



BEA Tuxedo

BEA Tuxedo
Domains コンポーネント

BEA Tuxedo リリース 8.0J
8.0 版
2001 年 10 月

Copyright

Copyright © 2001 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc.

All other company names may be trademarks of the respective companies with which they are associated.

BEA Tuxedo Domains コンポーネント

Document Edition	Date	Software Version
8.0J	2001 年 10 月	BEA Tuxedo リリース 8.0J

目次

このマニュアルについて	
対象読者	vii
e-docs Web サイト	vii
マニュアルの印刷方法	viii
関連情報	viii
サポート情報	ix
表記上の規則	ix
1. Domains について	
BEA Tuxedo の Domains コンポーネントとは	1-1
相互運用するビジネス・オペレーション	1-2
マルチ・ドメイン・コンフィギュレーションを構築する	1-3
マルチ・ドメイン・アプリケーションの設定および管理用ツール	1-4
ドメイン・ゲートウェイの種類	1-5
ドメイン・ゲートウェイでサポートされる機能	1-7
ドメイン・ゲートウェイを使用したアプリケーションの例	1-8
ドメイン・ゲートウェイでサポートされるメッセージング・パラダイム	1-10
ローカル・サービスとリモート・サービス間での要求 / 応答型の通信	1-11
ローカル・サービスとリモート・サービス間での会話型通信	1-12
データ記憶域のキューを使用したメッセージング	1-12
型付きバッファを使用してデータをパッケージ化する	1-13
Domains のトランザクション・タイムアウトとブロッキング・ タイムアウトを定義する	1-14
Domains の接続方法を指定する	1-15
Dynamic Status 機能を使用してリモート・サービスの可用性を 判定する	1-15
接続方針と Dynamic Status 機能の関係	1-17
Domains コンフィギュレーション・ファイルとは	1-18
DMCONFIG ファイルのセクションの記述	1-18
Domains 関連の新しい用語	1-20
Domains コンフィギュレーション・ファイルを変換する	1-20
DMCONFIG ファイルをバイナリ形式に変換する	1-21
BDMCONFIG ファイルをテキスト形式に変換する	1-22
BEA Tuxedo システムの Domains の特徴	1-22

2. ATMI Domains の計画とコンフィギュレーション	
複数の BEA Tuxedo アプリケーションで Domains を構成する	2-2
オプション 1: アプリケーションを再コンフィギュレーションする	2-3
アプリケーションの統合用のコンフィギュレーション・ファイル	2-4
オプション 1 の制限事項	2-7
オプション 2: アプリケーションを別々の BEA Tuxedo Domains として	
再定義する	2-8
コンフィギュレーション・ファイルを変更する	2-8
DMCONFIG ファイルを追加する	2-11
Domains のサンプル・アプリケーション : creditapp	2-12
creditapp の README ファイル	2-13
Domains 環境を設定する	2-17
Domains のサンプル・アプリケーション (simpapp) を	
コンフィギュレーションする	2-18
コンフィギュレーションの作業	2-19
lapp の環境変数を設定する	2-20
例	2-20
ubbconfig ファイルで lapp の Domains 環境を定義する	2-21
サーバの定義	2-21
lapp 用のアプリケーション・コンフィギュレーション・ファイルの例	2-22
DMCONFIG ファイルで lapp の Domains パラメータを定義する	2-23
lapp 用のドメイン・ゲートウェイ・コンフィギュレーション・	
ファイルの例	2-24
lapp 用のアプリケーション・コンフィギュレーション・ファイルと	
ドメイン・ゲートウェイ・コンフィギュレーション・ファイルを	
コンパイルする	2-25
rapp の環境変数を設定する	2-26
例	2-26
UBBCONFIG ファイルで rapp の Domains 環境を定義する	2-26
rapp 用のアプリケーション・コンフィギュレーション・ファイルの例	2-27
DMCONFIG ファイルで rapp の Domains パラメータを定義する	2-28
rapp の Domain ゲートウェイ・コンフィギュレーション・	
ファイルの例	2-28
rapp 用のアプリケーション・コンフィギュレーション・ファイルと	
ドメイン・ゲートウェイ・コンフィギュレーション・ファイルを	
コンパイルする	2-29
Domains 間でデータを圧縮する	2-30
サービス要求をリモート・ドメインにルーティングする	2-31
Domains でセキュリティを設定する	2-33
BEA Tuxedo アプリケーションのセキュリティと Domains の	
セキュリティの関係	2-34
Domains のセキュリティ・メカニズム	2-35

ドメイン用のアクセス制御リスト (ACL) を作成する	2-37
インポートされたリモート・サービスに対して標準の BEA Tuxedo アクセス制御リストを使用する	2-37
リモート・ドメインの ACL 方針を設定する	2-38
リモート・ドメインのクリデンシャル方針を設定する	2-40
Domains の認証機能を設定する	2-41
T_DM_PASSWORDS MIB クラスの定義	2-42
Domains パスワードを設定する	2-42
Domains 間のセキュリティのコーディング例	2-43
例 1: セキュリティとして NONE を設定する	2-43
例 2: セキュリティとして APP_PW を設定する	2-46
ドメイン間の接続を設定する	2-47
起動時に接続する (ON_STARTUP 方針)	2-47
クライアントから要求されたときに接続する (ON_DEMAND 方針)	2-48
メッセージの受信時にのみ接続する (INCOMING_ONLY 方針)	2-49
再接続間隔を設定する (ON_STARTUP のみ)	2-50
Domains 間の接続を制御する	2-52
Domains 間の接続を確立する	2-52
Domains 間の接続を解除する	2-52
接続状況のレポートを作成する	2-53
Domains 環境でのフェイルオーバーとフェイルバックを設定する	2-54
リンク・レベルのフェイルオーバーをサポートするよう Domains を コンフィギュレーションする	2-54
Domains レベルのフェイルオーバーとフェイルバックを設定する	2-55
3. CORBA Domains の計画とコンフィギュレーション	
マルチ・ドメイン環境 CORBA Domains の概要	3-1
ドメイン間通信	3-2
マルチ・ドメイン・コンフィギュレーションの要素	3-3
複数の CORBA ドメインを設定する	3-5
コンフィギュレーション・ファイル	3-5
ドメイン・コンフィギュレーション・ファイル (DMCONFIG)	3-7
factory_finder.ini ファイル	3-14
ローカル・ファクトリ	3-20
CORBA ドメイン・コンフィギュレーションの種類	3-20
直接接続ドメイン	3-21
間接接続ドメイン	3-21
例: 複数の CORBA ドメインの設定	3-22
UBBCONFIG ファイルの例	3-23

4. Domains の管理	
Domains の実行時管理コマンドを使用する	4-1
DMADM と Domain ゲートウェイ・グループの移行方法	4-3
管理インターフェイス dmadm(1) を使用する	4-4
Domains 管理サーバ DMADM(5) を使用する	4-5
ゲートウェイ管理サーバ GWADM(5) を使用する	4-6
ゲートウェイ・プロセスを使用する	4-7
Domains 環境でのトランザクションの管理	4-8
トランザクション管理機能	4-9
Domains で TMS 機能を使用する	4-9
ゲートウェイによる Domains 間でのトランザクションの調整方法	4-10
トランザクションで GTRID マッピングを使用する	4-12
密結合関係と疎結合関係の定義	4-12
Domains をまたがるグローバル・トランザクション	4-13
ログ機能によるトランザクションのトラッキング	4-20
ログ機能のしくみ	4-20
失敗したトランザクションの回復	4-22

このマニュアルについて

このマニュアルでは、データ依存型の BEA Tuxedo ATMI 環境またはオブジェクト指向の BEA Tuxedo CORBA 環境で BEA Tuxedo® ドメインを設定および管理する方法について説明します。

このマニュアルでは、以下の内容について説明します。

- 「第 1 章 Domains について」では、BEA Tuxedo の Domains コンポーネントの概要を示します。
- 「第 2 章 ATMI Domains の計画とコンフィギュレーション」では、BEA Tuxedo ATMI 環境のドメインの計画および設定方法について説明します。
- 「第 3 章 CORBA Domains の計画とコンフィギュレーション」では、BEA Tuxedo CORBA 環境のドメインの設定方法について説明します。
- 「第 4 章 Domains の管理」では、BEA Tuxedo ドメインの管理方法について説明します。

対象読者

このマニュアルは、主にミッション・クリティカルな BEA Tuxedo システムをサポートするパラメータの設定を行うシステム管理者を対象としています。また、BEA Tuxedo プラットフォームの知識があることを前提としています。

e-docs Web サイト

BEA 製品のマニュアルは BEA 社の Web サイト上で参照することができます。BEA ホーム・ページの [製品のドキュメント] をクリックするか、または <http://edocs.beasys.co.jp/e-docs/index.html> に直接アクセスしてください。

マニュアルの印刷方法

このマニュアルは、ご使用の Web ブラウザで一度に 1 ファイルずつ印刷できます。Web ブラウザの [ファイル] メニューにある [印刷] オプションを使用してください。

このマニュアルの PDF 版は、Web サイト上にあります。また、マニュアルの CD-ROM にも収められています。この PDF を Adobe Acrobat Reader で開くと、マニュアル全体または一部をブック形式で印刷できます。PDF 形式を利用するには、BEA Tuxedo Documents ページの [PDF 版] ボタンをクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader をお持ちではない場合は、Adobe Web サイト (<http://www.adobe.co.jp/>) から無償で入手できます。

関連情報

以下のマニュアルには、BEA Tuxedo ソフトウェアについての関連情報が掲載されています。

- 『BEA Tuxedo システムのインストール』 CD に同梱されているドキュメント
- 『BEA Tuxedo リリース・ノート』 CD に同梱されているドキュメント
- 『BEA Tuxedo アプリケーションの設定』 BEA Tuxedo オンライン・マニュアル CD で参照できます。このガイドには、BEA Tuxedo システムのセットアップおよび管理方法についての情報が記載されています。
- 『BEA Tuxedo アプリケーション実行時の管理』 BEA Tuxedo オンライン・マニュアル CD で参照できます。このガイドでは、BEA Tuxedo アプリケーションの実行時の管理方法について説明しています。
- 『BEA Tuxedo CORBA アプリケーションのスケーリング、分散、およびチューニング』 BEA Tuxedo オンライン・マニュアル CD で参照できます。このガイドには、BEA Tuxedo CORBA 環境で実行する CORBA アプリケーションの調整方法が記載されています。

BEA Tuxedo ATMI および BEA Tuxedo CORBA 環境の設定と管理方法の詳細については、<http://edocs.bea.com/> の 『CORBA Bibliography』を参照してください。

サポート情報

皆様の BEA Tuxedo マニュアルに対するフィードバックをお待ちしています。ご意見やご質問がありましたら、電子メールで docsupport-jp@bea.com までお送りください。お寄せいただきましたご意見は、BEA Tuxedo マニュアルの作成および改訂を担当する BEA 社のスタッフが直接検討いたします。

電子メール メッセージには、BEA Tuxedo 8.0 リリースのマニュアルを使用していることを明記してください。

BEA Tuxedo に関するご質問、または BEA Tuxedo のインストールや使用に際して問題が発生した場合は、www.bea.com の BEA WebSUPPORT を通じて BEA カスタマ・サポートにお問い合わせください。カスタマ・サポートへの問い合わせ方法は、製品パッケージに同梱されているカスタマ・サポート・カードにも記載されています。

カスタマ・サポートへお問い合わせの際には、以下の情報をご用意ください。

- お客様のお名前、電子メール・アドレス、電話番号、Fax 番号
- お客様の会社名と会社の住所
- ご使用のマシンの機種と認証コード
- ご使用の製品名とバージョン
- 問題の説明と関連するエラー・メッセージの内容

表記上の規則

このマニュアルでは、以下の表記規則が使用されています。

規則	項目
太字	用語集に定義されている用語を示します。
Ctrl + Tab	2 つ以上のキーを同時に押す操作を示します。
イタリック体	強調またはマニュアルのタイトルを示します。

規則	項目
等幅テキスト	<p>コード・サンプル、コマンドとオプション、データ構造とメンバ、データ型、ディレクトリ、およびファイル名と拡張子を示します。また、キーボードから入力する文字も示します。</p> <p>例：</p> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
等幅太字	<p>コード内の重要な単語を示します。</p> <p>例：</p> <pre>void commit ()</pre>
等幅イタリック体	<p>コード内の変数を示します。</p> <p>例：</p> <pre>String <i>expr</i></pre>
大文字	<p>デバイス名、環境変数、および論理演算子を示します。</p> <p>例：</p> <pre>LPT1 SIGNON OR</pre>
{ }	<p>構文の行で選択肢を示します。かっこは入力しません。</p>
[]	<p>構文の行で省略可能な項目を示します。かっこは入力しません。</p> <p>例：</p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
	<p>構文の行で、相互に排他的な選択肢を分離します。記号は入力しません。</p>

規則	項目
...	<p>コマンド行で次のいずれかを意味します。</p> <ul style="list-style-type: none"> ■ コマンド行で同じ引数を繰り返し指定できること ■ 省略可能な引数が文で省略されていること ■ 追加のパラメータ、値、その他の情報を入力できること <p>省略符号は入力しません。</p> <p>例：</p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
<p>.</p> <p>.</p> <p>.</p>	<p>コード例または構文の行で、項目が省略されていることを示します。省略符号は入力しません。</p>



1 Domains について

ここでは、次の内容について説明します。

- BEA Tuxedo の Domains コンポーネントとは
- マルチ・ドメイン・コンフィギュレーションを構築する
- ドメイン・ゲートウェイを使用したアプリケーションの例
- ドメイン・ゲートウェイでサポートされるメッセージング・パラダイム
- Domains のトランザクション・タイムアウトとブロッキング・タイムアウトを定義する
- Domains の接続方法を指定する
- Domains コンフィギュレーション・ファイルとは
- Domains コンフィギュレーション・ファイルを変換する
- BEA Tuxedo システムの Domains の特徴

BEA Tuxedo の Domains コンポーネントとは

BEA Tuxedo のアプリケーション・プログラミング・フレームワークを使用すると、アプリケーション処理の分散に伴う複雑さを透過的に処理できるため、オンライン・トランザクション処理 (OLTP) 用のオープンな分散アプリケーションを簡単に開発できます。このフレームワークは、次のコンポーネントで構成されています。

- さまざまなコンピュータやアプリケーション・プログラムの異種性、およびこれらのアプリケーション・プログラムの位置を透過的に扱う、拡張されたクライアント / サーバ型モデル。
- 集中管理型のサブシステム。アプリケーション管理者は、連係するすべてのマシンを単一のアプリケーションとして制御できます。

ビジネスの範囲が拡大すると、社内アプリケーション開発者は、異なる部門を機能別にグループ化し、各グループを個別に管理しつつ、サービスやデータは共有できるような組織を作ることと求められます。各アプリケーションの機能、地理性、機密度、および将来の拡張の見込みの違いにより、複数のアプリケーションを単一の分散アプリケーションにまとめることが常に適切であるとは限りません。また、企業は、異なるトランザクション処理モニタ (BEA 社の TOP END、Transarc の Encina、IBM の CICS、Bull の TDS、Bull の TP8、ICL の TPMS など) を使用して OLTP サービスを管理するほかの企業と提携して、ビジネスの拡大を図る場合もあります。

機能別に分けられた各グループは、1 つまたは複数のコンピュータを含むアプリケーションとして定義され、ほかのアプリケーションとは独立して管理されます。このように、特定の機能を持ったアプリケーションをドメインと呼びます。実際の組織では、ドメイン名にドメインの機能を付けます。たとえば、「課金ドメイン」や「オーダ・エントリ・ドメイン」のようになります。BEA Tuxedo ドメインとは、1 つの BEA Tuxedo コンフィギュレーション・ファイルで管理する、単一のコンピュータまたは複数のコンピュータによるネットワークを指します。

相互運用するビジネス・オペレーション

BEA Tuxedo システムの Domains 機能は、ビジネス・ドメイン間の相互運用性を実現するフレームワークを提供します。このフレームワークは、拡張された BEA Tuxedo のクライアント / サーバ型モデルに基づいています。相互運用性とは、単にドメイン間の通信を意味するだけではありません。Domains は、ローカル・ドメインのユーザからリモート・ドメインのサービスに対する要求、またはリモート・ドメインのユーザからローカル・ドメインのサービスに対する要求を透過的に処理できるため、組織内のビジネス・アプリケーション間の壁は、事実上なくなります。アプリケーション・プログラマは、ATMI インターフェイスを使用して、リモート・ドメインが提供するサービスにアクセスしたり、リモート・ドメインで実行されるサービスを定義します。

BEA Tuxedo アプリケーションは、Domains 機能を使用しつつ、ほかの管理ドメインで動作するアプリケーションと連携することができます。BEA Tuxedo システムには、異なるトランザクション処理システムを実行するドメインを含む、大規模なアプリケーションを制御するための共通のフレームワークが用意されています。

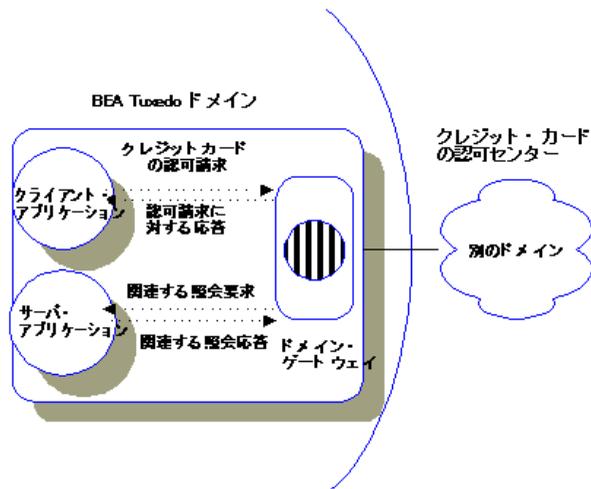
マルチ・ドメイン・コンフィギュレーションを構築する

マルチ・ドメイン・コンフィギュレーションを構築するには、次のタスクを考慮する必要があります。

- 既存の BEA Tuxedo アプリケーションをほかのドメインと統合する
- ドメイン間での相互運用性を保証する
- ドメイン間でのサービスへのアクセスを維持または制限する
- ドメイン間でのサービス要求を受け付けるか、または拒否する

Domains では、非同期性の高い、マルチタスク型およびマルチスレッド型のゲートウェイを使用して、これらのタスクを実現します。BEA Tuxedo に組み込まれているドメイン・ゲートウェイ (DGW) は、リモート・ドメインとの間で送受信される要求を処理するサーバです。トランザクションでは、どんな要求を処理することもできます。次の図は、1 つの BEA Tuxedo ドメインが、ドメイン・ゲートウェイを介して別のドメインと通信する方法を示しています。

図 1-1 ゲートウェイを介した双方向通信



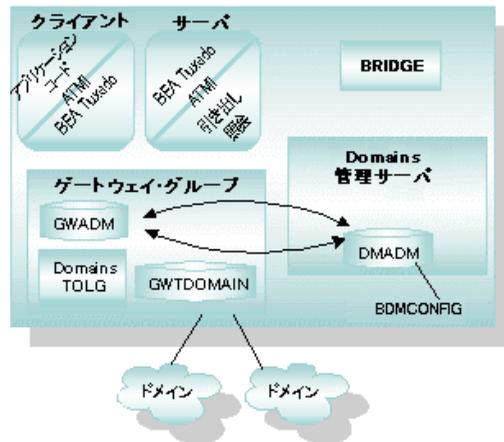
この図では、ゲートウェイがクレジット・カードの認可要求を処理し、別のドメインに送信する様子を示しています。このゲートウェイは、認可要求に対する応答も処理します。

