

Micro FocusR Enterprise Server

構成と管理

第 1 版
2003 年 5 月

このマニュアルでは、Enterprise Server の構成方法と管理方法について説明します。

このマニュアルは管理者を対象にしていますが、Enterprise Server を使用してデバッグを行う開発者にとっても役立つ情報が記載されています。

第 1 章 : はじめに

ここでは、Enterprise Server について説明します。

1.1 Enterprise Server の概要

Enterprise Server は、COBOL アプリケーションプログラムの実行環境を提供します。この実行環境はエンタープライズサーバとして知られています。このマニュアルおよび関連マニュアルでは、用語「エンタープライズサーバ」は明記しない限り、製品ではなく、実行環境を意味します。

COBOL アプリケーションは、各種のクライアントが発行したサービス要求に応じてエンタープライズサーバ内で実行されます。Net Express の一部である Interface Mapping Toolkit では、既存の COBOL アプリケーションをサービスとして公開できます。公開された COBOL サービスは、Web サービスとしてクライアントで、または J2EE コネクタや COM を介して起動できます。

サービスを実装する COBOL プログラムは、パラメータブロックを読み込んで処理し、処理した結果ブロックを返します。この COBOL プログラムは、非対話式プログラムである必要があります。サービスへのインターフェイスは、サービスの定義によって公開します。すべての COBOL アプリケーションがサービスとして公開するのに適しているわけではありません。詳細は、『分散コンピューティング』の『Enterprise Server へのアプリケーションの実装』の章を参照してください。

エンタープライズサーバは、COBOL アプリケーションのセッション管理と状態管理を提供します。また、オプションで外部リソースマネージャとのインターフェイスを取り、リソースの更新の整合を取ります。

エンタープライズサーバは、初期化済みのサービス実行プロセスプールを提供します。COBOL アプリケーションはこのプール内で実行されます。COBOL アプリケーションは独自のアドレス空間で実行されるため、エンタープライズサーバ内で実行されている他のプログラムから分離されます。複数のプロセスを実行することで、クライアントの要求メッセージに対応する COBOL プログラムを同時に複数実行できます。

Enterprise Server をインストールすると、エンタープライズサーバが 1 つ自動的に設定されます。エンタープライズサーバは新規に作成することもできます。

1.2 ライセンス

Enterprise Server を実行するにはライセンスが必要です。詳細は、[『ディプロイライセンスガイド』](#)を参照してください。

1.3 前提条件

唯一の前提条件として、クライアント側とサーバ側の両方のマシンで TCP/IP 通信ソフトウェアが実行されている必要があります。

1.4 エンタープライズサーバのアーキテクチャ

エンタープライズサーバにはいくつかの構成要素があります。図 1-1 に、エンタープライズサーバのアーキテクチャの概要を示します。

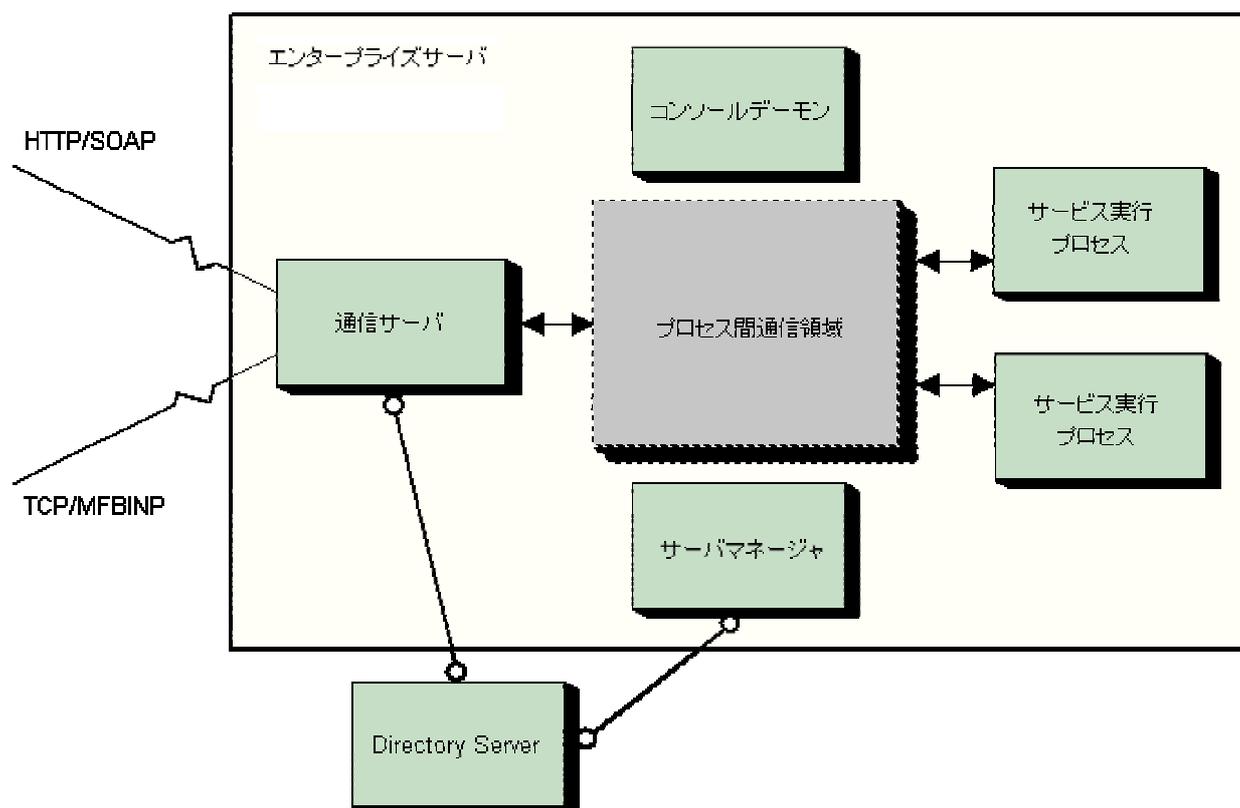


図 1-1 : エンタープライズサーバのアーキテクチャ

エンタープライズサーバには、複数のプロセスと、プロセス間通信で使用する領域が 1 つあります。

また、コンソールデーモン、サーバマネージャ、通信サーバがそれぞれ 1 つあります。これらのプロセスは次の機能を実行します。

- コンソールデーモンは、エンタープライズサーバのプロセスとログメッセージを管理します。
- サーバマネージャは、プロセス間通信領域の作成と管理を行い、コンソールデーモンと連動してプロセスを管理します。

- 通信サーバは、着信クライアント要求を受信し、その要求を該当するサービスに送信します。また、通信サーバは要求の処理結果を受信し、クライアントに返します。要求は次のどちらかのクライアントから発行されます。
 - Web サービスを要求するクライアント。これらのクライアントからの要求は、HTTP で SOAP (Simple Object Access Protocol) メッセージとして受信します。
 - サービスを要求する J2EE クライアント。J2EE クライアントからの要求は、Micro Focus 製の符号化と転送プロトコルを使用して受信します。

すべての種類の要求では、基本的な通信機構として TCP/IP を使用します。

エンタープライズサーバには、1 つ以上のサービス実行プロセスがあります。サービス実行プロセスは、名前が示すように、クライアント要求を処理する COBOL プログラムを実際に実行するプロセスです。

プロセス間通信領域には、エンタープライズサーバ内で利用できるすべてのサービスの定義を保持する共有メモリがあります。これらのサービスの定義は、図 1-1 に示す Directory Server で行われます。Directory Server はエンタープライズサーバの外側に位置し、複数のエンタープライズサーバの情報を保持します。また、プロセス間通信領域は、エンタープライズサーバのプロセス間で要求と応答を受け渡すためにも使用します。

1.5 サービス実行プロセス

サービスを提供する COBOL アプリケーションは、サービス実行プロセス内で実行されます。エンタープライズサーバは、複数のサービス実行プロセスを含むことができます。サーバを起動すると、サーバ内のすべてのサービス実行プロセスはクライアント要求を処理できます。サービス実行プロセスが要求の処理を完了すると、別の要求の処理が可能になります。

要求は次のように処理されます。

1. クライアント要求は通信サーバで受信され、共有メモリ領域内のキューに置くことでその実行をスケジュールします。
2. サービス実行プロセスの 1 つが使用可能になると、要求の処理を開始します。サービス実行プロセスは、Enterprise Server の環境で動作するように適合された COBOL ランタイムシステムである、アプリケーションコンテナを呼び出します。
3. アプリケーションコンテナは、要求ハンドラを呼び出して、受け取った要求を解読します。Enterprise Server が処理できる各種のクライアント要求を処理する Micro Focus 製の要求ハンドラがあります。この要求ハンドラは、Interface Mapping Toolkit が作成したマッピング情報を使用して、要求パラメータを COBOL アプリケーションの呼び出しインターフェイスにマップします。
4. COBOL アプリケーションが要求の処理を完了すると、アプリケーションコンテナは要求ハンドラを呼び出して出力パラメータをクライアントが解読できる形式にマップします。
5. アプリケーションコンテナは、応答を通信マネージャに渡します。通信マネージャは、受け取った応答をクライアントに返します。

図 1-2 に、サービス実行プロセスの構成要素を示します。

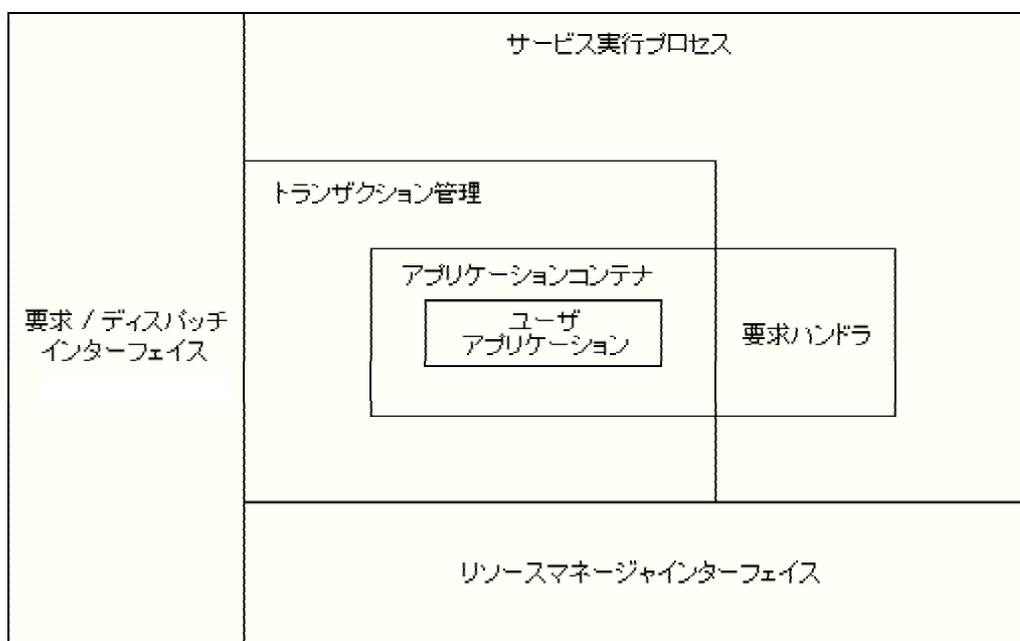


図 1-2： サービス実行プロセスの構成要素

トランザクション管理構成要素は、コンテナ管理サービスのトランザクションを管理します。つまり、COMMIT と ROLLBACK トランザクションを管理します。リソースマネージャインターフェイスは、データベースやファイルアクセスなど、リソースへの要求を処理します。トランザクション管理とリソースマネージャの詳細は、『[リソース管理](#)』を参照してください。

