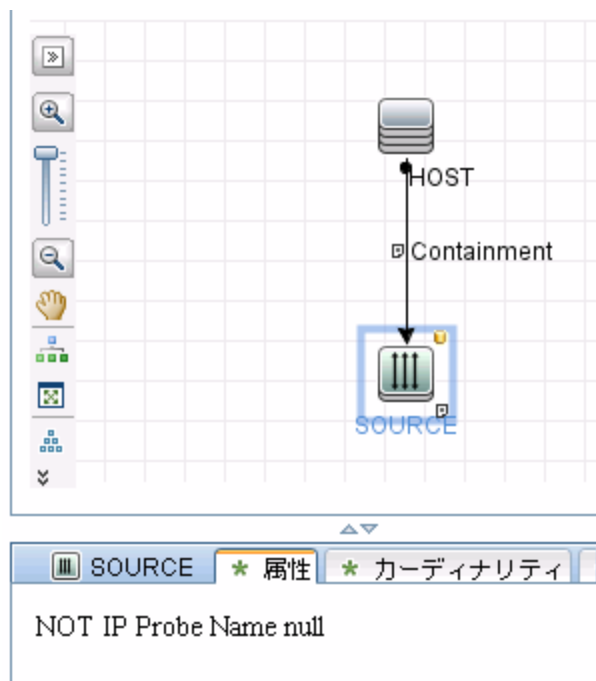
コマンド出力属性	CI タイプ	CI 属性
IPAddress	IpAddress	IP address
IPSubnet	IpAddress	IP Network Address
MACAddress	Interface	Interface MAC Address

## トリガ・クエリ



## アダプタ

- 入力クエリ:



• 使用スクリプト :

- networking.py
- Host\_connection\_by\_powershell.py
- Host\_discoverer.py
- Host\_win.py
- Host\_win\_shell.py
- Host\_win\_wmi.py
- Networking\_win.py
- Networking\_win\_shell.py
- Networking\_win\_wmi.py

• トリガ CI データ :



名前	値
host_cmdbid	\${HOST.root_id:NA}
host_key	\${HOST.host_key:NA}
ip_address	\${SOURCE.ip_address}
ip_domain	\${SOURCE.ip_domain}
mac_addr	\${NA}

検出された CIT

- Composition
- Containment
- DnsServer
- Interface
- IpAddress
- IpSubnet
- Membership
- Node
- Parent
- PowerShell
- RunningSoftware
- Terminal Server
- Windows

## 作成または変更されたエンティティ

エンティティ名	エンティティ・タイプ	エンティティの説明
powershell.xml	CIT	PowerShell プロトコルを表す
Host Connection by Powershell.xml	ジョブ	メイン・ジョブ
Powershell_host_connection.xml	アダプタ	ジョブ・アダプタ
Host_connection_by_powershell.py	スクリプト	デイスカバリ・スクリプト
Host_win.py	スクリプト	デイスカバリ・スクリプト
Host_win_shell.py	スクリプト	デイスカバリ・スクリプト
Networking_win.py	スクリプト	デイスカバリ・スクリプト
Networking_win_shell.py	スクリプト	デイスカバリ・スクリプト
Networking_win_wmi.py	スクリプト	デイスカバリ・スクリプト
Host_win_wmi.py	スクリプト	デイスカバリ・スクリプト

## トラブルシューティングおよび制限事項

本項では、Host Connection by PowerShell Discovery ディスカバリのトラブルシューティングと制限事項について説明します。

**問題：**アクセス拒否のエラー・メッセージが表示されます。

PowerShell プロトコルで Windows 2008 SP2 の宛先を検出中に、次のエラー・メッセージが表示される場合があります。

Connecting to remote server failed with the following error message: Access is denied. For more information, see the about\_Remote\_Troubleshooting Help topic.

このメッセージは、宛先ホストの検出を試行しているユーザがローカルの Administrator ユーザでない場合に 표시됩니다(ユーザが Administrator グループのメンバであるかどうかは関係ありません)。

**解決策：**PowerShell の追加の設定が必要です。

**LocalAccountTokenPolicy** キーを変更し、Administrator グループのユーザが管理者権限を使用してリモート接続できるようにする必要があります。検出するホストの PowerShell で、次のコマンドを実行します。

```
Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name LocalAccountTokenFilterPolicy -Value 1 -Type DWord
```

この特殊なケースの詳細については、「他のドメインの管理者のリモート処理機能を有効にする方法」(<http://technet.microsoft.com/ja-jp/library/dd347642.aspx>)を参照してください。

## 第70章: Layer2 ディスカバリ

### 本章の内容

概要 .....	1049
サポートされるデバイス .....	1050
Layer2 オブジェクトの検出方法 .....	1051
保存されたファイルを使用して Layer2 接続を検出する方法 .....	1054
CDP または LLDP MIB を使用して Layer2 接続を検出する方法 .....	1056
シェルによる Layer2 トポロジの検出方法 .....	1056
Layer2 Topology Bridge-based by SNMP ジョブ .....	1057
Layer2 Topology by Shell ジョブ .....	1058
Layer2 Topology CDP-LLDP based by SNMP ジョブ .....	1060
Layer2 Topology VLAN-based by SNMP ジョブ .....	1060
Process Layer2 Saved Files ジョブ .....	1061
Merge VLANs by Ports ジョブ .....	1063
VLAN ports by SNMP ジョブ .....	1065
VLANS by SNMP ジョブ .....	1066
L2 Bridge by SNMP アダプタ .....	1068
Layer2 Topology by Shell アダプタ .....	1069
CDP/LLDP Neighbors Layer 2 Devices by SNMP アダプタ .....	1072
L2 Vlan by SNMP アダプタ .....	1073
Merge VLANs アダプタ .....	1075
Process Layer2 Collected Files アダプタ .....	1078
VMS Catalyst by SNMP アダプタ .....	1079
Catalyst Vlans by SNMP アダプタ .....	1080
関係 .....	1082
トラブルシューティングおよび制限事項 .....	1083

## 概要

Layer2 パッケージでは、スイッチ・ツリー・トポロジを含む Layer2 トポロジ(スイッチ間のバックボーン・リンク)を検出するほか、スイッチ・ポートへのエンド・ユーザ接続(スイッチとホスト間の Layer2 CI)を検出します。

このパッケージでは、次のリソースから情報を検出できます。

- **フォワーディング・データベース(FDB)**

デバイス・タイプに基づいて、FDB は BRIDGE-MIB, QBRIDGE MIB, または STATISTICS-MIB で検出されます。

- **Cisco Discovery Protocol MIB(CDP-MIB)**

この MIB には、CDP により収集された、直接接続された Cisco ネットワーク・デバイスに関する情報が含まれます。

- **Link Layer Discovery Protocol MIB(LLDP-MIB)**

この MIB には、LLDP により収集された、直接接続されたネットワーク・デバイスに関する情報が含まれます。

### **Layer2 Topology Bridge-based by SNMP ジョブおよび Layer2 Topology VLAN-based by SNMP**

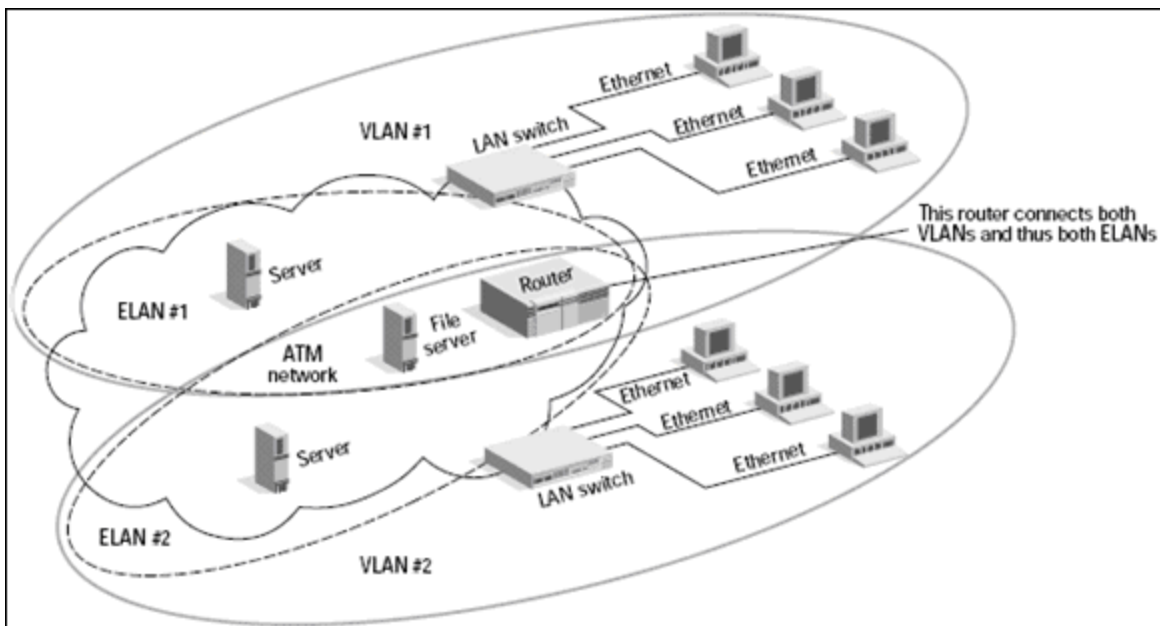
ジョブは、フォワーディング・データベースから Layer2 接続情報を選択し、スイッチとエンドポイント・ホスト間に Layer2 CI を作成して、スイッチ間の Layer2 接続に関する情報をプローブのファイル・システム上のファイルに格納します。

**Process Layer2 Saved Files** ジョブは、プローブのファイル・システム(Layer2 Topology Bridge-based by SNMP ジョブおよび Layer2 Topology VLAN-based by SNMP ジョブで作成)からデータを選択し、スイッチ間の Layer2 接続を作成します。

**Layer2 Topology CDP-LLDP based by SNMP** ジョブは、CDP または LLDP MIB から Layer2 接続情報を選択し、検出されたデバイスから、直接接続されたネットワーク・デバイスへの Layer2 接続をレポートします。

この Layer2 パッケージは、SNMP プロトコルをベースにしています。

次の図は、重複するVLAN/ELANを接続するルータを表しています。



## サポートされるデバイス

このディスカバリでは、次のデバイスがサポートされています。

- 3Com
- Cisco
- H3C
- HP ProCurve

## Layer2 オブジェクトの検出方法

このタスクでは、Layer2 オブジェクトの検出方法について説明します。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

Layer2 オブジェクトを検出するには、SNMP プロトコルが必要です。SNMP プロトコル資格情報を定義するときは、ポートおよびコミュニティ認証パラメータを使用可能にします。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - その他

- 環境内の検出対象のすべてのスイッチに SNMP アクセスが可能であることを確認します。これは、Layer2 トポロジを詳細に検出するための重要な要件です。

### 3. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次の順序でジョブをアクティブ化します。

- Host Connection by SNMP** ジョブをアクティブ化します。このジョブは、SNMP CI を CMDB に保存します。

**注:** Layer2 ディスカバリは、次の理由で接続ジョブに基づきます。

- スイッチ・ポートとホスト間の Layer2 接続は、ホストの MAC アドレスに基づきます。これらの MAC アドレスは、ネットワーク接続ジョブによって検出されます(ホスト・インタフェース)。
- Layer2 ジョブのトリガは、検出されたスイッチのタイプによって決定されます。スイッチのクラスとタイプは、Layer2 モジュールの Host Networking by SNMP ジョブによって検出されます。

- Host Networking by SNMP** ジョブをアクティブ化します。このジョブは、SNMP ルートおよびシステム・テーブルを使用してホスト・ネットワーキング・トポロジを検出します。このジョブは、環境で

検出されたスイッチのすべての SNMP エージェントで実行する必要があります。検出される Layer2 リンク名は、このディスカバリによって決定されます(Layer2 CI の名前は、検出対象の宛先ネットワーク・インタフェース・アダプタの関連するインタフェースの名前および説明と同じです)。

c. **VLANS by SNMP** ジョブをアクティブ化します。

このジョブのトリガは、**snmp\_of\_catalyst\_switch** クエリです。Switch CIT は、次のいずれかです。

- SNMP オブジェクト
- スwitchに接続されている SNMP エージェント

SNMP\_Net\_Dis\_Catalyst\_Vlans.py スクリプトは、ELAN テーブルごとに VLAN、ELAN の名前、VLAN 番号を取得します。

d. **VLAN ports by SNMP** ジョブをアクティブ化します。

このジョブのトリガは、**catalyst\_vlan** クエリです。これは、次のスイッチへの接続がある VLAN オブジェクトです。

- SNMP オブジェクトを持つスイッチ
- スwitch

トリガは、SNMP 自体ではなく、VLAN オブジェクトに配置されます。これは、VLAN オブジェクトを認証するときに、特別なコミュニティ文字列を使用する必要があるためです(検出されたスイッチの SNMP オブジェクトで検出された通常のコミュニティ文字列は使用できません)。このコミュニティ文字列は、<COMMUNITY>@<VLAN NUMBER> という値を持ちます。たとえば、コミュニティ文字列が **public**、検出された VLAN 番号が **16** の場合、コミュニティ文字列は **public@16** です。

SNMP\_Net\_Dis\_VMS\_catalyst.py スクリプトは、基本 MAC テーブルと Port number If index テーブルを取得します。

SNMP プロトコル・パラメータの詳細については、『HP Universal CMDB Data Flow Management Guide』の「SNMP Protocol」を参照してください。

e. **Layer2 Topology Bridge-based by SNMP** ジョブをアクティブ化します。

このジョブのトリガは、**catalyst\_bridge\_no\_vlan** クエリです。これは、次のスイッチへの接続があるブリッジ・オブジェクトです。



- SNMP オブジェクトを持つスイッチ
- スイッチ

このジョブ(**Layer2 Topology Bridge-based by SNMP**)と次のジョブ(**Layer2 Topology VLAN-based by SNMP**)の両方が `bridgePortDisc.py` スクリプトを使用します。このスクリプトにおけるこれらのジョブの違いは次のとおりです。

Cisco ネットワーク・デバイスの場合：

- **Layer2 Topology Bridge-based by SNMP** は、通常の SNMP コミュニティ認証を使用します。このジョブは、検出されたスイッチに VLAN が無いときにのみ、ブリッジでトリガされます。
- **Layer2 Topology VLAN-based by SNMP** は、スイッチ上で検出されたそれぞれの VLAN でトリガされます。このジョブは、関連する特別なコミュニティ認証を使用します。詳細については、[「VLAN ports by SNMP ジョブをアクティブ化します。」\(1052ページ\)](#)を参照してください。

他のネットワーク・デバイスの場合：

- **Layer2 Topology Bridge-based by SNMP** は、Layer2 接続を検出するために BRIDGE-MIB を使用します。
- **Layer2 Topology VLAN-based by SNMP** は、Layer2 接続を検出するために QBRIDGE-MIB または STATISTICS-MIB を使用します。

#### 注：

- Layer2 Topology Bridge-based by SNMP ジョブは、標準設定の VLAN ごとに Layer2 接続を検出します(標準設定の VLAN は #1 です)。Layer2 Topology VLAN-based by SNMP ジョブは、標準設定の VLAN を含むすべての VLAN について、すべての Layer2 接続を検出します。したがって、次のようになります。
- 標準設定の VLAN についてのみ Layer2 接続を検出する場合は、VLANs by SNMP ジョブと VLAN ports by SNMP ジョブを実行する必要はありません。
- VLANs by SNMP ジョブを実行して、ネットワーク・デバイスに接続されている VLAN CI がレポートされる場合、Layer2 Topology Bridge-based by SNMP ジョブは、Layer2 接続を検出しません。これは、デバイスがジョブのトリガ TQL クエリから除外されるためです。検出された VLAN を使用するデバイスで Layer2 接続を検出する

には、Layer2 Topology VLAN-based by SNMP ジョブを使用する必要があります。

- VLAN を保持するスイッチのブリッジで Bridge-based Layer2 ジョブをディスパッチすると、標準設定の VLAN Layer2 トポロジのみが検出されます。
- Layer2 Topology Bridge-based/VLAN-based by SNMP ジョブでは、Layer2 接続をホストするスイッチのみがレポートされます。スイッチ間のレポートを取得するには、Process Layer2 Saved Files ジョブも実行する必要があります。

f. **Layer2 Topology VLAN-based by SNMP** ジョブをアクティブ化します。

このジョブのトリガは、**catalyst\_vlan\_with\_bridge** クエリです。これは、**bridge\_mac** 属性に値が設定された VLAN オブジェクトです。これは、次のいずれかにも接続している必要があります。