ドメイン・ゲートウェイは、ドメイン間のすべての通信を管理します。ゲートウェイ・プロセスには、ドメイン・ゲートウェイ・グループを実行時に管理するゲートウェイ管理サーバ (GWADM)、および BEA Tuxedo アプリケーション全体に渡る Domains のコンフィギュレーション情報を実行時に管理する Domains 管理サーバ (DMADM) があります。

マルチ・ドメイン・アプリケーションの設定および管理用ツール

次の図は、BEA Tuxedo システムに用意されている、マルチ・ドメイン・コンフィギュレーションを設定および管理するためのツールを示しています。

図 1-2 Domains の管理用ツール



Domains 管理ツール	説明
<code>dmadmin(1)</code>	ドメイン・ゲートウェイ・グループを動的にコンフィギュレーション、監視、および調整するコマンドです。アプリケーションの実行中に <code>BDMCONFIG</code> ファイルを更新するには、このコマンドを使用します。このコマンドは、管理コマンドを変換するフロント・エンド・プロセスとして機能します。管理コマンドは、DMADM サーバにより宣言される一般的な管理サービス、つまり、DMADM サービスに要求を送信します。この結果、DMADMIN サービスは、 <code>BDMCONFIG</code> ファイル内の情報を検証、取得、または更新する関数を呼び出します。

Domains 管理ツール	説明 (続き)
DMCONFIG(5) BDMCONFIG	DMCONFIG は、マルチ・ドメイン・コンフィギュレーション用のテキスト形式のコンフィギュレーション・ファイルです。BDMCONFIG は、バイナリ形式のコンフィギュレーション・ファイルです。
dmloadcf(1) および dmunloadcf(1)	dmloadcf は、DMCONFIG ファイルを読み取り、構文をチェックし、オプションでバイナリ形式の BDMCONFIG コンフィギュレーション・ファイルをロードします。 dmunloadcf は、BDMCONFIG コンフィギュレーション・ファイルをバイナリ形式からテキスト形式に変換します。
DMADM(5)	実行時に Domains のコンフィギュレーションを管理する管理サーバです。DMADM は、ゲートウェイ・グループの登録サービスを提供します。このサービスは、GWADM サーバの初期化処理で要求されます。登録サービスにより、ゲートウェイ・グループが要求するコンフィギュレーション情報がダウンロードされます。DMADM サーバは、登録済みのゲートウェイ・グループのリストを管理し、コンフィギュレーション・ファイル (BDMCONFIG) が変更されると、変更内容をリスト内のゲートウェイ・グループに伝播します。
GWADM(5)	特定のゲートウェイ・グループを実行時に管理する管理サーバです。このサーバは、DMADM サーバに登録し、対応するゲートウェイ・グループが使用するコンフィギュレーション情報を取得します。GWADM は、DMADMIN からの要求、つまり、指定したゲートウェイ・グループの実行時オプションでの統計情報や変更に対する要求を受け付けます。GWADM は、「I-am-alive」メッセージを定期的に DMADM サーバに送信します。DMADM から応答が返されない場合、GWADM は再度登録を行います。このプロセスにより、GWADM サーバは、そのグループの Domains コンフィギュレーションに関する最新の情報を常に保持できます。
GWTDOMAIN(5)	接続されたドメイン (BEA Tuxedo Domains) 内のクライアントとサーバからメッセージを受信し、転送するゲートウェイ・プロセスです。

ドメイン・ゲートウェイの種類

BEA Tuxedo システムには、リモート・ドメインとの通信に使用するさまざまなネットワーク・トランスポート・プロトコルに対応するため、いくつかのゲートウェイが用意されています。同じ通信プロトコルおよび同じトランザクション・コミット・プロトコルを使用するリモート・ドメインへのアクセスは、特定のローカル・ドメインに対して定義されたコンフィギュレーションをインプリメントするゲートウェイ・グループを介して実現できます。用意されているドメイン・ゲートウェイは、次のとおりです。

- BEA Tuxedo Domains (TDomains) ゲートウェイ (GWTDOMAIN ゲートウェイ)
TCP/IP などのネットワーク・トランスポート・プロトコル上で機能する、特別に設計された TP プロトコルを使用して、複数の BEA Tuxedo アプリケーション間での相互運用性を実現します。
注記 GWTDOMAIN ゲートウェイを MSSQ セットのメンバとして指定することはできません。応答キューは指定しません。REPLYQ=N と指定してください。
GWTDOMAIN ゲートウェイの設定は、「再起動可能」に設定しておくことをお勧めします。
- BEA eLink OSI TP ゲートウェイ BEA Tuxedo アプリケーションと、OSI TP 標準を使用するほかのトランザクション処理アプリケーションとの間の相互運用性を実現します。OSI TP は、国際標準化機構 (ISO: International Standards Organization) によって定義された分散トランザクション処理用のプロトコルです。
- BEA eLink Adapter for Mainframe SNA ゲートウェイ BEA Tuxedo ドメイン内のクライアントとサーバ、およびリモート SNA ドメイン内の MVS/CICS または MVS/IMS 環境内のクライアントとサーバとの間の相互運用性を実現します。複数の SNA ネットワークとの通信もサポートします。
- BEA eLink Adapter for Mainframe TCP のゲートウェイ
 - BEA eLink Adapter for Mainframe TCP For CICS ゲートウェイ BEA Tuxedo 領域内の非トランザクション・タスクから CICS アプリケーション・プログラムのサービスへのアクセス、またはその逆のアクセスを実現します。また、ネットワーク・トランスポート・プロトコルである TCP/IP を使用して、BEA Tuxedo ドメインが CICS 環境と通信できるようにします。
 - BEA eLink Adapter for Mainframe TCP For IMS ゲートウェイ IMS システム内のクライアントおよびサーバ・トランザクションと、BEA Tuxedo ドメイン、CICS システム、または別の IMS システムとの間の透過的な通信を実現します。
- BEA TOP END Domain Gateway (TEDG) TOP END システムと BEA Tuxedo ドメインとの間の相互運用性を実現します。

ドメイン・ゲートウェイでサポートされる機能

ドメイン・ゲートウェイでは、次の機能がサポートされています。

- **管理** ゲートウェイは、ほかの BEA Tuxedo サーバとまったく同様に、起動したりシャットダウンできます。実行時の管理は、管理サーバの DMADM によって行われます。アプリケーション管理者は、DMADM を使用して、ドメイン・コンフィギュレーション・ファイルを変更したり、ゲートウェイ・グループの性能を調整できます。DMADM 管理サーバを起動してからゲートウェイ・グループを起動してください。
- **ATMI** ゲートウェイは、ドメインと BEA Tuxedo システムをつなぐプログラミング・インターフェイスにアクセスすることにより、次のメッセージング・モデルにアクセスできます。
 - **要求/応答モデル** BEA Tuxedo システムを使用するアプリケーション・プログラムは、ほかのドメインで動作しているアプリケーションに対してサービスを要求できます。また、リモート・アプリケーションは、ローカル・サーバに対してサービスを要求できます。このドメイン間の処理を行うため、アプリケーション・プログラムを変更する必要はありません。
 - **会話型モデル** アプリケーション・プログラムは、ほかのドメインで動作しているプログラムと会話型の通信を行うことができます。リモート・ドメインは、ローカル・サーバが提供する会話型サービスを利用して会話を行うことができます。このドメイン間の処理を行うため、アプリケーション・プログラムを変更する必要はありません。
 - **キューの処理モデル** BEA Tuxedo システムを使用するアプリケーション・プログラムは、キューにデータを保存できます。どのクライアントやサーバも、リモート・ドメインのキューにメッセージやサービス要求を保存できます。キューに保存されたすべての要求は、安全性を確保するため、トランザクション・プロトコルを使用して送信されます。このドメイン間の処理を行うため、アプリケーション・プログラムを変更する必要はありません。
- **マルチドメイン間の対話** ゲートウェイは、複数のドメインと通信することができます。
- **マルチネットワークのサポート** ゲートウェイは、TCP/IP や IPX/SPX など、さまざまなネットワーク・プロトコルを介して、ほかのドメインと通信することができます。ただし、リンク先のネットワーク・ライブラリの機能による制限を受けます。つまり、ゲートウェイは通常、1 種類のネットワークをサポートします。
- **トランザクション管理** アプリケーション・プログラムは、トランザクション内でほかのドメインと相互運用できます。ゲートウェイは、ドメイン間で行われるトランザクションのコミットまたはロールバックを調整します。

- 型付きバッファのサポート ゲートウェイは、アプリケーションで定義されるすべてのバッファ型に対して、符号化または復号化の操作を実行できます。

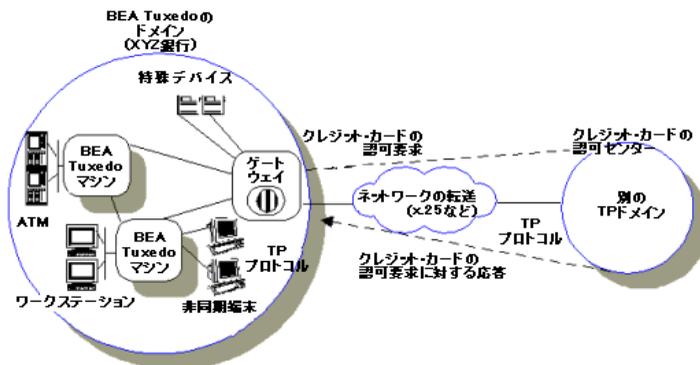
関連項目

- 『BEA Tuxedo システム入門』の第3章の43ページ「マルチ・ドメイン・コンフィギュレーション」
- 第1章の8ページ「ドメイン・ゲートウェイを使用したアプリケーションの例」
- 第1章の10ページ「ドメイン・ゲートウェイでサポートされるメッセージング・パラダイム」

ドメイン・ゲートウェイを使用したアプリケーションの例

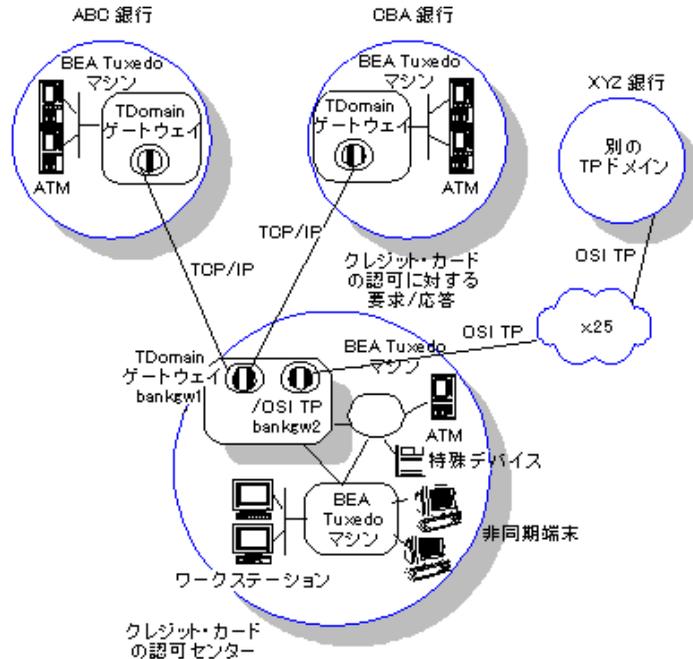
次の図は、リモート・ドメインに対してサービス(クレジット・カードの認可)を要求する BEA Tuxedo アプリケーションを示します。

図 1-3 高いレベルから見た 2 つのドメイン間での通信



アプリケーションは、リモート・ドメインからのサービス要求（残高の照会など）も受け付けます。ゲートウェイ・プロセスでは、双方向のトランザクション制御を実現でき、ほかのドメインと相互運用できるようにローカル・ドメインをコンフィギュレーションするための管理ツールを提供します。BDMCONFIG（マルチ・ドメイン・アプリケーション用のコンフィギュレーション・ファイル）では、エクスポートされたサービス、インポートされたサービス、アドレス指定、および使用されるアクセス制御リストを指定します。次の図では、Domains 環境の例をさらに詳しく示します。

図 1-4 Domains 環境の例



この例には、BEA Tuxedo システムの制御下で動作するクレジット・カードの認可センターがあり、認可センターには、bankgw1 (TCP/IP プロトコルを使用) と bankgw2 (OSI TP プロトコルを使用) という 2 つのゲートウェイ・グループがあります。bankgw1 は、2 つの BEA Tuxedo のリモート・ドメイン (ABC 銀行と CBA 銀行) へのアクセスを提供します。bankgw2 は、OSI TP プロトコルを使用して、1 つのリモート・ドメイン (XYZ 銀行) へのアクセスを提供します。

この例では、別のドメインである ABC 銀行が、クレジット・カード認可システムに対するサービス要求を生成しています。サービス要求は、bankgw1 グループ内で動作するゲートウェイによって受信されます。このゲートウェイは、ローカル・サーバが提供するクレジット・カード認可サービスに対し、リモート・ドメインの代わりにサービス要求を発行します。サーバは、要求を処理してから、応答をゲートウェイに送信します。ゲートウェイは、応答を ABC 銀行に転送します。

クレジット・カード認可センターからサービス要求を発行することもできます。たとえば、認可センターは、XYZ 銀行に残高照会を要求できます。これは、ドメインの機能、つまり、ローカル・サーバとして動作するゲートウェイを使用することにより実現できます。このゲートウェイは、ほかのドメインのサービスを、ローカル・サービスとして宣言します。

Domains には、受信した要求を制御する「ローカル・ドメイン」という概念があります。また、ドメインには、アプリケーション用の一般的なアドレス指定のフレームワークが用意されています。ローカル・ドメインは、アプリケーションの一部すなわちローカル・サービスの部分集合を、リモート・ドメインで使用できるようにします。各ローカル・ドメインは、常に 1 つのゲートウェイ・サーバ・グループによって表現されます。

ドメイン・ゲートウェイでサポートされる メッセージング・パラダイム

BEA Tuxedo のクライアント / サーバ型モデルの機能は、ドメイン・ゲートウェイに含まれる以下のメッセージング・パラダイムによってサポートされています。

- 第 1 章の 11 ページ「ローカル・サービスとリモート・サービス間での要求 / 応答型の通信」
- 第 1 章の 12 ページ「ローカル・サービスとリモート・サービス間での会話型通信」
- 第 1 章の 12 ページ「データ記憶域のキューを使用したメッセージング」

ローカル・サービスとリモート・サービス間での要求 / 応答型の通信

ドメイン・ゲートウェイは、ATMI インターフェイスで定義された要求 / 応答型のモデルをサポートしています。BEA Tuxedo アプリケーションでは、リモート・サービスを、ローカル・サービスと同じように要求できます。

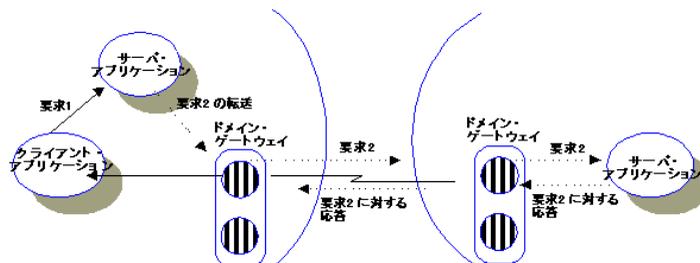
ATMI 関数のサポート

以下の BEA Tuxedo の ATMI 関数は、論理的に 1 つのアプリケーション内で使用するように制限されており、ドメイン間での使用はサポートされていません。

- `tpinit(3c)/tpterm(3c)` BEA Tuxedo アプリケーションは、リモート・ドメインの環境にはアタッチされません。リモート・ドメインへのアクセスは、ドメイン・ゲートウェイを使用して行われます。したがって、リモート・アプリケーションに `tpinit()/tpterm()` シーケンスは必要ありません。
- `tpadvertise(3c)` および `tpunadvertise(3c)` これらの関数は、ドメイン間では使用できません。ドメイン・ゲートウェイでは、ドメイン間の動的なサービスの宣言はサポートされません。
- `tpnotify(3c)` および `tpbroadcast(3c)` Domains では、これらのプリミティブによって提供される、任意通知型の通信パラダイムはサポートされません。
- イベントのポスト (`tppost(3c)`) およびイベントの通知 (`tpsubscribe(3c)`) ドメイン間では、これらの関数はサポートされません。

アプリケーションの移植性を維持するため、`tpforward(3c)` がサポートされています。転送された要求は、ドメイン・ゲートウェイにより、単純なサービス要求として解釈されます。次の図は、このプロセスを示しています。この図では、`tpforward` を使用して、リモート・サービスを要求する簡単な流れを示します。

図 1-5 `tpforward` を使用してリモート・サービスに要求を送信する



ローカル・サービスとリモート・サービス間での会話型通信

ATMI は接続指向型のインターフェイスです。このインターフェイスを使用すると、クライアントは、会話パラダイムでプログラミングされたサービスとの会話を確立し、保持することができます。

BEA Tuxedo アプリケーションは、`tpconnect(3c)` を使用してリモート・サービスとの会話を確立し、`tpsend(3c)` と `tprecv(3c)` を使用してこのサービスと通信し、`tpdiscon(3c)` を使用して会話を終了します。ドメイン・ゲートウェイは、リモート・サービスとの会話を保持し、BEA Tuxedo の会話型サービスの定義と同じ方法で戻り値 (TPSUCCESS または TPFALL を返す `tpreturn`) を返し、接続を切断します。

注記 接続指向型の ATMI 関数では、半二重会話を使用できます。会話サービスでは `tpforward(3c)` を使用できません。

会話型のリモート・サービスを設定するには、`DMCONFIG` ファイルの `DM_REMOTE_SERVICES` セクションで、`CONV=Y` と指定します。

データ記憶域のキューを使用したメッセージング

BEA Tuxedo システムでは、メッセージを永続記憶域 (ディスク) や非永続記憶域 (メモリ) に登録して、後で処理や検索を行うことができます。ATMI には、メッセージをキューに追加 (`tpenqueue`) したり、キューから読み取る (`tpdequeue`) ためのプリミティブが用意されています。応答メッセージやエラー・メッセージをキューに登録しておき、後でクライアントに返すこともできます。キューの作成、一覧表示、および変更を行うための管理コマンド・インタプリタ (`qmadmin`) も用意されています。また、メッセージをキューに登録したり、キューから取り出す要求を受け付けるサーバ (`TMQUEUE` サーバ)、キューから取り出したメッセージを処理するために転送するサーバ (`TMQFORWARD` サーバ)、およびキューの処理を伴うトランザクションを管理するサーバ (`TMS_QM` サーバ) の 3 つのサーバが用意されています。

