

CA Catalyst

CA Configuration Automation® コネクタ ガイド
バージョン 12.8



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

CA の事前の書面による承諾を受けずに本ドキュメントの全部または一部を複写、譲渡、開示、変更、複本することはできません。本ドキュメントは、CA が知的財産権を有する機密情報です。ユーザは本ドキュメントを開示したり、
(i) 本ドキュメントが関係する CA ソフトウェアの使用について CA とユーザとの間で別途締結される契約または (ii) CA とユーザとの間で別途締結される機密保持契約により許可された目的以外に、本ドキュメントを使用することはできません。

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

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

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

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

本ドキュメントの制作者は CA です。

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

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

CA Technologies 製品リファレンス

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

- CA Configuration Automation®
- CA Catalyst
- CA IT Process Automation Manager (CA IT PAM)
- CA NSM
- CA Service Desk Manager (CA SDM)
- CA Spectrum® Service Assurance (CA Spectrum SA)

CA への連絡先

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

目次

第 1 章: 概要	7
本書の内容	7
用語	8
CA Configuration Automation コネクタ	10
統合シナリオ	11
 第 2 章: インストール	 13
サポートされるオペレーティング環境	13
インストールに関する注意事項	14
CA Configuration Automation コネクタのインストール	14
 第 3 章: 環境設定	 21
CA Configuration Automation コネクタの設定	21
 第 4 章: コネクタと CA Catalyst のインタラクション	 25
コネクタ操作から送信	25
USM データ マッピング	26
重大度マッピング	27
関係のマッピング	27
ルール コンプライアンスのマッピング	28
変更の検出のマッピング	28
バックグラウンドプロセス マッピング	28
2 項関係マッピング	30
クラスタ マッピング	32
コンピュータ システム マッピング	33
ファイル マッピング	38
ハイパーバイザ マネージャ マッピング	38
InterfaceCard マッピング	39
場所マッピング	39
メディア ドライブ マッピング	40
メモリ マッピング	41
オペレーティング システム マッピング	41
ユーザ マッピング	42

ポート マッピング	43
プロセッサ マッピング	44
プロビジョニング ソフトウェア マッピング	44
ルータ マッピング	46
サービス マッピング	46
StorageArray マッピング	47
StorageVolume マッピング	47
ディスク パーティション マッピング	48
IPConfig マッピング	48
仮想化マネージャ マッピング	51
仮想システム マッピング	51
2 項関係スコープ マッピング	51
カスタム マッピング	54
Catalyst ジョブの表示	55

第 5 章: CA Configuration Automation コネクタのアンインストール 57

第 6 章: トラブルシューティング 59

CA Catalyst コンテナと Windows 2012 Server との互換性	59
RHEL6 マシンでのグラフィカル モード インストールの失敗	60
CA Configuration Automation コネクタのインストールの確認	61
CA Configuration Automation コネクタ ステータスを確認します。	62
ccaConnector.log ファイルのエラーと例外トレース	62
CA Configuration Automation サーバ接続例外	63
データベース例外	65
JVM ポート バインディング例外	66
CA Configuration Automation コネクタから CA Catalyst サーバヘデータが正常に送信されているこ とを確認します。	67
チェックサム テーブルの照合	69
CCA から Catalyst に CI が正常にエクスポートされていない場合は、データベースをクリーン アップします。	70

第 1 章：概要

このセクションには、以下のトピックが含まれています。

[本書の内容](#) (P. 7)

[用語](#) (P. 8)

[CA Configuration Automation コネクタ](#) (P. 10)

[統合シナリオ](#) (P. 11)

本書の内容

このガイドでは、CA Configuration Automation に対して CA Catalyst コネクタをインストールおよび設定する方法について説明します。

一般的には、CA Catalyst コネクタにより、一意のより広いコンテキストで視覚化、分析および管理のために、CA CMDB、Spectrum Service Assurance (CA Spectrum SA)、および CA IT Process Automation Manager (CA IT PAM) などのコンシューミング製品に製品データが開示されます。

このガイドでは CA Configuration Automation コネクタに固有の情報が含まれています。CA Catalyst コネクタと CA Catalyst インフラストラクチャに関する一般情報、すべてのコネクタに適用される情報、およびカスタム コネクタ統合に関する情報については、CA Catalyst で提供されている「コネクタ ガイド」を参照してください。

用語

以下のリストには、ユーザが初めて **CA Catalyst** コネクタの **CA Configuration Automation** または他のコンシューム製品との統合を行っている場合に有用だと考えられる概念と用語が含まれています。

コネクタ

コネクタとは、コネクタ データをコンシュームする製品から外部製品へのリンクで、このドキュメントではドメインマネージャとして参照されています。各コネクタでは、それぞれのドメインマネージャから情報を取得し、視覚化と分析のためにコンシューミング製品にコネクタ フレームワークを通して情報を送信します。コネクタは、オブジェクト作成などのソース ドメイン マネージャでのデータに関する着信操作も実行できます。**CA Catalyst** コネクタでは、さまざまなコンシューミング製品との統合を有効にするために、統合化コネクタ フレームワークを使用しています。

USM

統合化サービス モデル (**USM**) とは、すべてのコネクタからのデータの変換先の共通のオブジェクト タイプとプロパティのスキーマです。**USM** スキーマにより、同一フォーマットの共通インターフェースでのすべてのドメイン マネージャからのデータの分析が有効になります。

構成アイテム(CI)

構成アイテム (**CI**) はドメイン マネージャによって管理された IT エレメントの表現形です。各 **CI** は、**ComputerSystem**、**Database**、**Process**、**Relationship** などの (**USM** スキーマ内で定義された) タイプに属します。サービスは **CI** の集合であり、サービス内での **CI** の関係はユーザが定義します。

コネクタは **USM** スキーマに当てはめるために、統合化製品からの管理対象オブジェクトを変換し、さらにコンシューミング製品へオブジェクトを **CI** としてインポートします。

サービス

サービスとは、複数のドメイン マネージャによって管理された構成アイテムを含むことができる個別のビジネス機能を表します。たとえば、給料支払簿サービスには **Microsoft SCOM** によって管理された **Active Directory** データベース、セキュリティ製品によって管理されたユーザストア、メインフレーム製品によって管理されたバッチ ジョブ、ネットワーク製品によって管理されたルータ、アプリケーション管理製品によって管理されたアプリケーションなどが含まれる場合があります。

アラート

アラートは、フォールト条件およびサービス低下をレポートする **CA Catalyst** のメカニズムです。インフラストラクチャアラートは、元は（**CA NSM** イベントや **CA Spectrum** アラームなどの）ドメインマネージャの 1 つによってレポートされたフォールト状態です。アラートは対応する **CI** に関連付けられ、関連付けられたアラートの重大度により **CI** の状態および最終的にはサービスへの影響が決まります。サービスアラートは、モデル化されたサービスの分析に基づいて **CA Catalyst** により生成された状態です。サービスアラートが生成されるのは、1 つまたは複数の **CI** の状態が組み合わさって、全体的品質やサービスに関連する危険水準に影響が及ぶ場合です。

コネクタからの送信操作

コネクタからの送信操作とは、ドメインマネージャから **CA Catalyst** や **CA CMDB** のようなコンシューミング製品へのデータのインポートを行うためにコネクタが呼び出す操作です。コネクタからの送信操作は、すべてのコネクタでサポートされています。

コネクタへの着信操作

コネクタへの着信操作では、コンシューミング製品でのインポートされたデータへの変更の結果として、ドメインマネージャデータストアでの変更を呼び出します。たとえば、**CA Catalyst** での **CI** 調整により、**CI** プロパティの値を変更できます。次に、着信操作をサポートするコネクタでは、そのデータが調整されたデータと一致するように、ソースドメインマネージャでの変更を実行できます。または、**CA Catalyst** により真のソースとして定義されているドメインマネージャで **CI** が削除されると、着信操作をサポートするコネクタは、その **CI** のレコードを備えた他のドメインマネージャ内の **CI** を削除できます。

CCA コネクタでは、現在、着信操作はサポートされていません。

CA Configuration Automation コネクタ

CA Configuration Automation では、ユーザの企業でのターゲットの仮想サーバと物理サーバ上のソフトウェア設定変更がモニタされます。その場合、ユーザの企業の全体にわたるサーバおよびソフトウェアの検出に、CA ネットワーク ディスカバリ ゲートウェイ (NDG) が使用されます。このようにして検出されたサーバとそれらのソフトウェア コンポーネントは個別に管理できますが、サービスに組織化して、CA Configuration Automation の UI を使用して管理することもできます。これにより、管理中の任意のコンポーネントへの設定変更が、ユーザのデータ センターでこれらのサーバとサービスをモニタする任を負うメンバにより認識されていることを保証する多くの操作を自動または手動で実行できます。

以下の 4 つの操作は、CCA コネクタを使用して、アラートを送信するように設定できます。

- サーバ変更検出
- サーバルール コンプライアンス
- サービス変更検出
- サービス ルール コンプライアンス

前述の 4 つの操作の結果に基づいてアラートを送信するように CA Configuration Automation を設定するには、CA Configuration Automation 内で以下の設定手順を実行する必要があります。

- `catalyst.events.enabled` プロパティは、「CA Configuration Automation 製品ガイド」またはオンライン ヘルプの「CA Configuration Automation プロパティの表示と編集」セクション内に述べられているように、True に設定されている必要があります。

注: `catalyst.events.enabled` プロパティが True に設定されていないと、以下が発生します。

- イベントがコネクタへ送信されない。
- CA Configuration Automation からコネクタへの CI 同期が有効ではなく、CI 追加および削除のイベントは生成されません。

- サーバまたはサービス用の管理プロファイルを作成する際に、管理プロファイルが手動で実行されるときにコネクタを使用してアラートが送られるようにするために、[管理オプション] ページで [変更検出 - アドホック] チェック ボックスをオンにします。
- サーバまたはサービス用の管理プロファイルを作成する際に、管理プロファイルがスケジュールされたジョブとして実行されるときにコネクタを使用してアラートが送信されるようにするために、[管理オプション] ページで [変更検出 - スケジュール済み] チェック ボックスをオンにします。

注: CA Configuration Automation サーバを再起動した場合、CCA コネクタをホストする Catalyst コンテナを再起動する必要があります。

統合シナリオ

CCA コネクタは CCA サーバ とのインターフェースとして働き、CA Catalyst のインフラストラクチャを活用する製品での使用のためにサーバとソフトウェア コンポーネントの構成データを開示します。CA Configuration Automation をコンシューミング製品と統合すると以下の利点があります。

