

Micro Focus® Enterprise Server

構成と管理

第 6 版
2006 年 1 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。

はじめに





第 1 章 : はじめに

ここでは、Enterprise Server について説明します。

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」は実行環境を、「Enterprise Server」は製品名を意味します。

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。COBOL 開発システムの一部である Interface Mapping Toolkit を使用すると、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、または J2EE コネクタや COM を介してクライアントから起動できます。

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。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、『Interface Mapping Toolkit』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。

ライセンス

エンタープライズサーバを実行するには、ライセンスが必要です。詳細は、『[実行ライセンス管理ガイド](#)』を参照してください。

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。

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。

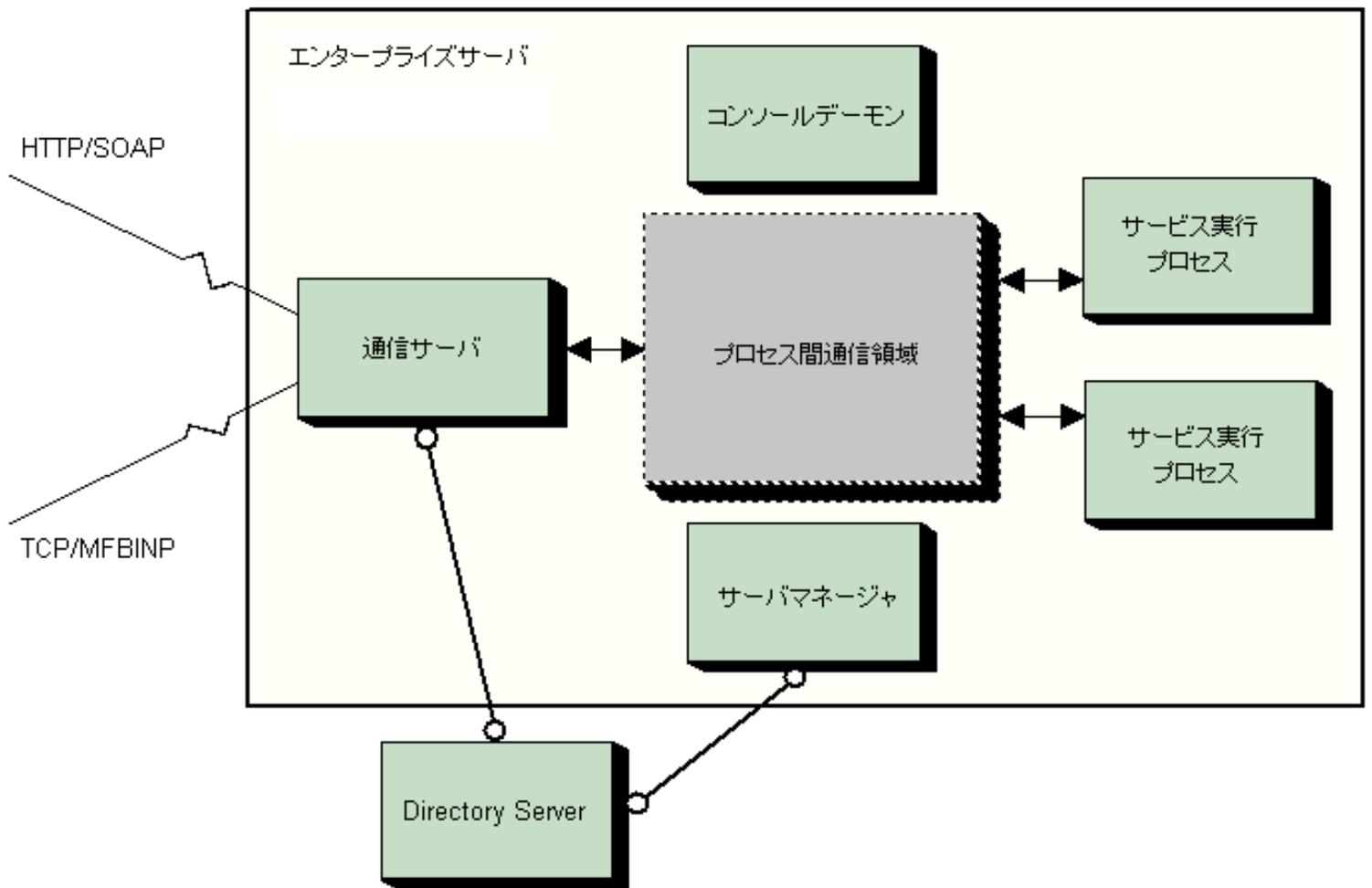


図 1-1: エンタープライズサーバのアーキテクチャ

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。

プロセスにはコンソールデーモン、サーバマネージャ、通信プロセスがあります。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。
- 通信プロセスは、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、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は 1 つの通信プロセスとともに作成されますが、信頼性と耐障害性を高めるために通信プロセスを追加することもできます。

エンタープライズサーバは、2 つのサービス実行プロセスとともに作成されます。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行うプロセスです。サービス実行プロセスは追加することができます。

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領域があります。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。

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もつことができます。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が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プロセスで受信され、共有メモリ領域内のキューに置かれることでその実行がスケジューリングされます。
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。

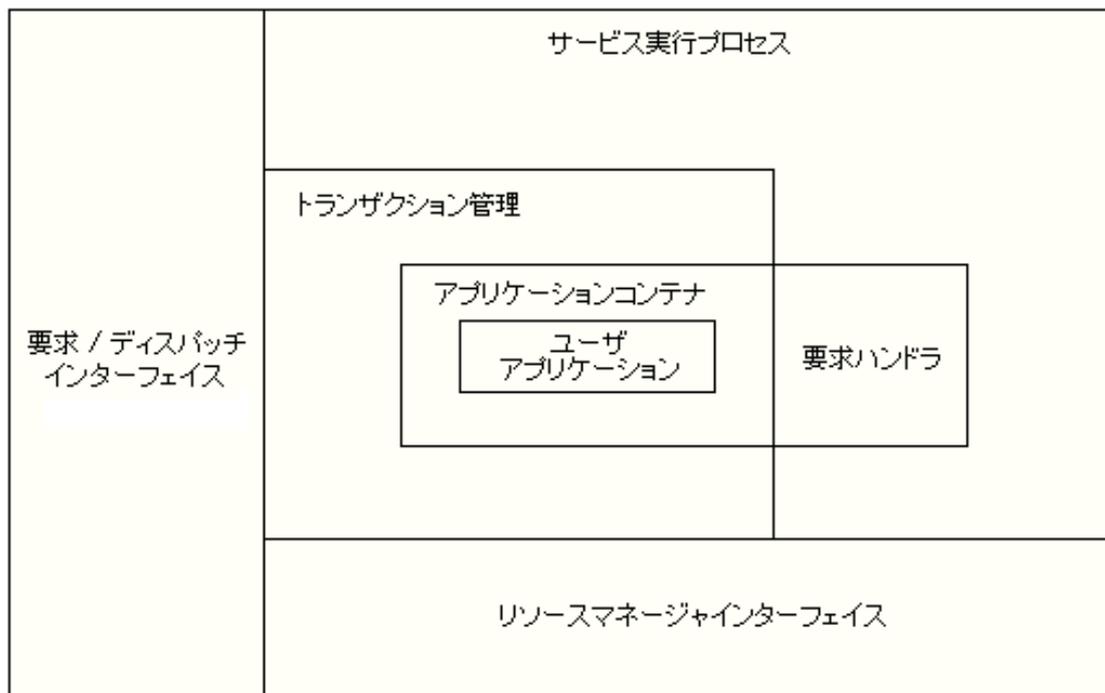


図 1-2: サービス実行プロセスの構成要素

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』](#)

[一ス管理](#)』を参照してください。

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
- 通信プロセス
- サービスリスナー
- 要求ハンドラ
- 実装パッケージ
- XA リソース

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、サーバから独立して存在することはできません。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 COBOL 開発システムからデプロイするのではなく、手動で設定する場合は、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。

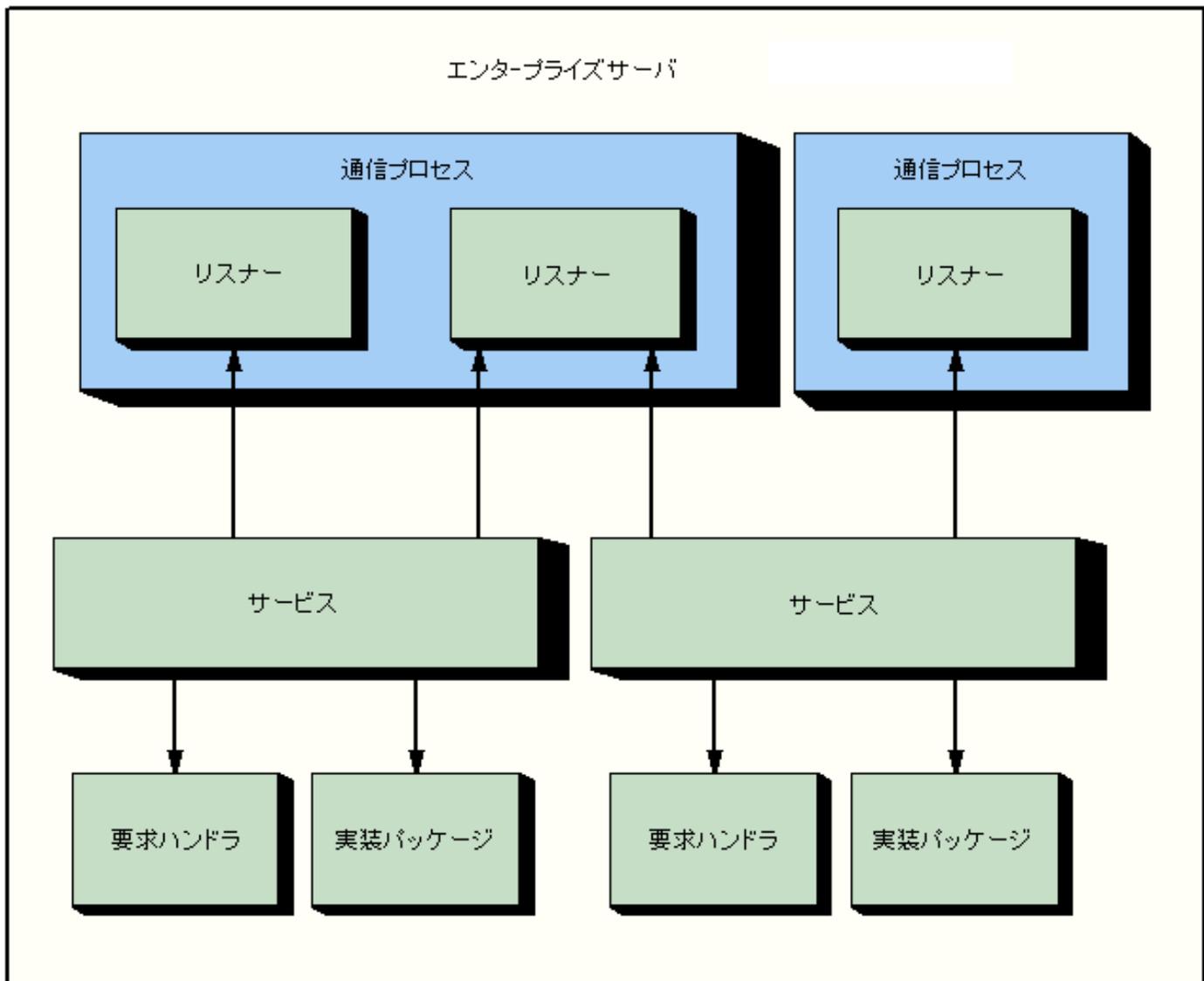


図 1-3: エンタープライズサーバ内のオブジェクト

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。

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。

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 応答があります。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。

通信プロセス

通信プロセスオブジェクトは、通信プロセスを定義します。これについては、『[エンタープライズサーバのアーキテクチャ](#)』の節で説明しています。通信プロセスにはサービスリスナーがあります。新しいエンタープライズサーバを追加すると、そのサーバには、2 つのシステムサービスのリスナーを含む 1 つの通信プロセスが作成されます。そのエンタープライズサーバで予測されている作業負荷に対応するため、必要に応じてリスナーを追加できます。プロセスを希望どおりに定義した後は、既存の通信プロセスをコピーして通信プロセスを追加することができます。複数の通信プロセスをもつことの利点は次のとおりです。

通信障害が起きてもサーバが堅牢であること。1 つの通信プロセスに障害が発生しても、他のプロセスは継

続いてクライアントの要求を受信して処理し、応答を返すことができます。

サーバが処理できるクライアント要求の数が増えること。通信プロセスはスレッドアーキテクチャを使用しているため、通信プロセスの数が増えると利用可能なスレッドの数も増えることになります。

通信プロセスの詳細は、『[通信プロセスとサービスリスナー](#)』の章を参照してください。

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 サービスの 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、『[通信プロセスとサービスリスナー](#)』の章、および『構成』の章にある『[デプロイサービスとデプロイリスナー](#)』を参照してください。

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、通信プロセス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。SOAP プロトコル用の要求ハンドラは、ユーザ出口を使用してカスタマイズすることができます。

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。

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。パッケージは、IDT (interface definition table; インターフェイス定義テーブル) に関連付けられています。IDT には、クライアント要求を COBOL プログラムインターフェイスにマップするため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。

XA リソース

XA リソースオブジェクトには、XA 互換のデータベースマネージャに関する情報が含まれています。1つまたは複数のコンテナ管理サービスに対してデータベースアクセスを行うには、このデータベースマネージャが必要です。

XA リソースの詳細は、この章の『[リソース管理](#)』の節と、『構成』の章の『[リソースマネージャ](#)』の節を参照してください。

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。詳細は、『構成』の章の『[Fileshare](#)』の節を参照してください。

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。

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 COBOL 開発システムへのディ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。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。

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) に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 の環境でサポートされていないリソースマネージャを使用する場合 (readme の『[補足情報](#)』の節を参照)。

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) でファイルに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の実行方法は、Net Express、Server Express、Micro Focus Server といった従来の実行環境でのアプリケーションの実行方法と類似していますが、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。

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

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。

リソースマネージャとして動作するエンタープライズサーバ

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す。

サービスインターフェイスのディプロイ

Interface Mapping Toolkit を使用して COBOL プログラムをマップした後、COBOL プログラムをエンタープライズサーバにディプロイする必要があります。ディプロイ処理の際、開発者は、Interface Mapping Toolkit の一部であるディプロイツールを用いてインターフェイスをディプロイします。このツールはエンタープライズサーバにインターフェイスを自動的にディプロイするだけでなく、COBOL アーカイブファイル (.car ファイル) にサービスをディプロイするために必要なファイルをすべてパッケージ化します。通常、運用環境でサービスを実行する準備がいったん整うと、ディプロイはユーザ側の責任になります。ディプロイツールが最後に使用されたときに作成された .car ファイルから、インターフェイスをディプロイします。

サービスが EJB インターフェイスとしてマップされている場合は、以下の作業も実行します。

- EJB を J2EE アプリケーションサーバにディプロイする。
- リソースアダプタをディプロイする。

.car ファイルからのディプロイと、EJB およびリソースアダプタのディプロイの詳細は、『[インターフェイスのディプロイ](#)』の章を参照してください。Interface Mapping Toolkit を使用したインターフェイスの自動ディプロイの詳細は、『Interface Mapping Toolkit』の『[Interface Mapping Toolkit](#)』の章にある『ディプロイツール』の節を参照してください。

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。

Enterprise Server をいったん構成すると、Web インターフェイスのかわりに Enterprise Server のコマンドを使用して管理作業ができるようになります。コマンド行インターフェイスの詳細は、『[サーバ](#)』の章を参照してください。

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バにサービス、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 へ直接ディプロイする場合は、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
- サービスの再ディプロイ

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、サーバ内のサービス実行プロセスの数を増やします。

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。

マップしたインターフェイスのエンタープライズサーバへのディプロイと、EJB としてマップしたサービスをディプロイするために必要な他の作業についての詳細は、『[インターフェイスのディプロイ](#)』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。ESMAC (起動されたエンタープライズサーバでのみ使用できる管理機能のセット) については、『[ESMAC 使用によるサーバの管理](#)』の章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](#)』、『[通信プロセスとサービスリスナー](#)』、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。

トラブルシューティングの問題については、『[トラブルシューティング](#)』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 とご使用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。

はじめに

Enterprise Server を初めて起動する場合は、次の作業が必要です。

1. ユーザアカウントストラテジを決定して実装する
2. Directory Server のセキュリティを設定する
3. Directory Server の他の構成オプションを設定する
4. 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け取り、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バに追加します。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。

ユーザアカウントストラテジ

COBOL プログラムがエンタープライズサーバ内のサービスとして実行される場合、COBOL プログラムは、サービス実行プロセスを開始したサーバマネージャプロセスからセキュリティ認証情報を継承します。サーバマネージャは、順に casstart プログラムにより開始されます。インタラクティブなユーザがコマンド行から casstart を実行する場合、COBOL サービスプログラムは、そのユーザのセキュリティ認証情報を使用します。

ただし、より一般的には、Enterprise Server は Directory Server への Web インターフェイスである Micro Focus Enterprise Server Administration を使用して開始します。Enterprise Server Administration (ES Admin) は、通常は Windows システムサービスとして実行されます (コントロールパネルの [サービス]、または net start コマンドの出力では Micro Focus Directory Server としてリストされます)。Windows システムサービスは、スタートアップのオプションで指定したユーザアカウントで実行されます。スタートアップのオプションは、コントロールパネルの [サービス] を使用して表示および変更ができます。

Net Express または Enterprise Server for Windows をインストールすると、MFDS はシステ

ムサービスとして、ローカルシステム ユーザアカウントを使用し、「デスクトップとの対話をサービスに許可」オプションを選択してインストールされます。これには 2 つの利点があります。

Windows の起動時に MFDS が開始される。MFDS を起動するためにシステムにログインする必要はありません。Enterprise Server にログインして手動で開始するのではなく、Windows の起動時に自動的に開始する場合には、この設定が必要です。実行するユーザアカウントに関係なく、どのシステムサービスでも Windows の起動時に開始するように設定できる点に注意してください。

「デスクトップとの対話をサービスに許可」オプションが使用できる。その他のユーザアカウントには、このオプションはありません。Enterprise Server の実行中は、Enterprise Server コンソールデーモンのウィンドウが表示されます。Enterprise Server がサービスとして実行され、「デスクトップとの対話をサービスに許可」オプションが選択されていない場合 (MFDS がローカルシステム以外の任意のユーザアカウントを使用している場合)、コンソールデーモンのウィンドウは表示されません。この場合でも、コンソールデーモンのメッセージは ES Admin によって表示できることに注意してください。

ただし、ローカルシステム アカウントには、ネットワークファイルアクセスの特権はありません。つまり、MFDS で開始したエンタープライズサーバ内で実行され、デフォルトの構成を使用する COBOL サービスプログラムは、ネットワークファイルをオープンできません。COBOL サービスプログラムからのネットワークファイルへのアクセスを可能にするには、次のいずれかの操作を行います。

Windows にログインし、casstart コマンドを使用してコマンド行からエンタープライズサーバを開始する。COBOL サービスプログラムはログインユーザのネットワーク認証情報を使用し、ログインユーザがアクセス権限をもつファイルをオープンできます。

この方法の不便な点は、手動で行わなければならないということです。つまり、エンタープライズサーバを ES Admin ではなく手動で (casstop コマンドを使って) 停止しなければなりません。Windows の起動時に Directory Server を自動的に開始することはできません。ユーザがログオフすると、エンタープライズサーバはシャットダウンされます。

コントロールパネルの [サービス] で MFDS を無効にし、適切なネットワークファイルアクセスをもつユーザでログインしてから、mfds コマンドを使用してコマンド行から開始する。ES Admin からエンタープライズサーバを開始すると、MFDS のユーザ認証情報が継承されます。

この方法の不便な点は、上記と同じような点ですが、この場合は ES Admin を使用して Enterprise Server の起動と停止ができます。

MFDS サービスのユーザアカウントを変更する。適切なネットワークファイルアクセスをもつユーザアカウントで MFDS サービスを実行すると、COBOL サービスプログ

ラムはこれらのファイルをオープンできます。確実にネットワークファイルをオープンするために、ドライブのマッピングではなく UNC パス (つまり、「¥¥server ¥share¥path-to-file¥filename」という形式のファイルパス名) がプログラムに対して必要になる点に注意してください。

この方法の不便な点は、主に、Enterprise Server がコンソールデーモンのウィンドウを表示できないということです。ただし、Enterprise Server のセキュリティは向上します。これは、ローカルデスクトップへのアクセスにセキュリティ上のリスクがあるためです。ローカルシステム アカウント (ローカルセキュリティ目的で管理者に相当するアカウント) ではなく、通常のユーザアカウント (特権が少ないアカウント) で COBOL アプリケーションを実行すると、さらに安全です。

