



コンテンツ展開の管理

この章は次のトピックで構成されています。

- コンテンツ展開の管理、1 ページ

コンテンツ展開の管理

概要

Catalog Deployer は、サービス設計者、カタログ発行者、財務管理者、および組織構築リソースが開発サイト、テストサイト、および本番サイト間でアプリケーションエンティティを移行するためのコンテンツ展開と構成管理に使用できます。

また、Catalog Deployer は変更管理プロセスと変更履歴を顧客に提供することによって、コンテンツの変更や Service Catalog のすべてのモジュールで使用される組織エンティティの変更に対する信頼性の高い制御を可能にします。

Catalog Deployer では、シスコによりブランド化コンテンツライブラリ内にパッケージされた事前設定サービスの展開もサポートされています。このようなサービスは、そのままの状態で、または組織要件に対してサービス定義をカスタマイズするためのテンプレートとして使用できるため、実行可能なサービス カタログの実装に必要な時間と労力が大幅に削減されます。

Catalog Deployer を使用したコンテンツ展開

Catalog Deployer には 2 つのコンテンツ展開方法が用意されています。

- ソース サイトは Catalog Deployer を使用して、ターゲット サイトへの送信やそこでの展開用にサービス定義とエンティティのパッケージを「抽出」したり組み立てたりできます。この場合、すべてのオペレーションを Catalog Deployer モジュール内で実行でき、外部プログラムは不要です。

- また、パッケージはエクスポート用に生成できます。エクスポートされたファイルは、Catalog Deployer 経由でターゲットサイトの Service Catalog のインスタンスにインポートされます。これらはテキスト形式の XML ファイルであるため、構成管理またはソースコード制御システムにも格納できます。ブランド化コンテンツライブラリはエクスポートファイルの形で配布されるため、標準的な Catalog Deployer 機能を使用して、ライブラリで使用可能なサービスをインポートしたり展開したりできます。

他のすべての Service Catalog モジュールと同様に、Catalog Deployer は使用のための権限を付与できます。ユーザには、展開履歴の表示、展開パッケージの作成と組み立て、特定のサイトへのパッケージのインポートや展開を行う権限、またはこれらをユーザの責務に最適になるように組み合わせた権限を持たせることができます。このような機能はすべて、Organization Designer モジュール内の標準ロールまたはカスタムロール経由で割り当てることができます。

主な特徴と機能

主な機能は次のとおりです。

- ユーザインターフェイスによる使用が簡単。展開用パッケージが作成され、データ入力、対象サイトに送信、またはファイルシステムにエクスポートできます。
- 2つのコンテンツ転送方法。サイト間の XML パッケージ送信またはエクスポート/インポート機能によるファイルベース送信。
- 1つ以上のターゲットサイトでのサービス展開（関連エンティティを含む）。
- ターゲットサイトにおける組織展開（OU、グループ、キューリー、ロール、人員、役職）。
- ターゲットサイトにおける電子メールテンプレートおよび Service Link エージェントの展開。
- 変更管理サポート（コンテンツ開発者/管理者およびカタログパブリッシャ間の責務の分離が選択可能）。
- ホット展開：ユーザがアプリケーションを使用可能なときにコンテンツを展開可能。
- オンラインオプションによる抽出/展開履歴の表示。
- サイト保護レベルおよび Entity Homes のサポート。
- ブランド化コンテンツライブラリからターゲットサイトに展開する前に、サービス定義のレビューが可能。

Catalog Deployer は、カスタマイズされたサービスカタログの基礎を形成するためのテンプレートコンテンツを提供する Service Catalog の試みの最初で使用することができます。また、Catalog Deployer は開発作業のビルトおよびメンテナンスの段階で使用して、テスト済みコンテンツを開発から他の環境にプロモートしたり、複数の環境を同期させたりもできます。

構成管理

以降の項で、Catalog Deployer を使用したリリースと構成の管理に関する機能について説明します。

- アプリケーションのロールおよび機能, (8 ページ)
- Catalog Deployer の設定, (12 ページ)
- Catalog Deployer パッケージ, (19 ページ)
- サンプルの展開シナリオ, (34 ページ)

サービス カタログおよびポートフォリオの開発

この章の次の項では、Cisco コンテンツライブラリをインストールしてサービス カタログやサービス ポートフォリオの基礎を提供するための Catalog Deployer の使用に関する機能を説明します。

- Catalog Deployer パッケージ, (19 ページ) (コンテンツのインポートおよび展開に関する項のみ)
- ブランド化コンテンツ ライブラリ, (42 ページ)

Catalog Deployer のアーキテクチャ

Catalog Deployer には、アプリケーションにある論理エンティティに基づいて編成された、データベース中立のデータ転送インフラストラクチャおよびインターフェイスがあります。論理エンティティの例としては、サービス定義、データディクショナリ、人員、および組織単位などがあります。完全なリストは、この項の最後にあります。ただし、異なるデータベースやアプリケーション環境間のカスタム コンテンツのサポートは制限されています。Catalog Deployer でサポートされないエンティティのリストについては、[サポートされていないエンティティ, \(6 ページ\)](#) を参照してください。

実装、サイト、およびエンティティ ホームについては、[表 9 : 主な用語, \(47 ページ\)](#) に説明があり、これらは Catalog Deployer の操作に不可欠です。

特に、論理エンティティのホームサイトは Catalog Deployer での構成管理に欠かせない部分です。Catalog Deployer ではオプションですが、論理エンティティに対してホームサイトのシステムを作成することによって、サイト間の論理エンティティの参照整合性と、サイト間の設定情報の整合性の両方の管理が可能になります。

論理エンティティ ホームサイトの背後にある概念は、特定のサイトは、他のサイトに比べ、論理エンティティタイプについて、より信頼性の高いバージョンのデータを持っているというものです。たとえば、ユーザ情報の認証および収集に自分の LDAP ディレクトリを使用しているカスタマーの場合、人員と組織単位について最も正確なデータがあるのは本番サイトということになります。このため、本番サイトが、人員および組織単位のホームサイト（記録のサイト）になります。もし、本番サイト以外のサイトで人員の作成や組織単位の編集が可能な場合、このデータを

他のシステムに展開すると、記録システムとの同期が失われ、実装におけるデータの品質が低下します。

そのため、アプリケーションフレームワークは、エンティティのホーム サイト以外のサイトでユーザがこれらのエンティティを修正することができないように、保護レベルを論理エンティティに割り当てる能够性を有するように設計されています。サイト管理者は、[Catalog Deployer の設定](#)（12 ページ）に説明されているように、4 つの設定から 1 つを選択することにより、論理エンティティに提供する保護のレベルを選択できます。

Catalog Deployer はエンティティの一部としてエンティティの権限を展開します。権限がそのホームサイトのエンティティから除去された場合、このアプリケーションは、Catalog Deployer がこの除去の複製のために使用できる削除タブ（またはトランザクションログ）を残しません。たとえば、サービスをオーダーする権限がサービス定義の組織単位から除去されたとき、Service Catalog はこの事実を保持せず、権限を除去するだけです。

権限が変更されたサービスを展開すると、サービス定義とその権限の間のすべての関連付けが対象サイトにドロップされ、ソースサイトで有効な権限に従って再作成されます。ただし、権限がカスタム ロールまたはグループに付与されている場合に、ロールまたはグループがソースサイトから削除されても、ロールまたはグループは対象サイトから削除されません。Catalog Deployer はエンティティの削除をプロバゲートできません。

Catalog Deployer でサポートされている/サポートされていないエンティティ

次の表は、Catalog Deployer によってサポートされているすべての論理エンティティと、それらに関連付けられたアプリケーション モジュールの一覧を示したものです。

表 1: サポートされている/サポートされていないエンティティ

モジュール	サポートされているエンティティ
Service Designer	<p>サービス定義（提供、フォーム、フォームセクション、計画、承認、権限）</p> <p>サービス定義とともに自動的に展開されるコンポーネントエンティティ。</p> <ul style="list-style-type: none"> サービス グループ ディクショナリおよびディクショナリ グループ アクティブ フォーム コンポーネントおよびコンポーネント グループ キーワード、カテゴリ、およびプレゼンテーションの各要素 スクリプト機能およびライブラリ <p>サービス定義によって参照されるエンティティ。</p> <ul style="list-style-type: none"> Email Templates Organization Designer のエンティティ Service Link のエージェントおよび変換
サービス項目マネージャ (Service Item Manager)	<p>サービス定義とともに自動的に展開されるコンポーネントエンティティ。</p> <ul style="list-style-type: none"> サービス項目（サービス項目ベースのディクショナリまたはテーブルベースのデータ取得ルールによって参照される場合） 標準（テーブルベースのデータ取得ルールによって参照される場合）サービス項目と標準を別々に展開することもできます。
Service Link	<ul style="list-style-type: none"> エージェント (Agents) 選択されたエージェントに関連付けられた変換も展開されます。

モジュール	サポートされているエンティティ
Organization Designer	<ul style="list-style-type: none"> • キュー • 組織 • グループ (Groups) • ロール • 役職 • 人員 (People)
管理 (Administration)	<ul style="list-style-type: none"> • Email Templates • ターゲットタイプを使用して、インフラストラクチャ/アプリケーションのテンプレートのスタックコンポーネントを新しい環境にエクスポートできます。カスタムパッケージでは、ターゲットタイプはデフォルトで表示されます。
需要管理	<ul style="list-style-type: none"> • アカウント定義 • 契約テンプレート • 課金料率

Catalog Deployer は、上記のエンティティをソースサイトのデータベースからターゲットサイトのデータベースにコピーします。Catalog Deployer は、ファイルシステムに格納されている情報の移動、コピーは行いません。Catalog Deployer は、JavaScript 関数に関連付けられたライブラリの定義をコピーします。

サポートされていないエンティティ

Catalog Deployer は、同じバージョンの Service Catalog を使用して実装されている 2 サイト間でトランザクションデータベースに格納されている論理エンティティを移行します。エンティティは .xml の形式でテキストストリームに抽出されます。これは展開用パッケージの一部であり、パッケージの作成に使用される仕様（任意）も含まれています。そのパッケージは、バックアップメカニズムとして、および別のサイトのターゲットデータベースにエンティティ定義を展開するためのソースとして機能するために、ソースデータベースから抽出できます。エンティティ定義は変更されずに対象環境に展開されます。



(注)

Catalog Deployer は、.bmp 形式のイメージ（プレゼンテーション要素）の展開をサポートしていません。Service Designer は現在、このようなイメージの指定ができないようになっています。古い形式のイメージは、.jpg または.gif など、Web 上のプレゼンテーションに最適化された別の形式に変換する必要があります。

Catalog Deployer は論理エンティティ以外の Service Catalog のいずれのコンポーネントにも影響しません。これらのコンポーネントのいくつかに対する変更は、Service Link によって処理される構成管理シナリオ（カスタマーが設計および構成した設定項目の実装間での同期）の一部でないため、これ以上ここでは検討する必要がありません。次に例を示します。

- ・ソフトウェア コンポーネントは Service Catalog インストーラでインストールして設定する必要があります。
- ・データマートおよびレポーティングテーブルのコンテンツは Service Catalog インストーラで作成して Extract-Transform-Load (ETL) プロセスで入力する必要があります。
- ・スキーマは Service Catalog のインストールの一部としてアップグレードされます。
- ・カスタマイズされたすべての Service Catalog コンポーネント、または追加コンポーネントは Service Catalog インストーラの [カスタマイゼーション (Customizations)] オプションにより、このガイドに記載されている手順ですべてのサーバにインストールする必要があります。このようなカスタマイズには一般に、拡張サービス組織によって提供される API や、ディレクトリ統合で使用されるカスタム マッピングのサポートが含まれます。

ただし、追加の設定項目は、Catalog Deployer によって処理される論理エンティティに対する変更と組み合わせて展開しなければならない場合があります。これらの詳細を、影響を受ける論理エンティティとともに、以下の表に示します。

表 2: サポートされていないエンティティ

設定項目	制御および移行する追加のアーティファクト
外部ディクショナリ	データベースで実行される DML および DDL スクリプト
データソース	外部ディレクトリや、データ取得ルールで参照される SQL ベースのオプションリストまたはテーブルを参照するデータソースの指定
データ取得ルール	ルールに直接入力される SQL ステートメント（異なるデータベースタイプ間では互換性がない場合がある）
ISF スクリプトライブラリ	アプリケーションサーバに展開されるライブラリ (JavaScript) テキスト ファイル

設定項目	制御および移行する追加のアーティファクト
カスタム アダプタ	Service Link の Adapter Development Kit (ADK) によって生成される展開ファイル

アプリケーションのロールおよび機能

重要なロールは次のように定義されます。

表 3: 重要なロール

ロール	定義
Catalog Publisher	サービス カタログの作成と管理、ランタイム アプリケーションへのカタログコンテンツの展開、カタログのルック アンド フィールと構造の設定を行います。また、サービス定義と提供計画の変更に従って、展開されたカタログコンテンツの継続的な更新を行います。
Service Designer	カスタマー サイトでサービス定義を設計します。サービス設計者は、カスタマー IT チームのカスタマーに提供されるサービスについて豊富な専門知識を持ち、Service Designer および Organization Designer の使用法に習熟しています。サービス設計者は、技術的には中級ですが、一般的にはエンジニアというよりはビジネス アナリストです。 サービス設計者は、開発サイトでサービス定義を頻繁に作成したり変更したりしなければならない場合があります。ステージング サイトまたは本番サイトへ作業をパブリッシュするために Catalog Deployer の使用が許可される人もいます。