ドメイン・ゲートウェイは、ドメイン全体に対してキューのメッセージ処理サービスをサポートします。

関連項目

- 『BEA Tuxedo システム入門』の第2章の15ページ「要求 / 応答型通信」
- 『BEA Tuxedo システム入門』の第2章の10ページ「会話型通信」
- 『BEA Tuxedo システム入門』の第2章の13ページ「キュー・ベースの通信」

型付きバッファを使用してデータをパッケージ化する

BEA Tuxedo アプリケーションでは、クライアントとサーバとの間でデータを送信するために、型付きバッファが使用されます。型付きバッファのメカニズムにより、アプリケーション・プログラムは、アプリケーションのクライアント側およびサーバ側で稼働中のマシンのデータ表現スキーマを意識せずにデータを転送することができます。

ドメイン・ゲートウェイは、異なるマシン表現を持つワークステーション、BEA Tuxedo マシン、およびリモート・ドメインから送信される要求を受信し、処理できます。型付きバッファのスイッチは、サービス要求に付属するデータを復号化します。管理者は、適切なアプリケーション用の型付きバッファのスイッチを定義する必要があります。

データ依存型ルーティングは、指定した条件に一致するデータ内のフィールドに基づいて実行されます。ただし、データが符号化されると、データの内容を判別できず、データを正しくルーティングできません。さらに、ドメイン・ゲートウェイは、次の理由により、データの内容を知る必要があります。

- ゲートウェイは、要求されたサービスに対して適切なリモート・ドメインを選択するために、データ依存型ルーティングを適用しなければならない場合があります。リモート・ドメインに対するデータ依存型ルーティングの条件は、Domains コンフィギュレーション・ファイルで定義されています。
- 各ドメインでインプリメントまたは使用されるネットワーク・プロトコルに応じて、異なるデータ形式が使用される可能性があります。

したがって、ドメイン・ゲートウェイは、符号化されたサービス要求を受信すると、常にそれを復号化しようとします。

OSI 用語では、抽象構文 (データの構造) と転送構文 (データ転送に使用する特定の符号化) を明確に区別します。各型付きバッファでは、特定のデータ構造 (抽象構文) と、そのデータ構造を特定の転送構文 (たとえば XDR) にマッピングするのに必要な符号化規則 (型付きバッファの動作) を暗黙的に定義します。

BEA Tuxedo システムには、定義済みのバッファ型 (STRING、CARRAY、FML、FML32、VIEW、VIEW32、X_C_TYPE、X_OCTET、X_COMMON、および XML) と、これらの型を XDR 転送構文にマッピングするための符号化規則が用意されています。

注記 バッファ型をカスタマイズするには、`TUXDIR/lib/tmtypesw.c` 内の `tm_typesw` 配列にインスタンスを追加し (`tuatypes(5)` と `typesw(5)` を参照)、新しいバッファ型用のルーチンを指定します (`buffer(3c)` を参照)。

関連項目

- 『BEA Tuxedo システム入門』の第 2 章の 24 ページ「型付きバッファ」
- 『C 言語を使用した BEA Tuxedo アプリケーションのプログラミング』の第 3 章の 28 ページ「バッファのカスタマイズ」
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `tuatypes(5)`
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `typesw(5)`

Domains のトランザクション・タイムアウトとブロッキング・タイムアウトを定義する

BEA Tuxedo システムには、トランザクション・タイムアウトとブロッキング・タイムアウトという 2 種類のタイムアウト・メカニズムが組み込まれています。トランザクション・タイムアウトは、サービス要求を処理するトランザクションの期間を定義するために使用します。このタイムアウト値は、`tpbegin(3c)` を呼び出して、トランザクションを開始するときに定義されます。一方、ブロッキング・タイムアウトは、個々のサービス要求の期間、つまり、1 つのサービス要求に対する応答をアプリケーションが待つ時間を定義するために使用します。

BEA Tuxedo のトランザクション・メカニズムは、Domains のフレームワークにもそのまま適用されます。ドメイン・ゲートウェイは TMS 機能をインプリメントしているため、BBL で生成される TMS_TIMEOUT メッセージの処理を要求されます。したがって、BEA Tuxedo と Domains で使用されるトランザクション・タイムアウトのメカニズムが同じであることが必要となります。

ただし、ドメイン・ゲートウェイでは、BEA Tuxedo のブロッキング・タイムアウトのメカニズムは使用できません。ブロッキング・タイムアウトのメカニズムでは、各クライアントまたは各サーバに割り当てられたレジストリ・スロット内の情報が使用されます。つまり、レジストリ・スロット内の情報がローカルの BBL で使用され、BLOCKTIME で指定された期間を過ぎてもブロックされたままになっている要求元が検出されます。一方、ドメイン・ゲートウェイは、複数のサービス要求を同時に処理できるマルチタスク・サーバであり、レジストリ・スロットのメカニズムを使用できません。したがって、ブロッキング・タイムアウトが発生すると、エラーまたは障害を通知する応答メッセージがドメイン・ゲートウェイから要求元に送信され、サービス要求に関連するあらゆるコンテキストをクリーンアップします。

Domains の接続方法を指定する

ユーザは、以下のいずれかの接続方針を選択して、ローカル・ドメイン・ゲートウェイからリモート・ドメインへの接続条件を指定できます。

- ブート時に接続する (ON_STARTUP)。
- クライアント・プログラムにより、リモート・サービスが要求されたときに接続する (ON_DEMAND)。
- 受信時に接続するが、自動的には行わない (INCOMING_ONLY)。

Dynamic Status 機能を使用してリモート・サービスの可用性を判定する

ゲートウェイ・プロセス (GWTDOMAIN) は、1 つ以上のリモート・ドメインからインポートされたサービスを掲示板で宣言します。通常、これらのサービスは、リモート・サービスが有効かどうかにかかわらず、宣言されたままの状態になります。

BEA Tuxedo ドメイン・ゲートウェイの Dynamic Status 機能では、BEA Tuxedo システムによって判定される、リモート・サービスのステータスをレポートします。

Dynamic Status が実行されると、ローカル・ゲートウェイとリモート・ゲートウェイ間のネットワーク接続のステータスによって、リモート・サービスのステータスが決定されます。リモート・サービスが存在するドメインへの接続が可能な場合、リモート・サービスは利用可能であると見なされます。一方、リモート・ドメインへのネットワーク接続が不可能な場合、そのドメインのサービスは利用不可能であると見なされます。この方針が呼び出されるのは、接続方針が ON_STARTUP (ローカル・ドメイン・ゲートウェイが起動時にリモート・ドメインへの接続を試みる時) または INCOMING_ONLY (ローカル・ドメイン・ゲートウェイが起動時にリモート・ドメインへの接続を試みない時) の場合です。

ゲートウェイは、各サービスに対して、サービスのインポート元であるリモート・ドメインを監視するほか、どのリモート・ドメインがアクセス可能であるかも監視します。この方法により、ゲートウェイは、リモート・ドメインに対する要求のロード・バランシングを実行します。サービスのインポート元であるすべてのリモート・ドメインがアクセス不可能になると、そのサービスは、掲示板で中断されます。

たとえば、コンフィギュレーション・ファイルの DM_REMOTE_SERVICES セクションを以下のように指定することにより、RSVC というサービスを2つのリモート・ドメインからインポートするとします。

```
RSVC RDOM=R1
RSVC RDOM=R2
```

R1 および R2 への接続が可能な場合、ゲートウェイは、RSVC サービスに対する要求のロード・バランシングを実行します。R1 への接続が不可能になると、ゲートウェイは、RSVC サービスへのすべての要求を R2 に送信します。R1 および R2 への接続が両方とも不可能になると、ゲートウェイは、掲示板の RSVC を中断します。RSVC に対する以降の要求は、ローカル・サービスが別のゲートウェイにルーティングされるか、または失敗して TPENOENT が返されます。

注記 接続方針が ON_DEMAND の場合は、クライアントがリモート・サービスを要求したとき、または管理コマンド connect が実行されたとき、のいずれかの場合に接続が試行されます。

接続方針と Dynamic Status 機能の関係

Dynamic Status 機能は、すべての Domains コンフィギュレーションで使用できるわけではありません。使用できるかどうかは、ドメイン間の接続方針によって決まります。次の表では、それぞれの接続方針と Dynamic Status 機能の関係について説明します。

表 1-1 接続方針と Dynamic Status 機能の関係

接続方針	Dynamic Status
ON_STARTUP	<p>オン</p> <p>リモート・ドメインからインポートされたサービスは、そのリモート・ドメインへの接続がアクティブである間は宣言されています。接続の確立方法は、次のとおりです。</p> <ul style="list-style-type: none"> ■ 自動接続 ■ 手動による <code>dmadmin</code> コマンドの実行 ■ 受信時接続
ON_DEMAND	<p>オフ</p> <p>リモート・ドメインからインポートされたサービスは、継続的に宣言されています。接続の確立方法は、次のとおりです。</p> <ul style="list-style-type: none"> ■ クライアント要求 ■ 手動による <code>dmadmin</code> コマンドの実行 ■ 受信時接続
INCOMING_ONLY	<p>オン</p> <p>初期状態では、リモート・サービスは中断されています。ドメイン・ゲートウェイは、リモート・ドメインからの受信時接続に対して使用できます。リモート・サービスは、ローカル・ドメイン・ゲートウェイが受信時接続を確立したとき、または、<code>connect</code> コマンドを手動で実行したときに宣言されます。接続の確立方法は、次のとおりです。</p> <ul style="list-style-type: none"> ■ 手動による <code>dmadmin</code> コマンドの実行 ■ 受信時接続

Domains コンフィギュレーション・ファイルとは

ドメインに関するすべてのコンフィギュレーション情報は、BDMCONFIG というバイナリ形式のファイルに格納されています。このファイルのテキスト版である DMCONFIG は、任意のテキスト・エディタで作成および編集できます。Domains の使用中に、`dmadmin(1)` コマンドを実行すると、システムの稼働中にコンパイル済みの BDMCONFIG ファイルを更新することができます。マルチ・ドメイン・アプリケーションでは、参加しているドメインごとに別々の BDMCONFIG ファイルを作成する必要があります。

BDMCONFIG ファイルへのアクセスは、Domains の管理サーバ DMADM(5) にアクセスすることによって実現できます。ゲートウェイ・グループが起動すると、ゲートウェイ管理サーバ GWADM(5) が、そのグループに必要なコンフィギュレーション・ファイルのコピーを DMADM サーバに要求します。GWADM サーバおよび DMADM サーバは、コンフィギュレーションに対する実行時の変更が、対応するドメイン・ゲートウェイ・グループに反映されるようにします。

DMCONFIG ファイルのセクションの記述

次の表は、DMCONFIG ファイルの各セクションの説明です。

表 1-2 DCONFIG ファイルのセクション

セクション	目的
DM_LOCAL_DOMAINS	特定のドメイン・ゲートウェイ・グループの環境を記述します。このセクションでは、論理アプリケーション名の LDOM を、リモート・ドメインからアクセスできるローカル・サービスの部分集合に割り当てます。このセクションにある複数のエントリを使用すると、1つの BEA Tuxedo アプリケーションに複数のゲートウェイ・グループを定義できます。各ゲートウェイ・グループでは、異なるドメインに対するアクセスを実現できます。
DM_REMOTE_DOMAINS	この Domains アプリケーションのクライアントとサーバがアクセスできるリモート・ドメインを指定します。

セクション	目的
DM_LOCAL_SERVICES	DM_LOCAL_DOMAINS セクションのローカル・ドメイン (LDM) によって提供されるローカル・サービスを記述します。サービスの指定内容により、リモート・ドメインからローカル・サービスに対するアクセスを制限することもできます。つまり、リモート・ドメインは、指定されたサービスだけにアクセスできます。
DM_REMOTE_SERVICES	リモート・ドメインによって提供されるサービスを記述します。また、(LDM パラメータを使用して、リモート・サービスへのアクセスを実現するローカル・ゲートウェイ・グループを命名します。
DM_ROUTING	データ依存型ルーティングの基準を指定します。ゲートウェイは、この基準を使用して、サービス要求を特定のリモート・ドメインに転送します。
DM_ACCESS_CONTROL	アクセス制御リストを指定します。このリストには、ローカル・サービスの要求を許可されたリモート・ドメイン (RDM) が一覧表示されています。DM_LOCAL_SERVICES セクションの ACL パラメータを使用するには、ACL= <i>name_of_list</i> を設定して、特定のローカル・サービスに対するアクセス許可を、アクセス制御リストに登録されたリモート・ドメインだけに制限します。
DM_dmtyp	特定の Domains コンフィギュレーションに必要なパラメータを定義します。現在、 <i>dmtyp</i> に指定できる値は、OSITP、SNAX、TOPEND、または TDOMAIN です。ここでは、TDOMAIN のみを扱います。OSITP および SNAX については、BEA eLink for Mainframe のマニュアルを参照してください。TOPEND については、『Using the BEA Tuxedo TOP END Domain Gateway』を参照してください。各ドメイン・タイプは、別々のセクションで指定する必要があります。 DM_TDOMAIN のセクションには、リモート・ドメインに関連するエントリを異なるネットワーク・アドレスを使って複数指定し、「ミラーリングした」ゲートウェイをインプリメントできます。詳細および「ミラーリングした」ゲートウェイの例については、DMCONFIG(5) を参照してください。

Domains 関連の新しい用語

BEA Tuxedo のリリース 7.1 以降では、Domains 用の MIB でローカル・ドメインとリモート・ドメインとの相互作用を記述するため、クラスと属性の用語が改善されています。改善された用語は、DM_MIB のクラス、リファレンス・ページ、エラー・メッセージ、DMCONFIG ファイルの構文、および各種の DMCNFIG エラー・メッセージに適用されました。

旧バージョンとの互換性のため、BEA Tuxedo リリース 7.1 より前に使用された DMCNFIG 用語と、改善された Domains 用の MIB 用語との間にエイリアスが提供されています。BEA Tuxedo リリース 7.1 以降の DMCNFIG では、両方のバージョンの用語を使用できます。詳細については、DM_MIB(5) リファレンス・ページの「Domains 関連の新しい用語」を参照してください。

関連項目

- 第 2 章の 17 ページ「Domains 環境を設定する」
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の DMCNFIG(5)
- 第 1 章の 20 ページ「Domains コンフィギュレーション・ファイルを変換する」

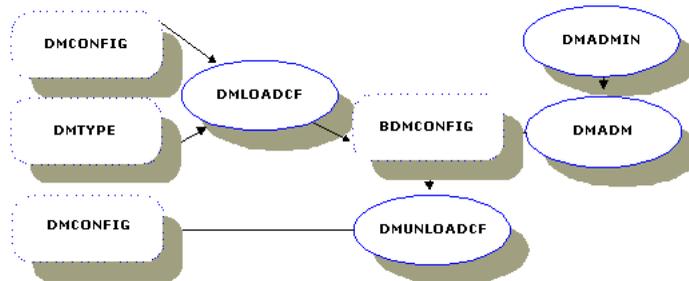
Domains コンフィギュレーション・ファイルを変換する

このセクションでは、テキスト形式の Domains コンフィギュレーション・ファイル (DMCNFIG) をバイナリ形式 (BDMCNFIG) に変換する方法、およびその逆の変換を行う方法について説明します。

DMCONFIG ファイルをバイナリ形式に変換する

dmloadcf(1) コマンドを実行すると、DMCONFIG (テキスト・ファイル) が解析され、Domains のコンフィギュレーション情報が BDMCONFIG というバイナリ・ファイルにロードされます。このコマンドは、環境変数 BDMCONFIG を使用して、コンフィギュレーションが格納されているディレクトリを示します。BDMCONFIG ファイルは、TUXCONFIG ファイル (バイナリ形式の UBBCONFIG ファイル) と同じデバイスに格納できます。

図 1-6 コンフィギュレーション用のコマンドとコンフィギュレーション・ファイルの関係



dmloadcf(1) コマンドに `-c` オプションを指定して実行すると、コンフィギュレーションで指定された各ローカル・ドメインに必要な IPC 資源の量を計算できます。

前の図に示すように、dmloadcf コマンドは、`$TUXDIR/udataobj/DMTYPE` ファイルを使用します。このコマンドは、DMTYPE ファイルをチェックして、コンフィギュレーション・ファイルで指定されたドメイン・タイプが有効であるかどうかを検証します。各 Domains のインスタンスには、ドメインの `type` が指定されています。type は、`TUXDIR/udataobj/DMTYPE` ファイル内でタグとして使用されます。ファイル内の各行の形式は、次のとおりです。

```
dmtype:access_module_lib:comm_libs:tm_typesw_lib:gw_typesw_lib
```

このファイルには、次のような TDOMAIN 用のエントリがあります。

```
TDOMAIN:-lgwt:-lnwi -lnws -lnwi::
```

BDMCONFIG ファイルをテキスト形式に変換する

バイナリ形式の Domains コンフィギュレーション・ファイルをアンロードする (テキスト形式に変換する) には、`dmunloadcf(1)` コマンドを実行します。

BEA Tuxedo システムの Domains の特徴

- **エイリアス機能** ローカル・アプリケーションとリモート・アプリケーションの間でマッピング・サービス名を定義できます。この結果、命名方法が異なる複数のアプリケーションを簡単に統合できます。
- **可用性** 障害に対応するため、代替サーバを指定しておくことができます。
- **スケーラビリティとモジュール化の促進** プログラマは、モジュール化を実現し、障害から切り離し、非依存的に拡張できるようなアプリケーションを構築できます。リモート・アプリケーションで使用されるサービスの記述を Domains のコンフィギュレーションに追加すると、ほかのトランザクション処理アプリケーションとの相互運用を簡単に実現することができます。
- **セキュリティ機能** 特定のリモート・ドメインからローカル・リソースへのアクセスを制限するための、アクセス制御リスト (ACL: Access Control List) 機能が用意されています。Domains には、暗号やパスワードによるセキュリティ機能も用意されています。
- **透過性と独立性** サービスがどのように分散されるかについて、アプリケーション・プログラマ側で認識しておく必要はありません。サービスは、クライアントと同じマシン、ローカル・ドメイン内のほかのマシン、またはリモート・ドメインで利用できます。クライアント・アプリケーションのプログラマは、サービスに対するインプリメンテーションの変更、サービスの位置、ネットワーク・アドレスなどを知っておく必要はありません。
- **トランザクション管理と信頼性** Domains 機能は、BEA Tuxedo のトランザクション管理機能に統合されています。

2 ATMI Domains の計画とコンフィギュレーション

この章では、BEA Tuxedo ATMI 環境での Domains の計画方法と設定方法を説明します。BEA Tuxedo CORBA 環境での Domains の設定方法については、第 3 章の 1 ページ「CORBA Domains の計画とコンフィギュレーション」を参照してください。