Windows 上の Enterprise Server では (COBOL サービスプログラムにネットワークファイルアクセスが必要であってもなくても)、MFDS とその下で実行される COBOL サービスプログラム用に特定のユーザアカウントを作成することをお奨めします。このアカウントに適切なアクセス権を設定し、COBOL プログラムに必要なアクセス権はいっさい与えないようにしてください。さらにセキュリティを高めるには、このユーザに対して特定のオブジェクト (ディレクトリ、ファイル、レジトリキー) へのアクセスを許可または拒否できるように ACL を設定して、COBOL サービスプログラムが実行できる作業を制御します。

casstart、casstop および mfdss コマンドの詳細は、次のヘルプトピックを参照してください。『[コマンド](#)』

ファイアウォールの設定

Directory Server およびエンタープライズサーバを実行しているマシンでファイアウォールが有効で、リモートクライアントが Directory Server およびエンタープライズサーバへ接続する場合は、使用しているポートに対してファイアウォールがアクセスを許可していることを確認する必要があります。

たとえば、Directory Server は、デフォルトでポート 86 を使用する設定になっています。このポートへの TCP および UDP アクセスを許可するよう、ファイアウォールを設定する必要があります。同じように、デフォルトのエンタープライズサーバである ESDEMO には、ポート 9003 を使用する Web サービスリスナーがあります。リモートクライアントがこのリスナーへ要求を送信できるようにするには、ファイアウォールがこのポートへのアクセスを許可しなければなりません。

リモートユーザがファイアウォールを介して Enterprise Server の機能にアクセスできるようにする場合には、固定のポート番号を使用して、アクセスを制御できるようにすることをお奨めします。

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、Directory Server と、それが管理するすべてのエンタープライズサーバに対して、誰でも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、ユーザ グループ、およびアクセス権を使用することで提供されます。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。Directory Server には、セキュリティを有効にしてから使用するための設定済みユーザアカウントがいくつかあります。

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 は Directory Server Administrators グループのメンバーなので、Directory Server で変更を行うために必要なアクセス権があります。このユーザ ID の不正利用を防ぐための処置が必要です。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。新しいユーザ ID を作成したり、事前設定済みのユーザ ID を削除したりすることもできます。

方法

セキュリティとアクセス権の詳細は、『[ユーザ、グループおよびアクセス権](#)』の章を参照してください。

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](#)

エンタープライズサーバ

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](#)

エンタープライズサーバとそれに含まれるオブジェクト (サービス、リスナーなど) にはそれぞれ特有の「構成情報」フィールドがあります。これらのフィールドは、各オブジェクトタイプの「追加」および「編集」ページでアクセスできます。「構成情報」フィールドのエントリでは .ini ファイル形式を使用し、セクション名が角かっこで囲まれ、その後に関数名/値の組が続きます。

```
[ASection]
name1=value1
name2=value2
```

ほとんどの設定では、大文字と小文字を区別しません。例外を次に示します。

「Web」リスナーの仮想パス名は、大文字と小文字を区別します。

UNIX では、環境変数の設定 ([ES-Environment] での) と、ファイルまたはディレクトリ名を含む値では大文字と小文字を区別します。

このフィールドの情報については、その種類のオブジェクトに関連する章を参照してください。

[第 7 章: サーバ](#)

[第 8 章: 通信プロセスとサービスリスナー](#)

[第 9 章: サービス、パッケージ、および要求ハンドラ](#)

以降では、エンタープライズサーバの属性について詳細に説明します。

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](#)』の節を参照)。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)。共有メモリ領域のサイズは、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できなくなることがあります。

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	補助トレースがアクティブでない場合 エントリ数 × 24	
	補助トレースがアクティブな場合 エントリ数 × 24 × 11	
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 × 144	
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)	
クライアント	ピーク時の同時クライアントの数 × 64	
合計		

変数の内容は、次のとおりです。

service-name-length サービスの名前の長さ

handler-name-length	要求ハンドラの名前の長さ
IDT-name-length	IDT の名前の長さ
application-path-length	パッケージ内の COBOL アプリケーションへのパスの長さ
module-name-length	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
IDT-length	パッケージの IDT ファイルのサイズ

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には、短期間に集中するサーバの要求を処理する機能があります。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。

サービス実行プロセスの数

エンタープライズサーバの作成時には、サービス実行プロセスが 2 つあります (『はじめに』の章の『[サービス実行プロセス](#)』の節にある図『サービス実行プロセスの構成要素』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、サービス実行プロセスのどれか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。詳細については、『[パフォーマンスに関する考慮事項](#)』の節を参照してください。

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。

[方法](#)

通信プロセスの数

通信プロセスは、クライアントとエンタープライズサーバの間の通信を処理し、多数のサービスリスナーで構成されています。エンタープライズサーバの作成時には、通信プロセスが1つあります。何らかの理由で通信に障害が発生した場合、エンタープライズサーバは、通信プロセスが再起動されるまで作業が何も処理できなくなります。エンタープライズサーバの通信への耐障害性を高めるために、エンタープライズサーバそれぞれに対して1つまたは複数の追加通信プロセスを作成できます。追加の通信プロセスは、既存のプロセスをコピーして作成します。新しい通信プロセスは、元の通信プロセスの完全なクローンですが、固定のポート番号は複製されません。新しく作成されたエンタープライズサーバの中で最初の通信プロセス、Communications Process 1 は、Web サービスのリスナーと、ディプロイリスナーを持ちます。新しいエンタープライズサーバを作成する場合、この通信プロセス定義を変更してからコピーすることもできます。この定義の変更は、サーバに対して計画した特定の作業負荷に応じて行います。

方法

障害に対応するためには、少なくとも2つの通信プロセスが必要です。オペレーティングシステムのスレッド制限に対処するために、3つ以上必要になることもあります。最大数は、32です。

ディプロイリスナーの詳細は、『[ディプロイサービスとディプロイリスナー](#)』の節を参照してください。通信プロセスとリスナーの管理についての詳細は、『[通信プロセスとサービスリスナー](#)』の章を参照してください。

環境変数

エンタープライズサーバは、起動時に Directory Server から環境を継承します。つまり、サードパーティソフトウェアのディレクトリ設定など、サーバの作業負荷に必要な環境変数は、Directory Server を起動したセッションで設定する必要があります。

「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。

環境変数の形式は、次のとおりです。

```
[ES-Environment]
environment-variable-name=environment-variable-setting
```

文字列内の要素は、次のようにセミコロンで区切ります。

```
[ES-Environment]
COBPATH=c:¥adirectory;c:¥anotherdirectory
```

注：(特に UNIX の場合には) エンタープライズサーバでは、COBDIR 環境変数を設定しないことをお奨めします。これは、製品の位置を示すために \$COBDIR を使用しているためです。\$COBDIR を設定した場合の結果は未定です。エンタープライズサーバのレベルでサービスプログラムの位置を指定する場合には、COBPATH 環境変数を使用します。エンタープライズサーバのレベルで COBPATH 環境変数を設定する場合、「パッケージの追加」または「パッケージの編集」ページの「パッケージパス」で、「\$COBPATH」も設定する必要があります。

環境変数を指定する場合、すでに作成されている他の変数の解決済み値を、環境変数の値の一部として使用することができます。この場合は、次のように、環境変数の先頭にドル記号 (\$) を付加します。

```
FILEROOT=d:¥data  
FILEA=$FILEROOT¥mydata.dat
```

この値は、d:¥data¥mydata.dat となります。

他の環境変数の参照を示す記号としてではなく、環境変数の実際の値の一部としてドル記号を使用したい場合は、次のように円記号 (¥) 文字を挿入してドル記号をエスケープします。

```
FILEA=¥$¥$fsserver1¥mydata.dat
```

この値は、\$\$fsserver1¥mydata.dat となります。

[方法](#)

Fileshare

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にアクセスする場合は、FHREDIR 環境変数を設定して、FHREDIR 構成ファイルの位置を指定する必要があります。たとえば、次のように指定します。

```
[ES-Environment]  
FHREDIR=home/mydir/client.cfg
```

FHREDIR の構成は静的なので、FHREDIR 構成ファイルの内容はエンタープライズサーバを初期化する時に読み込まれます。動的に変更することはできません。つまり、FHREDIR 構成ファイルには、ディプロイされているサービス、またはエンタープライズサーバにディプロイされる可能性があるサービスすべてのエントリを指定する必要があります。サービスがエンタープライズサーバに動的にディプロイされ、Fileshare に対して異なる構成を必要とする場合、エンタープライズサーバを停止して再起動する以外の方法はありません。

エンタープライズサーバがすべて同じ構成ファイルを使用する場合、各サーバにそれぞれ個別の構成エントリを使用せず、\$FHREDIR をコマンド行で設定してから Directory Server を起動することもできます。

エンタープライズサーバを起動する前に、エンタープライズサーバのサービスからアクセスする可能性のある Fileshare サーバは、すべて起動しておいてください。エンタープライズサーバは、Fileshare サーバを停止する前に停止してください。

Fileshare の詳細は、『[Fileshare ガイド](#)』を参照してください。

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。この構造体を、xa_switch_t と呼びます。XA 互換のリソースマネージャにアクセスするには、スイッチモジュールが必要です。このモジュールは、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のに必要です。

Net Express では、Oracle、IBM DB2、および SQL サーバ用のスイッチモジュールと、汎用 1 フェーズのスイッチモジュールのソースが、Net Express のインストールディレクトリ 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。

esoraxa.cbl	Oracle
esdb2bxa.cbl	IBM DB2
esmssql.cbl	SQL Server
esodbcxa.cbl	ODBC 用の汎用 1 フェーズコミット

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
ORA10	Oracle 10
DB2	IBM DB2

MSSQL	SQL Server
ODBC	ODBC の汎用 1 フェーズコミット

DB2 用のスイッチモジュールをビルドする場合は、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。

SQL Server と、ODBC スwitchモジュールの汎用 1 フェーズコミットを使用するには、Enterprise Server でそれらを定義する必要があります。詳細は、ヘルプトピック『[SQL サーバ XA スwitchモジュール](#)』および『[ODBC 用汎用 1 フェーズコミット XA スwitchモジュール](#)』を参照してください。

Enterprise Server を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。

他の XA 互換のデータベースマネージャのサポートを追加できます。サポートを追加するには、データベースのスイッチモジュールを作成してコンパイルします。方法については、データベースベンダのマニュアルを参照してください。

Web インターフェイスの「サーバの編集」ページ > 「プロパティ」ページ > 「XA リソース」ページを使って XA リソースマネージャの詳細を指定します。また、XA リソースマネージャ定義を編集および削除することもできます。リソースマネージャを個別に有効にして、エンタープライズサーバの起動時にどのリソースをアクティブにするかを制御できます。

[方法](#)

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。

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。

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、CPU バウンドサービスの場合は、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ヘッドが小さくなるためです。

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。

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。

ディプロイサービスとディプロイリスナー

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ィプロイする Interface Mapping Toolkit の機能は、ディ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ィプロイサービスがあります。このディプロイサービスを変更することも、異なる構成でディ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ィ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。ディ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](#)

ディプロイサービス

すべてのディプロイサービスでは、Service Class 属性を「MF deployment」に設定する必要があります。ディプロイサービスは、対話タイプ「Web」を使用するリスナーに関連付ける必要があります。

ディ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/mfdeploy.exe/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 および accept パラメータの値は変更できませんが、新しくディプロイサービスを作成する際には省略することができます (これらの値はデフォルトです)。

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=<ES>/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。

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=<ES>/bin
uploads=<ES>/deploy
<default>=/dev/null
```

仮想パス「cgi」は、ディプロイされる 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「<ES>」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、「<ES>/deploy」はデフォルトのディ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。

ディプロイを実行するクライアントが指定する 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、リスト内のエントリは 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、クライアントが指定する URL 内の最初のディレクトリが、リスト内のどのエントリとも一致しない場合に使用されます。デフォルトのディレクトリは、存在しないディレクトリにする必要があります。そうすると、通信プロセスが URL を完全パスに変換すると要求は失敗します。これにより、クライアントがマシン上のディレクトリを任意に閲覧することができなくなります。通信プロセスはいずれにしても安全なデフォルトを使用するので、デフォルトのディレクトリを指定する必要はありません。

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。

Copyright © 2006 Micro Focus (IP) Ltd. All rights reserved.

◀ はじめに

インターフェイスのディプロイ ▶

第 3 章：インターフェイスのディプロイ

ここでは、運用環境で使用する .car ファイルからサービスインターフェイスをディプロイする方法と、EJB と EJB サービスインターフェイスのリソースアダプタをディプロイする方法を説明します。

サービスインターフェイス

Interface Mapping Toolkit を使用して生成したサービスインターフェイスを実行するには、エンタープライズサーバにディプロイする必要があります。サービスインターフェイスはマッピング情報、COBOL プログラム、および必須データからなります。

通常、開発者が開発の際に使用するインターフェイスは、Interface Mapping Toolkit のディプロイツールを使用して自動的にディプロイします。運用環境でインターフェイスを使用する準備が整った場合は、ディプロイツールが最後に実行されたときに生成された .car ファイルを使用してディプロイする必要があります。

.car ファイルをディプロイするには、2 つの方法があります。

mfdepinst コマンドを使用
手動処理

mfdepinst コマンドによるディプロイ

mfdepinst コマンドは、エンタープライズサーバへのサービスのディプロイ処理の一部をサーバ側で処理します。このツールは、.car ファイルに含まれているサービスをエンタープライズサーバにインストールします。これは、『[.car ファイルの手動ディプロイ](#)』の節に説明されている、.car ファイルを手動で解凍して、Enterprise Server Administration を使用してファイルを追加することと同じです。

mfdepinst コマンドは、.car ファイルからファイル (.int や .idt など) を抽出して、新しいサービスやパッケージ用に Directory Service ディレクトリを更新します。また、エンタープライズサーバが動作中の場合には、それらの新しいサービスを通知して使用できるようにします。

詳細は、ヘルプトピックの『[mfdepinst](#)』を参照してください。

.car ファイルの手動ディプロイ

サービス用の COBOL アーカイブファイル .car 内にあるファイルを解凍し、Enterprise

Server Administration を使用してエンタープライズサーバにサービスを追加することによって、手動でデプロイすることができます。 .car ファイルは、デプロイツールの使用時に作成されます。

.car ファイルは、デプロイフォルダ myservice¥Repos¥ myservice.deploy に格納されています。 COBOL アーカイブの重要なファイルを、次に示します。

インターフェイス定義テーブル (interface definition table; IDT)。たとえば、myservice.idt。

実装パッケージ。たとえば、myprogram.int および myprogram.idy。

必須データ。たとえば、mydata.idx および mydata.dat。

マニフェストファイル manifest.mf。デプロイメントプロセスを制御します。

サービスインターフェイスを手動でデプロイする場合、次の作業を行うために Enterprise Server Administration を使用します。

1. 実装パッケージを追加する。 IDT、実行形式ファイル、およびデータファイルのパスと名前を指定します。
2. サービスを追加する。 このサービスで使用するリスナーと、サービスの機能を提供する実装パッケージを指定します。

方法

EJB およびリソースアダプタ

COBOL サービスを実行する EJB は、エンタープライズサーバで動作する COBOL サービスおよび WebSphere や WebLogic などのサードパーティの J2EE アプリケーションサーバで動作する EJB とクライアントで実行されます。

.car ファイルから (または、開発者であれば Interface Mapping Toolkit のデプロイツールを使用して) COBOL サービスをデプロイします。 『[サービスインターフェイス](#)』の節を参照してください。

EJB とクライアントでは、次に示すファイルを J2EE アプリケーションサーバにデプロイします。

エンタープライズアーカイブファイル myservice.ear。 このファイルは、Java クライアントが生成されるときに自動的に作成されます。 このファイルには、次の内容が記述されています。

- EJB アーカイブファイル myservice.jar。 このファイルは、クライアントが生成されるときに作成されます。 また、サービスインターフェイスをデプロイするデプロイツールを使用するときにも作成されます。
- クライアントアーカイブファイル myservice.war。 このファイルは、クライアントが生成されるときに作成されます。

.ear ファイルが生成されず、.jar ファイルのみが存在する場合は、かわりに .jar ファイルをデプロイできます。J2EE アプリケーションは、デプロイする前に .jar ファイルを .ear ファイルにパッケージ化することがあります。その方法は簡単です。

提供されているリソースアダプタの 1 つである mfcobol*.rar。リソースアダプタを使用すると、EJB が J2EE アプリケーションサーバを通じてエンタープライズサーバと通信します。リソースアダプタは、J2EE コネクタとも呼ばれ、リソースアーカイブファイル (.rar) にパッケージ化されています。

J2EE Reference Implementation (RI) サーバ (開発専用)、BEA WebLogic、IBM WebSphere、その他多くの J2EE アプリケーションサーバがサポートされています。各アプリケーションサーバは、デプロイ要件や使用するユーティリティは異なります。サポートされているアプリケーションサーバとそのバージョンについての詳細は、Readme ファイルの『[サードパーティのソフトウェア](#)』を参照してください。

J2EE アプリケーションサーバを迂回する方法には、次の 2 つがあります。

リソースアダプタを使用せずに COBOL エンタープライズサーバに直接アクセスする、Java bean を生成する方法。この方法では J2EE アプリケーションサーバの管理オーバーヘッドは防げますが、トランザクションサポートなどの利点が失われます。詳細は、COBOL 開発システムに付属の『Java と COBOL』の『Java インターフェイスマッピングとリソースアダプタの使用法』の章にある『J2SE の管理非対象コネクションの使用法』の節を参照してください。

独自にコードを作成してリソースアダプタを呼び出してエンタープライズサーバにその要求を送信する方法。その場合は、接続は管理されない状態になります。作成したコードは、J2EE アプリケーションサーバの環境ではなく、J2EE サーバのライブラリを含む Java 2 Standard Edition (J2SE) 環境で実行されます。詳細は、COBOL 開発システムに付属の『Java と COBOL』の『Java インターフェイスマッピングとリソースアダプタの使用法』の章にある『J2SE の管理非対象コネクションの使用法』の節を参照してください。

リソースアダプタ

Micro Focus 製のリソースアダプタを使用すると、J2EE 上にデプロイした EJB がエンタープライズサーバ上の COBOL と連携できるようになります。リソースアダプタは、J2EE コネクタとも呼ばれます。

次のリソースアダプタがあります。

mfcobol-notx.rar - トランザクションはサポートしません。

mfcobol-localtx.rar - ローカルトランザクションをサポートします。

mfcobol-xa.rar - XA トランザクションをサポートします。

mfcobol*ora.rar - Oracle J2EE アプリケーションサーバ用の上記のリソースアダプタ

です。

リソースアダプタには、J2EE 1.3 用のバージョンと、J2EE 1.4 用のバージョンの 2 つがあり、COBOL 開発システムの base¥bin¥j2ee13 および base¥bin¥j2ee14 のそれぞれのディレクトリに格納されています。また、Enterprise Server が収録されている CD にも格納されています。

EJB を Enterprise Server 上で動作する COBOL と連携させるには、Java 環境にリソースアダプタをデプロイする必要があります。一度にデプロイできるのは、1 つのリソースアダプタのみです。このリソースアダプタは、エンタープライズサーバに送られるすべてのバイナリ要求を処理します。

WebLogic、Sun 社の Reference Implementation、および Oracle では、これらのアーカイブファイルはデプロイディスクプリタを含んでいて、リソースアダプタをデプロイするために必要なすべての情報を提供します。weblogic-ra.xml などのデプロイディスクプリタは、アーカイブファイル mfcobol-notx.rar で確認できます。次の行が含まれているか確認してください。