- SNMP オブジェクトを持つスイッチ
- スイッチ

**bridgePortDisc.py** スクリプトの詳細については、[「Layer2 Topology Bridge-based by SNMP ジョブをアクティブ化します。」\(1052 ページ\)](#)を参照してください。

## 保存されたファイルを使用して Layer2 接続を検出する方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

Layer2 オブジェクトを検出するには、SNMP プロトコルが必要です。SNMP プロトコル資格情報を定義するときは、ポートおよびコミュニティ認証パラメータを使用可能にします。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次の順序でジョブを実行します。

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by SNMP** ジョブを実行して、ターゲット・ホストおよびターゲット・ホストへの接続を検出します。
- c. 使用する環境に応じて、**Layer2 Topology Bridge-based by SNMP** ジョブと **Layer2 Topology VLAN-based by SNMP** ジョブのいずれか、またはその両方を実行します。
- d. 上記のジョブをすべて実行したら、**Process Layer2 Saved Files** ジョブを実行します。

## CDP または LLDP MIB を使用して Layer2 接続を検出する方法

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

Layer2 オブジェクトを検出するには、SNMP プロトコルが必要です。SNMP プロトコル資格情報を定義するときは、ポートおよびコミュニティ認証パラメータを使用可能にします。

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. ディスカバリの実行

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

次の順序でジョブを実行します。

- a. **Range IPs by ICMP** ジョブを実行してターゲット IP を検出します。
- b. **Host Connection by SNMP** ジョブを実行して、ターゲット・ホストおよびターゲット・ホストへの接続を検出します。
- c. **Layer2 Topology CDP-LLDP based by SNMP** ジョブを実行します。

## シェルによる Layer2 トポロジの検出方法

シェルによる Layer2 トポロジの検出を行うには、次のジョブを実行します。

1. Range IP by ICMP
2. Host Connection by Shell
3. Layer2 Topology by Shell

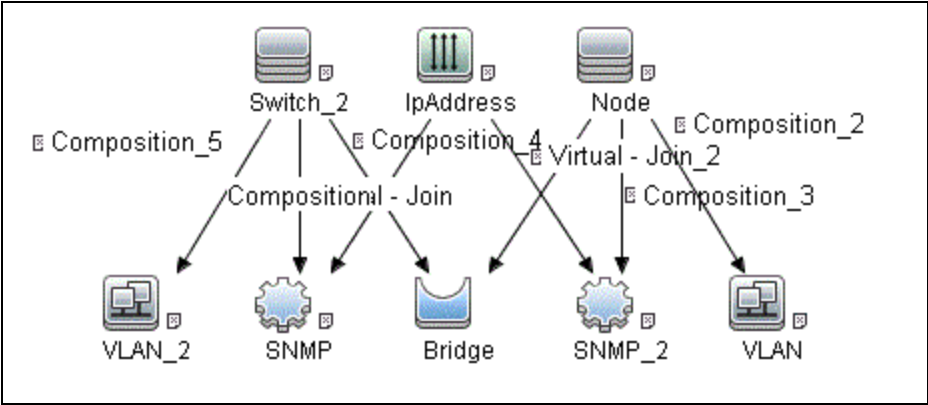
## Layer2 Topology Bridge-based by SNMP ジョブ

本 項 ではこのジョブに関する詳 細を説 明します。

### アダプタ

このジョブでは, L2 Bridge by SNMP アダプタを使用します。

### トリガ TQL クエリ



ノード名	条件
Switch_2	CI Type 等しい switch OR NodeRole 含む switch
IpAddress	NOT IP Probe Name null
Node	CI Type 等しい switch OR NodeRole 含む switch
SNMP	NOT Reference to the credentials dictionary entry null
SNMP_2	NOT Reference to the credentials dictionary entry null

### ディスカバリ・フロー

このディスカバリでは次の操作を実行します。

- 対象マシンへの接続。
- VLAN 関連データの検出。

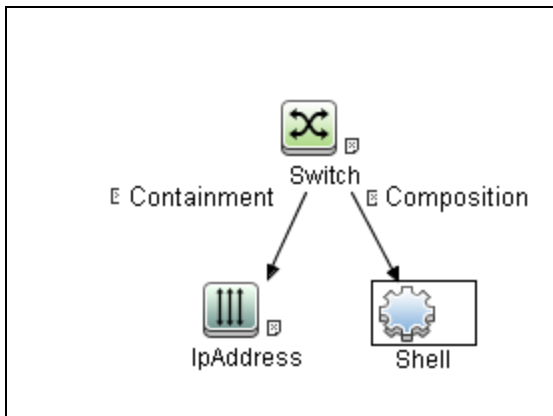
## Layer2 Topology by Shell ジョブ

このジョブは, Cisco Nexus 1000, 5000, 7000 シリーズのスイッチにおける Layer2 関連 データをレポートします。

### アダプタ

ID : Layer2 Topology by Shell

### トリガ TQL



### パラメータ

なし

### 前提条件

- SSH プロトコル資格情報を設定します。これに関する詳細については、『HP UCMDB Discovery and Integrations Content Guide - Supported Content』の SSH プロトコル資格情報に関する項を参照してください。

### ディスカバリ・フロー

Layer2 Topology by Shell ジョブのディスカバリ・フローを次に示します。

1. コマンド **sh int** を使用して, インタフェースの詳細を取得します。
2. コマンド **sh vlan all-ports** を使用して, 構成されている VLAN とポートを取得します。

3. コマンド **sh cdp neighbours detail** を使用して、接続されているデバイスのリストを取得します。

## Layer2 Topology CDP-LLDP based by SNMP ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、CDP/LLDP neighbors layer 2 devices by SNMP アダプタを使用します。

### トリガ TQL クエリ

ノード名	条件
Node	CI Type 等しい switch OR CI Type 等しい switchrouter OR NodeRole 含む switch OR NodeRole 含む router
SNMP	NOT Reference to the credentials dictionary entry null

## Layer2 Topology VLAN-based by SNMP ジョブ

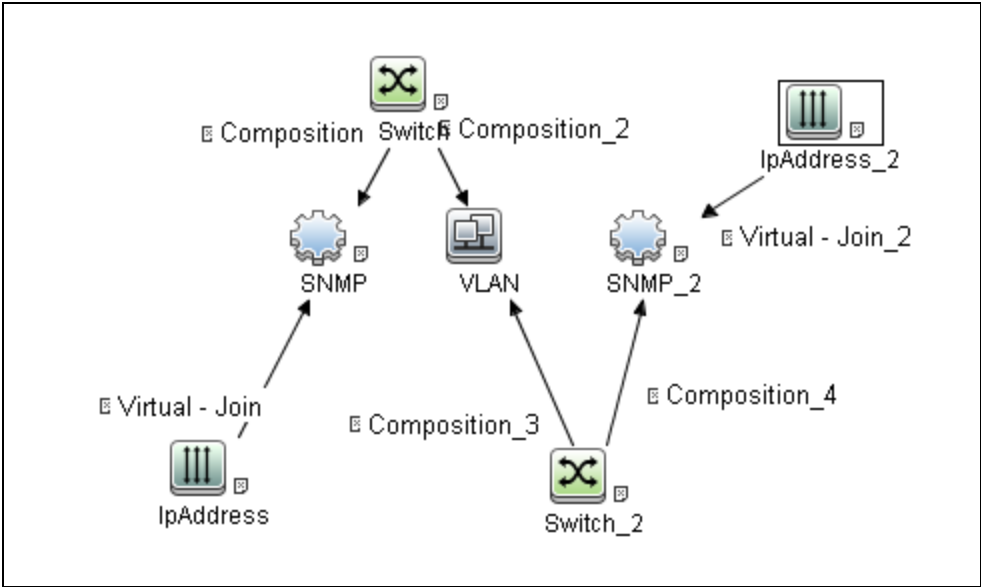
本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、L2 Vlan by SNMP アダプタを使用します。



トリガ TQL クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
IpAddress_2	NOT IP Probe Name null
SNMP	NOT Reference to the credentials dictionary entry null
SNMP_2	NOT Reference to the credentials dictionary entry null
VLAN	NOT VLAN Bridge MAC null

ディスカバリ・フロー

このディスカバリでは次の操作を実行します。

- 対象マシンへの接続。
- VLAN 関連データの検出。

Process Layer2 Saved Files ジョブ

本項ではこのジョブに関する詳細を説明します。

## ディスカバリ・フロー

このジョブは、1 つのスイッチ・インタフェースの MAC と別のスイッチ・インタフェースの MAC 間のすべての接続を検出します。この接続が存在する場合、ディスカバリは Layer2 接続を作成します。

## アダプタ

このジョブでは、Process Layer2 Collected Files アダプタを使用します。

## トリガ TQL クエリ



## Merge VLANs by Ports ジョブ

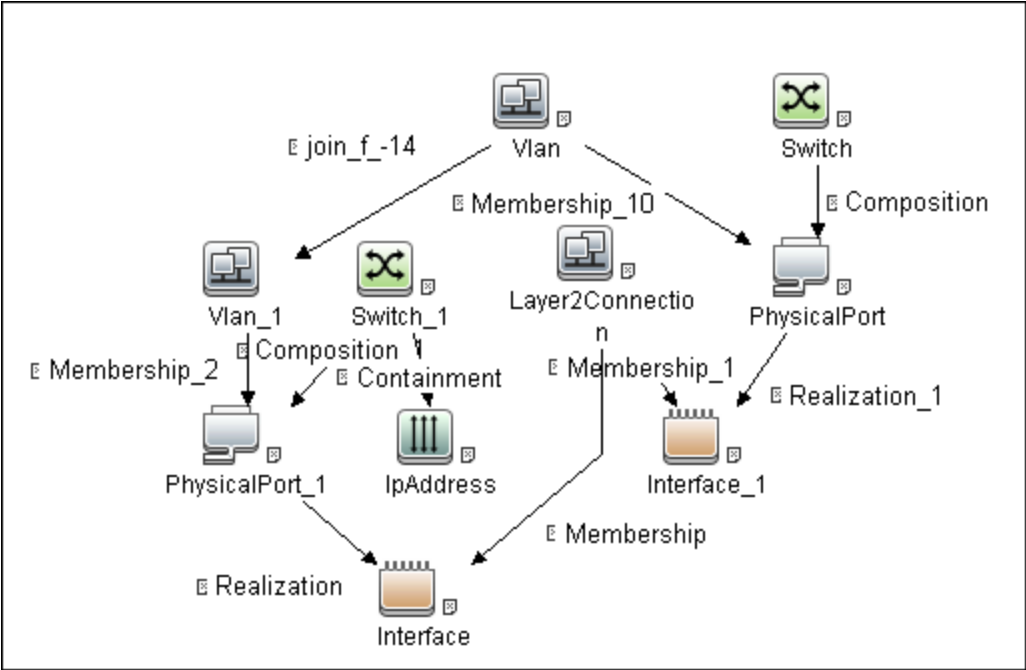
このジョブの機能は、エンリッチメントまたは調整の機能と似ています。これは、UCMDB 内にすでに存在するデータでのみ機能します。また、次のようなトポロジの場合、VLAN がマージされます。

- 1. VLAN に関連するポートが Layer2 接続によって接続されており、かつ
- 2. VLAN ID が同じである。

### アダプタ

このジョブでは、Merge VLANs アダプタを使用します。

### トリガ TQL クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null

### ディスカバリ・フロー

このディスカバリでは次の操作を実行します。

- 対象マシンへの接続。
- VLAN 関連データの検出。

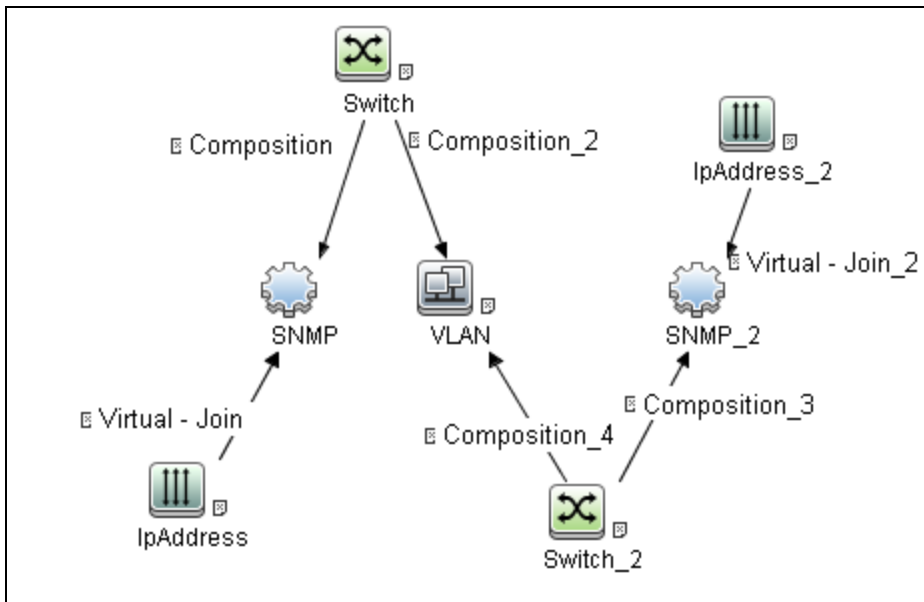
## VLAN ports by SNMP ジョブ

本項ではこのジョブに関する詳細を説明します。このジョブの使用は推奨されないため、[ディスカバリ モジュール] > [Tools and Samples] > [Deprecated Jobs]に移動されています。

### アダプタ

このジョブでは、VMS Catalyst by SNMP アダプタを使用します。

### トリガ TQL クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null
IpAddress_2	NOT IP Probe Name null
SNMP	NOT Reference to the credentials dictionary entry null
SNMP_2	(SNMP Description 類似 %atalyst% OR SNMP Description 類似 (大文字小文字の区別なし) %cisco%) AND NOT Reference to the credentials dictionary entry null
Switch	DiscoveredOsName 類似 %atalyst% OR DiscoveredModel 類似 %atalyst% OR DiscoveredOsName 類似 (大文字小文字の区別なし) %cisco% OR DiscoveredModel 類似 (大文字小文字の区別なし)

ノード名	条件
	%cisco%

## ディスカバリ・フロー

このディスカバリでは次の操作を実行します。

- 対象マシンへの接続。
- VLAN 関連データの検出。

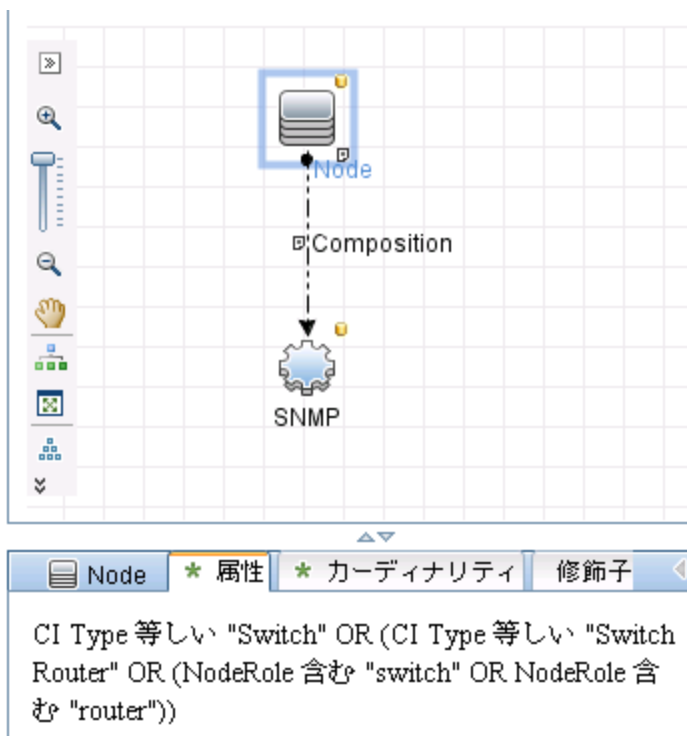
## VLANS by SNMP ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、Catalist VLANS by SNMP アダプタを使用します。

### トリガ TQL クエリ



## ディスカバリ・フロー

このディスカバリでは次の操作を実行します。

- 対象マシンへの接続。
- VLAN 関連データの検出。

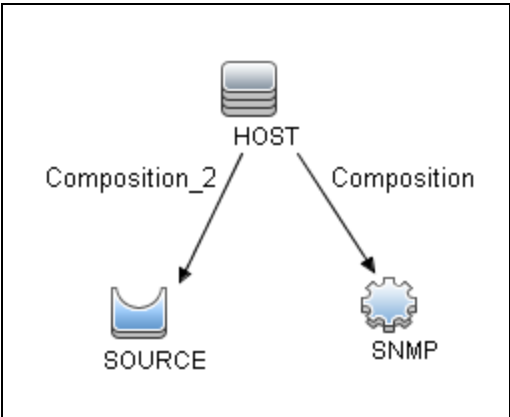
## L2 Bridge by SNMP アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Bridge

