

# Oracle WebLogic Server の 自動サービス移行

*Oracle ホワイト・ペーパー*  
2008 年 7 月



## **注 :**

本書は、オラクルの一般的な製品の方向性を示すことが目的です。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。下記の事項は、マテリアルやコード、機能の提供を確約するものではなく、また、購買を決定する際の判断材料とはなりません。オラクルの製品に関して記載されている機能の開発、リリース、および時期については、弊社の裁量により決定いたします。

## はじめに

今日のミッション・クリティカルなアプリケーションは、高可用性のサービスを必要とします。Oracle WebLogic Server は、エンタープライズ・クラスのアプリケーションの構築を容易にするメッセージング、トランザクション、およびほかのシステム・サービスを提供します。これらのサービスをクラスタ環境で実行して、スケーラブルで高可用性のアプリケーションを構築できます。Oracle WebLogic Server の以前のリリースでは、障害が発生したクラスタ・サーバーから正常なサーバーに管理者が高可用性のサービスを手動で移行できるサポートが提供されました。Oracle WebLogic Server 10.3 では、WLS Clustering インフラストラクチャにサポートが追加され、管理者が介入することなく高可用性のサービスが自動的に移行されるようになりました。本書では、通常の自動サービス移行フレームワークのさまざまな要素と WebLogic Server 10.3 リリースのこのフレームワークの詳細な説明とベスト・プラクティスを示します。

## 用語と定義

| 用語                   | 定義  |
|----------------------|---|
| <b>WebLogic ドメイン</b> | WebLogic Server ドメインは、管理用の WebLogic Server リソースの論理グループです。各ドメインには、Administration Server と呼ばれる特別な WebLogic Server インスタンスがあります。これは、ドメインのすべてのリソースを構成および管理する中核です。通常、ドメインには、Managed Server と呼ばれる追加の WebLogic Server インスタンスも含まれます。 |
| <b>WebLogic クラスタ</b> | WebLogic Server クラスタは、スケーラビリティと信頼性を向上させるために統合された一連の WebLogic Server インスタンスです。クラスタのすべてのサーバーは、同じ WebLogic ドメインに属します。<br><br>クラスタは、単一の WebLogic Server インスタンスとしてクライアントに表示されます。   |
| <b>ノード・マネージャ</b>     | ノード・マネージャは、リモートの場所から Administration Server および Managed Server インスタンスのライフ・サイクルを制御できる WebLogic Server ユーティリティです。  |

|                             |  |
|-----------------------------|--|
|                             | ノード・マネージャ・プロセスは、特定のマシンに関連づけられています。サーバー・インスタンスがノード・マネージャ・プロセスと同じマシンに存在する限り、ノード・マネージャ・プロセスを使用して WebLogic Server ドメインのサーバー・インスタンスを制御できます。 |
| ターゲット                       | さまざまな J2EE サブシステム・サービスのコンテナとしてホストおよび使用できる WebLogic Server の管理エンティティです。たとえば、サーバー、クラスタ、移行可能なターゲットです。                                     |
| サーバー・ターゲット                  | ターゲットとして使用される単一の WebLogic Server インスタンスです。   |
| クラスタ・ターゲット                  | JNDI などの一般的なリソースをレプリケートおよび共有し、クラスタ・サービスを提供する単一のエンティティとして動作するクラスタ化された 1 つ以上のサーバーです。   |
| 移行可能なターゲット                  | 1 つ以上の候補サーバーを含む仮想ターゲットです。任意の時点に 1 つのアクティブ・サーバーのみを使用します。  |
| 候補サーバー                      | 移行プロセスで User Preferred Server のスタンバイとして使用できるサーバーのリストです。  |
| User Preferred Server (UPS) | 移行可能なシステムを開始する際にユーザーが推奨するホストされたサービスを実行するサーバー・インスタンスです。   |
| ソース・サーバー                    | サービスの移行"元"のサーバー・インスタンスです。  |
| 移行先サーバー                     | サービスの移行"先"のサーバー・インスタンスです。  |
| SAN                         | ストレージ・エリア・ネットワークです。  |

## Oracle WebLogic Server サービスの概要

WebLogic Server は、クラスタ・サービスおよびシングルトン・サービスを提供します。通常、"クラスタ"サービスは、クラスタのすべてのノードで使用できます。たとえば、クラスタに配置された Web アプリケーションは、クラスタの各サーバーでアプリケーションのインスタンスを使用します。セッション・レプリケーションにアプリケーションが構成される場合、クラスタのすべてのノードで実行されるクラスタ・レプリケーション・サービスは、クラスタの別のノードにセッションをレプリケートします。