```
<connection-factory-name>CCIMFCobol_v1.0</connection-factory-name>  
<jndi-name>eis/MFCobol_v1.0</jndi-name>
```

WebLogic、Sun 社の Reference Implementation、および Oracle 以外のアプリケーションサーバでは、リソースアダプタをデプロイするときに次に示す情報を指定する必要があります。

接続ファクトリの名前 CCIMFCobol_v1.0
Enterprise Server の JNDI 名 eis/MFCobol_v1.0

注：リソースアダプタは、Enterprise Server または COBOL 開発システムのどちらかがインストールされたマシン上にデプロイする必要があります。

デプロイディスクプリタ

デプロイディスクプリタは、デプロイする構成要素を記述した xml ファイルです。J2EE アプリケーションサーバには、デプロイするアーカイブファイル (.jar、.war、および .ear) 用のデプロイディスクプリタが必要です。実行時に、アプリケーションサーバはディスクプリタを処理し、それに従って動作します。

EJB またはクライアント生成時に、汎用的なデプロイディスクプリタが作成され、関連するアーカイブファイルにパッケージ化されます。いくつかの J2EE アプリケーションサーバは、追加のデプロイディスクプリタを必要とします。これらは、サポートされているほとんどのアプリケーションサーバ用に自動的に生成されます。

詳細は、COBOL 開発システムに付属の『Java と COBOL』の『Java インターフェイスマッピングとリソースアダプタの使用法』の章にある『デプロイディスクプリタ』の節を参

照してください。

CustomRecord および RuntimeProperties サポート

CustomRecord インターフェイスと RuntimeProperties インターフェイスのサポートは、EJB の場合には mfejlib.jar、J2SE Java bean の場合には mfj2se.jar で提供されます。

J2SE Java bean の場合は、mfj2se.jar が Java ランタイムシステムで利用可能であり、Java クラスパス上にあることを確認する必要があります。

EJB の場合、インターフェイスのサポートは、EJB 用のクライアント生成時にアプリケーションの .ear ファイルに自動的にインクルードされます。手動でクライアントを作成する場合は、.ear ファイルに mfejlib.jar を手動で追加します。このファイルには、J2EE 1.3 用のバージョンと、J2EE 1.4 用のバージョンの 2 つがあり、COBOL 開発システムの base¥bin¥j2ee13 および base¥bin¥j2ee14 のそれぞれのディレクトリに格納されています。また、Enterprise Server が収録されている CD にも格納されています。

Sun 社の Reference Implementation にデプロイする場合は、mfejlib.jar への正しいパスが指定されていることを確認する必要があります。

[方法](#)

EJB のデプロイ

EJB のデプロイには、J2EE マシン上でサードパーティのアプリケーションサーバを、EJB、リソースアダプタ、および Enterprise Server がそれぞれ互いの位置を認識するように構成する作業が含まれます。

EJB のデプロイには、サードパーティのアプリケーションサーバ用のインターフェイスを使用します。多くのアプリケーションサーバは、WebLogic Server Console などの Web ベースのインターフェイスを提供し、デプロイプロセスを手順に従って行います。ほとんどのサードパーティのアプリケーションサーバは、クライアント、すべての必要なデプロイディスクリタ、および他のファイルとともにアプリケーションの .ear ファイル内にパッケージ化された EJB が必要です。.ear ファイルは、Interface Mapping Toolkit を使用してクライアントを作成するときに自動的に生成されます。

アプリケーションの .ear ファイルがない場合は、.ear ファイルを作成するためにサーバの管理インターフェイスや jar コマンドを使用します。

[方法](#)

J2EE Reference Implementation

J2EE Reference Implementation (RI) は、アプリケーションサーバとして機能します。これ

は実際の運用ではなく、開発の際に使用します。これは、J2EE ソフトウェア開発キット (software development kit; SDK) の一部として提供されています。EJB とリソースアダプタをデプロイするには、これをインストールする必要があります。

J2EE RI に関連するのは、次のものです。

J2EE RI サーバ自体。

管理ユーティリティ。これは、主にリソースアダプタを Java Naming and Directory Interface (JNDI) 名にバインドするために使用します。JNDI 名は、容易に識別できるようにリソースに付けた名前です。ヘルプを表示するには、次のコマンドを使用します。

```
j2eeadmin -help
```

デプロイツールユーティリティ。これは、リソースアダプタと EJB をデプロイします。このユーティリティには、グラフィカルユーザインターフェイス (graphical user interface; GUI) とコマンド行インターフェイスがあります。ヘルプを表示するには、次のコマンドを使用します。

```
deploytool -help
```

サーバ上にデプロイしたモジュールを変更した場合は、Sun 社のマニュアルでは、サーバを停止し再開するよう推奨しています。

注：エンタープライズサーバを停止し再開する場合は、J2EE RI サーバも停止し再開する必要があります。

J2EE Reference Implementation のインストール

J2EE RI を使用するには、次に示すソフトウェアが必要です。これらは、[Sun 社の Web サイト](#)からダウンロードできます。

JDK (Java Development Kit; Java 開発キット)。JDK をインストールした場合は、JAVA_HOME 環境変数を JDK のインストールディレクトリに設定します。

J2EE SDK。これには、J2EE RI のソフトウェアとマニュアルが含まれています。

SDK をインストールした場合は、J2EE_HOME 環境変数を SDK のインストールディレクトリに設定します。

環境変数とパスを設定するサンプルスクリプトを、次に示します。SDK と JDK は、デフォルト位置にインストールされていると仮定しています。

```
set java_home=c:\sun\appserver\jdk
set j2ee_home=c:\j2skee
set path=c:\j2skee\bin;c:\sun\appserver\jdk\bin;%path%
```

インストールの内容を検証するには、Web ブラウザを開き、URL `http://localhost:8000` を入力します。インストールが成功した場合は、J2EE のデフォルトのホームページが表示されます。

J2EE RI へのリソースアダプタのディプロイ

リソースアダプタのディプロイには、J2EE RI マシン上でサードパーティのアプリケーションサーバを、EJB、リソースアダプタ、および Enterprise Server がそれぞれ互いの位置を認識するように構成する作業が含まれます。

リソースアダプタを J2EE RI へディプロイするには、次の作業を行います。

1. リソースアダプタをディプロイツールユーティリティの `-deployConnector` コマンドを使用してディプロイする。
2. `j2eeadmin` ユーティリティの `-addConnectorFactory` を使用して接続ファクトリを追加する。これにより、リソースアダプタが Enterprise Server の JNDI 名 (`eis/MFCobol_v1.0`) にバインドされます。
3. EJB のリソースリファレンスで指定することで、リソースアダプタが EJB によって確実に使用されるようにする。『[J2EE RI へ EJB のディプロイ](#)』を参照してください。

次のサンプルスクリプトでは、デフォルトポート 9003 を使用して、リソースアダプタ `mfcobol-notx.rar` をローカルマシン `localhost` 上で動作する J2EE RI サーバにディプロイします。J2EE は `j2sdkee` にインストールされていると仮定しています。

```
set RAR=mfcobol-notx.rar
copy "c:\NetExpress\base\bin\%RAR%" c:\j2sdkee\lib\connector
call deploytool -deployConnector c:\j2sdkee\lib\connector\%RAR% localhost
call j2eeadmin -addConnectorFactory eis/MFCobol_v1.0 %RAR% Trace=true
call deploytool -listConnectors localhost
```

[方法](#)

J2EE RI への EJB のディプロイ

EJB を J2EE RI へディプロイするには、ディプロイツール UI やディプロイツールコマンドを使用します。EJB をディプロイする場合は、最初にディプロイツール UI の使用をお奨めします。ディプロイツール UI を使用すると、必要なディプロイディスクプリタを作成し、`.ear` ファイルをパッケージ化します。

[方法](#)

J2EE RI へのディプロイのサンプル EJB

サンプルエンタープライズアーカイブファイル `MapDemo-RI.ear` は、開発システムとともに

出荷され、Examples¥Net Express IDE¥mapdemo¥Client¥Java¥lib に格納されています。

サンプル MapDemo-R1.ear は、次のファイルを含んでいます。

MapDemo.war - MapDemo COBOL サービスを使用するクライアントソフトウェア (JSP、servlet、Bean クラスなど) を含む Web アーカイブファイル。このクライアントは生成されず、ハンドコードされます。

JMapServ.jar - 既製の EJB などを含むアーカイブファイル。

mfejblib.jar - 提供されている EJB が使用する CustomRecord と RuntimeProperties の Micro Focus サポートを含むアーカイブファイル。

さまざまなデプロイディスクプリタとマニフェストファイル。

サンプルアプリケーション MapDemo-R1.ear をデプロイするには、次に示すデプロイツールを使用します。

```
deploytool -deploy MapDemo-R1.ear localhost
```

localhost は、J2EE RI サーバが動作しているマシン名です。

MapDemo アプリケーションを表示するには、ブラウザに次の URL を入力します。

```
http://localhost:8000/MapDemo/MapDemo.jsp
```

サードパーティアプリケーションサーバのデプロイ

開発と運用で使用する、サードパーティアプリケーションサーバがいくつかサポートされています。たとえば、WebSphere や WebLogic があります。サポートされている全アプリケーションサーバの一覧は、Readme の『[サードパーティのソフトウェア](#)』を参照してください。

ほとんどのサードパーティアプリケーションサーバには、Web ベースの管理コンソールが付属しています。この管理コンソールを使って、EJB とリソースアダプタをデプロイします。このコンソールで、デプロイを手順に従って行います。また、オンライン情報を利用できます。

[方法](#)

サーバ上にデプロイしたモジュールを変更した場合は、サーバを停止し再開するよう推奨しています。

トラブルシューティング

デプロイした EJB が正常に機能しない場合には、次のことを確認してください。

アプリケーションサーバを停止し再開しましたか? ディプロイと構成の変更を有効にするために、これを行うことが必要な場合があります。WebLogic マシンをリブートするたびに、WebLogic サーバも [スタート] メニューから起動する必要があります。必要なエンタープライズサーバが動作していますか?

エンタープライズサーバに必要なサービスがディプロイされていますか? ディプロイに成功していますか?これらは、Enterprise Server Administration Console (<http://localhost:86>)、`deploy¥...¥deploylog.txt`、および `workarea¥myes¥console.log` を参照することで確認できます (myes は ESDEMO などのエンタープライズサーバ名で COBOL 開発システムのインストールディレクトリにあります)。

リソースアダプタのログファイル `cobcon.log` に情報が書き込まれていませんか? このログファイルは、Java システムの `user.home` 属性で定義したディレクトリに置かれています。このディレクトリは、デフォルトではユーザのプロファイルディレクトリと同じです。ログファイルに何の記述もない場合には、障害はリソースアダプタが起動する前に発生しています。障害は、アプリケーションサーバの設定時に発生した可能性があります。バックアップログファイル `cobcon1.log` および `cobcon2.log` が存在するか確認してください。

通信サーバのログファイル `log.html` に情報が書き込まれていませんか? 『トラブルシューティング』の章にある『[通信サーバのログファイル](#)』の節を参照してください。

ディプロイツールを使用して EJB を作成した場合に、アプリケーションサーバの J2EE コネクタクラスを使用しましたか?たとえば、WebLogic EJB の生成では `WebLogic.jar`、WebSphere EJB の生成では `j2ee.jar` を使用する必要があります。これらのクラスはディプロイツールの Classpath 設定で指定します。

リソースアダプタの JNDI 名が `eis/MFCobol_v1.0` に設定されていますか? この JNDI 名は、EJB のリソースリファレンスにマップされていますか?

各 EJB の JNDI 名は、それを参照する際に使用される名前と同じですか? EJB に JNDI 名を割り当てた場合は、その EJB を使用するすべてのクライアントや他のソフトウェアはその同じ JNDI 名を使用する必要があります。

EJB とリソースアダプタは、アプリケーションサーバに適切にディプロイされましたか?

アプリケーションサーバのログファイルにどのような障害が記録されましたか?サポートされているすべてのアプリケーションサーバには、`logs` サブディレクトリがあります。

第 4 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。

Web インターフェイスを使用して、インターネットやイントラネットでエンタープライズサーバを監視および管理するには、次に示すブラウザのどれかをインストールする必要があります。

Internet Explorer V6.0 以降
Netscape V6.0 以降
Mozilla V1.3 以降

注：デフォルトでは、Directory Server はポート 86 を使用します。Directory Server マシンでファイアウォールが有効な場合は、このポートへのアクセス許可が設定されていることを確認してください。

Enterprise Server Administration を使用する前に

Enterprise Server Administration を一般的な方法で使用するには、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。

Directory Server の起動と停止

Enterprise Server をインストールすると、Directory Server が Windows サービスとして自動的に実行されます。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。

また、mfds コマンドを使用して Directory Server をコマンド行から起動することもできます。コマンド行から Directory Server を起動する方法は、サービスとして実行しているアプリケーションをデバッグする場合に便利です。mfds コマンドを使用する前に、Windows サービスとして実行している Directory Server を停止する必要があります。mfds コマンドを使用して Directory Server を起動した場合は、Web インターフェイスのメニューオプションを使用して停止する必要があります。

また、mfds コマンドには、Directory Server をインストール解除するオプション、および Windows サービスとして再インストールするオプションもあります。

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。

Net Express を実行している場合には、IDE メニューの [ツール > Enterprise Server > Administration] をクリックします。

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。

[方法](#)

アクセスとアクセス権

Enterprise Server Administration で表示される画面と、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。

使用中のユーザアカウントが属しているユーザグループに対して割り当てられているアクセス権。

詳細は、『[ユーザ、グループおよびアクセス権](#)』の章を参照してください。

実際の画面

ログオンすると、図 4-1 に示すホームページが表示されます。実際の表示画面は、アクセスが制限されているかどうかによって決まります。アクセスが制限されている場合は、ログオンしたアクセス権レベルに応じて決まります。図 4-1 に、Schema Administrator アクセス権レベルのホームページを示します。

図 4-1: 「Enterprise Server Administration」のホームページ

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。

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](#)』

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、ホームページ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。ホームページの最初の表示では、テーブルは要約モードで表示され、サーバ内の各オブジェクトの数が表示されます。展開モードに切り替えるには、サーバテーブルの左上で をクリックします。展開モードでは、各オブジェクトの情報が表示されます。要約モードに戻るには、 をクリックします。サーバテーブルには、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。

サーバテーブルには、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。

編集するサーバ、または表示する他のテーブル(サービステーブルなど)を選択するとすぐに、ページはタブ表示に変わります。これらのタブを使用して、通信プロセス、サービス、要求ハンドラ、および実装パッケージの表示を切り替えることができます。図 4-2 に、「サーバの編集」ページを示します。先頭行のタブ名に続くカッコ内の数字は、サーバに登録されているそのタイプのオブジェクト数を示します。

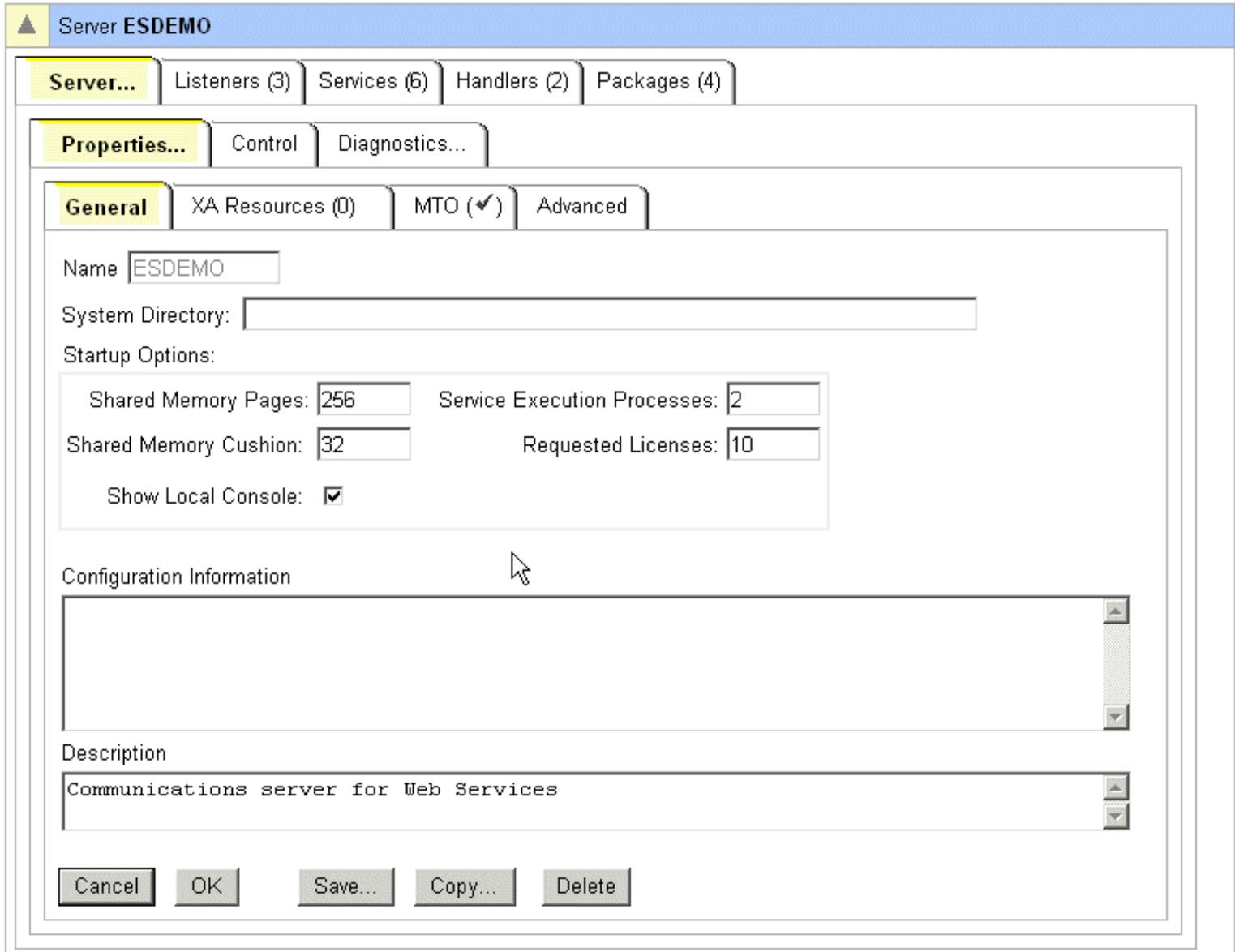


図 4-2: 「サーバの編集」ページでタブを表示

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。

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 [追加] をクリックします。

左側パネルの上部のメニューで [ホーム] をクリックするか、またはサーバの詳細ページか任意のオブジェクトの詳細ページの左上にある  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。

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。一部のページには、自動的に画面を更新する時間間隔 (秒数) を指定するための「自動画面更新間隔」フィールドもあります。デフォルト (かつ最小) の値は 10 秒です。空白にすると、自動更新は行いません。

マニュアルの表示

Enterprise Server Administration Web インターフェイスを使用しているときに、ページの左側にあるメニューで [目次] をクリックすると、別ウィンドウに Enterprise Server のマニュアルが表示されます。このリンクが正しく動作するための条件は次のとおりです。

使用しているエンタープライズサーバに、[virtual paths] セクションに次のエントリを持つ Web リスナーが必要です。