### 入力 TQL クエリ



### トリガ CI データ

名前	値
bridgeld	\${SOURCE.bridge_basemacaddr}
credentialsId	\${SNMP.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SNMP.application_ip}

### 使用スクリプト

- bridgePortDisc.py
- networking\_win.py
- SNMP\_Networking\_Utils.py



## 検出された CIT

- Bridge
- Composition
- Interface
- Layer2Connection
- Membership
- Node
- PhysicalPort
- Realization

## Layer2 Topology by Shell アダプタ

本項ではこのアダプタに関する詳細を説明します。

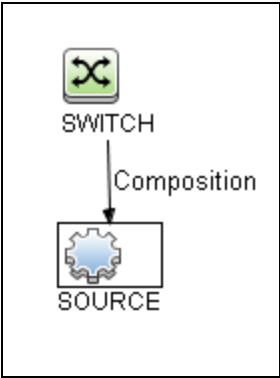
### ID

Layer2 Topology by Shell

### 入力 CIT

Shell

入力 TQL



トリガ CI データ

名前	値
credentialsId	SOURCE.credentials_id
hostId	SWITCH.root_id
ip_address	SOURCE.appliaction_ip
ip_domain	\${NA}
Protocol	SOURCE:root_class

使用スクリプト

- layer2.py
- layer2\_shell\_discoverer.py
- switch\_layer2\_by\_shell.py
- TTY\_Connection\_Utils.py

検出された CIT

- Composition
- Interface

- Layer2Connection
- Membership (layer2\_connection,interface)
- Node
- Realization
- Switch
- PhysicalPort

## パラメータ

なし

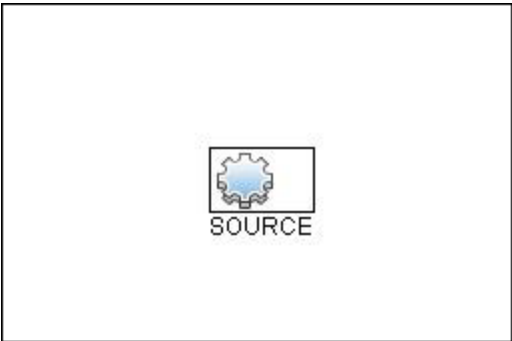
# CDP/LLDP Neighbors Layer 2 Devices by SNMP アダプタ

本項ではこのアダプタに関する詳細を説明します。

## 入力 CIT

SNMP

## 入力 TQL クエリ



## トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
ip_address	\${SOURCE.application_ip}

## 使用スクリプト

- networking\_win.py
- SNMP\_Networking\_Utils.py
- SNMP\_CDP\_LLDP.py

## 検出された CIT

- Composition
- Containment
- Interface
- IpAddress
- Layer2Connection
- Membership
- Node

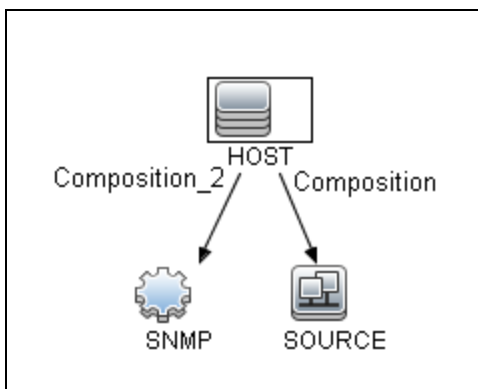
## L2 Vlan by SNMP アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Vlan

### 入力 TQL クエリ



## トリガ CI データ

名前	値
bridgeld	\${SOURCE.vlan_bridgemac}
credentialsId	\${SNMP.credentials_id}
hostId	\${HOST.root_id}
hostModel	\${HOST.discovered_model:NA}
hostOs	\${HOST.discovered_os_name:NA}
ip_address	\${SNMP.application_ip}
smpDescription	\${SNMP.snmp_description:NA}
snmpCommunityPostfix	\${SOURCE.vlan_id}

## 使用スクリプト

- bridgePortDisc.py
- networking\_win.py
- SNMP\_Networking\_Utils.py

## 検出された CIT

- Bridge
- Composition
- Interface
- Layer2Connection
- Membership
- Node
- PhysicalPort
- Realization

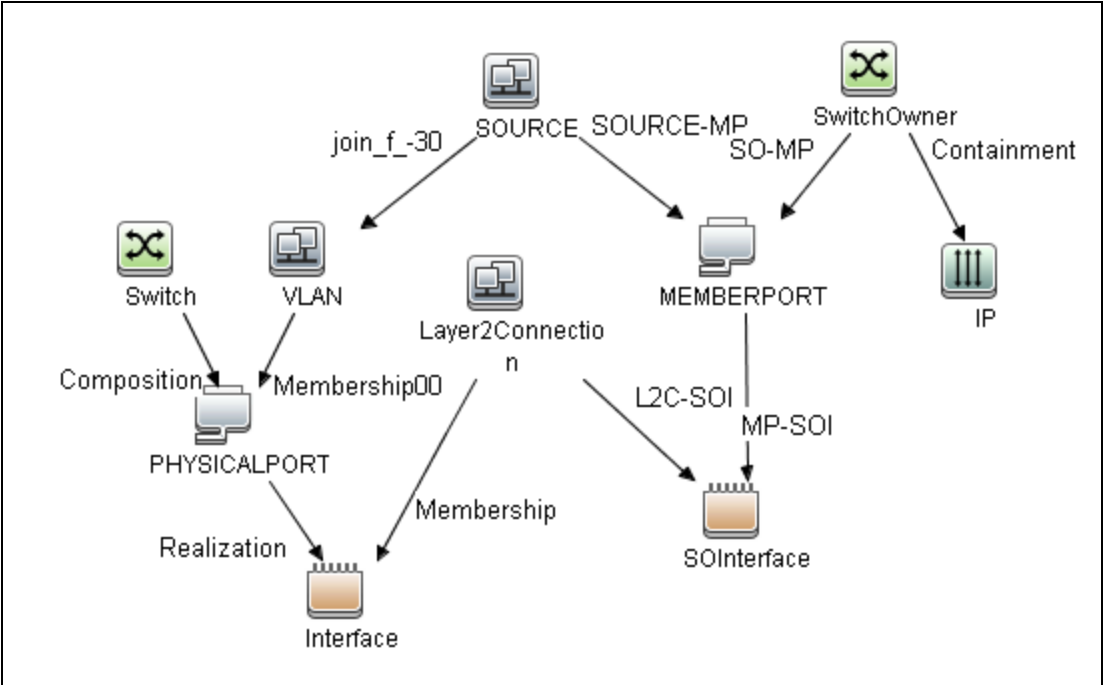
## Merge VLANs アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

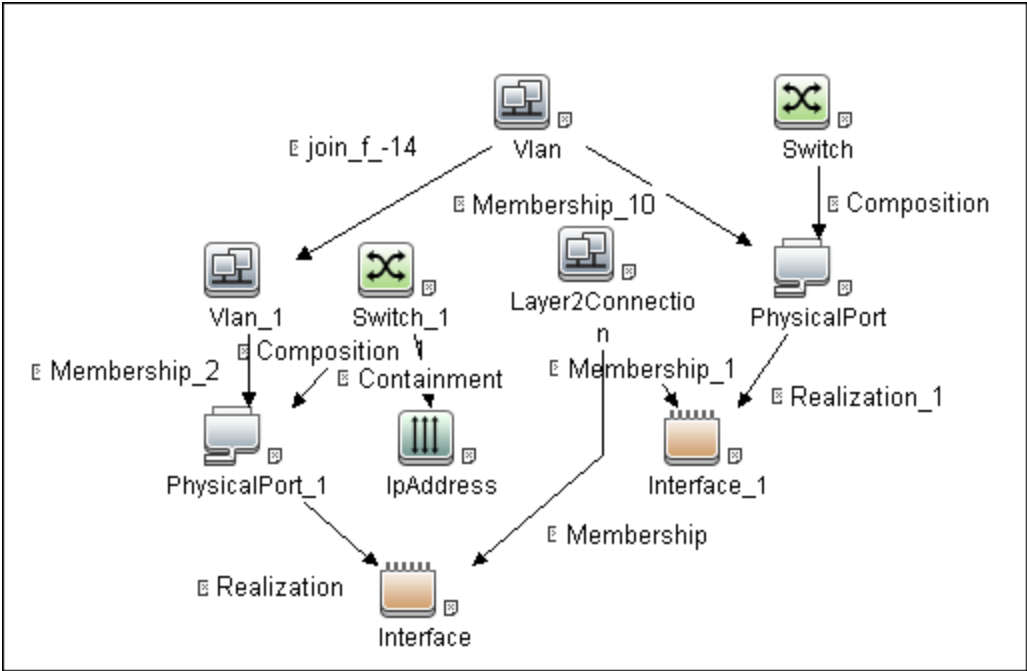
Vlan

### 入力クエリ



ノード名	条件
IP	NOT IP Probe Name null

トリガ TQL クエリ



ノード名	条件
IpAddress	NOT IP Probe Name null

トリガ CI データ

名前	値
memberId	\${MEMBERPORT.root_id}
portId	\${PHYSICALPORT.root_id}
vlanId	\${SOURCE.vlan_id}

使用スクリプト

merge\_vlans\_by\_ports.py

検出された CIT

- Membership



- PhysicalPort
- Vlan

## Process Layer2 Collected Files アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

Discovery Probe Manager

### 入力 TQL クエリ



### 使用スクリプト

processL2Files.py

### 検出された CIT

- Composition
- Interface
- Layer2Connection
- Membership
- Node

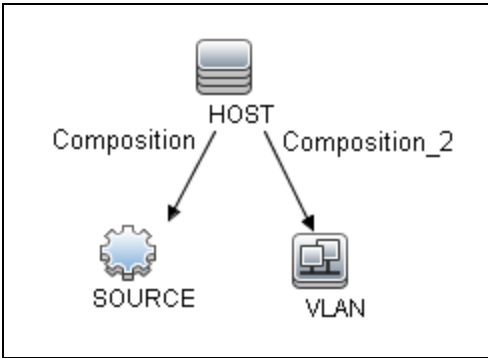
# VMS Catalyst by SNMP アダプタ

本 項 ではこのアダプタに 関 する 詳 細 を 説 明 し ます。

## 入 力 CIT

SNMP

## 入 力 TQL クエリ



## トリガ CI データ

名 前	値
credentialsId	\${SOURCE.credentials_id}
hostId	\${SOURCE.application_ip_domain}
ip_address	\${HOST.root_id}
snmpCommunityPostfix	\${VLAN.vlan_id}

## 使 用 スクリプト

SNMP\_Net\_Dis\_VMS\_catalyst.py

## 検 出 さ れ た CIT

- Bridge
- Composition

- Containment
- Dependency
- Membership
- PhysicalPort
- Vlan

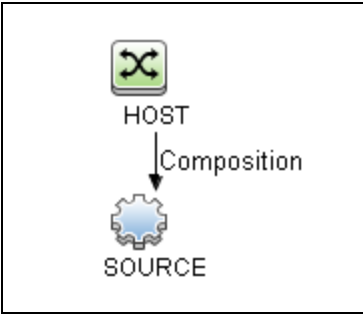
## Catalyst Vlans by SNMP アダプタ

本項ではこのアダプタに関する詳細を説明します。

### 入力 CIT

SNMP

### 入力 TQL クエリ



### トリガ CI データ

名前	値
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}
hostModel	\${HOST.discovered_model:NA}
hostOs	\${HOST.discovered_os_name:NA}
ip_address	\${SOURCE.application_ip}

名前	値
smpDescription	\${SOURCE.snmp_description:NA}

使用スクリプト

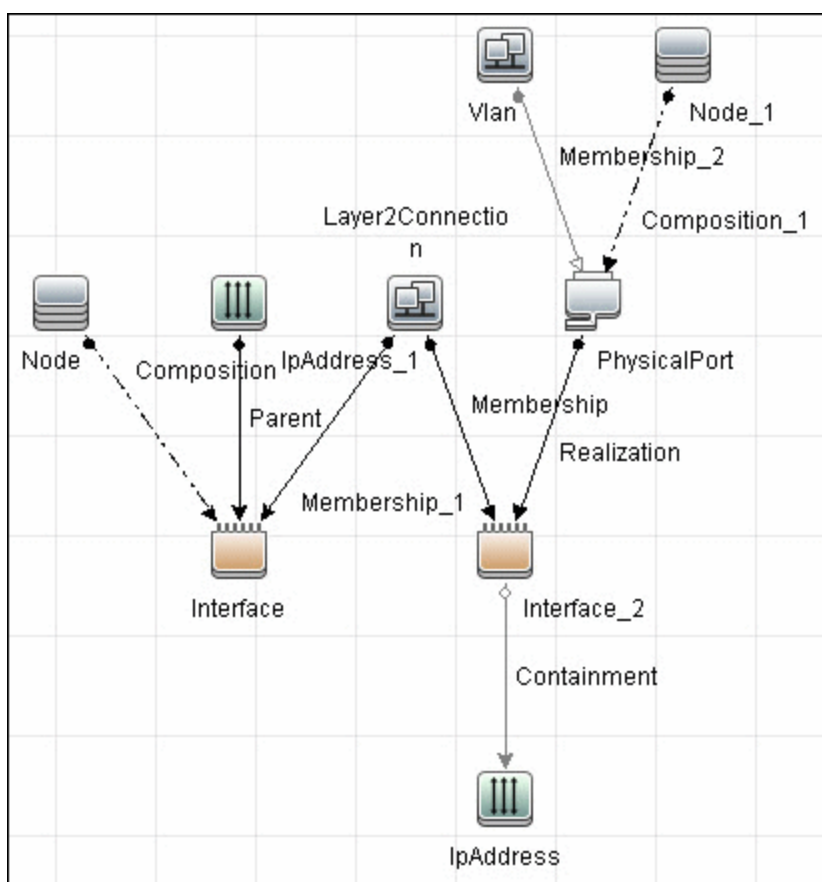
SNMP\_Net\_Dis\_Catalyst\_Vlans.py

検出された CIT

- Bcast Domain
- Composition
- ELAN
- ELAN-VLAN Map
- Membership
- PhysicalPort
- Vlan

## 関係

- Layer2 スイッチをそのポートに直接接続するか, VLAN 経由で接続できます。
- Bridge CIT は, ポートが配置されている基本 MAC アドレス(ネットワーク・インタフェース・カード)を表します。
- スイッチの各ポートを, Layer2 CI によってホストやインタフェース・オブジェクト(エンド・ユーザのマシン)に接続するか, バックボーン・リンクによってポート・スイッチに接続できます。



## トラブルシューティングおよび制限事項

本項では Layer2 ディスカバリのトラブルシューティングと制限事項について説明します。

- ディスカバリで空の結果が返された場合は、検出された SNMP エージェント(または、特別なコミュニティ認証を使用している SNMP エージェント)へのアクセス権があること、および要求されたすべての MIB テーブルが Data Flow Probe マシンからの SNMP 要求に回答していることを確認します。MIB テーブルの詳細については、該当するスクリプトを参照してください。
- レポートされたブリッジの MAC アドレスが 000000000000, "", または null の場合、アダプタは結果をレポートしません。
- 取得した基本ブリッジの MAC (1.3.6.1.2.1.17.1.1 テーブルから取得)が、宛先データの所定の bridgeId と同じでない場合、アダプタはゼロの結果を返します。  
SNMP\_Dis\_L2\_Bridge の場合は、bridge\_basemacaddr によって bridgeId が設定されます。  
SNMP\_Dis\_L2\_VLAN の場合は、vlan\_bridgemac によって bridgeId が設定されます。

## 第71章: No-Credentials ディスカバリ

### 本章の内容

概要 .....	1085
Data Flow Probe マシンの設定方法 .....	1086
Host Fingerprint with nmap の検出方法 .....	1087
Range IPs by nmap の検出方法 .....	1087
Host Fingerprint using nmap ジョブ .....	1088
Range IPs by nmap ジョブ .....	1090
トラブルシューティングおよび制限事項 .....	1092



## 概要

Nmapとは、未処理のIPパケットを使用してネットワークで使用可能なホスト、ホストが提供しているサービス、実行されているオペレーティング・システムなどを判断するネットワーク検索用のユーティリティです。