環境設定管理

CCA コネクタにより、サーバ、プロパティ、および関係のための CI がコンシューミング製品に提供されます。CA Configuration Automation では、最初に一連のネットワーク スキャンの使用により、広範囲のサーバ情報が検出されます。次に、サーバ、サーバのプロパティ、ソフトウェア コンポーネントおよびそれらの関係の詳細が取得されます。この情報は、ネットワークのインベントリ、および企業内でサーバがどのように関連するかを、他の CA 製品と他社製品が把握する上で不可欠です。

CCA コネクタでは、CA Configuration Automation ルール コンプライアンス および変更の検出操作からの結果を詳述するアラートも送信されます。操作が CA Configuration Automation 上で実行される際に、これらのアラートはそれぞれの CI と関連付けられます。またコネクタによって提供される同期により、CA Configuration Automation での構成変更が、コンシューミング製品にも作成できるようになります。

ビジネス サービス管理

CA Configuration Automation サービスと CI データの CA Catalyst への開示により、別のさらに広いビジネス サービス コンテキストでのデータを評価できます。CA Catalyst では、複数の管理対象製品からのデータが収集され、インポートされたサービス、あるいはインポートされた CI からのモデルサービスをモニタできるようになります。サービス モデルには、任意の数の製品によって管理された CI が含まれる場合があります、それにより、以下のアラートが発生する際に、ビジネスへの影響についてのより明確なビューが操作に付加されます。

- CA Configuration Automation 管理によるサービスとサーバの変更の検出サマリ内の構成の逸脱。
- CA Configuration Automation 管理によるサービスとサーバのルール コンプライアンス サマリ内のコンプライアンス ルールの逸脱。

さらに、CA Configuration Automation からインポートされた CI は、サービス モデリング機能を使用して、サービスへグループ化できます。これらの操作が CA Configuration Automation 上で実行される際に、これらの CI では CA Configuration Automation ルール コンプライアンスおよび変更の検出のアラートが受信されます。

注: CA Configuration Automation ユーザはルール コンプライアンスと変更の検出の操作をスケジュールするか、あるいは手動で実行できます。これらのどちらの操作でも CA Catalyst にアラートを送信できます。これらのアラートには、コンテキスト URL 内での起動も含まれていて、スケジュールされていると、ルール コンプライアンスおよび変更の検出の操作が表示されます。これらの URL では、CA Configuration Automation の構成プロパティに基づいて、指定された時間のみの結果が表示されます。

第 2 章: インストール

この章では CCA コネクタのインストールについて説明します。

このセクションには、以下のトピックが含まれています。

[サポートされるオペレーティング環境 \(P. 13\)](#)

[インストールに関する注意事項 \(P. 14\)](#)

[CA Configuration Automation コネクタのインストール \(P. 14\)](#)

サポートされるオペレーティング環境

CCA コネクタでは、以下のリリースがサポートされています。

- CA Catalyst r3.2
- CA Configuration Automation r12.8

CCA コネクタでは、以下のオペレーティング システムでのインストールがサポートされています。

- Microsoft Windows Server 2012 Standard エディション
- Microsoft Windows Server 2008 (32 ビット版および 64 ビット版) リリース 2 Standard、Enterprise、および Datacenter (最新のサービス パックを適用)
- Red Hat Enterprise Linux 5.0 および 6.0

インストールに関する注意事項

CCA コネクタのインストールを計画する際には、以下のポイントを考慮します。

- 必要な CA Configuration Automation Java SDK は、CCA コネクタといっしょにインストールされます。これにより、CA Configuration Automation との統合が提供されます。
- CCA コネクタのインストールプログラムでは、UNC の共有はサポートされていません。
- CCA コネクタのインストールプログラムでは、以下の CCA サーバ情報の提供が求められます。
 - ホスト名または IP アドレス
 - 認証情報（ユーザ名とパスワード）
 - ポート番号

CA Configuration Automation コネクタのインストール

以下の場所に CA Configuration Automation コネクタをローカルにインストールできます。

- CCA サーバ
- Catalyst サーバ
- CCA サーバと同じドメインにある Linux または Windows のシステム

重要: コネクタをインストールする前に、コンテナ インストーラのバージョン 3.2.0.0 以降がコンピュータにインストールされることを確認します。

次の手順に従ってください:

1. コネクタ パッケージから以下のいずれかのファイルをダブルクリックします。

- Connector_CCA.exe (Windows)

- Connector_CCA.bin (Linux)

コンピュータがすべての前提条件を満たしている場合、インストーラの [開始画面] ページが表示されます。コンピュータにコンテナがインストールされない場合、[見つからない前提条件コンポーネント] ページが表示されます。

2. 必要に応じて、コンテナをインストールします。

- a. [見つからない前提条件コンポーネント] ページで、[今すぐインストール] をクリックします。

インストーラにより、以下のデフォルトの場所にコンテナが配置されます。

<インストーラの起動ディレクトリ>/*Container* フォルダ。

インストーラがデフォルトの場所でコンテナを発見できない場合、ファイル選択ダイアログ ボックスが表示されます。

- b. コンテナの `install.exe` または `install.bin` ファイルを選択し、インストールを続行します。

重要: Linux へのインストールでは、コンテナのインストーラを起動できるように、`install.bin` ファイルに実行権限があることを確認してください。

3. インストーラの [開始画面] ページで [次へ] をクリックします。

4. [使用許諾契約] ページで、許諾契約を承認し、[次へ] をクリックします。

[インストール フォルダ] ページが表示され、利用可能なコンテナと対応するノード名のリストが表示されます。

5. コンテナをクリックします。

対応するノード名を解決できない場合、Catalyst コンテナ のサーバの構成ページが表示されます。

6. コンテナ ノードの名前を入力し、[次へ] をクリックします。

選択したコンテナにコネクタが存在し、そのコネクタのバージョンがインストーラのバージョンより低い場合、プロンプトが表示され、アップグレードするように促します。そうでない場合は、バージョン競合のダイアログ ボックスが表示され、上位バージョンへのアップグレード中であることが示されます。

注: 各コンテナでは、CA Configuration Automation コネクタが 1 つだけ存在できます。

7. [アップグレード] をクリックして、アップグレード処理を確定します。

コネクタ インスタンスが選択したコンテナに存在しない場合、インストーラは新規インストールとして続行されます。Catalyst サーバの構成ページで、以下のフィールドに入力します。

Host name

Catalyst サーバ名を定義します。

HTTP Port

Catalyst サーバがリスンする HTTP ポートを定義します。

8. CCA コネクタの構成ページの以下のフィールドに入力します。

CCA Server Host Name

コネクタがアラームや更新プログラムをモニタする CCA サーバ の名前を定義します。

CCA Server Port

CCA サーバ がリスンするポートを定義します。

デフォルト : 8080

CCA Server User

CCA サーバ にアクセスできるユーザを定義します。

デフォルト : ccaadmin

CCA Server Password

指定したユーザに関連付けるパスワードを定義します。

Verify CCA Server Password

パスワードが正しく入力されたことを確認します。

CCA Notification Listener Port

製品からイベントを受信するポートを定義します。

デフォルト： 7071

HTTPS

ターゲット CCA サーバで HTTPS が有効かどうかを指定します。

X.509 Certificate Authentication

ターゲット CCA サーバでクライアント認証が有効かどうかを指定します。

Certificate Path

CCA サーバユーザと関連付ける証明書へのパスを定義します。

Certificate Password

証明書ファイルに関連付けるパスワードを定義します。

9. [次へ] をクリックします。

CCA サーバに既存の登録済みコネクタがある場合は、別のサーバを選択するようにプロンプトが表示されます。

注: CCA サーバがサポートするコネクタ インスタンスは 1 つだけです。

10. [データベース サーバ] 画面で、以下のフィールドに入力します。

注: [データベース サーバ] および [データベース構成] ページに表示されるデータベースの詳細は、[CCA Connector 構成] ページに表示される CA Configuration Automation サーバの詳細に対応している必要があります。

データベース タイプ

CCA サーバ が使用するデータベースのタイプを定義します。

サーバ名

CCA データベースが常駐するコンピュータの名前を定義します。

ポート番号

CCA データベース ホストがリスンするポートを定義します。

デフォルト： 1433

インスタンス名(オプション)

CCA データベース インスタンス名を定義します。

11. [次へ] をクリックします。

12. [データベースの構成] ページで以下のフィールドに入力します。

データベース名

CCA データベース名を定義します。

デフォルト : cca

データベース ユーザ

CCA データベース管理者ユーザの名前を定義します。

デフォルト : cca

データベース ユーザ パスワード

指定した CCA データベース管理者ユーザに関連付けるパスワードを定義します。

パスワードの再入力

パスワードが正しく入力されたことを確認します。

13. [次へ] をクリックします。

14. [変更検出アラート メトリックしきい値レベル] ページで、以下のフィールドに入力します。

Alert Metric

以下のいずれかのメトリックを指定します。メトリックとしきい値が組み合わさって、アラートの重大度レベルが決定されます。

CountChange

変更検出操作が、ソース サーバからターゲット サーバへの変更回数に関するアラート重大度に基づいていることを定義します。

CountSource

変更検出操作が、ソース サーバへの変更回数に関するアラート重大度に基づいていることを定義します。

CountTarget

変更検出操作が、ターゲット サーバへの変更回数に関するアラート重大度に基づいていることを定義します。

CountTotal

変更検出操作が、変更合計回数に関するアラート重大度に基づいていることを定義します。

情報のしきい値

指定された [変更の検出] メトリック用の変更の最小数を定義します。これはアラートに情報レベルの重大度を割り当てるためのものです。

デフォルト : 0

注: 各重大度レベルのしきい値を増加させてください。情報レベルの値を最小に設定し、マイナー レベルの値を次に小さな値に設定します。以下、同じように処理します。

Minor Threshold

指定された [変更の検出] メトリック用の変更の最小数を定義します。これはアラートにマイナー レベルの重大度を割り当てるためのものです。

デフォルト : 5

Major Threshold

指定された [変更の検出] メトリック用の変更の最小数を定義します。これはアラートにメジャー レベルの重大度を割り当てるためのものです。

デフォルト : 10

Critical Threshold

指定された [変更の検出] メトリック用の変更の最小数を定義します。これはアラートにクリティカル レベルの重大度を割り当てるためのものです。

デフォルト : 20

Fatal Threshold

指定された [変更の検出] メトリック用の変更の最小数を定義します。これはアラートに致命的レベルの重大度を割り当てるためのものです。

デフォルト : 30

15. [次へ] をクリックします。
16. [インストール サマリ] ページで選択内容を確認し、[インストール] をクリックします。
コネクタをインストールし、適切な CA Configuration Automation および CA Catalyst インスタンスと統合します。

