



アップグレード計画

この章では、アップグレードの考慮事項および CVP アップグレードを実行するために使用できる方法に関する概要を示します。

- [アップグレード方法, 1 ページ](#)
- [アップグレードに関する考慮事項, 4 ページ](#)

アップグレード方法

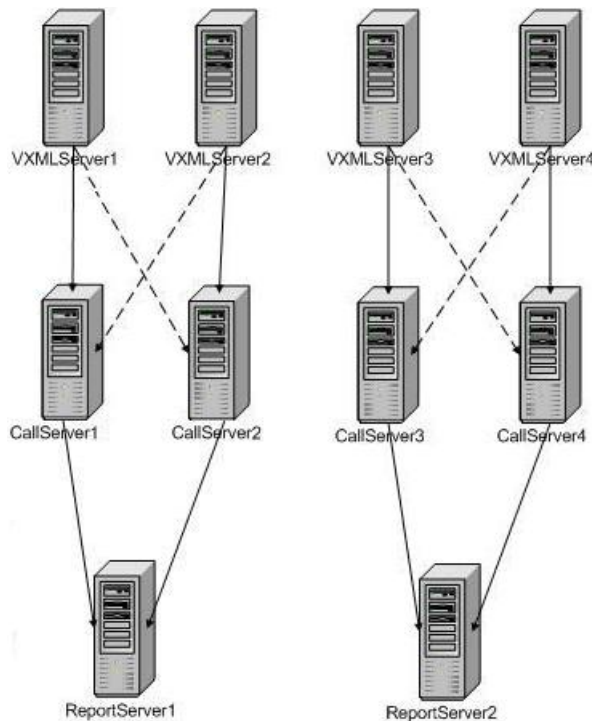
通常、アップグレードを実行する期限付きメンテナンス ウィンドウがあります。多数の CVP Server が存在するときに、1 回のメンテナンス ウィンドウですべてのサーバをアップグレードできないことがあります。CVP には、大規模な CVP 展開でアップグレードプロセスを分割するのに役立つアップグレード方法があります。CVP ユニットおよびマルチフェーズアップグレードの概念を使用して、複数のメンテナンス ウィンドウで実行できる複数のステップにサーバのアップグレードを分割できます。

CVP ユニット

一部の Unified CVP 展開は、コールパスが CVP Server のサブセットのみを通過するように、論理的に分割されます。サーバのこれらの論理的なグループ化を CVP ユニットと呼びます。CVP ユニットは、コールの処理に関与する可能性のあるすべての VXML Server、コールサーバ、および Reporting Server によって構成されます。これには、フェールオーバーマシン、VXML ゲートウェイがコールを転送できるその他の CVP ユニット、およびサブルーチン要素によって呼び出される VXML アプリケーションを格納するすべての VXML サーバが含まれます。

複数の CVP ユニットが含まれる CVP 展開では、一度に 1 つのユニットをアップグレードできません。この方法は、コールセンターが H.323 から SIP に移行していて、かつコール処理を継続しながら、リスクを最小限に抑える必要がある場合に、非常に役立つことがあります。CVP ユニットは、ユニット内のすべてのコンポーネントがアップグレードされるまで、オフラインのままである必要があります。

Unified CVP 展開の例（VXML Server、コールサーバ、Reporting Server、およびコールパス）を以下に示します。



図には、2つの論理 CVP ユニットが示されています。1つのユニットから他方のユニットに移動するコールパスがないため、これらのユニットは論理ユニットになります。

- CVP ユニット 1 : VS1、VS2、CS1、CS2、RS1
- CVP ユニット 2 : VS3、VS4、CS3、CS4、RS2

以下のシナリオは、CVP ユニットをアップグレードする方法を示す例です。

シナリオ 1 : メンテナンス ウィンドウで関連するサーバの CVP ユニットのアップグレード

- メンテナンス ウィンドウ 1 : RS1、CS1、CS2、VS1、VS2
- メンテナンス ウィンドウ 2 : RS2、CS3、CS4、VS3、VS4

各論理ユニットは個別にアップグレードできます。CVP ユニットのアップグレードは、次の項で説明されているマルチフェーズアップグレード方法を使用して、さらに分割できます。

マルチフェーズアップグレード

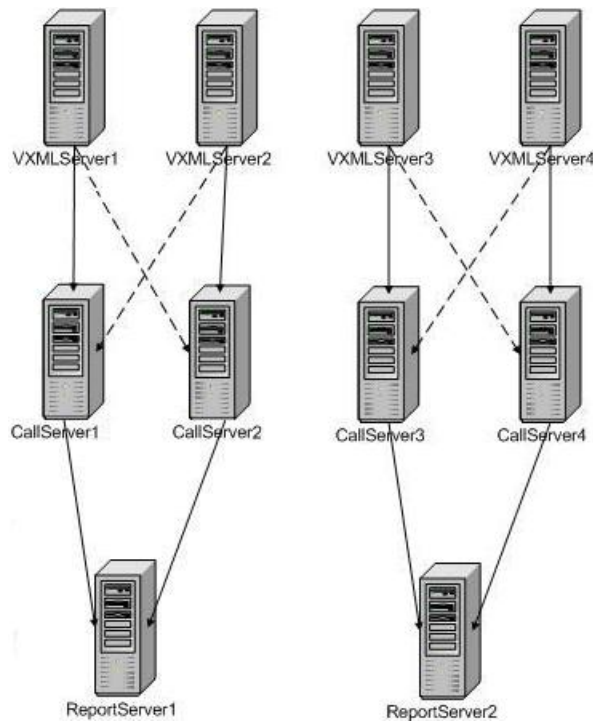
マルチフェーズアップグレードは、Unified CVP Server のサブセットをアップグレードし、コール処理を再開する機能です。マルチフェーズアップグレードプロセスでは、時間の経過でアップグレードを分割する追加の方法を提供します。CVP 展開に複数の CVP ユニットが存在する場合、マルチフェーズアプローチを使用して各ユニットをアップグレードできます。