```
docs=<ES>/docs
```

エンタープライズサーバが起動されています。

デフォルトのエンタープライズサーバである ESDEMO には、この構成情報をもつ Web リスナーがあります。新しくエンタープライズサーバを作成すると、必要な構成情報をもつ Web リスナーが自動的に作成されます。

第 5 章：ユーザ、グループおよびアクセス権

エンタープライズサーバで実行するアプリケーションは、重要なビジネス機能を実行し、また機密情報を処理することがあります。このため、これらのアプリケーションではセキュリティを高めるための手段を考慮する必要があります。このような手段のひとつは、エンタープライズサーバの構成と実行を行う機能を制限することです。

一般には、Micro Focus Directory Server の Web インターフェイスである Enterprise Server Administration を使用してエンタープライズサーバを管理します。Directory Server は、エンタープライズサーバの構成情報を保存します。

ここでは、Micro Focus Directory Server と Enterprise Server Administration の機能へのアクセスを制御する方法について説明します。

前提条件

アクセスを制御するには、Directory Server を制限モードで実行する必要があります。制限モードで実行しない場合、Enterprise Server Administration の機能はすべて、ログオンせずにアクセスできます。

[方法](#)

これ以降では、Directory Server が制限モードで実行されていると仮定します。

概要

Directory Server の内部でオブジェクトとスキーマの設定を表示および変更する機能は、システム全体で、ユーザアカウント、ユーザグループおよびアクセス権によって制御されます。管理機能を使用するユーザは、ログオンしなければなりません。ユーザが使用できる機能は、アクセス権で制御されます。アクセス権は、ユーザグループに設定します。設定されたグループ内のユーザは、これらのアクセス権で管理されるアクションを実行できます。

ユーザおよびグループについて

制限モードで実行中の Directory Server の Enterprise Server Administration に接続すると、要約画面が表示されます。この画面は、Directory Server が管理するエンタープライズサーバの基本情報を詳細に示します。

他のアクションを実行する前に、ログオンする必要があります。ログオンするには、ユー

ザアカウントが必要です。 Directory Server には事前に設定済みのユーザアカウントが多数ありますが、別のアカウントを作成することもできます。

ログオンした時に使用できる機能は、そのユーザが属しているユーザグループに設定されたアクセス権によって決まります。

どのユーザも、複数のグループに所属できます。 グループに関しても、Directory Server には事前に設定済みのグループがいくつかありますが、別のグループを作成することもできます。 事前に設定済みのグループにはそれぞれ、特定の管理作業を実行するための適切なアクセス権があります。 各グループには、それぞれ 1 つまたは複数のユーザが設定されていますが、Micro Focus アプリケーションが使用する System グループは例外です。

グループ	説明	事前に設定されているユーザ
All Users	<p>これは、アクセス権が最も制限されているグループです。 デフォルトの Enterprise Server アクセス権は、読み取り専用設定されています。 このアクセス権により、グループ内のユーザは、エンタープライズサーバの構成情報を参照できます。 すべてのユーザはこのグループに属しています。</p> <p>このグループには、次の管理作業が有効になるデフォルトの Enterprise Server アクセス権が設定されています。</p>	<p>schemaadmin、adddelete、modify、administrator</p>
Add/Delete	<p>サーバの追加および削除 サービス、リスナー、パッケージの、サーバへの追加とサーバからの削除</p>	<p>adddelete、administrator</p>
	<p>サービス、リスナー、パッケージのサーバへの追加</p>	

General Administrators	<p>このグループには、エンタープライズサーバに対する管理作業がすべて有効になるデフォルトのエンタープライズサーバアクセス権が設定されています。また、ユーザの管理を除くすべての Directory Server 管理作業も許可されています。</p>	administrator
Directory Server Administrators	<p>このグループには、Directory Server およびエンタープライズサーバのすべての管理作業に必要なアクセス権が設定されています。</p>	schemaadmin
Modify	<p>このグループには、エンタープライズサーバの構成情報の変更を有効にするデフォルトのエンタープライズサーバアクセス権が設定されています。</p>	modify
System	<p>これは、Micro Focus アプリケーションを使用するためのグループです。</p>	

アクセス権について

アクセス権は、Directory Server の機能の使用を制御します。たとえば、ユーザおよびグループを管理するためのアクセス権があります。アクセス権はユーザグループに設定します。

アクセス権には 2 つのカテゴリがあります。

Directory Server Administration	<p>Directory Server のリポジトリを管理します (たとえば、ユーザアカウントの管理など)。</p>
Server	<p>Directory Server が制御するエンタープライズサーバを管理します。</p>

アクセス権の中には、他のアクセス権を含むものがあります。たとえば、エンタープライズサーバを変更できるアクセス権には、サーバを起動および停止できるアクセス権も含まれています。アクセス権を割り当てると、そのアクセス権に含まれるすべてのアクセス権も自動的に付与されます。

Directory Server のアクセス権

Directory Server の使用可能なアクセス権を示します。

名前	説明
Save Repository	リポジトリのコピーをファイルシステムディレクトリに保存することを許可します。 ファイルシステムディレクトリからのデータのインポートを許可します。
Import Repository	含まれるアクセス権： Save Repository ファイルシステムディレクトリからリポジトリを復元することを許可します。復元すると、現在のリポジトリの情報はすべて上書きされます。
Restore Repository	含まれるアクセス権： Save Repository、Import Repository すべてのエンタープライズサーバ、およびすべてのユーザとグループを含め、Directory Server リポジトリ全体の削除を許可します。
Delete Repository	含まれるアクセス権： Save Repository、Import Repository Directory Server のオプションの変更を許可します。
Change MF Directory Server options	含まれるアクセス権： Save Repository、Import Repository 個々のエンタープライズサーバでのアクセス権の割り当てを許可します。
Set Server Permissions	含まれるアクセス権： Save Repository、Import Repository、Restore Repository、Delete Repository、Change MF Directory Server

Directory Server がネットワーク上で検出できる Directory Server のリストの表示を許可します。

Display Directories

含まれるアクセス権： Save Repository、 Import Repository、 Restore Repository、 Delete Repository、 Change MF Directory Server、 Set Server Permissions

Directory Server のシャットダウンを許可します。

Shutdown MF Directory Server

含まれるアクセス権： Save Repository、 Import Repository、 Restore Repository、 Delete Repository、 Change MF Directory Server、 Set Server Permissions

ユーザアカウントおよびユーザグループの管理、アクセス権の設定を許可します。

Administer Users

含まれるアクセス権： Save Repository、 Import Repository、 Restore Repository、 Delete Repository、 Change MF Directory Server、 Set Server Permissions、 Display Directories、 Shutdown MF Directory Server

サーバのアクセス権

サーバのアクセス権を示します。

名前	説明
Read	エンタープライズサーバの構成情報の表示を許可します。 エンタープライズサーバの起動と停止を許可します。
Start/Stop	含まれるアクセス権： Read エンタープライズサーバの構成情報の変更を許可します。
Modify	含まれるアクセス権： Read、 Start/Stop

リスナー、サービス、パッケージを、エンタープライズサーバに追加することを許可します。グループのデフォルトアクセス権として設定された場合は、エンタープライズサーバの追加も許可します。

Add

含まれるアクセス権： Read、Start/Stop、Modify

エンタープライズサーバの削除と、エンタープライズサーバからリスナー、サービス、パッケージを削除することを許可します。

Delete

含まれるアクセス権： Read、Start/Stop、Modify

グループにはデフォルトのサーバアクセス権を設定できますが、個々のエンタープライズサーバに対してこのデフォルトを無効にすることもできます。たとえば、オペレータグループに、サーバの起動と停止ができるデフォルトアクセス権を設定したとします。この設定は、オペレータグループが Modify アクセス権も持つように指定すると、テストサーバでは無効にできます。

注：新しいサーバの作成をグループに許可するには、グループのデフォルトアクセス権を Add に設定する必要があります。

ユーザおよびグループの管理

Enterprise Server Administration 内の画面を使用して、ユーザおよびグループの管理ができます。この管理を行うには、適切なアクセス権をもつグループに属するユーザアカウントでログオンする必要があります。事前に設定済みのユーザである schemaadmin がこの作業に適しています。

[方法](#)

アクセス権の設定

Enterprise Server Administration 内の画面を使用して、各グループにアクセス権を設定します。この設定を行うには、適切なアクセス権をもつグループに属するユーザアカウントでログオンする必要があります。事前に設定済みのユーザである schemaadmin がこの作業に適しています。

[方法](#)

最適な実施方法

できるだけ早い段階で、設定済みアカウントのパスワードを変更します。

アクセスする必要があるユーザのみがアクセス権をもつことを確認します。

ユーザが、その作業を行うために必要なアクセス権のみをもつことを確認します。

管理ユーザの数を制限します。

前バージョンからの移行

Directory Server の前バージョンでは、各ユーザに 4 種類のアクセス権レベルのいずれかを割り当てることでユーザのアクションを制御していました。そのレベルは、Modify、Add/Delete、Administrator、Schema Administrator です。

Micro Focus Studio にアップグレードする場合、Directory Server のユーザアカウントは、新しいグループとアクセス権モデルに自動的に移行されます。各ユーザは、そのアクセス権レベルに相当するアクセス権をもつデフォルトグループに割り当てられます。

アクセス権レベル	相当するグループ
Modify	Modify
Add/Delete	Add/Delete
Administrator	General Administrator
Schema Administrator	Directory Administrator

結果として、ユーザは以前と同じように Enterprise Server Administration にアクセスできます。

前回のリリースでは、ユーザ情報は CCIUSERS.dat ファイルに格納されていました。Enterprise Server Administration のユーザのインポート機能を使用すると、前バージョンで実行しているその他の Micro Focus Directory Server の CCIUSERS.dat ファイルからユーザを手動でインポートすることができます。この場合、制限のないモードで Directory Server を実行する必要があります。

第 6 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。

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
- 通信プロセス
- サービスリスナー
- 要求ハンドラ
- 実装パッケージ
- XA リソース

リポジトリファイルを常に使用可能な状態に保ち、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、最新の保存場所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。この最新の保存場所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。

デフォルトでは、エンタープライズサーバの情報のみが保存されます。バックアップと復元の対象に CCI サーバを含めるには、「構成オプション」ページで CCI サーバを指定します。

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。

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、C:\Documents and Settings\All Users\Application Data\Micro Focus\NetExpress\5.0\MFDS\ です (%SYSTEMDRIVE% は c: であることが前提です)。

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、次の作業が行えます。

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は、次の作業が行えます。

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。バックアップ内のサーバ名がすべて現在のリポジトリ内のサーバ名と一致すると、復元は失敗します。この動作は無効にできます。C:\Documents and Settings\All Users\Application Data\Micro Focus\NetExpress\5.0\MFDS\ivp から復元を行うと、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。

リポジトリ全体を削除する。

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 7 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。

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、CICS、および JCL 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。

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については、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 CCI (Common Communications Interface; 共通通信インターフェイス) 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

コマンド行インターフェイス

サーバの管理には、多数のコマンドが使用できます。これらは、次のとおりです。

casstart	エンタープライズサーバを起動します。
casstop	エンタープライズサーバを停止します。
cassi	サービス実行プロセスを開始します。
casdump	システムダンプを要求します。
casdup	トレースまたはダンプデータセットをフォーマットします。

詳細については、ヘルプトピックの『[コマンド](#)』を参照してください。

サーバ情報

「Micro Focus 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』の章の『[実際の画面](#)』の節にある『「Enterprise Server Administration」のホームページ』の図を参照してください。図 7-1 に、展開モードのサーバテーブルを示します。

MICRO FOCUS Enterprise Server Administration
Version 1.03.63
user.microfocus.com (12.220.78.128)

Status: MDS00001 OK Fri Aug 27 09:55:30 2004
[SCHEMA ADMINISTRATOR-schemaadmin-b43f780a412ef231] [Page id: 1000]

Refresh Auto-refresh interval (seconds)

1 - 1 of 1 Servers

Type	Name	Current Status	Communications Processes	Licenses	Last Status Change	Status Log	Services	Request Handlers	Implementation Packages	Description
MFES	ESDEMO	Stopped Start	1 top:12.220.78.128*:9000 (user.microfocus.com) 3 Listeners Details	- / 10	08/27/04 09:35:39	CASST00011 Server manager termination completed successfully 9:35:37 19 minutes 51 seconds in "Stopped" state since 09:35:39	1 Test [1] 2 Deployer [1] Details	1 MFRHBINP 2 MFRHSOAP Details	0	Communications server for Web Services

[Add...](#)

図 7-1: 展開モードのサーバテーブル

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
- 名前
- 現ステータス
- 通信プロセス

エンタープライズサーバの場合は、各プロセスの通信プロセスの一覧を表示します。

通信プロセス番号。この番号の背景が青色の場合、通信プロセスは自動開始プロセスです。つまり、エンタープライズサーバの起動時に自動的に開始されます。番号の背景が灰色の場合は自動開始プロセスではありません。エンタープライズサーバが、Directory Server からの管理メッセージを受信するために使用するネットワークアドレス。通信プロセスが所有するリスナーの数。

CCI サーバの場合は、クライアントの要求を受信するために使用するネットワークアドレスのみを

表示します。

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。

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

- ライセンス：ライセンス情報。 スラッシュ記号の前にある数字は、このサーバに割り当てられているライセンスの数です。 ダッシュ記号は、ライセンスが割り当てられていないことを示します。 スラッシュ記号の後にある数字は、このサーバが要求するライセンスの数です。 アスタリスク記号は、無制限のライセンス数が要求されたことを示します。
- 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

- ステータスログ：要約モードと同じ。
- ライセンス：要約モードと同じ。
- サービス：このサーバにディ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」となります。 サーバに関連付けられたオブジェクトにチェックマークが付いている場合は、実行可能であることが示されます。

通信プロセスが開始されている場合は、通信プロセスにチェックマークが付きます。

リスナーが開始され、通信プロセスが開始されている場合には、リスナーにチェックマークが付きます。

サービスは、それに関連付けられている 1 つまたは複数のリスナーが開始されている場合に、チェックマークが付きます。

要求ハンドラが有効で、サーバが起動されている場合、その要求ハンドラにチェックマークが付きます。

実装パッケージが有効で、サーバが起動されている場合、その実装パッケージにチェックマークが付きます。

サーバが起動されている場合、XA リソースにはチェックマークが付きます。

構成情報

「サーバの追加」ページおよび「サーバの編集」ページの「構成情報」フィールドで、エンタープライズサーバに特有のデータを設定できます。 このフィールドには、次のセクションを設定できます。

[ES-Environment]

このセクションの設定内容は、このエンタープライズサーバで実行されるプログラムの環境に追加されます。 このセクションを使用して、プログラムに必要な任意の環境変数 (DDNAME など) を設定したり、システム変数または COBOL ランタイム変数 (PATH、COBPATH など) を設定することができます。 UNIX では、このセクションでの設定は大文字と小文字を区別します。 環境変数の設定についての詳細は、『構成』の章の『[環境変数](#)』の節を参照してください。

「サーバの追加」ページおよび「サーバの編集」ページの「構成情報」フィールドを使用して、そのエンタープライズサーバに属する通信プロセスのパラメータを設定することもできます。 サーバオブジェクトの通信プロ

セス設定は、そのエンタープライズサーバのすべての通信プロセスに適用されます。1つの通信プロセスだけにパラメータを設定するには、「通信プロセスの追加」ページまたは「通信プロセスの編集」ページの「構成情報」フィールドを使用します。入力可能な文字については、『通信プロセスとサービスリスナー』の章の『[構成情報](#)』の節を参照してください。

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。少なくとも、1つの通信プロセスが開始されています。
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、次回サーバモニタがサーバがアクティブな状態にあるかどうかを確認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。

サーバで回復不可能な通信エラーが発生することがあります。つまり、通信プロセスがすべて非アクティブになったり、応答しなくなるようなエラーです。このような場合、サーバプロセスはシャットダウンします。「サーバの詳細」ページで「サーバ開始」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。

起動スクリプトと停止スクリプトの使用

起動スクリプトと停止スクリプトを用意して、エンタープライズサーバの起動および停止の前後に行う動作を指定することができます。起動スクリプトには `casstart` コマンド、停止スクリプトには `casstop` コマンドをそれぞれ記述する必要があります。これらのコマンドを記述しない場合、サーバは起動も停止もされません。コマンドの詳細は、ヘルプトピックの『[casstart](#)』および『[casstop](#)』を参照してください。

Directory Server は、スクリプトの環境の環境変数をいくつか設定します。設定した環境変数はスクリプト内で使用できます。

ES_HOME	サーバの作業ディレクトリ (ログファイルの位置)。この値は、「サーバの追加」ページ (または「サーバの編集」ページ) > 「プロパティ」ページ > 「全般」ページの「システムディレクトリ」フィールドから取得します。
---------	---

ES_SERVER	サーバ名 (casstart または casstop で -r スイッチが指定されていない場合、Enterprise Server は、この値をサーバ名として使用することに注意してください)。
MFDS_PORT	Directory Server が受信に使用するポート (通常は 86)。

たとえば、次の起動スクリプトを使用して、サーバの環境変数を設定します。