Nmapは、オペレーティング・システムの結果の精度(例: 精度 80%)も計算します。

DFMは次のジョブで nmap ユーティリティを使用します。

- **Host Fingerprint using nmap**

このジョブは、ホスト CI で `host_osaccuracy` 属性の Nmap の精度値をレポートします。

- **Range IPs by NMAP**

このジョブはライブ・ホストでレポートします。

## Data Flow Probe マシンの設定方法

**Host Fingerprint using nmap** ジョブまたは **Range IPs by nmap** ジョブのいずれかまたは両方を実行することになっているすべての Data Flow Probe マシンで次の手順を実行します。

1. **C:\hp\UCMDB\DataFlowProbe\tools\nmap\_install** から **nmap-6.25-setup.exe** を実行します。
2. ライセンス条件を承認して **[I agree]** をクリックします。 **[Choose Components]** ダイアログ・ボックスが開きます。
3. **[Nmap Core Files]**, **[Register Nmap Path]**, および **[WinPcap 4.2.1]** を選択します。
4. **[Next]** をクリックします。

**[Choose Install Location]** ダイアログ・ボックスが開きます。

5. 標準設定の場所を受け入れるか別の場所を入力します。 **[Install]** をクリックします。

Nmap がインストールされます。WinPcap インストール・ダイアログ・ボックスは、Nmap のインストールが完了した直後に開きます。

6. ライセンス条件を承認して **[Next]** をクリックします。 **[Choose Install Location]** ダイアログ・ボックスが開きます。
7. 標準設定の場所を受け入れるか別の場所を入力します。 **[Install]** をクリックします。

**[Finished]** ダイアログ・ボックスが開きます。

8. **[Finish]** をクリックします。WinPcap の **[Options]** ダイアログ・ボックスが開きます。
9. チェック・ボックスをクリアして **[Next]** をクリックします。
10. **[Finish]** をクリックします。

次のソフトウェアが Data Flow Probe マシンに追加されます。

- Nmap 6.2.5
- winpcap-nmap 4.1.2
- Microsoft Visual C++ Redistributable - x86 2010

確認するには、 **[プログラムの追加/削除]** ウィンドウにアクセスしてください。

## Host Fingerprint with nmap の検出方法

このタスクでは、**Host Fingerprint using nmap** ジョブを使用したホスト、オペレーティング・システム、ネットワーク・インタフェース、アプリケーション、実行中のサービスの検出方法について説明します。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

### 2. 前提条件 - Data Flow Probe マシンの設定

[「Data Flow Probe マシンの設定方法」\(1086ページ\)](#)を参照してください。

### 3. ディスカバリの実行

このジョブは、検出されたすべての IP アドレスでトリガされます。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Range IPs by nmap の検出方法

本項の内容

### 1. 前提条件 - Data Flow Probe マシンの設定

[「Data Flow Probe マシンの設定方法」\(1086ページ\)](#)を参照してください。

### 2. ディスカバリの実行

**Range IPs by nmap** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## Host Fingerprint using nmap ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは **OS\_Fingerprint** アダプタを使用します。

### パラメータ

パラメータを表示するには、**[Universal Discovery] > [ディスカバリ モジュール/ジョブ] > [Network Infrastructure] > [No-Credentials Discovery] > [Host Fingerprint using nmap] > [プロパティ] タブ > [パラメータ]** 表示 枠に移動します。

パラメータの上書きの詳細については、『HP Universal CMDB Data Flow Management Guide』の「Parameters Pane」を参照してください。

パラメータ	説明
Create_Application_CI	<b>True</b> : ポート・フィンガープリント情報に基づいてアプリケーション CI を作成する。
Perform_Port_Fingerprints	<b>True</b> : 開いているポートの検出を試みる。
discover_os_name	<b>True</b> : ホスト OS を検出する。これは不正確な場合もある。
nmap_host_timeout	nmap が単一ホストのスキャンに費やせる時間(秒単位)。
nmap_location	nmap 実行可能ファイルへの完全パス。 例 : <b>C:\Program Files (x86)\Nmap\nmap.exe</b>  注: 空の場合、ジョブはシステム・パスを検索します。
scan_known_ports_only	<b>portNumberToPortName.xml</b> ファイルに示されるポートをスキャンする。 標準設定 : <b>False</b>
scan_these_ports_only	スキャンするポートの範囲を制限する。例 : T:1-10,42,U:1-30(TCP ポート 1 から 10 および 42 および UDP ポート 1-30 を検出する)。このパラメータが空のままの場合、Nmap の標準設定が使用される。

## 検出された CIT

検出された CIT を表示するには, [リソース] 表示 枠 で 特定 のアダプタを選択します。

詳細については、『HP Universal CMDB Data Flow Management Guide』の「Discovered CITs Pane」を参照してください。

## Range IPs by nmap ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは **IpRange\_by\_nmap** アダプタを使用します。

### パラメータ

パラメータを表示するには、**[Universal Discovery] > [デイスカバリ モジュールジョブ] > [Network Infrastructure] > [Basic] > [Range IPs by nmap] > [プロパティ] タブ > [パラメータ]** 表示 枠に移動します。

パラメータの上書きの詳細については、『HP Universal CMDB Data Flow Management Guide』の「Parameters Pane」を参照してください。

名前	説明
excludePatternsList	セミコロンで区切られたワイルドカード・パターンのリスト。いずれかのパターンに一致した IP アドレスは、スキップされます。パターンには、数字、ドット、ワイルドカード *(0 個以上の文字に一致)、? (正確に 1 文字に一致)を含めることができます。
nmap_location	<b>nmap</b> 実行可能ファイルへの完全パス。 例: <b>C:\Program Files (x86)\Nmap\nmap.exe</b>  <b>注:</b> 空の場合、ジョブはシステム・パスを検索します。
range	ping を実行する IP の範囲 (セミコロンで区切る)。例: 1.2.3.0-1.2.3.10;1.2.3.50-1.2.3.60

### 検出された CIT

IpAddress

### デイスカバリ・フロー

デイスカバリは、次に示すようにプローブで指定された各範囲について実行されます。

1. 範囲で IP をフィルタリングします(**excludePatternsList** パラメータで指定されたパターンを適用)。
2. フィルタリングされた IP で ping スキャンを実行します。

注: Nmap は一度に 10 個のフィルタリングされた IP の ping スキャンを実行します。そのため、100 個の IP が確認のために渡されると、nmap コマンドは 10 回実行されます。これは、特に Windows におけるコマンド・ライン・サイズ制限のためです。

3. 次の範囲を処理する前に検出した IP を UCMDB に送信します。

## トラブルシューティングおよび制限事項

本項では、No-Credentials デイスカバリのトラブルシューティングと制限事項について説明します。

エラー・メッセージ	理由	解決策
Can't parse XML document with Nmap results.Skipped.	nmap.exe が失敗し、有効な XML ファイルを作成できなくなった。	<ul style="list-style-type: none"> <li>Nmap ジョブを再起動してみる。</li> <li>Nmap ジョブのスレッド数を減らしてみる。</li> </ul>
Error nmap result file is missing	nmap.exe が失敗し、XML ファイルを作成できなくなった。	<ul style="list-style-type: none"> <li>Nmap ジョブを再起動してみる。</li> <li>Nmap ジョブのスレッド数を減らしてみる。</li> </ul>
The system cannot execute the specified program (in the communication log file)	Windows システムが Nmap アプリケーションを起動できない。	<p>次のことを確認する。</p> <ul style="list-style-type: none"> <li>正しい Nmap バージョンをダウンロードして、インストールされていること。</li> <li>WinPcap がインストールされていること。</li> </ul> <p>これらのインストールの詳細については、<a href="#">「前提条件 - プロトコル資格情報の設定」(1087ページ)</a>を参照してください。</p> <p>Nmap および WinPcap をインストールしていてもこのエラー・メッセージがまだ通信ログに表示される場合、次のパスから <b>vcredist_x86.exe</b> をインストールする：  <b>C:\hp\UCMDB\DataFlowProbe\runtime\probeManager\discoveryResources</b></p>
Nmap is not installed on Probe machine	Nmap が Probe マシンにインストールされていない。	<p>コマンド・ラインから Nmap を起動してみる。</p> <p>Nmap がインストールされていることを確認する。インストールの詳細については、<a href="#">「前提条件 - プロトコル資格情報の設定」(1087ページ)</a>を参照してください。</p>



## 第72章: ネットワーク接続のアクティブおよびパッシブ・ディスカバリ

### 本章の内容

概要 .....	1094
サポートされるバージョン .....	1094
トポロジ .....	1095
プロセスの検出方法 .....	1096
TCPトラフィック・ジョブ .....	1097
Network Connectivity Data Analyzer ジョブ .....	1098
TcpDiscoveryDescriptor.xml ファイル .....	1101
サーバ検出方法 .....	1101
フィルタリング .....	1102
レポートイング .....	1103

## 概要

このモジュールのすべてのジョブは、Data Flow Probe の PostgreSQL データベースに対してクエリを実行し、**Host Resources** および **Host Applications**, **TCP By Shell/SNMP**, または **Collect Network Data by Netflow** の各ジョブで挿入されたネットワーク接続情報を取得します。

Host Resource ジョブの詳細については、[「ホストのリソースおよびアプリケーションのデイスカバリ」](#)(711ページ)を参照してください。

Data Flow Probe には PostgreSQL データベースが組み込まれているため、NetFlow 用に別の PostgreSQL インスタンスをインストールする必要はありません。代わりに、専用のスキーム(歴史的背景により netflow と呼ぶ)にデータを保存します。

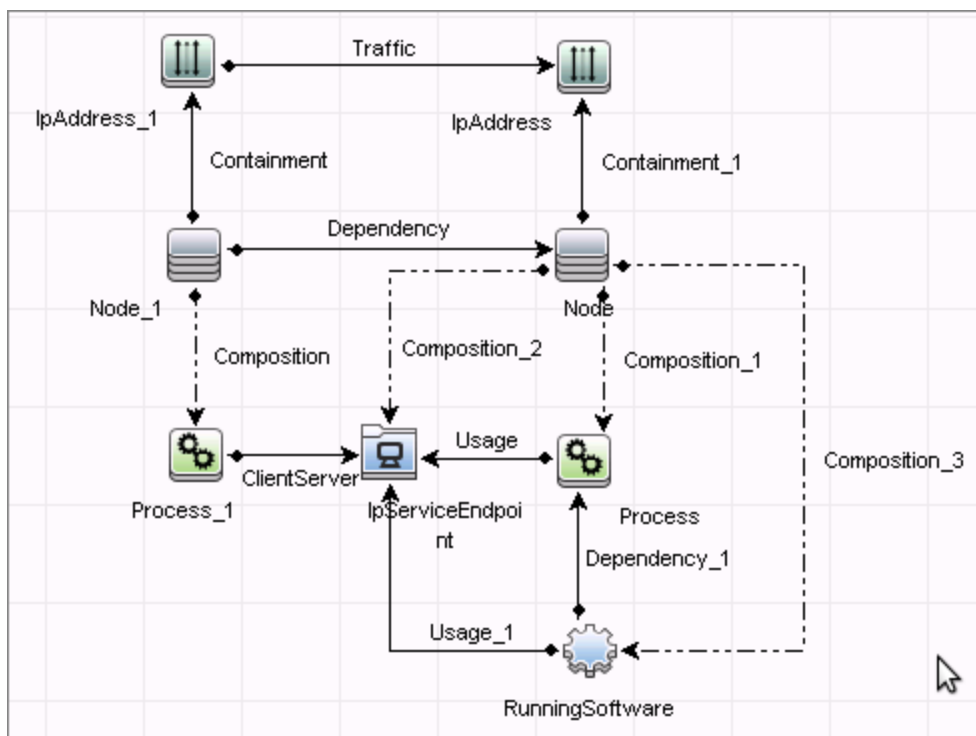
## サポートされるバージョン

このデイスカバリは NetFlow バージョン 5 および 9 をサポートします。

**注:** 標準設定では、NetFlow バージョンのサポートが有効になっています。任意のバージョンを無効にするには、**NetFlow.properties** ファイルで **flow.collector.V{netflow\_version}.enabled** プロパティを **false** に設定します。

## トポロジ

### ネットワーク接続のパッシブ・ディスカバリ



## プロセスの検出方法

このタスクでは、プロセスの検出方法について説明します。

本項の内容

### 1. 前提条件 - プロトコル資格情報の設定

ネットワーク接続を検出するには、次のプロトコルを定義します。

- SNMP プロトコル
- NTCMD プロトコル
- SSH プロトコル
- Telnet プロトコル
- WMI プロトコル

資格情報の詳細については、『HP Universal CMDB Discovery and Integration Content Guide - Supported Content』ドキュメントの「Supported Protocols」を参照してください。

注: これらのどのプロトコルも必須ではありませんが、WMI のみではネットワーク・データが取得されません。

### 2. ディスカバリの実行

次のジョブを次の順序で実行します。

- **TCP Data by Shell** または **TCP Data by SNMP** ジョブを実行し、リモート・マシンから収集した TCP 情報を Probe の PostgreSQL データベースにポピュレートします。詳細については、[「TCPトラフィック・ジョブ」\(1097ページ\)](#)を参照してください。
- **Network Connectivity Data Analyzer** ジョブを実行します。ジョブの詳細については、[「Network Connectivity Data Analyzer ジョブ」\(1098ページ\)](#)を参照してください。

## TCP トラフィック・ジョブ

**TCP Data by Shell** ジョブおよび **TCP Data by SNMP** ジョブでは、TCP トラフィックに関する情報を収集できます。これらのジョブは CI を CMDB に送信しませんが、Data Flow Probe のデータベース内の既存のデータに対してクエリを実行します。

これらのジョブは以下のパラメータによって強化されます。このパラメータを使用することで TCP データをキャプチャできるほか、キャプチャ間の時間遅延を設定できます。

パラメータ	説明
<b>CaptureProcessInformation</b>	<p><b>true</b> : プロセス情報をキャプチャして Data Flow Probe のデータベースに格納する。CI はレポートされない。Host Resources ジョブおよび Host Applications ジョブで使用される方法と同じ方法でプロセスをキャプチャする。</p> <p>Host Resource ジョブの詳細については、『HP UCMDB ディスカバリ / インテグレーション・コンテンツ・ガイド - ディスカバリ・モジュール』ドキュメントの「ホストのリソースおよびアプリケーションのディスカバリ」を参照してください。</p>
<b>DelayBetweenTCPSnapshots</b>	<p>TCP スナップショットのキャプチャから次のキャプチャまでの秒数。標準設定は 5 秒。1 回のジョブ呼び出し中にいくつかの TCP スナップショットを取ることは、より詳細なデータを取得する上で役立つ可能性がある。たとえば、リモートの Windows システムで <b>netstat -noa</b> コマンドを実行して TCP 情報を収集する場合は、このパラメータを使用して、コマンドの実行中に 5 秒間隔でプロセス情報をキャプチャできる。</p>
<b>NumberOfTCPSnapshots</b>	<p>取得する TCP スナップショットの数。</p>
<b>IsofPath</b>	<p>UNIX マシンのプロセス通信の検出を可能にする lsof コマンドのパス。標準設定値は <b>/usr/local/bin/lsof,lsof,/bin/lsof</b>。</p>
<b>useLSOF</b>	<p><b>true</b> : ディスカバリは <b>Isof</b> ユーティリティを使用して、UNIX マシン上のポートとプロセス間のマッピングの検出を試みる。</p> <p><b>標準設定</b> : True</p>
<b>useNetstatOnly</b>	<p>追加コマンド (lsof および pfiles) を実行するか、または netstat コマンドのみを使用するかを指定する。</p> <p><b>標準設定</b> : False</p>

## Network Connectivity Data Analyzer ジョブ

このジョブを使用すると、ユーザは IT サーバ・インフラストラクチャから TCP 通信情報をキャプチャして UCMDb 内でモデル化できます。カスタマイズしたトポロジをレポートするようにこのジョブを設定できます。詳細については、「[TcpDiscoveryDescriptor.xml ファイル](#)」(1101 ページ)を参照してください。