"シングルトン"サービスは、あらゆる時点でクラスタの单一のノードにのみ存在するサービスで、特定のサービス品質 (QOS) を提供し、もっとも重要な点として、データの一貫性を維持します。この例として、アプリケーション・メッセージの单一のコピーだけがプロデューサからの永続的な送信操作のたびにストレージに保存されることを保証するJMS (Java Messaging Service) サービスがあります。しかし、この高度なQOSにはコストがかかります。シングルトン・サービスをホストするノードで問題が発生した場合、問題が解決するまでサービスを使用できなくなることは明らかです。これは、エンタープライズ・アプリケーション環境の課題です。

WebLogic Server でのこの課題に対処する方法の詳細を確認する前に、現在 WebLogic Server で提供されているさまざまなシングルトン・サービスの概要を説明します。

現在の一連のシングルトン・サービスを以下のカテゴリにグループ化できます。

- 永続ストア・サービス
- JMS 関連サービス
- JTA 関連サービス
- ユーザー定義のシングルトン・サービス

## 永続ストア・サービス

WebLogic Server は、多くのサブシステムで共有できる永続ストア・サービスを提供して、さまざまな永続データを格納します。永続ストア・サービスを使用する特定の永続ストアおよびサブシステムは、すべて同じ WebLogic Server インスタンスで実行する必要があります。各 WebLogic Server には、"デフォルト"のファイル・ベースの永続ストアが含まれます。追加のカスタム・ストアを構成して、1 つ以上のサブシステム・サービスに関連づけることができます。WebLogic Server は、ファイル・ベースおよびデータベース表ベースのカスタム・ストアをサポートします。

カスタム・ストアは移行可能です。デフォルト・ストアも移行できますが、JTA サービス移行の一部としての暗黙的な移行のみです（JTA以外の移行可能なサービスでデフォルト・ストアを使用できません）。ストアの詳細については、[1]を参照してください。

## JMS 関連サービス

### JMS サービス

WebLogic JMS サービスは、WebLogic Server のパブリッシュ・サブスクライブおよび Point-to-Point メッセージング機能を提供します。このサービスは、JMS の物理的な宛先（キューおよびトピック）をホストするコンテナとして動作する "JMSServer" と呼ばれる構成エンティティで管理されます。通常、WLS サーバー・インスタンスを対象とします。

永続的なメッセージングをサポートしてデータの一貫性を維持するため、特定の JMS Server のすべての永続状態（メッセージ、トランザクションや確認応答などの内部状態）をクラスタの単一の場所に格納します。これは、同じ WebLogic Server インスタンスにホストされる、関連した永続ストアです。JMS の詳細については、[2] を参照してください。

#### SAF サービス

WebLogic Server 9.0 の別のメッセージング・サービスとして、Store-And-Forward (SAF) サービスが導入されました。名前が示すとおり、SAF サービスは、ローカルのクラスタまたはドメインのメッセージを格納し、送信先が使用できる場合にリモートのクラスタまたはドメインの最終的な送信先にメッセージを転送します。このサービスは、"SAF Agent" と呼ばれる構成エンティティで公開および管理されます。この構成エンティティは JMS Server に似ていますが、SAF インポート済み送信先をホストするコンテナとして動作します。対象の観点での JMS Server と SAF Agent の違いは、SAF Agent がクラスタまたは WebLogic Server インスタンスのリストを対象とし、JMS Server が単一の WebLogic Server インスタンスを対象とすることです。

JMS サーバーと同様に、SAF エージェントは、永続状態情報を格納するために関連する永続ストアを使用します。このため、不正なサーバーおよびストアによる類似した問題に直面し、サービスの停止時間が発生します。

SAF エージェントは、Web-Service Reliable Messaging (WS-RM) をサポートするよう構成できます。これらのシナリオでは、WS-RM を使用してインポート済み送信先をホストするために、移行できない SAF Agent のみが選択されます。SAF サービスの詳細については、[3] を参照してください。

#### パス・サービス

パス・サービスは、名前のついた一連のメッセージ (JMS メッセージの "Unit-of-Order") とクラスタの特定の JMS サーバーまたは SAF エージェントのマッピングを格納するために使用されるオプションの永続マップ・シングルトン・サービスです。同じシーケンス名のメッセージを物理的に同じ場所に固定して順序を適用する方法を提供します。このサービスは、"Path Service" と呼ばれる構成エンティティで公開および管理されます。クラスタごとに単一のパス・サービス・エンティティのみ構成できます。JMS Server および SAF Agent と同様に、関連する永続ストアを使用してデータを保存します。詳細については、[4] を参照してください。