```
set MY_VARIABLE=value
rem start the enterprise server
echo Enterprise Server %ES_SERVER% is starting
casstart
```

ここで、value は環境変数の値です。

サーバが完全に停止した後にのみ実行するコマンドが停止スクリプトに含まれている場合は、casstop コマンドを実行した後、そのサーバの cascd プロセスが終了するまでスクリプトを一時停止させます。これは、casstop コマンドが、サーバが実際にシャットダウンを完了する前に戻ってしまうためです。Windows でこれを行うには、サードパーティのツールが必要です。

次に、ログファイルをバックアップするか、またはサーバがシャットダウンを完了した後にのみ開始する他のタスクを実行できます。

サーバの [開始] または [停止] ボタンをクリックすると、指定したコマンドにより Windows バッチファイルが生成され、このバッチファイルが次に実行されます。

また、「サーバの無応答時」スクリプトも作成できます。このスクリプトは、サーバで実行中の通信プロセスがすべて「無応答」状態になったときに実行されます。たとえば、サーバがハングまたはクラッシュした場合、または、Directory Server とエンタープライズサーバ (これらが別々のシステムで実行されている場合) の間のネットワーク接続に問題がある場合などです。このスクリプトを使用して、システム管理者に問題を通知することができます。

起動、停止、および「サーバの無応答時」スクリプトは、「サーバの編集」ページ > 「プロパティ」ページ > 「高度な設定」ページで指定できます。

□ 方法

サーバの停止と再起動

Directory Server を停止する前、またはマシンをシャットダウンする前に、サーバをすべて停止する必要があります。Enterprise Server Administration を使用して Directory Server を停止する際には、先に全サーバを停止するというオプションがあります。

システムの再起動時に、「構成オプション」ページの「起動時にサーバを復元」がチェックされていない場合、Directory Server は、シャットダウンする前に停止されていなかったサーバ (Directory Server のリポジトリでまだ「開始」にマーク付けされている) を自動的に起動しません。「構成オプション」ページで [サーバモニタ] が有効になっている場合は、サーバがまだアクティブな状態にあるかをチェックし、ホームページにある Enterprise Server Administration サーバテーブルで「応答なし」にマーク付けします。この場合、各サーバは、手動で再起動する必要があります。各サーバの [詳細] をクリックし、次に [サーバの開始] をクリックしてサーバを再起動します。

[起動時にサーバを復元] が選択されている場合、Directory Server を再起動すると、以前に「開始」にマーク付けされていたサーバのリストを読み取り、サーバが実行中かチェックします。サーバが実行中でない場合は、サーバを再起動します。

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。

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

要求されたライセンス数の変更

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 XA リソースの追加

サーバに定義されている XA リソースの編集 (サーバが起動されていない場合)

XA リソースの削除 (サーバが起動されていない場合)

[方法](#)

ステータスが「開始」のサーバではいずれも、サーバの監視および制御を行うための、追加機能セットが使用できます。これらの追加機能を総称して Enterprise Server Monitor and Control と呼びます。詳細は、『[ESMAC 使用によるサーバの管理](#)』の章を参照してください。

CCI サーバ

Add/Delete またはそれ以上のアクセス権をもつ場合は、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にのみ、CCI サーバを削除します。

[方法](#)

第 8 章：通信プロセスとサービスリスナー

ここでは、通信プロセスとサービスリスナーオブジェクト、およびこれらを使用して実行できる作業について説明します。

はじめに

通信プロセスはエンタープライズサーバの構成要素で、クライアントとエンタープライズサーバの間の通信を処理します。通信プロセスには、多数のサービスリスナーがあります。エンタープライズサーバには、作成時には通信プロセスが 1 つありますが、通信プロセスは追加することができます。必要な通信プロセスの数については、『構成』の章の『[通信プロセスの数](#)』の節を参照してください。

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。

- 既存の通信プロセスをコピーして、通信プロセスを追加する
- 必要に応じて通信プロセスを更新または削除する
- 通信プロセスの開始と停止
- 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

通信プロセスとリスナー情報

リポジトリ内に保存されている、通信プロセスおよびリスナーの情報を表示するには、ホームページで、対象とするエンタープライズサーバの「通信プロセス」カラムで [詳細] をクリックします。通信プロセスおよびリスナーのテーブルを図 8-1 に示します。

Enterprise Server Administration > ESDEMO > Listeners
Version 1.03.63
user.microfocus.com (12.220.78.128)

Status: MDS00001 OK Fri Aug 27 10:00:04 2004
[SCHEMA ADMINISTRATOR-schemaadmin-b43f780a412ef231] [Page id: 1100]

Server ESDEMO

Server... Listeners (3) Services (2) Handlers (2) Packages (0) Refresh

Listener Display Filters Process: All Conversation Type: All Status: All

Communications Process 1 Auto-start Statistics Change Copy... Delete...

Listeners	Process ID	Control Channel Address	Status	Last Status Change	Status Log	Version
3 Add...		tcp:12.220.78.128*:9000 (user.microfocus.com)	Stopped	08/27/04-09:35:36	OK	1.3.5

Name	Address	Status	Last Status Change	Status Log	Conversation Type	Custom Configuration	Description
Edit... HTTP Echo	tcp:12.220.78.128*:9002 (user.microfocus.com)	Stopped	08/27/04-09:35:31	OK	http-echo		Simple HTTP test
Edit... Web	tcp:12.220.78.128*:1533* (user.microfocus.com)	Stopped	08/27/04-09:35:31	OK	Web	[virtual paths] cgi=<AEE>.bin uploads=<AEE>/deploy <default>=/dev/null docs=<es>/docs	Basic HTTP web server
Edit... Web Services	tcp:12.220.78.128*:9003 (user.microfocus.com)	Stopped	08/27/04-09:35:32	OK	Web Services and J2EE		Web Services via HTTP/SOAP

図 8-1: 通信プロセスとサービスリスナーのテーブル

通信プロセスの情報は、次のとおりです。

自動開始：エンタープライズサーバの起動時に、通信プロセスが自動的に開始されるかどうかを示します。

リスナー：通信プロセスが所有するリスナーの数。

プロセス ID：通信プロセスのプロセス ID。通信プロセスが実行中の場合にのみ表示されます。

コントロールチャンネルアドレス：エンタープライズサーバの構成要素が通信のために使用する内部的なアドレス。IP アドレスまたはポート番号の末尾にあるアスタリスクは、アドレスのこの部分が固定ではなく、起動時にホストシステムから与えられることを示します。

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：この通信プロセ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。

バージョン：通信プロセスソフトウェアのバージョン番号です。

通信プロセスには名前はなく、かわりに連続番号があります。新しくエンタープライズサーバを追加した時に自動的に作成される通信プロセスは Communications Process 1 です。その後作成される通信プロセスには、作成順に昇順の番号が付けられます。通信プロセスを削除した場合、既存の通信プロセスの番号は変わりませんが、次に作成される通信プロセスには、削除されたプロセスと同じ番号、つまり使用可能な最小の番号が付けられます。

リスナーの情報は、次のとおりです。

名前：リスナー名。

アドレス：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。IP アドレスまたはポート番号の末尾にあるアスタリスクは、アドレスのこの部分が固定ではなく、起動時に通信プロセスから与えられることを示します。

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：リスナーが処理するクライアント要求の種類を示します。値は、次のとおりです。

- Web Services and J2EE : SOAP または J2EE 要求。
- Web : World Wide Web (HTTP) 要求。Enterprise Server は、サービスのデプロイに Web リスナーを使用します。
- Fileshare : Fileshare クライアント要求。
- TN3270 : TN3270 ターミナルからの要求。
- MTO ISC : CICS Intersystem Communication (ピアツーピア) および、非 CICS プログラムからの CICS トランザクション要求。
- その他の要求種類のコネクタモジュール名。

TN3270 および MTO ISC は、Mainframe Transaction Option を使用した場合にのみ使用できます。

カスタム構成データ : 詳細は『[構成情報](#)』の節を参照してください。

説明

リスナー表示フィルタは、表示するリスナー情報を制御します。リスナー情報は、最初の通信プロセスの情報のすぐ上に表示されます。リスナー表示フィルタには、次の 3 種類があります。

プロセス (通信プロセス)
 会話タイプ
 ステータス

それぞれのフィルタには、フィールドに応じて多数の異なるオプションがあります。たとえば、「プロセス」フィルタを使用すると、リスナーの詳細をまったく表示しないか、全リスナーの詳細を表示するか、特定の通信プロセスに対してリスナーの詳細を表示するかを選択できます。また、「ステータス」フィルタでは、ステータスに関係なく全リスナーの詳細を表示するか、特定のステータスのリスナーのみ詳細を表示するかを選択できます。

たとえば、通信プロセス 2 のリスナーの詳細を表示するには、「リスナー表示フィルタ、プロセス」で [2] を選択します。Web サービスと J2EE のリスナーのみを表示するには、「リスナー表示フィルタ、会話タイプ」で [Web サービスと J2EE] を選択します。すべてのリスナーの詳細を表示するには、3 つすべてのフィルタで [すべて] を選択します。すべてのフィルタを組み合わせると、表示するレベルを最大限に制御できます。

通信プロセスおよびリスナーのテーブルを最初に表示すると、「プロセス」フィルタは [なし] に、他のフィルタは [すべて] に設定されています。

構成情報

通信プロセスまたはリスナーに特有の構成データについては、「通信プロセスの編集」ページ、「リスナーの追加」ページ、「リスナーの編集」ページの「構成情報」フィールドを使用できます。

通信プロセス

「通信プロセスの編集」ページの「構成情報」フィールドには、次のセクションと設定を指定できます。

[CCI]
 timeout-grain=timeout-grain-number-of-seconds
 [ISC]
 listener-wait=listener-wait-number-of-seconds
 [listeners]
 logging=logging-level
 [MFCC]
 trace=trace-setting

[threading]
limit=number-of-threads

パラメータを、次に示します。

timeout-grain

CCI タイムアウトを秒数で指定します。 MFCS タイムアウトは、この値の倍数に切り上げられます。

構文：

timeout-grain=timeout-grain-number-of-seconds

プロパティ：

デフォルト： 5 秒

備考：

MFCS (communications process; 通信プロセス) では、会話とリスナーのセットはデータを受信するタイムアウト値をそれぞれ独自に設定します。ただし、MFCS は CCI の一部として動作しており、CCI はプロセス全体に対してグローバルなタイムアウトを 1 つだけ保持しています。このため、MFCS タイムアウト値の精度は、CCI タイムアウトの設定による制限を受けます。デフォルトより大きな値を設定すると、MFCS タイムアウトの精度は低くなりますが、CPU の使用、割り込み、その他の要因が軽減されパフォーマンスが向上します。

listener-wait (MTO のみ)

ISC から他の MTO インスタンスまたは別の ISC 互換システムへの接続を開始する場合に、ローカルな ISC リスナーが使用可能になるまでの待機秒数を指定します。

構文：

listener-wait=listener-wait-number-of-seconds

プロパティ：

デフォルト： 10 秒

備考：

MFCS には、ISC からの通信をサポートするために使用可能な ISC リスナーが必要です。これは、起動時の ISC からの接続が設定されている場合に最も便利です。

logging

リスナーを起動する際に、追加ログ処理を有効にします。

構文：

logging=logging-level

パラメータ :

次の値が指定できます。

logging-level	0	追加ログ処理なし (デフォルト)
	1 または Yes	基本ログ処理。追加の CCI エラー情報も記録します。
	2	CCI のロード処理および構成、リスナーのアドレス情報も記録します。

プロパティ :

デフォルト : 0

trace

mf-client.dat ファイルとともに、MFCC (common client; 共通クライアント) のトレースを制御します。

構文 :

trace=trace-setting

パラメータ :

次の値が指定できます。

trace-setting	0	MFCC トレースを無効にします。
	ゼロ以外の任意の値	MFCS の起動時に MFCC トレースを有効にします。

プロパティ :

デフォルト : 0

備考 :

MFCC トレースはデフォルトでは無効ですが、mf-client.dat ファイルで有効にすることができます。MFCS は起動時に MFCC を使用して、コントロールチャンネルが動作していることを確認します。MFCC トレースが mf-client.dat ファイルで有効になっている場合は、このオプションを「0」に設定して MFCS 内での MFCC トレースを無効にできます。または、ゼロ以外の値に設定して、MFCS の起動時に MFCC トレースを有効にして、障害を診断することもできます。

limit

スレッド数の制限を指定します。

構文 :

limit=number-of-threads

プロパティ :

デフォルト : Windows では 2000、UNIX では 200

備考 :

MFCS は、システムの制限を超えないようにするために、開始するスレッドの数を制限します。現在、MFCS はアクティブな会話ごとに 1 つのスレッドを使用するため、この制限は同時に行われる会話の数に直接影響します。オペレーティングシステムは、指定された数のスレッドを MFCS が実際に作成することを許可しないことがあるので注意してください。

サービスリスナー

リスナーオブジェクトの使用可能な構成設定は、リスナーの会話タイプに応じて決まります。構成設定は、会話タイプによって次のようにグループ化されます。

Web Services と J2EE

この会話タイプに設定はありません。

Web

これらのオプションは、各要求に対してチェックされます。

[virtual paths] セクションは、URL で指定したトップレベルのパス要素と、それに対応する実際のファイルシステムディレクトリの間で変換を行うために使用します。たとえば、URL 「http://host/path/to/file」の場合、「path」のエントリに対して [virtual paths] セクションが照会されます。このセクションのエントリでは、大文字と小文字を区別します。この設定の詳細については、『構成』の章の『[ディプロイサービスとディプロイリスナー](#)』の節を参照してください。

[allow] セクションは、指定したディレクトリ以外で Web コネクタがサービスを行うファイルを制限します。

[options]

logging=logging-level

[virtual paths]

<default>=default-directory

element=file-system-path

[allow]

element=list-of-filenames

logging

記録する情報を制御します。

構文 :

logging=logging-level

パラメータ :

次の値が指定できます。

logging-level yes または 1 Web 要求の処理での、パス変換や CGI プログラム実行など、さまざまな情報を記録します。
その他の値 ログ処理なし

プロパティ :

デフォルト : ログ処理なし

<default>

認識されない URL のトップレベルのパス要素がマップされるディレクトリを指定します。

構文 :

default=default-directory

プロパティ :

デフォルト : /dev/null

備考 :

デフォルトのディレクトリは、存在しないディレクトリにする必要があります。これにより、通信プロセスが URL を完全パスに変換すると要求は失敗します。

element in [virtual paths] Section

element がマップされるファイルシステムパスを指定します。

構文 :

element=file-system-path

パラメータ :

element がマップされるファイルシステムパス。element は、URL のパスのトップレベル (先頭) 要素でなければなりません。値の先頭には、特殊なトークンを指定できます。

element <ES>: Windows の基本インストールディレクトリへのパスか、または UNIX の場合には \$COBDIR に置き換えられます。

<ASEE>: ES の従来同義語

<\$name>: 環境変数 name の内容に置き換えられます。

element in [allow] Section

1 つ以上のファイル名またはワイルドカードのパターン指定を、スペースで区切ってリストします。この仮想

パス上のファイルがコネクタでサービスされます。

構文：

```
element=list-of-filenames
```

パラメータ：

element element がマップされるファイルシステムパス。
ゼロ個以上のファイル名またはファイル名パターンを、スペースで区切ったリスト。これらのファイル名のいずれかに一致するファイルのみがサービスされます。使用できるワイルドカードは、「?」（任意の 1 文字に一致）と「*」（ゼロ個以上の文字に一致）です。

list-of-filenames

プロパティ：

デフォルト： 全ファイルへのサービスを許可

Fileshare

Fileshare コネクタは、各会話の開始時にこれらの設定をチェックします。設定の変更は、次の会話から有効になります。

[Operation]

timeout=number-of-seconds

[Trace]

control=control-flow-trace

data=data-trace

file=pathname

timeout

クライアントからデータを受信する場合のタイムアウトを秒数で設定します。

構文：

```
timeout=number-of-seconds
```

プロパティ：

デフォルト： 通信プロセスのタイムアウト。7200 秒 (2 時間) です。

control

コネクタの制御フローのトレースを有効にします。

構文：

```
control=control-flow-trace
```

パラメータ :

control-flow-trace さまざまなコネクタの動作 (エンタープライズサーバに要求を渡すなど) をトレースする場合は「yes」に設定します。

プロパティ :

デフォルト : 制御フローのトレースは無効です。

data

送信および受信したデータのトレースを有効にします。

構文 :

data=data-trace

パラメータ :

data-trace データをトレースする場合は「yes」に設定します。

プロパティ :

デフォルト : データのトレースは無効です。

file

トレースメッセージを書き込むファイルの完全パスを指定します。

構文 :

file=pathname

プロパティ :

デフォルト : トレースメッセージは MFCS ログに書き込まれます (HTML 形式で)。

TN3270

TN3270 コネクタは、以下の設定を定期的にチェックします。設定の変更は、現在の会話および新しい会話に影響を及ぼします。

[logging]

client-close=client-close-logging

[Operation]

default-terminal-type=terminal-type-name

```
Timeout=number-of-seconds  
TN3270E=negotiation-type  
[Trace]  
trace=control-flow-trace  
data=data-trace  
length=trace-length  
sem=send-semaphore-trace  
file=trace-filename
```

client-close

クライアントの接続解除のログ処理を制御します。

パラメータ：

client-close-logging 「yes」に設定すると、クライアントの正常な接続解除が記録されます。

プロパティ：

デフォルト： クライアントの異常な接続解除のみを記録します。

default-terminal-type

クライアントから送信された正しくないターミナルタイプを無効にします。

パラメータ：

terminal-type-name 「ibm-3278-2」などの、有効なターミナルタイプ名。

プロパティ：

デフォルト： なし

備考：

TN3270 の会話ネゴシエーションでは、エミュレートしているターミナルタイプをサーバに通知します。TN3270E は、許可されているターミナルタイプのセットを指定しますが、Classic TN3270 では要件はそれほど厳しくありません。TN3270E クライアントの中には、仕様に準拠していないものもあります。コネクタは、不正なターミナルタイプを受信した場合、通常はクライアントに対して別のタイプを指定するよう求めます。クライアントの中には (NetManage ViewNow など) 正しく応答しないものがあります。このような場合は、次の設定を行うことをお奨めします。

Timeout

受信タイムアウトを秒数で指定します。

構文：