1.6 Directory Server

Directory Server は、サービスと他の関連する構成要素を定義する一連の情報を管理するプロセスです。Directory Server が取り扱う構成要素はオブジェクトと呼ばれます。Directory Server のリポジトリには、オブジェクトに関する情報が格納されます。Directory Server は、次の情報を管理します。

- サーバ
- サービス
- サービスリスナー
- 要求ハンドラ
- 実装パッケージ

Interface Mapping Toolkit を使用して COBOL アプリケーションをサービスとしてデプロイすると、サービスの情報のほとんどが自動的に作成されます。管理者は、必要に応じて情報を編集できます。

サーバオブジェクトは、Directory Server の最上位オブジェクトです。他のオブジェクトは、サーバと切り離すことはできません。図 1-3 に、2 つのサービスがあるエンタープライズサーバを示します。矢印はオブジェクト間の関係を示します。サーバに 1 つ以上のサービスを手動で設定する場合は、Net Express からデプロイするのではなく、すべてのオブジェクトを作成してからオブジェクト間の関係をビルドします。

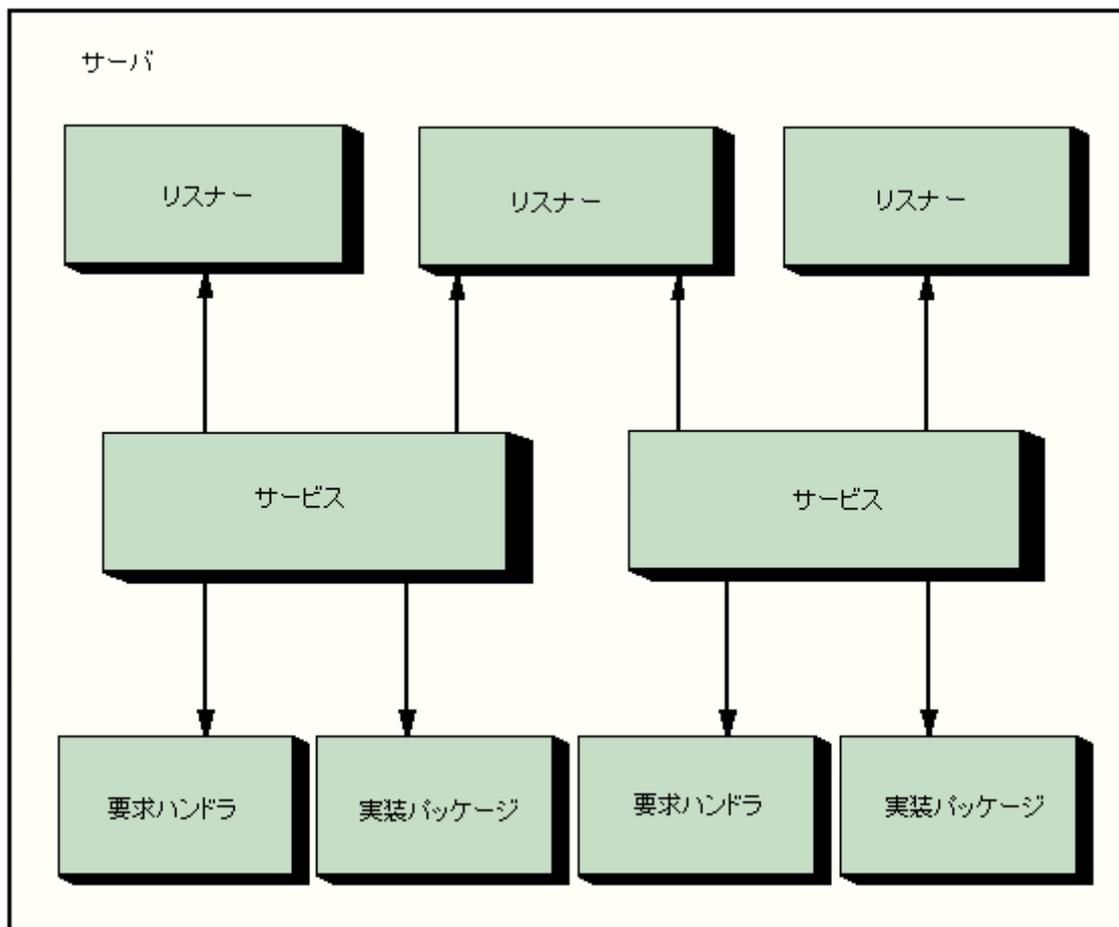


図 1-3 : エンタープライズサーバ内のオブジェクト

デフォルトでは、Directory Server はエンタープライズサーバが実行されているマシンで実行されます。

1.6.1 サーバ

Directory Server は、次の 2 種類のサーバオブジェクトに関する情報を格納します。

- エンタープライズサーバ ([『エンタープライズサーバのアーキテクチャ』](#)を参照)。エンタープライズサーバは、複数のサービスにアクセスできます。

- CCI サーバ。Micro Focus 製の共通通信インターフェイス (CCI) を使用し、CCITCP2 プロセスに登録しないでネーミングサーバとして Directory Server を使用するサービスを提供します。CCI サーバは下位互換性のためにサポートされています。

サーバの詳細は、『[サーバ](#)』の章を参照してください。

1.6.2 サービス

サービスオブジェクトはサービスを定義します。サービスにより、クライアントは、特定のビジネス機能にアクセスすることができます。

サービスには複数のオペレーションを含むことができます。たとえば、計算プログラムでは加算、減算、乗算、および除算を行います。各作業は、クライアントが、その機能へのインターフェイスを理解している場合に使用できる計算サービス内のオペレーションです。

エンタープライズサーバでは、エンタープライズサーバにデプロイするサービスのほかに、1 つ以上のシステムサービスを含むことができます。デフォルトのエンタープライズサーバには、次の 2 つのシステムサービスがあります。

- デプロイサービス。Net Express から起動されているエンタープライズサーバへサービスを自動的にデプロイできます。
- テストサービス。これは、エコーサービスです。Web ブラウザを開いてサービスリスナーのアドレスを入力すると、サーバが起動されている場合には HTML 応答がありません。この HTML 応答は、要求が受信されていることを示します。

ESDEMO 以外のエンタープライズサーバへデプロイする場合は、デプロイサービスが必要になります。

サービスには 1 つ以上のサービスリスナーが必要です。ビジネス機能を提供するサービスの場合は、要求ハンドラが 1 つ、サービスに関連する実装パッケージが 1 つそれぞれ必要です (システムサービスの場合にはハンドラやパッケージは必要ありません)。

サービスの詳細は、『[サービス、パッケージ、および要求ハンドラ](#)』の章を参照してください。

1.6.3 サービスリスナー

サービスリスナーオブジェクトには、サービスのために着信クライアント要求を受信するネットワークアドレスが含まれます。サービスには複数のリスナーを含むことができ、1 つのリスナーでは複数のサービスのためにクライアント要求を受信できます。

各サービスリスナーは 1 つのコネクタに関連付けられています。コネクタは、リスナーが処理する種類の要求を処理するロジックを含むソフトウェアです。

Enterprise Server には次のコネクタを含め、複数のコネクタが用意されています。

- soap コネクタ。Web サービス SOAP 要求と、J2EE 要求のために使用する Micro Focus 製の Binary Protocol の両方を処理します。
- http-switch コネクタ。サービスのディプロイに使用し、簡単な Web サーバ機能を提供します。
- http-echo コネクタ。接続テストに使用でき、HTTP 要求に対して短い HTML 応答を返します。

サービスリスナーの詳細は、『[サービスリスナー](#)』の章および『構成』の章にある『[ディプロイサービスとディプロイリスナー](#)』を参照してください。

1.6.4 要求ハンドラ

要求ハンドラオブジェクトは要求ハンドラを定義します。要求ハンドラは、クライアントからのサーバへのアクセス要求を受信し、要求をビジネス機能を提供するアプリケーションが解読できる形式に変換するプログラムです。アプリケーションが作業を完了すると、要求ハンドラはアプリケーションの出力をクライアントが解読できる形式に変換し、通信サーバを介して出力をクライアントに返します。

SOAP プロトコルと Micro Focus 製のバイナリプロトコル用の要求ハンドラが提供されています。提供されている要求ハンドラでサポートされていないメッセージ形式を処理する要求ハンドラを独自に作成できます。ユーザ出口を使用して、既存の要求ハンドラをカスタマイズすることもできます。

要求ハンドラの詳細は、『[サービス、パッケージ、および要求ハンドラ](#)』の章を参照してください。

1.6.5 実装パッケージ

実装パッケージオブジェクトには、サービスを提供する COBOL アプリケーションの情報が含まれています。

実装パッケージの詳細は、『[サービス、パッケージ、および要求ハンドラ](#)』の章を参照してください。

1.7 リソース管理

データベースやファイルなどの外部リソースを使用するサービスを提供する COBOL アプリケーションは、独自のリソースを管理することも、独自のリソースをアプリケーションコンテナに管理させることもできます。独自のリソースを管理するサービスは、アプリケーション管理サービスと呼ばれます。独自のリソースを管理しないサービスは、コンテナ管理サービスと呼ばれます。

サービスが Fileshare を使用してファイルを更新する場合は、サービスが実行されるエンタープライズサーバに対して fhredir.cfg ファイル名とそのファイルの位置を指定する必要があります。Fileshare の詳細は、『Fileshare ガイド』を参照してください。

1.7.1 アプリケーション管理サービス

サービスは、次の場合に独自のリソースを管理する必要があります。

- アプリケーションが、ファイルやデータベース (またはその両方) をトランザクションを使用してアクセスし、トランザクションロジックを含む場合。つまり、COMMIT と ROLLBACK トランザクションを含む場合。
- アプリケーションが、Enterprise Server の環境でサポートしていないリソース管理を使用する場合。『[コンテナ管理サービス](#)』を参照してください。

アプリケーションがトランザクションを使用しない (つまり、COMMIT と ROLLBACK を含まない) でファイルをアクセスするサービスは、アプリケーション管理サービスです。

トランザクションを使用するアプリケーション管理サービスの場合は、必要なすべてのトランザクションロジックを含む必要があります。つまり、すべてのリソースをコミットまたはロールバックする必要があります。

アプリケーション管理サービスは、Net Express または Application Server などの従来の実行環境でのアプリケーションと同じように実行されます。主な違いのうちの 1 つは、サービスの実行終了時にプロセスが終了しないことです。

1.7.2 コンテナ管理サービス

サービスとして実行しているアプリケーションがデータベースとファイルを使用し、トランザクションロジックを含まない場合は、アプリケーションコンテナに独自のリソースを管理させることができます。

アプリケーションコンテナは、必要なリソースマネージャを事前に知る必要があります。リソースは、サービスごとではなく、エンタープライズサーバごとに定義します。XA 互換のリソースマネージャのみ定義できます。定義できるデータベースリソースマネージャの種類は次のとおりです。

- IBM DB2
- Oracle 8
- Oracle 9

エンタープライズサーバとそのサービス実行プロセスを起動すると、エンタープライズサーバマネージャは、そのエンタープライズサーバ内のサービスに必要なすべてのリソースへの接続を開きます。コンテナ管理サービスを起動すると、エンタープライズサービスマネージャは必要なファイルアクションとデータベースアクションをすべて実行します。