17. [インストール完了] ページ上で [完了] をクリックします。
インストール処理により、ログ ファイルが作成されます。[インストール サマリ] ページにエラーが表示された場合は、次のファイルを見直して、インストールの問題を解決します。

```
%CATALYST_HOME%\CCA Connector Uninstall folder ¥ Logs¥  
CA_Configuration_Automation_Connector_Install%timestamp%.log
```

注: インストール完後、CCA サーバを再起動します。

第 3 章：環境設定

この章では、インストール後に CCA コネクタを設定する方法について説明します。

CA Configuration Automation コネクタの設定

インストールの後、インストール中に定義したコネクタのプロパティを変更し、さらにその他のプロパティを編集して、コネクタ動作を最適化させたり、統合された製品の変更に合わせて調節できます。

CCA コネクタを設定する方法

1. レジストリ サービスが含まれる **Catalyst** ホスト サーバにログインし、次に、以下のディレクトリに移動します。
`¥topology¥physical¥<CCA_connector_host_server>¥modules¥configuration`
2. XML エディタで `CCACConnector_<CCAServerHost>_<CCA Server port>.xml` ファイルを開き、以下の設定パラメータのうちの任意のものを変更してから、ファイルを保存します。

host

コネクタが CI とアラートを収集している CCA サーバ ホストを指定します。

port

ターゲット CA Configuration Automation ホスト上で CA Configuration Automation が実行されているポートを指定します。

username

SDK を介して CA Configuration Automation にアクセスできる有効なユーザ名を指定します。

password

指定したユーザ名に対する暗号化されたパスワードを指定します。

secure

SDK が HTTPS の安全な通信を使用するかどうかを指定します。安全な通信を有効にするには、値を `true` に設定します。

retry_count

接続失敗があった場合に再試行する回数を指定します。

postfix

SDK webservice にアクセス可能なサーバ上のエンドポイントを指定します。

デフォルト： ¥cacca¥services¥SDKService

cd_alert_metric

以下のいずれかのメトリックを指定します。これにより、変更検出アラートの重大度レベルが決まります。

CountChange

ソースからターゲットで発生した変更の数を指定します。

CountSource

ソースでのみ発生した変更の数を指定します。

CountTarget

ターゲットでのみ発生した変更の数を指定します。

CountTotal

変更検出操作によって判別された変更の総数を指定します。

cd_fatal_threshold

指定された［変更の検出］メトリック用の変更の最小数を指定します。これはアラートに致命的レベルの重大度を割り当てるためのものです。

cd_critical_threshold

指定された［変更の検出］メトリック用の変更の最小数を指定します。これはアラートにクリティカルレベルの重大度を割り当てるためのものです。

cd_major_threshold

指定された［変更の検出］メトリック用の変更の最小数を指定します。これは、アラートにメジャーレベルの重大度を割り当てるためのものです。

cd_minor_threshold

指定された [変更の検出] メトリック用の変更の最小数を指定します。これはアラートにマイナー レベルの重大度を割り当てるためのものです。

cd_information_threshold

指定された [変更の検出] メトリック用の変更の最小数を指定します。これはアラートに情報レベルの重大度を割り当てるためのものです。

notification_listen_port

CA Configuration Automation からイベントを受信するポートを指定します。

delete_thread_interval

タイマ スレッドの間隔を指定します。 イベントがオンにされていない場合、CA Configuration Automation での CI の削除は、CA Catalyst と同期される必要があります。 このオプションはデフォルトでは存在しません。必要に応じて、手動で追加する必要があります。

デフォルト（オプションが追加された場合）：15 分間

client_auth_cert_file

CCA サーバでクライアント認証が有効な場合、CCA サーバに接続するように設定したユーザの証明書のパスを指定します。これは、この手順の `username` パラメータにすでに設定したユーザの証明書である必要があります。

client_auth_cert_password

`client_auth_cert_file` パラメータで設定された証明書ファイルのパスワードを指定します。

client_auth_keystore_type

`client_auth_cert_file` に設定された証明書ファイル用のキーストアタイプを指定します。デフォルトは、PKCS12（CCA サーバからダウンロードされる証明書の形式）です。これ以外にサポートされるタイプは Java キーストア タイプを表す JKS のみです。

プロパティの変更内容が保存されます。

重要： Launch in Context 詳細のパラメータ内の値を編集しない場合は、`CCAConnector_<CCAServerHost>_<CCA Server port>.xml` ファイル内のほかのプロパティも変更しません（手順 3 を参照してください）。

3. (オプション) ファイル内のこれらの値のいずれかを変更した場合は、**Launch in Context** 詳細のパラメータ内のホスト、ポートまたはその両方の値を変更します。

注: **Launch in Context** 詳細の機能に関連付けられた他のどの値も変更しません。

プロパティの変更内容が保存されます。

4. コネクタを停止し、再起動します。

コネクタが再起動します。

重要: コネクタでは、開始操作と停止操作を短い間隔で繰り返さないようにしてください。停止および開始の都度、対応するコマンドがコネクタに送信されます。開始操作と停止操作を短い間隔で繰り返すと、これらのコマンドがコネクタ上にキューイングされ、コネクタでは、キュー内のコマンドがすべて処理されるまで開始と停止が繰り返されます。

第 4 章: コネクタと CA Catalyst のインタラクション

この章では、CCA コネクタがどのように CA Catalyst と対話するか、また、CA Configuration Automation エンティティがどのように USM スキーマにマッピングされるかについて説明しています。

注: CCA コネクタまたは CCA サーバが再起動されると、さまざまなコンプライアンス CI のコンプライアンス ステータスが [不明] にリセットされます。このステータスは、CCA サーバでの次の手動あるいはスケジュールによるコンプライアンス操作（つまり、変更の検出あるいはルール コンプライアンス）が終わると、[準拠] か [非準拠] に更新されます。

このセクションには、以下のトピックが含まれています。

[コネクタ操作から送信](#) (P. 25)

[USM データ マッピング](#) (P. 26)

[Catalyst ジョブの表示](#) (P. 55)

コネクタ操作から送信

CCA コネクタはコネクタ操作からの送信を呼び出して、コンシューミング製品に以下の CA Configuration Automation データをインポートすることができます。

自動的な CI、サービス、および関係の同期

CA Configuration Automation CI およびサービス定義をインポートして、継続的に CA Configuration Automation 内の CI、サービスおよび関係に対する更新に同期します。

CI と関係の更新

CA Configuration Automation の追加および削除をモニタするために CcaSiloConnector クラス ライブラリを使用します。

アラート

CA Configuration Automation アラートに対する追加および変更をモニタするために `CcaSiloConnector` クラス ライブラリを使用します。具体的には、コネクタは `Operations Manager Connector Framework` を使用して、CA Configuration Automation と CA Catalyst 間でアラート データを同期します。

CI のタイプおよびクラス

示されたクラスとして以下のオブジェクト タイプをインポートします。

- サービス
- サーバ
- コンポーネント
- コンプライアンス ステータス (ルール コンプライアンス操作と変更の検出操作の結果)
- 関係 (CA ネットワーク ディスカバリ ゲートウェイ (NDG) および仮想化関係からの階層関係)

USM データ マッピング

コネクタは、サービスと CI をインポートする際に、クラス、プロパティ、関係および重大度を正規化して、USM スキーマに当てはめます。このセクションでは、CA Configuration Automation クラス、重大度、および関係とインポートの後のそれらの USM マッピングをリスト表示します。

注: CI プロパティ マッピングの詳細については、レジストリ サービス内の `¥topology¥physical¥<CCA_connector_host_server>¥modules¥policy` にある CCA コネクタ ポリシー ファイルを参照してください。

重大度マッピング

以下のテーブルでは、CCA コネクタがどのように CA Configuration Automation アラート重大度を USM 重大度にマッピングするかを説明しています。

CA Configuration Automation 重大度	USM 重大度
情報	マイナ
警告 :	メジャー
エラー	クリティカル

関係のマッピング

以下のテーブルでは、CCA コネクタが、どのように CA Configuration Automation コンポーネントのプロパティ値を USM コンポーネント関係にマッピングするかを説明しています。

CA Configuration Automation コンポーネント	USM コンポーネント関係
comp_uuid	Source_uuid
Srvr_uuid	Target_uuid
relationship	HasAccessTo/HasRequirementFor

以下のテーブルでは、CCA コネクタが USM 仮想化関係にどのように CA Configuration Automation コンポーネント プロパティ値をマップするかを説明しています。

CA Configuration Automation コンポーネント	USM 仮想化関係
source_uuid (サーバ)	Source_uuid
target_uuid (サーバ)	Target_uuid
relationship	IsManagedBy/IsHostedBy

ルール コンプライアンスのマッピング

以下のテーブルでは、CCA コネクタが USM ComplianceStatus にどのように CA Configuration Automation ルール コンプライアンスの結果をマッピングするかを説明しています。

CA Configuration Automation RuleComplianceSummary	USM ComplianceStatus
ComplianceStatus と連結されたサーバまたはサービス UUID	MdrElementId
CA Configuration Automation をコンテキスト内で起動してサマリを表示するための URL の更新	UrlParams

変更の検出のマッピング

以下のテーブルでは、CCA コネクタがどのように CA Configuration Automation 変更の検出の結果を USM ComplianceStatus にマッピングするかを説明しています。

CA Configuration Automation ChangeDetectionSummary	USM ComplianceStatus
ComplianceStatus と連結されたサーバまたはサービス UUID	MdrElementId
CA Configuration Automation をコンテキスト内で起動してサマリを表示するための URL の更新	UrlParams

バックグラウンド プロセス マッピング

以下の表は、CCA コネクタが CA Configuration Automation バックグラウンドプロセスを USM プロセスにどのようにマップするかを表しています。

サービス	[acm_os_svc]	プロパティ名
サービス表示名	display_name	NamedAliases
サービス キー名	svc_name	ProcessName/ProductName
サービス ログオン名	logon_as	usm-core2: LogOnAs
サービス パス	path	usm-core2: ExecutablePath