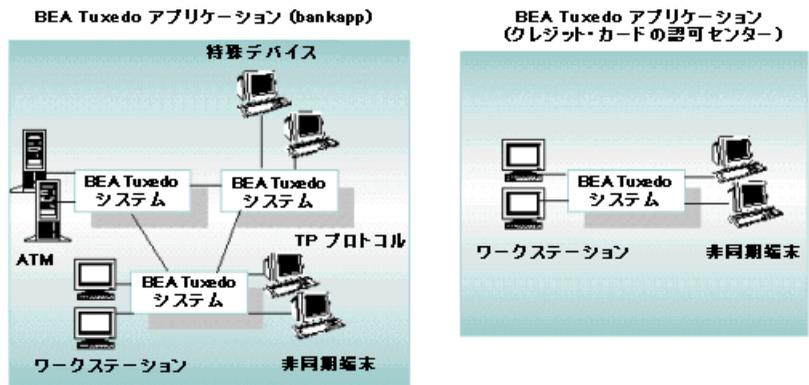
ここでは、次の内容について説明します。

- 複数の BEA Tuxedo アプリケーションで Domains を構成する
- Domains のサンプル・アプリケーション : creditapp
- Domains 環境を設定する
- Domains 間でデータを圧縮する
- サービス要求をリモート・ドメインにルーティングする
- Domains でセキュリティを設定する
- ドメイン間の接続を設定する
- Domains 環境でのフェイルオーバーとフェイルバックを設定する

複数の BEA Tuxedo アプリケーションで Domains を構成する

ある銀行が、次の図のように、2 つの BEA Tuxedo アプリケーション (bankapp およびクレジット・カード認可センター) を開発したとします。

図 2-1 2 つの BEA Tuxedo アプリケーション



bankapp アプリケーションは、各支店にある ATM を本店に接続し、クレジット・カード認可センターは、クレジット・カード用のアプリケーションを処理します。しばらくして、この銀行が、より良い顧客サービス、つまり、bankapp アプリケーションがクレジット・カードの認可センターと直接通信できれば、口座の開設時にその場でクレジット・カードを発行できることに気づいたとします。

bankapp は、BEA Tuxedo ソフトウェアに同梱されているサンプル・アプリケーションです。クレジット・カードの認可用アプリケーションは、bankapp を拡張した架空のアプリケーションです。

bankapp をマルチマシン・アプリケーションとしてインプリメントする方法については、`TUXDIR/apps/bankapp/ubbmp` にあるコンフィギュレーション・ファイルのサンプル・コードを参照してください。

この場合、次のいずれかのオプションを使用できます。

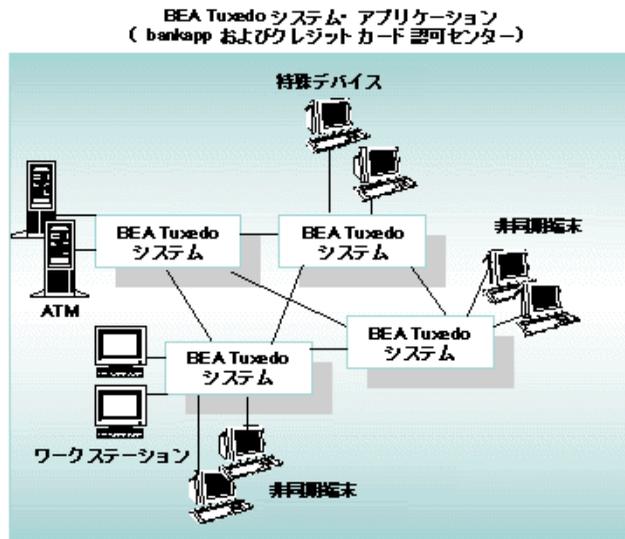
- 第 2 章の 3 ページ「オプション 1: アプリケーションを再コンフィギュレーションする」

- 第 2 章の 8 ページ「オプション 2: アプリケーションを別々の BEA Tuxedo Domains として再定義する」

オプション 1: アプリケーションを再コンフィギュレーションする

1 つ目の方法として、次の図に示すように、2 つの BEA Tuxedo アプリケーションを 1 つに統合することができます。

図 2-2 2 つの BEA Tuxedo システム・アプリケーションを統合する



2 つのアプリケーションを 1 つに統合する場合は、以下の変更が行われます。

- `OPTION=LAN` が指定され、`NETWORK` セクションが組み込まれます。
- サーバの移行を有効にするには、`OPTION=MIGRATE` を指定します。この場合、バックアップ・マスタ・サイトが定義されます。
- ゲートウェイ・サーバが、3 つのサーバ (TLRA、ACCTA、および CRDT) として再定義されます。
- クレジット認可サービスが追加されます。

アプリケーションの統合用のコンフィギュレーション・ファイル

次のリストは、統合後のコンフィギュレーション・ファイルの例です。

リスト 2-1 統合されたアプリケーションのコンフィギュレーション・ファイルの例

```
*RESOURCES
IPCKEY          76666
UID             0000
GID             000
PERM            0660
MAXACCESSERS   40
MAXSERVERS     35
MAXSERVICES    75
MAXCONV        10
MASTER        SITE1,SITE2
SCANUNIT       10
MODEL          MP
LDBAL          Y
OPTIONS        LAN,MIGRATE
MAXGTT         100
MAXBUFTYPE    16
SCANUNIT       10
SANITYSCAN     5
DBBLWAIT       6
BBLQUERY       50
BLOCKTIME      2
#
#
*MACHINES
#
mach1 LMID=SITE1
        TUXDIR="/home/mylogin/tuxroot"
        APPDIR="/home/mylogin/bankapp"
        ENVFILE="/home/mylogin/bankapp/ENVFILE"
        TLOGDEVICE="/home/mylogin/bankapp/TLOG"
        TLOGNAME=TLOG
        TUXCONFIG="/home/mylogin/bankapp/tuxconfig"
        ULOGPFX="/home/mylogin/bankapp/ULOG"
        TYPE="type1"
#
mach2 LMID=SITE2
        TUXDIR="/home/mylogin/tuxroot"
        APPDIR="/home/mylogin/bankapp"
```

```

ENVFILE="/home/mylogin/bankapp/ENVFILE"
TLOGDEVICE="/home/mylogin/bankapp/TLOG"
TLOGNAME=TLOG
TUXCONFIG="/home/mylogin/bankapp/tuxconfig"
ULOGPFX="/home/mylogin/bankapp/ULOG"
TYPE="type2"

#
mach3 LMID=SITE3
TUXDIR="/home/mylogin/tuxroot"
APPDIR="/home/mylogin/bankapp"
ENVFILE="/home/mylogin/bankapp/ENVFILE"
TLOGDEVICE="/home/mylogin/bankapp/TLOG"
TLOGNAME=TLOG
TUXCONFIG="/home/mylogin/bankapp/tuxconfig"
ULOGPFX="/home/mylogin/bankapp/ULOG"
TYPE="type2"

#
mach4 LMID=SITE4
TUXDIR="/home/mylogin/tuxroot"
APPDIR="/home/mylogin/bankapp"
ENVFILE="/home/mylogin/bankapp/ENVFILE"
TLOGDEVICE="/home/mylogin/bankapp/TLOG"
TLOGNAME=TLOG
TUXCONFIG="/home/mylogin/bankapp/tuxconfig"
ULOGPFX="/home/mylogin/bankapp/ULOG"
TYPE="type1"

#
*GROUPS
#
DEFAULT:  TMSNAME=TMS_SQL      TMSCOUNT=2
BANKB1    LMID=SITE1           GRPNO=1
OPENINFO="TUXEDO/SQL:/home/mylogin/bankapp/bankd11:bankdb:readwrite"
BANKB2    LMID=SITE2           GRPNO=2
OPENINFO="TUXEDO/SQL:/home/mylogin/bankapp/bankd12:bankdb:readwrite"
BANKB3    LMID=SITE3           GRPNO=3
OPENINFO="TUXEDO/SQL:/home/mylogin/bankapp/bankd13:bankdb:readwrite"
BANKB4    LMID=SITE4           GRPNO=4
OPENINFO="TUXEDO/SQL:/home/mylogin/bankapp/bankd14:bankdb:readwrite"
#
#
*NETWORK
#
SITE1     NADDR="<network address of SITE1>"
          BRIDGE="<device of provider1>"
          NLSADDR="<network listener address of SITE1>"
SITE2     NADDR="<network address of SITE2>"
          BRIDGE="<device of provider2>"
          NLSADDR="<network listener address of SITE2>"

```

2 ATMI Domains の計画とコンフィギュレーション

```
SITE3      NADDR="<network address of SITE3>"
           BRIDGE="<device of provider3>"
           NLSADDR="<network listener address of SITE3>"
SITE4      NADDR="<network address of SITE4>"
           BRIDGE="<device of provider4>"
           NLSADDR="<network listener address of SITE4>"

#
*SERVERS
#
DEFAULT:  RESTART=Y MAXGEN=5 REPLYQ=Y CLOPT="-A"
# bankapp 用のサーバ
TLR       SRVGRP=BANKB1   SRVID=2
TLR       SRVGRP=BANKB2   SRVID=3           RQADDR=tlr2           CLOPT="-A -- -T 600"
TLR       SRVGRP=BANKB3   SRVID=4
XFER      SRVGRP=BANKB1   SRVID=10
XFER      SRVGRP=BANKB2   SRVID=6
XFER      SRVGRP=BANKB3   SRVID=8
ACCT      SRVGRP=BANKB1   SRVID=11
ACCT      SRVGRP=BANKB2   SRVID=7
ACCT      SRVGRP=BANKB3   SRVID=13
BTADD     SRVGRP=BANKB1   SRVID=12
BTADD     SRVGRP=BANKB2   SRVID=14
BTADD     SRVGRP=BANKB3   SRVID=16
# クレジット・カード認可センター用のサーバ
TLRA      SRVGRP=BANKB4   SRVID=5           CLOPT="-A -- -T 600"
ACCTA     SRVGRP=BANKB4   SRVID=9
CRDT      SRVGRP=BANKB4   SRVID=15
#
#
*SERVICES
#
DEFAULT:  LOAD=50          AUTOTRAN=N
# bankapp 用のサービス
BR_ADD    PRIO=20          ROUTING=BRANCH_ID
TLR_ADD   PRIO=20          ROUTING=BRANCH_ID
WITHDRAWAL PRIO=50          ROUTING=ACCOUNT_ID
DEPOSIT   PRIO=50          ROUTING=ACCOUNT_ID
TRANSFER  PRIO=50          ROUTING=ACCOUNT_ID
INQUIRY   PRIO=50          ROUTING=ACCOUNT_ID
CLOSE_ACCT PRIO=40          ROUTING=ACCOUNT_ID
OPEN_ACCT PRIO=40          ROUTING=BRANCH_ID
# クレジット・カード認可センター用のサービス
WITHDRAWALA PRIO=50
INQUIRYA    PRIO=50
OPENCA      PRIO=40
CLOSECA     PRIO=40
DEPOSITA    PRIO=50
OPEN_ACCT2  PRIO=40
```

```
OPENC                PRIO=40

#
#
*ROUTING
#
ACCOUNT_ID           FIELD=ACCOUNT_ID
                     BUFTYPE="FML"
                     RANGES="10000-39999:BANKB1,
                               40000-69999:BANKB2,
                               70000-109999:BANKB3,
                               *:*"

BRANCH_ID            FIELD=BRANCH_ID
                     BUFTYPE="FML"
                     RANGES="1-3:BANKB1,
                               4-6:BANKB2,
                               7-10:BANKB3,
                               *:*"

#
```

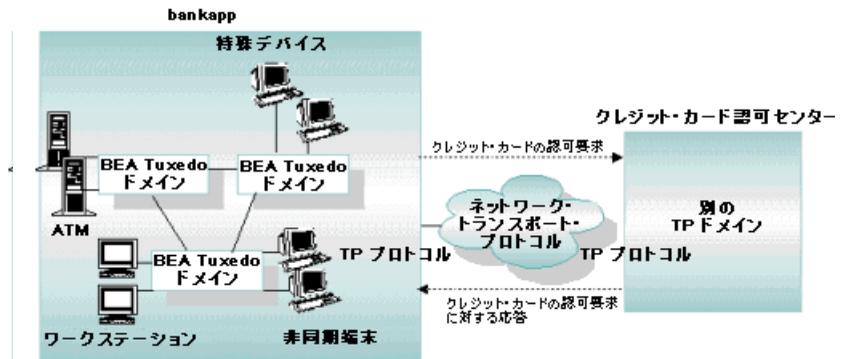
オプション 1 の制限事項

- 小規模なアプリケーションを 2 つ管理するより、大規模なアプリケーションを 1 つ管理する方が、手間がかかる場合があります。前者の場合は、2 つのアプリケーションをそれぞれ別の管理インターフェイスで管理できます。
- ネットワーク接続されたアプリケーションの場合、すべてのサーバを起動するのに時間がかかり、ネットワーク全体に掲示板を複製転送する必要も生じます。アプリケーションが複数のより小さい単位に分かれていれば、これらを同時に起動することができます。

オプション 2: アプリケーションを別々の BEA Tuxedo Domains として再定義する

次の図では、統合されたアプリケーションを 4 つの BEA Tuxedo ドメイン (TDomains) に再コンフィギュレーションした結果を示しています。4 つのうち、3 つのドメインは左側にあります。

図 2-3 Domains のコンフィギュレーション



コンフィギュレーション・ファイルを変更する

統合したアプリケーションを TDomain として再構成するには、UBBCONFIG ファイルを次のように変更します。

- MODEL を SHM に変更します。
- NETWORK セクションを削除します。
注記 複数のドメインを使用する環境でも、必要に応じて MP モードを使用し、NETWORK セクションを記述することができます。
- ドメイン固有のサーバ (DMADM、GWADM、GWTDOMAIN など) を SERVERS セクションに追加します。

次のリストは、変更後の UBBCONFIG ファイルの例です。

リスト 2-2 変更後の UBBCONFIG ファイル

```
*RESOURCES
IPCKEY          76666
UID             7901
GID             601
PERM            0660
MAXACCESSERS   40
MAXSERVERS     35
MAXSERVICES    75
MAXCONV        10
MASTER        SITE1
SCANUNIT       10
MODEL          SHM
LDBAL          Y
MAXGTT         100
MAXBUFTYPE    16
SCANUNIT       10
SANITYSCAN     5
BBLQUERY       50
BLOCKTIME     2
#
*MACHINES
sfexpz         LMID=SITE1
               TUXDIR="/home/mylogin/tuxroot"
               APPDIR="/home/mylogin/creditapp"
               ENVFILE="/home/mylogin/creditapp/ENVFILE"
               TLOGDEVICE="/home/mylogin/creditapp/TLOG"
               TLOGNAME=TLOG
               TUXCONFIG="/home/mylogin/creditapp/tuxconfig"
               ULOGPFX="/home/mylogin/creditapp/ULOG"
               TYPE="type1"

#
#
#
*GROUPS
DEFAULT:      LMID=SITE1
BANKB1        GRPNO=1  TMSNAME=TMS_SQL          TMSCOUNT=2
OPENINFO="TUXEDO/SQL:/home/mylogin/creditapp/crtdll1:bankdb:readwrite"
BANKB2        GRPNO=2
BANKB3        GRPNO=3
BANKB4        GRPNO=4
DMADMGRP     LMID=mach1 GRPNO=5
#
#
```

2 ATMI Domains の計画とコンフィギュレーション

```
#
*SERVERS
#
DEFAULT: RESTART=Y MAXGEN=5 REPLYQ=Y CLOPT="-A"
GWADM          SRVGRP=BANKB2      SRVID=30
                REPLYQ = N RESTART = Y GRACE = 0
GWTDOMAIN      SRVGRP=BANKB2      SRVID=31
                REPLYQ = N RESTART = Y GRACE = 0
GWADM          SRVGRP=BANKB3      SRVID=24
                REPLYQ = N RESTART = Y GRACE = 0
GWTDOMAIN      SRVGRP=BANKB3      SRVID=25
                REPLYQ = N RESTART = Y GRACE = 0
GWADM          SRVGRP=BANKB4      SRVID=20
                REPLYQ = N RESTART = Y GRACE = 0
GWTDOMAIN      SRVGRP=BANKB4      SRVID=21
                REPLYQ = N RESTART = Y GRACE = 0
DMADM          SRVGRP="DMADMGRP"   SRVID=50
                REPLYQ = N RESTART = Y GRACE = 0
TLRA           SRVGRP=BANKB1      SRVID=2      CLOPT="-A -- -T 100"
BTADD          SRVGRP=BANKB1      SRVID=3
ACCTA          SRVGRP=BANKB1      SRVID=4
CRDT           SRVGRP=BANKB1      SRVID=5
CRDTA          SRVGRP=BANKB1      SRVID=6
*SERVICES
DEFAULT:       LOAD=50
INQUIRYA      PRIO=50
WITHDRAWALA   PRIO=50
OPEN_ACCT2    PRIO=40
OPENC         PRIO=40
OPENCA        PRIO=40
CLOSECA       PRIO=40
BR_ADD        PRIO=20
TLR_ADD       PRIO=20
```

DMCONFIG ファイルを追加する

さらに、次のような DMCNFIG ファイルを 4 つ作成する必要があります。

リスト 2-3 DMCNFIG ファイルの例

```
#
#
*DM_LOCAL_DOMAINS
#
#
QDOM1  GWGRP=BANKB2
        TYPE=TDOMAIN
        DOMAINID=QDOM1
        BLOCKTIME=10
        MAXDATALEN=56
        MAXRDOM=89
        DMTLOGDEV= "/home/mylogin/creditapp/DMTLOG"
        AUDITLOG= "/home/mylogin/creditapp/AUDITLOG"

QDOM2  GWGRP=BANKB3
        TYPE=TDOMAIN
        DOMAINID=QDOM2
        BLOCKTIME=10
        MAXDATALEN=56
        MAXRDOM=89
        DMTLOGDEV= "/home/mylogin/creditapp/DMTLOG"
        AUDITLOG= "/home/mylogin/creditapp/AUDITLOG"
        DMTLOGNAME= "DMTLOG_TDOM2"

QDOM3  GWGRP=BANKB4
        TYPE=TDOMAIN
        DOMAINID=QDOM3
        BLOCKTIME=10
        MAXDATALEN=56
        MAXRDOM=89
        DMTLOGDEV= "/home/mylogin/creditapp/DMTLOG"
        AUDITLOG= "/home/mylogin/creditapp/AUDITLOG"
        DMTLOGNAME= "DMTLOG_TDOM3"

#
*DM_REMOTE_DOMAINS
#
#
TDOM1  TYPE=TDOMAIN
        DOMAINID=TDOM1

TDOM2  TYPE=TDOMAIN
        DOMAINID=TDOM2
```

```
TDOM3    TYPE=TDOMAIN
          DOMAINID=TDOM3
#
#
*DM_TDOMAIN
#
QDOM1    NWADDR="0x0002DEEF93026927"
          NWDEVICE="/dev/tcp"
QDOM2    NWADDR="0x0002BEEF93026927"
          NWDEVICE="/dev/tcp"
QDOM3    NWADDR="0x0002CEEF93026927"
          NWDEVICE="/dev/tcp"
TDOM1    NWADDR="0x0002DEEF93026947"
          NWDEVICE="/dev/null"
TDOM2    NWADDR="0x0002BEEF9302691D"
          NWDEVICE="/dev/tcp"
TDOM3    NWADDR="0x0002CEEF9302690E"
          NWDEVICE="/dev/tcp"
#
#
#
*DM_LOCAL_SERVICES
#
#
WITHDRAWALA
INQUIRYA
OPENCA
CLOSECA
```

Domains のサンプル・アプリケーション： creditapp

BEA Tuxedo システムには、creditapp というサンプル・アプリケーションが同梱されています。creditapp は、前に説明した例 (bankapp とクレジット・カードの認可用アプリケーションを別々のドメインにする例) で取り上げた、実行可能な架空のアプリケーションです。