#### Message Driven Bean (MDB)

移行可能な JMS サーバーでホストされる移行先をリスニングする構成の Message Driven Bean は、ホストしている JMS Server の移行先とともに移行されます。詳細については、[5] を参照してください。

#### JTA 関連サービス

クラッシュ後のトランザクション・リカバリを正常に処理するために、JTA Transaction Recovery Service (TRS) が設計されました。トランザクション・リカバリ・サービスの自動移行では、状態監視サブシステムを活用して、移行可能なターゲットでホストされるサービスの状態を監視します。

プライマリ・サーバーで障害が発生した場合、移行可能なサービス・フレームワークは、トランザクション・リカバリ・サービスをバックアップ・サーバーに自動的に移行します。移行フレームワークは、構成された候補サーバーからバックアップ・サーバーを選択します。移行アクションを完了する前にバックアップ・サーバーで障害が発生した場合、トランザクション・リカバリ・サービスを別のバックアップ・サーバーに手動で再移行する必要があります。

手動または自動のサービス移行の場合、障害が発生した移行可能なサーバーから移行される可能性のあるマシンでアクセスできる共有ストレージ・システムにコードを保存するため、デフォルトの永続ストアを構成する必要があります。つまり、トランザクション・リカバリ・サービスを移行できる可能性のあるサーバーを、元のサーバーのデフォルト・ストア・ファイルにアクセスできるサーバーにだけ制限する必要があります。

TRSの詳細については、[6]を参照してください。

## ユーザー定義のシングルトン・サービス

また、開発者は、WebLogic Serverを使用して、クラスタのシングルトン・サービスとして使用できるカスタム・サービスを作成できます。シングルトンとして動作させるには、クラスに"weblogic.cluster.singleton.SingletonService"インターフェースを実装する必要があります。

SingletonServiceインターフェースには、次のメソッドが含まれます。

- `public void activate()`

クラスタのノードでサービスがアクティブになる場合にこのメソッドが呼び出されます。

- `public void deactivate()`

クラスタのノードでサービスが非アクティブになる場合にこのメソッドが呼び出されます。

ユーザー定義のシングルトン・サービスの詳細については、[7]を参照してください。

## 移行プロセス

前項で説明したように、シングルトン・サービスは最高品質のサービスを提供しますが、"シングル・ポイント障害"の影響を非常に受けやすくなります。この問題に対処するため、WebLogic Serverは、"移行"と呼ばれるソリューションを提供します。

WebLogic Serverの移行とは、クラスタ化されたWebLogic Serverインスタンスまたは障害が発生した場合にクラスタ化されたインスタンスで実行されていたサブシステム・コンポーネントを移動するプロセスです。

障害が発生した場合に特定の物理マシンから別の物理マシンにサーバー・インスタンス全体を移動するプロセスは、*Whole Server Migration* (WSM) と呼ばれます。また、特定のサーバー・インスタンスから別のサーバー・インスタンスに影響するサブシステム・サービスだけを移動するプロセスを*Service Migration*といいます。

WebLogic Server 9.0 リリースでは、*Whole Server Migration*プロセスが導入されました。このプロセスでは、なんらかの理由で移行可能なサーバーが使用できなくなつた場合（停止してネットワークの接続が切断された場合やホスト・マシンで

障害が発生した場合など) にサーバー・インスタンスが自動的に移行されます。サーバー・インスタンスで障害が発生した場合、可能であれば移行可能なサーバーが同じマシンで自動的に再起動します。障害が発生したマシンで移行可能なサーバーを再起動できない場合は、別のマシンに移行されます。また、管理者は、サーバー・インスタンスの移行を手動で開始できます。詳細については、0 項を参照してください。

Service MigrationがWebLogic Server 7.0 に追加されました。この手法は、障害が発生したサブシステム・サービスのみを、不正なサーバー・インスタンスから使用できる正常なサーバー・インスタンスに手動で移行するために必要なインフラストラクチャを提供します。

ただし、手動の移行には、サーバー/サービス障害を検出して移行を開始するため、ユーザーの介入が必要になります。これにより、WLS ユーザーの全体の管理および所有コストが増加し、予期せずにサービスを使用できなくなる可能性があります。