サービス	[acm_os_svc]	プロパティ名
サービス 起動	startup	StartupType
オープン ポート	[acm_open_ports]	プロパティ名
ポート	port	AccessedViaTcpPort/usm-core2:AccessedViaUdpPort
プロトコル	protocol	TCP/UDP
プロセス名	name	ProcessName/ProductName
プロセス パス	path	usm-core2:ExecutablePath
通信関係	[acm_comm_relshps]	プロパティ名
アプリケーション名 1	テーブル内に列はありませんが、ビューへのアクセスの結果として戻りました。	ProductName
アプリケーション名 2	テーブル内に列はありませんが、ビューへのアクセスの結果として戻りました。	ProductName
Netstat 通信関係	[acm_netstat_relshps]	プロパティ名
プロセス名 1	process_name_1	ProcessName
実行可能パス 1	exec_path_1	usm-core2:ExecutablePath
プロセス名 2	process_name_2	ProcessName
実行可能パス 2	exec_path_2	usm-core2:ExecutablePath

2 項関係マッピング

以下の表は、CCA コネクタが CA Configuration Automation 2 項関係を USM 関係にどのようにマップするかを表しています。

BP からのネットワーク詳細 (Windows)	[acm_param]	プロパティ名
クラスタ ステータス	"name" = "Cluster Status" の場合、 "value" 列に注意してください	usm-core2:MemberStatus
クラスタ化済み	"name" = "Clustered" の場合、 "value" 列に注意してください	Semantic= "HasMember" ComputerSystem
通信関係	[acm_comm_relshps]	プロパティ名
IPv4 アドレス 1	ipv4_addr_1	usm-core2:SourceIPv4 Address
IPv6 アドレス 1	ipv6_addr_1	usm-core2:SourceIPv6 Address
ポート 1	port_1	usm-core2:SourceTransport LayerPort
プロトコル	protocol	usm-core2:TransportLayer
関係タイプ	relshp_typ	Semantic: IsConnectedTo,HasAccess To, IsConnectedTo,HasAccess To
IPv4 アドレス 2	ipv4_addr_2	usm-core2:TargetIPv4 Address
IPv6 アドレス 2	ipv6_addr_2	usm-core2:TargetIPv6 Address
ポート 2	port_2	usm-core2:TargetTransportLayer Port

CTA 通信関係	[acm_ta_relshps]	プロパティ名
属性名	attr_name	usm-core2: ExtensionNameValuePairs
属性値	attr_value	usm-core2: ExtensionNameValuePairs
関係トラフィックのサマリ	[acm_traffic_summary]	プロパティ名
開始時間	start_tm	usm-core2: ExtensionNameValuePairs
停止時間	stop_tm	usm-core2: ExtensionNameValuePairs
パケット数	packet_cnt	usm-core2: ExtensionNameValuePairs
通信関係 - 設定	[acm_relshp]	プロパティ名
コンポーネント UUID	comp_uuid	SourceMdrElementID
関係タイプ	relshp_typ	Context = HasAccessTo/ HasRequirementFor
包含関係	[acm_srvr_inst/acm_comp]	プロパティ名
親	acm_svc.svc_name, acm_srvr_inst.srvr_name, acm_bp.name	SourceMdrElementID
子	acm_srvr_inst.srvr_name, acm_bp.name	TargetMdrElementID
関係タイプ	relshp_typ	Semantic:IsComposedOf
仮想関係	[acm_srvr_relshps]	プロパティ名
親	parent_srvr_uuid	SourceMdrElementID
子	child_srvr_uuid	TargetMdrElementID

仮想関係	[acm_svr_relshps]	プロパティ名
関係タイプ	relshp_typ	Context = IsManagerFor/IsHostFor

ストレージ関係	[acm_svr_stor_rel]	プロパティ名
親	lun_name	SourceMdrElementID
子	svr_name	TargetMdrElementID
関係タイプ	relshp_typ	Semantic:IsConnectedTo

ストレージ関係	[acm_svr_stor_rel]	プロパティ名
親	disk_name	SourceMdrElementID
子	lun_name	TargetMdrElementID
関係タイプ	relshp_typ	Semantic:IsComposedOf

クラスタ マッピング

以下の表は、CCA コネクタが CA Configuration Automation クラスタを USM にどのようにマップするかを表しています。

ブループリントからのネットワーク詳細 (Windows)	[acm_param]	プロパティ名
クラスタ名	"name" = "Cluster Name" の場合、"value" 列に注意してください。	GroupName

クラスタ	[acm_cluster]	プロパティ名
クラスタ名	cluster_name	GroupName/PrimaryDns Name
クラスタ IPv4 アドレス	ipv4_addr	PrimaryIPV4Address
クラスタ IPv6 アドレス	ipv6_addr	PrimaryIPV6Address

コンピュータ システム マッピング

以下の表は、CCA コネクタが CA Configuration Automation コンピュータ システムを USM にどのようにマップするかを表しています。

サーバ	[acm_srvr]	プロパティ名
サーバ UUID	srvr_uuid	MdrElementID
ビジネス プロセス	business_process	usm-core2: BusinessRelevance
変更時刻	modification_tm	LastModTimeStamp
ステータス	status	AdministrativeStatus (New, managed, unmanged)
コンテキスト内起動 URL	srvr_name	UrlParams
サーバ	[acm_srvr_inst]	プロパティ名
ホスト名	srvr_name	PrimaryDnsName
製造元	manufacturer	Vendor
モデル	mdl	Model
シリアル番号	serial_number	PhysSerialNumber
Windows ドメイン名	windows_domain_name	usm-core2:PrimaryDnsDomain
NIS ドメイン名	nis_domain_name	usm-core2:PrimaryNisDomain
BIOS 名	bios_name	usm-core2:BiosName
BIOS 製造元	bios_manufacturer	usm-core2:BiosVendor
BIOS シリアル番号	bios_serial_number	BiosSystemID
SNMP システム名	snmp_sys_name	SysName
SNMP の説明	snmp_descr	Description
IPv4 アドレス	ipv4_addr	PrimaryIPV4Address
OS 名	os_name	usm-core2:OSName
OS バージョン	os_ver	PrimaryOSVersion
MAC アドレス	mac_addr	PrimaryMacAddress

サーバ	[acm_srvr_inst]	プロパティ名
BIOS ファームウェアバージョン	bios_firmware_ver	usm-core2:BiosVersion
アーキテクチャ	architecture	ComputerSystem ProcessorType
CPU 速度	cpu_speed	ProcessorSpeedInGHz
CPU 数	number_of_cpus	NumberOfCores
作成者	created_by	CreationUserName
作成時間	creation_tm	CreationTimestamp
OS ファミリ	os_family	PrimaryOSType
リフレッシュ時間	rfrsh_tm	LastModTimestamp
検出時間	discvd_tm	usm-core2: DiscoveryTimestamp
IPv6 アドレス	ipv6_addr	PrimaryIPv6Address
元のホスト名	orig_host_name	NamedAliases

ブループリントからのネットワーク詳細(すべてのオペレーティングシステム)	[acm_param]	プロパティ名
ドメイン	"name" = "Domain" の場合、 "value" 列に注意してください	usm-core2:PrimaryDns Domain
IPv6 アドレス	"name" = "IPv6 Address" の場合、 "value" 列に注意してください	PrimaryIPv6Address
ネットマスク	"name" = "Netmask" の場合、 "value" 列に注意してください	usm-core2:IPv4NetMask or IPV6NetMask
NIC の数	"name" = "Number of NICs" の場合、 "value" 列に注意してください	usm-core2:NumberOf InterfaceCards
プライマリ IP アドレス	"name" = "Primary IP Address" の場合、 "value" 列に注意してください	PrimaryIPv4Address

ブループリントからのネットワーク詳細 (Windows)	[acm_param]	プロパティ名
HBA 数	"name" = "Number of HBAs" の場合、"value" 列に注意してください	usm-core2: NumberOfHbas
プライマリ HBA ワールドワイド名	"name" = "Primary HBA Worldwide Name" の場合、"value" 列に注意してください	usm-core2: PrimaryWWName
ブループリントからのネットワーク詳細 (Linux)	[acm_param]	プロパティ名
HBA 数	"name" = "Number of HBAs" の場合、"value" 列に注意してください	usm-core2: NumberOfHbas
プライマリ HBA ワールドワイド名	"name" = "Primary HBA Worldwide Name" の場合、"value" 列に注意してください	usm-core2: PrimaryWWName
ブループリントからのネットワーク詳細 (HP-UX)	[acm_param]	プロパティ名
HBA 数	"name" = "Number of HBAs" の場合、"value" 列に注意してください	usm-core2: NumberOfHbas
プライマリ HBA ワールドワイド名	"name" = "Primary HBA Worldwide Name" の場合、"value" 列に注意してください	usm-core2: PrimaryWWName
ブループリントからのネットワーク詳細 (AIX)	[acm_param]	プロパティ名
HBA 数	"name" = "Number of HBAs" の場合、"value" 列に注意してください	usm-core2: NumberOfHbas

ブループリントからのネットワーク詳細 (AIX)	[acm_param]	プロパティ名
プライマリ HBA ワールドワイド名	"name" = "Primary HBA Worldwide Name" の場合、 "value" 列に注意してください	usm-core2: PrimaryWWName
ブループリントからのハードウェア詳細 (すべてのオペレーティングシステム)	[acm_param]	プロパティ名
BIOS ファームウェア日付	"name" = "BIOS-Firmware Date" の場合、 "value" 列に注意してください	usm-core2: BiosDate
BIOS ファームウェアバージョン	"name" = "BIOS-Firmware version" の場合、 "value" 列に注意してください	usm-core2: BiosVersion
CPU 数	"name" = "CPU Quantity" の場合、 "value" 列に注意してください	NumberOfCores
CPU 速度	"name" = "cpu speed" の場合、 "value" 列に注意してください	ProcessorSpeedInGHz
CPU タイプ	"name" = "CPU Type" の場合、 "value" 列に注意してください	ProcessorType
ホスト名	"name" = "host name" の場合、 "value" 列に注意してください	PrimaryDnsName
製造元	"name" = "manufacturer" の場合、 "value" 列に注意してください	Vendor
モデル	"name" = "Model" の場合、 "value" 列に注意してください	Model
パッチ レベル	"name" = "Patch Level" の場合、 "value" 列に注意してください	usm-core2: OSPatchLevel
物理メモリ	"name" = "physical memory" の場合、 "value" 列に注意してください	MemoryInGB