マルチフェーズアップグレードでは、以下の順序で、すべてのコンポーネントをアップグレードする必要があります。

- 1 Operations Console (OAMP)
- 2 Reporting Server
- 3 コールサーバ
- 4 VXML Server

1つのカテゴリのすべてのサーバを1つのメンテナンス ウィンドウでアップグレードする必要はありません。ただし、Unified CVP 展開または CVP ユニットの次のコンポーネントのセットに移る前に、1つのタイプのすべての CVP コンポーネントをアップグレードする必要があります。

図 1: CVP Server とコールパス



以下のシナリオは、マルチフェーズアップグレードの実装方法を示す例です。

シナリオ 2: メンテナンス ウィンドウで特定のタイプのすべてのサーバをアップグレードする

- メンテナンス ウィンドウ 1: RS1、RS2
- メンテナンス ウィンドウ 2: CS1、CS2、CS3、CS4
- メンテナンス ウィンドウ 3: VS1、VS2、VS3、VS4

シナリオ 3 : メンテナンス ウィンドウでサーバタイプのサブセットをアップグレードする

- メンテナンス ウィンドウ 1 : RS1、RS2、CS1
- メンテナンス ウィンドウ 2 : CS2、CS3、CS4、VS1
- メンテナンス ウィンドウ 3 : VS2、VS3、VS4

シナリオ 4 : メンテナンス ウィンドウで CVP ユニットからサーバタイプのサブセットをアップグレードする

- メンテナンス ウィンドウ 1 : RS1、CS1
- メンテナンス ウィンドウ 2 : CS2、VS1、VS2
- メンテナンス ウィンドウ 3 : RS2、CS3
- メンテナンス ウィンドウ 4 : CS4、VS3、VS4

アップグレードに関する考慮事項

CVPユニットまたはマルチフェーズアップグレード、あるいはその両方を計画するときに、以下のアップグレードに関する考慮事項を確認してください。

CVP ユニットおよびマルチフェーズ アップグレード

製品のアップグレードを複数のステップに分割するときに、以下の点を考慮することは重要です。

全体的な考慮事項

- サーバでのコンポーネントの共存
 - マルチフェーズアップグレードは、コンポーネントが共存する展開ではサポートされません。
 - CVPユニットのアップグレードでは、すべての共存コンポーネントが同じユニットの一部である場合、共存コンポーネントがサポートされます。非共存コンポーネントと同様、ユニットを構成するすべてのコンポーネントがアップグレードされるまで、CVPユニットはオフラインのままである必要があります。たとえば、共存する VXML Server とコールサーバが同じ CVP ユニットに存在する場合、そのユニットはアップグレードできます。すべてのコンポーネントがアップグレードされると、ユニットはコール処理を開始します。

**注意**

展開がアップグレード順序に沿って実行されず、その展開がコールを処理した場合、一部のコールデータは失われるおそれがあります。たとえば、7.0(2) Reporting Server がメッセージを 9.0(1) コールサーバから受信する場合、そのコールに関する CVP 9.0(1) で導入された新しいフィールドやメッセージはいずれも保持されません。

Ops Console

- Unified CVP 9.0(1) では、展開ごとに 1 つの Ops Console Server がサポートされます。
- Ops Console は、常に最初にアップグレードする必要があります。最初にアップグレードしないと、Unified CVP 9.0(1) リリースで導入されたプロパティを設定できません。
- Ops Console には、最後にリリースされたバージョンのユーザ インターフェイスのみが表示されます。以前のバージョンのデバイス タイプのインターフェイス画面形式はサポートされません。たとえば、コールサーバがバージョン 8.0(1) であるとき、9.0(1) バージョンのコールサーバ設定画面では、該当する場合、新しい設定オプションとデフォルト値が表示されます。
- Ops Console が古いリリース バージョンのデバイス タイプのオンライン ビューを実行する場合、デフォルトを適用して、古いデバイス タイプでは不明な新しいプロパティを読み込もうとします。
- 古いバージョンのデバイス タイプをサポートし続けるには、Ops Console を使用して、既存のプロパティの属性を削除または変更することはできません。
- 以前のバージョンのコンポーネントが新しいバージョンにしか適用されない設定プロパティを受け取る場合、トレース メッセージをログに記録します。
- Unified CVP デバイス タイプのバージョン情報は、すべての Unified CVP 9.0(1) コンポーネントで使用できます。アップグレードされていない CVP デバイス タイプのバージョン情報は、使用できません。