WebLogic Server 10.3 では、サービス障害の検出とリカバリのプロセス全体が向上した移行フレームワークで完全に自動化されます。*Automatic Service Migration (ASM)* フレームワークは、シングルトン・サービスの状態を事前に監視し、クラスタの使用できる正常なサーバー・インスタンスに障害が発生したサービスを自動的に移行します。このため、移行の実行時間が大幅に削減され、これらのサービス全体の可用性が向上します。

このホワイト・ペーパーの後半では、クラスタ環境のシングルトン・サービスの可用性を高めるさまざまなインフラストラクチャ、フレームワーク、ユーティリティ、およびツールを説明します。

## サービス移行フレームワーク

サービス移行フレームワークには、移行可能なターゲット、状態監視サービス、リース・サービス、およびノード・マネージャ・プロセスが含まれます。

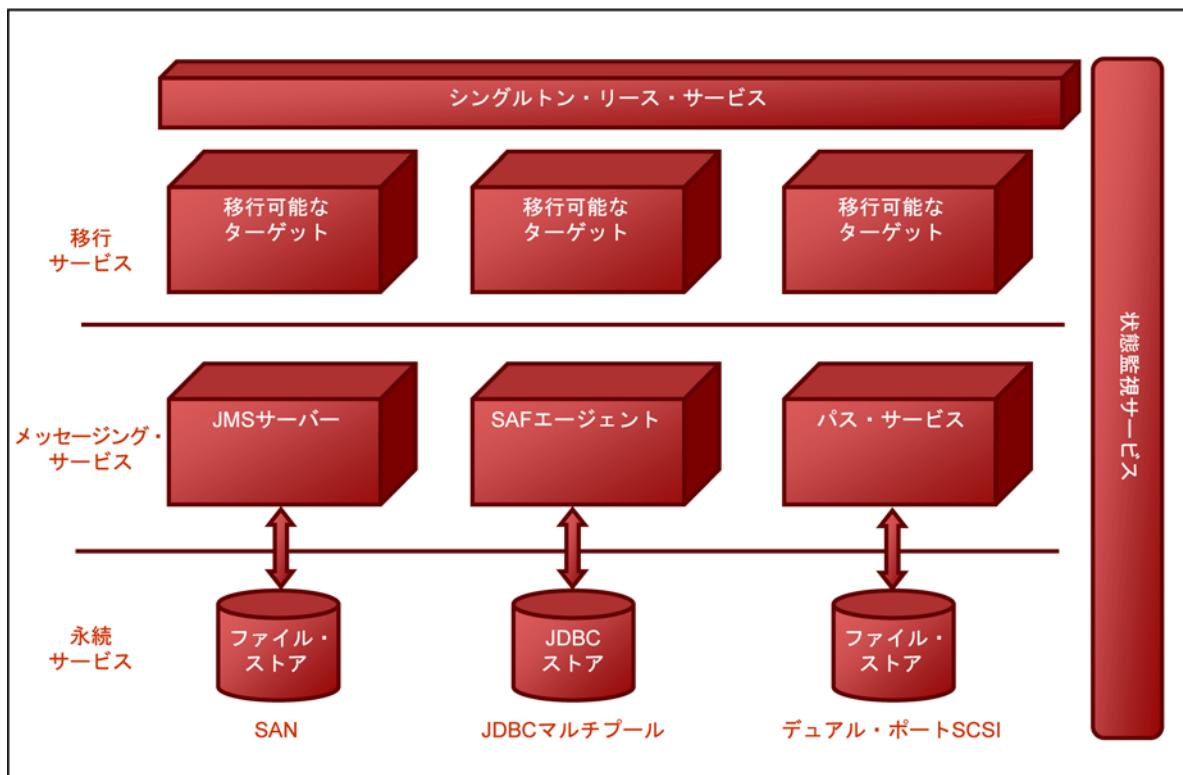


図 1 : JMS ASM の高レベルのアーキテクチャ

### 移行可能なターゲット

移行可能なターゲットは、多くの移行可能なサービスをホストできる論理的なターゲットです。一括して実行および移行をおこなうサービスの集合として使用できます。移行可能なターゲットは、クラスタに関連づけられます。また、現行ホスト・サーバーと呼ばれるクラスタの単一のノードでのみ常にホストされます。移行可能なターゲットのすべての移行可能なサービスは、移行可能なターゲットをホストするクラスタ・ノードで実行されます。移行可能なターゲットが移行されると、このターゲットでホストされるすべてのサービスも移行されます。

### サブシステムの状態監視サービス

サーバー・サブシステムの状態監視サービスは、サーバーの通常の処理またはパフォーマンスに影響を与えるさまざまな悪条件（メモリの不足やデータ永続性を妨げる IO エラーなど）を検出し、適切な処理を実行してサービスの中止を少なくします。