ブループリントからのハードウェア詳細(すべてのオペレーティングシステム)		
	[acm_param]	プロパティ名
シリアル番号	"name" = "serial number" の場合、"value" 列に注意してください	PhysSerialNumber
システム GUID	"name" = "System GUID" の場合、"value" 列に注意してください	ComputerSystem BiosSystemID
合計ディスク サイズ	"name" = "Total Disk size" の場合、"value" 列に注意してください	StorageInGB
ブループリントからのストレージ詳細(すべてのオペレーティングシステム)		
	[acm_param]	プロパティ名
論理ドライブ数	"name" = "Number of Logical Drives" の場合、"value" 列に注意してください	usm-core2:NumberOfDiskPartitions
物理ドライブ数	"name" = "Number of Physical Drives" の場合、"value" 列に注意してください	usm-core2:NumberOfPhysicalDrives
合計ディスク サイズ	"name" = "Total Disk size" の場合、"value" 列に注意してください	StorageInGB
メモリ		
	[acm_srvr_inst]	プロパティ名
メモリ スロットの合計	mem_tot_slots	usm-core2:NumberOfMemorySlots
サーバのファイバチャネル ワールドワイドノード名		
	acm_srvr_fc_wwnn	プロパティ名
ワールドワイドノード名	wwnn	usm-core2:PrimaryWWName

ファイル マッピング

以下の表は、CCA コネクタが CA Configuration Automation ファイル システムを USM にどのようにマップするかを表しています。

ブループリントからのストレージ 詳細(すべてのオペレーティング システム)	[acm_cfg_param]	プロパティ名
サイズ (論理ドライブ当たり)	"name" = "Size " の場合、"value" 列に注意してください	usm-core2:MaxSizeInMB
ファイル システム	[acm_file_sys]	プロパティ名
ファイル システム名	name	FilePathUrl,[FileType="Volume"]
ファイルシステム サイズ	file_sys_size	usm-core2:MaxSizeInMB
ファイルシステム マウント場 所	mount_location	usm-core2:MountedFile システム

ハイパーバイザ マネージャ マッピング

以下の表は、CCA コネクタが CA Configuration Automation ハイパーバイザ マネージャを USM にどのようにマップするかを表しています。

仮想化	[acm_srvr_inst]	プロパティ名
サーバ ビジュアル タイプを含む	ve_host_typ	usm-core2:VirtualizationEnvironment

InterfaceCard マッピング

以下の表は、CCA コネクタが CA Configuration Automation ストレージプロセッサを USM にどのようにマップするかを表しています。

ストレージ プロセッサ	acm_stor_processor	プロパティ名
名前	name	DeviceSysName
IP アドレス	ip_addr	DeviceIPv4Address
IP アドレスの DNS 名	dns_name	DeviceDnsName

場所マッピング

以下の表は、CCA コネクタが CA Configuration Automation の場所を USM にどのようにマップするかを表しています。

サーバ	[acm_srvr_inst]	プロパティ名
SNMP の場所	snmp_location	LocationName

サーバ	[acm_srvr]	プロパティ名
location	location	LocationName

サービス	[acm_svc]	プロパティ名
location	location	LocationName

メディアドライブ マッピング

以下の表は、CCA コネクタが CA Configuration Automation メディア ドライブを USM にどのようにマップするかを表しています。

BP からのストレージ詳細(すべての OS)	[acm_cfg_param]	プロパティ名
説明 (物理ドライブ当たり)	"name" = "Description " の場合、説明 "value" 列に注意してください	
サイズ (物理ドライブ当たり)	"name" = "Size " の場合、"value" 列に注意してください	CapacityInMB
物理ディスク	[acm_physical_disk]	プロパティ名
物理ディスク インデックス	disk_ix	ContainingIndex
物理ディスク名	disk_name	ラベル
物理ディスク サイズ	disk_size	CapacityInMB
物理ディスク インターフェース タイプ	intf_typ	usm-core2: DriveInterfaceType
物理ディスク メディア タイプ	media_typ	DriveType
物理ディスク モデル	mdl	モデル
CD/DVD ドライブ	[acm_cd_dvd_drive]	プロパティ名
CD DVD ドライブの説明	descr	Description,[DriveType="OpticalDrive-DVD"]
CD DVD ドライブ デバイス ID	device_id	usm-core2: OSDriveName
CD DVD ドライブ メディアの種類	media_typ	TypeName/DriveType
テープドライブ	[acm_tape_drive]	プロパティ名
テープ ドライブ説明	descr	Description,[DriveType="TapeDrive"]

テープドライブ	[acm_tape_drive]	プロパティ名
テープ ドライブ デバイス タイプ	device_typ	TypeName/DriveType
テープ ドライブ 製造元	manufacturer	Vendor

メモリ マッピング

以下の表は、CCA コネクタが CA Configuration Automation メモリを USM にどのようにマップするかを表しています。

メモリ	[acm_srvr_inst]	プロパティ名
メモリ 容量	mem_cap	SizeInMB
メモリ タイプ	mem_typ	MemoryType/Model
メモリ 速度	mem_speed	usm-core2: SpeedInGHz
使用中メモリ スロット数	mem_slots_in_use	usm-core2: SlotsInUse
物理メモリ	physical_mem	MemoryInGB

オペレーティング システム マッピング

以下の表は、CCA コネクタが CA Configuration Automation オペレーティング システムを USM にどのようにマップするかを表しています。

ブループリントからのハードウェア詳細(すべてのオペレーティング システム)	[acm_param]	プロパティ名
仮想メモリ	"name" = "virtual memory" の場合、"value" 列に注意してください	usm-core2:VirtualMemoryInGB
オペレーティング システム	[acm_srvr_inst]	プロパティ名
OS 名	os_name	NamedAliases

オペレーティング システム	[acm_srvr_inst]	プロパティ名
OS 詳細	os_detail	説明
OS タイプ	os_typ	OSType
OS メジャー バージョン	os_ver_major	MajorVersion
OS マイナー バージョン	os_ver_minor	MinorVersion
OS ビルド バージョン	os_ver_build	BuildNumber
パッチ レベル	os_patch_level	usm-core2: OSPatchLevel
OS バージョン	os_ver	Version
OS カーネル	os_kernel	usm-core2: OSKernelVersion

メモリ	[acm_srvr_inst]	プロパティ名
仮想メモリ	virt_mem	usm-core2:VirtualMemory InGB

ユーザ マッピング

以下の表は、CCA コネクタが CA Configuration Automation ユーザ（個人）を USM にどのようにマップするかを表しています。

サーバ	[acm_srvr_inst]	プロパティ名
SNMP 連絡先	snmp_contact	UserName

サーバ	[acm_srvr]	プロパティ名
ビジネス所有者	business_owner	UserName
IT 所有者	it_owner	UserName

サービス	[acm_svc]	プロパティ名
ビジネス所有者	business_owner	UserName
IT 所有者	it_owner	UserName

ポート マッピング

以下の表は、CCA コネクタが CA Configuration Automation ポートを USM にどのようにマップするかを表しています。

ブループリントからのネットワーク詳細 (Windows)	[acm_cfg_param]	プロパティ名
説明	"name" = "Description " の場合、 "value" 列に注意してください	説明
IP アドレス	"name" = "IP Address " の場合、 "value" 列に注意してください	PrimaryIPV4Address, PrimaryIPV6Address or OtherIPAddresses
物理アドレス	"name" = "Physical address " の 場合、"value" 列に注意してく ださい	PrimaryMacAddress または OtherMacAddresses
速度	"name" = "Speed" の場合、 "value" 列に注意してください	NomSpeedInBitsPerSec
ネットワーク インターフェース カード	[acm_srvr_nic]	プロパティ名
MAC アドレス	mac_addr	PrimaryMacAddress
インターフェース インデックス	net_adapter	IfIndex
インターフェース速度	speed	NomSpeedInBitsPerSec
インターフェース二重性	duplex	usm-core2:IsFullDuplex
インターフェース ネゴシエーション	aneg	usm-core2:DuplexIs Negotiated
DNS ドメイン	dns_domain	usm-core2: PrimaryDnsDomain
ネットワーク インターフェース カード	[acm_intf_ipv4_addr]	プロパティ名
IPv4 アドレス	ipv4_addr	PrimaryIPV4Address

ネットワーク インターフェース カード	[acm_intf_ipv6_addr]	プロパティ名
IPv6 アドレス	ipv6_addr	PrimaryIPv6Address または OtherIPAddresses

プロセッサ マッピング

以下の表は、CCA コネクタが CA Configuration Automation プロセッサを USM にどのようにマップするかを表しています。

プロセッサ	[acm_srvr_inst]	プロパティ名
プロセッサ名	processor_name	NamedAliases
プロセッサ アーキテクチャ	processor_architecture	ProcessorType
プロセッサの説明	processor_descr	説明
プロセッサ製造元	processor_manufacturer	Vendor
プロセッサ最大クロック速度	processor_max_clock_speed	SpeedInGHZ
プロセッサ L2 キャッシュ サイズ	processor_l2_cache_size	usm-core2: L2CacheInMB
プロセッサ L2 キャッシュ速度	processor_l2_cache_speed	usm-core2: L2CacheSpeedInGHZ
論理プロセッサ数	processor_logical_cnt	usm-core2: NumberOfCores

プロビジョニング ソフトウェア マッピング

以下の表は、CCA コネクタが CA Configuration Automation のプロビジョニング ソフトウェアを USM にどのようにマップするかを表しています。

コンポーネント	[acm_comp]	プロパティ名
コンポーネント UUID	comp_uuid	MdrElementID
コンポーネント バージョン	comp_ver	バージョン
コンポーネント修飾子	comp_qual	SoftwarePathUrl

コンポーネント	[acm_comp]	プロパティ名
リフレッシュ時間	rfrsh_tm	usm-core2:LastRefreshTimestamp
作成時間	creation_tm	CreationTimestamp
不明	missing	AdministrativeStatus = "Missing-InSubseqDiscover"

コンポーネント	[acm_bp]	プロパティ名
コンポーネント名	name	usm-core2: BlueprintName

コンポーネント	[acm_param]	プロパティ名
製品名	"name" = "Product name" の場合、 "value" 列に注意してください	Product name
ベンダ	"name" = "Vendor" の場合、 "value" 列に注意してください	Vendor
カテゴリ	[acm_bp].cat	SoftwareCategories
コンテキスト内起動 URL	[acm_srvr].srvr_name, [acm_bp].name, [acm_comp].comp_qual	UrlParams

インストール済みアプリケーション	[acm_os_appl]	プロパティ名
アプリケーション名	name	ProductName
アプリケーション発行元	publisher	Vendor
アプリケーションバージョン	ver	バージョン
アプリケーションアーキテクチャ	アーキテクチャ	ProcessorEnvironments
アプリケーションインストール日付	install_date	usm-core2:InstallDate
アプリケーションインストール場所	install_location	SoftwarePathUrl