ロール	定義
Organization Builder	<p>カスタマーサイトで Organization Designer の組織エンティティを設計します。組織ビルダーは、Service Catalog を正常に展開して使用するためには必要な組織単位、グループ、およびロールの設定に関する組織のニーズについて豊富な専門知識を持っています。</p> <p>組織ビルダーは、Organization Designer モジュールの使用法に習熟している必要があります。組織ビルダーは、技術的に高度である必要はありませんが、アプリケーション管理の IT リソースである必要があります。</p>
Site Administrator	<p>カスタマーサイトでアプリケーションの権限を管理します。この担当者は、システムの最初のユーザになる場合があります。Global Configurations の設定など、アプリケーションのバックエンドのすべてにアクセスできます。言語や課金カテゴリなどのリストの管理も行います。</p>
変更マネージャ	<p>カスタマーサイトで実装の変更要求を承認します。この担当者は、通常、本番サイトの安定性に責任を負っています。本番サイトへの展開に先立ち、すべての変更内容を理解しておく必要があります。このロールはオプションですが、正式な変更制御プロセスが確立されたカスタマーサイトで必要になります。</p>
Finance Designer	<p>カスタマーのアカウントを作成し、各アカウントまたは OU のクォータを定義するための協定を設定し、サービス品目運用の課金料率を定義します。カスタマーのサービス品目の消費と課金/チャージバックを管理します。</p>

次の表は、Catalog Deployer モジュールに関して、システム定義のロールおよび各ロールに付与されるデフォルト機能を説明したものです。

表 4 : 事前定義されたロールおよび機能

定義済みの役割	機能					
	拡張サービス展開の管理	カスタム展開の管理	展開のインポート	展開用パッケージの展開	ブランドコンテンツライブラリのパッケージング	
基本サービス展開の管理						
Catalog Designer and Administrator	P	P	P			
Organization Designer			P			
Site Administrator	P	P	P	P	P	P
Catalog Publisher	P	P	P	P	P	
Licensed Content Publisher	P	P	P	P	P	P

Service Catalog の実装と構成管理

この項では、Service Catalog の実装方法と、業界標準の構成（または変更）管理プラクティスをサポートする関連プロセスの設計方法を説明します。

以下では、一般的な方法と（Catalog Deployer 設定管理など）Service Catalog アプリケーションで使用可能な方法について説明し、両者を比較します。初期開発からテスト、展開、およびメンテナンスまで、展開の段階に一致した、構成管理に対するさまざまなアプローチを説明します。

Catalog Deployer のベスト プラクティス

Catalog Deployer は、IT 業界標準の構成管理方法およびテクノロジーで採用されているベスト プラクティスを実装します。そのため、これらのプラクティスを確認しておくと役に立ちます。

- ・ソフトウェア設定項目（つまり、IT アプリケーションを構成する個々のソフトウェアモジュールまたはコンポーネント）に対する変更は開発環境で行われます。
- ・同じ（開発）環境において、変更されたソフトウェアは一般に予備テスト（単体テスト）を受けます。

- ・単体テストにソフトウェアが合格すると、ソフトウェアの「ソース」のコピーがソースコード管理システムにチェックインされ、リリース候補のラベルが付けられます。「ソース」は、テキストエディタで作成されて保守されるコードを含む、複数のタイプのアーティファクトで構成される場合や、あるいは、ますます一般的になりつつあるものとして、統合開発環境の一部になっているメタデータ リポジトリ内に、または XML ファイルに格納された仕様で構成される場合があります。
- ・保存されたソースは、厳重に制御されたテスト環境に展開され、そこで厳しいテストが行われます。テスト環境は、異なるテストの実行結果を確実に比較できるようにするために、テストの各セットの前に再初期化される場合があります。
- ・テスト実行者は、問題を見つける責任はありますが、それを診断または修正する責任はありません。すべての問題は、開発チームに報告されます。開発チームは、開発環境を使用して、問題の修正、コードの再テスト、および修正されたソースのコピーの保存を行います。
- ・修正、抽出、展開、テストの各ステップは、ソフトウェアがすべてのテスト条件（これらはパフォーマンス対策または機能要件またはその組み合わせの場合があります）を満たすことをテストチームが認定するまで繰り返されます。
- ・テスト環境に展開され（そしてテストされ）た同じソースが、本番環境に展開されます。

構成管理

Service Catalog アプリケーションの場合は、開発されるソフトウェアの多くが、カテゴリ、グループ、タスク、チェックリストなどの関連要素を含むサービス定義です。

Catalog Deployer は、お客様が、サービス定義などのコンテンツや関連サービスを展開するために使用できます。

一般的な構成管理シナリオと Catalog Deployer が果たす役割には次のようなものがあります。

- ・新規のサービス定義または拡張されたサービス定義を開発環境で開発して単体テストを行います。
- ・Catalog Deployer を使用して、開発環境から新規のサービス定義または更新されたサービス定義を抽出します。作成された展開パッケージは、必要に応じて、エクスポートされ、ソースコードの制御下に置かれます。
- ・Service Catalog データベースにない他のコードリソース（JavaScript ライブラリなど）もソースコードの制御下に置く場合があります。
- ・Catalog Deployer を使用して、テスト環境または品質管理（QA）環境にサービス定義を展開します。
- ・サービスをテストします。問題が見つかった場合は、サービスが指定要件を満たしたものと認定されるまで、前のステップを繰り返します。つまり、開発でコードを修正して、抽出し、テスト環境に再展開します。
- ・テスト環境で受け入れられると、サービスは、コードをテスト環境に最初に展開した同じ手順に従って本番環境に展開されます。

このシナリオは業界標準に従ったものです。業界標準では、開発環境においてのみ、サービス定義の変更が行われます。また、テスト環境および本番環境にソースコードの制御されたセットを展開するには、自動化されたプロセスが使用されます。

ただし、このシナリオは不完全です。ほとんどの実装では、人員と事業単位（組織のタイプ）が、本番環境に動的に追加されたり、人員が初めて Service Catalog にログインしたときに彼らのためにサービスをオーダーしたり、審査または承認を実施するために割り当てられたりします。したがって、Catalog Deployer は、これらのエンティティを本番サイトから開発に戻す移行のときも使用しなければなりません。それにより、まだ開発中のサービス定義または他の Service Catalog 設定項目（たとえば、承認）でこれらのエンティティを使用できます。さらに、サービス定義は、電子メールテンプレート、グループ、キー、サービスチーム組織、およびロールなどの関連エンティティを参照する場合があります。これらのいずれかが対象環境に存在しない場合、足りないものを対象環境に展開しなければ、サービスの展開パッケージは正常に展開されません。

Catalog Deployer の設定

Catalog Deployer の設定には次の作業が含まれます。

- 1 Service Catalog をインストールして、Catalog Deployer をサポートするようにクライアントワークステーションを設定するための前提条件を満たす。
- 2 Catalog Deployer を含む Service Catalog のバージョンをインストールする。Catalog Deployer はすべての Service Catalog サイトの一部として自動的にインストールされます。



(注)

サービスパックを含むすべてのサイトで、同じ Service Catalog のバージョンを使用する必要があります。

- 3 開発と本番の Service Catalog インスタンス内で実装とサイトを設定する。
- 4 ソースサイトとターゲットサイトでアプリケーションサーバ JDBC データソースを設定する。
- 5 Administration モジュールと Organization Designer モジュールを使用して、実装での働きに適した Catalog Deployer 機能に担当者がアクセスできるようにする。



(注)

さまざまなデータベースからインポート（Catalog Deployer または REX を使用して）された「個人」と「エージェント」データのパスワードをターゲットデータベースでリセットする必要があります。これは、キー暗号キーが Prime Service Catalog データベースのインスタンスごとに異なるためです。「個人」と「エージェント」のパスワードがリセットされなかった場合は、それらをターゲットデータベースで正しく復号化できません。

クライアントワークステーションの設定

Catalog Deployer によって生成される展開パッケージは、含まれるエンティティの圧縮された XML 表現を含みます。Catalog Deployer はパッケージのファイルベース送信を使用します。その中で、

パッケージコンテンツは、パッケージのエクスポートおよびターゲット サイトへの以降のインポートにより、あるサイトから別のサイトへ転送されます。

このファイルベースの転送をサポートするには、暗号化されたページをディスクに保存できるようユーザのブラウザを設定する必要があります。次の手順を実行します。

-
- ステップ1** [ツール (Tools)]>[インターネットオプション (Internet Options)]>[詳細設定 (Advanced)]を選択します。
- ステップ2** [暗号化されたページをディスクに保存しない (Do Not Save Encrypted Package to Disk)] チェック ボックスをオンにします。
- ステップ3** [OK] をクリックします。
-



(注) すべての展開がサイト間の直接送信で行われている場合、この設定は不要です。

実装とサイトの設定

Catalog Deployer を使用する前に、開発インスタンスおよび本番インスタンスを使用して実装を設定する必要があります。実装は、Catalog Deployer が Service Catalog サービス定義とその他の Service Catalog エンティティを移行するサイトを集めたものです。

実装とサイトの設定手順の概要は以下のとおりです。詳細は以降のセクションで説明します。

-
- ステップ1** 実装に名前を割り当てます。この名前は一般に会社やプロジェクトを表します。この名前はマニュアル記載のみを目的とし、Catalog Deployer によって使用されることはありません。
- ステップ2** 実装内のサイトに名前を付けます。
実装内では、各サイトに一意の名前を付ける必要があります。Service Catalog は、エンティティの変更（追加、修正、削除）をユーザーに許可する保護レベルを Service Designer および Organization Designer のページに割り当てるために、この固有の名前を使用してサイトを識別します。
- 通常、開発 (DEV) と本番 (PROD) の2つのサイトが必要です。設定管理および移行計画で必要になった場合は、追加のサイト (STAGE、TEST、QA など) を使用できます。
- ステップ3** 各論理エンティティのホーム サイトの指定
各論理エンティティにホーム サイトを (1つ) 指定します。論理エンティティのホーム サイトを決定する際に、以下のルールが役立ちます。
- 論理エンティティのホームは、一般に、1つのサイトのみにすべきです。これにより、エンティティ インスタンスに対する変更をよりよく管理できます。
 - サービスの定義に関係する論理エンティティのホームは、明らかに開発インスタンスにすべきです。

Catalog Deployer の設定

- システムの本番使用によって作成されるエンティティのホームは、本番サイトにすべきです。Import Person イベントを含むシングルサインオン (SSO) またはディレクトリ統合が有効な場合、本番の論理エンティティ ホームは人を含みます（したがって、同じテーブルにあるキーも含みます）。自動化されたユーザ作成ファシリティによって組織単位が作成された場合、組織単位のホームも本番サイトに置く必要があります。

(注) 論理エンティティのホームがテストサイトやステージサイトに置かれることは、ほぼありません。これは、このようなサイトは通常、公開データおよびテスト対象の新しく開発されたコードを使用して再構築されるためです。

ステップ4 各サイトのデータソースの作成

Catalog Deployer でコンテンツをサイトに展開するためには、Service Catalog アプリケーションを実行しているアプリケーションサーバで、そのサイトの JDBC データソースを設定する必要があります。たとえば、開発サイトで開発したサービスをテストおよび本番に展開するには、開発サイトは本番サイトおよびテストサイトの両方のデータソースを含んでいなければなりません。本番サイトから開発に組織エンティティを展開するには、本番インスタンスは開発インスタンスのデータソースを含んでいなければなりません。

クラスタ化されたサイトの場合、サイトを構成する各ノードがデータソースにアクセスできる必要があります。

ステップ5 各サイトでのプロセスの繰り返し

実装、サイト、および論理エンティティを指定するデータは、順番に、論理エンティティ内に格納されます。

これらの仕様は、実装を構成するすべての Service Catalog インスタンスに存在しなければなりません。

また、すべてのサイトで適切なデータソースを作成する必要があります。

Administration モジュールでの実装の設定

Catalog Deployer を使い始めるには、Administration モジュールで実装環境の設定を行う必要があります。Administration で設定した値により、ソースサイトは対象サイトを認識できるようになります。

アプリケーションサーバコンソールの JDBC データソースのページでデータソースを定義した後で、次の手順を実行する必要があります。

実装を設定するには、次の手順に従います。

ステップ1 Administration 特権を持ったユーザとして開発インスタンスにログインします。**ステップ2 モジュールのドロップダウンメニューから、[管理 (Administration)] を選択します。****ステップ3 [Settings (設定)] タブをクリックします。****ステップ4 右側のメニューから、[Entity Homes] を選択します。**

[論理エンティティのホームの仕様 (Logical Entity Home Specification)] ページが表示されます。

これらの設定は、Catalog Deployer と Service Designer モジュールおよび Organization Designer モジュールの動作に影響します。設定が正しくないと、間違ったデータが本番サイトを含む Service Catalog アプリケーションサイトに書き込まれる可能性があります。これらの設定の変更は、システム管理者だけが行うようになります。