### アダプタ

このジョブでは、Network\_Connectivity\_Data\_Analyzer アダプタを使用します。

• アダプタ・パラメータ

パラメータ	説明
<b>acceptedServices</b>	<p>レポートするサービスのリストを指定する(ssh, oracle, mysql など)。</p> <p>値を <b>known_services</b> に設定した場合, サーバで実行中のソフトウェアのうち, <b>portNumberToPortName.xml</b> ファイル内で <b>discover="1"</b> として設定されているサービスを表しているソフトウェアのみがレポートされる。</p> <p>値を '*' に設定すると, 検出されたすべてのサービスがレポートされる。</p>
<b>discoveryDescriptorFile</b>	IP 範囲による分析およびレポート方法を定義するために使用するジョブ構成ファイルのフル・パス。
<b>includeOutscopeClients</b>	<p><b>True</b> : 範囲外のクライアントのレポートを有効にする。</p> <p><b>False</b> : 範囲外のクライアントのレポートを無効にする。</p>
<b>includeOutscopeServers</b>	<p><b>True</b> : 範囲外のサーバのレポートを有効にする。</p> <p><b>False</b> : 範囲外のサーバのレポートを無効にする。</p>
<b>reportIpTrafficLink</b>	<p><b>True</b> : トラフィック・リンクのレポートを有効にする。</p> <p><b>False</b> : トラフィック・リンクのレポートを無効にする。</p>
<b>reportNodeDependencyLink</b>	<p><b>True</b> : 依存関係リンクのレポートを有効にする。</p> <p><b>False</b> : 依存関係リンクのレポートを無効にする。</p>
<b>reportServerRunningSoftware</b>	<p><b>True</b> : サーバで実行中のソフトウェアのレポートを有効にする。</p> <p><b>False</b> : サーバで実行中のソフトウェアのレポートを無効にする。</p>

## 検出された CIT

- Client-Server : DFM は, どのマシンがサーバでどのマシンがクライアントかを決定します。

- 一方のエンドがリスナ・ポートとして検出された場合、そのエンドがサーバとみなされます。
  - 一方のエンドが **StatisticBasedApproach** の最小条件を満たす場合 (サーバ検出方法に関する項を参照)、そのエンドがサーバとみなされます。
  - どちらのエンドにもポートへの接続が 1 つしかない場合、DFM はポートと **portNumberToPortName.xml** ファイルをチェックして、そのエンドがサーバかどうかを識別します ([**アダプタ管理**] > [**リソース**] ペインで [**リソースの検索**] ボタンをクリックし、[**名前**] フィールドに「**portNumberToPortName.xml**」と入力して [**次を検索**] ボタンをクリックします)。
- Composition
  - Containment
  - Dependency : 検出されたクライアントとサーバ間でリンクを設定します。
  - IpAddress
  - IpServiceEndpoint
  - Node
  - Traffic : 検出された IP アドレス間でリンクを設定します。
  - Usage



## TcpDiscoveryDescriptor.xml ファイル

**TcpDiscoveryDescriptor.xml** ファイルは、IP 範囲による分析およびレポートを行うためのルールを定義します。

本項の内容

- [「サーバ検出方法」\(1101ページ\)](#)
- [「フィルタリング」\(1102ページ\)](#)
- [「レポートイング」\(1103ページ\)](#)

### サーバ検出方法

**serverDetectionApproach** タグには、クライアント・サーバ関係を解決するために使用する方法のリストが含まれます。

<b>ListenPortsBasedApproach</b>	<b>LISTEN</b> または <b>ESTABLISHED</b> の接続状態に基づいて関係を解決します。この方法を使用できるようにするには、プロセス間ディスカバリを実行する必要があります。ポートがリッスン用に使われている場合、そのホストはサーバとして解決されます。したがって、接続のもう1つのメンバーは自動的にクライアントとして解決されます。ポートがリッスン用に使われていない場合は、この逆です。
<b>KnownPortsBasedApproach</b>	<b>portNumberToPortName.xml</b> ファイルで定義されている既知のサーバ・ポート・リストに基づいて関係を解決します。
<b>StatisticBasedApproach</b>	最小条件に基づいて関係を解決します。条件の値がゼロの場合、その条件は考慮されません。有効な条件は次のとおりです。 <ul style="list-style-type: none"><li>• <b>minClients</b> : ホストをサーバとして示すための最小接続数。</li><li>• <b>minPackets</b> : ホストをサーバとして示すための、ホストにより送受信された最小の合計パケット数。</li><li>• <b>minOctets</b> : ホストをサーバとして示すための、ホストにより送受信された最小の合計オクテット数。</li></ul>

**注:** 方法のアクティブ属性を **false** に設定するか、方法に関与するタグをコメント・アウトまたは削除すると、方法を非アクティブにできます。

## フィルタリング

**Filtering** セクションでは、検出されたクライアントとサーバに適用するフィルタ・ルールを定義します。

注: ホストに対して少なくとも1つのフィルタが適用された場合、そのホストはフィルタされます。

フィルタ・タイプは次のとおりです。

- ・「[範囲フィルタ](#)」(1102ページ)
- ・「[サービス・フィルタ](#)」(1103ページ)

### 範囲フィルタ

範囲フィルタでは、IP 範囲に基づいてフィルタリングを行います。

例：

```
範囲フィルタ定義
<ranges>
  <include>
    <range>probe_ranges</range>
  </include>
  <exclude>
    <range>outscope_clients</range>
  </exclude>
</ranges>
```

最終のレポート・トポロジに含める必要がある範囲を **<include>** タグで定義する必要があります。除外する必要がある範囲を **<exclude>** タグで定義する必要があります。特定の範囲を定義するには、次のキーワードを使用する必要があります。

キーワード	説明
probe_ranges	プロトコル・マネージャを使用して定義したすべての範囲を含めます。
outscope_clients	プローブ範囲の外側にあるすべてのクライアント IP を含めます。
outscope_servers	プローブ範囲の外側にあるすべてのサーバ IP を含めます。
ddm_related_connections	プローブ IP を含めます。ユーザはディスカバリ・プロセス中に開始された DFM 関連の接続をフィルタできます。

## サービス・フィルタ

サービス・フィルタでは、指定したサービス・リストに従って、検出されたサーバのフィルタリングを行います。**portNumberToPortName.xml** ファイル内の定義に従って、サービス名と、関連するポート間のマッピングを行います。

例：

### 範囲フィルタ定義

```
<services>
  <include>
    <service name="*" />
  </include>
  <exclude>
    <service name="ssh" />
  </exclude>
</services>
```

最終のレポート・トポロジに含める必要があるサービスを **<include>** タグで定義します。除外する必要があるサービスを **<exclude>** タグで定義します。**service name** の値が "\*" (アスタリスク) の場合、すべてのサーバが検出されます。

注：サービスのアクティブ属性を **false** に設定するか、サービスに関与するタグをコメント・アウトまたは削除すると、サービスを非アクティブにできます。

## レポーティング

**Reporting** セクションでは、フィルタ・ルールと、アクティブなレポータのリストを定義します。**configuration** タグは、すべてのレポータの標準設定のフィルタ・ルールを定義します。このボディ内で **<filtering>** タグを定義することで、レポータはフィルタ・ルールを上書きできます。各レポータは、レポートされるトポロジに関与します。

注：レポータのアクティブ属性を **false** に設定するか、レポータに関与するタグをコメント・アウトまたは削除すると、レポータを非アクティブにできます。

次のレポータを使用できます。

- 標準設定：詳細については、「[標準設定のレポータ](#)」(1104ページ)を参照してください。
- clientProcess**：詳細については、「[クライアント・プロセス・レポータ](#)」(1105ページ)を参照してください。

- **clientServerLink** : 詳細については、「[クライアント・サーバ・リンク・レポータ](#)」(1106ページ)を参照してください。
- **ipTrafficLink** : 詳細については、「[IPトラフィック・リンク・レポータ](#)」(1107ページ)を参照してください。
- **nodeDependencyLink** : 詳細については、「[ノード依存関係リンク・レポータ](#)」(1108ページ)を参照してください。
- **serverProcess** : 詳細については、「[サーバ・プロセス・レポータ](#)」(1109ページ)を参照してください。
- **serverRunningSoftware** : 詳細については、「[サーバで実行中のソフトウェアのレポータ](#)」(1110ページ)を参照してください。

## • 標準設定のレポータ

アクティブになっているレポータがない場合、ジョブは、**containment** 関係でリンクされている IP CI とノード CI のみを返します。



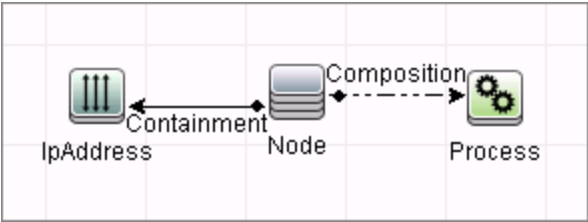
・ クライアント・プロセス・レポータ

このレポータはクライアント・プロセスをレポートします。

レポータ定義

```
<reporting>  
  <reporter name="clientProcess" active="true"/>  
</reporting>
```

トポロジ



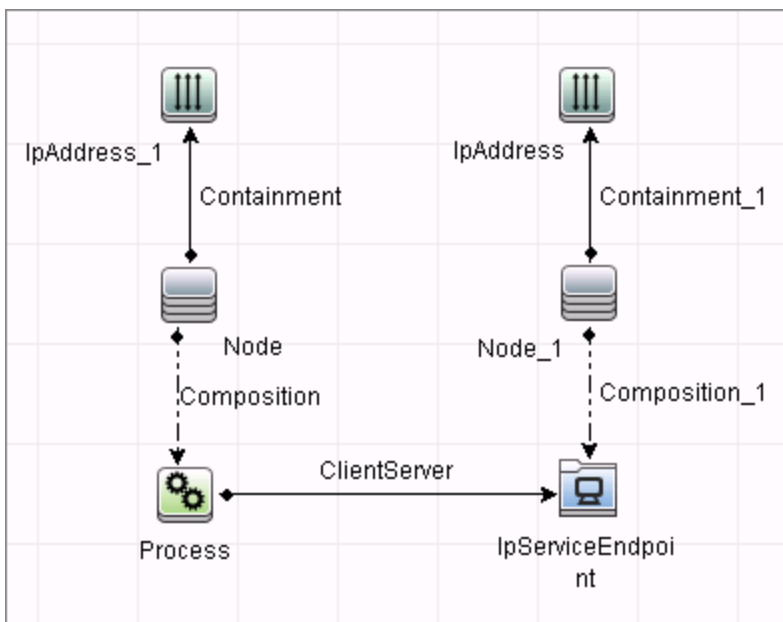
## ・ クライアント・サーバ・リンク・レポータ

このレポータは、クライアント・プロセス通信のエンドポイントと、(clientProcess active="false" の場合でも) クライアントとサーバ間のクライアント・サーバ・リンクをレポートします。

### レポータ定義

```
<reporting>  
  <reporter name="clientServerLink" active="true"/>  
</reporting>
```

### トポロジ



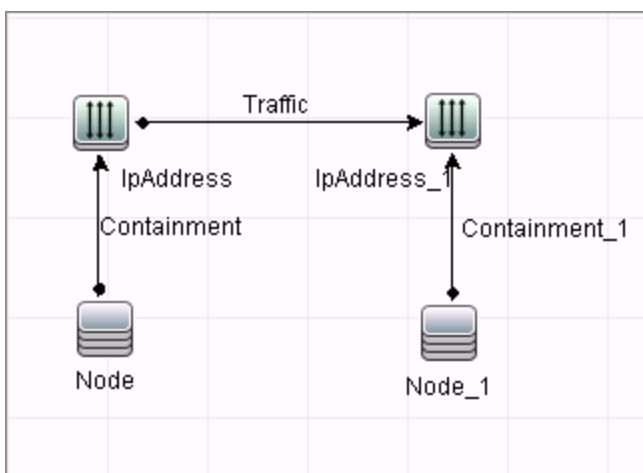
## ・ IP トラフィック・リンク・レポーター

このレポーターは、IP 間のトラフィック・リンクをレポートします。**reportTrafficDetails** 属性は、リンクの **octetCount**, **packetCount**, および **portset** 属性をジョブでレポートする必要があるかどうかを示します。

### レポーター定義

```
<reporting>
  <reporter name="ipTrafficLink" active="true" reportTrafficDetails="true"/>
</reporting>
```

### トポロジ



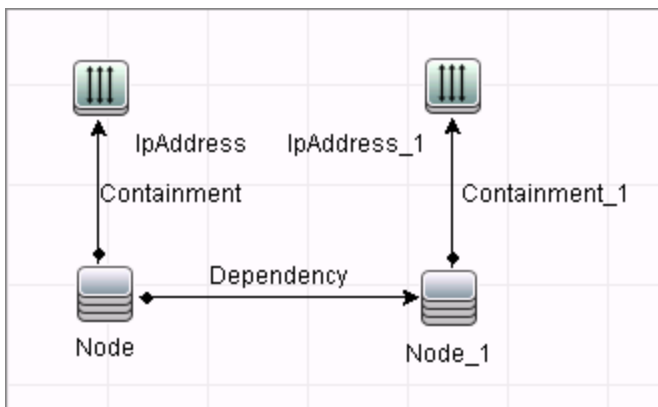
## ・ ノード依存関係リンク・レポータ

このレポータは、検出されたノード間の依存関係リンクをレポートします。

### レポータ定義

```
<reporting>  
  <reporter name="nodeDependencyLink" active="true"/>  
</reporting>
```

### トポロジ





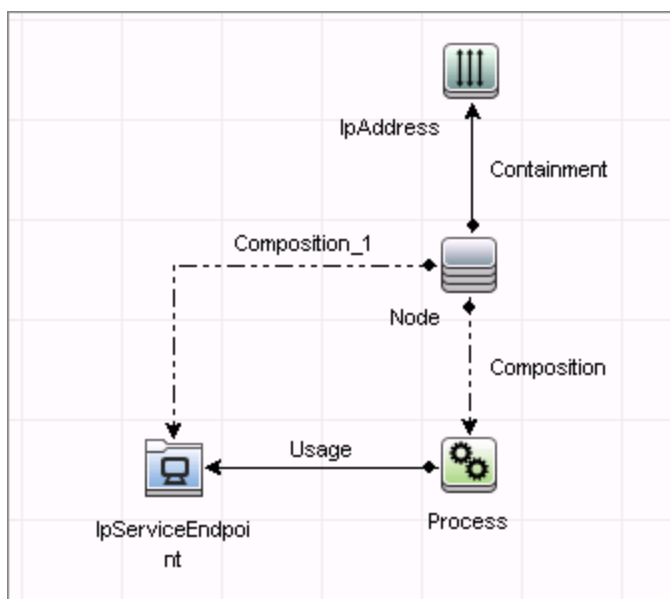
## ・ サーバ・プロセス・レポータ

このレポータは、サーバ・プロセスをレポートします。**linkWithCommunicationEndpoint** 属性は、検出された通信エンドポイントとプロセスをレポータが('usage' リンクで)リンクする必要があるかどうかを示します。

### レポータ定義

```
<reporting>
  <reporter name="serverProcess" active="true" linkWithCommunicationEndpoint="true"/>
</reporting>
```

### トポロジ



## ・サーバで実行中のソフトウェアのレポータ

このレポータは、通信エンドポイントと('usage' リンクで)リンクされている、サーバで実行中のソフトウェア、およびサーバ・プロセスをレポートします。**linkWithProcess** 属性は、サーバプロセスと検出された実行中のソフトウェアをレポータが('dependency' リンクで)リンクする必要があるかどうかを示します。

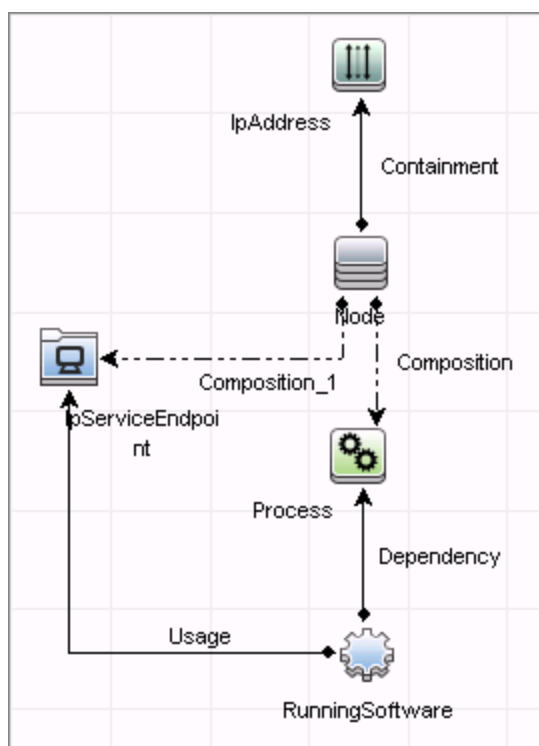
サーバで実行中のソフトウェアのレポートは **acceptedServices** パラメータによって異なります。

- **known\_services** に設定した場合、サーバで実行中のソフトウェアのうち、**portNumberToPortName.xml** ファイル内で **discover="1"** として設定されているサービスを表しているソフトウェアのみがレポートされます。
- '\*' に設定すると、検出されたすべてのサービスがレポートされます。
- TCP ポートまたはサービス名が含まれる場合、ジョブは、そのポートと連携している実行中のソフトウェアのみをレポートします。