インストール済みアプリケーション	[acm_os_appl]	プロパティ名
パッチ	is_patch	ReleaseType

ルータ マッピング

以下の表は、CCA コネクタが CA Configuration Automation ルータを USM にどのようにマップするかを表しています。

サーバ	[acm_srvr_inst]	プロパティ名
ルータ	is_router	設定されている場合、 ComputerSystem がルータとしてマップされます

サービス マッピング

以下の表は、CCA コネクタが CA Configuration Automation サービスを USM にどのようにマップするかを表しています。

サービス	[acm_svc]	プロパティ名
サービス UUID	svc_uuid	MdrElementID
サービス名	svc_name	ServiceName
説明	descr	Description
作成者	created_by	CreationUserName
作成時間	creation_tm	CreationTimestamp
ビジネス プロセス	business_process	usm-core2:BusinessRelevance
変更時刻	modification_tm	LastModTimestamp
コンテキスト内起動 URL	svc_name	UrlParams
コンテキスト内起動ビジュアル URL	svc_name	usm-core2:ExtensionNameValue Pairs

StorageArray マッピング

以下の表は、CCA コネクタが CA Configuration Automation ストレージ システムを USM にどのようにマップするかを表しています。

ストレージ システム	acm_stor	プロパティ名
名前	name	NamedAliases
シリアル番号	serial_number	PhysSerialNumber
製造元	manufacturer	Vendor
モデル/タイプ	mdl_typ	Model
ストレージ容量	storage_cap	HardDriveCapacityInGB
ストレージのファイバ チャンネル ワールドワイド名		
ワールドワイド ノード名	wwnn	usm-core2:PrimaryWWName
ワールドワイド ポート名	wwpn	usm-core2:OtherWWNames

StorageVolume マッピング

以下の表は、CCA コネクタが CA Configuration Automation ストレージ論理ユニット番号を USM にどのようにマップするかを表しています。

ストレージ LUN	acm_stor_lun_ex	プロパティ名
Network Address Authority	net_addr_authority	PhysSerialNumber
ストレージ容量	storage_cap	PhysSerialNumber
名前	name	Label

ディスク パーティション マッピング

以下の表は、CCA コネクタが CA Configuration Automation ディスク パーティションを USM にどのようにマップするかを表しています。

論理パーティション	[acm_disk_partition]	プロパティ名
論理パーティション名	partition_name	usm-core2:NamedAliases
ドライブ指定	drive_designation	usm-core2:OSDriveName
論理パーティション インデックス	partition_ix	usm-core2:ContainingIndex
論理パーティション ファイルシステム	filesystem	usm-core2:PartitionType
論理パーティションがブート可能	is_boot_partition	usm-core2:IsBootable
論理パーティションがプライマリ	is_pri_partition	usm-core2:IsPrimary
論理パーティション サイズ	partition_size	usm-core2:CapacityInMB

IPConfig マッピング

以下の表は、CCA コネクタが CA Configuration Automation IPConfig 値を USM にどのようにマップするかを表しています。

サーバ	[acm_srvr_inst]	プロパティ名
SNMP IP 転送	snmp_ip_fwding	usm-core2:DoesIPForwarding
ドメイン サーバ IPv4 アドレス	domain_srvr_ipv4_addr	usm-core2:PrimaryDnsServer IPV4Address
ドメイン サーバ IPv6 アドレス	domain_srvr_ipv6_addr	usm-core2:PrimaryDnsServer IPV6Address
ドメイン サーバ名	domain_server_name	PrimaryDnsServer

ブループリントからのネットワーク詳細 (Windows)	[acm_cfg_param]	プロパティ名
ブロードキャスト アドレス	"name" = "Broadcast Address " の場合、"value" 列に注意してください	BroadcastIPv4Address または BroadcastIPv6Address
DNS	"name" = "DNS " の場合、"value" 列に注意してください	DnsServerAddresses (IPv4 または IPv6)
ゲートウェイ	"name" = "Gateway " の場合、"value" 列に注意してください	GatewayIPv4Address または GatewayIPv6Address
プライマリ WINS	"name" = "Primary Wins" の場合、"value" 列に注意してください	PrimaryWins IPv4Address または Primary WinsIPv6Address
セカンダリ WINS	"name" = "Secondary Wins " の場合、"value" 列に注意してください	OtherWinsAddresses
サブネット マスク	"name" = "Subnet Mask " の場合、"value" 列に注意してください	usm-core2: IPv4NetMask or IPv6NetMask
ネットワーク インターフェース カード	[acm_srvr_nic]	プロパティ名
デフォルト IPv4 ゲートウェイ	default_gateway_ipv4_addr	GatewayIPv4Address
IPv4 DHCP サーバ	dhcp_srvr_ipv4_addr	DhcpServerIPv4Address
デフォルト IPv6 ゲートウェイ	default_gateway_ipv6_addr	GatewayIPv6Address IPv6
IPv6 DHCP サーバ	dhcp_srvr_ipv6_addr	DhcpServerIPv6Address
ネットワーク インターフェース カード	[acm_intf_ipv4_addr]	プロパティ名
サブネット マスク	subnet_mask	usm-core2:IPConfig usm-core2: IPv4NetMask or IPv6NetMask
IPv4 サブネット	ipv4 subnet	usm-core2: IPv4Subnet

ネットワーク インターフェース カード	[acm_intf_ipv6_addr]	プロパティ名
IPv6 サブネット	ipv6_subnet	usm-core2: IPV6Subnet または OtherIPSubnets (IPv4/IPv6)
ネットワーク インターフェース カード	[acm_intf_dns_srvr]	プロパティ名
DNS サーバ IPv4	srvr_ipv4_addr	PrimaryDnsServer IPV4Address および OtherDnsServer Addresses
DNS サーバ IPv6	srvr_ipv6_addr	PrimaryDnsServer IPV6Address および OtherDnsServer Addresses
ネットワーク インターフェース カード	[acm_intf_wins_srvr]	プロパティ名
WINS サーバ IPv4	srvr_ipv4_addr	PrimaryWinsServer IPV4Address および OtherWinsServer Addresses
WINS サーバ IPv6	srvr_ipv6_addr	PrimaryWinsServer IPV6Address および OtherWinsServer Addresses
ストレージ iSCSI イニシエータ	acm_stor_iscsi_initr_inst	プロパティ名
イニシエータの識別子	initr_id	Extension Name Value Pairs
イニシエータの IP アドレス	[acm_stor_iscsi_initr_ip].initr_ip_addr	Static IPV4 Address

仮想化マネージャ マッピング

以下の表は、CCA コネクタが CA Configuration Automation 仮想化マネージャを USM にどのようにマップするかを表しています。

サーバ	[acm_srvr_inst]	プロパティ名
マネージャ仮想化タイプ	virt_mgmt_typ	ProcessDistinguishingID/Product Name

仮想システム マッピング

以下の表は、CCA コネクタが CA Configuration Automation 仮想システムを USM にどのようにマップするかを表しています。

仮想化	[acm_srvr_inst]	プロパティ名
ゲスト サーバ仮想化タイプ	ve_guest_typ	usm-core2:VirtualizationEnvironment
ゲスト サーバ論理名	ve_guest_logical_name	ComputerName
ゲスト サーバ起動モード	ve_guest_startup	usm-core2:IsAutomaticallyStarted

2 項関係スコープ マッピング

以下の表は、CCA コネクタが CA Configuration Automation 関係を USM にどのようにマップするかを表しています。

親	関係タイプ	子タイプ	スコープ	コメント
サービス	HasMember	ComputerSystem	サービス スコープ	
	HasDetail	ComplianceStatus	サービス スコープ	
	HasContact	Person	サービス スコープ	
	IsResidentOf	Location	サービス スコープ	
コンピュータシステム	IsManagedBy	ComputerSystem	スコープなし	仮想化関係

親	関係タイプ	子タイプ	スコープ	コメント
	HasDetail	ComplianceStatus	コンプライアンス ステータスがサービス レベルの場合はサービス スコープ、それ以外の場合はコンピュータ システム スコープ	
	HasContact	Person	スコープなし	
	HasDetail	IPConfig	コンピュータ システム スコープ	
	IsHostFor	BackgroundProcess	コンピュータ システム スコープ	
	IsConnectedTo	BackgroundProcess	スコープなし	通信関係
	IsConnectedTo	ComputerSystem	スコープなし	通信関係
	IsResidentOf	ComputerSystem	スコープなし	
プロビジョニング ソフトウェア	HasAccessTo	ComputerSystem	サービスを使用して検出されたコンポーネントの場合はサービス スコープ、それ以外の場合はスコープなし	設定関係「使用」
	HasRequirementFor	ComputerSystem	サービスを使用して検出されたコンポーネントの場合はサービス スコープ、それ以外の場合はスコープなし	設定関係「通信」
	HasAccessTo	BackgroundProcess	サービスを使用して検出されたコンポーネントの場合はサービス スコープ、それ以外の場合はスコープなし	設定関係「使用」
	HasRequirementFor	BackgroundProcess	サービスを使用して検出されたコンポーネントの場合はサービス スコープ、それ以外の場合はスコープなし	設定関係「通信」

親	関係タイプ	子タイプ	スコープ	コメント
	IsAffectedBy	ProvisionedSoftware	サービスを使用して検出されたコンポーネントの場合はサービス スコープ、それ以外の場合はスコープなし	ネストされたコンポーネント
	IsHostedBy	ComputerSystem	サービスを使用して検出されたコンポーネントの場合はサービス スコープ、それ以外の場合はスコープなし	
ポート	HasDetail	IPConfig	コンピュータ システム スコープ	
	IsPartOf	ComputerSystem	コンピュータ システム スコープ	
クラスタ	HasMember	ComputerSystem	スコープなし	
バックグラウンド プロセス	IsConnectedTo	BackgroundProcess	スコープなし	通信関係
	IsConnectedTo	ComputerSystem	スコープなし	通信関係
メディア ドライブ	IsPartOf	ComputerSystem	コンピュータ システム スコープ	
オペレーティング システム	IsAffectedBy	ComputerSystem	コンピュータ システム スコープ	
仮想化システム	IsHostedBy	ComputerSystem	コンピュータ システム スコープ	仮想化関係
メモリ	IsPartOf	ComputerSystem	コンピュータ システム スコープ	
プロセッサ	IsPartOf	ComputerSystem	コンピュータ システム スコープ	

カスタム マッピング

CCA コネクタでは、USM スキーマに当てはめる製品データをマッピングするコネクタ ポリシーが使用されます。各コネクタには、以下のディレクトリにある XML ファイル内にデフォルト ポリシーが含まれています。