アプリケーションコンテナは、XA コマンドを使用してデータベースを管理します。

アプリケーションコンテナは、サービスの作成時およびサービスの Net Express へのデプロイ時に得た情報を使用して、リソースのコミットまたはロールバックを決定します。詳細は、Net Express に付属の『分散コンピューティング』の『Interface Mapping Toolkit による COBOL

Web Services の作成』の章を参照してください。ただし、アプリケーションコンテナ自体にエラーがある場合には、リソースは必ずロールバックされます。

1.7.3 コンテナ管理サービスとアプリケーション管理サービスの混在

コンテナ管理サービスとアプリケーション管理サービスを同一のエンタープライズサーバ内に混在させることができます。ただし、この環境では、アプリケーション管理サービスはすべてのリソースをコミットまたはロールバックすることがより重要となります。これは、コンテナ管理アプリケーションのためにアプリケーションコンテナがリソースをコミットまたはロールバックする場合に、前回実行されたアプリケーション管理サービスの未処理の更新を含め、未処理のすべての更新をコミットまたはロールバックしてしまうためです。

1.7.4 リソースマネージャとして動作する Enterprise Server

J2EE クライアントは、IBM WebSphere または BEA WebLogic などの J2EE アプリケーションで実行されます。サービスが J2EE クライアントからの要求を処理する場合に、J2EE アプリケーションサーバは、トランザクションに関連するリソースへのすべての更新を管理するトランザクションマネージャとすることができます。この場合には、エンタープライズサーバはリソースマネージャのように動作し、COBOL サーバがコンテナ管理であろうとアプリケーション管理であろうと、J2EE アプリケーションサーバからのコミット要求またはロールバック要求に応答しません。

1.8 Enterprise Server の管理

Enterprise Server Administration ツールを使用して Enterprise Server を構成および管理できます。このツールを使用して、Directory Server が管理するエンタープライズサーバに関する情報を表示および変更できます。Enterprise Server Administration には Web ブラウザからアクセスします。

管理作業は、特別な管理作業と日常の管理作業の 2 つに分類できます。エンタープライズサービスにサービス、サービスリスナー、要求ハンドラ、および実装パッケージをすべて定義した後は、日常の管理作業はそれほどありません。開発者が Interface Mapping Toolkit を使用して Enterprise Server へ直接デプロイする場合は、必要なオブジェクト定義は自動的に作成されます。

特別な管理作業には、次の作業があります。

- Directory Server の構成
- サービス、要求ハンドラなどのオブジェクトの作成
- オブジェクトの削除
- サービスの再デプロイ

日常の管理作業には、次の作業があります。

- サーバの開始と停止

- 必要に応じた オブジェクトの編集
- イベントに対応する他のオブジェクトの状態の変更。たとえば、ネットワーク障害に応じてリスナーを使用不能にします。
- システムのパフォーマンスを最大にするパフォーマンスの監視とシステムのチューニング。たとえば、サーバ内のサーバ実行プロセスの数を増加します。

1.9 他の章について

Directory Server とエンタープライズサーバを構成してから、Enterprise Server を起動します。構成の詳細は、『[構成](#)』の章を参照してください。

Directory Server への Web インターフェイスについては、『[Enterprise Server Administration の概要](#)』を参照してください。

Directory Server が管理するオブジェクトの詳細は、『[サーバ](#)』、『[サービスリスナー](#)』、および『[サービス、パッケージ、および要求ハンドラ](#)』の章を参照してください。

Micro Focus 社が提供している要求ハンドラをカスタマイズする方法は、『[要求ハンドラのユーザ出口](#)』の章を参照してください。

第 2 章：構成

ここでは、インストール後の Enterprise Server の構成方法について説明します。この構成では、Directory Server と個別のエンタープライズサーバの両方の属性も設定します。また、パフォーマンスの考慮事項についても説明します。

2.1 はじめに

Enterprise Server を初めて起動する場合は、次の処理が必要です。

1. Directory Server のセキュリティを設定する
2. Directory Server の他の構成オプションを設定する
3. エンタープライズサーバの構成オプションを設定する

Enterprise Server をインストールすると、デフォルトのエンタープライズサーバ ESDEMO が 1 つ作成されます。このサーバはディプロイシステムサービスを提供します。Interface Mapping Toolkit を使用して COBOL サービスを自動的にディプロイすると、ディプロイサービスはサービスを受信し、サービスとその構成要素 (オペレーションとパッケージ) をエンタープライズサービスに追加します。ディプロイサービスおよび関連するリスナーの構成が必要になる場合があります。詳細は、『[ディプロイサービスとディプロイリスナー](#)』を参照してください。

2.2 Directory Server のセキュリティの設定

Directory Server は、セキュリティを使用可能または使用不能にして実行できます。セキュリティを使用不能にして Directory Server を実行すると、すべてのユーザはエンタープライズサーバと Directory Server に変更を加えることができます。常にセキュリティを使用可能にして Directory Serverity を実行することをお奨めします。

セキュリティは、ユーザ ID とパスワードを使用することで提供されます。各ユーザ ID にはアクセス権レベルが関連付けられています。Directory Server をインストールした直後はセキュリティが使用不能になっています。一部のユーザ ID はセキュリティを使用可能にした場合にのみ使用するために提供されています。

管理者としての最初の作業は、セキュリティを使用可能にし、ユーザのログオン情報を作成することです。

セキュリティを使用可能にするには、Enterprise Server Administration への Web インターフェイスを開始する必要があります。次に、「構成オプション」ページで「ユーザアクセス制限」を選択します。

[方法](#)

セキュリティを使用可能にすると、ユーザ ID とパスワードの入力がすぐに求められます。次の構成済みのログオン情報を入力します。

「ユーザ ID:」 schemaadmin
「パスワード:」 schemaadmin

ユーザ ID schemaadmin には最も高いアクセス権レベル Schema Administrator が割り当てられているため、このユーザ ID の不正使用を防ぐ必要があります。schemaadmin のパスワードはすぐに変更してください。ユーザ ID schemaadmin を保護した後に、Schema Administrator アクセス権レベルをもつ新しいユーザ ID の作成が必要になる場合があります。他のアクセス権レベルを割り当てたユーザ ID の新規作成や、すでに定義されているユーザ ID の削除も必要になる場合があります。

[方法](#)

セキュリティとアクセス権レベルの詳細は、『[Enterprise Server Administration の概要](#)』の章を参照してください。

2.3 Directory Server

Directory Server のさまざまな属性を構成できます。たとえば、エンタープライズサーバが使用可能かどうかの監視、ジャーナルに記録するイベントの種類、Directory Server のクライアントがどのくらいの時間非アクティブの場合に自動的にログオフするかなどがあります。これらのオプションの詳細は、「構成オプション」ページのヘルプを参照してください。

[方法](#)

2.4 Enterprise Server

エンタープライズサーバのさまざまな属性を構成できます。たとえば、使用可能なメモリ量、起動するサービス実行プロセスの数、トレースする情報などがあります。ほとんどの属性は、適切な値に設定されています。これらの属性の詳細は、「サーバの追加」ページと「サーバの編集」ページのそれぞれのヘルプを参照してください。

[方法](#)

以後では、一部のさらに複雑な属性について詳細に説明します。

2.4.1 共有メモリ領域

共有メモリ領域は、エンタープライズサーバがその実行に必要なすべての情報を格納するメモリ領域です（『はじめに』の章にある『エンタープライズサーバのアーキテクチャ』の図を参照）。共有メモリ領域のサイズは、ページ数で表します（1 ページは 4 KB）。共有メモリのサイズは、サーバ内のオブジェクトのすべての定義と現在のクライアント要求をすべて格納するのに十分な大きさである必要があります。サーバの実行中に共有メモリが不足した場合は、サーバのパフォーマンスが著しく低下し、クライアント要求の処理が妨げられます。

エンタープライズサーバに割り当てる共有メモリ領域のページ数の概算は、ここで説明する計算式を使用してください。

実際の共有メモリの要件は作業負荷によって異なりますが、安全を考えて大きめに割り当ててください。Enterprise Server は、処理量が最大でない限り物理ページ数の使用を最小にすることで共有メモリのリソースの使用を最適化します。

次の表の中央カラムに示した計算式を使用して、それぞれの計算結果を右側のカラムに書き込みます。右側のカラムに書き込んだ値の合計が共有メモリの最小必要バイト数です。合計バイト数を 4096 で除算し、その結果の端数を切り上げて、必要な共有メモリのページ数を求めます。

項目	計算式	計算結果
オーバヘッド	固定サイズ	8192
共有メモリ領域管理	共有メモリ領域サイズ / 4096	
トレース	エントリ数 × 24	
ローカルトレース	サービス実行プロセスの数 × (エントリ数 × 24)	
サービス	サービスの数 × (128 + service-name-length)	
サービス実行プロセス	サービス実行プロセスの数 × 128	
要求ハンドラ	要求ハンドラの数 × (128 + handler-name-length)	
パッケージ	パッケージの数 × (128 + IDT-name-length + application-path-length + module-name-length)	
常駐 IDT	常駐 IDT の数 × IDT-length	
クライアント要求	同時に処理されるクライアント要求の数 × (256 + クライアント要求の平均サイズ)	
合計		

変数の内容は、次のとおりです。

service-name-length は、サービスの名前の長さです。

handler-name-length は、要求ハンドラの名前の長さです。

IDT-name-length は、IDT の名前の長さです。

application-path-length は、パッケージ内の COBOL アプリケーションへのパスの長さです。

module-name-length は、アプリケーションを格納しているモジュールの名前の長さです。

IDT-length は、パッケージの IDT ファイルのサイズです。

2.4.2 共有メモリクッション

共有メモリクッションは、共有メモリ領域の一部です。共有メモリクッションの機能は、短期間に集中するサーバの要求を処理します。エンタープライズサーバが開始されたときは、共有メモリクッションは使用不能です。着信クライアント要求では使用されません。応答をクライアントに返す必要があるときに、その応答を処理する十分な空き領域が共有メモリにない場合にのみ使用されます。

共有メモリクッションのサイズは、ページ数で表します (1 ページは 4 KB)。共有メモリクッションのサイズは、共有メモリ領域サイズの 10 % に設定してください。

2.4.3 サービス実行プロセスの数

エンタープライズサーバには、1 つ以上のサービス実行プロセスがあります (『はじめに』の章にある『[サービス実行プロセスの構成要素](#)』の図を参照)。サービス実行プロセスは着信クライアント要求を処理します。応答がクライアントに返されると、サービス実行プロセスは別の要求を処理できます。すべてのサービス実行プロセスが要求を処理中である場合は、どれかのサービス実行プロセスが要求の処理を完了するまで、着信要求の処理は待機します。エンタープライズ内のサービス実行プロセスの数は、パフォーマンスに影響します。影響の程度はサービス実行プロセスの数によって異なります。特定のエンタープライズサーバに最適なサービス実行プロセスの数を見つけ出す必要があります。

サービス実行プロセスの実行中にサービス実行プロセスの数を変更できます。ただし、その変更はサーバが停止されるまで有効です。サーバを停止すると、サービス実行プロセスの数はサーバを追加したときの値またはサーバの詳細を編集したときの値に戻ります。

2.4.4 環境変数

「サーバの追加」ページまたは「サーバの編集」ページの「構成情報」で環境変数を設定できます。環境変数はサーバ内で実行されているすべてのサービスに適用されます。ただし、Interface Mapping Toolkit を使用してサービスの作成時に設定したサービスレベルの環境変数は、サーバレベルの設定を上書きします。

エンタープライズサーバ内のどれかのサービスが Fileshare を使用してネットワーク上のファイルをアクセスする場合は、FHREDIR 環境変数を設定する必要があります。