### リース・サービス

リース・サービスは、クラスタのサーバー・インスタンスにリースを付与して、移行可能なターゲットをホストします。これらのリースは、移行可能なターゲットの排他的な所有権を保証します。移行可能なサービスは、移行可能なターゲットのリースを保持するサーバーにホストされます。移行可能なサービスを継続的にホストできるように、各サーバーは定期的にリースを更新します。

サーバーがクラッシュした場合またはネットワーク・パーティションなどのほかの障害によってリースを更新できない場合、障害が発生したサーバーの移行可能なサービスがリースを取得する正常なサーバーに移行されます。

リース・サービス・フレームワークでは、クラスタの单一の管理サーバー・インスタンスが"Singleton Master"として動作します。つまり、クラスタのすべてのシングルトン・サービスおよび移行可能なサービスの状態を監視し、障害が発生した場合にクラスタ内の異なる WebLogic Server インスタンスにこれらのサービスを移行して、移行可能なサービスの移行可能なターゲットがクラスタ内の複数のサーバーで実行されないことを保証します。

WebLogic クラスタには、リース情報を処理する 2 つのオプションがあります。

- データベース・リース

リース情報がデータベースに格納されます。ただし、データベースの高い可用性を実現する必要があります。また、リース情報にアクセスする場合、各クラスタ・ノードでデータベースに接続できる必要があります。データベースを使用できず、自動サービス移行フレームワークが動作しない場合、クラスタのすべての移行可能なターゲットとシングルトン・サービスが非アクティブになります。

- コンセンサス

リース情報がクラスタ・ノードのメモリ内に格納されます。高い可用性のデータベースは必要ありませんが、ノード・マネージャ・プロセスで起動されるすべてのクラスタ・サービスが必要です。適切な動作をおこなうには、コンセンサス・リースで常にノード・マネージャ・プロセスを実行する必要があります。次の項でノード・マネージャを説明します。

リース・サービスの詳細については、[8]を参照してください。

## ノード・マネージャ

サービス移行フレームワークは、ノード・マネージャを使用して、オプションの移行前後のスクリプト（このホワイト・ペーパーの後半で説明します）を実行します。また、コンセンサス・リースでは、ノード・マネージャでサーバーを起動および監視する必要があります[9]。

## サービス移行フレームワークの構成

### 移行基盤（リース・オプション） - クラスタ

移行基盤は、移行サービス・フレームワークで使用されるリース・サービス・タイプ（データベースまたはコンセンサス）を決定します。詳細については、前の項を参照してください。設定はクラスタ全体に適用されます。

移行可能なターゲットは、次の構成可能なプロパティを提供します。JTA トランザクション・リカバリ・サービスに明示的なMTがないことに注意してください。

## User Preferred Server (UPS) - 移行可能なターゲット

デフォルトで、各 WebLogic Server インスタンスに WLS サーバー・インスタンス名の移行対象（移行可能）が含まれます。

それぞれの移行可能なターゲットで、管理者は User Preferred Server としてクラスタの単一のノードを選択する必要があります。移行サービス・フレームワークは、使用可能な場合に常に UPS のサービスを起動します。

## 候補サーバー - 移行可能なターゲット

候補サーバーは、移行可能なターゲットを移行できる一連のサーバーを定義します。これは、クラスタのノードのサブセットです。移行可能なターゲットに候補サーバーが定義されていない場合、クラスタのすべてのノードが候補サーバーになり、移行可能なターゲットをクラスタのすべてのノードに移行できます。

## 移行ポリシー

移行可能なターゲットの移行ポリシーは、状態障害または管理停止が発生した場合に自動的に移行をおこなうかどうかを制御します。自動移行を無効にする"Manual"ポリシーと 2 つの自動移行ポリシー（"Exactly-Once"および"Failure-Recovery"）があります。デフォルト・ポリシーは"Manual"です。

### Exactly-Once

Exactly Once ポリシーは、候補リストの少なくとも 1 つのサーバーが実行されている場合にサービスを常にアクティブにする必要があることを示します。最初に、User-Preferred Server (UPS) でサービスが起動されます。しかし、UPS を使用できない場合、クラスタのほかの候補サーバーのサービスがアクティブになります。サービスをホストするサーバーが停止した場合（正常な停止または強制的な停止）、または障害が発生した場合、サービスが別の候補サーバーに移行されます。