`CATALYST_HOME¥resources¥Core¥Catalogpolicy`

`CATALYST_HOME`

CA Catalyst のインストール ディレクトリを定義します。

デフォルト ポリシーにより、このセクション内に含まれたマッピングがすべて確立されます。コネクタ ポリシーをカスタマイズすることにより、製品変更やカスタマイズに基づいたマッピングを追加でき、またはデフォルト ポリシーではサポートされていないエンティティをサポートできます。たとえば、デフォルト ポリシーではサポートされていない製品へのクラスを追加している場合、コネクタ ポリシーを編集することにより、これらのクラスを USM タイプにマッピングできます。

注: コネクタ ポリシーの書き込みとカスタマイズの詳細については、「*CA Catalyst コネクタ ガイド*」を参照してください。

注: 失敗した CI、およびエラーを解決する方法の詳細については、「[トラブルシューティング \(P. 59\)](#)」を参照してください。

第 5 章: CA Configuration Automation コネクタのアンインストール

不要になった場合、CCA コネクタをアンインストールできます。

次の手順に従ってください:

1. (Windows) [スタート]-[プログラム]-[CA]-[Catalyst]-[Uninstall_CA Configuration Automation Connector for <CCA Server Host>_<CCA Server Port> on the connector system] を選択します。

(Linux) Uninstall CA Configuration Automation Connector for <CCA Server Host>_<CCA Server Port> from the <CCA Connector installation directory>/Uninstall_CA Configuration Automation Connector for <CCA Server Host>_<CCA Server Port> を実行します。

[Uninstall CCA Connector] ダイアログ ボックスが表示されます。

2. [Restart Container] チェック ボックスをオンにしてコンテナ サービスを再起動し、[Uninstall] をクリックします。

CCA コネクタがアンインストールされます。[アンインストール完了] ページに、発生したエラーを含むアンインストール結果のリストが表示されます。

注: [Restart Container] チェック ボックスをオンにしていない場合は、CCA コネクタがアンインストールされた後にコンテナを再起動することをお勧めします。

3. アンインストール プログラムにより CCA コネクタ OSGi バンドル `com.ca.cca.catalyst.connector_<installed version>.jar` が <Catalyst Home>\container\system\com\ca\catalyst\3.0.0\bundles ディレクトリから削除されたことを確認します。削除されていない場合は、手動で削除します。

注: 手順 2 で [Restart Container] チェック ボックスをオンにしなかった場合は、CCA コネクタ ノード上の CA Catalyst コンテナ サービスを再起動してください。

4. [完了] をクリックします。アンインストール ログ ファイルが作成されます。アンインストール画面にエラーが表示される場合は、ログ ファイルを確認します。

```
%CATALYST_HOME%\CCA Connector Uninstall folder \Logs\
CA_Configuration_Automation_Connector_UnInstall%timestamp%.log
```

Catalyst コンテナをアンインストールする場合は、以下のタスクを実行します。

- <Catalyst Home>\Uninstall_CA Catalyst Container\Uninstall CA Catalyst Container. から CA Catalyst Container Uninstall を起動します。
- Catalyst レジストリ UI の [Actions] メニューを使用して、<connector node name> ディレクトリ全体を削除します。Catalyst レジストリ UI は、/topology/physical/ にあります。

第 6 章: トラブルシューティング

このトラブルシューティング セクションでは、CA Catalyst コンテナに関する作業で発生する一般的な問題について説明します。また、これらの問題を解決する方法も示します。CA Catalyst コンテナの診断およびトラブルシューティングの詳細については、「[CA Catalyst 実装ガイド](#)」を参照してください。

このセクションには、以下のトピックが含まれています。

[CA Catalyst コンテナと Windows 2012 Server との互換性](#) (P. 59)

[RHEL6 マシンでのグラフィカルモードインストールの失敗](#) (P. 60)

[CA Configuration Automation コネクタのインストールの確認](#) (P. 61)

[CA Configuration Automation コネクタ ステータスを確認します。](#) (P. 62)

[ccaConnector.log ファイルのエラーと例外トレース](#) (P. 62)

[CA Configuration Automation コネクタから CA Catalyst サーバへデータが正常に送信されていることを確認します。](#) (P. 67)

CA Catalyst コンテナと Windows 2012 Server との互換性

CA Catalyst コンテナは Windows 2012 と互換性がありません。CA Catalyst コンテナを Windows 2012 にインストールするには、Windows 7 または Windows 2008 互換モードを使用し、サポート対象の Windows バージョンにインストールされた CA Catalyst Server にそれをポイントする必要があります。

CA Catalyst コンテナが Windows 2012 サーバと互換性を持つようにするには、以下の手順に従います。

1. インストーラの exe ファイルを右クリックし、[プロパティ] を選択します。
2. [互換性] タブをクリックし、[互換モードでこのプログラムを実行する] オプションをクリックします。
3. インストーラがサポートするオペレーティング システムを選択します。たとえば Windows 7 を選択します。
4. 管理者特権を取得するために [管理者としてこのプログラムを実行する] オプションを選択し、[OK] をクリックします。

5. `catalyst_install.bat`、または CA セットアップ ランチャから `setup.exe` ファイルを実行します。

システムが互換モードになっているため、今回は CA Catalyst インストールまたはセットアップが実行されます。

RHEL6 マシンでのグラフィカル モード インストールの失敗

問題の状況:

CCA コネクタを RHEL6 マシンにインストールすると以下のエラーが発生します。

Graphical installers are not supported by the vm (vm ではグラフィカル インストーラはサポートされていません)

解決方法:

このエラーは以下の理由によって発生している可能性があります。

- `/usr/lib/libXtst.so` ファイルがシステムにインストールされていない。
- `$DISPLAY` 環境変数が正しく設定されていない。
- GUI インストーラを実行するために必要な X Windows ライブラリが使用可能になっていない。

CCA コネクタを RHEL マシンにインストールするには、以下のいずれかを実行してください。

- `libXtst-1.2.1-2.el6.i686.rpm` パッケージに含まれる `libXtst.so` ライブラリをインストールする。
- `$DISPLAY` 環境変数を有効なディスプレイに設定する。

注: 64 ビットの RHEL マシンにログインしている場合は、以下のコマンドを実行して不足している 32 ビットの依存コンポーネントをインストールしてください。

```
yum install xulrunner.i686
```

CA Configuration Automation コネクタのインストールの確認

症状:

CA Configuration Automation コネクタが正しくインストールされていることを確認するにはどうすればよいですか。

解決方法:

Catalyst_CCAConnector_InstallDebug.log または CatalystInstallDebug.log ファイルにエラーがないことを確認します。Windows の場合、このログ ファイルは %TEMP% ディレクトリにあります。Linux の場合、このログ ファイルは / ディレクトリにあります。ログ ファイルのどちらかにエラーがある場合は、詳細について CA Technologies のテクニカルサポートにお問い合わせください。ログ ファイルのどちらにもエラーがない場合は、以下の手順に従います。

1. 以下の URL を使用して、レジストリ UI にログインします。
`https://registryserver:port/registry/carbon/admin/login.jsp`
2. [Browse] セクション内のフォルダを展開し、以下のディレクトリを見つけます。
`¥topology¥physical¥<CA Configuration Automation Connector NODE>`
3. ディレクトリ内に `connector-modules.xml` および `startup.properties` ファイルがあることを確認します。
ファイルのいずれかがディレクトリ内に存在しない場合、CA Configuration Automation コネクタのインストールは失敗しています。
4. CA Configuration Automation コネクタをアンインストールしてから、再インストールします。

CA Configuration Automation コネクタ ステータスを確認します。

症状:

CA Configuration Automation コネクタが正常に開始されていることを確認するにはどうすればよいですか。

解決方法:

次の手順に従ってください:

1. 以下の URL を使用して、CA Catalyst 管理 UI にログインします。

`http://<CA Catalyst-Server>:port/adminui`

2. [Catalyst Node] パネル内の CA Configuration Automation コンテナ ノードをクリックして展開します。

利用可能な CA Configuration Automation コネクタと CA Configuration Automation コネクタのステータスが表示されます。ステータスが [RUNNING] としてマークされている場合、CA Configuration Automation コネクタは正常に開始されています。

注: CA Catalyst コネクタが開始された直後は、操作を実行せずに、コネクタ ノードが [RUNNING] 状態に移行するのを待機してください。

ccaConnector.log ファイルのエラーと例外トレース

ccaConnector.log ファイルには、エラーと例外トレースが表示されます。ccaConnector.log ファイルに表示される例外を確認し、例外ごとに解決を行います。Windows の場合、ログ ファイルは以下の場所にあります。

`C:\Program Files (x86)\CA\Catalyst\<ContainerID>\container\data\log`

Linux の場合、ログ ファイルは以下の場所にあります。

`/opt/CA/Catalyst/<ContainerID>/container/data/log`

ccaConnector.log ファイルに表示される例外を以下に示します。

- CA Configuration Automation サーバ接続例外
- データベース例外
- JVM ポート バインディング例外

CA Configuration Automation サーバ接続例外

症状:

CA Catalyst コネクタ アプリケーションを起動するときに CA Configuration Automation サーバ接続例外が ccaConnector.log ファイルに表示されます。

解決方法:

次の手順に従ってください:

1. CCA サーバ がオンラインで、CA Configuration Automation サーバからサーバにアクセスできることを確認します。
2. ファイアウォールが、CA Configuration Automation コネクタ サーバから必要な CCA サーバ ポートへのアクセスをブロックしていないことを確認します。

重要: CCA サーバ を再起動すると、CA Configuration Automation コネクタをホストしている CA Catalyst コンテナ サービスが再起動されます。

3. コネクタのインストール後に CCA サーバ のパスワードを変更した場合は、レジストリ UI を使用し、以下の手順で CCAConnector_<CCA Server Host>_<CCA Server Port>.xml 内の password パラメータを変更してください。
 - a. コマンドプロンプトで、以下のディレクトリを開きます。
`%CATALYST_HOME%\tools\encrypt`
注: ディレクトリに java.exe があることを確認してください。
 - b. `encrypter.bat <New Password>` コマンドを実行します。
 - c. `encrypter.bat <New Password>` コマンドから暗号化された文字列出力をコピーします。
 - d. 以下の URL を使用して、レジストリ UI にログインします。
`https://registryserver:port/registry/carbon/admin/login.jsp`
 - e. 以下のディレクトリを参照します。
`¥topology¥physical¥<CCA Connector Server>¥modules¥configuration¥CCAConnector_<CCA Server Host>_<CCA Server Port>.xml`
 - f. [Content] パネルで、[Edit as Text] をクリックします。
 - g. `encrypter.bat` コマンドからコピーした値で password プロパティの値を置換します。