環境変数の形式は、次のとおりです。

[ES-Environment]

environment-variable-name=environment-variable-setting

次に使用例を示します。

```
[ES-Environment]
FHREDIR=c:¥client.cfg
```

[方法](#)

2.4.5 リソースマネージャ

エンタープライズサーバ内にデータベースにアクセスするコンテナ管理サービスがある場合は、アプリケーションコンテナが対話する必要があるリソースマネージャの情報を指定する必要があります。XA 互換のリソースマネージャのみ使用できます。XA インターフェイスは、スイッチテーブルと呼ばれる構造体を記述します。スイッチテーブルは、リソースマネージャに実装された xa_ ルーチンの名前を保持します。この構造体の名前は xa_switch_t です。Micro Focus 社は、リソースマネージャが実装した XA スイッチデータ構造体のアドレスを取得するために、スイッチモジュールをいくつか提供しています。

Net Express では、Oracle および IBM DB2 用のスイッチモジュールのソースファイルがディレクトリ install-dir¥base¥source¥enterpriseserver¥xa に格納されています。次のソースファイルがあります。

esora8xa.cbl	Oracle 8
esora9xa.cbl	Oracle 9
esdb2bxa.cbl	IBM DB2

このディレクトリには、必要なスイッチモジュールをビルドするために使用できるバッチファイル build.bat も格納されています。バッチファイルを実行するコマンドは、次のとおりです。

```
build database-name
```

database-name には、次のどれかを指定できます。

ORA8	Oracle 8
ORA9	Oracle 9
DB2	IBM DB2

DB2 用のスイッチモジュールを作成する場合は、LIB 環境変数に DB2 LIB ディレクトリへのパスが含まれていることを確認する必要があります。

Enterprise Server を Net Express の一部としてではなくスタンドアロンで実行している場合は、Net Express で必要なスイッチモジュールをビルドし、スタンドアロンの Enterprise Server で使用できるようにする必要があります。

Web インターフェイスの「サーバ詳細の編集」ページで 1 つのリソースマネージャの詳細を指定できます。

方法

「サーバの追加」ページまたは「サーバの編集」ページの「構成情報」で複数のリソースマネージャの詳細を指定できます。形式は、次のとおりです。

```
[ESXRM]
rmtype=xa
rmlabel=resource-manager-id
rmname=resource-manager-name
rmmodule=switch-module-name
rmopenstring=open-string
rmclosestring=close-string
```

パラメータの内容は、次のとおりです。

resource-manager-id	特定の XA 構成を識別するために内部的に使用するリソースマネージャ ID です。リソースマネージャ ID は、エンタープライズサーバ内で一意である必要があります。
resource-manager-name	リソースマネージャを認識する名前です。xa_switch_t 構造体の「name」フィールドで返された名前と一致する必要があります。
switch-module-name	xa_switch_t 構造体をエンタープライズサーバに返すエントリポイントを含むスイッチモジュール実行可能ファイル (.dll) の位置です。
open-string	xa_open() 呼び出しでリソースマネージャに渡される文字列です。通常、文字列には少なくともデータベース名、およびデータベースに接続するユーザ ID とパスワードが含まれます。この文字列の内容は、データベース固有です。 DB2 の場合は、文字列の内容については、IBM 社の『DB2 管理の手引き』を参照してください。詳細は、『アプリケーション・プログラミングの手引き』を参照してください。 Oracle の場合は、『Oracle アプリケーション開発者ガイド 基礎編』を参照してください。 リソースマネージャが動的登録をサポートし、ax_reg 関数を提供するモジュールの名前を要求するデータベースベンダの場合は、casaxlib を指定します。
close-string	xa_close() 呼び出しでリソースマネージャに渡される文字列です。close-string を指定する必要があるかどうかは、データベースベンダのマニュアルを参照してください。

次に使用例を示します。

```
[ESXRM]
rmtype=xa
```

```
rmlabel=XADB2
rmname=DB/2
rmmodule=c:%netexpress%base%source%enterpriseserver%xa%esdb2xa.dll
rmopenstring=db=sample,uid=myuser,pwd=myspasswd,axlib=casaxlib
rmclosestring=
```

[方法](#)

2.5 パフォーマンスに関する考慮事項

パフォーマンスを向上するためにエンタープライズサーバをどのように構成するかは、いくつかの要因によって決まります。明らかな要因の 1 つは、予期されるサーバの作業負荷です。クライアント要求が着信する頻度と、これらの要求に対して求められている応答速度を考慮する必要があります。

他の 2 つの要因は、サーバ内で実行するサービスの種類とサービスの要求の種類です。サービスは、I/O バウンドまたは CPU バウンドに分類できます。クライアント要求は、短期実行または長期実行に分類できます。

2.5.1 I/O バウンドサービス

多数の入出力要求を実行するサービスはその実行時に、要求の応答を待機する間は休止した状態になります。ただし、入出力要求の処理では CPU の負荷を考慮すべき場合があります。I/O バウンドサービスの場合には、サービス実行プロセスの数を増加すべきか検討する必要があります。

2.5.2 CPU バウンドサービス

入出力要求を実行しない (またはほとんどない) サービスでは、一般に CPU の負荷が高くなります。通常、I/O バウンドサービスの場合には、サービス実行プロセスの数が少ないほどよく動作します。実行されている作業間での CPU の切り替えによるオーバーヘッドが小さくなるためです。

2.5.3 短期実行クライアント要求

短期実行クライアント要求は、クライアントとサービスの間で 1 回のみ行われる要求です。クライアント要求が着信し、サービスが実行され、応答がクライアントに返されます。Web サービスクライアントからの要求は常に短期実行です。

短期実行クライアント要求の場合には、サービス自体が I/O バウンドか、または CPU バウンドかのみを考慮する必要があります。

2.5.4 長期実行クライアント要求

長期実行クライアント要求は、同じクライアントがサービスを繰り返し要求し、サービスのそれぞれの起動の間データを保持する必要がある場合の要求です。Java 用語では、これらの要求はステートフル要求です。クライアントが WebLogic または WebSphere などのアプリケーションサーバで実行されているステートフル Java bean の場合には、サービスは Java bean が実行されている間実行されます。このような状況では、サービス自体が I/O バウンドではなく CPU バウンドの場合でもサービス実行プロセスの数を増加する必要があります。

エンタープライズサーバが要求がステートフルであることを検出すると、新しいサービス実行プロセスが自動的に開始されます。サービス実行プロセスの数が動的に増加することでマシンのリソースが消費され、エンタープライズサーバのパフォーマンスが影響されます。アプリケーションで実際にステートフルな相互運用が必要ない限り、長期実行クライアント要求の使用は最小にすることをお奨めします。

2.6 デプロイサービスとデプロイリスナー

サービスを実行中のエンタープライズサーバに自動的にデプロイする Net Express Interface Mapping Toolkit の機能は、デプロイサービスと呼ばれるシステムサービスを使用してサービスを実行します。エンタープライズサーバ ESDEMO には、「Deployer」と呼ばれるデフォルトのデプロイサービスがあります。このデプロイサービスを変更することも、異なる構成でデプロイサービスを新規に作成することもできます。また、作成したエンタープライズサーバにデプロイサービスを追加することもできます。

方法

Deployer サービスは、Web と呼ばれるリスナーを使用します。Web リスナーでは、http-switch コネクタを使用します。Deployer サービスを変更する場合は、Web リスナーの構成の変更が必要になることもあります。デプロイサービスで使用する独自のリスナーを作成することもできます。

方法

2.6.1 デプロイサービス

すべてのデプロイサービスでは属性を、次のように設定する必要があります。

Service Class	MF deployment
Conversation Type	http-switch

デプロイサービスの構成情報は、次のようになります。

```
[MF client]
  scheme=http
  URL=virtual-directory-name-1/virtual-directory-name-2
  accept=application/x-zip-compressed
```

virtual-directory-name-1 は、ディプロイプログラム mfdeploy.exe を格納しているディレクトリです。virtual-directory-name-2 は、ディプロイ済みサービスを含むディレクトリです。

scheme パラメータと accept パラメータの値は変更できません。

ESDEMO の Deployer サービスの設定は、次のとおりです。

```
[MF client]
  scheme=http
  URL=/cgi/mfdeploy.exe/uploads
  accept=application/x-zip-compressed
```

ディプロイサービスの構成は、サービスのリスナーの構成と一致する必要があります。URL パラメータには、mfdeploy.exe プログラムの位置として、リスナーが使用する仮想ディレクトリと同じ名前を指定する必要があります。ディプロイ済みサービスを格納する仮想パスを、リスナーに対して構成された仮想ディレクトリ名に変更できます。

たとえば、次のようなリスナーの構成があります。

```
[virtual paths]
  <default>=/dev/null
  netexpress=<ASEE>/bin
  project1=c:/development/project1
  project2=c:/development/project2
```

「project1」という名前のディプロイサービスをこの構成で作成します。

```
[MF client]
  scheme=http
  URL=/netexpress/mfdeploy.exe/project1
  accept=application/x-zip-compressed
```

「project1」サービスを使用してディプロイしたサービスは c:/development/project1 に格納されます。project2 に対して同じようなサービスを作成できます。

ディプロイディレクトリには、.mfdeploy ファイルが存在する必要があります。必要に応じて、install-dir¥base¥deploy から .mfdeploy ファイルをコピーして変更してください。

2.6.2 ディプロイリスナー

ディプロイリスナーの構成情報には、「[virtual paths]」という名前のセクションがあり、URL の上位レベルディレクトリのリストと変更後のパスが後に続きます。たとえば、ESDEMO の Deployer サービスが使用する Web リスナーの構成は、次のようになります。

```
[virtual paths]
  cgi=<ASEE>/bin
```

```
uploads=<ASEE>/deploy
<default>=/dev/null
```

仮想パス「cgi」は、デプロイされる COBOL アーカイブファイルを格納する mfdeploy.exe プログラムの位置を指定します。仮想パス「uploads」は、アップロードされた COBOL アーカイブファイルを格納するディレクトリを mfdeploy.exe プログラムに示します。特別なトークン「<ASEE>」は、Enterprise Server の基本インストールディレクトリに変換されます。たとえば、Enterprise Server を c:\Programs\NetExpress にインストールした場合は、「<ASEE>/deploy」はデフォルトのデプロイ先ディレクトリである c:\Programs\NetExpress\base\deploy に変換されます。

URL の最初のディレクトリのみがリストと比較されるため、リスト内のエンタリは 1 つのディレクトリ名にする必要があります。

「<default>」ディレクトリは、URL 内の最初のディレクトリがリスト内のどのエンタリとも一致しない場合に使用されます。<default> ディレクトリを指定することは重要です。指定しない場合には、サーバは現在のドライブのルートディレクトリを使用し、ネットワーク上のそのドライブのファイルをすべて公開します。

別の使用例を次に示します。

```
[virtual paths]
<default>=c:/web
docs=c:/web/documents
images=d:/media/images
```

この構成では、URL <http://host:port/docs/a.html> はファイル c:/web/documents/a.html を返し、URL <http://host:port/images/gif/b.gif> はファイル d:/media/images/gif/b.gif を返します。

第 3 章 : Enterprise Server Administration の概要

ここでは、Directory Server への Web インターフェイスの使用方法について説明します。この Web インターフェイスは、Enterprise Server Administration と呼ばれます。

3.1 Enterprise Server Administration の最初の使用

Enterprise Server Administration を一般的に使用するには、構成を行う必要があります。詳細は、『[構成](#)』の章を参照してください。