このアプリケーションは、*TUXDIR/apps/creditapp* にあり、以下のファイルが含まれています。

リスト 2-4 creditapp のファイル

ACCT.ec	ACCTA.ec	AUDITC.c	BAL.ec	BALANCE.m
BALANCEA.m	BALC.ec	BTADD.ec	CBALANCE.m	CCLOSE.m
CDEPOSIT.m	CLOSE.m	COPEN.m	CRDT.ec	CRDTA.ec
CRMENU.m	CRMENU2.m	CTRANSFER.m	CWITHDRAW.m	DEPOSIT.m
DEPOSITA.m	FILES	HCBALANCE.m	HCCLOSE.m	HCCLOSE.m
HCOOPEN.m	HCWITHDRAW.m	HELP.m	HOPEN.m	OPEN.m
README	RUNME	RUNME.sh	SETUP.sh	TLR.ec
TLR1.ec	TLR2.ec	TLR3.ec	TLRA.ec	TRANSFER.m
WITHDRAW.m	WITHDRAWA.m	XFER.c	appinit.c	aud.h
aud.v	audit.c	auditcon.c	bank.flds	bank.flds.h
bank.h	cleanup.sh	crbank.sh	crbankdb.sh	crdt_app.mk
crdt_app2.mk	crdt_app3.mk	crdt_app4.mk	crdt_flds.h	crdtvar
crdtvar2	credit.flds	crtlog.sh	crtlog2	crtlog2.sh
domcon1	domcon2	domcon3	domcon4	driver.sh
envfile.sh	gendata.c	gentran.c	hostmk	listnr
populate.sh	run.sh	setenv	ubbdom1	ubbdom2
ubbdom3	ubbdom4	util.c		

creditapp の README ファイル

creditapp ディレクトリには、次の README ファイルが含まれています。README ファイルには、creditapp のインストールと実行を行うためのスクリプトが記述されています。また、元のスクリプトにはない情報もいくつか追加されています。

リスト 2-5 creditapp の README ファイル

アプリケーションの簡単な作成方法

creditapp は、bankapp アプリケーションと hostapp アプリケーションを拡張したアプリケーションです。

creditapp アプリケーションは、4 つのドメインで構成するように設計されています。したがって、このソフトウェアは、4 つのマシンにインストールする必要があります。RUNME.sh スクリプトに必要な手順が説明されています。

ステップ 1: creditapp のソフトウェアをコピーします。

\$HOME ディレクトリの下に新しいディレクトリを作成し、そのディレクトリに <TUXDIR>/apps/creditapp のすべてのソース・ファイルをコピーします。TUXDIR は、BEA Tuxedo システムのインストール先である

ルート・ディレクトリです。新しいディレクトリを `$HOME /creditapp` とするとします。この手順の残りのステップは、`$HOME/creditapp` ディレクトリで行われます。

ステップ 2: 残り 3 つのマシンに対して、次の作業を行います。

アプリケーション用に使用できるディレクトリに、`creditapp` ディレクトリを作成します。

このディレクトリを `$HOME/creditapp` と指定するとします。

各マシンの `$HOME/creditapp` と `TUXDIR` の絶対パスを書き留めておきます。これらの絶対パスは、`RUNME.sh` スクリプトで必要になります。

ステップ 3: 「マスタ・サイト」で、「`RUNME.sh`」スクリプトを実行します。

シェル・スクリプト「`RUNME.sh`」は、

4 ドメイン構成の `creditapp` アプリケーションを初期化、起動、シャットダウン、およびクリーンアップするために設計された対話型のプログラムです。シェルは対話型であり、コマンド行引数は必要ありません。ディレクトリ内で必要なのは、ステップ 1 で `TUXDIR/apps/creditapp` ディレクトリからコピーしたソースのみです。

環境変数 `RSH` と `RCP` の値を入力するか、またはデフォルト値をそのまま指定するかを尋ねるメッセージが表示されます。

`RSH` と `RCP` の値は、必要なスクリプトをリモートにコピーしたり、実行するときに使用されます。したがって、これらの値を入力しておくことは重要です。

以下は、重要な環境変数です。スクリプトにより、環境から `TUXDIR` と `APPDIR` の値が取り出され、(オプション 4 で) `BLKSIZE` の入力が必要になります。

<code>TUXDIR</code>	このソフトウェアのインストール先である、BEA Tuxedo システムのルート・ディレクトリ。
<code>APPDIR</code>	<code>creditapp</code> アプリケーションが置かれているディレクトリ。 <code>crdtvar.dml</code> では、デフォルトでこのディレクトリが作業ディレクトリに設定されます。 そのため、最初に示したとおり、 <code>\$HOME/creditapp</code> を作業ディレクトリとして使用できます。これは、ステップ 1 で <code>creditapp</code> ファイルのコピー先に指定したディレクトリです。
<code>BLKSIZE</code>	データベースの論理ブロック・サイズ (バイト単位)。 コンピュータの物理的ページ・サイズの整数倍でなければなりません (512 バイト、4096 バイトなど)。

`RUNME.sh` を起動すると、メニューが表示され、10 個のオプション (11 個目は「終了」オプション) が表示されます。
オプションの種類は、以下のとおりです。

- 1) コンフィギュレーション・ファイルおよび `makefile` を初期化する
- 2) ファイルをリモート・サイトにコピーする
- 3) `crdtapp` のクライアントとサーバを作成する
- 4) データベースを作成する
- 5) バイナリ形式の `tuxconfig` ファイルと `bdmconfig` ファイルを生成する
- 6) トランザクション・ログ・ファイルを作成する
- 7) アプリケーションを起動する
- 8) データベースにデータを追加する
- 9) アプリケーションをシャットダウンする
- 10) IPC 資源、データベース・ファイル、およびログ・ファイルをクリーンアップする
- q) 終了

サンプル・アプリケーションの作成と実行に関するすべてのプロセスを行うには、1 番のオプションから開始してください。

1 つのオプションが終了すると、次のオプションが表示されます。

オプション 1 コンフィギュレーション・ファイルおよび `makefile` を初期化します。
このオプションでは、アプリケーションに必要な `makefile`、`UBBCONFIG` ファイル、および `DMCONFIG` ファイルを設定します。

質問には必ず答えてください。

システム名を入力してください。使用しているマシンの `uname` を、
現在使用しているマシンから順に入力します。

各マシンの `TUXDIR` を入力してください。

各マシンの `APPDIR` を入力してください。

以降の質問にもすべて答えてください。

4 つの 16 進数の例は、「`beef`、`cfff`、`6774`、`aeef`」です。

NOTE: 各マシンには、一意の 16 進シーケンスが必要です。

オプション 2 ファイルをコンフィギュレーション内のほかのドメインにコピーします。

オプション 3 すべてのマシンにクライアントとサーバを作成します。

注) 各サイトでクライアントとサーバが正しく作成されたことを
確認してください。必要に応じて、各自でクライアントとサーバを作成してください。

特定のサイトで次のように入力します。

```
nohup make -f CRDT{$MACH}.mk2
```

`${MACH}` は、作成先のマシンの `uname` です。
次に例を示します。

```
nohup make -f CRDTtux1.mk2
```

オプション 4 各サイトでデータベースを作成します。

注) 各サイトの

```
crdt${MACH}.dm1 (プライマリ・サイトの場合) または  
crdt${MACH}.dm2 (リモート・サイト) で、ブロック・サイズの値が  
正しく設定されていることを確認してください。${MACH} は  
作成先のマシンの uname です。
```

オプション 5 `tuxconfig` ファイルおよび `bdmconfig` ファイルを生成します。

その他のオプションは、`bankapp` の場合と同様です。

オプション 8 データベースにデータを追加します。

メニューを終了するには、`q` を入力します。

CREDITAPP の実行

各マシンにはスクリプト `run.sh` が用意されています。

`run.sh` を実行します。

```
run
```

次のメッセージが表示されます。

```
Is this machine the Credit Card Authorization Center(y/n)?
```

プライマリ・マシンの場合は `y` を入力します。

それ以外のマシンの場合は `n` を入力します。

プライマリ・マシンでは、ほかの 3 つのマシンとは異なるメニューが表示されます。

すべての口座はプライマリ・マシンに置かれています。
これらの口座には、どのマシンからもアクセスできます。

```
口座 10000000 ~ 120000000
```

マシン 2、マシン 3、およびマシン 4 は、機能拡張された `bankapp` アプリケーションです。

マシン 2 には、10000 ~ 39999 の口座があります。
マシン 3 には、40000 ~ 79999 の口座があります。
マシン 4 には、80000 ~ 109999 の口座があります。

すべての処理は、/DOMAIN ソフトウェアを使用して行われます。

ULOG##### を指定して tail -f を実行すると、要求の処理内容が表示されます。

要求を処理するマシンで、tail -f ULOG##### と入力します。

には日付を指定します。

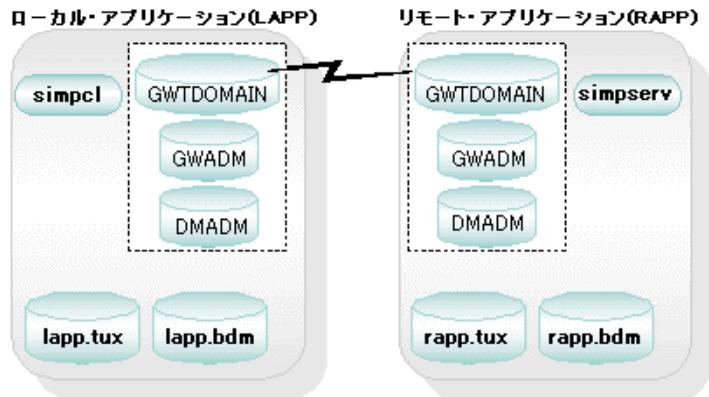
Domains 環境を設定する

Domains 環境を設定する際、管理者は、BEA Tuxedo ドメイン側で認識しておく必要のある、ほかのドメインの情報を指定する必要があります。たとえば、ほかのドメインからインポートされたサービス、リモート・ドメインへのアクセスに必要なアドレス指定パラメータとセキュリティ・パラメータ、アクセス制御リスト、ほかのドメインにエクスポートされたサービス、データ依存型ルーティングの指定、およびエクスポートされたサービスに対するアクセスを制御するパラメータを指定する必要があります。これらの情報は、UBBCONFIG コンフィギュレーション・ファイルとDMCONFIG コンフィギュレーション・ファイルで定義されます。

Domains のサンプル・アプリケーション (simpapp) をコンフィギュレーションする

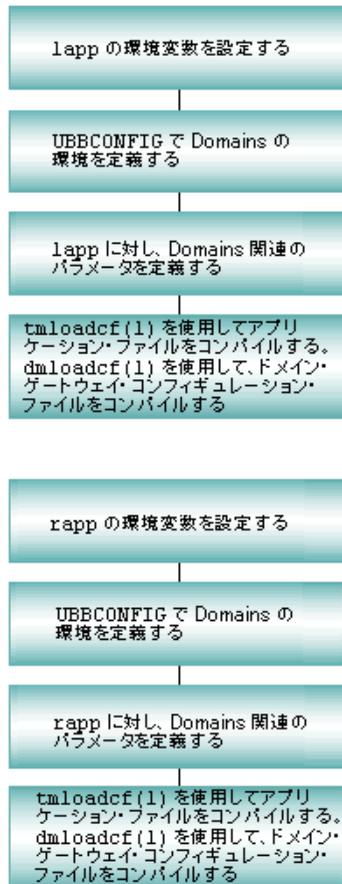
次の図に示す Domains の例は、ローカル・アプリケーション (lapp) とリモート・アプリケーション (rapp) の2つのアプリケーションで構成されています。どちらのアプリケーションも BEA Tuxedo システムに同梱されている simpapp に基づいています。lapp のクライアントは、TOUPPER サービスにアクセスできます。TOUPPER は rapp で宣言されています。

図 2-4 simpapp のローカル・アプリケーションとリモート・アプリケーション



コンフィギュレーションの作業

lapp (ローカル・アプリケーション) と rapp (リモート・アプリケーション) で構成される simpapp ドメインをコンフィギュレーションするには、次の作業を行う必要があります。



lapp の環境変数を設定する

アプリケーションを正しくコンフィギュレーションするには、次の環境変数を設定する必要があります。

- TUXDIR BEA Tuxedo システムのルート・ディレクトリ (例: /opt/tuxedo)
- TUXCONFIG アプリケーション・コンフィギュレーション・ファイル (例: lapp.tux)
- BDMCONFIG ドメイン・ゲートウェイ・コンフィギュレーション・ファイル (例: lapp.bdm)
- PATH TUXDIR/bin を含める必要があります
- LD_LIBRARY_PATH TUXDIR/lib を含める必要があります (このパス名は、オペレーティング・システムによって異なります)。

例

```
$ TUXDIR=/opt/tuxedo
$ TUXCONFIG=/home/lapp/lapp.tux
$ BDMCONFIG=/home/lapp/lapp.dom
$ PATH=$TUXDIR/bin:$PATH
$ LD_LIBRARY_PATH=$TUXDIR/lib:$LD_LIBRARY_PATH
$ export TUXDIR TUXCONFIG BDMCONFIG PATH LD_LIBRARY_PATH
```

ubbconfig ファイルで lapp の Domains 環境を定義する

ローカル・アプリケーションのサンプル・コンフィギュレーション・ファイルである `lapp.ubb` では、必須パラメータだけが定義されています。それ以外のパラメータには、デフォルト値が設定されます。

次の 2 つのサーバ・グループが定義されます。

- Domains 管理サーバ (DMADM) を含むサーバ・グループ
- ゲートウェイ管理サーバ (GWADM) とドメイン・ゲートウェイ (GWTDOMAIN) を含むサーバ・グループ

注記 GWTDOMAIN 以外のゲートウェイには、GWTDOMAIN 以外の実行可能ファイルを使用する必要があります。詳細については、BEA eLink for Mainframe のマニュアル、および『Using the BEA Tuxedo TOP END Domain Gateway』を参照してください。

サーバの定義

- DMADM Domains 管理サーバは、ドメイン・ゲートウェイ・グループに必要なコンフィギュレーション情報、つまり、バイナリ形式の Domains コンフィギュレーション・ファイルに格納されている情報を実行時に変更できます。DMADM は、登録済みのゲートウェイ・グループのリストを管理します。1 つの Domains アプリケーションに設定できる DMADM のインスタンスは 1 つだけです。
- GWADM ゲートウェイ管理サーバは、特定のドメイン・ゲートウェイ・グループを実行時に管理します。このサーバは、Domains のコンフィギュレーション情報を DMADM サーバから取得します。また、ゲートウェイ・グループを管理したり、トランザクション・ログを記録します。
- GWTDOMAIN Domains ゲートウェイ・サーバは、リモートの Domains との相互アクセスを実現し、複数の BEA Tuxedo ドメイン間での相互運用を可能にします。エクスポートやインポートに必要なローカル・サービスとリモート・サービスについての情報は、Domains コンフィギュレーション・ファイルに格納されています。Domains ゲートウェイ・サーバには、常に `REPLYQ=N` を指定する必要があります。

lapp 用のアプリケーション・コンフィギュレーション・ファイルの例

リスト 2-6 アプリケーション・コンフィギュレーション・ファイルの例 (lapp.ubb)

```
# lapp.ubb
#
*RESOURCES
IPCKEY          111111
MASTER         LAPP
MODEL          SHM

*MACHINES
giselle

                LMID=LAPP
                TUXDIR="/opt/tuxedo"
                APPDIR="/home/lapp"
                TUXCONFIG="/home/lapp/lapp.tux"

*GROUPS
LDMGRP         GRPNO=1  LMID=LAPP
LGWGRP         GRPNO=2  LMID=LAPP

*SERVERS
DMADM          SRVGRP=LDMGRP SRVID=1
GWADM          SRVGRP=LGWGRP SRVID=1
GWTDOMAIN     SRVGRP=LGWGRP SRVID=2  REPLYQ=N

*SERVICES
```

DMCONFIG ファイルで lapp の Domains パラメータを定義する

ローカル・ドメイン・ゲートウェイのサンプル・コンフィギュレーション・ファイル (lapp.dom) では、必須パラメータだけが定義されています。オプション・パラメータには、デフォルト値が設定されます。

DM_LOCAL_DOMAIN セクションでは、ローカル・ドメインとそれに関連するゲートウェイ・グループを指定します。このセクションの唯一のエントリ LAPP には、そのグループのドメイン・ゲートウェイ・プロセスで必要な次のパラメータを指定します。

- GWGRP には、アプリケーションで指定されたゲートウェイ・サーバ・グループの名前を指定します。
- TDOMAIN の TYPE は、ローカル・ドメインが別の BEA Tuxedo ドメインと通信することを示します。このパラメータは、ゲートウェイで使用されるプロトコルを示します。その他のオプションとして、SNA、OSI TP、TOP END Domain gateway、TCP for CICS、および TCP for IMS を指定することもできます。
- DOMAINID には、ドメイン・ゲートウェイの名前を指定します。この名前は、すべてのドメインで一意でなければなりません。

DM_REMOTE_DOMAINS セクションでは、既知のリモート・ドメインとその特性を指定します。このセクションのエントリは 1 つ (RAPP) だけです。TYPE は、ドメインの種類を識別するために使用されます。DOMAINSID は、一意なドメイン識別子です。

DM_TDOMAIN セクションでは、BEA Tuxedo の Domains コンポーネントに必要なアドレス指定情報を定義します。このコンフィギュレーション・ファイルで指定された、各ローカル・ドメインとリモート・ドメインのセクションには、次のエントリがあります。

- NWADDR では、ほかの BEA Tuxedo ドメインからの接続を受け付けるネットワーク・アドレス (ローカル・ドメイン・エントリ)、またはほかの BEA Tuxedo ドメインへの接続を確立するネットワーク・アドレス (リモート・ドメイン・エントリ) を指定します。

DM_LOCAL_SERVICES セクションでは、エクスポートされるサービスに関する情報を提供します。この例では、エクスポートされたサービスがないため、サンプル・ファイルのこのセクションにはエントリがありません。

DM_REMOTE_SERVICES セクションでは、インポートされるサービスに関する情報を提供します。この例では、TOUPPER サービスがインポートされています。ローカル・ドメイン内のクライアントは、このサービスにアクセスできます。

lapp 用の ドメイン・ゲートウェイ・コンフィギュレーション・ファイルの例

リスト 2-7 ドメイン・ゲートウェイ・コンフィギュレーション・ファイルの例 (lapp.dom)

```
#
# lapp.dom
#
*DM_LOCAL_DOMAINS
LAPP          GWGRP=LGWRP
              TYPE=TDOMAIN
              DOMAINID="111111"

*DM_REMOTE_DOMAINS
RAPP          TYPE=TDOMAIN
              DOMAINID="222222"

*DM_TDOMAIN
LAPP          NWADDR="//mach1:5000"
RAPP          NWADDR="//mach2:5000"

*DM_LOCAL_SERVICES
*DM_REMOTE_SERVICES
TOUPPER
```

lapp 用のアプリケーション・コンフィギュレーション・ファイルとドメイン・ゲートウェイ・コンフィギュレーション・ファイルをコンパイルする

ローカル・アプリケーション・コンフィギュレーション・ファイル (lapp.ubb) には、ローカル・アプリケーションの起動に必要な情報が含まれています。tmloadcf(1) を実行して、このファイルをバイナリ形式のファイル (lapp.tux) にコンパイルする必要があります。