- ステップ 5** [実装サイト (Implementation Sites)] セクションのページ下部にあるテキスト フィールドに、サイト名を入力します（例：「Development」）。
- ステップ 6** [データソースの選択 (Select a Data Source)] ドロップダウン メニューからデータ ソースを選択し、[新規追加 (Add New)] をクリックします。
サイト名のリストにこのサイト名が追加されます。
- ステップ 7** 他のサイトとして、本番サイトの名前を追加します。
- ステップ 8** QA または Test など、Catalog Deployer でリフレッシュする必要のある他のサイトが実装に含まれる場合は、これらも追加します。
- ステップ 9** ページ上部にある [実装名 (Implementation Name)] フィールドに実装名を入力します（例：My Cloud や Data Center Management）。
- ステップ 10** 現在のサイトを識別するために [このサイト (This Site is)] ドロップダウン メニューから開発サイトを選択し、[更新 (Update)] をクリックします。
- ステップ 11** エンティティの [ホームサイト (Home Site)] の割り当てを確認し、要件に合わないところを変更します。
完了したら、[更新 (Update)] をクリックします。
- ステップ 12** 定義した各サイトに適切な [サイト保護レベル (Site Protection Level)] を割り当てます。完了したら、[更新 (Update)] をクリックします。これらの保護レベルは、対応する論理エンティティの保守を行う Service Designer、Organization Designer、および Administration のページの動作を変更します。

表 5: 保護レベル

保護レベル	ホーム以外のサイトのユーザインターフェイスに対する影響
なし	なし：エンティティを作成および変更するための UI 要素がすべて使用可能なままになります。 保護レベル [なし (none)] は、実装の初期段階の開発サイトでのみ使用すべきです。このレベルでは、適切なロールを持つユーザなら、サイト内のすべてのエンティティを作成、変更、削除することができます。
作成のみ (Create only)	新しい論理エンティティの作成の制御が無効になります。
作成、変更 (Create, Modify)	論理エンティティの作成、更新の制御が無効になります。

保護レベル	ホーム以外のサイトのユーザインターフェイスに対する影響
作成、変更、削除 (Create, Modify, Delete)	<p>論理エンティティの作成、編集、および削除のすべての制御が無効になります。</p> <p>保護レベル [作成、変更、削除 (Create, Modify, Delete)] は、一般にすべてのテスト、ステージング、または QA サイトに適用すべきです。これらのサイトは、通常、(たとえば、パフォーマンステストやボリュームテストをサポートするために)公開から完全なデータベースをコピーする、または、公開に昇格する前の機能テストのために開発からサービスを展開することによって、リフレッシュされます。</p> <p>保護レベル [作成、変更、削除 (Create, Modify, Delete)] は、一般に本番サイトに適用すべきです。保護レベル [作成のみ (Create only)] では、エンティティ名の変更などのマイナー修正を保護エンティティに適用できます。</p>

(注) 1つの実装内のすべてのサイトで、同じサイト名、エンティティ ホーム設定、保護レベルを指定する必要があります。これは、ユーザがエンティティを誤って間違ったサイト（そこでは、エンティティが「同じ」エンティティの次の展開によって上書きされ、異なるサイトで開発および保守される可能性があります）で作成または修正することを防止します。

サイトのデータ ソースの設定

Catalog Deployer がパッケージを別のサイトに送信するためには、ターゲット サイトに対応する JDBC データ ソースがソース サイトと同じアプリケーションサーバ上で設定されている必要があります。JDBC データ ソースを設定する手順は、『Cisco Prime Service Catalog Installation and Upgrade Guide』に記載されている RequestCenter データ ソースを設定する手順と同じです。

Service Catalog でデータ ソースを検出するには、同じ JNDI 名プレフィックスを使用する必要があります。検出されたデータ ソースのリストが [管理 (Administration)] > [設定 (Settings)] > [データソースレジストリ (Data Source Registry)] に表示されます。

特定のデータ ソースは、レポートを作成する目的で使用され、Service Designer や Catalog Deployer のユーザに表示するためには使用されません。データ ソースをこれらのユーザーに必要なものだけに制限するには、これらのデータ ソースの [エンティティ ホーム 定義] に使用 (Use for Entity Home Definition) チェックボックスをオンにしてから、[更新 (Update)] をクリックします。

サイトの設定

最初に、1つのサイトを設定します。通常は、開発サイトから始めます。開発サイトは、ソースサイトに表示される他のターゲット サイトに移行されたサービス定義のソース サイトです。実

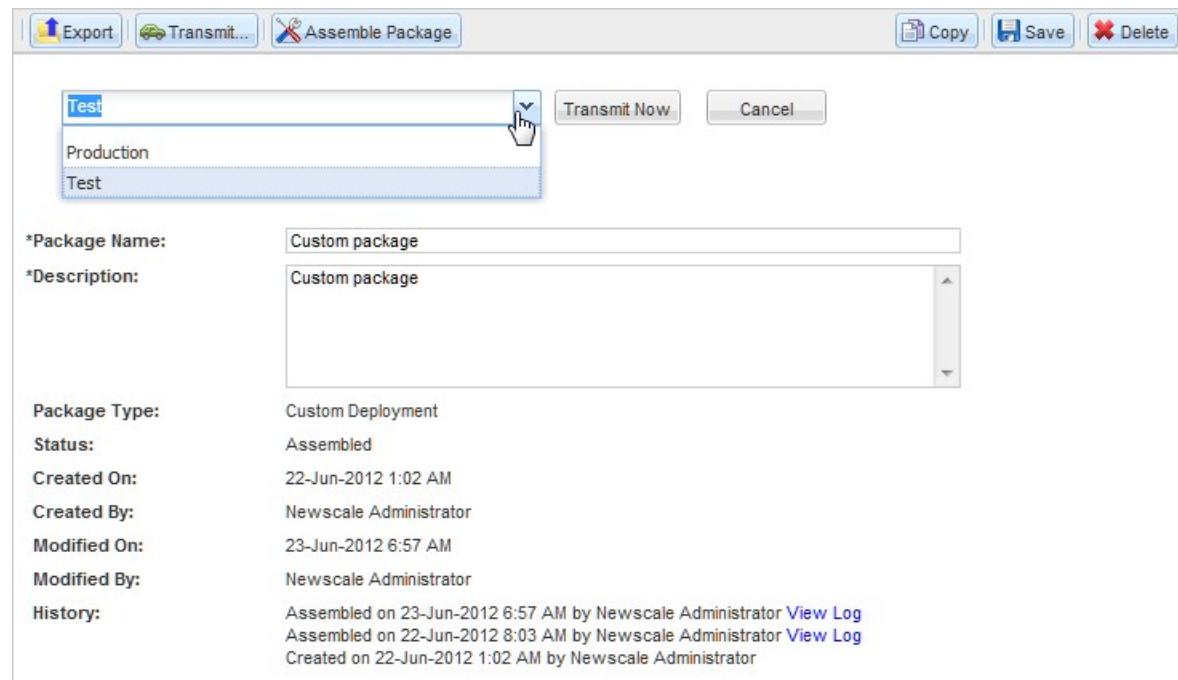
装全体の構成に応じて、どのサイトでもソースサイトまたはターゲットサイトとして、あるいは潜在的に両方のサイトとして、設定できます。

たとえば、テストサイトと本番サイトが見えるように開発サイトを設定、および開発サイトとテストサイトが見えるように本番サイトを設定します。このシナリオでは、まず開発からテストにエンティティを展開します。次にテストが成功したと仮定し、開発から本番にエンティティを展開します。次に、組織情報を本番サイトから、テストサイトと開発サイトの両方に移行します。

必要なデータ ソースの定義を含め、潜在的なターゲットサイトの設定をすべてのソースサイトで繰り返します。

設定が正しいことを確認するには、各サイトにログインして、Catalog Deployer に移動し、[アクション (Action)] ペインで [送信 (Transmit)] をクリックすることによって、パッケージを、今設定したターゲットサイトの1つに送信します。ドロップダウンメニューに選択可能なターゲットサイトの1つとして自分のサイトが表示されます。

図 1: ターゲットサイト



362511

Catalog Deployer のパフォーマンスの考慮事項

Catalog Deployer は、ソースサイトまたはターゲットサイトがオンラインで使用中の場合に実行する必要があります。ただし、この使用には何らかの制限を設けることをお勧めします。

Catalog Deployer の同時使用

Catalog Deployer は、パッケージの組み立てや展開だけでなく、サービスのプレビューにも大量のメモリを消費します。メモリ消費から生じる問題を避けるために、Catalog Deployer では、6人以上のユーザが同時にパッケージの組み立て、インポート、またはプレビューを行うことができないようになっています。6人目のユーザがパッケージの組み立て、インポート、またはプレビュー

を行おうとすると、Catalog Deployer からその旨のアラートが表示され、後で再試行するようにユーザに通知されます。通常、本番展開は1人の担当者が処理するため、このような状態が本番環境で発生する可能性はありません。しかし、開発環境やテスト環境で、複数のサービス設計者がそれぞれのコンテンツを展開のためにパッケージ化している場合は、この状態が発生することがあります。

ブラウザ セッションのタイムアウト

パッケージの組み立てまたは展開は、まだブラウザを介した操作と見なされています。そのため、Administration モジュールで設定された非活動用のセッションタイムアウトで、大規模なパッケージの組み立てや展開がタイムアウトすることはありません。

パッケージ サイズ

Catalog Deployer は、エンティティが複数のプライマリエンティティに必要な場合に、関連付けられたエンティティをパッケージ内で1回作成/更新するだけです。たとえば、展開パッケージ内の20のサービスにディクショナリ「IT Software Configuration」が必要な場合、Catalog Deployer はそのディクショナリをターゲットサイトで1回作成/更新するだけです。共通コンポーネントを持つサービスを同じパッケージにまとめることで、パッケージのサイズと、Catalog Deployer が行う必要のある冗長な作成/更新の数を減らすことができます。

大きいパッケージ（サイズによって定義）のパフォーマンスは時間がかかることが予想されます。また、このようなパッケージのアセンブリや展開は失敗する可能性があることにも留意してください。この場合、解決するには、パッケージをプライマリエンティティの比較的小さいセットに分けます。たとえば、20個のサービスで構成される1つのパッケージを作成する代わりに、各10個のサービスのパッケージを2個作成します。

ホット展開

原則として、Catalog Deployer は1回のデータベーストランザクションで完全なパッケージを展開できます。その場合、1つのコンポーネントの展開に失敗すると（たとえば、指定されたキーがターゲット内に存在しないために拡張パッケージ内のサービス定義を展開できない）、全体のパッケージ内容がロールバックされます。対象サイトのリポジトリは、展開が開始する前と同じ状態で維持されます。

ただし、実際には、この全か無かという展開によって問題が発生することがあります。一度エンティティが展開されると、完全なパッケージが展開され、データベーストランザクションがコミットされるまで、オンラインユーザはこれを使用できません。この間に、ユーザが進行中の展開によって更新されたサービスのオーダーを試みると、ユーザのセッションはハングし、サービス定義が使用可能になるまで待機することになります。最終的に、エラーメッセージが表示されるか（最悪の場合）、遅延の後サービスをオーダーできるようになります（最良の場合）。（このエラーが発生する可能性があるのは、ユーザが最初にサービスをオーダーする場合のみです。タスク実行者や承認者がこのサービスで作業する場合には発生しません）。

このシナリオを回避する確実な方法の1つは、公開システムが使用中は展開を許可しないことです。これは、定期的にスケジュールされたメンテナンス期間に公開システムに対する更新を行うという、業界で受け入れられているベストプラクティスに準拠しています。ただし、定期メンテナンスが実行されるまで展開を待機させることができない場合もあります。Service Catalog が稼働中に展開を実施しなければならない場合に問題が発生する可能性を最小限に抑えるために、サービスをより小さな論理パッケージにグループ化することをお勧めします。サービス設計者は、関

連するサービスを同じパッケージにグループ化することによって、共有コンポーネントの小さくなったりしたパッケージ サイズの恩恵を受けることもできます。

Catalog Deployer パッケージ

ここでは、Catalog Deployer の機能を簡単に説明します。展開パッケージタイプ（基本サービス、拡張サービス、およびカスタム）には、展開可能なプライマリ エンティティと、関連エンティティの管理展開に使用可能なオプションに違いがあります。

基本サービス展開用パッケージ

基本サービス展開用パッケージは、Service Designer でのサービス定義のインポートと同様の機能を果たします。ターゲットサイトで関連するエンティティ（キューや組織単位など）が見つからなかった場合は、それらのエンティティが展開で省略されるだけです。基本パッケージにはバンドルされたサービスを含めることができません。

サービス展開用パッケージは標準とサービス項目も収集します。ルックアップ Dynamic Data Retrieval (DDR) とサービス項目ベースのディクショナリで参照されたものが自動的にパッケージに含まれます。標準エントリも展開するかどうかを制御する Administration グローバル設定があります（標準エントリは必ず収集され、制御は展開ターゲットにのみ適用されます）。



(注)