```
Timeout=number-of-seconds
```

プロパティ :

デフォルト : 84600 秒 (1 日)

TN3270E

TN3270E (拡張 TN3270) ネゴシエーションを無効にします。

構文 :

TN3270E=negotiation-type

パラメータ :

negotiation-type 「no」に設定すると、TN3270E ネゴシエーションが無効になります。

プロパティ :

デフォルト : 拡張 TN3270 ネゴシエーションは有効です。

備考 :

この設定は、Classic TN3270 のみをサポートし、TN3270E ネゴシエーションを正しく行わないような、非常に特殊なクライアントの場合に役に立つことがあります (現在、そのようなクライアントは知られていません)。

trace

コネクタ内の制御フローのトレースを有効にします。

構文 :

trace=control-flow-trace

パラメータ :

control-flow-trace 「yes」に設定すると、制御フローのトレースが有効になります。

プロパティ :

デフォルト : トレースは無効です。

data

データのトレースを有効にします。

構文：

data=data-trace

パラメータ：

次の値が指定できます。

yes

または
すべてのデータフローをトレースします。

data-trace all

nvt NVT モードのみのデータフローをトレースします (基本的にはネゴシエーション中のみ)。

その他 データのトレースは無効です。

プロパティ：

デフォルト：
トレースは無効です。

length

データトレースに書き込むデータの最大バイト数を指定します。

構文：

length=trace-length

プロパティ：

デフォルト：
40 バイト

sem

送信セマフォでのオペレーションのトレースを有効にします。

構文：

sem=send-semaphore-trace

パラメータ：

send-semaphore-trace 「yes」に設定すると、送信セマフォでのオペレーションがトレースされます。

プロパティ：

デフォルト: トレースは無効です。

備考:

この設定は、送信セマフォに問題があると疑われる場合に役に立つことがあります。会話がハングしても、ESMAC がビジー表示をしないためです。

file

トレースメッセージを書き込むファイルの完全パスを指定します。 ファイルは HTML 形式です (Fileshare や ISC トレースの場合とは異なります)。

構文:

file=trace-filename

プロパティ:

デフォルト: トレースメッセージは MFCS ログに書き込まれます (HTML 形式で)。

MTO ISC

ISC コネクタは、各会話の開始時に [Operation] セクションの設定をチェックします。 設定の変更は、次の会話から有効になります。

ISC コネクタは、各会話の際に以下の [Trace] セクションを定期的にチェックするため、新しい会話と既存の会話の両方で設定の変更が有効になります。

[Operation]

timeout=number-of-seconds

[Trace]

trace=control-flow-trace

data-trace=data-trace-setting

allmax=dump-size

file=pathname

timeout

クライアントからデータを受信する場合のタイムアウトを秒数で設定します。

構文:

timeout=number-of-seconds

プロパティ:

デフォルト: 通信プロセスのタイムアウト。7200 秒 (2 時間) です。

trace

コネクタ内の制御フローのトレースを有効にします。

構文：

trace=control-flow-trace

パラメータ：

control-flow-trace さまざまなコネクタの動作 (エンタープライズサーバに要求を渡すなど) をトレースする場合は「yes」に設定します。

プロパティ：

デフォルト： トレースは無効です。

data-trace

送信および受信したデータのトレースを有効にします。

構文：

data-trace=data-trace-setting

パラメータ：

次の値が指定できます。

no データフローをトレースしません。

header CTG プロトコルヘッダーのフォーマットされたトレースを作成します。

fmh

data-trace-setting または CTG ヘッダー、要求ヘッダー、応答ヘッダーのフォーマットされたトレースを作成します。

structure

all ユーザデータのヘッダーおよびダンプのフォーマットされたトレースを作成します。

dump 全データのフォーマットしていないダンプを作成します。

プロパティ：

デフォルト： トレースは無効です。

allmax

data-trace-setting が「all」の場合、ユーザデータのダンプに書き込むデータの最大バイト数を設定します。

構文：

allmax=dump-size

パラメータ：

dump-size データをすべて書き込む場合は、0 (無制限) に設定します。

プロパティ：

デフォルト： 無制限。つまり、データをすべて書き込みます。

file

トレースメッセージを書き込むファイルの完全パスを指定します。 ファイルはテキストファイルです。

構文：

file=pathname

プロパティ：

デフォルト： トレースメッセージは MFCS ログに書き込まれます (HTML 形式で)。

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。

- 通信プロセスの属性の編集
- 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
- 通信プロセスの統計情報の表示

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

- 通信プロセスのコピー
- 通信プロセスの削除。 エンタープライズサーバに通信プロセスが 1 つしかない場合は、削除できません。
- 通信プロセスの開始
- 通信プロセスの停止
- 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)

新しく作成された通信プロセスは、「コピー」ページで「自動開始」をチェックしている場合はただちに開始されます。チェックしていない場合は、「通信プロセスとリスナー」ページでプロセスの [開始] ボタンをクリックして開始します。「自動開始」が有効になっている通信プロセスは、エンタープライズサーバの起動時に自動的に開始されます。

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

Copyright © 2006 Micro Focus (IP) Ltd. All rights reserved.

◀ サーバ

サービス、パッケージ、および要求ハンドラ ▶

第 9 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。

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連携して動作することで、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。システムサービスのサービスオブジェクトとは対照的に、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 1 つの実装パッケージが関連付けられている必要があります。サービスリスナーは、サービスオブジェクトに 1 つ以上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

サービスの手動作成

Interface Mapping Toolkit を使用して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、および要求ハンドラと関連付ける

追加、更新、および削除

エンタープライズサーバが実行中かどうかに関係なく、エンタープライズサーバにサービスと実装パッケージを追加し、さらにそれら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 (つまり、状態が「開始」の場合) は、サービスとパッケージを追加すると、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限り、更新も受け付けられてその変更がすぐに反映されます。

クライアント要求によりリソースが使用されている場合は、サービス、パッケージ、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。

サービス、パッケージ、および要求ハンドラを削除すると、物理ファイルではなく Directory Server リポジトリ内のオブジェクトが削除されます。開発者が、デプロイツールまたは `imtkmake` コマンドを使用してサービスをデプロイすると、そのたびに新しいディレクトリがエンタープライズサーバのデプロイディレクトリに作成され、デプロイされたファイルがすべて格納されます。開発者が、すでにデプロイされているサービスを再度デプロイすると、古いデプロイディレクトリは参照されなくなります。これは、古いサービスとパッケージオブジェクトがすでに削除されているためです。古いデプロイディレクトリは、時々削除してください。

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す。

Enterprise Server が拒否する更新を Directory Server が受け付けた場合は、使用中のエンタープライズサーバと 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。

サービス

サービスにより、特定のビジネス機能にアクセスすることができます。

Interface Mapping Toolkit を使用して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、『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) に応じて、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を含めることができます。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。デフォルトの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 つの要素で構成される名前。単純サービスの名前はこの形式です。
- `http://tempuri.org/wmapserv#Add` など、3 つの要素で構成される名前。名前を構成する 3 つの要素は次のとおりです。
 - サービスのネームスペース。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。

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 9-1 に、サービステーブルを示します。

Server ESDEMO										
Server...		Listeners (3)		Services (6)		Handlers (2)		Packages (4)		
Service Display Filters		Service:		Class: All		Handler: All		Package: All		
1 - 3 of 3 displayable namespaces from a total of 3								Show 10 service namespaces at a time		
Service Namespace	Operation	Service Class	Search Order	Listener(s)	Request Handler	Implementation Package	Current Status	Status Log	Custom Configuration	
Test	Test Edit...		1	1 CI 1 HTTP Echo tcp:10.120.63.180*:9002 (NWB-MBISSETXP.microfocus.com)			Available	OK		
Deployer	Deployer Edit...	MF deployment	1	1 CI 1 Web tcp:10.120.63.180*:2808* (NWB-MBISSETXP.microfocus.com)			Available	OK	[MF client] scheme=http URL=/cgi/mfdeploy.exe/uploads accept=application/x-zip-compressed	
Delete	#Add Edit...		1	1 CI 1 Web Services tcp:10.120.63.180*:9003 (NWB-MBISSETXP.microfocus.com)	MFRHSOAP	http://tempuri.org/wmapserv#Add	Available	OK		
	#Next Edit...		1	1 CI 1 Web Services tcp:10.120.63.180*:9003 (NWB-MBISSETXP.microfocus.com)	MFRHSOAP	http://tempuri.org/wmapserv#Next	Available	OK		
	#Read Edit...		1	1 CI 1 Web Services tcp:10.120.63.180*:9003 (NWB-MBISSETXP.microfocus.com)	MFRHSOAP	http://tempuri.org/wmapserv#Read	Available	OK		
	#Delete Edit...		1	1 CI 1 Web Services tcp:10.120.63.180*:9003 (NWB-MBISSETXP.microfocus.com)	MFRHSOAP	http://tempuri.org/wmapserv#Delete	Available	OK		
Add...										

図 9-1: 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バ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。

サービスクラス: 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。

探索順序: 検索のために並べ替える場合に、同じ名前のサービスを並べる順序です。

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が表示されます。

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: 詳細は、『[構成情報](#)』の節を参照してください。

説明

サービス表示フィルタは、表示するサービス情報を制御します。サービス情報は、サービステーブルの一番上に表示されます。サービス表示フィルタには、次の 4 種類があります。

サービス: このフィルタを使用するには、表示するサービス名を指定します。入力する文字は、サービスを一意に識別できる文字だけで十分です。

クラス

ハンドラ

パッケージ

サービス以外のフィルタには、オプションが 3 つあります。

なし: このフィルタではサービスを表示しない。

一部: このフィルタでは値をもつサービスを表示する。

すべて: このフィルタではすべてのサービスを表示する。

たとえば、クラス 1 のサービスに関してサービスの詳細を表示する場合は、「サービス表示フィルタ、クラス」で「一部」を選択します。関連する要求ハンドラをもたないサービスのみを表示する場合は、「サービス表示フィルタ、ハンド

virtual-directory-name-1/ mfdeploy.exe を含むディレクトリにマップされた、リスナーの構成内にある仮想ディレクトリに対応していなければなりません。

virtual-directory-name-2 新しいディプロイディレクトリが作成される仮想ディレクトリに対応していなければなりません (このディレクトリには、.mfdeploy ファイルが必要です)。

プロパティ :

デフォルト: /cgi/mfdeploy.exe/
uploads

備考 :

単一の Web リスナーの下に複数のディプロイサービスを置き、ディプロイパッケージに対して別の位置を指示することができます。この場合、リスナーの構成に仮想ディレクトリを追加し、適切なディレクトリをここに指定します。詳細は、『構成』の章の『[ディプロイサービスとディプロイリスナー](#)』の節を参照してください。

accept

このサービスが受信する要求データの種類を MFCC に通知します。

構文 :

accept=request-data-type

パラメータ :

request-data-type 「application/x-zip-compressed」に設定します。

プロパティ :

デフォルト: な
し

listener

新しいサービスを所有するリスナーの名前を指定します。デフォルトは「Web Services」です。サービスを別の名前のリスナーにディプロイしたい場合は、この値を設定します。

構文 :

listener=listener-name

プロパティ :

デフォルト: Web
ト: Services

備考 :

サービスを別の名前のリスナーにディプロイしたい場合は、この値を設定します。

server

新しいサービスとパッケージを所有するサーバの名前を指定します。デフォルトでは、ディプロイサービスが属するサーバになります。この値を設定して、同じシステムの異なるエンタープライズサーバにディプロイされるサービスへのディプロイ要求を、エンタープライズサーバで受信することができます。

構文：

server=server-name

プロパティ：

デフォルト： ディプロイサービスが属するサーバ

備考：

この値を設定して、同じシステムの異なるエンタープライズサーバにディプロイされるサービスへのディプロイ要求を、エンタープライズサーバで受信することができます。

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

サービスのログ処理のオンとオフの切り替え。 ログ処理がオンの場合、サービスの要求および結果ブロックは、エンタープライズサーバのダンプデータセットに書き込まれます。

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

単純サービスまたはオペレーションの削除

オペレーションをもつサービスの削除。 オペレーションをもつ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](#)

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。

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 9-2 に、パッケージテーブルを示します。

Server ESDEMO									
Server...		Listeners (3)		Services (6)		Handlers (2)		Packages (4)	
1 - 4 of 4 packages								Show <input type="text" value="10"/> packages at a time	
	Name	Current Status	Status Log	Package Module	IDT	Package Path	Custom Configuration		
Edit...	http://tempuri.org/wmapserv#Add	Available	OK		E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X\wmapserv.idt	E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X			
Edit...	http://tempuri.org/wmapserv#Next	Available	OK		E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X\wmapserv.idt	E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X			
Edit...	http://tempuri.org/wmapserv#Read	Available	OK		E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X\wmapserv.idt	E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X			
Edit...	http://tempuri.org/wmapserv#Delete	Available	OK		E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X\wmapserv.idt	E:\Program Files\Microsoft Focus\Net Express\Base\deploy\wmapserv.3mETH12X			
Add...									

図 9-2: 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

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: このフィールドは使用しません。

説明

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

パッケージをサービスに関連付ける

パッケージのサービスとの関連付けを解除する

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。

エンタープライズサーバへのパッケージの追加

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)

Enterprise Server Administration の外部で、古いディプロイディレクトリを削除できます。

方法

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読解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: Microsoft 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。

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 9-3 に、要求ハンドラテーブルを示します。

Enterprise Server Administration > ESDEMO > Handlers
Version 1.03.83
NWB-MBISSETXP.microfocus.com (10.120.63.180)

Status MDS00001 OK [SCHEMA...]

Server ESDEMO

Server... Listeners (3) Services (2) **Handlers (2)** Packages (0) Refresh

	Module Name	Module Path	Current Status	Status Log	Custom Configuration	Description
Edit...	MFRHBINP		Enabled	OK		Handles J2EE requests
Edit...	MFRHSOAP		Enabled	OK		Handle SOAP requests
Add...						

図 9-3: 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

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: このフィールドは使用しません。

説明

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

要求ハンドラのサービスとの関連付け

要求ハンドラのサービスとの関連付け解除

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。

エンタープライズサーバへの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 10 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。

監視機能

Micro Focus Server Administration は、Directory Server のアクティビティを監視するために、次の 4 つの機能を備えています。

- ディレクトリページ
- 統計ページ
- セッションリスト
- ジャーナル

ディレクトリページ

ディレクトリページでは、ローカルネットワーク内の Directory Server のその他すべてのインスタンスを表示するテーブルで、各インスタンスについて次の情報を表示します。

- Directory Server のバージョン番号
- 実行しているマシンの IP アドレス
- 実行しているマシンの DNS 解決されたホスト名。ホスト名の横にある右向きの矢印をクリックすると、Enterprise Server Administration のウィンドウが新しく開き、IP アドレスを覚えなくても、他の Directory Server およびそのオブジェクトにアクセスできるようになります。

注：ローカルネットワーク上の Directory Server の全インスタンスに対して、適切なレベルのセキュリティを設定し、ディレクトリページを使用した不正なアクセスが行われる可能性を少なくしてください。

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 5 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

Directory Server が使用するファイルパス

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、サービスをディプロイしたにもかかわらず、ディ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が変更されたかどうかをこのページで確認できます。

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。

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ブルフィックス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き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。

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。ジャーナルをエクスポートして、ジャーナルを検索したり、ジャーナルに注釈を付けたりすることができます。

Schema Administrator アクセス権をもつ場合は、次の作業ができます。

ディレクトリページの表示

[方法](#)

Copyright © 2006 Micro Focus (IP) Ltd. All rights reserved.

◀ サービス、パッケージ、および要求ハンドラ

ESMAC 使用によるサーバの管理 ▶

第 11 章 : ESMAC 使用によるサーバの管理

ここでは、Enterprise Server Monitor and Control を使用して、エンタープライズサーバを管理する方法を説明します。

概要

Enterprise Server Monitor and Control (ESMAC) は、エンタープライズサーバ上で実行するサービス、パッケージ、およびハンドラの表示および管理ができます。

ESMACは、Enterprise Server Administration と連動しています。ただし、ESMAC には Enterprise Server Administration にはない機能があります。ESMAC では、次の作業ができます。

- サービス実行プロセス、サービス、パッケージなど、エンタープライズサーバを監視および管理します。
- サーバとアプリケーションの現在のアクティビティとパフォーマンスを監視します。また、アクティビティとパフォーマンスの傾向を監視します。
- アプリケーションの診断情報 (ダンプとトレース) を取得します。

ESMAC は、ESMAC の使用者およびリソースの表示と更新を行うアクセス権を定義できます。ESMAC ユーザの詳細は、『[セキュリティ](#)』の節を参照してください。

実際の画面

ESMAC インターフェイスは、ブラウザを使用して表示します。インターネットやイントラネットでエンタープライズサーバを監視および管理するには、次に示すブラウザのどれかをインストールする必要があります。

- Internet Explorer V 6.0 以降
- Netscape V 6.0 以降
- Mozilla V1.3 以降

ESMAC を初めて表示すると、図 11-1 に示す「サーバ情報」ページが表示されます。

Home

Server

Monitor

Control

SEPs

Clients

Diagnostics

Log A

Dump B

Trace C

Blocks

Display

Resources

Services

Package

Handlers

Users

Env.Vars.

Chg.User

Help:

[Menu](#)

[This Page](#)

[Contents](#)

Server Information Refresh

Times:	Start: 9/29/2003 - 15:51:00
	Elapsed: 0 - 0:01:32
Counts:	Transactions: 6 Dumps: 0 Per Hour: 234 Trace Blocks: 0 Active Clients: 0 Limit: 0
Sizes:	Max Tasks: 2 HTTP Out: 63 k segments Diagnostics Size: 0 k ID Timeout: 0 (minutes) Shared Memory: 256 pages (4k) SM Total: 1024 k SM Cushion: 131 k SM Free:
States:	Force Phase In: Yes Dump: A Dump on ABEND: Sys:Yes Tran:No Trace: A Active trace:
Activity:	Avg. Latency(ms): Avg. Task(ms): Task/Sec:
	15:51

Server Information Refresh

図 11-1 : ESMAC の「サーバ情報」ページ

ESMAC を使用すると、次のことが可能になります。

現在、接続しているサーバの情報を取得します。

サーバ上のサービスを監視します。

サービス実行プロセス、アクティブな診断データセット、トレースコントロールフラグセットなどのサーバに関連付けられている各パラメータを表示および変更します。

サービス実行プロセス (service execution processe; SEP) 情報の表示、SEP の停止、およびトレース情報の取得を行います。

サーバ上で動作するクライアントプロセスの情報を表示します。

コンソールログ、ダンプ、およびトレース情報を表示します。

サーバ上に定義されているサービス、パッケージ、および要求ハンドラの情報を表示します。

サーバ管理者のアクセス権を削除および変更できます。

サーバに関連付けられている環境変数を表示します。

管理者としてログインします。

ESMAC インターフェイスは、次の 2 つの領域に分割されています。

メニュー。これは、ページの左側サイドバーにあり、ESMAC を管理する複数のボタンがあります。ページのメイン本体。これは、監視および管理情報を表示します。これらの情報とコントロールは、メニューでボタンをクリックすることで利用可能になります。

サイドバー

ESMAC メニューは、ページの左側サイドバーに表示されます。ユーザの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。

本体

管理および監視情報は、ページ本体に表示されます。

ページ本体は、メニューでボタンをクリックすると変更されます。通常、ページ本体は、1 つ以上のコントロールがあり、エンタープライズサーバで必要な情報を入力または変更できます。

ステータス行

ステータス行は、ページの一番上に表示されます。ステータス行は、次の内容が表示されます。

- 接続しているエンタープライズサーバ名。
- エンタープライズサーバが実行しているサーバ名。
- ユーザ名。
- 現在、表示されているページの日時。

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、サーバを追加した直後に「サーバ情報」ページの [画面更新] をクリックすると、アクティビティの詳細が更新されます。

セキュリティ

ESMAC にアクセスするユーザを設定した場合は、設定直後にそのユーザのアクセス権を設定する必要があります。

デフォルトのセキュリティは、システムの設定によって異なります。すでに定義されている特別なユーザ `mfuser` をシステムに設定することも設定しないこともできます。

- `mfuser` が定義されていない場合は、すべてのユーザはデフォルトですべてのアクションを実行できます。
- `mfuser` が定義されている場合は、アクセス権はデフォルトで `mfuser` 用に定義されているアクセス権と同じです。

エンタープライズサーバを開発に使用する場合は、`mfuser` を設定しないことをお奨めします。エンタープライズサーバを運用に使用する場合は、`mfuser` を設定することをお奨めします。`mfuser` には、最小の権限のみを与えます。

ユーザ名およびパスワードを作成し、「ユーザ」ページでアクセス権を設定します。

[方法](#)

[ユーザ] をクリックすると、管理者は現在の設定を表示できます。すべてのユーザのリストが「ユーザ」ページに表示されます。「ユーザ」ページで [詳細] をクリックすると、ユーザのアクセス権の詳細な情報が表示されます。

読み取りや更新のアクセス権がない場合は、使用可能なオプションが ESMAC で制限されています。「ユーザ」ページでは、次の操作を行えます。

- サーバ情報の表示。詳細は、『[サーバ情報](#)』の節を参照してください。
- パフォーマンスの監視。詳細は、『[サービスの監視](#)』の節を参照してください。
- 環境変数の表示。詳細は、『[環境変数](#)』の節を参照してください。
- 異なるユーザに変更します。

サーバへの接続

監視するサーバに接続するために、Enterprise Server Administration から ESMAC を開始します。

方法

ESMAC メニューの [ホーム] をクリックすると、Enterprise Server Administration に戻ることができます。

サーバ情報

ESMAC は、現在表示しているエンタープライズサーバのさまざまな情報を提供します。メニューで [サーバ] をクリックすると、サーバ情報が「サーバ情報」ページに表示されます。

「サーバ情報」ページは、エンタープライズサーバの操作を監視できる情報を提供します。詳細は、次のとおりです。

- 日時：エンタープライズサーバの開始日時、経過日時など。
- カウント：サービス数、ダンプ数、トレースブロック数など。
- サイズ：サーバ上の最大タスク数、使用可能なメモリ量、診断ファイルサイズなど。
- 状態：ダンプ状態、現在のダンプデータセット設定など。
- アクティビティ：タスク完了の平均時間。

これらの情報は、サーバの状態をすばやく測定できます。たとえば、次のような場合があります。

- 現在、実行しているサービスを知りたい場合があります。「アクティビティ」情報で、タスクの完了が予期している時間より長い場合など、サービスが予想外の動作をしているかどうかわかります。
- ダンプを取得している環境を知りたい場合があります。「状態」情報で、サービスアベンドではなくシステムアベンドでダンプを取得しているなど、現在のダンプ状態がわかります。

「サーバ情報」ページで表示されている情報によっては、エンタープライズサーバへの変更が必要な場合があります。この場合は、他の ESMAC ページを使用して変更できます。これらの変更については、以後で説明します。

サービスの監視

エンタープライズサーバ上でサービスを実行する場合は、サービスが期待通り開始および停止しており、予想より多くのリソースを使用していないことを確認する必要があります。

「監視」ページは、サービスのパフォーマンスを表示します。詳細は、次のとおりです。

潜在平均時間 (ミリ秒)。
タスクの完了の平均時間 (ミリ秒)。
1 秒あたりの完了タスク数。

たとえば、タスクの完了の平均時間が長い場合は、サービスが長期実行しているか、ステートフルであることを示しています。1 秒あたりの完了タスク数が不意に予期していた数より上昇した場合は、タスクのエラーにより、他のタスクが繰り返し呼び出されている可能性があります。

方法

エンタープライズサーバの管理

ESMAC を使用してエンタープライズサーバの各パラメータを管理できます。ESMAC メニューの [管理] をクリックすると、「管理」ページが表示され、各パラメータの管理ができます。

たとえば、『[サーバ情報](#)』の節で説明されているような、ダンプの状態を含めてエンタープライズサーバのさまざまな情報を表示できます。タンプの状態を変更する場合は、「管理」ページで行います。

「管理」ページで管理する内容は、次のとおりです。

サーバ上で実行している SEP 数。
ダンプとトレースパラメータは、補助トレースとダンプファイルのサイズおよびダンプとトレースを使用するデータセットなどを含んでいます。
プログラムが起動されるたびにリロードするかどうか。
トレースコントロールフラグセット。
HTTP 出力用に共有メモリを通じて渡されるデータブロックのサイズ。
タイムアウト値は、エンタープライズサーバのユーザ ID 用に設定されたタイムアウト値を保有していないすべてのユーザに使用されます。

また、このページからエンタープライズサーバをシャットダウンすることもできます。シャットダウン時には、サーバのダンプが得られます。

方法

サービス実行プロセス

サービス実行プロセスは、エンタープライズサーバ内にあります。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。サービス実行プロセスの詳細は、『Micro Focus Enterprise Server 構成と管理』の『構成』の章にある『[サービス実行プロセス](#)』の節を参照してください。

「SEP」(サービス実行プロセス) ページを表示すると、エンタープライズサーバ上で実行している SEP の情報を表示できます。情報を表示するには、メニューの [SEP] をクリックします。

「SEP」(サービス実行プロセス) ページは、次の SEP の情報も含め、現在のエンタープライズサーバ上のすべての SEP 詳細を表示します。

プロセス ID。
SEP の種類。
タスク数。
該当する SEP で実行するサービス数。

SEP の終了および SEP 用のトレースの作成と表示もできます。

方法

サービスとの対話操作

エンタープライズサーバでサービスとして実行するプログラムでは、ACCEPT FROM CONSOLE 文および DISPLAY UPON CONSOLE 文が使用できます。DISPLAY UPON CONSOLE 文が実行されると、表示テキストがコンソールログに表示され、サービスは継続します。ACCEPT FROM CONSOLE 文が実行されると、サービスはいったん停止し、ESMAC のページを使用してユーザに入力を求めます。サービスの実行を継続するには、情報を入力する必要があります。

方法

問題の診断

ESMAC は、ダンプとトレース情報を作成および表示し、サーバコンソールログを確認できます。

サーバコンソールログ

サーバコンソールログは、領域の情報を提供します。このコンソールログには、サーバの開始や停止時の詳細、SEP の作成情報などが含まれます。

メニューの診断グループの「ログ」をクリックし、[表示] をクリックすると、コンソールログが表示されます。

ESMAC に戻るには、ブラウザの [戻る] をクリックします。

ダンプとトレース

システムダンプまたはトランザクションダンプを指定することができ、システムトレースを作成できます。これらは、「管理」ページで行います。「管理」ページは、ダンプとトレースを格納するデータセットも指定できます。ダンプとトレースを表示するには、メニューの診断グループのオプションを使用します。

方法

「SEP」ページと「クライアント」ページのオプションを使用して、SEP とクライアントプロセスのトレースも取得できます。

方法

システムダンプとトレースの詳細は、『トラブルシューティング』の章にある『[診断サーバの使用方法](#)』の節を参照してください。

サービス、パッケージ、および要求ハンドラ

サービス、パッケージ、および要求ハンドラの情報は、リソースグループで行います。

[サービス] をクリックすると、「サービス」ページが表示され、エンタープライズサーバ上で実行している各サービスのリストが表示されます。「サービス」ページでサービスの 1 つを表示し、[詳細] をクリックすると、そのサービスの詳細な情報が表示されます。「サービス」ページは、選択したサービスの情報を表示します。[パッケージステータス]、[詳細] の順にクリック、および [ハンドラステータス]、[詳細] の順にクリックす

ると、それぞれ関連するパッケージと要求ハンドラの情報が表示されます。パッケージと要求ハンドラ情報へのリンクは、そのサービスに影響する問題として提供されます。

必要に応じて、[パッケージ] と [ハンドラ] をクリックすると、それぞれパッケージと要求ハンドラを個別に表示できます。

新しいサービス、パッケージ、要求ハンドラを設定するには、Enterprise Server Administration を使用します。詳細は、『Micro Focus Enterprise Server 構成と管理』にある『[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。

ダンプデータセットに記録されたサービスへの入力と出力を指定できます。デフォルトでは、指定できません。

方法

環境変数

各環境変数は、エンタープライズサーバの開始時に設定されます。これらの環境変数のエラーは、サーバ上のアプリケーションの実行に影響を与えます。

ESMAC は、エンタープライズサーバ用に設定されたすべての環境変数の完全なリストを表示できます。表示するには、メニューの [環境変数] をクリックします。「環境領域」ページに表示されます。

Enterprise Server Administration を使用して、エンタープライズサーバ用の環境変数を設定できます。『Micro Focus Enterprise Server 構成と管理』の『構成』の章にある『[環境変数](#)』の節を参照してください。

第 12 章：トラブルシューティング

ここでは、Enterprise Server で実行しているアプリケーションが引き起こす一般的な問題の診断の方法を説明し、それを解決するアドバイスを提供します。

情報の収集

すべてのトラブルシューティングにおいて、問題が発生したときのエンタープライズサーバの状態および問題に至るイベントについて、できるだけ多くの情報の収集が必要です。

診断補助として、次の情報を使用できます。

- ジャーナルと呼ばれる Directory Server ログファイル
- エンタープライズサーバログファイル
- 通信プロセス (MFCS) ログファイル
- さまざまなダンプとトレース

問題が発生した場合は、現在のログファイル、ダンプ、およびトレースのみでなく、各ファイル、ディレクトリ、領域、および複数のオペレーティングシステムツールの出力結果もキャプチャする必要があります。必要なときにデータを容易に取得するために、障害に先立ってデータキャプチャのメカニズムを用意しておくことが重要です。実稼動システムでは、まずエンタープライズサーバのサービスをクライアントに復元してから、キャプチャしたデータを解析することをお奨めします。サポートするオペレーティングシステムをリブートし、エンタープライズサーバが開始した直後に、正常な状態のエンタープライズサーバ上でデータキャプチャを行うこともお奨めします。障害が発生したデータのキャプチャと比較するために、この正常なデータのキャプチャを保持する必要があります。

エンタープライズサーバがクライアント要求に応答しているかどうか、および ESMAC からの要求に応答しているのかも確認する必要があります。応答がない場合、通信プロセス (MFCS) の問題であることを示します。

エンタープライズサーバログファイル

エンタープライズサーバはメッセージをコンソールログファイルに記録します。通常、コンソールログファイルは C:\Documents and Settings\user-ID\My Documents\Micro Focus\Net Express 5.0\WORKAREA\es-name\console.log にあります (user-ID は、Enterprise Server を実行しているユーザアカウント)。エンタープライズサーバが開始されると、既存の console.log ファイルは console.bak に名前が変わり、既存の console.bak ファイルは消去されます。データ収集には、console.log と console.bak の両方が必要です。エンタープライズサーバを再開する前に、ログファイルを安全な場所にコピーしてください。

[方法](#)

Directory Server ジャーナル

Directory Server は、イベントをジャーナルに記録します。通常、ジャーナルは C:\Documents and Settings\All Users\Application Data\Micro Focus\NetExpress\5.0\journal.dat にあります。最新のエントリーは、Enterprise Server Administration の Web インターフェイスのページ上でも使用できます。

最大ファイルサイズ、表示されるエントリー数、およびログレベルは、Enterprise Server Administration の「構成オプション」ページで変更できます。ジャーナルが最大ファイルサイズ設定値に達した場合は、最新のジャーナルエントリーにより古いジャーナルエントリーが上書きされます。トラブルシューティングを行うために、最大ファイルサイズを 256 KB 以上に増やす必要がある場合があります。

ログ処理には、次の 3 つのレベルがあります。

エラーのみ
 エラーと警告
 すべての情報

トラブルシューティングを行うために、最大のログレベルに設定する必要がある場合があります。

方法

通信プロセスログファイル

MFCS (Communications processes; 通信プロセス) は、メッセージを通信プロセスログファイルに記録します。通常、ログファイルは C:\Documents and Settings\user-ID\My Documents\Micro Focus\Net Express 5.0\WORKAREA\es-name\log.html にあります (user-ID は、Enterprise Server を実行しているユーザアカウント)。複数の通信プロセスが実行されている場合は、各メッセージの先頭の日時に、[2] のように角かっこでインスタンス識別子が続きます。

デフォルトでは、管理者によって削除されるまで連続的に増大するログファイルは 1 つのみです。ただし、mf-server.dat ファイルを循環ログファイルとして使用し、ログレベルを変更できます。通常、mf-server.dat ファイルは、install-dir\base\bin (Windows の場合) または \$COBDIR/etc (UNIX の場合) にあります。ファイルは ini ファイル形式で、セクションタグが角かっこで囲まれ、その後「name=value」の組が続きます。

循環ログファイルの場合は、MFCS が log-1.html、log-2.html などのような名前の複数のログファイルを使用します。ログファイルが設定サイズに達すると、MFCS は次のファイルに移ります (既存の場合は最初に削除されます)。ログファイルが設定数に達すると、log-1.html に戻ります。これにより、ログサイズの合計が、特定のサイズを超過しないようになります。

構文は、次のとおりです。