ローカル・ドメイン・ゲートウェイ・コンフィギュレーション・ファイル (lapp.dom) には、1つのドメインのドメイン・ゲートウェイがほかのドメインと通信するために使用する情報が含まれています。dmloadcf(1) を実行して、このファイルをバイナリ形式のファイル (lapp.bdm) にコンパイルする必要があります。

両方のコンフィギュレーション・ファイルをコンパイルするには、次のセッション例に示す手順を実行してください。

```
$ cd /home/lapp
$ TUXCONFIG=/home/lapp/lapp.tux; export TUXCONFIG
$ tmloadcf -y lapp.ubb
$ BDMCONFIG=/home/lapp/lapp.dom; export BDMCONFIG
$ dmloadcf -y lapp.dom
```

ローカル・ドメインとリモート・ドメインを作成したら、tmboot(1) を使用してアプリケーションを起動できます。2つのドメインのうち、どちらから起動してもかまいません。dmadmin(1) を使用するとアプリケーションを監視できます。2つのアプリケーションが起動されたら、ローカル・アプリケーション内のクライアントは、リモート・アプリケーションにある TOUPPER サービスを呼び出すことができます。

```
$ tmboot -y
```

rapp の環境変数を設定する

アプリケーションを正しくコンフィギュレーションするには、次の環境変数を設定する必要があります。

- TUXDIR BEA Tuxedo システムのルート・ディレクトリ (例: /opt/tuxedo)
- TUXCONFIG アプリケーション・コンフィギュレーション・ファイル (例: rapp.tux) の絶対パス名
- BDMCONFIG ドメイン・ゲートウェイ・コンフィギュレーション・ファイル (例: rapp.bdm) の絶対パス名
- PATH TUXDIR/bin を含める必要があります
- LD_LIBRARY_PATH TUXDIR/lib を含める必要があります (このパス名は、オペレーティング・システムによって異なります)。

例

```
$ TUXDIR=/opt/tuxedo
$ TUXCONFIG=/home/rapp/rapp.tux
$ BDMCONFIG=/home/rapp/rapp.dom
$ PATH=$TUXDIR/bin:$PATH
$ LD_LIBRARY_PATH=$TUXDIR/lib:$LD_LIBRARY_PATH
$ export TUXDIR PATH LD_LIBRARY_PATH TUXCONFIG BDMCONFIG
```

UBBCONFIG ファイルで rapp の Domains 環境を定義する

リモート・アプリケーションのサンプル・コンフィギュレーション・ファイルである rapp.ubb では、必須パラメータだけが定義されています。オプション・パラメータには、デフォルト値が設定されます。

次の 3 つのサーバ・グループが定義されます。

- 最初のサーバ・グループ (SRVGP=RDMGRP) には、Domains 管理サーバ (DMADM) が含まれます。

- 2 番目のサーバ・グループ (SRVGP=RGWGRP) には、ゲートウェイ管理サーバである GWADM、およびドメイン・ゲートウェイである GWTDOMAIN が含まれます。
- 3 番目のサーバ・グループ (SRVGP=APPGRP) には、アプリケーション・サーバである simpserv が含まれます。

次の 4 つのサーバが定義されます。

- DMADM Domains 管理サーバ
- GWADM ゲートウェイ管理サーバ
- GWTDOMAIN Domains ゲートウェイ・サーバ
- simpserv simpapp のアプリケーション・サーバ。文字列を小文字から大文字に変換する TOUPPER サービスを宣言します。

rapp 用のアプリケーション・コンフィギュレーション・ファイルの例

リスト 2-8 アプリケーション・コンフィギュレーション・ファイルの例 (rapp.ubb)

```
# rapp.ubb
#
*RESOURCES
IPCKEY          222222
MASTER         RAPP
MODEL          SHM

*MACHINES
juliet

                LMID=RAPP
                TUXDIR="/opt/tuxedo"
                APPDIR="/home/rapp"
                TUXCONFIG="/home/rapp/rapp.tux"

*GROUPS

RDMGRP          GRPNO=1 LMID=RAPP
RGWGRP          GRPNO=2 LMID=RAPP
APPGRP          GRPNO=3 LMID=RAPP

*SERVERS
DMADM           SRVGRP=RDMGRP SRVID=1
GWADM           SRVGRP=RGWGRP SRVID=1
```

```
GWTDOMAIN      SRVGRP=RGWGRP SRVID=2  REPLYQ=N
simpserv       SRVGRP=APPGRP SRVID=1
*SERVICES
TOUPPER
```

DMCONFIG ファイルで rapp の Domains パラメータを定義する

リモート・ドメイン・ゲートウェイのサンプル・コンフィギュレーション・ファイル `rapp.dom` では、必須パラメータだけが定義されています。オプション・パラメータには、デフォルト値が設定されます。

このコンフィギュレーション・ファイルは、ローカル・ドメイン・ゲートウェイのコンフィギュレーション・ファイルに似ています。唯一の相違点は、エクスポートおよびインポートされるサービスです。

`DM_LOCAL_SERVICES` セクションは、各ローカル・ドメインによってエクスポートされたサービスに関する情報を提供します。この例では、`TOUPPER` サービスがエクスポートされ、`DM_LOCAL_SERVICES` セクションに組み込まれます。インポートされるサービスはないため、サンプル・ファイルの `DM_REMOTE_SERVICES` セクションにはエントリがありません。

rapp の Domain ゲートウェイ・コンフィギュレーション・ファイルの例

リスト 2-9 ドメイン・ゲートウェイ・コンフィギュレーション・ファイルの例 (`rapp.dom`)

```
# rapp.dom
#
*DM_LOCAL_DOMAINS
RAPP          GWGRP=RGWGRP
              TYPE=TDOMAIN
              DOMAINID=" 222222 "
*DM_REMOTE_DOMAINS
```

```
LAPP          TYPE=TDOMAIN
              DOMAINID="111111"

*DM_TDOMAIN

RAPP          NWADDR="//mach2:5000"

LAPP          NWADDR="//mach1:5000"

*DM_LOCAL_SERVICES
TOUPPER
*DM_REMOTE_SERVICES
```

rapp 用のアプリケーション・コンフィギュレーション・ファイルとドメイン・ゲートウェイ・コンフィギュレーション・ファイルをコンパイルする

リモート・アプリケーション・コンフィギュレーション・ファイル (rapp.ubb) には、1つのドメインのドメイン・ゲートウェイがほかのドメインと通信するために使用する情報が含まれています。このファイルをバイナリ形式のファイル (rapp.tux) にコンパイルする必要があります。

リモート・ドメイン・ゲートウェイ・コンフィギュレーション・ファイル (rapp.dom) には、ドメイン・ゲートウェイがほかのドメインとの通信に必要なコンテキストを初期化するための情報が含まれています。このコンフィギュレーション・ファイルは、ローカル・ドメイン・ゲートウェイのコンフィギュレーション・ファイルに似ています。唯一の相違点は、エクスポートおよびインポートされるサービスです。このファイルは、バイナリ形式のファイル (rapp.bdm) にコンパイルする必要があります。

```
$ cd /home/rapp
$ TUXCONFIG=/home/rapp/rapp.tux; export TUXCONFIG
$ tmloadcf -y rapp.ubb
$ BDMCONFIG=/home/rapp/rapp.dom; export BDMCONFIG
$ dmloadcf -y rapp.dom
```

ローカル・ドメインとリモート・ドメインを作成したら、`tmboot(1)` を使用してアプリケーションを起動できます。2つのドメインのうち、どちらから起動してもかまいません。`dmadmin(1)` を使用するとアプリケーションを監視できます。2つのアプリケーションが起動されたら、ローカル・アプリケーション内のクライアントは、リモート・アプリケーションにある `TOUPPER` サービスを呼び出すことができます。

```
$ tmboot -y
```

関連項目

- 第1章の18ページ「Domains コンフィギュレーション・ファイルとは」
- 第2章の30ページ「Domains 間でデータを圧縮する」
- 第2章の31ページ「サービス要求をリモート・ドメインにルーティングする」
- 第1章の20ページ「Domains コンフィギュレーション・ファイルを変換する」
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `DMCONFIG(5)`
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `UBBCONFIG(5)`

Domains 間でデータを圧縮する

ドメイン間で送信されるデータを圧縮すると、性能を高めることができます。圧縮を設定するには、`DMCONFIG` の `CMPLIMIT` パラメータを設定します。

関連項目

- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `DMCONFIG(5)`
- 『BEA Tuxedo アプリケーション実行時の管理』の第4章の1ページ「ネットワーク・データの圧縮」

サービス要求をリモート・ドメインにルーティングする

サービス要求を特定のリモート・ドメインに送信するため、ゲートウェイで使用されるデータ依存型ルーティング情報は、DMCONFIG ファイルの DM_ROUTING セクションで指定されます。サポートされる型付きバッファは、FML32、VIEW32、FML、VIEW、X_C_TYPE、および X_COMMON です。

ドメインのルーティング・テーブルを作成するには、次の項目を指定します。

- ルーティング・エントリで有効なバッファ型
- ルーティング・エントリとルーティング・フィールドの名前
- ルーティング・フィールドの範囲、および関連するリモート・ドメインの名前。

これらのフィールドを次の表で説明します。

ルーティング・テーブルのフィールド	説明
バッファ型	<p>このルーティング・エントリで有効なデータ・バッファの型とサブタイプのリストです。型には、FML32、VIEW32、FML、VIEW、X_C_TYPE、および X_COMMON があります。FML 型には、サブタイプを指定できません。それ以外の型では、サブタイプが必要です。(ワイルドカード)値は使用できません。重複する型とサブタイプの組み合わせを、同じルーティング基準名には指定できません。型とサブタイプの組み合わせが一意であれば、1つの基準名を複数のルーティング・エントリで指定できます。1つのルーティング・エントリに対して複数のバッファ型を指定する場合は、各バッファ型に対するルーティング・フィールドのデータ型は同じでなければなりません。</p> <p><i>type</i> の有効な値は次のとおりです。 <code>[[:subtype1[,subtype2 . . .]]</code> <code>[;type2[:subtype3[,subtype4 . . .]]] . . .</code></p> <p>型とサブタイプの組み合わせは 32 あり、合計の最大長は 256 文字です。サブタイプの有効な値には、セミコロン (;)、コロン (:)、カンマ (,)、およびアスタリスク (*) を指定できません。</p> <p>例 :FML</p>
ドメインのルーティング基準	<p>ルーティング・エントリの名前 (識別子) です。有効な値は、1 ~ 15 文字で構成する文字列です。</p> <p>例 :ROUTTAB1</p>

ルーティング・テーブルの フィールド	説明 (続き)
ルーティング・フィールド名	<p>ルーティング・フィールドの名前です。このフィールドの値は、FML フィールド・テーブルで識別された名前 (FML パッファの場合)、または FML VIEW テーブルで識別された名前 (VIEW、X_C_TYPE、または X_COMMON パッファの場合) と見なされます。</p> <p>有効な値は、1 ~ 30 文字で構成する識別子です。</p> <p>例 : FIELD1</p>
範囲	<p>ルーティング・フィールドのリモート・ドメイン名 (RDOM) に関連する、数値のセット (数値を含む必要あり) と英数字文字列 (文字列値を含む必要あり) で構成される値です。ルーティング・フィールドには、FML でサポートされている任意のデータ型を指定できます。</p> <p>string 型、carray 型、および文字型のフィールドの文字列値は、次の条件を満たす必要があります。</p> <ul style="list-style-type: none"> ■ 一重引用符で囲み、先頭に符号が付きません。 ■ short 型および long 型の整数値は数値の文字列であり、オプションで先頭に正の符号または負の符号を指定できます。 ■ 浮動小数点数は、C コンパイラまたは atof() で受け付けられる形式です。つまり、符号 (オプション)、数字の文字列 (オプションで小数点を追加)、e または E (オプション)、符号または空白文字 (オプション)、整数、という形式で指定します。 <p>フィールド値が範囲と一致する場合、関連する RDOM 値には、要求のルーティング先のリモート・ドメインを指定します。RDOM 値に * を指定すると、ゲートウェイ・グループが認識する任意の要求が送信されることを示します。</p> <p>このフィールドで有効な値は、範囲 /RDOM の組み合わせをカンマで区切った順序付きのリストです。範囲は、(a) 単一の値 (符号付き数値または一重引用符で囲んだ文字列)、または、(b) lower-upper (lower と upper は共に一重引用符で囲んだ符号付き数値または文字列) の形式で表します。なお、lower は、upper 以下でなければなりません。</p> <p>範囲 /RDOM の組み合わせでは、範囲と RDOM はコロン (:) で区切られます。関連する FIELD のデータ型の最小値を指定するには、MIN を使用します。</p>

ルーティング・テーブルの
フィールド

説明 (続き)

- 文字列と `carrays` の最小値は、ヌル文字列です。
 - 文字フィールドの最小値は、0 です。
 - 数値の場合、そのフィールドに格納できる最小値になります。
- 関連する FIELD: のデータ型の最大値を指定するには、MAX を使用しま
す。

- 文字列と `carray` の最大値は、8 進の 255 文字の無限文字列です。
- 文字フィールドの最大値は、単一の 8 進の 255 文字です。
- 数値の場合、数値としてフィールドに格納できる最大値になります。

したがって、MIN - -5 は、-5 以下のすべての数を表します。6 - MAX は、6 以上のすべての数を表します。範囲にメタ文字 * (ワイルドカード) を使用すると、既にエントリに表示された範囲以外にある任意の値が対象となります。各エントリで使用できるワイルドカードは 1 つだけであり、フィールドの最後に指定する必要があります。ワイルドカードの後に範囲を指定すると無視されます。

例 : 1-100:REMDOM3

Domains でセキュリティを設定する

BEA Tuxedo ATMI 環境には、次のアプリケーション用セキュリティ・メカニズムが用意されています。

- 認証 ユーザまたはシステム・プロセスに対して指定されている ID を証明する機能です。ID 情報を安全に記録および転送し、必要に応じて ID 情報を利用可能にします。デフォルトの認証プラグインには、認証なし (NONE)、アプリケーション・パスワードを使用 (AP_PW)、およびユーザ・レベルの認証を実行 (USER_AUTH) という 3 つのセキュリティ・レベルが用意されています。
- 認可 ID またはその他の情報に基づいて、リソースへのアクセスを制御します。デフォルトの認可プラグインには、オプションのアクセス制御リスト (ACL) と必須のアクセス制御リスト (MANDATORY_ACL) という 2 つのセキュリティ・レベルが用意されています。
- 監査 操作要求とその結果に関する情報を安全に収集、格納、および配布します。デフォルトの監査機能は、BEA Tuxedo のイベント・ブローカとユーザ・ログ (ULOG) の機能によってインプリメントされます。

- 暗号化 データを符号化し、ユーザが認識できない形式に変換するセキュリティ・メカニズムです。デフォルトでは、RC4 形式の対象鍵による暗号化が設定されています。
- セキュリティ・プラグイン・インターフェイス カスタマイズされた認証、認可、監査など、サードパーティ製のセキュリティ・システムのインストールを可能にします。ただし、プラグイン・インターフェイスは、BEA Tuxedo リリース 7.1 以降のソフトウェアを実行するアプリケーションでのみ利用できます。セキュリティ・プラグイン・インターフェイスを使用するドメインでセキュリティを設定する方法については、『BEA Tuxedo のセキュリティ機能』の第 2 章の 23 ページ「ドメイン間のリンクの確立」を参照してください。

BEA Tuxedo アプリケーションのセキュリティと Domains のセキュリティの関係

個々のアプリケーションと Domains コンフィギュレーションでは、用意されている BEA Tuxedo のセキュリティ・メカニズムが異なりますが、互換性はあります。

- BEA Tuxedo ATMI 環境には、Domains コンフィギュレーション用に次のセキュリティ・メカニズムが用意されています。
 - リモード・ドメインの認証
 - リモート・ドメインにエクスポートされたローカル・サービスのアクセス制御
 - ドメイン間の通信をセキュリティ保護する暗号化
- BEA Tuxedo アプリケーションのセキュリティが `USER_AUTH` 以上のレベルに設定され、かつ、`DMCONFIG` ファイルの `DM_REMOTE_DOMAINS` セクションにある `ACL_POLICY` パラメータと `CREDENTIAL_POLICY` パラメータが `GLOBAL` に設定されている場合、ドメイン間での要求の送信時にユーザ ID も送信され、`ACL` によるチェックが行われます。
- BEA Tuxedo アプリケーションのセキュリティが `USER_AUTH` 以上のレベルに設定され、かつ `DMCONFIG` ファイルの `DM_REMOTE_DOMAINS` セクションにある `ACL_POLICY` パラメータと `CREDENTIAL_POLICY` パラメータが `LOCAL` に設定されている場合、要求の送信時にユーザ ID は送信されません。`ACL` によるチェックは、要求元のドメインの Domain ID によって処理されます。

- BEA Tuxedo アプリケーションのセキュリティ・レベルを `NONE` を割り当てていても、Domains の接続セキュリティを `DM_PW` に設定することはできません。ただし、Domains コンフィギュレーションでアプリケーション・パスワードを使用するには、参加する各アプリケーションのセキュリティ・レベルを、`APP_PW` (または `USER_AUTH`、`ACL`、`MANDATORY_ACL`) に設定しておく必要があります。

Domains のセキュリティ・メカニズム

特定のドメインを多数のユーザが使用する場合があるため、BEA Tuxedo の標準のアプリケーション・パスワード機構では、十分なセキュリティ機能を実現できないことがあります。したがって、Domains には、別のセキュリティ・メカニズムが用意されています。

- アクセス制御リスト リストに指定されたりリモート・ドメインだけがローカル・ドメインのリソースを利用できるようにする機能です。このセキュリティ・レベルは、`DMCONFIG` の `DM_ACCESS_CONTROL` セクションで設定します。
- Domains の認証機能 各リモート・ドメインを正しく識別するためには、認証技術が必要です。Domains には、パスワード・セキュリティのレベルが3つ用意されています。`NONE` は、認証機能を指定しないことを示します。`APP_PW` は、アプリケーション・パスワードを使用して認証を行うことを示します。このパスワードは、2つのドメインで一致しなければなりません。`DM_PW` は、ローカル・ドメインまたはリモート・ドメインに対して1対1で特定のパスワードを設定し、認証を行うことを示します。セキュリティ・レベルを指定するには、関連するローカル・ドメインのアクセス・ポイントの `DM_LOCAL_DOMAINS` セクションにある `SECURITY` パラメータに、必要なレベル (`NONE`、`APP_PW`、または `DM_PW`) を設定します。
- リンク・レベルの暗号化 ドメイン間の通信で暗号化を使用すると、データの機密性を保持できます。この方法により、ドメイン・ゲートウェイ間で送受信される BEA Tuxedo メッセージやアプリケーション生成されたメッセージの内容を、ネットワーク上の侵入者に知られないようにすることができます。このセキュリティ・メカニズムを設定するには、`DMCONFIG` ファイルで `MINENCRYPTBITS` パラメータと `MAXENCRYPTBITS` パラメータを設定します。
- ローカル Domains アクセス リモート・ドメインからローカル・サービスへのアクセスを制限します。サービスがリモート・ドメインにエクスポートされない限り、そのサービスをリモート・ドメインから利用することはできません。サービスをエクスポートするには、サービスの `DMCONFIG` ファイルの `DM_LOCAL_SERVICES` セクションにエントリを指定します。