### レポータ定義

```
<reporting>  
  <reporter name="serverRunningSoftware" active="true" linkWithProcess="true"/>  
</reporting>
```

## トポロジ



## 第16部: Tools and Samples > Discovery Samples

## 第73章: Dynamic Credential Sample ジョブ

本項ではこのジョブに関する詳細を説明します。

### はじめに

このジョブは、リモート・マシンに接続するための資格情報を動的に作成して使用方法を示すサンプルとして機能します。

### トリガ TQL

なし

### アダプタ情報

このジョブでは、Dynamic Credential Sample アダプタを使用します。

### アダプタ・タイプ

Jython

### 入力 CIT

Node

### 入力 TQL



### 使用スクリプト

DynamicCredentialSample.py

# 第74章: Config Parser アダプタ

このアダプタは、実行中のソフトウェアの構成ファイルを取得し、解析して、IP アドレスをレポートするための方法を示すサンプルとして機能します。

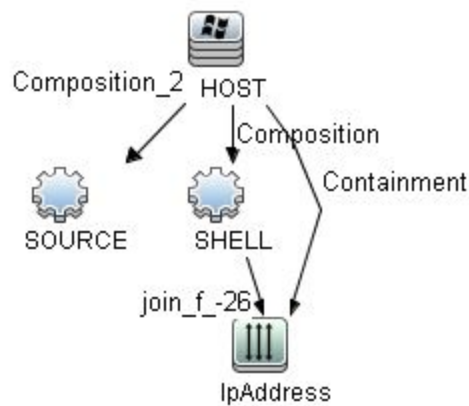
## アダプタ・タイプ

Jython

## 入力 CIT

RunningSoftware

## 入力 TQL



## トリガ CI データ

名前	値
Protocol	\${SHELL.root_class}
credentialsId	\${SHELL.credentials_id}
host_ips	\${IpAddress.name}
ip_address	\${SHELL.application_ip}
product_name	\${SOURCE.discovered_product_name}
version	\${SOURCE.version}

使用スクリプト

Config\_Parser\_Main.py

必要な権限

権限	操作	使用状況の詳細	オブジェクトとパラメータ
Shell	exec	Any command	Windows/UNIX : Any command -l

必要なディスカバリ・プロトコル

- NTCMD Protocol
- SSH Protocol
- Telnet Protocol
- Universal Discovery Protocol

検出された CIT

IpAddress

グローバル構成ファイル

globalSettings.xml

アダプタ・パラメータ

名前	値
configFileLocation	C:\configuration.txt
groupOfMatchResult	1
productName	Dataflow Probe
regexMatcher	^IP=(.*)\$

# 第75章: Execute Command アダプタ

このアダプタは、ホストでコマンドを実行する方法を示すサンプルとして機能します。

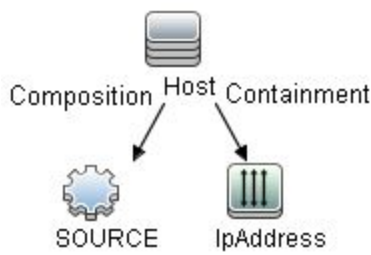
## アダプタ・タイプ

Jython

## 入力 CIT

Shell

## 入力 TQL



## トリガ CI データ

名前	値
Protocol	\${SOURCE.root_class}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${SOURCE.root_container}
host_ips	\${IpAddress.name}
ip_address	\${SOURCE.application_ip}
language	\${SOURCE.language:NA}

## 使用スクリプト

Execute\_Command\_Main.py



## 必要な権限

権限	操作	使用状況の詳細	オブジェクトとパラメータ
Shell	exec	Any command	Windows/UNIX : Any command -l

## 必要なディスカバリ・プロトコル

- NTCMD Protocol
- SSH Protocol
- Telnet Protocol
- Universal Discovery Protocol

## 検出された CIT

Node

## グローバル構成ファイル

globalSettings.xml

## アダプタ・パラメータ

名前	値
remoteCommandUNIX	echo "UNIX testing"
remoteCommandWin	echo "Windows testing"
storeToAttribute	data_note

## 第17部: Tools and Samples > Discovery Tools

# 第76章: File Monitor by Shell ジョブ

本項では File Monitor by Shell ジョブに関する詳細を説明します。

## トリガ・クエリ

なし

## ジョブ・パラメータ

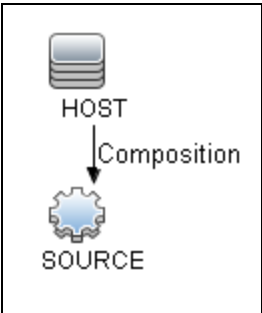
標準設定では、パラメータは上書きされず、アダプタからの値が使用されます。

## アダプタ情報

このジョブでは、FILE\_Mon アダプタを使用します。

## 入力 CIT

Shell



## トリガ CI データ

名前	値
codepage	\${SOURCE.codepage:NA}
connected_os_credentials_id	\${SOURCE.connected_os_credentials_id:NA}
credentialsId	\${SOURCE.credentials_id}
hostId	\${HOST.root_id}

名前	値
ip_address	\${SOURCE.application_ip}
language	\${SOURCE.language:NA}
Protocol	\${SOURCE.root_class}

## 使用スクリプト

- file\_ver\_lib.py
- file\_mon\_utils.py
- file\_mon.py

## 検出された CIT

- Composition
- ConfigurationDocument

## アダプタ・パラメータ

名前	標準設定値	説明
binary_file_extensions	exe, bin, dat	このパラメータは、どのタイプのファイルがバイナリかを定義し、該当するファイルからのコンテンツが読み取られないようにする。
discoverUnixHiddenFiles	false	このフラグは、Unix 系のシステムで非表示ファイルを検出するかどうかを決定する。 <b>true</b> の場合、すべての非表示ファイルが検出される。このパラメータは、extensions パラメータに割り当てられた値に依存しない。
extensions	cfg, conf, txt, xml, exe, bin, dat	このパラメータは、デイスカバリで探すファイル拡張子を定義する。
folders	C:\temp,D:\temp	ファイルの収集元であるフォルダのリスト。
recursively	false	このフラグは、フォルダを再帰的にスキャンするかどうかを決定する。

## グローバル構成ファイル

globalSettings.xml

## トラブルシューティングおよび制限事項

本項では **File Monitor by Shell** ジョブを実行する場合の、ファイル・ディスカバリのトラブルシューティングと制限事項について説明します。

- **File Monitor by Shell** は自動的にトリガされません。これは、このジョブのトリガ TQL クエリがないためです。すべての宛先に自動トリガがあると、Data Flow Probe でメモリ不足エラーが発生する可能性があります。この問題を解決するには、トリガされた CI を手動で追加します。
- **File Monitor by Shell** ジョブを実行する場合、2Mb を超えるファイルが検出されるとメモリ不足エラーが発生する可能性があります。

## 第77章: HP ServiceGuard and Oracle RAC ディスカバリ

### 本章の内容

概要 .....	1123
サポートされるバージョン .....	1123
Link DB DataFiles and Clustered FS ジョブの実行方法 .....	1123
Link DB Datafiles With Clustered FileSystem アダプタ .....	1124
Link DB DataFiles and Clustered FS ジョブ .....	1126

## 概要

このジョブは、HP ServiceGuard および Oracle RAC のサポートの一環です。導入されたメカニズムによって、FS リソースを介して Oracle データベース・インスタンスと HP ServiceGuard パッケージ間の間接リンクをレポートできます。

## サポートされるバージョン

このジョブは、Oracle RAC 10i が付属する HP-UX 10 および HP-UX 11 をサポートしています。

## Link DB DataFiles and Clustered FS ジョブの実行方法

### 1. 前提条件

このジョブは単に複雑なエンリッチメントであるというだけなので、資格情報は必要ありません。したがって、唯一の前提条件はジョブをトリガするために特定のトポロジがUCMDBに存在する必要があるということです。

### 2. ディスカバリの実行

次のジョブを実行します。

- a. **Range IPs by ICMP** ジョブを実行します。
- b. **Host Connection by Shell** ジョブを実行します。
- c. **Host Resources by Shell** ジョブおよび **Host Applications by Shell** ジョブを実行します。
- d. **Service Guard Cluster Topology** ジョブを実行します。
- e. **Oracle Topology by SQL** ジョブを実行します。
- f. **Link DB DataFiles And Clustered FS** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

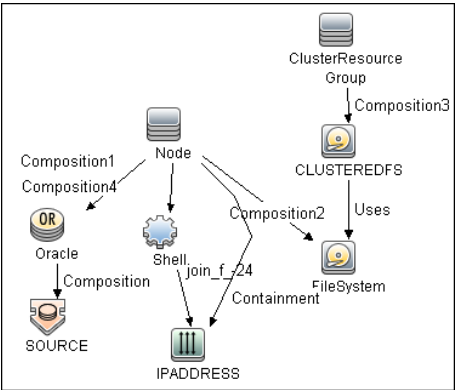
# Link DB Datafiles With Clustered FileSystem アダプタ

本項ではこのアダプタに関する詳細を説明します。

## 入力 CIT

DB Data File

## 入力クエリ



## トリガ CI データ

名前	値
dbFileId	\${SOURCE.root_id}
dbFilePath	\${SOURCE.name}
fsId	\${CLUSTEREDFS.root_id}
mountPoints	\${CLUSTEREDFS.mount_point}

## 使用スクリプト

- linkDbDatafileAndFs.py



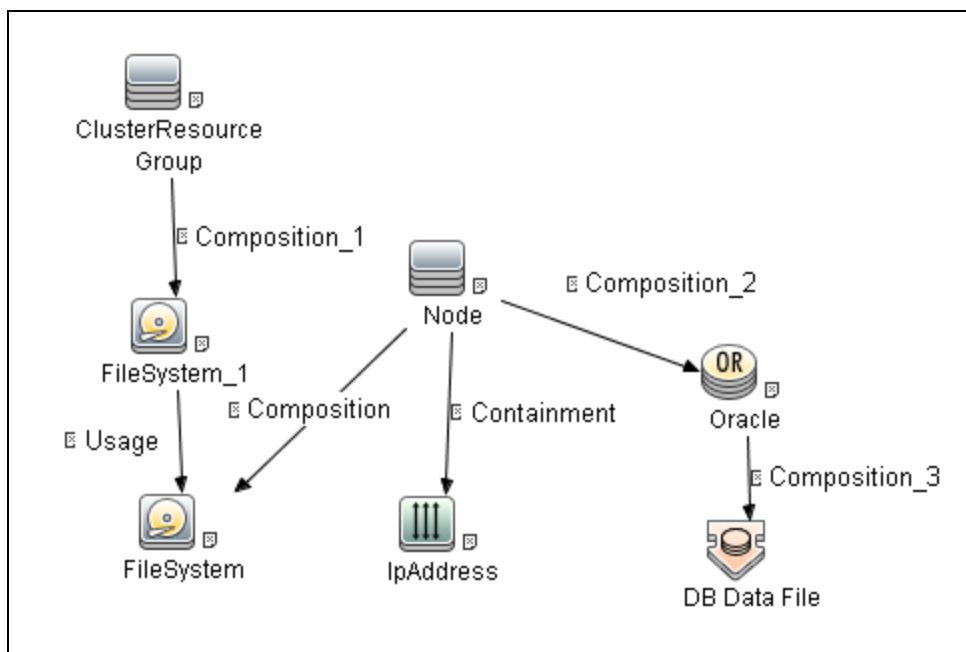
## 検出された CIT

- DB Data File
- FileSystem
- Node
- Usage

## Link DB DataFiles and Clustered FS ジョブ

本項ではこのジョブに関する詳細を説明します。

### トリガ・クエリ



### ディスカバリ・フロー

DB Data File と File System をリンクする方法は次のとおりです。

1. **Service Guard Cluster Topology by TTY** ジョブは、ServiceGuard パッケージのマウント・ポイントである File System Object をレポートします。そのため、これらは File System パッケージ・リソースです。
2. **Oracle Topology by SQL** ジョブは、Oracle DB および DB Data File をレポートします。
3. ClusteredResource Group に FS リソースを持つポロジがあり、このクラスタの少なくとも 1 つに検出された DB Data File で実行されている Oracle データベースがある場合、**Link DB DataFiles and Clustered FS** ジョブはすべてのマウント・ポイントおよび DB Data File を探します。このジョブは、これらの間の有効な関係があればそれを探し、新しいリンクとしてそれぞれをレポートします。

# 第78章: クラスタ化ソフトウェアの統合

## 本章の内容

概要 .....	1128
サポートされるソフトウェア .....	1128
クラスタ化ソフトウェアの統合方法 .....	1128
Merge Clustered Software ジョブ .....	1129

## 概要

このドキュメントでは、**Merge\_Clustered\_software** ディスカバリ・パッケージの使用法と機能について説明します。このパッケージでは、クラスタ・ノードに特定の RunningSoftware が存在することを示す CI をクラスタ化サービスに統合できます。

## サポートされるソフトウェア

このディスカバリ・パッケージでは、次のディスカバリがサポートされています。

- 次を含む HP ServiceGuard クラスタ：
  - Oracle データベース
  - Oracle TNS リスナ
  - Oracle iAS
- 次を含む Microsoft Cluster Server (MSCS)：
  - Microsoft SQL Server

## クラスタ化ソフトウェアの統合方法

ディスカバリ・ジョブをアクティブ化します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

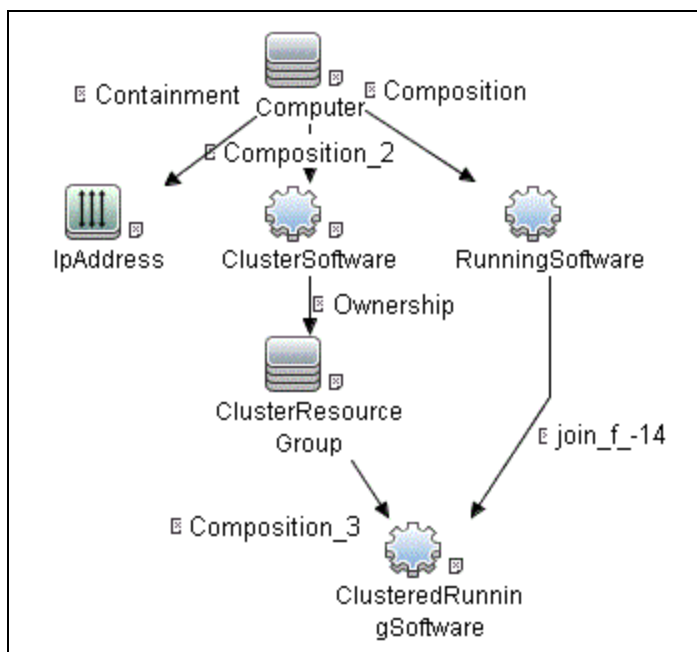
**注:** ディスカバリの範囲を広げるには、該当する CIT 名をパラメータに追加して、トリガ TQL クエリと入力 TQL クエリを更新します。その他の変更は不要です。

## Merge Clustered Software ジョブ

本項ではこのジョブに関する詳細を説明します。

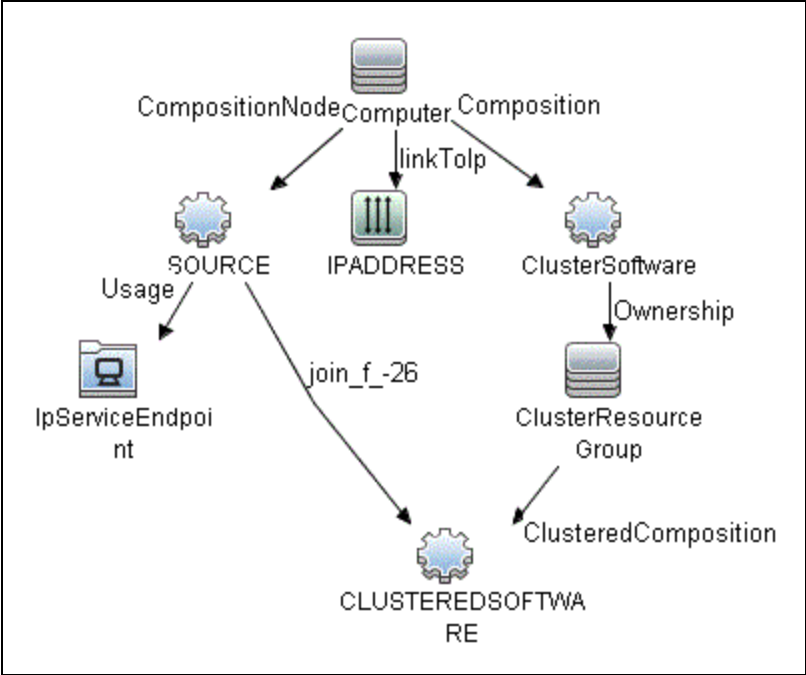
### トリガ TQL クエリ

次の図は、クラスタ化ソフトウェアを統合するためのトリガ TQL クエリを示しています。



### 入力 TQL クエリ

次の図は、クラスタ化ソフトウェアを統合するための入力 TQL クエリを示しています。



トリガ CI データ

名前	値
className	\${SOURCE.root_class}
clusteredContainer	\${ClusterResourceGroup.root_id}
clusteredUcmdbIds	\${CLUSTEREDSOFTWARE.root_id}
discProdName	\${CLUSTEREDSOFTWARE.root_id}
ipServiceEndpointIds	\${IpServiceEndpoint.root_id:}
localHostId	\${SOURCE.root_container}
localSoftwareId	\${SOURCE.root_id}
productName	\${SOURCE.product_name:NA}
remoteSoftwareName	\${CLUSTEREDSOFTWARE.name}
softwareName	\${SOURCE.name}