3.2 Web インターフェイスのアクセス

Enterprise Server をインストールすると、Directory Server が Win32 サービスとして自動的に実行されます。Web インターフェイスを使用して Directory Server にアクセスするには、Web ブラウザを起動して Directory Server が実行されているアドレスを入力します。Directory Server の構成と行う作業によっては、Enterprise Server Administration にログオンする必要があります。ログオンするには、ユーザ ID とパスワードが必要です。

Directory Server は、Windows のコントロール パネルを使用して、停止および再起動することができます。

[方法](#)

3.3 アクセスレベルとアクセス権レベル

アクセス可能なレベル、つまり Enterprise Server Administration を使用して実際に行える作業は、次の 2 つの要因によって決まります。

- 「構成オプション」ページの「ユーザアクセス」の設定。設定する値は次のどちらかです。

Restricted Enterprise Server Administration を使用するには、ログオンする必要があります。

Unrestricted ログオンすることなく、すべてのユーザは Enterprise Server Administration が許可するアクションを実行できます。

- ログオン時に指定するユーザ ID。各ユーザ ID には、実行可能なアクションを指示するアクセス権レベルが関連付けられています。次の 4 つのアクセス権レベルがあります。

Modify Directory Server リポジトリに格納されている値を変更できます。

Add/Delete	Modify レベルに加えて、オブジェクトを追加および削除できます。通常、このレベルがプログラムクライアントに必要な最高レベルのアクセス権となります。
Administrator	Add/Delete レベルに加えて、リポジトリのバックアップと復元、リポジトリへのオブジェクトのインポート、および Directory Server のシャットダウンができます。
Schema Administrator	Administrator レベルに加えて、ユーザの追加、変更、削除、およびリポジトリの削除ができます。

Directory Server を Restricted モードで実行する場合は、Schema Administrator アクセス権をもつユーザはすべてのユーザに対してユーザ ID とそのパスワードを作成し、常に最新のユーザ情報を管理する必要があります。

方法

3.4 実際の画面

ログインすると、図 3-1 に示すホームページが表示されます。

Enterprise Server Administration
Version 1.00.87
TECH-CPQ-1.microfocus.co.jp (10.18.11.122)

Status | MDS00001 OK | Wed Jul 23 15:55:46 2003
[SCHEMA ADMINISTRATOR-Developer] [Page id: 1000]

画面更新 1ページに 10 件を表示 << 前へ 次へ >>

1 - 1 / 1 サーバー								
	タイプ	名前	バージョン	アドレス	現ステータス	ステータスログ	オブジェクト	説明
編集...	MFES	ESDBMO	1.1.0	top:10.18.11.122*:9000 (TECH-CPQ-1.microfocus.co.jp)	開始 Details	OK	<ul style="list-style-type: none"> 3 リスナー 詳細 2 サービス 詳細 2 ハンドラ 詳細 0 パッケージ 	Communications server for Web Services

追加...

図 3-1 : 「Enterprise Server Administration」のホームページ

3.4.1 メニュー

Enterprise Server Administration メニューはホームページの左側パネルに表示されます。メニューオプションは、アクションなどの見出しの下部にグループ化されています。ユーザのア

アクセス権レベルに該当するメニューオプションのみが表示されます。Enterprise Server Administration のすべてのアクションをメニューを使用して行う必要はありません。多くのアクションはテーブル内のプッシュボタンで利用できます。

3.4.2 ステータス行

ステータス行は、Enterprise Server Administration 見出しの下部で、ページの上部に表示されます。ステータス行には、Directory Server の状態についての情報が表示されます。エラーメッセージは赤色で表示されます。

ステータス行に表示されるメッセージの詳細は、Enterprise Server の次のヘルプトピックを参照してください。『[Directory Server エラーメッセージ](#)』

3.4.3 テーブル

Directory Server のオブジェクトの情報はテーブル内に表示されます。ログオン後に最初に表示されるテーブルには、Directory Server リポジトリのサーバについての情報が表示されます。このテーブルは、要約モードまたは展開モードのどちらかで表示されます。要約モードでは、サーバ内の各オブジェクトの数が表示されます。展開モードでは、各オブジェクトの情報が表示されます。どちらのモードにも、特定の種類のオブジェクトに関する詳細を示すページに移動できる、いくつかの [詳細] プッシュボタンがあります。

ほとんどのテーブルには、選択したオブジェクトの属性を更新できるページに移動できる [編集] ボタンが 1 つ以上、選択した種類のオブジェクトを追加できる [追加] ボタンが 1 つ表示されます。

テーブルには、白色のカラムとベージュ色のカラムがあります。白色のカラムの情報はエンタープライズサーバと CCI サーバのどちらにも適用され、ベージュ色のカラムはエンタープライズサーバにのみ適用されます。CCI サーバの情報は、『サーバ』の章にある『[はじめに](#)』を参照してください。

3.4.4 ナビゲーション

Enterprise Server Administration には 2 つの機能があり、ナビゲーションモードはそれぞれ異なります。

- 左側パネルのメニュー項目を選択して、Directory Server 情報の保存や復元、ユーザの新規作成などのさまざまな管理作業を実行します。選択可能なメニュー項目はアクセス権限レベルによって異なります。
- ページの中央にあるボタン (通常は、現在表示されているテーブル内のボタン) をクリックしてリポジトリ内のオブジェクトを操作します。たとえば、新しい要求ハンドラを追加する場合は、次の作業を行う必要があります。
 1. 要求ハンドラのテーブルを表示するには、ホームページ内のサーバのテーブルで「要求ハンドラ」の [詳細] をクリックします。

2. 「要求ハンドラの追加」ページに進むには、要求ハンドラのテーブルの左下で [ハンドラの追加] をクリックします。

ほとんどのページの上部には一連のボタンが表示されます。これらのボタンをクリックして、サーバ、リスナー、サービス、要求ハンドラ、実装パッケージのそれぞれのページに進むことができます。

左側パネルの上部のメニューで [ホーム] をクリックすると、いつでもホームページに戻ることができます。

警告: 前のページに戻るために、Web ブラウザのツールバーにある [戻る] ボタンを使用することはお奨めしません。[戻る] ボタンを使用すると、すでに有効でなくなった管理要求を再送することがあります。

3.4.5 ページの最新表示

ほとんどのページには、[画面更新] ボタンがあります。アクションを実行した後にページのテーブルの内容を最新表示する必要がある場合は、このボタンをクリックします。たとえば、サーバを追加した場合は、ホームページの [画面更新] ボタンをクリックすると、追加したサーバがサーバのテーブルに表示されます。

第 4 章 : Directory Server リポジトリ

ここでは、Directory Server リポジトリについて説明します。

4.1 はじめに

Directory Server リポジトリは、Directory Server が管理するオブジェクトの情報を格納する一連のファイルです。

リポジトリには、Directory Server が管理する次の種類のオブジェクトすべてに関する情報を保持します。

- サーバ
- サービス
- サービスリスナー
- 要求ハンドラ
- 実装パッケージ

リポジトリファイルを使用可能な状態に保ち、常に安全な状態にすることは重要です。変更を行いその変更を失いたくない場合は、必ずリポジトリを定期的に (できれば遠隔地に) 保存してください。システム障害が原因でリポジトリ内のデータを失った場合には、最新の保存位置からデータを復元できます。登録されているすべてのサーバステータスが「停止」になった場合のみにリポジトリを保存することをお奨めします。そうでない場合にリポジトリを保存すると、事実上は実行中のシステムのスナップショットを保存することになります。この最新の保存位置からデータを復元する場合、スナップショットがサーバの実際の状態と必ずしも一致しないことがあります。たとえば、いくつかのサーバが実行されていない可能性があります。復元する場合には、データが保存されたときのサーバの状態を無視して、サーバステータスを「停止」としてデータを復元するオプションがあります。

特定のサーバに変更を加えた場合は、対象のサーバとそのオブジェクトに関する情報のみを保存できます。

4.2 リポジトリ情報

リポジトリに関する情報のいくつかの項目は「統計」ページに表示されます。

- 「合計」 - リポジトリにある現在のオブジェクトの総数。
- 「リポジトリステータス」 - リポジトリの現在の状態です。たとえば、Directory Server の起動時にリポジトリから Directory Server にロードされたオブジェクトの数。
- 「リポジトリロケーション」 - リポジトリが格納されているディレクトリ。デフォルトでは、Net Express の一部として Enterprise Server をインストールした場合は `install-dir¥base¥mfds¥`、Enterprise Server をスタンドアロンでインストールした場合は `install-dir¥mfds¥` です。

4.3 実行できる作業

Administrator アクセス権またはそれ以上のアクセス権レベルをもつ場合は、次の作業ができます。

- リポジトリを別ディレクトリに保存する。
- 現在のエンタープライズサーバの情報のみを別ディレクトリに保存する。
- バックアップからリポジトリを復元する。現在のリポジトリ全体が置換されます。Net Express の一部として Enterprise Server をインストールした場合、および Enterprise Server をスタンドアロンでインストールした場合は、install-dir¥base¥mfds¥ivp および install-dir¥mfds¥ivp からそれぞれ Enterprise Server がインストールされた初期状態にリポジトリを復元できます。
- バックアップからオブジェクトを現在のリポジトリにインポートする。リポジトリに存在しないオブジェクトのみをインポートできます。重複するオブジェクトは無視されます。

さらに、Schema Administrator アクセス権レベルをもつ場合には、リポジトリ全体を削除できます。

警告: リポジトリ全体を削除したり復元したりすると、クライアントアプリケーションで接続エラーが発生したり、データが失われたりすることがあります。

[方法](#)

第 5 章：サーバ

ここでは、サーバオブジェクトと実行できる作業について説明します。

5.1 はじめに

サーバは、Directory Server が管理する最上位レベルのオブジェクトです。サーバには 2 つの種類があります。

- Micro Focus Enterprise Server

いくつかの異なる種類のクライアントが発行したサービス要求に対応するために実行されている COBOL プログラムの実行環境を提供するサーバです。

エンタープライズサーバの詳細は、『[はじめに](#)』の章を参照してください。

COBOL 開発システムと同一のシステムで実行されている場合には、エンタープライズサーバはデバッグモードで実行できます。Enterprise Server にデプロイされたアプリケーションをデバッグする方法の詳細は、COBOL 開発システムのマニュアルを参照してください。

- Micro Focus CCI Server

Micro Focus 製の共通通信インターフェイス (CCI) を使用し、CCITCP2 プロセスに登録しないでネーミングサーバとして Directory Server を使用するサービスを提供するサーバです。CCI サーバを新規作成したり、CCI サーバの状態を変更したりできません。Directory Server と同一のネットワーク上で CCI サーバが実行され、ネーミングサーバとして Directory Server を使用するように構成されている場合には、CCI サーバが実行時のサーバテーブルに「登録済み」という状態で表示されます。CCI サーバが停止すると、サーバテーブルから CCI サーバが削除されます。CCI の詳細は、『CCI の設定』マニュアルを参照してください。

管理者としての作業は、次のとおりです。

- 必要に応じて、サーバを追加、更新、または削除する
- サーバの開始と停止
- サーバの状態を監視し、問題を調査する

5.2 サーバ情報

「Application Server Administration」のホームページのサーバテーブルに、リポジトリに格納されているサーバの情報が表示されます。すべてのサーバに共通の情報は白色の背景で表示されます。エンタープライズサーバに適用される情報はベージュ色の背景で表示されます。

サーバテーブルは、要約モードまたは展開モードのどちらかで表示されます。Enterprise Server Administration を初めて起動すると、サーバテーブルは要約モードで表示されます。サーバテーブルの左上部にあるアイコンをクリックすると、展開モードに切り換わります。要約モードのサーバテーブルについては、『Enterprise Server Administration の概要』の章にある『「Enterprise Server Administration」のホームページ』の図を参照してください。図 5-1 に、展開モードのサーバテーブルを示します。

Enterprise Server Administration
Version 1.00.87
TECH-CPQ-1.microfocus.co.jp (10.18.11.122)

Status: MDS00001 OK Wed Jul 23 15:41:34 2003

[SCHEMA ADMINISTRATOR-Developer] [Page id: 1000]

画面更新 1ページに 10 件を表示 << 前へ 次へ >>

1 - 1 / 1 サーバー								
	タイプ	名前	バージョン	アドレス	現ステータス	ステータスログ	オブジェクト	説明
編集...	MFES	ESDBMD	1.1.0	bop:10.18.11.122:9000 (TECH-CPQ-1.microfocus.co.jp)	開始 Details	OK	<ul style="list-style-type: none"> 3 リスナー 詳細 2 サービス 詳細 2 ハンドラ 詳細 0 パッケージ 	Communications server for Web Services
追加...								