- メインフレームへのユーザ ID のマッピング ドメイン内のユーザ ID と外部のユーザ ID を相互にマッピングできるメカニズムを提供します。現在、このメカニズムは、BEA eLink for Mainframe-SNA で、IBM LU6.2 メインフレームの RACF (リモート・アクセス制御機能) ユーザ名と相互にマッピングするために使用されています。このメカニズムを使用するには、次のコンフィギュレーション用コマンド `dmadmin` を参照してください。
 - `addumap` ローカル・ドメインとリモート・ドメインのペアのリモート・ユーザ・マッピングに、ローカル・ユーザ・マッピングを追加します。マッピングは、インバウンド、アウトバウンド、または両方、のいずれかで定義されます。
 - `addusr` リモート・ドメインのリモート・ユーザおよびパスワード・テーブルに、リモート・ユーザ名とパスワードを追加します。
 - `delumap` ローカル・ドメインとリモート・ドメインのペアのローカル・ユーザからリモート・ユーザへのマッピングを削除します。
 - `delusr` リモート・ドメインのリモート・ユーザおよびパスワード・テーブルから、リモート・ユーザ名とパスワードを削除します。
 - `modusr` リモート・ドメインのパスワード・テーブルで、リモート・パスワードを変更します。

関連項目

- 第 2 章の 37 ページ「ドメイン用のアクセス制御リスト (ACL) を作成する」
- 第 2 章の 41 ページ「Domains の認証機能を設定する」
- 第 2 章の 43 ページ「Domains 間のセキュリティのコーディング例」
- 『BEA Tuxedo コマンド・リファレンス』の `dmadmin(1)`

ドメイン用のアクセス制御リスト (ACL) を作成する

ドメイン用の ACL を作成するには、DMCONFIG ファイルの DM_ACCESS_CONTROL セクションで、ACL の名前と、ドメイン・インポート VIEW リストに含まれるリモート・ドメインのリストを指定する必要があります。次の表では、これらの 2 つのフィールドを説明します。

ドメイン用の ACL のフィールド	説明
ドメイン用の ACL の名前	この ACL の名前です。 1 ~ 30 文字で構成した文字列が有効です。この名前は出力可能です。コロン (:)、#、および復帰改行文字は指定できません。 例 :ACLGRP1
リモート・ドメインのリスト	このアクセス制御リストでアクセスを許可されたリモート・ドメインのリストです。 このフィールドには、1 つまたは複数のカンマで区切ったリモート・ドメイン名を指定できます。 例 :REMDOM1 ,REMDOM2 ,REMDOM3

インポートされたリモート・サービスに対して標準の BEA Tuxedo アクセス制御リストを使用する

リモート・ドメインからインポートされたリモート・サービスは、単に BEA Tuxedo ドメイン内のサービスと見なされます。したがって、標準の BEA Tuxedo ACL メカニズムを使用して、これらのサービスへのアクセスを特定のユーザ・グループに制限できます。

BEA Tuxedo のアクセス制御リストの使い方については、『BEA Tuxedo コマンド・リファレンス』の次のエントリを参照してください。tpacladd(1)、tpaclmod(1)、tpacldel(1)、tpusradd(1)、tpusrmod(1)、tpusrdel(1)、tpgrpadd(1)、tpgrpmod(1)、および tpgrpdel(1) に関する記述。

リモート・ドメインの ACL 方針を設定する

管理者は、次のコンフィギュレーション・パラメータを使用して、BEA Tuxedo リリース 7.1 以上のソフトウェアを実行するリモート・ドメインで、アクセス制御リスト (ACL: Access Control List) 方針の設定と制御を行います。

パラメータ名	説明	設定
DMCONFIG の ACL_POLICY (DM_MIB の TA_DMACLPOLICY)	リモート・ドメイン・アクセス・ポイントごとに、DMCONFIG ファイルの DM_REMOTE_DOMAINS セクションで指定される場合があります。特定のリモート・ドメイン・アクセス・ポイントに対するこのパラメータの値により、ローカル・ドメイン・ゲートウェイがリモート・ドメインから受信したサービス要求の ID を変更するかどうかが決まります。*	LOCAL または GLOBAL。デフォルト値は LOCAL です。 LOCAL は、このリモート・ドメインから受信したサービス要求の ID を、このリモート・ドメイン用の LOCAL_PRINCIPAL_NAME パラメータで指定されたプリンシパル名に変更することを示します。 GLOBAL は、ローカル・ドメインが、リモート・ドメインから受信したインバウンドのサービス要求のクレデンシャルを使用することを示します。リモート・ドメインからクレデンシャルを受信しなかった場合、サービス要求はクレデンシャルなしでサービスに送信されますが、通常、その要求は失敗します。 注記 このパラメータは、ローカル・ドメインがリモート・ドメインからのクレデンシャルを「受け付ける」かどうかを制御します。関連パラメータである CREDENTIAL_POLICY パラメータは、ローカル・ドメインがリモート・ドメインにクレデンシャルを「送信する」かどうかを制御します。

* リモート・ドメイン・アクセス・ポイントは、RDOM (アールドム) または単に「リモート・ドメイン」とも呼ばれます。

パラメータ名	説明	設定
DMCONFIG の LOCAL_PRINCIPAL_NAME (DM_MIB の TA_DMLOCALPRINCIPALNAME)	リモート・ドメイン・アクセス・ポイントごとに、DMCONFIG ファイルの DM_REMOTE_DOMAINS セクションで指定される場合があります。特定のリモート・ドメイン・アクセス・ポイントに対し、ACL_POLICY パラメータに LOCAL (デフォルト値) が設定された場合、ローカル・ドメイン・ゲートウェイは、リモート・ドメインから受け取ったサービス要求の ID を、LOCAL_PRINCIPAL_NAME のプリンシパル名に変更します。	1 ~ 511 文字。指定しない場合、リモート・ドメイン・アクセス・ポイントの DOMAINID 文字列がデフォルト値になります。

* リモート・ドメイン・アクセス・ポイントは、RDOM (アールドム) または単に「リモート・ドメイン」とも呼ばれます。

ACL 方針の詳細については、『BEA Tuxedo のセキュリティ機能』を参照してください。

リモート・ドメインのクリデンシャル方針を設定する

管理者は、次のコンフィギュレーション・パラメータを使用して、BEA Tuxedo リリース 8.0 以上のソフトウェアを実行するリモート・ドメインのクリデンシャル方針の設定と制御を行います。

パラメータ名	説明	設定
DMCONFIG の CREDENTIAL_POLICY (DM_MIB の TA_DMCREREDENTIALPOLICY)	リモート・ドメイン・アクセス・ポイントごとに、DMCONFIG ファイルの DM_REMOTE_DOMAINS セクションで指定される場合があります。特定のリモート・ドメイン・アクセス・ポイントに対するこの値により、ローカル・ドメイン・ゲートウェイが、リモート・ドメインから受信したサービス要求の ID を変更するかどうかが決まります。*	LOCAL または GLOBAL。デフォルトは LOCAL です。 LOCAL の場合、リモート・ドメインの呼び出しに要求元ユーザのクリデンシャルがアタッチされません。 GLOBAL の場合、リモート・ドメインの呼び出しに要求元ユーザのクリデンシャルがアタッチされます。 注記 このパラメータは、ユーザのクリデンシャルをリモート・ドメインに送信するかどうかを制御します。ACL_POLICY パラメータも似ていますが、ACL_POLICY パラメータの場合は、着信したクリデンシャルをドメインで受け付けるかどうかを制御します。

* リモート・ドメイン・アクセス・ポイントは、RDOM (アールドム) または単に「リモート・ドメイン」とも呼ばれます。

クリデンシャル方針の詳細については、『BEA Tuxedo のセキュリティ機能』を参照してください。

Domains の認証機能を設定する

ドメイン・ゲートウェイでは、リモート・ドメインからローカル・ドメインへの接続、またはその逆の接続に対する要求を認証できます。ただし、認証メカニズムはオプションであり、TUXCONFIG ファイルで指定されている BEA Tuxedo のメカニズムと互換性があります。

アプリケーション管理者は、リモート・ドメインからの接続に対して、セキュリティ機能をいつ実行するかを定義できます。特定のローカル・ドメインで使用するセキュリティのレベルを指定するには、DMCONFIG ファイルの DM_LOCAL_DOMAINS セクションで SECURITY パラメータを設定します。パスワード・セキュリティには 3 つのレベルがあります。

- セキュリティなし (NONE オプションを使用) リモート・ドメインからの接続は認証されません。
- アプリケーション・パスワード (APP_PW オプションを使用) リモート・ドメインからの接続は、TUXCONFIG ファイルで定義されたアプリケーション・パスワードを使用して認証されます。BEA Tuxedo のアプリケーション・パスワードは、`tmloadcf(1)` で要求されます。TUXCONFIG ファイルの SECURITY オプションが有効であると、パスワードの入力が求められます。パスワードは、TUXCONFIG ファイルとともに、コンフィギュレーション内のほかのノードに自動的に伝播されます。パスワードを動的に更新するには、`tmadmin` コマンドを使用します。
- リモート Domains パスワード (DM_PW オプションを使用) BEA Tuxedo Domains では、この機能を使用して、複数の BEA Tuxedo ドメイン間でのセキュリティ機能を強化します。ローカル・ドメインとリモート・ドメインの間の接続は、BDMCONFIG ファイルの DM_PASSWORDS セクションで定義されたパスワードを使用して認証されます。`dmloadcf` の実行後、これらのパスワードは、`dmadmin(1)` コマンドの `passwd` サブコマンドにより、バイナリ形式のコンフィギュレーション・ファイルに追加されます。各エントリには、リモート・ドメインから特定のローカル・ドメインにアクセスするためのパスワードと、ローカル・ドメインからリモート・ドメインにアクセスするためのパスワードが格納されています。

TUXCONFIG で SECURITY パラメータが設定されていない場合、つまり、デフォルトの NONE が指定されるか、または明示的に NONE が設定された場合でも、Domains コンフィギュレーションのドメイン・ゲートウェイでは、DM_PW レベルのセキュリティを適用できます。DM_PW オプションが選択された場合、各リモート・ドメインに対して、BDMCONFIG ファイルの DM_PASSWORDS セクションでパスワードを定義する必要があります。つまり、パスワードが設定されていないリモート・ドメインからの接続は、ドメイン・ゲートウェイによって拒否されます。

T_DM_PASSWORDS MIB クラスの定義

T_DM_PASSWORDS クラスは、TDOMAIN タイプのローカル・アクセス・ポイントとリモート・アクセス・ポイントでドメイン間認証を行うためのコンフィギュレーション情報を示します。T_DM_PASSWORDS クラスには、リモート・ドメインごとに以下のエントリがあります。

- TA_DMLACCESSPOINT— パスワードが適用されるローカル・ドメイン・アクセス・ポイントの名前。
- TA_DMRACCESSPOINT— パスワードが適用されるリモート・ドメイン・アクセス・ポイントの名前。
- TA_DMLPWD— ローカル・ドメイン・アクセス・ポイント (TA_DMLACCESSPOINT で指定) とリモート・ドメイン・アクセス・ポイント (TA_DMRACCESSPOINT で指定) の間で接続認証に使用されるローカル・パスワード。
- TA_DMRPWD— リモート・ドメイン・アクセス・ポイント (TA_DMRACCESSPOINT で指定) とローカル・ドメイン・アクセス・ポイント (TA_DMLACCESSPOINT で指定) の間で接続認証に使用されるリモート・パスワード。

注記 パスワードは、暗号化されて安全に格納されます。

Domains パスワードを設定する

BEA Tuxedo の Domains パスワード (DM_PW) は、次のように `dmadmin(1)` コマンドを使用して設定します。

```
passwd [-r] local_domain_name remote_domain_name
```

管理者に対し、指定したローカル・ドメインとリモート・ドメインの新しいパスワードを入力するよう求められます。

関連項目

- 第 2 章の 43 ページ「Domains 間のセキュリティのコーディング例」
- 『BEA Tuxedo コマンド・リファレンス』の `dmadmin(1)`
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `DMCONFIG(5)`

- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の DM_MIB(5)

Domains 間のセキュリティのコーディング例

DMCONFIG ファイルの DM_LOCAL_DOMAINS セクションの SECURITY パラメータでは、ローカル・ドメインのセキュリティ・タイプを指定します。認証が必要な場合は、ローカル・ドメインとリモート・ドメインの間で接続が確立されるたびに認証が行われます。2つのドメインのセキュリティ・タイプに互換性がない場合、またはパスワードが一致しない場合、接続を確立することはできません。

例 1: セキュリティとして NONE を設定する

ローカル・ドメインで SECURITY に NONE が設定されると、ほかのドメインからの接続は認証されません。SECURITY に NONE が設定されても、ローカル・ドメインは、SECURITY に DM_PW が設定されたリモート・ドメインには接続できます。ただし、このような接続を確立する前に、`dmadmin(1)` を実行するか、または DM_MIB(5) を使用して、両方のドメインでパスワードを定義しなければなりません。

リスト 2-10 アプリケーションと Domains に対して NONE を設定する

```
DOM1:SECURITY in UBBCONFIG set to NONE
      SECURITY in DMCNFIG set to NONE
DOM2:SECURITY in UBBCONFIG set to NONE
      SECURITY in DMCNFIG set to DM_PW
```

この例では、DOM1 にはセキュリティが設定されていません。DOM2 には DM_PW セキュリティが設定されています。

要求側では、UBBCONFIG と DMCONFIG の関連属性が次のように設定されます。

```
UBBCONFIG
SECURITY=NONE

DMCONFIG
*DM_LOCAL_DOMAINS
DOM1
  DOMAINID=DOM1
  SECURITY=NONE

*DM_REMOTE_DOMAINS
DOM2 DOMAINID="DOM2"
```

応答側では、UBBCONFIG と DMCONFIG の関連属性が次のように設定されます。

```
UBBCONFIG
SECURITY=NONE

DMCONFIG
*DM_LOCAL_DOMAINS
DOM2
  DOMAINID=DOM2
  SECURITY=DM_PW

*DM_REMOTE_DOMAINS
DOM1 DOMAINID="DOM1"
```

TUXCONFIG ファイルと BDMCONFIG ファイルで必要な属性が設定された後、DOM1 と DOM2 でアプリケーションを起動します。

```
DOM1 上 :
dmadmin
passwd DOM1 DOM2
  Enter Local Domain Password:foo1
  Reenter Local Domain Password:foo1
  Enter Remote Domain Password:foo2
  Reenter Remote Domain Password:foo2
```

```
DOM2 上 :
dmadmin
passwd DOM2 DOM1
  Enter Local Domain Password:foo2
  Reenter Local Domain Password:foo2
  Enter Remote Domain Password:foo1
  Reenter Remote Domain Password:foo1
```

両方のドメインでパスワードを設定すると、接続を確立でき、リモート・ドメイン上でサービスを実行できます。

リスト 2-11 アプリケーションに対して NONE を設定し、Domains に対して DM_PW を設定する

要求側では、UBBCONFIG と DMCONFIG の関連属性が次のように設定されます。

```
UBBCONFIG
  SECURITY=NONE

DMCONFIG
  *DM_LOCAL_DOMAINS
  DOM1
  DOMAINID=DOM1
  SECURITY=DM_PW

  *DM_REMOTE_DOMAINS
  DOM2 DOMAINID="DOM2"
```

応答側では、UBBCONFIG と DMCONFIG の関連属性が次のように設定されます。

```
UBBCONFIG
  SECURITY=NONE

DMCONFIG
  *DM_LOCAL_DOMAINS
  DOM2
  DOMAINID=DOM2
  SECURITY=DM_PW

  *DM_REMOTE_DOMAINS
  DOM1 DOMAINID="DOM1"
```

TUXCONFIG ファイルと BDMCONFIG ファイルで必要な属性を設定したら、DOM1 と DOM2 でアプリケーションを起動します。

```
DOM1 上 :
dmadmin
passwd DOM1 DOM2
Enter Local Domain Password:foo1
Reenter Local Domain Password:foo1
Enter Remote Domain Password:foo2
Reenter Remote Domain Password:foo2

DOM2 上 :
dmadmin
passwd DOM2 DOM1
Enter Local Domain Password:foo2
Reenter Local Domain Password:foo2
Enter Remote Domain Password:foo1
Reenter Remote Domain Password:foo1
```

両方のドメインでパスワードを設定すると、接続を確立でき、リモート・ドメイン上でサービスを実行できます。

例 2: セキュリティとして APP_PW を設定する

UBBCONFIG の SECURITY パラメータが APP_PW 以上のレベルに設定されている場合、DMCONFIG の SECURITY には、NONE、APP_PW、または DM_PW を設定できます。1 つの DMCNFIG ファイルには、ドメインのビューを複数定義できるため (ただし、ローカル・ドメインの定義ごとに 1 つのビューを指定)、各ビューに対して異なるセキュリティ・メカニズムを割り当てることができます。

注記 ローカル・ドメイン・アクセス・ポイントに対して、DMCONFIG の SECURITY に APP_PW が設定されている場合、UBBCONFIG の SECURITY には APP_PW 以上のセキュリティ・レベルを設定しなければなりません。

リスト 2-12 アプリケーションと Domains に対して APP_PW を設定する

```
DOM1:SECURITY in UBBCONFIG set to APP_PW
      SECURITY in DMCNFIG set to APP_PW
DOM2:SECURITY in UBBCONFIG set to APP_PW
      SECURITY in DMCNFIG set to APP_PW
```

この例では、DOM1 と DOM2 の両方で APP_PW セキュリティが適用されます。

要求側では、UBBCONFIG と DMCNFIG の関連属性が次のように設定されます。

```
UBBCONFIG
  SECURITY=APP_PW

DMCNFIG
  *DM_LOCAL_DOMAINS
DOM1
  DOMAINID=DOM1
  SECURITY=APP_PW

  *DM_REMOTE_DOMAINS
DOM2 DOMAINID="DOM2"
```

応答側では、UBBCONFIG と DMCNFIG の関連属性が次のように設定されます。

```
UBBCONFIG
  SECURITY=APP_PW

DMCNFIG
  *DM_LOCAL_DOMAINS
```

```
DOM2
  DOMAINID=DOM2
  SECURITY=APP_PW

*DM_REMOTE_DOMAINS
DOM1 DOMAINID="DOM1"
```

TUXCONFIG ファイルと BDMCONFIG ファイルを作成した後、DOM1 と DOM2 でアプリケーションを起動します。

ドメイン間の接続を設定する

ユーザは、ローカル・ドメイン・ゲートウェイからリモート・ドメインへの接続の条件を指定することができます。この条件を指定するには、Domains コンフィギュレーション・ファイルの CONNECTION_POLICY パラメータに値を割り当てます。接続方針は次の 3 種類です。