### Failure-Recovery

Failure-Recovery ポリシーは、User-Preferred Server (UPS) が正常に起動されている場合にのみサービスが起動されることを示します。管理者が手動で UPS を停止した場合（正常な停止または強制的な停止）、failure-recovery サービスは移行されません。一方、内部エラーで UPS の障害が発生した場合は、状態監視サービスで検出されます。また、クラッシュした場合は、サービスが自動的に別の候補サーバーに移行されます。

移行可能なサービスをクラスタの特定のサーバーで起動する必要があり、サーバーが停止してサービスを移行できない場合に、このポリシーが役立ちます。ただし、サーバーがクラッシュした場合は、サービスを移行する必要があります。

### Manual

Manual ポリシーは、サービスの User-Preferred Server が停止した場合（正常な停止または強制的な停止）、または障害が発生した場合にサービスが自動的に移行されないことを示します。管理者は、障害が発生したノードから正常なノードに移行可能なターゲットを手動で移行する必要があります。

## アクティブ前/非アクティブ後の移行スクリプト

移行可能なサービスが元のマシンで非アクティブになったあとと移行可能なサービスがターゲット・マシンでアクティブになる前に、これらのオプションのスクリプトが移行フレームワークで実行されます。移行フレームワークは、移行元と移行先の WebLogic Server のノード・マネージャに問い合わせて、これらのスクリプトを実行します。これらのスクリプトは、移行可能なサービスで使用する必要がある共有パーティション/ファイル・システムのマウントおよびアンマウントなどの移行前後のクリーンアップ・タスクを実行するメカニズムを提供します。

スクリプトで障害が発生した場合、非アクティブ後のスクリプトの障害を移行可能なサービスの移行を妨げる致命的なエラーとしてマークできます。このような場合、管理者は、スクリプトの障害の修正アクションを実行したあとに移行可能なターゲットを手動で移行する必要があります。

### Restart-In Place

移行可能なターゲットは、別のクラスタ・ノードに移行する代わりに同じクラスタ・ノードで障害が発生したサービスを非アクティブにして再アクティブ化する "restart-in-place" オプションを提供します。

サービスの移行可能なターゲットが不正でも、サーバーの状態が適切な場合（つまり、RUNNING 状態）、移行フレームワークはサービスの再起動だけをおこないます。サーバーが不正な場合、フレームワークは、すべてのインプレース再起動をスキップしてすぐに移行段階に進みます。再起動の回数とその時間間隔は、構成可能です。指定された回数の再起動がおこなわっても移行可能なターゲットを再起動できない場合、フレームワークはターゲットを別のサーバーに移行します。

移行可能なターゲットの構成の詳細については、[10]を参照してください。

## 移行構成ルールおよびベスト・プラクティス

自動サービス移行機能を使用する場合、クラスタの"移行基盤"を構成し、自動移行を有効にして、オプションで移行前後のスクリプトを使用するかどうかを指定する必要があります。

メッセージング・サブシステム・サービスを移行する必要があり、"デフォルト" 永続ストアを使用できない場合、サービス・プロバイダ・エンティティと同じ移行可能なターゲットも対象とするカスタム永続ストアを定義する必要があります。

SAFAgent、JMSServer、およびカスタム・ストアのすべてで移行可能なターゲットを共有できます。SAF メッセージの一貫性を維持するため、WebLogic Server は、既存の SAF エージェントが移行可能なターゲットに変更されることを防止します。代わりに、既存の SAF エージェントを削除し、同じ値で新しい SAF エージェントを構成して、メッセージの順序を維持するために移行可能なターゲットにする必要があります。

## ターゲット・ルール

### JMS サーバー

JMS サーバーは、移行可能なターゲットを使用しない場合、特定のクラスタ・メンバーを直接対象とし、ホスト・サーバーのデフォルト・ストアまたはカスタム・ストアを使用できます。ただし、移行可能なターゲットを対象とする場合、カスタム永続ストアを使用し、そのカスタム・ストアと同じ移行可能なターゲットを対象にする必要があります。JMS サーバー、SAF エージェント、およびカスタム・ストアは、移行可能なターゲットを共有できます。

### SAF エージェント

SAF エージェントは、移行可能なターゲットを使用しない場合、SAF エージェントとクラスタの各サーバーでデフォルト永続ストアを使用して、クラスタ全体またはクラスタの複数のサーバーのリストを対象にできます。ただし、移行可能なターゲットを対象にする場合、単一のターゲットのみを使用できます。また、カスタム永続ストアを使用する必要があります。JMS サーバー同様、カスタム・ストアで使用される同じ移行可能なターゲットを対象にする必要があります。SAF エージェント、JMS サーバー、およびカスタム・ストアは、移行可能なターゲットを共有できます。