図 5-1：展開モードのサーバテーブル

5.2.1 すべてのサーバに適用される情報

ホームページのサーバテーブルに表示される次の情報は、両方の種類のサーバに適用されます。

- 要約モード
 - 「タイプ」 - サーバの種類。
 - 「名前」 - サーバ名。
 - 「アドレス」 - サーバが使用するネットワークアドレス。CCI サーバはこのアドレスでクライアント要求を受信します。エンタープライズサーバはこのアドレスを使用して、Directory Server からの管理メッセージを受信します。
 - 「現ステータス」 - 現在の状態。
- 展開モード
 - 「前回のステータス変更」 - 最後に状態が変更された時刻。

5.2.2 エンタープライズサーバに適用される情報

ホームページのサーバテーブルに表示される次の情報は、エンタープライズサーバのみに適用されます。

- 要約モード
 - 「バージョン」 - エンタープライズサーバの実行可能バージョンを示します。サーバが起動されると、この情報が更新されます。
 - 「ステータスログ」 - このサーバで発生した最新のイベントについての情報を表示します。
 - 「オブジェクト」 - このサーバに対して登録されている各種類のオブジェクト数のリストです。
 - 「説明」
- 展開モード
 - 「バージョン」 - エンタープライズサーバの実行可能バージョンを示します。サーバが起動されると、この情報が更新されます。
 - 「ステータスログ」 - このサーバで発生した最新のイベントについての情報を表示します。
 - 「サービスリスナー」 - このサーバに対して登録されているすべてのサービスリスナーのリストです。
 - 「デプロイサービス」 - このサーバにデプロイされたサービスのリストです。最大でリスト内の5つのサーバが表示されます。6つ以上のサーバがある場合は、[詳細] をクリックしてリスト全体を表示します。
 - 「要求ハンドラ」 - このサーバに対して登録されているすべての要求ハンドラのリストです。
 - 「実装パッケージ」 - このサーバに対して登録されているすべての実装パッケージのリストです。
 - 「説明」

サーバが実行中の場合にはそのステータスは「開始」となり、サーバに関連付けられたすべての使用可能なオブジェクトには、展開モードのサーバテーブルでオブジェクトの次にチェックマークが付き、実行可能であることが示されます。

5.3 サーバステータス

エンタープライズサーバステータスは、次のどれかです。

停止	サーバは現在実行されていません。
停止中	サーバの終了処理が現在実行中です。通常、これは一時的なステータスで、すぐに「停止」のステータスに切り替わります。
開始	サーバは現在実行中で、クライアント要求とサーバ要求に応答できます。
開始中	サーバの開始処理が現在実行中です。通常、これは一時的なステータスで、すぐに「開始」のステータスに切り替わります。
応答なし	サーバの前のステータスは「開始」でしたが、サーバモニタはその後サーバにアクセスできません。ネットワークの接続エラーが発生したか、またはステータスが「停止」とマーク付けされる前にサーバが異常終了した可能性があります。

ネットワーク通信が復元され、サーバが実行中でサーバモニタが実行されてい

る場合は、サーバモニタが次にサーバがアクティブな状態にあるかチェックする際にサーバステータスは「開始」にマーク付けされます。

サーバで回復不能な通信エラーが発生し、サーバプロセスがアクティブな状態の場合は、「サーバの詳細」ページで [サーバー停止] をクリックしてサーバをシャットダウンできます。

サーバで回復不能な通信エラーが発生し、サーバプロセスがシャットダウンした状態の場合は、「サーバの詳細」ページで [サーバー開始] をクリックしてサーバを開始できます。

無効 どの状態にも該当しないエラーが発生しました。

CCI サーバステータスは常に「登録済み」です。この状態は、CCI サーバの起動が Directory Server に通知されたことを示します。CCI サーバが停止すると、登録が解除されてサーバのリストから削除されます。

5.4 実行できる作業

実行できる作業は、ユーザのアクセス権レベルのみでなく、サーバの種類によっても異なります。サーバが起動されると、Add/Delete またはそれ以上のアクセス権をもつ場合はモニタ機能が利用できます。

5.4.1 エンタープライズサーバ

Modify またはそれ以上のアクセス権をもつ場合は、次の作業ができます。

- サーバの属性の編集

Add/Delete またはそれ以上のアクセス権をもつ場合は、次の作業もできます。

- サーバの追加
- サーバの開始
- サーバの停止
- サーバへのサービスリスナーの追加
- サーバへのサービスの追加
- サーバの削除
- サーバのコピー。サーバをコピーする場合は、サーバに関連付けられたすべてのオブジェクト (サービスリスナー、サービス、要求ハンドラ、および実装パッケージ) もコピーします。
- サーバの実行時詳細の表示
- サービス実行プロセス数の変更
- サーバのコンソールログの表示。コンソールログは、エンタープライズサーバのさまざまな構成要素からのメッセージで構成されます。表示できるメッセージの詳細は、

Enterprise Server の次のヘルプトピックを参照してください。『[Enterprise Server エラーメッセージ](#)』

- サーバの通信コンソールログの表示
- サーバの診断ダンプの作成
- サーバの診断ダンプの表示
- サーバへのサービスの追加

[方法](#)

5.4.2 CCI サーバ

Add/Delete またはそれ以上のアクセス権レベルをもつ場合は、CCI サーバを削除できます。CCI サーバが異常終了し、Directory Server から CCI サーバが登録解除されない場合のみに、CCI サーバを削除します。

[方法](#)

第 6 章 : サービスリスナー

ここでは、サービスリスナーオブジェクトと実行できる作業について説明します。

6.1 はじめに

サービスリスナーは、サービスのために着信クライアント要求を受信するネットワークアドレスを含むオブジェクトです。サービスには複数のリスナーを含むことができ、1つのリスナーでは複数のサービスのためにクライアント要求を受信できます。

管理者としての作業は、次のとおりです。

- 必要に応じて、リスナーを追加、更新、または削除する
- リスナーをサービスに関連付ける、またはその関連付けを解除する
- リスナーの開始と停止

6.2 リスナーの情報

ホームページ内のサーバのテーブルで「リスナー」にある [詳細] をクリックすると、リポジトリ内に格納されているリスナーの情報が表示できます。図 6-1 に、リスナーのテーブルを示します。

The screenshot shows the Enterprise Server Administration interface. At the top, it says "Enterprise Server Administration > ESDEMO > リスナー". Below that, there's a status bar showing "Status: MDS00001 OK" and a timestamp "Wed Jul 23 15:38:49 2003". There are navigation buttons: "画面更新", "<< サーバー", "サービス", "バンドラ", and "バックアップ". The main content is a table titled "登録済みリスナー ESDEMO".

名前	エンドポイント	現在のステータス	前回のステータス変更	ステータスログ	会話タイプ	カスタム構成データ	説明
HTTP Echo	tcp:10.18.11.122*:8002 (TECH-CPO-1.microfocus.co.jp)	停止	07/08/03-23:06:25	OK	http-echo		Simple HTTP test
Web	tcp:10.18.11.122*:2847* (TECH-CPO-1.microfocus.co.jp)	停止	07/08/03-23:06:25	OK	http-switch	[virtual paths] ogf=c:ASCE>bin uploads=c:ASCE>Neploy	Basic HTTP web server
Web Services	tcp:10.18.11.122*:8003 (TECH-CPO-1.microfocus.co.jp)	停止	07/08/03-23:06:25	OK	MFES		Web Services via HTTP/SOAP

図 6-1 : サービスリスナーのテーブル

表示される情報は、次のとおりです。

- 「名前」 - リスナー名。
- 「エンドポイント」 - リスナーが着信クライアント要求を受信するネットワークアドレスです。
- 「ステータス情報」 - 現在の状態と状態の変更が最後に発生した時刻が表示されます。
- 「ステータスログ」 - このリスナーで発生した最新のイベントについての情報を表示します。
- 「会話タイプ」 - サービスリスナーが SOAP または J2EE クライアント要求を処理する場合は「MFES」です。他の値は、このリスナーが処理する別の要求の種類を示します。
- 「カスタム構成データ」
- 「説明」

6.3 実行できる作業

Modify またはそれ以上のアクセス権をもつ場合は、次の作業ができます。

- リスナーの属性の編集
- リスナーをサービスに関連付ける
- リスナーのサービスとの関連付けを解除する
- エンタープライズサーバが実行中の場合には、リスナーの開始と停止

Add/Delete またはそれ以上のアクセス権をもつ場合は、次の作業もできます。

- リスナーの追加
- リスナーの削除 (サービスに関連付けられていない場合のみ)

サービスリスナーを初めて追加した際は、サービスリスナーのステータスを「閉鎖」、「無効」、または「停止」のどれかに設定できます。サービスリスナーステータスを「開始」に設定するには、「リスナーの編集」ページを使用します。リスナーステータスの詳細については、コンテキストヘルプを参照してください。

[方法](#)

第 7 章：サービス、パッケージ、および要求ハンドラ

ここでは、サービス、パッケージ、および要求ハンドラのそれぞれのオブジェクトについて説明し、これらのオブジェクトの相互関係と実行できる作業について説明します。

7.1 はじめに

サービスオブジェクト、実装パッケージオブジェクト、および要求ハンドラオブジェクトはともに動作することで、サービスベースのビジネス機能を提供することができます。システムサービスとは対照的に、ビジネスサービスのサービスオブジェクトには、必ず 1 つの要求ハンドラと関連付けられた 1 つの実装パッケージが必要です。サービスオブジェクトには 1 つ以上のサービスリスナーを関連付けることができます。

これらの 3 種類のオブジェクトに対して行う管理作業は、次のとおりです。

- 必要に応じて、オブジェクトを追加、更新、または削除する
- オブジェクトの状態を監視し、問題を調査する

7.2 サービスの手動作成

Interface Mapping Toolkit を使用して Net Express からサービスをデプロイする場合には、必要なすべてのオブジェクトは自動的に作成され、それらの関係が完成します。ただし、エンタープライズサーバにサービスを手動で追加した場合は、追加したサービスのオブジェクトとそれらの関係を、次の順序で作成する必要があります。

1. エンタープライズサーバに実装パッケージを追加する
2. 1 つ以上のサービスリスナーを追加する
3. サービスを追加し、リスナー、実装パッケージ、または要求ハンドラと関連付ける

7.3 追加、更新、および削除

サービスと実装パッケージをエンタープライズサーバに追加し、エンタープライズサーバが実行中かどうかに関係なく、追加したサービスと実装パッケージを更新または削除できます。