Catalog Deployer で展開用のサービスを表示または選択するために Service Designer の権限は必要ありません。

基本サービス展開用パッケージの内容とその関連するエンティティを以下に示します。

表 6: 基本サービス展開用パッケージ

コンテンツ タイプ	操作	エンティティ タイプ/説明
プライマリ エンティティ	[サービスの内容 (Services content)] タブ経由で選択される	サービス定義：Service Designer の [サービスカタログ (Service Catalog)] オプションで定義されたサービス定義のすべての側面

コンテンツ タイプ	操作	エンティティ タイプ/説明
コンポーネント エンティティ	自動展開	<p>サービス定義によって参照されるすべてのエンティティ：</p> <ul style="list-style-type: none"> • カテゴリ • データ ディクショナリ • ディクショナリ グループ • アクティブ フォーム コンポーネント • アクティブ フォーム コンポーネント グループ • キーワード • 目的 • プレゼンテーション要素 • スクリプト機能 • スクリプト ライブラリ • サービス グループ • サービス項目とサービス項目 グループ • 標準と標準 グループ • 内線番号 (Extensions)
関連エンティティ	展開では、ターゲット サイトに存在しないエンティティ 関連付けが省略されます。	<p>サービス定義によって参照されるすべてのエンティティ：</p> <ul style="list-style-type: none"> • エージェント (Agents) • Email Templates • 役職 • グループ (Groups) • 組織 • 人員 (People) • ロール • キュー

デフォルトで、Catalog Deployer は標準データと標準定義を展開します。この動作を無効にするには、Administration 設定を [標準テーブルのエントリ(データ)の展開 (Deploy Entries (data) in Standards Tables)] に設定します。さまざまな環境でさまざまなデータセットを使用しなければならない場合や、データが外部ソースからファイルインポートを介して提供される場合などにこの方法を選択します。

拡張サービス展開用パッケージ

拡張サービス展開用パッケージは、指定したサービス定義とそのコンポーネントを展開します。このパッケージタイプでは、対象サイトでの展開中にサービス定義の関連エンティティを処理する方法のオプションを制御する機能が提供されます。

拡張サービス展開は、基本展開とは次の 2 つの点で大きく異なります。

- 親サービスを選択してそれがバンドルであることを示すことによって、バンドルを構成しているすべてのサービスを自動的に展開できます。
- ユーザは、関連付けられたエンティティが展開時にターゲットサイトに存在しないときに実行するアクションを指定できます。これは、基本サービス展開の動作を上書きするため、関連エンティティが 1 つも存在しない場合、自動的に失敗します。

関連付けられたエンティティがターゲットサイトに存在しないときの拡張サービス展開パッケージにおける Catalog Deployer の動作を制御するオプションについて次に要約を示します。

表 7: 拡張サービス展開用パッケージ

関連付けられた単位	使用可能なオプション	結果
エージェント (Agent)	省略 (Skip) 失敗 (Fail)	エージェントの参照が、提供タスクまたは承認タスクから削除されます 展開が失敗します
Eメールテンプレート (Email Template)	作成 (Create)	対象サイトに空のテンプレートが作成されます。
	省略 (Skip) 失敗 (Fail)	テンプレートの参照が、提供タスクまたは承認タスクから削除されます 展開が失敗します
役職 (Functional Position)	省略 (Skip) 失敗 (Fail)	役職との関連付けが削除されます。割り当てはデフォルトサービス キューに入ります 展開が失敗します

関連付けられた単位	使用可能なオプション	結果
グループ	省略 (Skip)	グループとの関連付けが削除されます。割り当てはデフォルトサービス キューに入ります
	失敗 (Fail)	展開が失敗します
組織	省略 (Skip)	タスクがデフォルト サービス キューに割り当てられます
	失敗 (Fail)	展開が失敗します
人員 (People)	省略 (Skip)	タスクがデフォルト サービス キューに割り当てられます
	失敗 (Fail)	展開が失敗します
キュー	作成 (Create)	キューは、対象サイトにその キューの組織が存在する場合のみ作成されます。存在しない場合、展開が失敗します
	省略 (Skip)	タスクがデフォルト サービス キューに割り当てられます
ロール	失敗 (Fail)	展開が失敗します
	省略 (Skip)	タスクがデフォルト サービス キューに割り当てられます
	失敗 (Fail)	展開が失敗します

サービスの定義を実質的に変更するので、[省略 (Skip)] オプションおよび[作成 (Create)] オプション（選択したエンティティ タイプにのみ使用可能）は、「クイック アンド ダーティ」展開にのみ有用と見なすべきです。サービスのフォームコンポーネントはそのまま転送されますが、提供計画に大幅な変更をもたらす可能性があります。

カスタム展開用パッケージ

カスタム展開用パッケージを使用すれば、展開中に、エンティティを個別に選択して、関連エンティティを処理する方法に関するオプションを制御することができます。カスタム展開は、人員、キュー、ロール、組織単位、エージェント、サービス項目、標準、課金レート、アカウント定義、契約テンプレート、カテゴリ、グループなどの組織エンティティ、役職、電子メールテン

プレートに使用されるのが一般的です。カスタムパッケージにはサービスを含めることができます。

サービス項目と標準展開の振る舞いを制御するよりきめの細かいオプションが用意されています。オプションで、サービス項目インスタンスと標準エントリを展開の一部として含めることができます。ただし、このオプションはパッケージ内のすべてのサービス項目タイプと標準に適用されます。一部の標準またはサービス項目のエントリだけを含めて他のエントリは含めないようにするには、それらを別々のパッケージに分離する必要があります。

インポート中に、[需要管理 (Demand Management)] > [課金レート (Billing Rate)] > [課金レート定義 (Billing Rate Definition)] で単位レート フラグが設定 (ソース XML) された場合は、Catalog Deployer でカスタムパッケージを作成するときに請求レートとして「マージ」オプションが選択されます。レート表がターゲット内に存在する場合は、「マージ」オプションが無視され、ターゲット内の既存のデータ レコードを削除することによって「上書き」として実行されます。

「単位レート」フラグが設定されたレート表に対して、次の検証を実施する必要があります。これは、需要管理、NSAPI、および REX フローにも実施する必要があります。

- 1 つの属性だけを課金可能にする必要があります。
- 1 つのデータ レコードだけがレート表データ内に存在する必要があります。
- 単一の請求可能な属性は、データ型「Number」だけにする必要があります (Integer/Long/Double/Money)。
- データ タブ内に複数のレコードが存在する場合は、ユーザが最初のタブの「単位レート」フラグをチェックできないようにしてください。

カスタム展開を使用してカテゴリを展開すると、ソース サイトのカテゴリ構造のすべてまたは一部を、ターゲットサイトに再作成できます。選択したカテゴリとそれらのサブカテゴリが展開され、カテゴリが関連付けられているサービスとカテゴリの関係が、ソース サイトでの関係と一致するように再確立されます。これは、基本展開パッケージで提供される機能を補完します。基本展開パッケージでは、展開しているサービスに関連付けられているカテゴリのみがパッケージに含まれます。

1 つのパッケージには、エンティティ間に相互依存性がある 2 つ以上のエンティティ タイプを含めることができます。たとえば、人員は既存の組織に関連付ける必要があります。Catalog Deployer は正しい順序で自動的にエンティティを展開します。そのため、これらの相互依存関係は維持できます。

さらに、同じタイプのエンティティ間に相互依存関係が存在することもあり、ある人員が別の人員をスーパーバイザとして指名したり、組織階層が存在したりします。このような場合も、Catalog Deployer によってエンティティは正しい順序で（たとえば、スーパーバイザ エンティティが最初に）展開されるため、関係を再設定できます。

展開する各エンティティ（役職以外）に対して、パッケージに含まれる新しい定義で対象サイトのエンティティ定義を上書きする、または展開を省略するオプションを選択できます。

Catalog Deployer はターゲットサイトにエンティティが存在するかどうかを調べるのみで、（たとえば、ソースがターゲットよりも新しいかどうかを確認するために）コンテンツの検査を行わないでの、エンティティ展開をスキップすることは危険です。[スキップ (Skip)] オプションは、

すでに存在するエンティティの再インストールを行わないでパフォーマンスを最適化できますが、その使用は限定された状況（たとえば、エンティティが存在するか不明であるターゲットサイトに大きいパッケージを初めて展開する場合など）で行ってください。いずれかの面で展開に失敗する（たとえば、ホーム OU がターゲットサイトと現在の展開パッケージに存在しない人を展開しようとした）場合は、その欠落を修正してパッケージを再展開できます。まだターゲットサイトに存在しないパッケージコンテンツのみが展開されます。

エンティティタイプ（電子メールテンプレート、役職、およびエージェント以外）ごとに、ターゲットサイトの他の関連エンティティとの関係の再設定に関して Catalog Deployer の動作を指定するオプションもあります。

- ソースサイトの関係を複製せずにエンティティ定義に対する変更のみを展開するには、[含めない (Do not include)] を使用します。このオプションは、たとえば、新しい組織のみ展開し、この組織と多数の人々（一部の人は対象サイトに存在しない可能性がある）との関係は展開しない場合に選択します。
- 一般的にはソースサイトの関連付けを含めることは必要です。このオプションを選択すると、対象サイトでソースサイト内の関係が複製されます。[スキップ (Skip)] オプションを使用すると、一部の関連付けられたエンティティがターゲットサイトに存在しない場合に展開を続行できます。たとえば、まだ開発中で展開されていないサービスにキーを割り当てていることがあります。[スキップ (Skip)] オプションを使用する場合は、予想される関連付けがスキップされた状態にならないように、ログを注意深く読んでください。

展開中にソースサイトからターゲットサイトにインポートされたデータもマージすることができます。このオプションは、サービス項目定義、標準定義、課金レート定義、および同意テンプレートに使用できます。



(注)

ポータルページとポートレット権限はロールを使って展開できます。展開は、ポータルページとポートレットが Portal Designer 経由でインポートされてから実施する必要があります。

展開用パッケージの作成と展開

展開用パッケージは Catalog Deployer で次のフローに従います。

ステップ1 展開用パッケージを作成します。Catalog Deployer にアクセスして使用する権限を付与されたユーザは、展開用パッケージを作成し、パッケージに含めるエンティティを選択できます。[展開用パッケージの作成](#)、(25 ページ) を参照してください。

ステップ2 パッケージをアセンブリします。基盤となるコンテンツが展開準備完了と見なされると、[アクション (Action)] ペインの [パッケージのアセンブリ (Assemble Package)] ボタンをクリックすると、展開用パッケージがアセンブリされます。この時点で、Catalog Deployer はコンテンツを抽出し、展開用パッケージの一部として保存されるコピーを作成します。パッケージのアセンブリが行われると、パッケージに含

まれるエンティティへの変更は、パッケージが再アセンブリされない限りキャプチャされません。展開用パッケージの組み立て、(27 ページ) を参照してください。

- ステップ3** パッケージを送信します。アセンブリされたパッケージは、展開のために対象サイトに送信されます。パッケージは、Catalog Deployer 経由で送信できます。または、ソースサイトおよびターゲットサイト間に直接接続がなければ、このステップをエクスポートおよびインポートプロセスを使用して「オフライン」で実行する場合があります。展開用パッケージの送信、(27 ページ) を参照してください。
- ステップ4** パッケージを展開します。パッケージを展開する権限を持つ担当者は、対象サイトで受信されたパッケージを選択し、展開を実行します。パッケージの展開、(30 ページ) を参照してください。
- ステップ5** ビュー レコードログファイルに、各サイトの Catalog Deployer 内で発生するすべてのアクティビティが記録されます。ログファイル、(33 ページ) を参照してください。
- ステップ6** 展開中にサポートされないエンティティについては、サポートされていないエンティティ、(6 ページ) を参照してください。



(注) パッケージ定義が変更された場合のみ、パッケージを保存する必要があります。パッケージのコンテンツは自動的に保存されます。

展開用パッケージの作成

展開用パッケージを作成するには、次の手順を実行します。

- ステップ1** [表示と検索 (View and Search)] ペインで [アクション (Action)] > [新規展開用パッケージ (New Deployment Package)] を選択します。
- ステップ2** [新規展開用パッケージ (New Deployment Package)] ウィンドウで、次の詳細情報を入力します。
- パッケージ名 (Package Name)
 - 説明
 - パッケージタイプ (Package Type)
- ステップ3** [保存 (Save)] をクリックします。



(注)

- 名前からユーザがパッケージコンテンツを推測できるようにするために、パッケージの命名標準を策定してください。たとえば、パッケージのタイプ、ソースサイト、コンテンツの説明、およびビルド番号とパッケージが作成された日付の情報を組み合わせて名前を構成できます。ソースサイトまたはターゲットサイトで同じ名前の2つのパッケージを作成することはできません。
- 特殊文字（たとえば、“\/*?”<0[]”など）を使用するパッケージ名を入力することはできません。スペースは使用できますが、ログ XML 出力ファイルでは下線（_）に置き換えられます。
- パッケージの説明は必須です。パッケージ名と説明は両方とも、パッケージが作成された後で変更できます。
- 必要に応じて、[表示と検索 (View and Search)] ペインまたは [コンテンツ (Content)] ペインでパッケージ名をクリックし、[パッケージ名 (Package Name)] と [説明 (Description)] フィールドを確認します。検索結果には、入力されたエンティティタイプまたは検索する値がパッケージ内のプライマリ エンティティだったパッケージのみが返されます。プライマリ エンティティとは、パッケージに含めるために選択されたエンティティ（サービス定義、組織単位、グループ、キー、人員、役職、ロール、カテゴリ、電子メールテンプレート）のことです。
- パッケージタイプはパッケージをいったん作成すると変更できません。パッケージタイプの詳細については、[ログ ファイル](#)、(33 ページ) を参照してください。