また、SAF エージェントを移行可能なターゲットにする場合、以下のトピックを検討してください。

#### 移行可能なターゲットへの SAF エージェントの変更

SAF メッセージの一貫性を維持するため、WebLogic Server は、既存の SAF エージェントが移行可能なターゲットに変更されることを防止します。代わりに、既存の SAF エージェントを削除し、同じ値で新しい SAF エージェントを構成して、移行可能なターゲットにする必要があります。

#### 増加したメッセージ・スループットのための移行可能な SAF エージェントの特定

SAF エージェントは、移行可能なターゲットを使用しない場合、増加したメッセージ・スループットに対応するためにクラスタ全体またはクラスタの複数のサーバーを対象にできます。ただし、移行可能なターゲットを対象にする場合、クラスタ全体を含むクラスタのほかのサーバーを対象にできません。このため、クラスタの個別のサーバーの複数の SAF エージェントに JMS 宿先をインポートしてスループットを増やす場合、クラスタの各サーバーに移行可能なターゲットを作成し、移行可能なターゲットごとに個別の SAF エージェントを作成する必要があります。

#### 一貫したサービス品質のための SAF エージェントの特定

WebLogic 管理者は、同じクラスタまたは同じサーバーの複数の SAF エージェントを構成および配置できます。そのため、移行可能な SAF エージェントと移行できない SAF エージェントの両方が同じサーバーに存在する可能性があります。このような場合、JMS クライアント・アプリケーションの動作は、メッセージを処理する SAF エージェントによって異なります。

たとえば、インポート済み送信先を複数の SAF エージェントに配置できます。インポート済み送信先に送信されたメッセージのすべての SAF エージェント間でロードバランシングがおこなわれます。SAF エージェントのリストに移行できないエージェントが含まれる場合、JMS クライアント・アプリケーションの HA（高可用性）が制限されることがあります。このため、推奨されるベスト・プラクティスは、同じレベルの HA 機能を提供する 1 つ以上の SAF エージェントにインポート済み送信先を配置することです。つまり、一貫した転送の品質および動作を実現するには、移行可能なターゲットまたは移行できないターゲットを対象とする一連の SAF エージェントすべてにインポート済み送信先を設定する必要があります。

### パス・サービス

パス・サービスは、移行可能なターゲットを使用しない場合、クラスタの単一のメンバーを対象にします。また、デフォルト・ファイル・ストアまたはカスタム・ストアを使用できます。ただし、移行可能なターゲットを対象にする場合、デフォルト・ストアを使用できないので、同じ移行可能なターゲットにカスタム・ストアを構成して対象にする必要があります。追加のベスト・プラクティスとして、パス・サービスおよびカスタム・ストアを移行可能なターゲットのユーザーのみにする必要があります。ただし、JMS サーバー、SAF エージェント、およびカスタム・ストアは、移行可能なターゲットを共有できます。

### 永続ストア

前述のとおり、すべての JMS 関連サービスは、JMS サービスと同じ移行可能なターゲットも対象とするカスタム永続ストアを必要とします。

クラスタの移行可能なターゲットのサーバーで使用できる共有ディスクに移行可能なカスタム・ファイル・ストアを構成できます。また、移行前後のスクリプトを使用して、バックアップ・サーバー・ターゲットに移行可能なカスタム・ファイル・ストアを移行できます。最高の信頼性を実現するには、高い可用性の共有ストレージ・ソリューション（ストレージ・エリア・ネットワークやデュアル・ポート・ディスクなど）を使用します。

### JTA トランザクション・リカバリ・サービス

JTA では、自動または手動の移行に移行可能なターゲットの構成は必要ありません。移行可能なターゲットがサーバー・レベルで JTA に自動的に定義されるためです。JTA のデフォルトの移行ポリシーは手動ですが、自動移行に構成された場合の JTA ポリシーは内部で failure-recovery に設定されます。つまり、User-Preferred Server が起動する場合のみ、トランザクション・リカバリ・サービスが起動します。管理者が UPS の正常な停止または強制的な停止をおこなう場合、このサービスは移行されません。ただし、内部エラーで UPS が停止した場合は、このサービスが別の候補サーバーに移行されます。

トランザクション・リカバリ・サービスを別のサーバーに移行するのではなく、クラッシュしたサーバーを再起動してトランザクション・リカバリ・サービスで不完全なトランザクションに対応することを推奨します。ただし、許容できる期間を超えてサーバーを使用できない場合、障害が発生したサーバーのトランザクション処理をバックアップ・サーバーで実行できるように、トランザクション・リカバリ・サービスを別のサービスに移行できます。