エンタープライズサーバが実行中の場合は（つまり、状態が「開始」の場合）、サービスとパッケージを追加するときに、その変更がすぐにエンタープライズサーバに反映されます。サービスがクライアント要求に回答していない場合は、更新が受け付けられてその変更がすぐに反映されません。

クライアント要求でリソースが使用されている場合は、サービス、パッケージ、リスナー、または要求ハンドラを削除できません。サービスが実行中かどうかに関係なく、パッケージがサー

ビスに関連付けられている場合はパッケージを削除できません。サービスにオペレーションがある場合は、サービスとそのすべてのオペレーションを 1 回でまとめて削除したり、オペレーションやパッケージを個別に削除したりできます。

エンタープライズサーバの実行中にリソースを追加、編集、または削除した場合は、オペレーションの成功や変更を示す管理メッセージがエンタープライズサーバコンソールに送信されません。

Enterprise Server が拒否する更新を Directory Server で受け付ける場合は、使用されているエンタープライズサーバで Directory Server リポジトリの同期が取れなくなることがあります。同期が取れなくなった場合には、サーバをいったん停止してから再起動して更新を同期させます。

7.4 サービス

サービスは、特定のビジネス機能にアクセスできます。

Interface Mapping Toolkit を使用して Net Express からデプロイしたサービスは、常に 1 つ以上のオペレーションを含みます。たとえば、Net Express の『入門書』で説明されているインターフェイスマッピングチュートリアルを実行する場合は、4 つのオペレーション (レコードの追加、レコードの読み取り、次のレコードの取得、レコードの削除) を含むサービス `wmapserv` を作成します。Enterprise Server のマニュアルでは、この種類のサービスに対して「オペレーションをもつサービス」という表現を使用します。

サービスの名前とサービスに関連付ける要求ハンドラ (ある場合) によって、エンタープライズサーバ内に作成するサービスには 1 つ以上のオペレーションが含まれます。サービスを作成しサービスに要求ハンドラを関連付けない場合、または指定した名前に有効な区切り文字が含まれない場合には、サービスとオペレーションは同じものとなります。Enterprise Server のマニュアルでは、この種類のサービスに対して「単純サービス」という表現を使用します。デフォルトエンタープライズサーバ `ESDEMO` の `Deploy` サービスなどのシステムサービスは単純サービスです。

サービス名には 2 つの形式があります。形式は、サービスにオペレーションが含まれるかどうかによって決定されます。次の 2 つの形式があります。

- `Test` または `my_service` など、1 つの部分で構成される名前。単純サービスの名前はこの形式です。
- `http://tempuri.org/wmapserv#Add` など、3 つの部分で構成される名前。名前を構成する 3 つの部分は次のとおりです。
 - サービスネームスペース。最初のシャープ (#) 文字より前の名前の部分です。
 - 区切り文字。ここではシャープ (#) 文字です。
 - オペレーションの名前。ここでは `Add` です。

オペレーションをもつサービスの名前はこの形式です。

Web サービスのサービス名では区切り文字として最初のシャープ (#) 文字を使用します。J2EE アプリケーションの一部を構成するサービスの名前では最後のピリオド (.) 文字を使用します。たとえば、mybinpservice.operation です。

サービス名には 1 つ以上のシャープ (#) 文字またはピリオド (.) 文字を含むことができます。Enterprise Server は最初の # 文字または最後のピリオド文字を区切り文字として認識します。

7.4.1 サービステーブル

ホームページ内のサーバのテーブルで「サービス」にある [詳細] をクリックすると、リポジトリ内に格納されているサービスの情報が表示できます。図 7-1 に、サービステーブルを示します。

Enterprise Server Administration > ESDEMO > サービス
Version 1.00.87
TECH-CPQ-1.microfocus.co.jp (10.18.11.122)

Status: MDS00001 OK Wed Jul 23 15:43:20 2003
[SCHEMA ADMINISTRATOR-Developer] [Page id: 3000]

画面更新 << サーバー リスナー ハンドラ パッケージ

1ページに サービスネームスペースを表示 << 前へ 次へ >>

編集... 1 - 2 / 2 サービスネームスペース ESDEMO											
サービス ネーム スペース	オペレーショ ン	サービス Class	探 索 順 序	リスナー	要 求 ハ ン ド ラ	実 装 パ ッ ケ ー ジ	前 回 の ス テ ー タ ス 変 更	現 ス テ ー タ ス	ス テ ー タ ス ロ グ	カスタム 構成 データ	説明
Test	Test <input type="button" value="編集..."/>		1	(HTTP Echo) tcp:10.18.11.122*:9002 (TECH-CPQ- 1.microfocus.co.jp)			-	Available	OK		Basic HTTP test service
Deployer	Deployer <input type="button" value="編集..."/>	MF deployment	1	(Web) tcp:10.18.11.122*:4077* (TECH-CPQ- 1.microfocus.co.jp)			-	Available	OK	[MF client] scheme=http URL=/cgi/mfdeploy.exe/uploads accept=application/x-zip-compressed	Deployment file-upload service

図 7-1 : サービステーブル

表示される情報は、次のとおりです。

- 「サービスネームスペース」 - サービスを識別する、エンタープライズサービス内で一意なラベルです。サービスにオペレーションが含まれる場合には、オペレーションをグループ化します。
- 「オペレーション」 - サービス内の個別のオペレーションのリストです。サービスが単純サービスの場合は、このリストには 1 つの項目のみがあり、サーバの名前と同じ名前となります。
- 「サービス Class」 - サービスの詳細な修飾子。クライアント要求は、名前付きサービスではなく、特定のクラスのサービスを検索できます。
- 「リスナー」 - サービスに割り当てられたリスナーのリストです。
- 「要求ハンドラ」 - サービスの要求ハンドラ。
- 「実装パッケージ」 - ビジネス機能を提供する実装パッケージです。
- 「ステータス情報」 - 現在の状態と状態の変更が最後に発生した時刻を表示できます。
- 「ステータスログ」 - このサービスで発生した最新のイベントについての情報を表示します。
- 「カスタム構成データ」
- 「説明」

7.4.2 実行できる作業

Modify またはそれ以上のアクセス権をもつ場合は、次の作業ができます。

- サービスの属性の編集
- サービスステータス (利用可能または無効) の変更

Add/Delete またはそれ以上のアクセス権をもつ場合は、次の作業もできます。

- サービスの追加
- サービスの削除。サービスを削除する場合は、サービスに関連付けられたオペレーションをすべて削除します。サービスに関連付けられたパッケージも削除できます。

[方法](#)

7.5 実装パッケージ

実装パッケージは、ビジネス機能を提供するアプリケーションを定義します。

7.5.1 パッケージテーブル

ホームページ内のサーバのテーブルで「パッケージ」にある [詳細] をクリックすると、リポジトリ内に格納されているパッケージの情報が表示できます。図 7-2 に、パッケージテーブルを示します。



Status	MDS00001 OK	Wed Jul 23 16:19:29 2003
--------	-------------	--------------------------

[SCHEMA ADMINISTRATOR-Developer] [Page id: k100]

画面更新 << サーバー リスナー サービス ハンドラ

1ページに 件を表示 << 前へ 次へ >>

編集... 1 - 2 / 2 パッケージ ESDEMO								
	名前	現ステータス	ステータスログ	パッケージモジュール	IDT	パッケージパス	カスタム構成データ	記述
編集...	BOOK.Add	Available	OK		e:\netx4\Base\deploy\6uoqhd\J\JMap Serv .idt	e:\netx4\Base\deploy\6uoqhd\J		from J\Map Serv .car
編集...	BOOK.Read	Available	OK		e:\netx4\Base\deploy\6uoqhd\J\JMap Serv .idt	e:\netx4\Base\deploy\6uoqhd\J		from J\Map Serv .car
追加...								

図 7-2 : パッケージテーブル

表示される情報は、次のとおりです。

- 「名前」 - 名前はエンタープライズサーバ内で一意です。
- 「現ステータス」 - 現在の状態。
- 「ステータスログ」 - このパッケージで発生した最新のイベントについての情報を表示します。
- 「パッケージモジュール」 - パッケージを含むライブラリの名前 (パッケージがライブラリの一部の場合)。
- 「IDT」 - このパッケージのインターフェイス定義テーブル (IDT) の名前。
- 「パッケージパス」 - パッケージモジュールの位置。
- 「カスタム構成データ」
- 「説明」

7.5.2 実行できる作業

Modify またはそれ以上のアクセス権をもつ場合は、次の作業ができます。

- パッケージの属性の編集
- パッケージステータス (利用可能または無効) の変更

Add/Delete またはそれ以上のアクセス権をもつ場合は、次の作業もできます。

- パッケージの追加
- パッケージの削除 (サービスに関連付けられていない場合のみ)

方法

7.6 要求ハンドラ

要求ハンドラは、クライアントからのサーバへのアクセス要求を受信し、要求をビジネス機能を提供するアプリケーションが解読できる形式に変換するソフトウェアです。Enterprise Server では、次の要求ハンドラが提供されています。

- MFRHSOAP - Simple Object Access Protocol (SOAP) を使用する要求を処理します。これらは Web サービス用の要求です。XML で記述されています。
- MFRHBINP - Micro Focus 製の J2EE コネクタが作成したバイナリ要求を処理します。

7.6.1 要求ハンドラテーブル

ホームページ内のサーバのテーブルで「要求ハンドラ」にある [詳細] をクリックすると、リポジトリ内に格納されている要求ハンドラの情報が表示できます。図 7-3 に、要求ハンドラテーブルを示します。

The screenshot shows the Enterprise Server Administration interface. At the top, it displays the title "Enterprise Server Administration > ESDEMO > ハンドラ" and the version "Version 1.00.87". Below this, the IP address "TECH-CPQ-1.microfocus.co.jp (10.18.11.122)" is shown. A status bar indicates "Status MDS00001 OK" and the date "Wed Jul 23 15:54:23 2003". There are navigation buttons for "画面更新", "<< サーバー", "リスナー", "サービス", and "パッケージ". The main content area contains a table titled "登録済みサービスの要求ハンドラ ESDEMO".

編集...	モジュール名	モジュールパス	現ステータス	ステータスログ	カスタム構成データ	記述
編集...	MFRHBINP		Enabled	OK		Handles J2EE requests
編集...	MFRHSOAP		Enabled	OK		Handle SOAP requests
追加...						