展開用パッケージへのコンテンツの追加

パッケージにはその作成時にステータス [未送信 (Not Transmitted)] が割り当てられます。[表示と検索 (View and Search)] ペイン内でこれをクリックすると、そのコンテンツが [内容 (Contents)] ペインに表示されます。

コンテンツをパッケージに追加するには、次の手順を実行します。

ステップ1 [内容 (Content)] ペインで、[追加 (Add)] ドロップダウン メニューをクリックし、そのパッケージのタイプで使用可能なエンティティのリストから、追加するエンティティ タイプを選択します。
[検索 (Search)] ダイアログ ボックスが表示され、コンテンツを検索および選択できます。

ステップ2 チェック ボックスをオンにしてコンテンツを選択します。

ステップ3 [追加 (Add)] をクリックして、選択したコンテンツを追加します。

コンテンツが [内容 (Content)] ペインに表示され、自動的に保存されます。エンティティを削除する必要がある場合は、チェックボックスをオンにして、[削除 (Remove)] と [はい (Yes)] を順にクリックします。コンテンツは、パッケージが送信されるまで、いつでも追加および削除できます。アセンブリされた後にコンテンツが変更された場合、パッケージを再アセンブリする必要があります。

パッケージ内容のレビュー

基本および拡張サービス展開用パッケージでは、サービス定義のレビューが可能です。

すでにパッケージに含まれているサービスの場合

パッケージの[内容 (Content)]ペインに移動し、レビューするサービスのチェックボックスをオンにして選択し、[レビュー (Preview)]をクリックします。

サービスのレビューはサービス定義の概要およびサービスフォームのレンダリングで構成されます。すべてのエンティティ参照がレビュー最後の[追加内容 (Additional Content)]セクションにまとめられています。これは、サービスを展開する前に対象環境にこれらのエンティティが存在することを確認するうえで役に立ちます。

展開用パッケージの組み立て

[アクション (Action)]ペインで、[パッケージの組み立て (Assemble Package)]をクリックしてパッケージを組み立てます。[OK]をクリックして、パッケージのアセンブリが成功したことを確認します。パッケージを組み立てた後、そのステータスは[未送信 (Not Transmitted)]のままでです。いずれかのコンポーネントの定義が変更された場合、または展開するエンティティの追加または削除を行った場合には再度組み立てることができます。パッケージ組み立てのログエントリには、指定されたプライマリエンティティと一緒に展開されるコンポーネントエンティティの両方が含まれます。[アクション (Action)]ペインの[履歴 (History)]セクションで[ログの表示 (View Log)]リンクをクリックすると、ログが表示されます。

展開用パッケージの送信

コンテンツは、次の場合に別のサイトへ直接送信できます。

- ターゲットサイトは [管理 (Administration)] > [設定 (Settings)] > [Entity Homes] で定義されています。
- ターゲットサイトに対応するデータソースは、アプリケーションサーバコンソールの JDBC データソースページで定義されています。

パッケージを送信するには、次の手順を実行します。

-
- ステップ1** [表示と検索 (View and Search)] ペインで、[表示 (View)] ドロップダウンメニューを使用してステータスが [未送信 (Not Transmitted)] のパッケージを表示します。
 - ステップ2** リストでパッケージを探します。パッケージ名をクリックすると、そのパッケージの情報が他のペインに表示されます。
 - ステップ3** パッケージがアセンブリされていない場合、[アクション (Action)] ペインで、[パッケージのアセンブリ (Assemble Package)] をクリックします。[OK] をクリックして、パッケージのアセンブリが成功したことを確認します。
 - ステップ4** [アクション (Action)] ペインで、[送信 (Transmit)] をクリックします。
 - ステップ5** ドロップダウンメニューからターゲットサイトを選択します。
(注) サイト管理者が設定したデータソースのみを選択できます。手順については、『Cisco Prime Service Catalog Designer Guide』を参照してください。サイトが一覧表示されない場合は、Catalog Deployer のエクスポート/インポート機能のみを使用するようにサイトが設定されている可能性があります。
 - ステップ6** [今すぐ送信 (Transmit Now)] をクリックします。
 - ステップ7** [OK] をクリックして、送信が成功したことを確認します。
展開パッケージがターゲットサイトに送信されます。ソースサイトの現在のパッケージのステータスは [送信済み (Transmitted)] に変更されます。送信したパッケージは、他のサイトに送信したり、エクスポートしたりできます。
-

展開用パッケージのエクスポート

パッケージを送信する代わりに、パッケージをエクスポートすることができます。パッケージをエクスポートすると、アセンブリされたパッケージのコンテンツで構成される XML ファイルが作成されます。次に、エクスポートファイルは、対象サイトにインポートされ展開されます。

パッケージをエクスポートすると、パッケージがテキストで表現されます。エクスポートしたパッケージは、オフラインのアーカイブとして使用したり、会社のソースコード制御システムにチェックインしたりできます。

パッケージをエクスポートするには、次の手順を実行します。

-
- ステップ1** [表示と検索 (View and Search)] ペインで、[表示 (View)] ドロップダウンメニューを使用して [未送信 (Not Transmitted)] または [送信済み (Transmitted)] ステータスのパッケージを表示します。
(注) 展開用に受信されたパッケージまたは展開されたパッケージは、エクスポートできません。

- ステップ2** リストでパッケージを探します。パッケージ名をクリックすると、そのパッケージの情報が他のペインに表示されます。
- ステップ3** パッケージがアセンブリされていない場合、[アクション (Action)]ペインで、[パッケージのアセンブリ (Assemble Package)]をクリックします。[OK]をクリックして、パッケージのアセンブリが成功したことを確認します。
- ステップ4** [アクション (Action)]ペインで、[エクスポート (Export)]をクリックします。
次の例に示されているように、[エクスポート (Export)]ダイアログボックスが表示されます。
- ステップ5** このダイアログボックスで、ファイル名のリンク（上記の例では、ファイル名のリンクは **Basic Service Package02** です）をクリックします。
[ファイルのダウンロード (File Download)]ダイアログボックスが表示されます。
- ステップ6** [保存 (Save)]をクリックします。
[名前を付けて保存 (Save As)]ダイアログボックスが表示されます。
- ステップ7** 必要に応じて、ファイルの名前を変更し、目的の宛先を選択します。
宛先は一般的には共有ドライブ上（Catalog Deployerの機能ですべてのユーザがアクセス可能）にします。
パッケージのトラッキングを円滑に行うには、ディレクトリの命名およびサブディレクトリの構造の標準を確立する必要があります。たとえば、同じ変更要求の一部として展開されるすべてのパッケージに対して、新しいサブディレクトリを作成するなどです。
- ステップ8** [保存 (Save)]をクリックします。
- ステップ9** [Close (閉じる)]をクリックして、[エクスポート (Export)]ダイアログボックスを閉じます。

セキュリティやその他の理由でソースから対象サイトへパッケージの直接送信が許可されていない場合、パッケージの送信の代わりにエクスポート/インポートを使用できます。結果は同じです。つまり、パッケージは [展開用に受信 (Received for Deployment)] ステータスでターゲットサイトに作成され、続いて展開できます。

展開用パッケージのインポート

エクスポートしたパッケージは、対象システムにインポートする必要があります。

パッケージをインポートするには、次の手順を実行します。

-
- ステップ1** [表示と検索 (View and Search)]ペインで、[アクション (Action)]>[インポート (Import)]を選択します。
[インポート (Import)]ダイアログボックスが表示され、インポートするファイルを探すように求められます。

ステップ2 [参照 (Browse)] をクリックします。

ステップ3 パッケージファイルを見つけて、選択します。

ステップ4 [開く (Open)] をクリックします。

ステップ5 [インポート (Import)] をクリックします。

展開用パッケージがインポートされ、[展開用に受信 (Received for Deployment)] ステータスが割り当てられて、開かれます。これで展開できるようになりました。

ステップ6 [インポート (Import)] ダイアログボックスを閉じます。



(注) 展開用パッケージを最初にエクスポートしたソースサイトに再インポートするには、まずソースサイトから同一のパッケージを削除する必要があります。ファイル名が変更された場合でも、アプリケーションは重複ファイルを認識します。

パッケージの展開

対象サイトに送信またはインポートしたパッケージを展開するには、以下の手順を実行します。

ステップ1 [表示と検索 (View and Search)] ペインで、[表示 (View)] ドロップダウンメニューを使用して [展開用に受信 (Received for Deployment)] ステータスのパッケージを表示します。

ステップ2 リストでパッケージを探します。パッケージ名をクリックすると、そのパッケージの情報が他のペインに表示されます。

(注) 展開オプションとソースサイトで選択した関連付けられたエンティティのルールに従って展開が実行されます。対象サイトのユーザは、タブをクリックして選択したエンティティを表示できますが、いかなる方法でもパッケージを変更することはできません。

ステップ3 [アクション (Action)] ペインで、[展開 (Deploy)] をクリックします。

ステップ4 [OK] をクリックして、展開が正常に完了したことを確認します。

展開が正常に完了すると、パッケージのステータスは [展開済み (Deployed)] に変更されます。

展開するパッケージのアセンブリされたコンテンツは、対象サイトのリリース レベルと一致する必要があります。Catalog Deployerは、パッケージがアセンブリされたアプリケーションバージョンを追跡し、異なるバージョン間での展開を許可しません。

基本または拡張サービスパッケージには、展開されるサービスで使用するすべてのコンポーネント設計要素が含まれます。ただし、データベースに格納されていない設計コンポーネントは展開用パッケージに含まれず、別に展開する必要があります。これらの要素には次のものが含まれます。

- 任意の ISF スクリプトによって参照される JavaScript ライブラリ

- 環境に追加され、データ取得ルールまたはオプションリストによって参照されるデータソース

展開は、原則として、展開が行われたときにインフライトであった要求によって使用されるサービスおよびコンポーネントの定義に影響を与えません。ただし、要求プロセスの動的性質により、以前に送信された要求は次のものによる影響を受けます。

- ルールまたは JavaScript 関数およびライブラリのコンテンツに対する変更
- 最終的な承認ステップにまだ合格していないサービス要求の提供計画に対する変更

展開スケジュールおよびサービスの設計者は、サービス定義の以前のバージョンに基づいたインフライトの要求が上記の要素の変更による影響を受けることを考慮する必要があります。

複数のパッケージを送信して展開する

1つの機能で複数のパッケージを送信して展開できます。Catalog Deployer は順番に各パッケージを処理し、それぞれのステータスを提供します。この手順により、わずかなユーザ操作で大量のパッケージを展開することが容易になります。

複数のパッケージを送信するには、次の手順を実行します。

-
- ステップ1** [表示と検索 (View and Search)] ペインで、[アクション (Action)] > [複数パッケージの送信 (Transmit Multiple Packages)] を選択します。
 - ステップ2** [送信するパッケージの追加 (Add Packages to Transmit)] をクリックして [検索 (Search)] ダイアログボックスを開きます。ここで、送信するパッケージを検索して選択できます。[検索 (Search)] ダイアログボックスでは、ステータスが [未送信 (Not Transmitted)] (組み立て済み) および [送信済み (Transmitted)] のパッケージのみを選択できます。
 - ステップ3** 必要なパッケージのチェック ボックスをオンにして選択します。
 - ステップ4** [追加 (Add)] をクリックして、選択したパッケージを追加します。パッケージが [パッケージ (Packages)] フォルダの下にアルファベット順に表示されます。選択したすべてのパッケージが送信されます。パッケージを削除する必要がある場合は、そのチェック ボックスをオンにして、[削除 (Remove)]、[はい (Yes)] の順にクリックします。
 - ステップ5** [ターゲットサイトを選択 (Select a target site)] フォルダの下で、ターゲットサイトのチェック ボックスをオンにして選択します。
 - ステップ6** [送信 (Transmit)] をクリックして送信を開始します。パッケージが一覧表示された順序 (アルファベット順) で送信されます。送信プロセスが完了すると、ステータス メッセージが表示され、各パッケージの送信の成否が示されます (送信失敗の最も一般的な理由は、ソースサイトとターゲットサイト間のバージョンの不一致です)。各パッケージのログファイルも更新されます。
-

同様のインターフェイスによる複数パッケージの展開

同様のインターフェイスを使用して複数のパッケージを展開できます。

-
- ステップ1** [表示と検索 (View and Search)]ペインで、[アクション (Action)]>[複数パッケージの展開 (Deploy Multiple Packages)]を選択します。
 - ステップ2** [複数パッケージの展開 (Deploy Multiple Packages)]ウィンドウの[展開するパッケージの追加 (Add Packages to Deploy)]をクリックして [検索 (Search)]ダイアログボックスを開きます。ここで、展開するパッケージを検索して選択できます。ステータスが [展開用に受信済み (Received for Deployment)] または [展開済み (Deployed)] の展開パッケージを選択できます。
 - ステップ3** 展開するパッケージのチェックボックスをオンにして選択します。
 - ステップ4** [追加 (Add)]をクリックして、選択したパッケージを追加します。
パッケージが [パッケージ (Packages)] フォルダの下にアルファベット順に表示されます。選択したすべてのパッケージが展開されます。パッケージを削除する必要がある場合は、そのチェックボックスをオンにして、[削除 (Remove)]、[はい (Yes)] の順にクリックします。
 - ステップ5** [展開 (Deploy)]をクリックして展開を開始します。
パッケージが一覧表示された順序 (アルファベット順) で展開されます。複数送信と同様に、複数展開でも各パッケージの展開の成否が示されます。この情報は、パッケージのログファイルでも利用できます。
-