## サブシステムのサービス移行とサーバー全体の移行

"移行プロセス"で説明したとおり、WebLogic Server は、自動サブシステム・サービス移行以外に WebLogic Server インスタンス全体の移行をサポートします。これは、"Whole Server Migration"と呼ばれます。WSM では、特定の物理マシンから別の物理マシンにサーバー・インスタンス全体の透過的なフェイルオーバーを実行します。このため、"同機種"のサービス移行のオプションが提供され、すべての一括移行が実行されます。新しいハードウェア全体の WLS インスタンスを起動する必要があるので、一般的にサービスを有効にする時間がかかります。一方、自動サブシステム・サービス移行 (ASM) は、おもに JMS や JTA などの"固定"サービスの移行に使用されます。WSM とは異なり、ASM は、サブシステム・サービスのフェイルオーバーおよび障害からのリカバリをサポートし、WSM と比較して高速なフェイルオーバーまたはリカバリ時間を提供します。

ユーザーは、WSM、ASM、または両方の使用を選択できます。使用する移行タイプの決定は、JMS/JTAサブシステム・サービスとともにすべてのアプリケーションを移行するかどうかなどの多くの要素に依存します。通常、基本的な使用には比較的簡単なWSMが推奨されています。しかし、高速のフェイルオーバー時間およびサービス移行の高度な制御を必要とする場合には、ASMが適しています。両方を構成する場合、サーバーはサブシステム・サービスを最初に移行します。これに失敗すると、WSMが実行されます。WSMの詳細については、[11]を参照してください。

## トラブルシューティング

次の WebLogic デバッグ・フラグによって、この機能面で発生する問題を診断できます。デバッグ・メッセージは、適切なサーバー・ログ・ファイルに記録されます。

| サブシステム領域                       | デバッグ・フラグ   |
|--------------------------------|--|
| シングルトン・モニター・アクションの情報           | -Dweblogic.debug.DebugSingletonServices=true   |
| コンセンサス・リースを使用する場合              | -Dweblogic.debug.DebugConsensusLeasing=true  |
| JMServer のデプロイメント/アンデプロイメントの追跡 | -Dweblogic.debug.DebugJMSBackEnd=true  |
| SAFAgent のデプロイメント/アンデプロイメントの追跡 | -Dweblogic.debug.DebugJMSSAF=true  |
| JMS モジュールのデプロイメント/アンデプロイメントの追跡 | -Dweblogic.debug.DebugJMSModule=true   |
| 永続ストアのデプロイメント/アンデプロイメントの追跡     | -Dweblogic.debug.DebugStoreIOPhysical=true<br>-Dweblogic.debug.DebugStoreIOLogical=true<br>-Dweblogic.debug.DebugStoreIOLogicalBoot=true |

## 結論

WebLogic Server 10.3 は、シングルトン・サービスの可用性を高める場合にユーザーが介入する必要がない、完全に自動化されたサービスの移行機能を提供します。ユーザーは、この機能を活用して、サーバー管理に関連する総所有コスト (TCO) を削減できます。また、エンタープライズ・アプリケーションの可用性を高めて、WebLogic Server プラットフォーム全体の投資収益率 (ROI) を向上できます。

## 付録 A : 参考資料および関連ドキュメント

- [1] [WebLogic永続ストア](#)
- [2] [WebLogic JMS サービス](#)
- [3] [WebLogic SAF サービス](#)
- [4] [WebLogicバス・サービス](#)
- [5] [MDBの移行](#)
- [6] [WebLogicトランザクション・リカバリ・サービス](#)
- [7] [ユーザー定義サービスの自動移行](#)
- [8] [自動移行サービスのリース](#)
- [9] [ノード・マネージャ管理者ガイド](#)
- [10] [移行可能なターゲットの構成](#)
- [11] [Whole Server Migration](#)



Oracle WebLogic Server の自動サービス移行  
2008 年 7 月

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

海外からのお問い合わせ窓口：  
電話 : +1.650.506.7000  
ファクシミリ : +1.650.506.7200  
[www.oracle.com](http://www.oracle.com)

Copyright © 2008, Oracle. All rights reserved.  
本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。  
本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による暗示を問わず、特定の目的に対する商品性もしくは適合性についての暗示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によつても再作成または送信することはできません。

Oracle は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。  
その他の名称はそれぞれの会社の商標です。