図 7-3 : 要求ハンドラテーブル

表示される情報は、次のとおりです。

- 「モジュール名」
- 「モジュールパス」

- 「現ステータス」 - 現在の状態。
- 「ステータスログ」 - この要求ハンドラで発生した最新のイベントについての情報を表示します。
- 「カスタム構成データ」
- 「記述」

7.6.2 実行できる作業

Modify またはそれ以上のアクセス権をもつ場合は、次の作業ができます。

- 要求ハンドラの属性の編集
- 要求ハンドラステータス (利用可能または無効) の変更

Add/Delete またはそれ以上のアクセス権をもつ場合は、次の作業もできます。

- 要求ハンドラの追加
- 要求ハンドラの削除。MFRHBINP は削除できません。

[方法](#)

第 8 章 : Directory Server の監視

ここでは、Directory Server のアクティビティを監視する機能について説明します。

8.1 監視機能

Application Server Administration は、Directory Server のアクティビティを監視するために、次の 3 つの機能を備えています。

- 「統計」ページ
- ジャーナル
- セッションリスト

8.1.1 「統計」ページ

「統計」ページでは、Directory Server およびその現在のアクティビティの統計と他の情報を表示します。次の 4 種類の情報が表示されます。

- リポジトリの位置と状態
- リポジトリ内の各オブジェクト (サーバ、リスナー、サービス、要求ハンドラ、およびパッケージ) の数、およびこれらのオブジェクトの総数。サービスの数は、実際のサービス処理の数を示します。
- 要求処理の統計
- 構成オプションの設定

問題が発生した場合には、このページに表示される情報が役立ちます。たとえば、サービスをデプロイしたにもかかわらず、デプロイしたサービスがサービステーブルに表示されない場合には、サービスとパッケージの数に変更されたかどうかをこのページで確認できます。

8.1.2 ジャーナル

ジャーナルは、Directory Server で行われたアクティビティを記録するファイルです。ジャーナルは使用をトレースできるため、サポート時に役立ちます。

次に示す 3 つのジャーナルのうち、どれか 1 つを選択できます。

- エラーのみ (デフォルト)
- エラーと警告
- すべての情報。このレベルを選択するとパフォーマンスが低下します。パフォーマンスの低下の度合いは、ファイルシステムのパフォーマンスやクライアントのトラフィックなど多数の要因によって決定されます。アクティブなプログラムのクライアントの数が多しシステムで最もパフォーマンスが低下します。

ジャーナルのレベルは「構成オプション」ページで設定できます。

8.1.3 セッションリスト

セッションリストは、Directory Server に接続されている Web ブラウザクライアントとプログラムクライアントのすべてのリストです。セッションはアクティブか、または休止状態でまだ削除されていない状態のどちらかです。これは、非アクティブになってから削除されるまでの時間がまだ経過していないためです。

ユーザのアクセスが制限されていない場合は、1 つの Web ブラウザクライアントセッション (セッション ID「Developer」) のみが表示されます。

ユーザアクセスが制限されている場合は、一意なユーザ ID で識別される各 Web ブラウザクライアントセッションユーザの個別のセッションが表示されます。

8.2 実行できる作業

Modify またはそれ以上のアクセス権をもつ場合は、次の作業ができます。

- 「統計」ページの表示
- ジャーナルの表示
- セッションリストの表示

Add/Delete またはそれ以上のアクセス権をもつ場合は、次の作業ができます。

- ジャーナル内のエントリをすべて削除する。削除後に、新しいジャーナルがすぐに開始されます。
- ジャーナルをテキストファイルとしてエクスポートする。このジャーナルのエクスポートは、ジャーナルの検索や注釈を付けるために行います。

[方法](#)

第 9 章：要求ハンドラのユーザ出口

ここでは、Micro Focus MFRHSOAP 要求ハンドラの動作のカスタマイズを可能にする、ユーザの出口プログラムの作成方法について説明します。

9.1 はじめに

Micro Focus MFRHSOAP 要求ハンドラにはユーザの出口点がいくつか用意されているため、入力パラメータと出力パラメータの追加処理を実行するユーザの出口プログラムを提供することができます。

必要なすべてのコードを要求ハンドラ自体にインクルードするよりもユーザの出口プログラムを提供した方が利点がある場合には、カスタム要求ハンドラとともに使用するユーザ出口プログラムを利用できます。

インターフェイスマッピング処理の一部として実行時に要求ハンドラが呼び出すユーザ出口プログラムとその出口点を指定する方法については、『分散コンピューティング』の『Interface Mapping Toolkit を使用した COBOL Web サービス』の章を参照してください。指定したユーザ出口プログラムの名前とその出口点は、マッピングされたサービスに関連付けられたインターフェイス定義テーブル (IDT) に組み込まれます。

9.2 提供されているユーザ出口点

図 9-1 に、要求ハンドラ、アプリケーションコンテナ、およびランタイムインターフェイスマッパーの相互関係を示します。

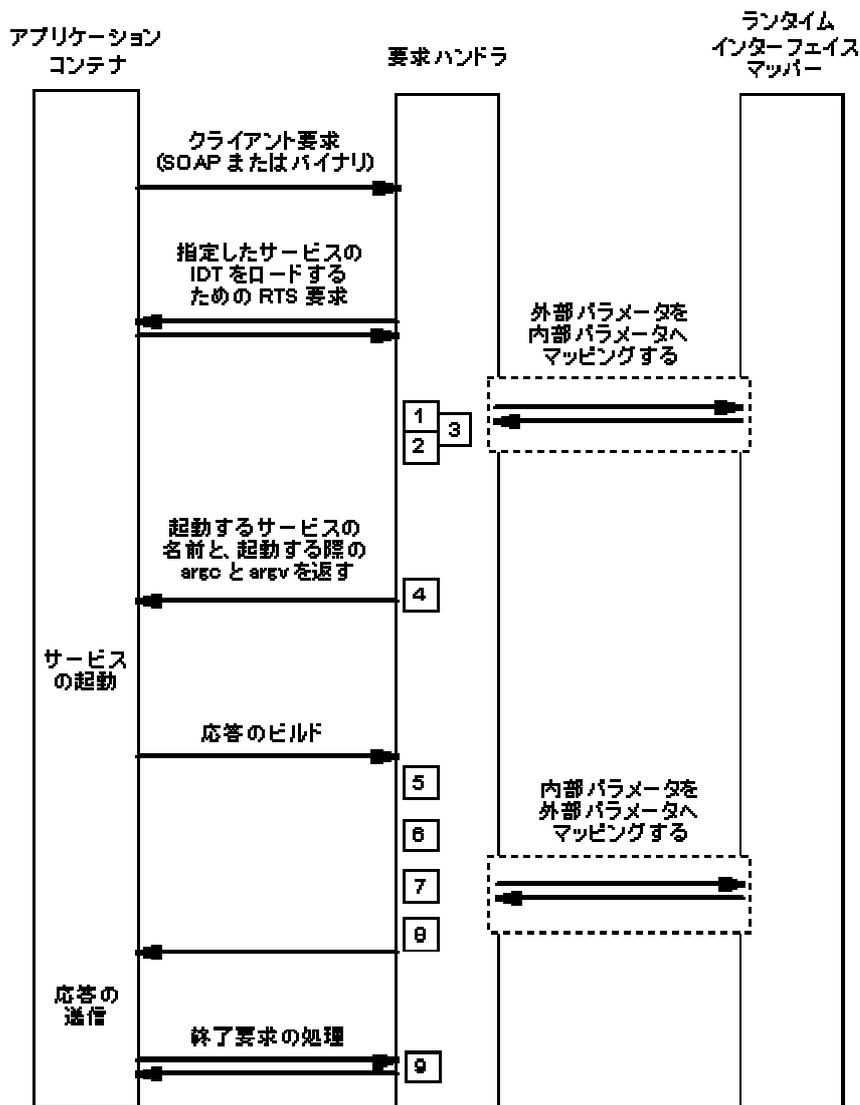


図 9-1 : 要求ハンドラの出口点

ユーザ出口点は、1 ~ 9 の番号がついた四角で表しています。次の表に、それぞれの出口点の名前と使用目的を示します。

番号	名前	説明
1	外部データ値変更 (入力パラメータ)	要求ハンドラのパラメータのデフォルトマッピングが行われる前に、クライアント要求で指定した外部データ値を変更できます。この出口点は、IDT で指定された外部-内部入力パラメータごとに呼び出されます。
2	内部データ値変更 (入力パラメータ)	要求ハンドラのパラメータのデフォルトマッピングで作成された内部データ値を無効にできます。この出口点は、IDT で指

- 定された外部-内部入力パラメータごとに呼び出されます。
- | | | |
|---|----------------------|-------------------------------------------------------------------------------------------------------|
| 3 | マップされない名前付き外部入力パラメータ | IDT 内で定義された入力パラメータマッピングに関連付けられていない入力パラメータを処理できます。この出口点は、マップされない名前付き入力パラメータごとに呼び出されません。 |
| 4 | 入力要求処理の終了 | クライアント要求で指定された外部パラメータ以外のパラメータ値を設定できます。また、出口点 1 ~ 3 に割り当てられたリソースを開放できます。 |
| 5 | 応答処理の開始 | クライアント応答をビルドするデフォルトマッピングを完全にバイパスできます。出口プログラムは、アプリケーションコンテナによってクライアントに返される応答を作成します。 |
| 6 | 内部データ値変更 (出力パラメータ) | 要求ハンドラのパラメータのデフォルトマッピングが行われる前に、ユーザアプリケーションが作成した内部データ値を変更できます。この出口点は、IDT で指定された内部-外部出力パラメータごとに呼び出されます。 |
| 7 | 外部データ値変更 (出力パラメータ) | 要求ハンドラのパラメータのデフォルトマッピングで作成された外部データ値を無効にできます。この出口点は、IDT で指定された内部-外部出力パラメータごとに呼び出されます。 |
| 8 | 応答処理の終了 | デフォルトマッピングで生成された応答を変更できます。 |
| 9 | 要求処理の終了 | 出口点 1 ~ 8 に割り当てられたリソースを開放できます。 |

9.3 構造体定義と型定義

COBOL プログラムと C プログラムのために、Enterprise Server には COBOL コピーブック (cbltypes.cpy) と C ヘッダーファイル (svext.h) が用意されています。これらのファイルでは、カスタム要求ハンドラの作成に必要な構造体とインターフェイスが定義されています。

各出口点で処理を行う出口ルーチンで必要になる情報を渡すために使用する IDP_EXIT_INFO 構造体が定義されています。この構造体へのポインタは、出口点識別子とともにそれぞれの出口ルーチンに渡されます。渡される情報と返される情報は、出口点によって異なります。

各出口点で渡される情報の定義については、次のヘルプトピックを参照してください。『[出口点](#)』

サポート関数の定義については、次のヘルプトピックを参照してください。『[関数](#)』

構造体の定義については、次のヘルプトピックを参照してください。『[構造体](#)』

9.4 連絡節のフィールド名

いくつかの出口点が提供されているため、入力フィールドまたは出力フィールドを操作できます。出口インターフェイスにフィールド名を渡す方法は、サービスを提供するプログラム内でフィールドを渡す方法が連絡節でどのように定義されているかによって異なります。次の2つの方法があります。

- 項目はレベル-01 で定義されます。この場合は、出口インターフェイスに渡されるフィールド名は、連絡節のフィールド名と同じです。
- 項目はレベル-01 の下位レベルで定義されます。この場合は、出口インターフェイスに渡されるフィールド名には、そのフィールドが属する集団項目の名前がプリフィックスとして付きます。それぞれの名前は、ピリオド文字 (.) で区切ります。たとえば、次のレコードのフィールド「e」は「a.b.c.e」として一意に識別されます。
- 01 a.
- 03 b.
- 05 c.
- 07 d pic x(4).
- 07 e pic x(4).
- 07 f pic x(4).
- 05 g pic x(4).
- 03 h pic x(4).

フィールド自体またはフィールドが属する集団項目が表の場合は、必要とされる表指標は名前に「(n)」を追加することで指定できます。n は表内のフィールドまたは項目の位置を表します。たとえば、次のレコードで「c」の4番目のフィールド「f」は、「a.b.c(4).f」として一意に識別されます。

```
01 a.  
03 b.  
05 c occurs 10.  
07 d pic x(4).  
07 e pic x(4).  
07 f pic x(4).  
05 g pic x(4).  
03 h pic x(4).
```

9.5 ユーザ出口プログラムのディプロイ

ユーザ出口プログラムをディプロイするには、ユーザ出口プログラムを Enterprise Server の bin ディレクトリにコピーします。Net Express の一部として Enterprise Server をインストールした場合は install-dir¥base¥bin、Enterprise Server をスタンドアローンでインストールした場合は install-dir¥bin です。