検出された CIT

- Node

- RunningSoftware

使用スクリプト

- mergeClusteredSoftware.py

作成または変更されたエンティティ

エンティティ名	タイプ	説明
mergeClusteredSoftware.py	スクリプト	ディスカバリ・スクリプト
Merge_Clustered_Software.xml	パターン	ディスカバリ・パターン
Merge Clustered Software.xml	ジョブ	ディスカバリ・ジョブ
mergeDiscClusteredSoft.xml	TQL クエリ	トリガ TQL クエリ

# 第79章: TCP/UDP Ports Discovery by Nmap

## 本章の内容

- 概要 .....1133
- サポートされるバージョン .....1133
- TCP/UDP Ports by Nmap の検出方法 .....1133
- TCP Ports ジョブ .....1134



## 概要

このディスカバリでは、特定のホストで、既知のサーバ・ポートの開いた TCP または UDP ポートを検出できます。

## サポートされるバージョン

このディスカバリでは、Nmap バージョン 6.25 以降をサポートしています。

## TCP/UDP Ports by Nmap の検出方法

### 1. 前提条件 - Data Flow Probe マシンの設定

TCP Ports ジョブを実行することになっているすべての Data Flow Probe マシンで次の手順を実行します。

- a. **C:\hp\UCMDB\DataFlowProbe\tools** から **nmap-6.25-setup.exe** を実行します。
- b. ライセンス条件を承認して **[I agree]** をクリックします。 **[Choose Components]** ダイアログ・ボックスが開きます。
- c. **[Nmap Core Files]**, **[Register Nmap Path]**, および **[WinPcap 4.1.2]** を選択します。
- d. **[Next]** をクリックします。

**[Choose Install Location]** ダイアログ・ボックスが開きます。

- e. 標準設定の場所を受け入れるか別の場所を入力します。 **[Install]** をクリックします。

Nmap がインストールされます。WinPcap インストール・ダイアログ・ボックスは、Nmap のインストールが完了した直後に開きます。

- f. ライセンス条件を承認して **[Next]** をクリックします。 **[Choose Install Location]** ダイアログ・ボックスが開きます。
- g. 標準設定の場所を受け入れるか別の場所を入力します。 **[Install]** をクリックします。

**[Finished]** ダイアログ・ボックスが開きます。

- h. **[Finish]** をクリックします。WinPcap の **[Options]** ダイアログ・ボックスが開きます。
- i. チェック・ボックスをクリアして **[Next]** をクリックします。
- j. **[Finish]** をクリックします。

次のソフトウェアが Data Flow Probe マシンに追加されます。

- Nmap 6.25
- winpcap-nmap 4.1.2
- Microsoft Visual C++ Redistributable - x86 2010

確認するには、**[プログラムの追加/削除]** ウィンドウにアクセスしてください。

## 2. ディスカバリの実行

次のジョブを実行します。

- a. **Range IPs by ICMP** ジョブまたは **Range IPs by nmap** ジョブを実行して、IP 範囲内で稼働中のマシンを検出します。
- b. **TCP Ports** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

## TCP Ports ジョブ

本項ではこのジョブに関する詳細を説明します。

### アダプタ

このジョブでは、**TCP Ports Discovery** アダプタを使用します。

## トリガ・クエリ



## ノード条件

ノード名	条件
IpAddress	NOT IP Probe Name null

## ジョブ・パラメータ

名前	標準設定値	説明
checkIfIPsReachable	true	このフラグは、ホストのポートが利用可能かどうかの確認をジョブが開始する前に、検出された IP に到達可能であることを確認する必要があるかどうかを示します。
checkOnlyKnownPorts	true	このフラグは、既知のポートのみをこのジョブで検出するかどうかを示します。このフラグによって、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータがキャンセルされることはありません。このフラグの <b>false</b> への設定は、 <b>ports</b> パラメータまたは <b>UDPports</b> パラメータ内の実際のポート範囲と合わせて使用する場合にのみ適用されます。
connectTimeout	5000	IP およびポートに接続するときのタイムアウトをミリ秒単位で示します。
nmapPath		nmap 実行可能ファイルのフルパスを示します (例 : C:\Program Files (x86)\Nmap\nmap.exe)。
pingTimeout	2000	ICMP ping のタイムアウトをミリ秒単位で示します。
ports	<b>JEE TCP Ports ジョブの場合 :</b> <ul style="list-style-type: none"> <li>weblogic,</li> <li>weblogicSSL,</li> </ul>	このパラメータには、デイスカバリの実行に使用される TCP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など) が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、また

名前	標準設定値	説明
	<p>websphere_ jmx, rmi</p> <p><b>Databases TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>oracle, db2, sybase, sql, mysql</li> </ul> <p><b>SAP TCP Ports ジョブの場合 :</b></p> <ul style="list-style-type: none"> <li>sap, sap_ jmx, sap_ http, sap_ https</li> </ul> <p><b>SAP TCP Ports ジョブの場合 : 標準設定値なし</b></p>	<p>は * の値が含まれている場合、ディスカバリはすべての既知の TCP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、<b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。</p>
<b>scanUDP</b>	false	<p>このフラグは、UDP ポートをスキャンするかどうかを示します。</p> <p><b>注:</b> UDP スキャンは、<b>useNMap=true</b> に設定されている場合にのみサポートされます(下記を参照)。</p>
<b>UDPports</b>		<p>このパラメータには、ディスカバリの実行に使用される UDP ポートのリストが格納されます。このリストには範囲、独立したポート番号、既知のプロトコル名 (http, ftp など)が含まれる可能性があり、カンマを使用して区切る必要があります。このリストが空の場合、または * の値が含まれている場合、ディスカバリはすべての既知の UDP ポートに対してのみ実行されます。1000-1100 などのポート範囲が入力された場合、<b>checkOnlyKnownPorts=true</b> が指定されていれば、ディスカバリは、その範囲内のすべての既知のポートに対してのみ実行されます。</p>
<b>useNMap</b>	<p><b>Databases TCP Ports ジョブおよび JEE TCP Ports ジョブの場合 :</b></p> <p>false</p>	<p>このフラグは、ポートのスキャン中に nmap を使用するかどうかを示します。</p> <p><b>注:</b> <b>nmapPath</b> にパスが指定されないと(上記を</p>

名前	標準設定値	説明
	<b>SAP TCP Ports</b> <b>ジョブおよび TCP</b> <b>Ports ジョブの場</b> <b>合 : true</b>	参照), システム・パスからの nmap が使用されま す。

注: ports パラメータまたは UDPports パラメータ内でポート名が割り当てられており, portNumberToPortName.xml 構成ファイル内で検出可能(isDiscovered=1)としてマークされているポートのみが検出対象になります。

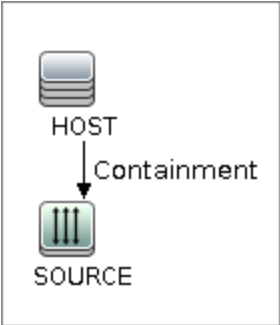
アダプタ情報

このアダプタでは, TCP ポートを検出します。

入力 CIT

IpAddress

入力クエリ



トリガ CI データ

名前	値
ip_address	\${SOURCE.name}
ip_domain	\${SOURCE.routing_domain}

## 使用スクリプト

- TcpPortScanner.py
- nmap.py

## グローバル構成ファイル

portNumberToPortName.xml

## 検出された CIT

- Composition
- Containment
- IpAddress
- IpServiceEndpoint
- Node

## 第18部: Tools and Samples > SSL Certificate Discovery

## 第80章: SSL Certificate ディスカバリ

### 本章の内容

概要 .....	1141
サポートされるバージョン .....	1141
トポロジ .....	1141
ディスカバリ・メカニズム .....	1141
SSL Certificates の検出方法 .....	1142
SSL Certificate Discovery by HTTPS ジョブ .....	1143
SSL Certificate Discovery アダプタ .....	1145
追加情報 .....	1146
トラブルシューティングおよび制限事項 .....	1146



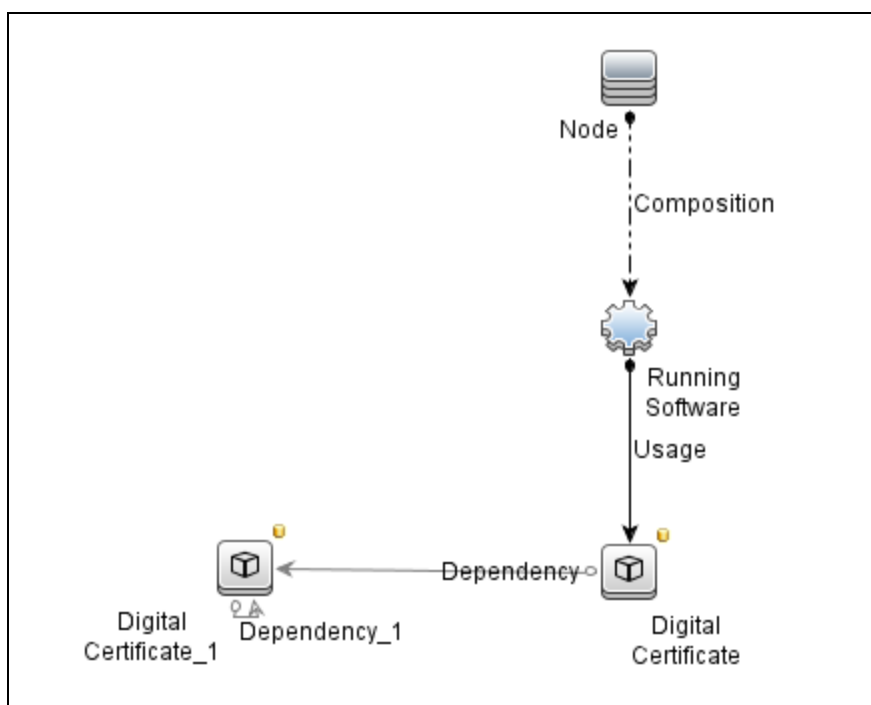
## 概要

このディスカバリでは、HTTPS プロトコルを使用しているとしてレポートされた RunningSoftware CI で使用されている SSL 証明書を検出できます。

## サポートされるバージョン

このディスカバリは、公開鍵基盤および権限管理基盤用の X.509 標準に従っている SSL 証明書の検出のみをサポートします。

## トポロジ



## ディスカバリ・メカニズム

ディスカバリ・フローには、次の手順が含まれます。

1. ホストに接続します。
2. SSL ハンドシェイクを実行します。
3. 証明書情報を取得します。

## SSL Certificates の検出方法

### 1. 前提条件

ディスカバリを正常に実行するには、**IpServiceName** または **ServiceNames** の属性値が **https** になっている **IpServiceEndpoint** CI が存在する必要があります。

### 2. ディスカバリの実行

- 次のジョブをクイック・モードで実行します。

**注:** クイック・モードでは、portNumberToPortName.xml に記述された既知のポートのみが検出されます。より完全なディスカバリを実行することもできます(以下の手順を参照)。このディスカバリでは、アプリケーション関連のジョブによって検出された標準設定でないポートが検索されます。

- i. **Range IPs by ICMP** ジョブ。ターゲット IP を検出します。
- ii. **Web Server Detection using TCP Ports** ジョブまたは **TCP Ports** ジョブ。
- iii. **SSL Certificate Discovery by HTTPS** ジョブ。
- **任意指定:** より完全なディスカバリを行う場合は、次のジョブを実行します。
  - i. **Host Connection by Shell** ジョブ: ターゲット・ホストと、ターゲット・ホストへのシェル接続を検出します。
  - ii. **Host Applications by Shell** ジョブ: ターゲット・ホストのアプリケーションを検出します。
  - iii. **任意指定:** (検出する必要があるアプリケーションに基づいて)次のいずれかのジョブを実行します。
    - **Apache Tomcat by Shell**
    - **JEE Glassfish by Shell**

- JEE JBoss by Shell
- JEE Weblogic by Shell
- JEE WebSphere by Shell
- Web Server by Shell
- IIS Applications by NTCMD or UDA

iv. **SSL Certificate Discovery by HTTPS** ジョブを実行します。

ジョブ実行の詳細については、『HP Universal CMDB Data Flow Management Guide』の「Module/Job-Based Discovery」を参照してください。

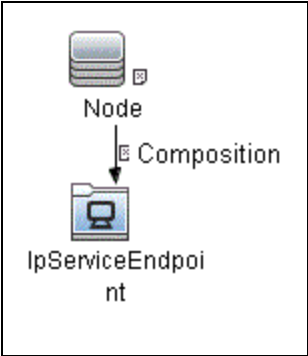
# SSL Certificate Discovery by HTTPS ジョブ

本項ではこのジョブに関する詳細を説明します。

## アダプタ

このジョブでは、**SSL Certificate Discovery** アダプタを使用します。

## トリガ TQL



ノード名	条件
Node	なし
IpServiceEndpoint	IpServiceName 等しい https OR ServiceNames 含む https

**パラメータ**

なし

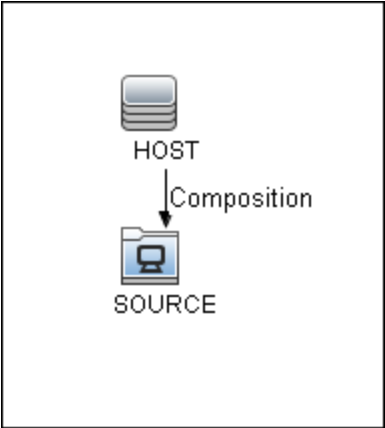
# SSL Certificate Discovery アダプタ

本項ではこのアダプタに関する詳細を説明します。

## 入力 CIT

IpServiceEndpoint

## 入力 TQL クエリ



## トリガ CI データ

名前	値
host_id	\${HOST.root_id}
https_port	\${SOURCE.network_port_number}
ip	\${SOURCE.bound_to_ip_address}

## 使用スクリプト

- ssl\_cert.py
- ssl\_cert\_discoverer.py
- ssl\_cert\_discovery\_by\_https.py
- distinguished\_name.py

## 検出された CIT

- Dependency
- Digital Certificate
- IpServiceEndpoint
- RunningSoftware
- Usage

## パラメータ

なし

## 追加情報

次のジョブは、ポートの名前を検出します。このポートが https ポートの場合、SSL Certificate Discovery のトリガに使用できます。

- IIS Discovery by Shell
- Apache TomCat Discovery by Shell
- Apache Discovery by Shell
- GlassFish ディスカバリ
- JBoss ディスカバリ
- Weblogic ディスカバリ
- WebSphere ディスカバリ
- SAP ディスカバリ

## トラブルシューティングおよび制限事項

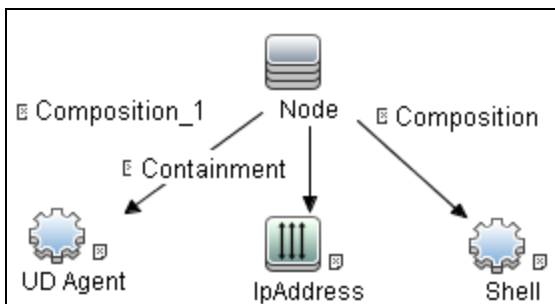
ブラウザが自動的に信頼する証明書については、このジョブでその認証局を取得することはできません。この場合、一部のルート証明書がUCMDBに表示されません。