展開用パッケージのクローズと再オープン

どの状態の展開用パッケージも、それが作成されたサイトの [表示と検索 (View and Search)] ペインでそれをクリックして、[アクション (Action)] ペインで [クローズ済みとしてマーク (Mark as Closed)] をクリックすることによってクローズすることができます。パッケージをクローズすると、そのステータスが [クローズ (Closed)] に変わるだけで、そのパッケージは「クローズ」パッケージのビューにしか表示されなくなります。これにより、探す対象のアクティブパッケージが少なくなるため、他のパッケージでの作業が容易になります。

クローズしたパッケージはいつでも（他のサイトに送信する必要がある、またはコンテンツをエクスポートしたい場合など）再オープンすることができます。パッケージを再オープンするには、[表示と検索 (View and Search)] ペインでそれをクリックして、[アクション (Action)] ペインで [再オープン (Reopen)] をクリックします。

展開用パッケージの削除

パッケージを削除するには、[表示と検索 (View and Search)] ペインでそれをクリックして、[アクション (Action)] ペインで [削除 (Delete)] をクリックします。削除を確定するには、[はい (Yes)] をクリックします。展開用パッケージを削除すると、パッケージと展開履歴が現在のサイトから完全に削除されます。パッケージ内容は圧縮されますが、パッケージコンポーネントを表すために必要な XML は非常に大きくなることがあります。したがって、パッケージを削除すると、リポジトリ/データベースで使用可能なスペースを取り戻せます。パッケージを過去にエク

サポートしたことがある場合は、パッケージの内容を復旧できます。ただし、現在のサイトのパッケージの完全な展開履歴を復旧することはできません。

展開用パッケージのコピー

多くの場合、ビルドプロセスの一部として同じエンティティを複数回展開する必要があります。たとえば、1つ以上のサービスを開発環境からテスト環境に展開したとします。次に、修復が必要ないいくつかの不具合をテスターが見つけると、開発環境でサービス定義が修復されて再展開されます。次に、本番環境に展開する前にテスト環境で修正が確認されます。

展開用パッケージのコピー機能はこのワークフローを容易にします。目的のコンテンツを持つ最初のパッケージが作成されたら、このパッケージをコピーできます。

パッケージをコピーするには、以下を実行します。

ステップ1 [表示と検索 (View and Search)] ペインでパッケージを選択します。

ステップ2 [アクション (Action)] ペインで、[コピー (Copy)] をクリックします。
[パッケージのコピー (Copy Package)] ダイアログボックスが表示されます。

ステップ3 パッケージの名前を変更します。

ステップ4 [パッケージのコピー (Copy Package)] をクリックします。

ステップ5 [OK] をクリックします。

新しいパッケージが作成され、[未送信 (Not transmitted)] のステータスで開かれます。このパッケージには、アセンブリされたパッケージコンテンツではなく、[コンテンツ (Content)] ペインで指定されたエンティティが含まれます。次に、必要に応じて、パッケージを再アセンブリできます。

ログ ファイル

[アクション (Action)] ペインの [履歴 (History)] セクションの [ログの表示 (View Log)] リンクに、Catalog Deployer によるアクションの詳細が表示されます。[ログの表示 (View Log)] リンクをクリックすると、[ログの表示 (View Log)] タブが開いて、[ログの詳細 (Log Details)] と [XML出力 (XML Output)] というサブタブが表示されます。ログには、パッケージが受けるすべてのアクティビティが記録されます。アセンブリログには、パッケージに含まれるすべてのエンティティが表示されます。展開ログには、ターゲットサイトで正常に抽出されたすべてのエンティティが表示されます。パッケージの展開に失敗した場合、エラーも表示されます。

ログには、すべてのパッケージの詳細（名前、説明）、組み立てまたは展開の時刻の日時スタンプ、および含まれているすべてのエンティティが含まれます。

既知のエラーおよび欠落

Catalog Deployer はエンティティの名前変更をサポートしていません。ソースサイトでエンティティ名を変更し、そのエンティティを展開しようとしたり、名前を変更したエンティティを参照するサービスを展開したりすると、CatalogDeployer は正常に機能しません。関連エンティティとの関連付けを設定（再設定）するために、CatalogDeployer はそのエンティティをターゲットサイトで名前により見つけようとします。展開の目的上、エンティティ「名」は、人間（ログイン名によって識別）および組織（組織タイプと名前の組み合わせによって識別）を除くすべてのエンティティの名前属性です。結果は次のようにになります。

- サービス定義のコンポーネントエンティティ（ディクショナリ、フォームコンポーネント）では、新しいエンティティが作成され、このエンティティとの関連付けが展開対象のサービスに設定されます。
- 展開対象のプライマリエンティティでは、新しいエンティティが新しい名前で作成されます。
- 関連付けられた（ダイレクトリエンティティおよび電子メールテンプレート）エンティティでは、選択したパッケージタイプおよび展開オプションに応じて、展開が失敗する場合、またはエンティティの関連付けがスキップされる場合があります。

回避策は次のとおりです。

- 該当する場合は、影響を受けるエンティティのサイト保護をターゲットサイトでオフにします。
- 名前がソースサイトの名前と一致するようにターゲットサイトのエンティティの名前を変更します。
- サイト保護をオンに戻します。
- エンティティを展開します。

展開プロセスの間は、エンティティ名の変更状況を注意深くモニタリングし、前述の説明のように、ターゲットサイトにエンティティが適用されることを確認する必要があります。その後、ターゲットサイトへのエンティティの展開を試みます。

目標テーブルと課金レートテーブルで参照されている測定単位は展開パッケージで収集されません。ターゲットサイトで Administration モジュールを介してサービスまたは課金レートに対して設定された新しい測定単位を作成してから、それらを参照するパッケージを展開する必要があります。

サンプルの展開シナリオ

ここでは、頻繁に行われるいくつかの開発シナリオおよび展開シナリオで使用する推奨手順の詳細を示します。

- 初期展開、(35 ページ)

- ・各サーバへのエンティティの配置、(36 ページ)
- ・新しいキューを使用するためのサービスの展開、(38 ページ)
- ・新しい電子メールテンプレートを使用するサービスの展開、(39 ページ)
- ・キューおよびサービス チームの名前変更、(40 ページ)

初期展開

新しい Service Catalog サイトを作成する必要があります。1 つのオプションは、既存の Service Catalog データベースをサイトにコピーし、そのデータベースを使用するようにインストールおよび設定を調整する方法です（この手順については、このガイド (http://www.cisco.com/en/US/products/ps13206/prod_technical_reference_list.html) を参照してください）。また、データベースの初期設定を行い、Catalog Deployer を使用して新しいサイトの生成に必要なすべてのエンティティを展開する方法もあります。

予備設定の実行

Catalog Deployer は、Service Catalog の進展に伴い一般に変化しない Service Catalog の設定面は展開しません。したがって、初期サイトに対して行ったように、これらの要素を定義する必要があります。

- ・ディレクトリ統合、承認、およびエンティティ ホームなど、Administration モジュールの設定を再入力します。これらの設定は開発の設定と同じにすべきですが、環境固有の設定項目は除きます（たとえば、開発とテストで LDAP ディレクトリが別の場合があります）。
- ・インストール手順にすべての Service Link カスタム アダプタが含まれていたこと、および Service Catalog アダプタに加えたすべての変更が再度適用されることを確認します。
- ・エージェントおよび変換を再入力します（既知のエラーおよび欠落）。

サービスの基盤エンティティの展開

Organization Designer によって保守されるエンティティ（人員、組織、グループ、ロール、および役職）は、サービス定義がこれらのエンティティを参照できるように、その前にサイトに存在していかなければなりません。そのため、これらのエンティティはすべて、サービスカタログを展開する前に展開する必要があります。また、これらのエンティティは、1つ以上のカスタム展開パッケージに含める必要があります。

さらに、カスタム電子メールテンプレートをすべて展開する必要があります。電子メールテンプレートは、それを使用するサービスが展開される前ならいつでも展開できます。

サービスの展開

基盤エンティティを展開したら、サービスを展開できます。初期展開では、パッケージサイズを管理できるサイズにし（つまり、サービス グループごとにグループ化し）、ログを慎重に確認し

ます。すべての関連エンティティが事前に展開されているはずなので、基本展開を使用する必要があります。

各サーバへのエンティティの配置

これで2つ（以上の）サイトが稼働している状態です。本番サイトのサービスはすべて、開発サイトのサービスと同一である必要があります。（開発サイトには初期展開の一部である他のサービスが存在することもありますが、それは問題ありません。）どの環境で、グループ、ロール、および組織のエンティティ定義とメンバーシップを定義するのかが問題です。すべてのケースにおいて、エンティティ保護レベルは、ホーム以外の環境にあるエンティティに対するメンテナンスは回避するように設定されます。

本番にのみ存在するディレクトリ関連エンティティ

最も単純なアプローチは、サービス定義を構成するすべてのエンティティのホームを開発インスタンスにし、人と組織のすべての関連エンティティのホームを本番インスタンスにする方法です。

これによって、Catalog Deployer の使用およびそれが展開するエンティティの保守が非常に簡単にになります。つまり、すべてのエンティティのすべての面が常に1つのサイトのみで保守されます。ただし、ディレクトリ関連エンティティの作成およびテストのワークフローは複雑になります。

- 1 新しいサービスを展開する前に、提供計画またはサービスの他の領域で参照されるキューリー、組織、グループ、ロール、および機能ユニットの一覧を作成します。
- 2 これらすべての関連エンティティが開発インスタンスにすでに存在する場合、これらのエンティティは本番サイトにも存在することになります（本番サイトがホームであるため）。したがって、サービスの作成およびテストを進めることができます。
- 3 これらのディレクトリ関連エンティティの中でまだ存在しないものがある場合は、本番サイトにログインして作成します。エンティティ定義、およびそのメンバーシップとロールを設定します。
- 4 本番サイトにログインして、新しいエンティティとその関連エンティティを含むカスタム展開パッケージを作成します。ターゲットインスタンスにまだ関連付けがないため、このパッケージでは新しいエンティティに対して [ソース関連付けの使用 (Use Source associations)] オプションを使用する必要があります。
- 5 カスタムパッケージを開発サイトに展開します。
- 6 展開した新しいエンティティを使用して、サービスを作成します。
- 7 サービスがテスト可能な状態になったら、対象サービスを含む基本サービスパッケージを作成します。
- 8 まず、ディレクトリエンティティを含んでいるカスタムパッケージをテストサイトに展開します。次に、基本サービスパッケージを展開します。
- 9 サービスがテストにパスしたら、同じ基本サービスパッケージを開発サイトから本番サイトに展開します。

このシナリオでは、ディレクトリ関連エンティティに関するすべての作業（サービス定義内の実際のエンティティ参照を除く）は1か所、つまり本番環境で行われます。これは推奨されるプ

ロセスです。つまり、すべての作業が 1か所で行われるため、確認とモニタリングが容易になります。ただし、このプロセスでは次のような不都合が生じことがあります。

- 開発プロセスにステップを追加することになります。エンティティ定義を最初に（または2度目あるいは3度目に）訂正しなかった場合は、本番に戻ってエンティティを修正し、再パッケージ、再展開および再テストを行わなければならなくなります。再パッケージを行う場合、プライマリ エンティティはターゲット サイトにのみ存在するサービスにすでに関連付けられているため、プライマリ エンティティに対して [ターゲット関連付けの使用 (Use Target Associations)] オプションを使用する必要があります。
- 考えられる別の難点として、本番環境で広範な開発作業を行っている場合があり、これは一部の組織ではセキュリティ リスクと見なされます。

本番と開発の両方に存在するディレクトリ関連エンティティ

代替アプローチは、サービス定義を構成するすべてのエンティティのホームを開発インスタンスにし（これは変更不可）、人員と組織の関連エンティティのホームを本番インスタンスと開発インスタンスの両方にすることです。

組織、グループ、ロール、人員などのエンティティのホームを両方のインスタンスにすることは、次のような利点があります。

- これらのエンティティの定義を論理的な場所、つまり開発環境で操作できます。作業はいくつかの手順を伴うことがあります。たとえば、カスタム ロールの定義を正常に行う場合です。開発、展開、およびテストを繰り返す必要はありません。単に開発およびテストを行って、終了したら展開するだけです。
- 論理的な場所、つまり本番環境の組織またはロールにメンバを割り当てることができ、そこではすべての人員情報がディレクトリ統合により自動的にリフレッシュされ、すべての Service Catalog ユーザが表現されることが保証されます。