- h. [Save Content] をクリックします。
- i. 解決方法を確認します。
 - CA Catalyst コンテナ サービスを再起動します。
 - CA Catalyst コンテナ アプリケーションを起動します。
 - CA Configuration Automation サーバ接続例外が ccaConnector.log ファイルに表示されていないことを確認します。

データベース例外

症状:

CA Catalyst コネクタ アプリケーションを起動するときにデータベース例外が ccaConnector.log ファイルに表示されます。

解決方法:

次の手順に従ってください:

1. データベース サーバがオンラインで、CA Configuration Automation コネクタ サーバからサーバに ping を実行できることを確認します。
2. ファイアウォールが、CA Configuration Automation コネクタ サーバから必要なデータベース ポートへのアクセスをブロックしていないことを確認します。
3. レジストリ UI から CCAConnector.xml ファイルの db.password パラメータを更新します。CA Configuration Automation コネクタのインストール後にパスワードを変更した場合は、password パラメータを更新します。パスワードを変更するには、以下の手順に従います。
 - a. コマンドプロンプトで、以下のディレクトリを開きます。
`%CATALYST_HOME%\tools\encrypt`
 注: java.exe がパスに存在していることを確認してください。
 - b. encrypter.bat <New Password> ファイルを実行します。
 - c. encrypter.bat <New Password> コマンドから暗号化された文字列出力をコピーします。
 - d. 以下の URL を使用して、レジストリ UI にログインします。
`https://registryserver:port/registry/carbon/admin/login.jsp`
 - e. 以下のディレクトリを参照します。
`¥topology¥physical¥<CCA Connector Server>¥modules¥configuration¥CCAConnector_<CCA Server Host>_<CCA Server Port>.xml`
 - f. [Content] パネルで、[Edit as Text] をクリックします。
 - g. encryptor ユーティリティからコピーした値で db.password プロパティの値を置換します。
 - h. [Save Content] をクリックします。
 - i. CA Catalyst コンテナ サービスおよび reverify を再起動します。

JVM ポート バインディング例外

症状:

CA Catalyst コネクタ アプリケーションを起動するときに JVM ポート バインディング例外が ccaConnector.log ファイルに表示されます。

解決方法:

次の手順に従ってください:

1. いずれかのプロセスが CA Configuration Automation コネクタのインストール時に notification_listen_port パラメータに使用されているポートを使用しているかどうかを確認します。
2. ポートの競合が CA Configuration Automation コネクタのインストール後に発生する場合は、レジストリ UI から CCACConnector.xml ファイルのパラメータを更新します。
 - a. 以下の URL を使用して、レジストリ UI にログインします。
`https://registryserver:port/registry/carbon/admin/login.jsp`
 - b. [Browse] セクション内のフォルダを展開し、以下のディレクトリを見つけます。
`¥topology¥physical¥<CCA Connector Server>¥modules¥configuration¥CCACConnector_<CCA Server Host>_<CCA Server Port>.xml`。 [Edit as Text] をクリックします。
 - c. CA Configuration Automation サーバのその他のポートと競合しない値で notification_listen_port の値を置換します。
 - d. [Save Content] をクリックします。
 - e. CA Catalyst コンテナ サービスを再起動します。

CA Configuration Automation コネクタから CA Catalyst サーバへデータが正常に送信されていることを確認します。

症状:

CA Configuration Automation コネクタから CA Catalyst サーバへデータが正常に送信されていることを確認するには、どうすればよいですか。

解決方法:

次の手順に従ってください:

1. 以下の URL を使用して、Catalyst USM Web ビューを開きます。

`http://catalystserver:8080/ca-rest/browse/type?mdr=all`

2. データ ソースから CA Configuration Automation を選択し、CI がページにリスト表示されていることを確認します。

CI が CA Configuration Automation データ ソースにリスト表示されない場合は、以下を確認します。

- a. CCA サーバを起動するときに、CA Configuration Automation コネクタの Catalyst コンテナ サービスを再起動していることを確認します。
- b. 接続の問題を回避するために、CA Catalyst サーバおよび CA Configuration Automation コネクタ サーバ ノードの時間設定が同じであることを確認します。
- c. `problem.log` ファイルに、CCA Catalyst コンテナと CA Catalyst サーバ コンテナのエラーがないことを確認します。Windows の場合、ログ ファイルは以下の場所にあります。

`C:\Program Files\CA\Catalyst\<ContainerID>\container\data\log`

Linux の場合、ログ ファイルは以下の場所にあります。

`/opt/CA/Catalyst/<ContainerID>/container/data/log`

エラーが存在する場合は、手順 f を実行し、エクスポートを確認します。

- d. Catalyst ジョブを使用して CA Configuration Automation コネクタからデータをインポートした場合は、ジョブが完了していることを確認します。

この検証は、統合をベースにした管理プロファイルで使用できます。ジョブが完了している場合、CCA サーバの [ログ] タブには、開始されたジョブとジョブの完了メッセージが表示されます。

- e. Catalyst ジョブの [エクスポート サマリ] タブ、または CI のエクスポートに使用された管理プロファイル ジョブのいずれかを確認します。[Catalyst CI カウント (失敗)] 列がゼロ以外の場合は、リンクをクリックして、失敗した CI の詳細を確認します。
- f. エクスポートの失敗の根本原因を特定します。
 - CA Configuration Automation コネクタの予測とチェック サムテーブルを照合します。
 - 以下の場合、データベースをクリーンアップしてください。
 - CI が CA Configuration Automation から CA Catalyst に正しくエクスポートされていない。
 - チェックサム エントリが CA Catalyst の CI に一致していない。

チェックサムテーブルの照合

CA Catalyst サーバに送信されるすべての CI に対して、CA Catalyst の永続性ストアが保持する CA Configuration Automation コネクタの予測とチェックサムテーブルとを照合します。

注: CA Configuration Automation コネクタは、CA Catalyst サーバに送信されるすべての CI のチェックサムテーブルを保持します。

次の手順に従ってください:

1. CA Configuration Automation データベースで以下のクエリを実行して、失敗した CI を検索します。

注: 対応するデータベース名、パスワード、およびその他の詳細に合わせて、クエリを変更してください。また、データベースには、CA Configuration Automation および CA Catalyst データベースに対するマシン間のデータベースアクセスを設定する必要がある場合があります。

- 失敗した CI を識別するために SQL データベースで以下のクエリを使用します。

```
select ci_id = CASE CHARINDEX( '|', ci_typ)
WHEN '0' THEN ci_id
      ELSE (ci_id + substring( ci_typ, CHARINDEX( '|', ci_typ) + 1,
len(ci_typ)))
END collate sql_latin1_general_cp1_ci_as from acm_catlst_ci_cksum
EXCEPT
select c_mdrelementid from [catalystdb].[dbo].t_ci_detail where
c_mdrproduct = 'CA:00033'
```

- 失敗した CI を識別するために ORACLE データベースで以下のクエリを使用します。

```
CREATE PUBLIC DATABASE LINK catalystdb CONNECT TO catalystadmin IDENTIFIED
BY <Password> USING '//<DBServer>/<DB SID>';
select CASE instr( ci_typ, '|')
HEN 0 THEN cast( ci_id as NVARCHAR2(36) )
      ELSE concat( cast( ci_id as NVARCHAR2(36) ), substr( ci_typ,
instr( ci_typ, '|') + 1, length(ci_typ)))
END as c_mdrelementid
from acm_catlst_ci_cksum
minus
select c_mdrelementid from t_ci_detail@catalystdb where c_mdrproduct =
'CA:00033'
```

2. 手順 1 で識別された不明な ID を以下のディレクトリの invalidCIs.log ファイルを確認します。

```
%CATALYST_HOME%\<ContainerID>\container\data\log
```

すべての属性のソース データが各 USM の正しい形式であることを確認してください。そうでない場合は、詳細について CA Technologies のテクニカル サポートにお問い合わせください。識別された ID が **BinaryRelationship** である場合は、ソースおよびターゲット CI がジョブの一部として正常にエクスポートされていることを確認します。

ソースおよびターゲット CI がエクスポートされていることを確認するには、関連する CI の **SourceMDRElementID** および **TargetMDRElementID** を USM Web ビューで検索してください。

CCA から Catalyst に CI が正常にエクスポートされていない場合は、データベースをクリーンアップします。

エクスポート処理中に多くの CI のエクスポートが失敗した場合は、データベースをクリーンアップするために CA Catalyst データベースからデータをパージします。

次の手順に従ってください:

1. 環境内のすべてのノード上で、CA Catalyst Container CatalystConnector サービスを停止します。
2. (オプション) CA Catalyst Administrator および CA Catalyst Registry サービスを停止します。
3. CA Catalyst データベース上の Microsoft SQL Server Management Studio などの適切なツールから、以下の SQL コマンドシーケンスを実行します。

```
delete from t_ci_detail
delete from t_ci_timestamp
delete from t_notebooks_timestamp
delete from t_tags
delete from t_rest_access
delete from t_connector_init_status
```

さまざまな USM タイプに対して、より多くのテーブルが存在します。**t_ci_detail** からデータが削除されると、テーブルからデータが削除されます。

4. 以下の SQL コマンド シーケンスを実行します。 CA Configuration Automation テーブルをクリーンアップしてデータを再発行するために、Microsoft SQL Server Management Studio などの適切なツールからコマンドを実行します。

```
delete from acm_catlst_ci_cksum  
delete from acm_catlst_ci_summary
```

5. 各 Catalyst ノードから以下のファイルを削除します。
CA_CATALYST_HOME が C:\Program Files\CA\Catalyst であると仮定します。

```
C:\Program Files\CA\Catalyst\<ContainerID>\container\data  
C:\Program Files\CA\Catalyst\<ContainerID>\container\CatalystDataStore  
C:\Program Files\CA\Catalyst\<ContainerID>\nls-store  
C:\Program Files\CA\Catalyst\<ContainerID>\solr\data
```

これらのファイルは、さまざまな用途に使用されます。たとえば、CatalystDataStore フォルダは、差分処理の一部としてコネクタの起動時に使用されます。これは、コネクタがダウンしている間に発生した変更を特定します。

注: CA_CATALYST_HOME がサーバ上の別のディレクトリを指している場合は、ファイル名を変更します。

6. (オプション) CA Catalyst Administrator および CA Catalyst Registry サービスを停止している場合は、開始します。
7. 環境内のすべてのノード上で、CA Catalyst Container CatalystConnector サービスを開始します。
8. Catalyst ジョブまたは管理プロファイルを再実行してデータをエクスポートし、それらが正常にエクスポートされていることを確認します。