```
[logging]
files=number-of-files
filesize=maxsize
dscontrol=none|standard|all
dirsvc=none|unusual|processing|not-found|all
```

すべてのパラメータは、オプションです。ただし、循環ログファイルを有効にするには、「files=」と「filesize=」の両方を設定する必要があります。また、「files=」は、2 以上の数を設定する必要があります。これらのパラメータは、上記で説明しているように循環ログファイルを構成します。両方とも設定すると、MFCS で必要になるログファイル領域の合計は「files*filesize (概算値)」を超えません。パラメータを、次に示します。

files

ログファイル数を指定します。

構文：

```
files=number-of-files
```

filesize

ログファイルの最大サイズをキロバイト (KB) で指定します。

構文：

```
filesize=maxsize
```

dscontrol

管理メッセージ用にログレベルを設定します。通常、これらのメッセージは、Directory Server (MFDS) から発行されますが、他のソースから発行されることもあります。メッセージは、構成を更新するなどの特定のアクションを行うように MFCS に通知します。

構文：

```
dscontrol=none|standard|all
```

パラメータ：

none	管理メッセージを記録しません。
standard	MFCS 構成または状態を更新する管理メッセージを記録します。エンタープライズサーバがまだ動作中かどうかを確認するために MFDS が使用する KEEPALIVE プロブメッセージなどの、情報を要求するのみのメッセージは記録しません。
all	すべての管理メッセージを記録します。

プロパティ：

デフォルト： standard

dirsvc

Directory Server リポジトリをクエリーまたは更新する MFCS の試行用のログレベルを設定します。構成の問題を診断する場合に、このレベルを設定すると役立つことがあります。

構文：

```
dirsvc=none|unusual|processing|not-found|all
```

パラメータ：

none	Directory Server アクティビティの情報を記録しません。
unusual	構成の問題を示す重複するオブジェクト名などの異常な結果のみ記録します。
processing	正常なプロセスの進捗についてのメッセージも記録します。これは、デフォルト設定です。
not-found	「not-found」状態を返す検索についてのメッセージも記録します。通常、不要なオプションの構成項目が見つからないことを MFCS が予期していることから、これらのメッセージは記録されません。場合によっては、すべての「not-found」結果を記録することが構成の診断に役立ちます。
all	正常な MLDAP アンバインドオペレーションを含む、より多くの情報を記録します。

プロパティ：

デフォルト： processing

ダンプとトレース

運用システムは、常にいくつかの診断スイッチをオンにして動作させることをお奨めします。これは、予期しない問題を診断する場合に必要です。診断はリソースを消費するため、問題を特定する機能を取るかパフォーマンスを取るかが常に問題になります。ただし、問題を再現させたり、問題が再現するまで待って診断したりすることを避けるため、最初に問題が発生したときに診断できるようにすることが重要です。診断の最小レベルは、次のとおり

です。

- 「アプリケーション」のトレースフラグをオンにします。
- 「システムアベンドのダンプ」をオンにします。
- 「補助トレース」をオンにし、単一のトレースデータセット内に最低限 10 分間のデータを格納できるダンプデータセットサイズを選択します。トレースポイントやトランザクション量を増やす場合は、より多くの診断データセットサイズが必要です。
- 「ローカルトレーステーブルサイズ」を 70 以上に設定します。

「タスク制御」、「記憶制御」、および「アプリケーションコンテナ」のトレースフラグもオンにする必要がある場合があります。

「サーバの編集」ページ > 「診断」ページ、または ESMAC の「管理」ページ上でトレースと内部的に起動されるダンプの設定を指定します。次に示すように、ESMAC の「管理」ページ上でトレースフラグをオンにすることは、「サーバの編集」ページ > 「診断」ページでトレースフラグをオンにすることと同じです。

「サーバの編集」ページ > 「診断」ページ	ESMAC 「管理」ページ
「アプリケーション」	API
「タスク制御」	KCP
「記憶制御」	SCP
「アプリケーションコンテナ」	RTS

「サーバの編集」ページ > 「診断」ページでの設定は、ESMAC 「管理」ページでの設定より優先します。

方法

システムまたはトランザクションアベンド時に作成されるダンプは、内部的に起動されるダンプです。「管理」ページでこれらを選択している限り、アベンドが発生したときにダンプが作成されます。外部的に起動されるダンプも取得できます。それは、コマンドで直ちに回答してダンプを取得します。外部的に起動されるダンプの開始方法は、次に示すように複数あります。

- 「サーバの編集」ページ > 「診断」ページ > 「ダンプ」ページ上で [ダンプの作成] をクリックします。
- ESMAC の「管理」ページ上で [ダンプ] をクリックします。
- casdump コマンドを実行します。詳細は、ヘルプトピックの『[casdump](#)』を参照してください。

注：Directory Server がシステムサービスとして動作しており、Enterprise Server Administration インターフェイスからサーバを開始している場合は、casdump コマンドは動作しません。これは、サーバがユーザ ID で動作していないからです。casdump コマンドが動作するためには、次の作業のどれかを行います。

- Directory Server をユーザ ID で動作させます。『Enterprise Server Administration の概要』の章にある『[起動と停止](#)』の節を参照してください。
- システムサービスとして Directory Server を動作させます。ただし、casstart コマンドを使用してエンタープライズサーバを開始します (ヘルプトピックの『[casstart](#)』を参照してください)。
- システムサービスとして Directory Server を動作させます。ただし、LOCAL_SYSTEM ではなくユーザ ID を使用するようにシステムサービスの定義を変更します。これにより、エンタープライズサーバのコンソールウィンドウが表示されなくなります。ただし、Administration インターフェイスにはコンソールが表示されます。

casdump コマンドは、柔軟性があります。最初に、「/d」オプションなしで実行します。これは、「共有メモリをロックしない」ことを意味します。

```
casdump /res-name
```

このオプションなしで実行すると、記憶連鎖の追跡やダンプ中に、実行中のエンタープライズサーバにより共有メモリが変更されて、コマンドが失敗することがあります。しかし、「/d」オプションなしでコマンドを実行する価値はあります。それは、連鎖の追跡やブロックのフォーマットを開始する前に、まず最初に共有メモリをブロックとしてすべてダンプするからです。

コマンドがハングした場合には、「/d」オプションをつけて再度実行します。

```
casdump /res-name /d
```

エンタープライズサーバが共有メモリをロックしたままの状態で作動しなくなると、このコマンドはハングする可能性があります。

調査のために弊社にダンプを送付するには、「/f」オプション付きのコマンドを実行した結果も必要になる場合があります。このオプションは、共有メモリの使用について詳細な情報である FAQE (空き領域キュー要素) をダンプします。

外部的に起動されるダンプは、ダンプ X データセット casdumpx.rec に格納されます。「/d」オプションを指定する場合は、現在のダンプデータセット casdumpa.rec に格納され、指定しない場合は casdumpb.rec に格納されます。

データキャプチャ

データキャプチャの最小項目は、次のとおりです。

エンタープライズサーバディレクトリ (通常は C:\Documents and Settings\user-ID\My Documents\Micro Focus\Net Express 5.0\WORKAREA\es-name にあります。user-ID は、Enterprise Server を実行しているユーザアカウントです)。このディレクトリには、次のものが含まれます。

- コンソールログ。console.log と console.bak。
- 通信プロセスログファイル (1 つまたは複数)。『[通信プロセスログファイル](#)』の節を参照してください。
- 診断データセット。casauxta.rec と casauxtb.rec。
- システムアベンドダンプ (casdumpa.rec、casdumpb.rec、または casdumpx.rec のうち 1 つ)。
- 外部的に起動されるダンプ (システムアベンドダンプのようなファイル名)。

ファイル cobver のコピー (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\cobver、Enterprise Server をスタンドアロンでインストールした場合は install-dir\bin\cobver)。使用している製品の名前とバージョンが保持されています。

install-dir\base\bin\mf-server.dat が存在すれば、このファイルのコピー。

Directory Server 構成ディレクトリ。通常は C:\Documents and Settings\All Users\Application Data\Micro Focus\NetExpress\5.0\MFDS です。

Directory Server ログファイル (通常は C:\Documents and Settings\All Users\Application Data\Micro Focus\NetExpress\5.0\journal.dat)。『[Directory Server ジャーナル](#)』の節を参照してください。

Fileshare や FaultFind など、他の構成要素のログ (使用可能な場合)。

マシン上で実行しているプロセスのリスト。データキャプチャの先頭と末尾の 2 つのリストを取得する必要があります。これらは、Windows タスク マネージャから取得できます。

共有メモリ領域とセマフォのリスト。これらは、Process Explorer などのサードパーティーツールで取得できます。

netstat コマンドで取得したネットワーク接続とリスニングソケットのリスト。

オプション。開いているファイルのリスト。これらは、Process Explorer などのサードパーティーツールで取得できます。

注：

Process Explorer など、情報取得に使用できるオペレーティングシステムツールにより、さまざまなオプションが提供されます。弊社のサポート窓口から、最も便利なオプションについてのアドバイスを得ることができます。

これらの項目の多くは、エンタープライズサーバディレクトリに格納されています。このため、ディレクトリ全体の取り入れから開始することをお奨めします。

データキャプチャは、非破壊的です。つまり、正常なシステムでは破損しません。また、ノンブロッキングです。つまり、不健全なシステムで不完全な状態を待ち続けません。

通信プロセスのチェック

データキャプチャを取得するときは、通信プロセス (MFCS) がまだ実行中であることもチェックする必要があります。実行中であれば、現在の動作をチェックできます。手順は次のとおりです。

1. エンタープライズサーバが動作しているマシンのホスト名を書き留めます。
2. Enterprise Server Administration の Web インターフェイスに進みます。ホームページ上のサーバのリストで、チェックしたいエンタープライズサーバを見つけます。「通信プロセス」カラムで、そのサーバの 1 つ以上の通信プロセスを一覧表示します。カラムに表示されているポート番号をそれぞれ書き留めます。(ポート番号は、2 番目のコロン後のアドレスの最後の部分です。)
3. Web ブラウザで、次の URL に進みます。

```
http://hostname:port/MF_KEEPALIVE
```

ここで、hostname は手順 1 のホスト名 (または IP アドレス) です。port は手順 2 の最初のポート番号です。

MFCS が正常に動作していると、次のような応答があります。

```
Server=ESDEMO
ServerType=GKC
Version=1.2.5
Status=Started
```

4. 「MF_KEEPALIVE」のかわりに「MF_STATISTICS」を使用して、手順 3 を繰り返します。これで、MFCS の現在の動作、会話の処理数、または他の情報を参照できます。

複数の通信プロセスが実行している場合は、それぞれをこの方法でチェックできます。手順 2 で記録したそれぞれのポート番号は、通信プロセスと一致します。特定のサーバの通信プロセスが一部無効になっている場合があります。このような通信プロセスは、チェックする必要がありません。

障害の種類

障害は、次に示すカテゴリに分類されます。

不正な出力 (アプリケーションやシステムの機能が、正しくない結果を生成する。または、予期しない動作を示す)。これらには、アプリケーションのアベンドが含まれます。

予期しないシステム終了。エンタープライズサーバ全体の終了またはサブシステム (サポートプロセス) の終了。

システムループ。これらには、アプリケーションループとエンタープライズサーバのサブシステムループが含まれます。エンタープライズサーバのプロセスで CPU 時間の異常累積を示し、システムハングのような場合と同じ外見的な兆候を見せます。

システムハング。エンタープライズサーバがクライアントへの応答を停止するように見える問題です (ループを除く)。

障害の最初の手がかりは、次から得ます。

コンソールログ。これには、問題に至るエンタープライズサーバのシステムアクティビティの履歴が読みやすく記述されています。情報レベルメッセージ以外のメッセージを最初に調査する必要があります。障害が発生してからしばらくの間エンタープライズサーバが正常に見えることがよくあります。このため、障害が発生する 10 ~ 15 分前を詳細に調査します (情報メッセージを含む)。

プロセスのリスト。障害が発生しているエンタープライズサーバ (サーバがシャットダウンしていない場合) に関連するプロセスを検査します。正常なデータキャプチャのリストと、問題のあるデータキャプチャのリストの間のプロセスに大幅な CPU 時間の相違がある場合は、ループの可能性があります。

これらの手がかりと、外見的な兆候を組み合わせることで、リストされている障害の種類の中の 1 つに障害を分類するための情報が十分に得られます。

不正な出力

この手順では、アプリケーションの問題とエンタープライズサーバのインフラストラクチャの問題を区別します。

アベンド以外のアプリケーションの障害は、アプリケーションの問題であることがほとんどです。アクティブなトレースデータセットと、ダンプ X データセット (casdumpx.rec) のシステムトレーステーブルを見て、障害が発生する前のアプリケーションのアクティビティを明らかにします。

アプリケーションのアベンドは、アプリケーションの問題であることも、エンタープライズサーバのインフラストラクチャの問題であることもあります。アクティブなダンプデータセットから RTS エラーダンプを見つけます。エラーダンプが見つかったら、ここから RTS エラーを判断して問題を分類します。障害が発生したモジュールは、アプリケーションまたはエンタープライズサーバのインフラストラクチャですか？

予期しない終了

システムコンソールログでシャットダウンの原因を示すメッセージを調べ、それらのメッセージから障害を分類します。

ループ

ループしているプロセス名を使用して障害を分類します。ループしているプロセスがサービス実行プロセス (Windows 上の cassi および UNIX 上の cassi32。両プロセスとも以降では cassi と呼びます) である場合は、ループはエンタープライズサーバのサブシステムまたはアプリケーション内にある可能性があります。アクティブなトレースデータセットとダンプ X データセットのシステムトレーステーブルを参照して、最後に制御されたモジュールを決定します。アプリケーションの場合には、プロセスに FaultFind を使用して、次の障害発生時に情報をより多く取得できるようにすることを検討してください。

ループしているプロセスが cassi であっても、最後に制御されたモジュールがアプリケーションの一部でない場合、またはループしているプロセスが cassi でない場合には、問題を弊社のサポート窓口へ照会してください。

システムハング

すべてのクライアント要求に対してエンタープライズサーバからの応答がないのか、または 1 種類のクライアント要求のみに対して応答がないのかを調べます。

1 種類の要求が失敗している場合

アプリケーションがリソースの競合の影響を受けている可能性を検討します。サービスリスナーが失敗している可能性を検討します。

すべてのクライアント要求が失敗している場合

「現在のステータス」カラムの [詳細] をクリックして、エンタープライズサーバの「詳細」ページの「要求サービス実行プロセス数」で数値を指定し、[更新] をクリックして、サービス実行プロセスの数を動的に増加させてみます。新しいプロセスは、少し遅れた後にリストに表示されます。成功した場合は、共有メモリがロックされていないこと、および問題の外見的な症状が存在していても、エンタープライズサーバがまだあるレベルで正常に機能していることを示します。

ダンプ X データセットを使用して、アクティブなサービス実行プロセス (cassi プロセス) の数をチェックします。各サービス実行プロセスに対して、ディスパッチ制御ブロックを検査します。現在、すべてのサービス実行プロセスがトランザクションにより占有されていますか？ 同じトランザクションが各 cassi プロセスを占有していますか、または各 cassi プロセスが最後に実行したコマンドは同じですか？ どちらかが「はい」であれば、次の可能性を考え

ます。

問題のある種類の要求数が多いため使用可能な cassi プロセスがすべてブロックされ、エンタープライズサーバ全体が障害を起こしているように見える。これは、顧客用アプリケーションに問題がある可能性があります。

実行時間の長い正常な要求が数多く到着したため使用可能な cassi プロセスがすべて占有され、エンタープライズサーバ全体が障害を起こしているように見える。これに該当しそうな場合は、サービス実行プロセス数を増加させて問題を修正することができます。

他の可能性としては、サーバの共有メモリがロックされている場合があります。最もよい確認方法は、ダンプ×データセットの最後のトレースエントリの日時と、ダンプの日時を比較します。2つの時間差が大きいほど、共有メモリがロックされている可能性が高くなります。共有メモリがロックされていない場合は、サーバが何を待っているかを調べます。共有メモリがロックされている場合は、どのプロセスが共有メモリをロック状態にしたかを調べます。

診断サーバの使用方法

診断サーバとして、問題の調査専用エンタープライズサーバを使用している場合には、診断補助からほとんどの値を取得できます。そのようなサーバは、問題が発生した使用サーバからダンプとトレースデータを安定して収集できます。

診断サーバの構成

診断サーバは通常のエンタープライズサーバであり、ESMAC で使用できる診断機能は、他のサーバで使用できる診断機能とまったく同じです。診断サーバは、自身の診断を生成できない構成であることが、独特な点です。これにより、他のサーバからの診断情報を処理することが可能になります。診断サーバの推奨構成を、図 12-1 に示します。

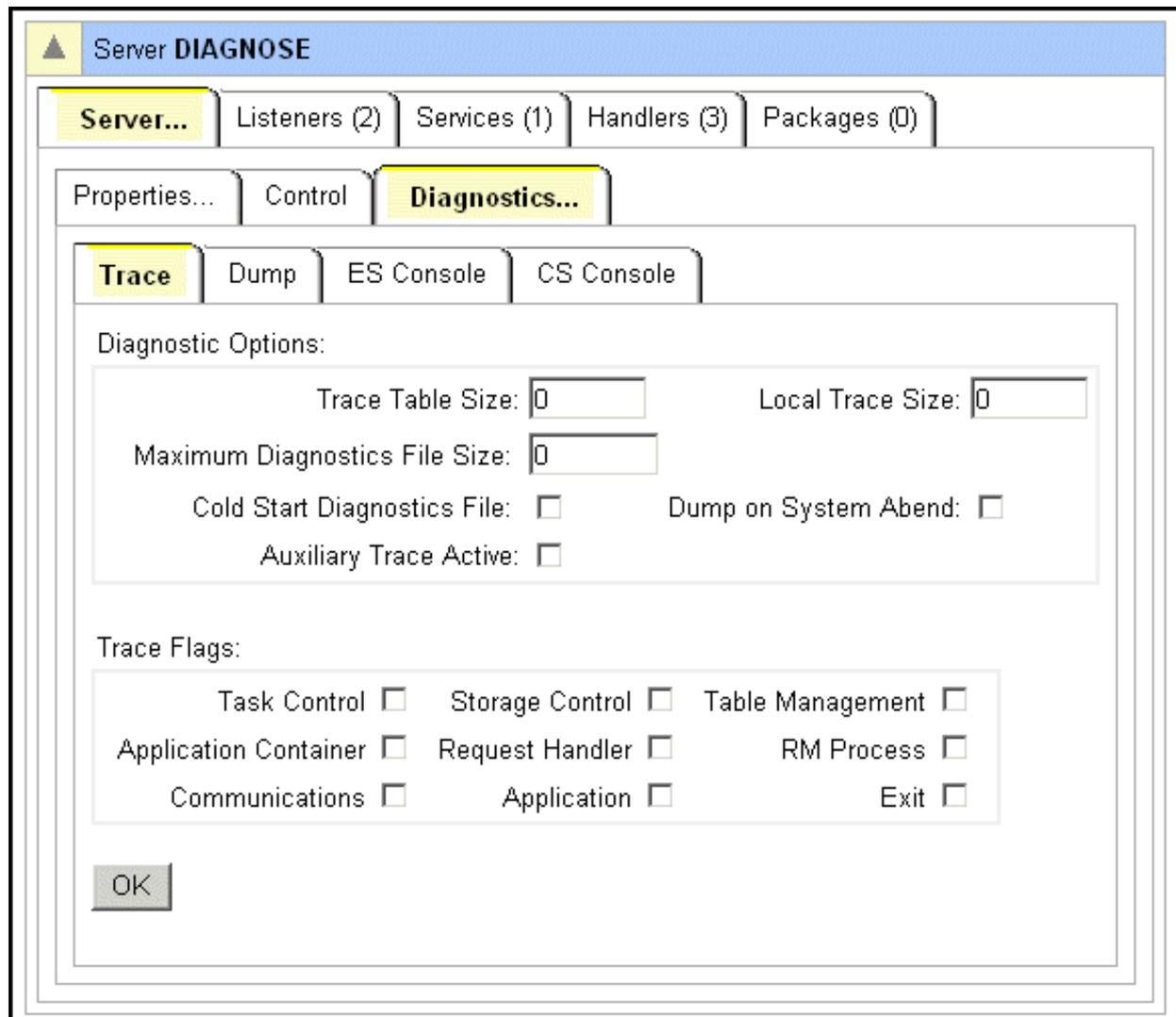


図 12-1 : 診断サーバの構成

診断データの移動

問題が発生したサーバから収集したデータの解析を始める前に、診断サーバにデータをコピーする必要があります。ここでは、収集したデータとログファイルを診断サーバに移動するコマンドについて説明します。%SystemDrive% は c: で、問題が発生したエンタープライズサーバのディレクトリ内に移動済みであると仮定します。

Net Express の一部として Enterprise Server をインストールした場合は、次の操作を実行します。