組織、グループ、ロール、人員などのエンティティのホームを両方のインスタンスにすることは、次のような欠点があります。

- ロールベース アクセス コントロールは現在、ある特定のエンティティ タイプへのアクセスを許可します。それ以上の細分化はできません。たとえば、開発環境でグループ（およびそのロール）のみを定義することを設計者に許可できるのではなく、本番環境でのグループメンバーシップの割り当てを許可するだけです。
- メンテナンスは複数の環境で発生しているため、パッケージの展開時には注意を払う必要があります。エンティティに対するソースとターゲットの適切な関連付けが維持されるようにする必要があります。

たとえば、メンバがさまざまな組織の人々で構成されるグループを作成しているとします。次の項では、グループにメンバを割り当てる方法について説明します。

本番と開発の両方に存在する人員とグループ

- 開発でグループを作成し、グループ経由で付与される機能および権限が正しいことを検証するためにソース（テスト）メンバを割り当てます。グループに必要なすべての人員が開発環境にいるとは限らないため、メンバリストは不完全で、場合によっては本番内の人員と互換性がありません（対応するエントリが本番内に存在しない「テスト」人員が何人かいる場合）。
- カスタム展開パッケージを使用して、ソースの関連付けによってグループを本番に展開しますが、参照されるエンティティが存在しなければ関連付けをスキップします。
- Organization Designer を本番で使用して、適切な人員をグループに関連付けます。両方のエンティティのホームが本番であり、「作成、削除、変更」の保護レベルでもエンティティを保守できるため、この関連付けは人員またはグループのいずれか便利な方のページで行うことができます。
- グループのメンバーシップが変化した場合に必要なことは、本番環境で Organization Designer に進み、適切な変更を加えるだけです。さらに、ディレクトリ統合にはある個人がメンバであるグループのリストを動的に指定する機能があります。エンタープライズディレクトリおよび対応するディレクトリマッピングを更新して、この機能を有効にしている場合、手動で更新する必要はありません。

新しいキーを使用するためのサービスの展開

新しいキーを使用するには、新しいサービスを開発するか、既存のサービスを拡張する必要があります。ベストプラクティスに従って、キーは、そのホームになっている対応するサービスチーム組織を持つようにすべきです。

キーおよび組織の「ホーム」を設定した場所に応じて、ここでは 2 つのシナリオが考えられます。それぞれに利点と欠点があります。（これは前項の説明の特殊ケースですが非常に頻繁に起きます）。

本番サイトの組織単位およびキー

これは、すべての作業を本番サイトの組織、人、キーに置く「クリーン」アプローチです。2007 よりも前の Service Catalog バージョンにおいて実行可能な唯一のアプローチだったため、それらのバージョンからアップグレードする人はその使用を継続することを好む場合があります。

- このアプローチのデメリットは、手動作業を本番サイトで行っている（キーおよび OU を作成している）ことで、一部の組織ではセキュリティ上の理由からこれは好まれない場合があります。
- このアプローチの利点は、キーおよび OU でのすべての作業（その定義だけでなく、メンバーの割り当て）が 1 箇所で行われることです。
- 本番サイトで OU およびキーを作成し、適切なロールを割り当てます。
- 本番サイトで、OU にメンバーを割り当てます。

- 新しい OU と キューを本番サイトから開発サイトに展開します。展開用パッケージに、OU 内のすべてのサービス実施者を含めます。既存の人を省略して人を展開するように設定し、展開のパフォーマンスを最適化することができます。
- 開発サイトでサービスを作成します。
- 開発サイトから本番サイトにサービスを展開します。

開発と本番の両方に存在する組織単位とキュー

このシナリオでは、エンティティの定義のホームが開発サイトにあり、そのメンバーシップのホームが本番サイトにあります（もちろん、この分割は Entity Homes によって実施することはできません。したがって、Organization Designer によるメンテナンスを許可するために、両方のサイトがエンティティのホームとして指定されます）。

- 開発サイトで OU と キューを作成し、適切なロールを割り当てます。新しい構成をテストするのに十分な数のメンバーを割り当てます。
- 開発サイトでサービスを作成します。
- 新しい OU および キューを開発から本番に展開する際にソースの関連付けを使用しますが、ターゲット サイトに存在しないものはすべてスキップします（「テスト」人員を除外するため）。
- 本番サイトで Organization Designer を使用（またはディレクトリ統合に依存）してメンバを OU に割り当てます。
- 開発サイトから本番サイトにサービスを展開します。

初期展開後の組織単位とキューの保守

前述の両シナリオでは、初期展開後に初期のキューと OU に対して可能な変更について確認しました。

- OU のメンバーシップの変更は、初期展開に関して処理されます。つまり、OU メンバーシップは本番で保守されます。それらの変更すべてを開発サイトに展開し直すことは重要ではありませんが、行う場合は、OU および 新規または影響を受けた人員のカスタム展開パッケージは、ソースの関連付けを使用して作成し、開発サイトに展開できます。
- OU に割り当てられた権限の変更はエンティティ定義のホーム サイトで適用されます。その後、対象の関連付けを使用して、新しい OU 定義がもう一方のサイトに展開されます。

新しい電子メール テンプレートを使用するサービスの展開

次の 2 つのパッケージが必要です。

- 開発では、電子メール テンプレートを含むカスタム パッケージを作成します。デフォルトの展開オプション（既存のエントリを置き換える）を使用します。
- 開発で、改訂したサービスを含む基本サービス パッケージを作成します。
- 本番サイトにカスタム パッケージを展開します。

- 4 基本サービス パッケージを本番サイトに展開します。

カスタム パッケージが基本サービス パッケージの前に展開されている場合は、それぞれのパッケージが作成時の状態で展開されます。

キューおよびサービス チームの名前変更

特定のキューに割り当てられているタスクを含む、複数のサービスを展開後、キューの名前がサービス実施者に誤解を与えていたため、名前を変更したいというフィードバックを受け取ることがあります。または、会社組織が再編成されていて、組織（サービスチームを含む）の名前を新しい組織を反映したものに変更する必要がある場合があります。

これは既知のエラーおよび欠落、(34 ページ) で説明した「既知のエラーおよび欠落」と衝突します。Catalog Deployer はサイト間でエンティティの名前を一致させて動作するため、名前変更されたエンティティとの一致は不可能です。場合によっては（上述のように）、Catalog Deployer は失敗してエラーを報告します。ただし、別のケースでは、Catalog Deployer はターゲットに新しいエンティティを作成します。名前変更されたキューに関しては、以前に展開されたすべてのサービスが引き続き古いキューを使用します。キューが名前変更され展開された後で展開されたサービスのみが新しいキューを参照します。これは明らかに設計者の意図に反します。

この動作を防ぐ唯一の方法は、名前を変更する必要があるエンティティを注意深くモニタリングおよび制御するプロセスを整備することです。このプロセスは、キューおよび組織のホームが本番サイトだけにあるか、本番サイトと開発サイトの両方にあるかに応じて若干異なりますが、いずれの場合も両方のサイトでエンティティの手動による名前変更（Organization Designer を介して）が必要です。

次のシナリオを考慮する必要があります。

開発と本番の両方に存在するエンティティ

- 1 要件に対応するために、Organization Designer が、通常のメンテナンス手順を使用して、開発インスタンスのキューとサービス チームの名前を変更します。
- 2 管理者は手動で（Organization Designer を介して）本番インスタンスのエンティティの名前を変更します。エンティティのホームは両方のインスタンスにあるため、通常は、エンティティ保護レベルによってこの手順は許可されます。
- 3 Service Designer が、名前を変更したキューとサービス チームを使用してサービスを修正します。処理されたサービスは、標準の手順を使用して展開できます。

本番にしか存在しないエンティティ

- 1 要件に対応するために、Organization Designer が、通常のメンテナンス手順を使用して、本番インスタンスのキューとサービス チームの名前を変更します。
- 2 管理者が開発インスタンスでエンティティ ホーム保護レベルを緩和します。この保護レベルの緩和は、キュー定義やメンバーシップへの手動での変更は通常許可されず、エンティティのホームは本番サイトにあるため必要になります。
- 3 Organization Designer が開発サイトのエンティティの名前を変更します。

- 4 変更が適用された後、管理者がエンティティ保護レベルを元に戻します。
- 5 Service Designer が、名前を変更したキーとサービス チームを使用してサービスを修正します。処理されたサービスは、標準の手順を使用して展開できます。

カテゴリとアイコンの変更

カテゴリは、基本サービスまたは拡張サービスの展開の一環として、コンポーネント エンティティとして展開されます。これは、主な目的が新規または修正したサービス定義を展開し、関連カテゴリも展開されるようにする場合には効果的な方法です。ただし、カテゴリ階層またはカテゴリに関連付けられているイメージが変更されている場合には、サービス パッケージの使用は効果的ではありません。

カスタム展開パッケージには、階層を参照している可能性があるサービスとは無関係に、カテゴリ階層のすべてまたは一部を展開するためのオプションがあります。カテゴリ構造がターゲット環境に展開されるときに、Catalog Deployer は各カテゴリおよびそのサービス間の関連付けを自動的に再設定します。

通常どおり、エンティティ名（このケースではカテゴリ）の単純な変更に対する警告が適用されます。もう適用されないカテゴリがある場合は、そのカテゴリのサービスへの関連付けを削除し、新しいカテゴリを作成して、置き換える必要があります。あるいは、ソースとすべての対象インスタンスでカテゴリの名前を変更します。

- 1 Catalog Designer がカテゴリのホームである開発サイトでカテゴリの名前を変更します。
- 2 管理者が、本番サイトでエンティティ保護機能をオフにします。
- 3 Catalog Designer が本番サイトでカテゴリの名前を変更します。
- 4 変更が適用された後、管理者がエンティティ保護レベルを元に戻します。
- 5 Service Designer が名前を変更したカテゴリを使用してサービスを処理します。処理されたサービスは、標準の手順を使用して展開できます。

Service Catalog アップグレード後のエンティティの名前変更

Service Designer は、アップグレードプロセスによって生成された固有な名前を持つエンティティを認識できます。

- すべてのサービス定義に対応するアクティブ フォーム コンポーネントがあります。アクティブ フォーム コンポーネントの名前は、「UPGD: Form」とその後に続くサービス名で構成されます。
- すべてのフォーム コンポーネントがフォーム コンポーネント グループに割り当てられます。グループの名前は、「UPGD: Form」とその後に続くサービス グループ（フォームに対応するサービス定義がある場所）の名前で構成されます。

Service Designer は、少なくとも、アップグレードプロセスで作成されたアーティファクトの名前を、よりユーザにわかりやすい名前に変更したいと考えます（また、再使用を促進するためにフォーム コンポーネントの構造のリファクタリングも望む場合もありますが、それは別の問題です）。次に示すのは「標準的な」名前変更シナリオです。

- 1 Service Designer がフォーム コンポーネントとフォーム コンポーネント グループの名前を開発サイト（2つのエンティティがホーム）で変更します。
- 2 管理者が、本番サイトでエンティティ保護機能をオフにします。
- 3 Catalog Designer が、本番サイトでフォーム コンポーネントとグループの名前を変更します。
- 4 変更が適用された後、管理者がエンティティ保護レベルを元に戻します。
- 5 Service Designer が、名前を変更したフォーム コンポーネントを使用してサービスに手を加えます。処理されたサービスは、標準の手順を使用して展開できます。

カスタムの役職の追加

開発サイトでOUに関連した役職を定義して、1つまたは2つの組織に対してその役職を保持している個人を指定しています。開発サイトのサービスを、その役職にタスクをルートするように変更しています。この場合、次の手順で、新規および変更済みのエンティティを展開します。

- 1 開発サイトでカスタム パッケージを作成します。
- 2 カスタム パッケージには、作成した役職とその役職に割り当てられている人を含めます。人の展開オプションでは、ソースの関連付けを使用します。
- 3 本番サイトにカスタム パッケージを展開します。
- 4 修正したサービスを含む基本サービス パッケージを作成します。
- 5 基本サービス パッケージを本番サイトに展開します。

ブラウザ キャッシュが有効な状態での環境への展開

Administration 設定で [ブラウザキャッシュ (Browser Cache)] 設定が有効になっている場合は、サービスカタログプレゼンテーション内のアイコンと埋め込み画像に加えられる変更は、ブラウザ キャッシュが削除されるまで有効になりません。アプリケーションユーザにそれぞれのブラウザ キャッシュの削除を要求するには、このガイドの手順に従ってブラウザ キャッシュのバージョンを上げてください。

ブランド化コンテンツライブラリ

シスコでは、特定の製品リリースにおいてブランド化コンテンツライブラリの形でコンテンツを提供します。ご使用のリリースに関してこのようなコンテンツライブラリが使用できるかどうかは、Cisco Technical Assistance Center (TAC) にお問い合わせください。使用可能なライブラリパッケージには次の2種類があります。

- サービス ライブラリ。サービス定義とそのコンポーネント エンティティが含まれます。
- カスタム ライブラリ。サービス ライブラリに関連付けられたエンティティが含まれます。

これらのライブラリ内のサービスは、エンドユーザ管理、アクセス管理、データセンター管理、およびその他の領域で一般的に必要なサービス用のモデルを提供します。ライブラリコンテンツはプレビューすることができ、必要な項目を開発環境に展開することができます。そのため、企業の要件に合わせたサービスの設計、設定、およびカスタマイズを迅速に行うことができます。