## 第19部: Tools and Samples > UD Agent Management

## 第81章: Install UD Agent ジョブ

本項では Install UD Agent ジョブに関する詳細を説明します。

### トリガ・クエリ



### ジョブ・パラメータ

標準設定では、ジョブ・パラメータはアダプタ・パラメータを上書きしません。

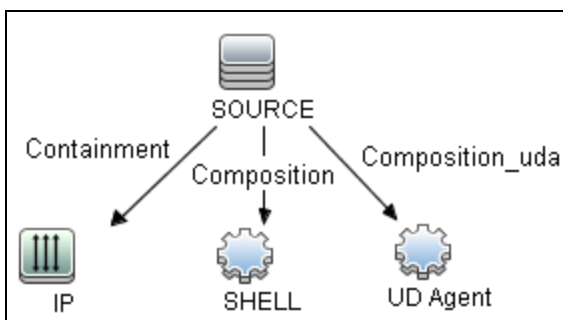
## アダプタ情報

このジョブでは、**InstallUDAgent** アダプタを使用します。

### 入力 CIT

Node

### 入力クエリ





## トリガ CI データ

名前	値
ProtocolList	\${SHELL.root_class}
codepage	\${SHELL.codepage:NA}
connected_os_credentials_id	\${SHELL.connected_os_credentials_id:NA}
credentialsId	\${SHELL.credentials_id}
hostId	\${SOURCE.root_id}
ipTaggingList	\${IP.ip_lease_time:NA}
ip_address	\${SHELL.application_ip}
macList	\${SHELL.arp_mac:NA}
nodeIpList	\${IP.name:NA}
nodeMacList	\${IP.arp_mac:NA}
shellId	\${SHELL.root_id}

## 検出された CIT

- Composition
- Node
- UDA

## アダプタ・パラメータ

名前	標準設定値	説明
CallhomeFrequency	3	このパラメータは、コールホーム要求の頻度を日単位で指定する。
EnableSoftwareUtilization	false	このフラグは、ソフトウェア使用率を有効にするかどうかを決定する。
PrimaryCallhomeProbeAddress		このパラメータは、プライマリのコールホーム・プローブ・アドレスを定義する。次のいずれかの形式を使用する。

名前	標準設定値	説明
		<ul style="list-style-type: none"> <li>IPAddress</li> <li>HostName</li> <li>HostNameOrIPv4Address:1977</li> <li>[IPv6Address]:1977</li> </ul> <p>例 :</p> <ul style="list-style-type: none"> <li>10.11.12.13</li> <li>probehost</li> <li>probehost:1977</li> <li>[2010:836b:4179::836b:4179]:1977</li> </ul>
RunUDAgentUnderRootAccount	true	このパラメータは、UD エージェントを Unix マシンのルート・アカウントで実行するかどうかを決定する。このパラメータが false の場合、UD エージェントは、インストールを行っているユーザ・アカウントで実行される。
SecondaryCallhomeProbeAddresses		<p>このパラメータは、セカンダリのコールホーム・プローブ・アドレスを指定する。</p> <p>PrimaryCallhomeProbeAddress の欄で示されている形式のいずれかを使用する。</p>
SoftwareUtilizationPeriod	365	ソフトウェア使用率データには、指定された期間におけるアプリケーションの使用日数が(割合として)示される。
UdAgentInstallCredentialId		このパラメータは、UD エージェントのインストールで使用する UD エージェントの資格情報 ID を指定する。このパラメータが空の場合、インストール・プロセスですべての ID が試行される。

## グローバル構成ファイル

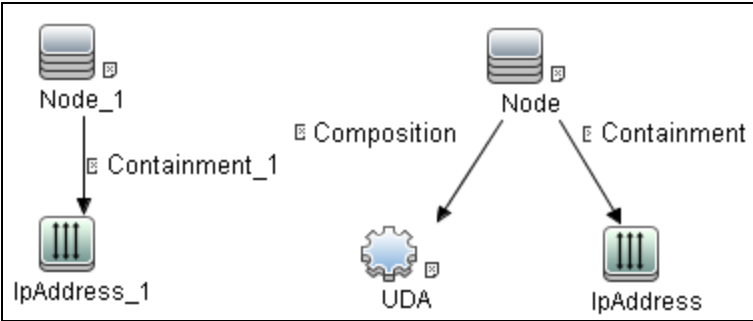
AgentsConfigurationByPlatform.xml

AgentsLifeCycleSettings.xml

# 第82章: Migrate DDMI Agent ジョブ

本項では Migrate DDMI Agent ジョブに関する詳細を説明します。

## トリガ・クエリ



## ジョブ・パラメータ

標準設定では、ジョブ・パラメータはアダプタ・パラメータを上書きしません。

## アダプタ情報

このジョブでは、**MigrateDDMItoUDAgentAdapter** アダプタを使用します。

## 入力 CIT

IpAddress

## 入力クエリ

なし

## トリガ CI データ

名前	値
architecture	\${UDA.architecture:NA}
codepage	\${UDA.codepage:NA}

名前	値
hostId	\${SOURCE.root_id}
ip_address	\${SOURCE.name}

## 検出された CIT

- Composition
- Node
- UDA

## アダプタ・パラメータ

名前	標準設定値	説明
BandwidthLimit	0	このパラメータは、アップロードまたはダウンロード時に適用する帯域幅制限を kb/s 単位で指定する。
CallhomeFrequency	3	このパラメータは、コールホーム要求の頻度を日単位で指定する。
EnableSoftwareUtilization	false	このフラグは、ソフトウェア使用率を有効にするかどうかを決定する。
PrimaryCallhomeProbeAddress		<p>このパラメータは、プライマリのコールホーム・プローブ・アドレスを定義する。次のいずれかの形式を使用する。</p> <ul style="list-style-type: none"> <li>• IPAddress</li> <li>• HostName</li> <li>• HostNameOrIPv4Address:1977</li> <li>• [IPv6Address]:1977</li> </ul> <p>例 :</p> <ul style="list-style-type: none"> <li>• 10.11.12.13</li> <li>• probehost</li> <li>• probehost:1977</li> </ul>

名前	標準設定値	説明
		<ul style="list-style-type: none"> <li>[2010:836b:4179::836b:4179]:1977</li> </ul>
RunUDAgentUnderRootAccount	true	このパラメータは、UD エージェントを Unix マシンのルート・アカウントで実行するかどうかを決定する。このパラメータが false の場合、UD エージェントは、インストールを行っているユーザ・アカウントで実行される。
SecondaryCallhomeProbeAddresses		<p>このパラメータは、セカンダリのコールホーム・プローブ・アドレスを指定する。</p> <p>PrimaryCallhomeProbeAddress の欄で示されている形式のいずれかを使用する。</p>
SoftwareUtilizationPeriod	365	ソフトウェア使用率データには、指定された期間におけるアプリケーションの使用日数が(割合として)示される。
codepage	NA	このパラメータは、検出されたマシンのコードページを指定する。

## グローバル構成ファイル

AgentsConfigurationByPlatform.xml

AgentsLifeCycleSettings.xml

# 第 83 章 : UDA Status Collector ジョブ

本 項 では UDA Status Collector ジョブに 関 する 詳 細 を 説 明 し ます。

## トリガ・クエリ



## ジョブ・パラメータ

標 準 設 定 で は、 ジョブ・パラメータはアダプタ・パラメータを上 書 きし ません。

## アダプタ情報

このジョブでは、 **UDAStatusCollector** アダプタを 使 用 し ます。

## 入 力 CIT

IpAddress

## 入 力 クエリ

なし

## トリガ CI データ

名 前	値
domain_name	\$(SOURCE.routing_domain:DefaultDomain)
ip_address	\${SOURCE.name:NA}

## 使 用 スクリプト

- icmp\_utils.py
- UDAStatusCollector.py

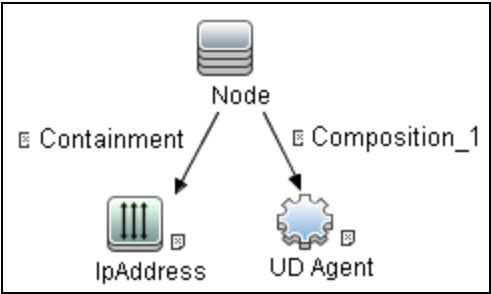
## アダプタ・パラメータ

名前	標準設定値	説明
excludePatternsList		このパラメータは、セミコロン(;)区切りのワイルドカード・パターンのリストを指定する。パターンのいずれかに一致する IP アドレスはスキップされる。パターンには、数字、ドット、*(ゼロ個以上の文字と一致)、または?(正確に 1 文字と一致)を含めることができる。
isCreateUDA	false	このフラグは、UDA CI を返すかどうかを決定する。
isIPv4PingEnabled	true	このフラグは、定義した範囲内のすべての IPv4 アドレスに対して ping を実行するかどうかを決定する。
isIPv6PingEnabled	true	このフラグは、定義した範囲内のすべての IPv6 アドレスに対して ping を実行するかどうかを決定する。
pingProtocol	1	このパラメータは、次の PING プロトコルのいずれか 1 つを指定する。 <ul style="list-style-type: none"> <li>• ICMP(1)</li> <li>• ECHO PORT(2)</li> <li>• ICMP と ECHO PORT(3)</li> </ul>
range	NA	このパラメータは、ping の対象の IP 範囲を指定する。  範囲はセミコロンで区切る。例 : 1.2.3.0-1.2.3.10;1.2.3.50-1.2.3.60
retryDiscover	2	このパラメータは、ping が失敗した場合の再試行回数を指定する。
threadPoolSize	10	このパラメータは、ポート 7 のエコーを実行するプール内のスレッドの数を指定する。
timeoutDiscover	3000	このパラメータは、ping タイムアウトをミリ秒単位で指定する。
virtualModeDiscover	false	このフラグは、仮想 IP を検出するかどうかを決定する。

# 第 84 章: Uninstall UD Agent ジョブ

本 項 では Uninstall UD Agent ジョブに関する詳 細を説 明します。

## トリガ・クエリ



## ジョブ・パラメータ

標準設定では、ジョブ・パラメータはアダプタ・パラメータを上書きしません。

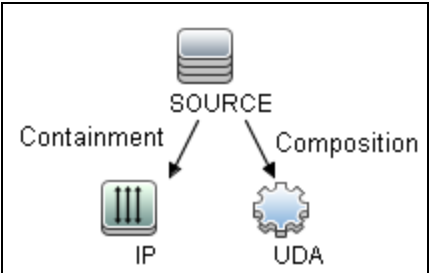
## アダプタ情報

このジョブでは、**UninstallUDAgent** アダプタを使用します。

## 入力 CIT

Node

## 入力クエリ





## トリガ CI データ

名前	値
ProtocolList	\${UDA.root_class}
agentId	\${UDA.root_id}
codepage	\${UDA.codepage:NA}
connected_os_credentials_id	\${UDA.connected_os_credentials_id:NA}
credentialsId	\${UDA.credentials_id:NA}
hostId	\${SOURCE.root_id}
ipTaggingList	\${IP.ip_lease_time:NA}
ip_address	\${UDA.application_ip}
macList	\${UDA.arp_mac:NA}
nodeGUID	\${SOURCE.ud_unique_id:NA}
nodeIpList	\${IP.name:NA}
nodeMacList	\${IP.arp_mac:NA}

## 検出された CIT

- Composition
- NTCMD
- Node
- SSH
- Telnet

## アダプタ・パラメータ

名前	標準設定値	説明
RemoveAgentData	false	このフラグは、エージェントのアンインストール後にリモート・ホストからエージェント・データを削除するか

名前	標準設定値	説明
		どうかを決定する。例 : ログ・ファイル, 使用率データ

グローバル構成ファイル

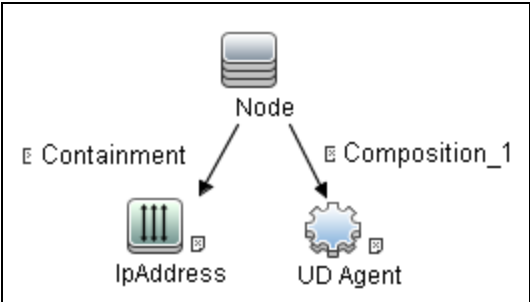
AgentsConfigurationByPlatform.xml

AgentsLifeCycleSettings.xml

# 第 85 章 : Update UD Agent ジョブ

本項では Update UD Agent ジョブに関する詳細を説明します。

## トリガ・クエリ



## ジョブ・パラメータ

標準設定では、ジョブ・パラメータはアダプタ・パラメータを上書きしません。

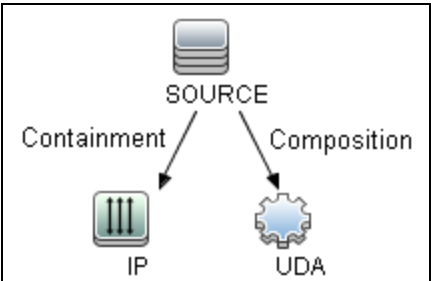
## アダプタ情報

このジョブでは、**UpdateUDAgent** アダプタを使用します。

## 入力 CIT

Node

## 入力クエリ



## トリガ CI データ

名前	値
ProtocolList	\${UDA.root_class}
agentId	\${UDA.root_id}
architecture	\${UDA.architecture:NA}
codepage	\${UDA.codepage:NA}
connected_os_credentials_id	\${UDA.connected_os_credentials_id:NA}
credentialsId	\${UDA.credentials_id:NA}
hostId	\${SOURCE.root_id}
ipTaggingList	\${IP.ip_lease_time:NA}
ip_address	\${UDA.application_ip}
macList	\${UDA.arp_mac:NA}
nodeIpList	\${IP.name:NA}
nodeMacList	\${IP.arp_mac:NA}
platform	\${UDA.platform:NA}

## 検出された CIT

- Composition
- Node
- UDA

## アダプタ・パラメータ

名前	標準設定値	説明
CallhomeFrequency	3	このパラメータは、コールホーム要求の頻度を日単位で指定する。
EnableSoftwareUtilization	false	このフラグは、ソフトウェア使用率を有効にするかどうかを決定する。

名前	標準設定値	説明
PrimaryCallhomeProbeAddress		<p>このパラメータは、プライマリのコールホーム・プローブ・アドレスを定義する。次のいずれかの形式を使用する。</p> <ul style="list-style-type: none"> <li>• IPAddress</li> <li>• HostName</li> <li>• HostNameOrIPv4Address:1977</li> <li>• [IPv6Address]:1977</li> </ul> <p>例 :</p> <ul style="list-style-type: none"> <li>• 10.11.12.13</li> <li>• probehost</li> <li>• probehost:1977</li> <li>• [2010:836b:4179::836b:4179]:1977</li> </ul>
RunUDAgentUnderRootAccount	true	<p>このパラメータは、UD エージェントを Unix マシンのルート・アカウントで実行するかどうかを決定する。このパラメータが false の場合、UD エージェントは、インストールを行っているユーザ・アカウントで実行される。</p>
SecondaryCallhomeProbeAddresses		<p>このパラメータは、セカンダリのコールホーム・プローブ・アドレスを指定する。</p> <p>PrimaryCallhomeProbeAddress の欄で示されている形式のいずれかを使用する。</p>
SoftwareUtilizationPeriod	365	<p>ソフトウェア使用率データには、指定された期間におけるアプリケーションの使用日数が(割合として)示される。</p>
UdAgentInstallCredentialId		<p>このパラメータは、UD エージェントのインストールで使用する UD エージェントの資格情報 ID を指定する。このパラメータが空の場合、インストール・プロセスですべての ID が試行される。</p>
UpgradeAgent	true	<p>このフラグは、エージェントをアップグレードするかどうかを決定する。このパラ</p>

名前	標準設定値	説明
		メータが false の場合、エージェント構成のみを更新する。

グローバル構成ファイル

AgentsConfigurationByPlatform.xml

AgentsLifeCycleSettings.xml

## ドキュメント・フィードバックの送信

このドキュメントについてコメントがある場合、電子メールで [ドキュメント・チームへの問い合わせ](#)ができます。このシステムで電子メールクライアントが設定されていれば、このリンクをクリックすることで、以下の情報が件名に記入された電子メールウィンドウが開きます。

**ディスカバリ / インテグレーション・コンテンツ・ガイド - ディスカバリ・モジュール(Universal CMDB コンテンツ・パック 17.00 (CP17))についてのフィードバック**

本文にご意見、ご感想を記入の上、[送信] をクリックしてください。

電子メールクライアントが利用できない場合は、上記の情報をコピーして Web メールクライアントの新規メッセージに貼り付け、cms-doc@hp.com 宛にお送りください。

お客様からのご意見、ご感想をお待ちしています。