```
copy *.rec "C:\Documents and Settings\username\My Documents\Micro Focus
\Net Express 5.0\WORKAREA\diagnostics-server-name"
copy console.log "C:\Documents and Settings\username\My Documents\Micro Focus\
Net Express 5.0\WORKAREA\diagnostics-server-name\console.aux"
copy console.bak "C:\Documents and Settings\username\My Documents\Micro Focus\
Net Express 5.0\WORKAREA\diagnostics-server-name\console.bak"
```

次は、diagnose と呼ばれる診断サーバへ必要データをコピーするサンプルスクリプトです。

```
copy *.rec "C:\Documents and Settings\username\My Documents\
Micro Focus\Net Express 5.0\WORKAREA\diagnose\
copy console.log "C:\Documents and Settings\username\My Documents\Micro Focus\
Net Express 5.0\WORKAREA\diagnose\console.aux"
copy console.bak "C:\Documents and Settings\username\My Documents\Micro Focus\
Net Express 5.0\WORKAREA\diagnose\console.bak"
```

データを診断サーバへ移動した後に、問題が発生したサーバを再開します。

コンソールログの表示

上記のコマンドを使用して問題のあるサーバからコピーされた診断ファイルが存在する診断サーバでのみ意味を持つコンソールログを表示する特別なオプションがあります。これらの特別なオプションを使用するには、次の操作を実行します。

[ログ] と [C (現在)] をクリックし、[表示] をクリックして、現在のコンソールログファイル (診断サーバの console.log ファイル) を表示します。

[ログ] と [B (バックアップ)] をクリックし、[表示] をクリックして、問題のあるサーバのバックアップコンソールログ (console.bak) を表示します。この操作は、診断サーバを起動した後にデータをコピーした場合のみに行えるので注意が必要です。これは、サーバの起動プロセスにより、問題のあるサーバからのバックアップコンソールログが診断サーバのバックアップコンソールログに置き換えられてしまうためです。

[ログ] と [A (補助)] をクリックし、[表示] をクリックして、問題のあるサーバのコンソールログ (console.aux) を表示します。

システムトレースの表示

ここで説明する内容は、ヘルプトピック『[システムトレースの表示](#)』で説明されています。オプションは複雑ですので、例を参考にしてください。

システムトレーステーブルのトレースを取得するには、メニューの診断グループ上の [トレース] と、A または B データセットを選択するために [A] または [B] をクリックします ([C] は、診断サーバに属するインメモリのトレーステーブルのため選択できません)。「ブロック」に、1 つのトレースインデックスエントリにまとめるトレースブロック数を入力します。[表示] をクリックします。トレースインデックスの表示を、図 12-2 に示します。

The screenshot displays the 'Trace A' interface. At the top, there is a 'Refresh' button. Below it, a header row indicates the columns: Action [Blocks] [Level] [PIDs] [Task IDs] and (mm/dd/yyyy) Time (HH:MM:SS.hh). The main content area shows three rows of trace data, each with a 'Details' button, a '10' input for blocks, a '0' dropdown for level, and checkboxes for various PIDs and Task IDs. The left sidebar includes a 'Diagnostics' section with radio buttons for Log, A, Dump, B, Trace, and C, and a '10 Blocks' input field. Other sections include Resources (Services, Package, Handlers), Users, Env.Vars., Chg.User, and Help (Menu, This Page, Contents).

図 12-2: 「トレースインデックス」ページ

各ブロックのサイズは、システムトレーステーブルのサイズと同じです。つまり、各ブロックには、「サーバの編集」ページの「トレーステーブルサイズ」で入力したトレースエントリ数が含まれています。インデックスエントリは、各エントリに対してタイムスタンプが表示されます。これは、ブロックの各組の先頭トレースエントリにおけるタイムスタンプです。これらのタイムスタンプは、検索対象を特定のイベントに絞るのに役立ちます。

実際のトレースを表示するには、1つのインデックスエントリから、次に示すフィールドを選択します。

「ブロック」([詳細]のすぐ右)に、トレースに含めたいブロック数を入力します。このデフォルト値は、メニューの診断グループ上の「ブロック」に入力した値です。ブロック数は、希望する値に変更できます。2つの「ブロック」コントロールを指定することにより、フォーマットされたトレースの表示内容をより正確に制御することができます。

「レベル」(右隣のコントロール)に、フォーマットされたトレースの詳細レベルを指定します。オプションを次に示します。

- 0 サービスの開始、モジュールの開始、モジュールの終了、およびサービスの終了。これは、デフォルト設定です。
- 1 エンタープライズサーバAPIの入口と出口
- 2 サブシステムのトレースエントリ
- 3 いくつかのデータのフォーマットが可能な、すべてのエントリ
- 4 すべてのトレースエントリ

「PID」(右隣のコントロール)で、確認するプロセスのプロセスIDのチェックボックスをチェックします。チェックボックスは、このトレースインデックスエントリのブロック内にトレースエントリをもつすべてのブ

ロセスに対して表示されます。プロセス ID は、プロセスの機能を示す先頭の文字と数字で記述されています。先頭の文字の意味を、次に示します。

- C 通信プロセス (MFCS)
- F Fileshare サーバ (内部)
- J ジャーナルコントロール
- M サーバマネージャ
- R 回復
- S サービス実行プロセス (SEP)
- U 不明
- Z 汎用的なターミナルとクライアント

「タスク ID」(右隣のコントロール) で、確認するタスクのタスク ID のチェックボックスをチェックします。チェックボックスは、このトレースインデックスエントリのブロック内にトレースエントリをもつすべてのタスクに対して表示されます。

「レベル」、「PID」、および「タスク ID」コントロールは、確認したい特定の情報を選択するためのフィルタです。「レベル」を 0 (デフォルト) のままにし、「PID」および「タスク ID」のチェックボックスに全くチェックをしないと、トレースインデックスエントリのブロック内にエントリをもつすべてのプロセスおよびタスクに対して最小レベルの情報を参照することができます。選択しても、トレース情報が生成されないことがあります。これは、プロセスとタスクの両方を選択し、選択したプロセス上で選択したタスクが動作していない場合に起こります (2 つ以上のプロセス上で 1 つのタスクが動作することはできません)。タスクが通信作業を要求すると、その通信作業は独自のプロセスとタスクを持ちます。2 つのタスク ID を自動で関連付ける方法はありません。フィルタにかかわらず、ブロックの最初のトレースエントリは常に表示されます。

参照したい情報を選択する例を示します。メニューの診断グループ上の「ブロック」に、10 を入力したとします。実際には、データセットには 60 ブロックの情報があるため、6 つのインデックスエントリが表示されます。特定の SEP を調査したく、その SEP がインデックスエントリの 3 番目と 4 番目にリストされているとします。3 番目のインデックスエントリの「ブロック」に 20 を入力します。対象の SEP のプロセス ID のチェックボックスをチェックし、[詳細] をクリックします。これにより、3 番目のインデックスエントリの最初のブロックから始まり、調査対象のトレースエントリを含むすべてのブロックで、該当の SEP のトレースエントリのみを参照できます。

各トレースエントリには、次の情報があります。

見出し	内容
None	トレースイベントがあれば、トレースイベントの説明。
Seq	昇順のエントリ連続番号
Task-Nbr	5 桁のタスク番号
ProcessID	5 桁のプロセス ID
ID	トレースされたイベント、コマンド、または命令の ID。4 バイトの 16 進数です。
hhmmssshh	トレースエントリが記録されたタイムスタンプ。タイムスタンプは、時、分、秒、および 100 分の 1 秒で示されます。
aaaa bbbb	8 バイトのエントリ固有データ

トレースの参照が終わったら、[戻る] をクリックして「トレースインデックス」ページに戻ります。

ダンプの表示

ここで説明する内容は、ヘルプトピック『[ダンプの表示](#)』で説明されています。トレースと同じように、オプションは複雑ですので、例を参考にしてください。

ダンプは、概要または詳細を取得することができます。概要は、ダンプの詳細の末尾に記述されます。ダンプは、概要または詳細のどちらでも、システムトレースとローカルトレース (SEP 用トレース) の両方を含みます。

ダンプを取得するには、メニューの診断グループ上の「ダンプ」を、次に [A] または [B] をクリックして A または

B データセットを選択するか、または [C] をクリックしてダンプ X データセット (外部的に起動されるダンプで作成) を選択します。「ブロック」は無視します。これは、出力に影響しません。次に [表示] をクリックします。「ダンプインデックス」ページの表示を、図 12-3 に示します。

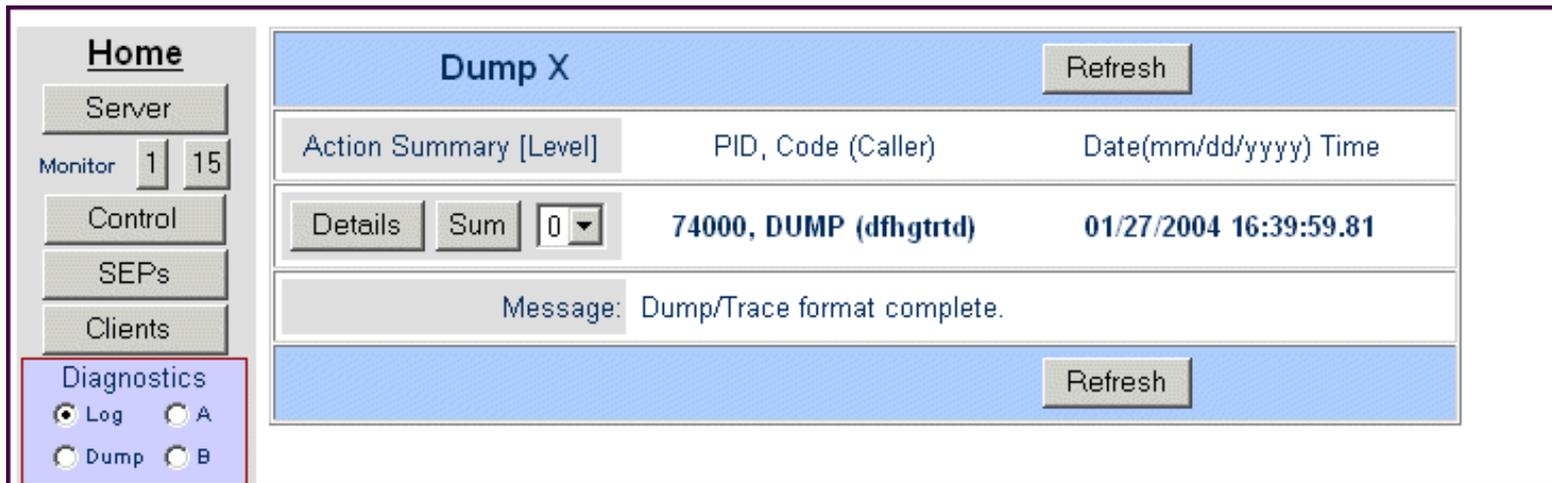


図 12-3 : 「ダンプインデックス」ページ

インデックスエントリは、各エントリに対して次の情報が表示されます。

プロセス ID
 ダンプを識別するコード
 ダンプの要求元
 日時

ダンプを表示するには、1 つのインデックスエントリから、「レベル」(右に 3 番目のコントロール) で、フォーマットされたダンプの詳細なトレースのレベルを指定します。オプションを次に示します。

- 0 サービスの開始、モジュールの開始、モジュールの終了、およびサービスの終了。
- 1 エンタープライズ サーバ API の入口と出口
- 2 サブシステムのトレースエントリ
- 3 何らかのデータのフォーマットが可能、すべてのエントリ
- 4 すべてのトレースエントリ これは、デフォルト設定です。

ローカルトレーステーブル (SEP 用トレース) には、レベル 0 と 1 しかありません。

このオプションは、ダンプの詳細には影響しますが、ダンプの概要には影響しません。

ダンプの詳細を参照するには [詳細] を、概要を参照するには [概要] をクリックします。

ダンプの参照が終わったら、[戻る] をクリックして「ダンプインデックス」ページに戻ります。

無効なクレデンシャルエラー

エンタープライズサーバの起動時に、コンソールログに次のような INVALID_CREDENTIALS エラーが表示されることがあります。

CASCD4002W MFDS へのバインディング処理中にエラー (INVALID_CREDENTIALS) が発生しました。
 CASCD0102F TX デーモンが初期化に失敗しました。

これは、Directory Server のユーザ ID とパスワード (暗号化形式) の詳細を保持している内部ファイルに問題があります。このエラーが発生した場合は、Micro Focus Directory Server のスキーマディレクトリ (通常は C:\Documents and Settings\All Users\Application Data\Micro Focus\NetExpress\5.0\MFDS) に次のファイルがある

ことを確認してください。

usr.dat
ugp.dat
ace.dat
prm.dat

これらのファイルは、最初に install-dir¥base¥bin¥cciusers.dat ファイルから作成されます。このファイルの読み取り、または初期セキュリティ情報の作成に問題がある場合は、このファイルを再度インポートしてデータを作り直すことができます。手順は次のとおりです。

1. Enterprise Server Administration のホームページで、左側にあるメニューの [ユーザ] をクリックします。
2. ユーザのリストの下にある [インポート...] をクリックします。

cciusers.dat ファイルの場所を確認するページが表示され、既存のユーザ情報をすべて置き換えるかどうか確認されます (デフォルトは [はい])。

注： [インポート...] は、Directory Server が制限モードで実行されている場合は使用できません。

3. [OK] をクリックします。

エンタープライズサーバを起動するには、特定のシステムユーザアカウントが存在していなければなりません。必要なユーザ ID は、mf_mdsa および mf_cs で、これらのユーザアカウントは製品とともに出荷されるデフォルトの cciusers.dat ファイルに含まれています。

Copyright © 2006 Micro Focus (IP) Ltd. All rights reserved.

第 13 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。

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、『Web サービスと COBOL』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。

提供されているユーザ出口点

図 13-1 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。

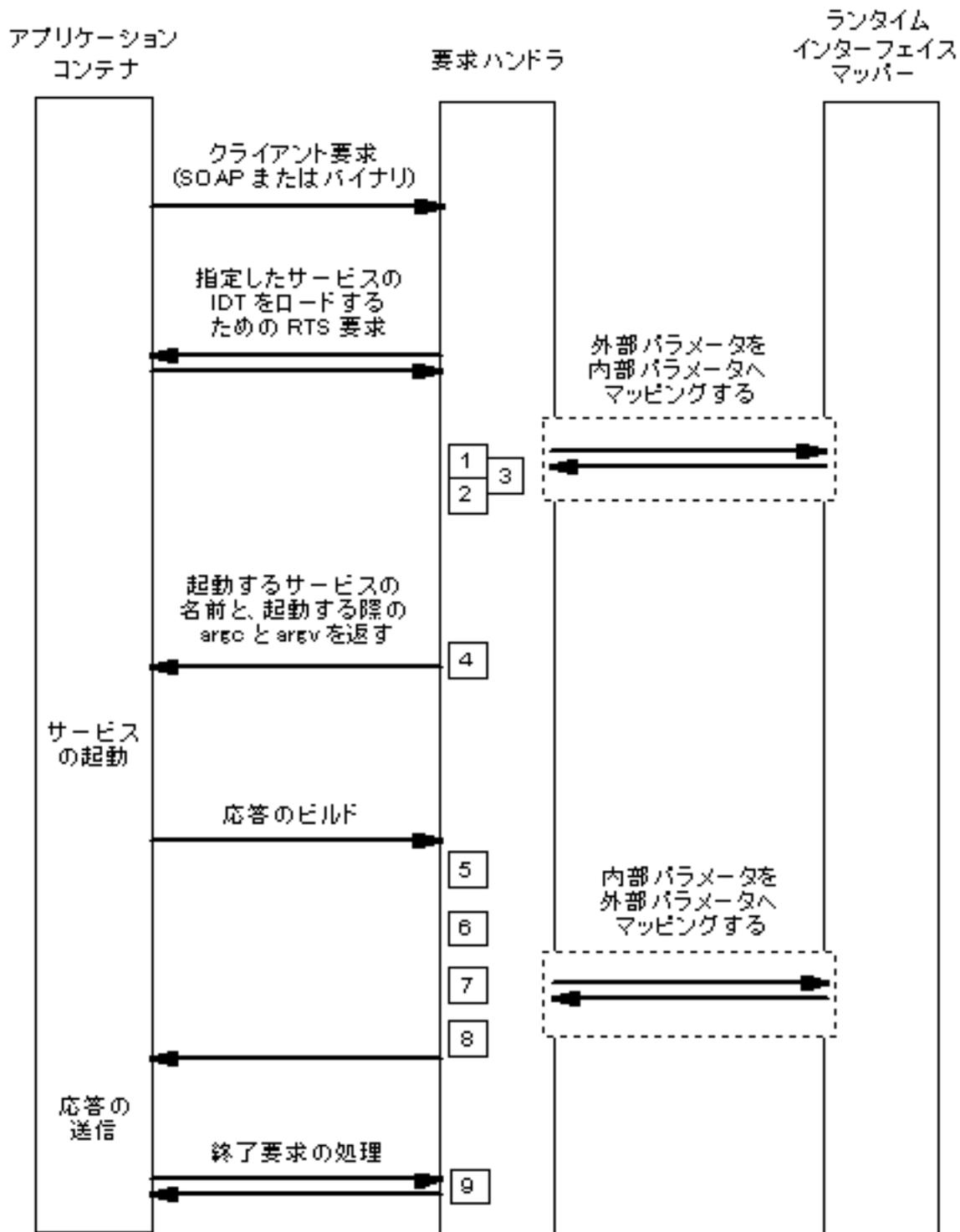


図 13-1: 要求ハンドラの出口点

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す。 |
| 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解放できます。 |
| 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解放できます。 |

構造体定義と型定義

COBOL プログラマと C プログラマのために、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、ユーザ出口および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](#)』

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).

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。

Copyright © 2006 Micro Focus (IP) Ltd. All rights reserved.



トラブルシューティング