カスタムライブラリには、サービス内で参照されるキューやサービスチーム（組織）などのエンティティが含まれています。このライブラリは、オプションで、対応するサービスライブラリと一緒にインストールできます。カスタムライブラリが展開されている場合は、サービスで事前定義の参照が使用されます。カスタムライブラリがインストールされていない場合は、必要に応じて、サービスで「デフォルトサービス キュー」が参照されたり、参照が削除されたりします。サービス設計者には、タスク（提供）計画の設定を完成させる責任があります。

ブランド化ライブラリの展開

一般に、サイト管理者はブランド化ライブラリのコンテンツを管理および展開する責任があります。必要に応じて、ブランド化コンテンツライブラリを展開する権限を持つカスタムロールを作成できます。ロールの作成と割り当ての詳細については、[組織の構築またはオンラインヘルプ](#)を参照してください。

以下の手順を使用して、ブランド化ライブラリを展開します。

-
- ステップ1** ライブラリを入手します。ライブラリと対応するユーザマニュアルはシスコのソフトウェアサイトからダウンロードできます。
 - ステップ2** ライブラリをインストールします。ライブラリは、クライアントワークステーションがアクセス可能なディレクトリ（Catalog Deployer がそこから実行されます）にインストールする必要があります。共有のネットワークドライブをライブラリの中心のストレージにできます。
 - ステップ3** ライブラリコンテンツの展開を担当するユーザが、この機能を含むロールを持っていることを確認します。
 - ステップ4** ライブラリをインポートします。ライブラリを Catalog Deployer にインポートします。[ライブラリパッケージのインポート](#) (44 ページ) を参照してください。
 - ステップ5** ライブラリコンテンツを確認します。ライブラリパッケージを展開する権限を持っているユーザが、対象サイトで受け取ったパッケージを選択し、状況に応じて、ライブラリコンテンツ（サービス）をプレビューします。ライブラリコンテンツをプレビューするには、プレビューするサービスのチェックボックスをオンにして選択し、[プレビュー (Preview)] をクリックします。
 - ステップ6** 選択したライブラリコンテンツを展開します。受信するパッケージを選択し、展開するパッケージコンテンツを選択して、展開を実行します。[ライブラリの展開](#) (46 ページ) を参照してください。
 - ステップ7** 展開ログを確認します。ライブラリコンテンツを展開すると、エンティティ定義の作成または更新に関する、すべての展開アクティビティの詳細なログが生成されます。
-

サイト管理者とカタログパブリッシャのロールには、ブランド化コンテンツライブラリから内容をインポート、プレビュー、または展開するための[展開のインポート (Import Deployments)]および[展開用パッケージの展開 (Deploy Deployment Package)]機能が含まれます。

ライブラリ パッケージのインポート

ライブラリ パッケージのインポートは、展開用パッケージのインポートと似ています。

ステップ1 [表示と検索 (View and Search)] ペインで、[アクション (Action)]>[インポート (Import)]を選択します（そのオプションが使用不可の場合は、現在のユーザに、「ブランド化ライブラリコンテンツの展開」機能を含むロールがありません）。

[インポート (Import)] ダイアログボックスが表示され、インポートするライブラリファイルを参照するように求められます。

ステップ2 [参照 (Browse)] をクリックします。

ステップ3 ライブラリ ファイルを見つけて、選択します。

ステップ4 [開く (Open)] をクリックします。

ステップ5 [インポート (Import)] をクリックします。

ライブラリがインポートされ、[展開用に受信 (Received for Deployment)] ステータスが割り当てられて、開かれます。

ステップ6 [インポート (Import)] ダイアログボックスを閉じます。

ライブラリ パッケージにはライブラリを識別し、その展開の詳細がわかる属性があります。

表 8: ライブラリ パッケージ属性のインポート

属性	説明
[パッケージ名 (Package Name)] と [説明 (Description)]	ライブラリの名前とそのコンテンツの簡潔な説明。
ライブラリ バージョン (Library Version)	ベンダーがリリースしたライブラリのバージョン。
ベンダー名	シスコ、またはコンテンツ提供者の名前。
アプリケーション バージョン (Application Version)	ライブラリの展開がサポートされているデータベースのバージョン。ライブラリは特定バージョンの Service Catalog に合わせてカスタマイズされています。
画像	ライブラリに関連付けられたイメージアイコン。
パッケージタイプ (Package Type)	サービス ライブラリまたはカスタム ライブラリ。

属性	説明
Status	ライブラリの現在のステータス。[受信済み (Received)] または [展開済み (Deployed)] のいずれか。
[作成日 (Created On)]/[作成者 (Created By)] [変更日 (Modified On)]/[変更者 (Modified By)]	Catalog Deployer は、このサイトのライブラリを作成した（つまり、ライブラリをインポートした）人と、最新の展開アクティビティの日付とそれを実行した人を自動的に追跡します。
履歴 (History)	このサイトのライブラリに対して実行されたすべての展開アクティビティの簡潔なログ。展開の詳細を見るためのより詳細なログへのリンクが付いています。

ブランド化ライブラリ パッケージの作成

適切なロールを持っているライセンスされたユーザは、[表示と検索 (View and Search)] ペインで [アクション (Action)] > [新しいライブラリパッケージ (New Library Package)] を選択することによって、新しいライブラリを作成できます。

手順は、次の例外を除いて、展開パッケージの作成手順 ([展開用パッケージの作成、\(25 ページ\)](#) を参照) と同じです。

- [ライブラリ バージョン (Library Version)] と [ベンダー名 (Vendor Name)] が必須です。これらは、ライブラリを識別する自由形式のテキストフィールドです。
- オレンジ色のボールアイコン ● がデフォルトで新しいライブラリに関連付けられ、[表示と検索 (View and Search)] ペインの最初の列に表示されます。ライブラリパッケージが保存されたら、[アクション (Action)] ペインで [イメージのアップロード (Upload Image)] をクリックすることによって、カスタムイメージアイコンも使用できます。イメージは JPG、PNG、または GIF 形式にすることができます。これらは 16 ピクセルの幅と 16 ピクセルの高さで表示されるため、この縦横比を維持するように作成する必要があります。イメージのサイズが正しくない場合は、ブラウザによってサイズが変更され、ぼやけたり、ノイズが入ったりすることがあります。
- 使用可能なパッケージ タイプは次のとおりです。
 - サービス ライブラリ。サービス定義とそのコンポーネントエンティティが含まれます。
 - カスタム ライブラリ。サービス ライブラリに関連付けられたエンティティが含まれます。
- ライブラリは、アプリケーションバージョン、ライブラリが作成された Service Catalog データベースのバージョンによっても識別されます。また、ライブラリは、それが作成されたも

のと同じバージョンのアプリケーションを実行している Service Catalog サイトにのみ展開できます。

- ライブラリ パッケージは、エクスポートはできますが、送信はできません。展開パッケージと同様に、ライブラリ パッケージをエクスポートする前に組み立てる必要があります。

ライブラリの展開

ライブラリはシスコから顧客サイトへ配布される新しいコンテンツの伝達手段です。ライブラリは開発サイトにのみ展開する必要があります。Catalog Deployerは、顧客の要件に合わせて以前にカスタマイズされた可能性のある既存のコンテンツを混乱させたり上書きしたりすることなくライブラリを展開できるように設計されています。そのため、ライブラリパッケージを展開するためのオプションは、ライブラリ コンテンツが、対象サイトに同じ名前で作成されているコンテンツを置き換えるのではなく、補完できるように設定されています。

- パッケージのプライマリ エンティティ（カスタム パッケージ内のサービスやディレクトリ 関連エンティティ）は、同名のエンティティがターゲットサイトで見つからない場合にのみ展開されます。ライブラリからの展開によって既存のコンテンツが置き換えられることはあります。
- サービス バンドルは常に親サービス定義とすべての子サービス定義を含みます。すべてのプライマリエンティティの場合のように、対象サイトに同じ名前のサービスがすでに存在している場合、サービス定義は省略されます（展開されません）。
- コンポーネント エンティティは、プライマリ エンティティとともに自動的に展開されるエンティティです。たとえば、サービスを展開することは、そのサービスによって使用されるディクショナリ、アクティブ フォーム コンポーネント、カテゴリ、およびキーワードを展開することになります。これらのコンポーネント エンティティは、対象サイトに同じ名前のエンティティがすでに存在している場合は展開されません。
- 関連付けられたエンティティは、プライマリエンティティによって参照されるエンティティです。たとえば、キー、人員、および組織はサービス タスク プランで参照でき、これらはサービスの関連エンティティ タイプです。プライマリエンティティの展開時に対象サイトに関連エンティティが存在する場合、それらのエンティティ間の関係が確立されます。関連付けられたエンティティが存在しない場合にはデフォルトの関係（たとえば、「デフォルト サービス キュー」に対する）が設定されます。

サポートエンティティ（キューや組織単位など）を含んでいるライブラリのコンテンツ全体は、同時に展開する必要があります。逆に、Catalog Deployer を使用すれば、展開するサービス ライブラリの内容を選択することができます。各サービスは、別々のトランザクションに展開されます。展開に失敗すると、現在のトランザクションがロールバックされます。

すべてのインポート アクションと展開アクションがライブラリ パッケージ用に記録されます。[履歴 (History)] セクションの [ログの表示 (View Log)] リンクに、Catalog Deployer によるアクションが詳しく表示されます。既知のエラーおよび欠落、(34 ページ) を参照してください。

用語

表 9：主な用語

用語	定義
パッケージのアセンブリ	コンテンツをパッケージに追加します。パッケージを組み立てると、パッケージに含まれているオブジェクトの現行定義がリポジトリから抽出され、パッケージに書き込まれます。これらのオブジェクトに対してアセンブリ以降に加えられた変更は、アセンブリされたパッケージに反映されません。
関連付けられた単位	プライマリ エンティティに関連するエンティティです。対象サイトで展開を成功させるために必要です。たとえば、サービス定義（プライマリエンティティ）は、タスクが割り当てられているいくつかの組織、サービスチームを参照する可能性があります（組織単位は、関連エンティティです）。
Catalog Deployer	あるサイトから別のサイトへのカタログコンテンツおよびディレクトリの展開を担当する Service Catalog モジュール。
コンポーネント エンティティ	サービスを展開する際に自動的に展開されるエンティティです。サービス コンポーネントエンティティには、カテゴリ、プレゼンテーション要素、ディクショナリ、ディクショナリグループ、サービスグループ、キーワード、および目的が含まれます。
データ ソース (Data Source)	アプリケーションサーバコンソールの JDBC データ ソースページで定義されるデータ ソース。Catalog Deployerによる直接のサイト間展開の対象となるサイトすべての接続情報を指定します。

用語	定義
展開用パッケージ	Catalog Deployer によって管理されるプライマリオブジェクト。展開用パッケージには、選択された、展開予定のエンティティ（サービス定義、キュー、グループなど）、および展開アクティビティ履歴が含まれます。展開用パッケージは、コンテンツを対象サイトに展開するために他のサイトに送信またはエクスポート/インポートすることができます。展開パッケージには、それが「組み立て」られ、その時点で含まれるデータがソースサイトから抽出されるまで、エンティティコンテンツは含まれません。
エンティティ (Entity)	Service Catalog ソフトウェア内で作成され保守されるオブジェクトの 1 つ。例：組織単位、サービス定義、キュー、および電子メールテンプレート。
エンティティ ホーム (Entity Home)	いずれのサイトが、サポートされる各エンティティタイプの正式サイトであるかを管理者が指定する、Administration モジュールの設定のページ。
パッケージのエクスポート	すでにアセンブリされたパッケージのコンテンツを含む XML ファイルを作成し、そのファイルをファイルシステムに書き込みます。ソースサイトから対象サイトへのパッケージのエクスポートは、パッケージ送信の代替手段です。
実装	直接または間接に接続されている 1 つ以上の Service Catalog サイトのグループ。
パッケージのインポート	以前にエクスポートされた XML 展開用パッケージを対象サイトにインポートすることにより、パッケージを作成します。パッケージはステータス [展開用に受信 (Received for Deployment)] で作成されます。
ライブラリ パッケージ	事前に組み立てられた、ファイルインポートとして使用可能なシスコのパッケージ。パッケージ内容は、サービスカタログの基礎を提供するために使用できるテンプレートサービス、または関連エンティティで構成されています。展開用パッケージと異なり、ライブラリの選択されたコンテンツは展開することができます。

用語	定義
プライマリ エンティティ	Catalog Deployer パッケージに含めるために選択できるエンティティ。
サイト	データベースおよび http アドレスを共有する 1 つ以上のコンピュータシステム（単一のコンピュータまたはクラスタ）の集合で、機能によって分類できます。たとえば、開発サイト、テストサイト、および本番サイトです。
ソース サイト	パッケージが作成されるサイトです。他のサイト上で展開するパッケージを送信またはエクスポートします。
対象サイト	パッケージが展開されるサイトです。送信またはインポートを使用してパッケージを受信します。
送信	データベース接続を使用して、あるサイトから別のサイトにパッケージを送信します。 ジョブを送信すると、ソースサイトでそのジョブの編集（再アセンブリやコンテンツ変更）はできません。ターゲットサイトは Catalog Deployer 経由で送信を受信します。