- 起動時に接続する (ON_STARTUP)。
- クライアント・プログラムにより、リモート・サービスが要求されたときに接続する (ON_DEMAND)。
- 受信時に接続するが、自動的には行わない (INCOMING_ONLY)。

接続方針が ON_STARTUP および INCOMING_ONLY の場合、Dynamic Status が呼び出されます。Dynamic Status とは、リモート・サービスのステータスのチェックとレポートを行う BEA Tuxedo の Domains の機能です。

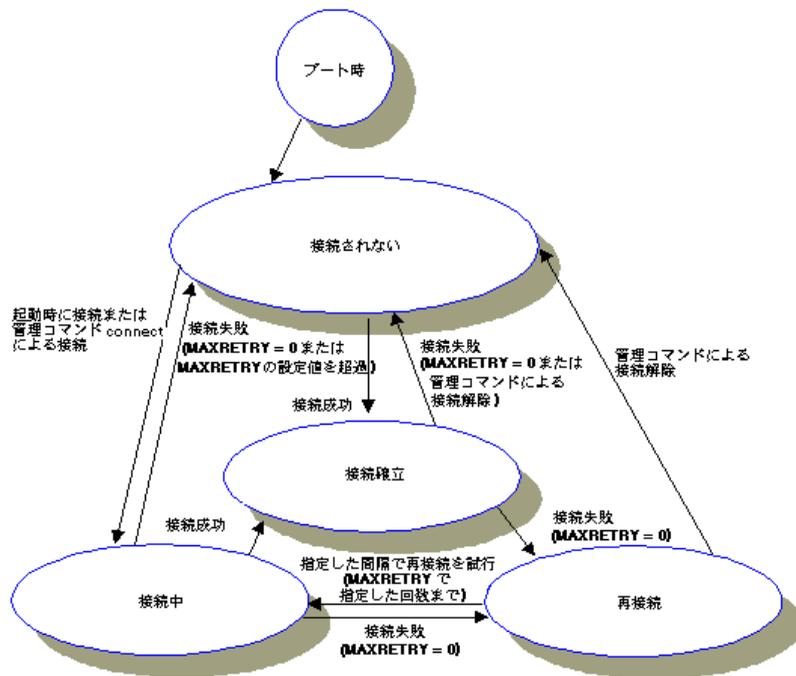
起動時に接続する (ON_STARTUP 方針)

接続方針が ON_STARTUP の場合、ドメイン・ゲートウェイは、ゲートウェイ・サーバの初期化時にリモート・ドメインへの接続を試みます。デフォルトでは、60 秒おきに、失敗した接続が再試行されるよう設定されています。RETRY_INTERVAL パラメータを使用して、再接続の間隔を変更することもできます。この方針では、Dynamic Status が呼び出されます。

```
CONNECTION_POLICY=ON_STARTUP
```

次の図は、ON_STARTUP が設定されている場合に、ゲートウェイによってどのように接続が確立されるかを示しています。

図 2-5 ON_STARTUP による接続



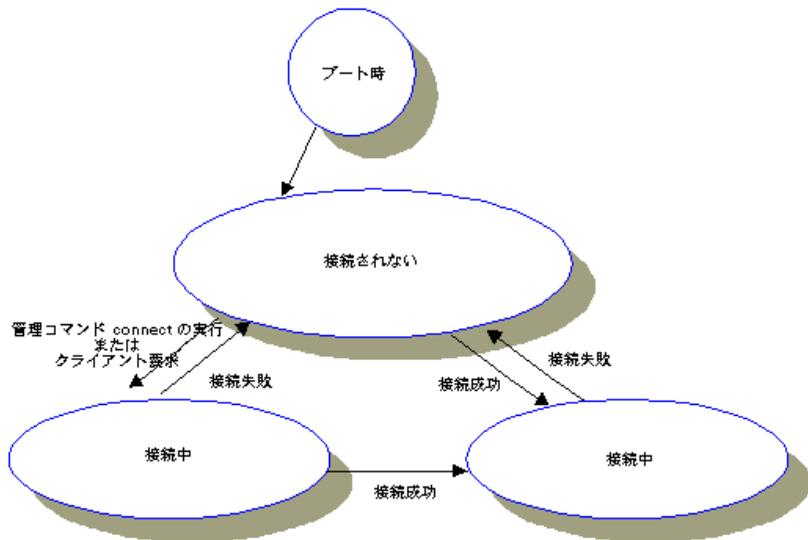
クライアントから要求されたときに接続する (ON_DEMAND 方針)

接続方針が ON_DEMAND の場合は、クライアントがリモート・サービスを要求したとき、または管理コマンド "connect" が実行されたとき、のいずれかの場合に接続が試行されます。CONNECTION_POLICY のデフォルト設定は、ON_DEMAND です。ON_DEMAND が接続方針として設定されていると、再接続は行われません。この方針では、Dynamic Status は呼び出されません。

CONNECTION_POLICY=ON_DEMAND

次の図は、ON_DEMAND が設定されている場合に、ゲートウェイによってどのように接続が確立されるかを示しています。

図 2-6 ON_DEMAND による接続



メッセージの受信時にのみ接続する (INCOMING_ONLY 方針)

接続方針が `INCOMING_ONLY` の場合、ドメイン・ゲートウェイは起動時にリモート・ドメインへの接続を試みません。`INCOMING_ONLY` が接続方針として設定されていると、再接続は行われません。この接続方針では、Dynamic Status が呼び出されます。

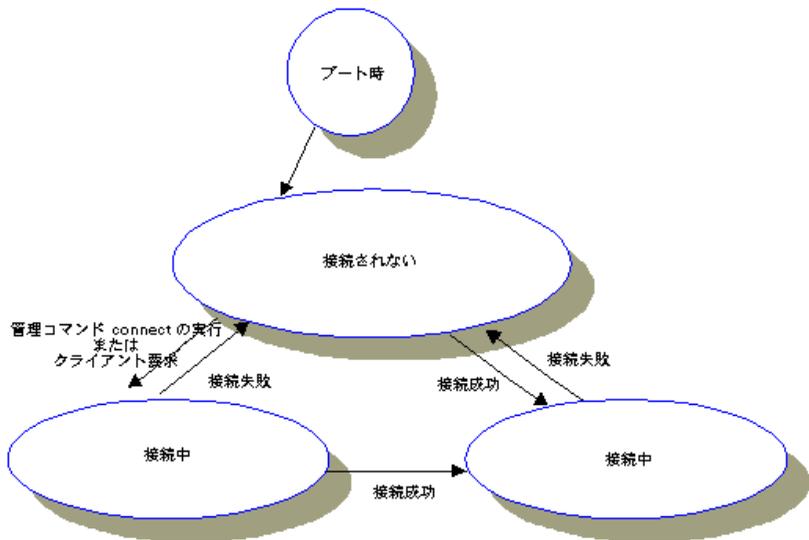
この方針を使用するには、Domains コンフィギュレーション・ファイルに次の行を入力します。

```
CONNECTION_POLICY=INCOMING_ONLY
```

注記 `dmadmin connect` コマンドを使用して、手動で接続を確立することもできます。

次の図は、`INCOMING_ONLY` が設定されている場合に、ゲートウェイによってどのように接続が確立されるかを示しています。

図 2-7 INCOMING_ONLY による接続 (受信時接続を受け付ける)



再接続間隔を設定する (ON_STARTUP のみ)

CONNECTION_POLICY パラメータに ON_STARTUP を設定すると、再接続間隔の設定機能が有効になります。この機能により、ドメイン・ゲートウェイからリモート・ドメインへの接続が失敗しても、自動的に再試行されます。管理者は、自動再接続の頻度を制御することができます。これは、ゲートウェイが次に再接続を行うまでの時間 (秒単位) を指定することによって実現できます。再接続間隔を指定するには、Domains コンフィギュレーション・ファイルの DM_LOCAL_DOMAINS セクションで RETRY_INTERVAL パラメータを設定します。

RETRY_INTERVAL=number_of_seconds

注記 指定できる値は 0 ~ 2147483647 秒です。

接続方針が ON_STARTUP であり、RETRY_INTERVAL パラメータに値が指定されていない場合は、デフォルト値の 60 秒が使用されます。

接続方針が ON_STARTUP の場合に限り、RETRY_INTERVAL パラメータは有効になります。それ以外の接続方針 (ON_DEMAND および INCOMING_ONLY) では、再接続は行われません。

最大再接続回数を設定する

MAXRETRY パラメータに値を割り当てることにより、接続の中止までにドメイン・ゲートウェイがリモート・ドメインへ再接続を試みる回数を指定することができます。最小値は 0 であり、最大値は、MAXLONG パラメータの値です。デフォルトでは最大値に設定されています。

- MAXRETRY に 0 を指定すると、自動再接続処理は行われません。サーバは、リモート・ゲートウェイへの接続を自動的に行いません。
- MAXRETRY に数値を指定すると、ゲートウェイは、指定された回数の再接続を行った後、接続を中止します。

注記 RETRY_INTERVAL は、SCANUNIT の倍数に切り上げられます。

- MAXRETRY=MAXLONG を指定すると、再接続処理が無限に繰り返されるか、または接続が確立されるまで繰り返されます。

MAXRETRY パラメータは、接続方針が ON_STARTUP の場合のみ有効です。それ以外の接続方針 (ON_DEMAND および INCOMING_ONLY) では、再接続は行われません。

表 2-1 MAXRETRY パラメータと RETRY_INTERVAL パラメータの設定例

設定値	結果
CONNECTION_POLICY=ON_STARTUP RETRY_INTERVAL=30 MAXRETRY=3	ゲートウェイは、30 秒おきに再接続を 3 回試行した後、接続を中止します。
CONNECTION_POLICY=ON_STARTUP MAXRETRY=0	ゲートウェイは、初期化時に接続を試行し、最初の試行が失敗した場合は再接続を行いません。
CONNECTION_POLICY=ON_STARTUP RETRY_INTERVAL=30	ゲートウェイは、接続が確立されるまで 30 秒おきに接続を試行します。

関連項目

- 第 2 章の 52 ページ「Domains 間の接続を制御する」
- 第 2 章の 55 ページ「Domains レベルのフェイルオーバーとフェイルバックを設定する」
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の DMCONFIG(5)

Domains 間の接続を制御する

管理者は、ドメイン間で確立する接続の数を制御できます。また、ローカル・ドメインとリモート・ドメインとの間の接続を解除することもできます。

Domains 間の接続を確立する

ローカル・ゲートウェイからリモート・ドメインの接続を確立するには、次のように、`connect (co)` サブコマンドを指定して `dmadmin` コマンドを実行します。

```
dmadmin co -d local_domain_name
```

デフォルトでは、指定したローカル・ドメインと、ローカル・ゲートウェイに対して設定されたすべてのリモート・ドメインとの間で接続が確立されます。1つのリモート・ドメインに対してのみ接続を確立したい場合は、`-R` オプションを指定して、そのドメインをコマンド行で次のように指定します。

```
dmadmin co -d local_domain_name -R remote_domain_name
```

接続に失敗した場合、再接続が行われるようドメインを設定してあれば、自動再接続機能により再接続が繰り返されます。

Domains 間の接続を解除する

ローカル・ゲートウェイとリモート・ドメインとの間の接続を解除するには（ただし、自動再接続機能は無効）、次のように、`disconnect (dco)` サブコマンドを指定して `dmadmin` コマンドを実行します。

```
dmadmin dco -d local_domain_name
```

デフォルトでは、ローカル・ゲートウェイに対して設定されたすべてのリモート・ドメインとの接続が解除されます。1つのリモート・ドメインとの接続だけを解除したい場合は、`-R` オプションを指定して、接続解除するドメインをコマンド行で次のように指定します。

```
dmadmin dco -d local_domain_name -R remote_domain_name
```

このコマンドにより、自動再接続機能は停止されます。このコマンドの実行時にアクティブな接続があるかどうかは無関係です。

接続状況のレポートを作成する

`printdomain` コマンドを使用すると、接続状況および再試行されている接続についてのレポートを作成できます。`connect` コマンドを実行すると、接続に成功したかどうかレポートされます。また、`printdomain` コマンドを使用すると、指定したローカル・ドメインに関する情報を出力できます。この情報には、リモート・ドメインのリスト、接続されているリモート・ドメインのリスト、および接続を再試行しているリモート・ドメインのリストが含まれます。

次の例は、`dmadmin` セッションを示しています。ここでは、`LDM` と呼ばれるローカル・ドメインに対して `printdomain` コマンド (省略形 `pd`) を実行しています。

```
$ dmadmin
dmadmin - Copyright (c) 1996 BEA Systems, Inc.
Portions * Copyright 1986-1997 RSA Data Security, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
TUXEDO is a registered trademark.

pd -d LDM
Local domain :LDM
  Connected domains:
    Domainid:RDM1
  Disconnected domains being retried:
    Domainid:RDM2

dco -d LDM -R RDM1
Operation completed successfully.Use printdomain(pd) to obtain results.

dco -d LDM -R RDM2
Operation completed successfully.Use printdomain(pd) to obtain results.

co -d LDM -R RDM3
Operation completed successfully.Use printdomain(pd) to obtain results.

pd -d LDM
Local domain :LDM
  Connected domains:
    Domainid:RDM3
```

Domains 環境でのフェイルオーバーとフェイルバックを設定する

Domains 環境では、2 種類のフェイルオーバーを実行できます。つまり、リンク・レベルのフェイルオーバーと Domains レベルのフェイルオーバーです。この節では、これらについて説明します。

- 第 2 章の 54 ページ「リンク・レベルのフェイルオーバーをサポートするよう Domains をコンフィギュレーションする」
- 第 2 章の 55 ページ「Domains レベルのフェイルオーバーとフェイルバックを設定する」

使用中のドメインでフェイルオーバーとフェイルバックの機能を有効にするには、まず Domains コンフィギュレーション・ファイルで、この機能がサポートされるように設定する必要があります。

Domains コンフィギュレーション・ファイルの詳細については、『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の DMCONFIG (5) を参照してください。

リンク・レベルのフェイルオーバーをサポートするよう Domains をコンフィギュレーションする

リンク・レベルのフェイルオーバーとは、プライマリ・リンクが失敗したときに、代替のネットワーク・リンクをアクティブにするメカニズムです。リンク・レベルのフェイルオーバーを使用するには、プライマリ・ゲートウェイと代替ゲートウェイを異なるリモート・ドメインに常駐させる（つまり、ゲートウェイのミラーリングを使用する）必要があります。現在、リンク・レベルのフェイルオーバーでは、同一のゲートウェイに対する複数の代替リンクはサポートされていません。

リンク・レベルのフェイルオーバーをインプリメントするには、次のように、Domains コンフィギュレーション・ファイル (DMCONFIG) の DM_TDOMAINS セクションで指定します。

```
*DM_TDOMAINS
RDOM1 NWADDR=//addr1:0
RDOM1 NWADDR=//addr2:0
```

最初のエントリは、リモート・ドメイン `RDOM1` に対するプライマリ・ネットワーク・リンクを示しています。2 番目のエントリは、代替リンクを示しています。

リンク・レベルのフェイルバックは手動で行います。プライマリ・リンクが復元されると、管理者は代替リンクを手動で解除する必要があります。代替リンクを手動で解除すると、処理中の要求がエラーになり、プライマリ・リンクで新規トラフィックが再開される場合があります。

注記 ゲートウェイのミラーリングの詳細については、『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `DMCONFIG(5)` を参照してください。

Domains レベルのフェイルオーバーとフェイルバックを設定する

Domains レベルのフェイルオーバーとは、プライマリ・リモート・ドメインで障害が検出されたときに、要求を代替リモート・ドメインに転送するメカニズムです。プライマリ・リモート・ドメインが復元されると、処理はプライマリ・リモート・ドメインに戻されます。

このフェイルオーバーとフェイルバックのレベルは、Dynamic Status に依存します。Domains レベルのフェイルオーバーとフェイルバックを有効にするには、`CONNECTION_POLICY` を `ON_STARTUP` または `INCOMING_ONLY` に設定してください。

Domains レベルのフェイルオーバーとフェイルバックでは、ネットワーク接続が確立されているリモート・ドメインはアクセス可能と定義され、ネットワーク接続が確立されていないリモート・ドメインはアクセス不可能と定義されます。

Domains レベルのフェイルオーバーとフェイルバックの前提条件

Domains レベルのフェイルバックを使用するには、`CONNECTION_POLICY` パラメータの値として `ON_STARTUP` または `INCOMING_ONLY` を指定します。

`ON_DEMAND` は、Domains レベルのフェイルバックには不適当です。この接続方針では、リモート・ドメインが常に有効であると見なされるためです。`ON_STARTUP` または `INCOMING_ONLY` を接続方法として指定しない場合、サーバは `RDOM` パラメータで指定した代替のリモート・ドメインに処理を転送できません。

注記 ネットワーク接続が確立されているリモート・ドメインはアクセス可能です。ネットワーク接続が確立されていないリモート・ドメインはアクセス不可能です。

フェイルオーバーをサポートするよう Domains をコンフィギュレーションする

フェイルオーバーをサポートするには、Domains コンフィギュレーション・ファイルで、特定のサービスを実行するリモート・ドメインのリストを指定する必要があります。特に、DM_REMOTE_SERVICES セクションで、RDOM パラメータの値などのリストを指定する必要があります。次のように、代替のドメインを指定することもできます。

```
RDOM=identifier_1, identifier_2, identifier_3
```

例

3 つのリモート・ドメイン (R1 (プライマリ・リモート・ドメイン)、R2、および R3) から TOUPPER と TOUPPER2 のサービスを利用することができるとします。Domains コンフィギュレーション・ファイルに次のエントリを追加します。

```
*DM_REMOTE_SERVICES
DEFAULT:RDOM=R1, R2, R3
TOUPPER
TOUPPER2
```

フェイルバックをサポートするよう Domains をコンフィギュレーションする

プライマリ・リモート・ドメインへのネットワーク接続が再び確立されると、フェイルバックが発生します。フェイルバックは、次の原因によって発生します。

- 自動再接続 (ON_STARTUP の場合のみ)
- 受信時接続
- 手動による `dmadmin connect` コマンドの実行

注記 自動再接続では、接続の再試行をオンに設定してください (指定例: `MAXRETRY>0`)

3 CORBA Domains の計画と コンフィギュレーション

BEA Tuxedo CORBA 環境の Domains は、コア ATMI BEA Tuxedo 環境の Domains を拡張したものです。ドメインは、完全に管理可能な構造体です。ドメインを参照するプログラミング・インターフェイスはありません。ドメイン関連の設定は、すべてコンフィギュレーション・ファイルで指定されます。この設定を行うのは管理者だけです。

この章は、以下の節で構成されています。

- 複数の CORBA ドメインを設定する
- CORBA ドメイン・コンフィギュレーションの種類
- 例：複数の CORBA ドメインの設定

マルチ・ドメイン環境 CORBA Domains の概要

企業には、地理的に分けられたアプリケーションや、業務別のアプリケーションなど、さまざまなアプリケーションがあるため、多数の独立したドメインが存在します。各ドメインは通常、個別のユニットとして管理されています。つまり、地理的条件（マシンが存在する場所）や組織上の部門（経理部門、生産部門、発送部門など）ごとに管理されています。

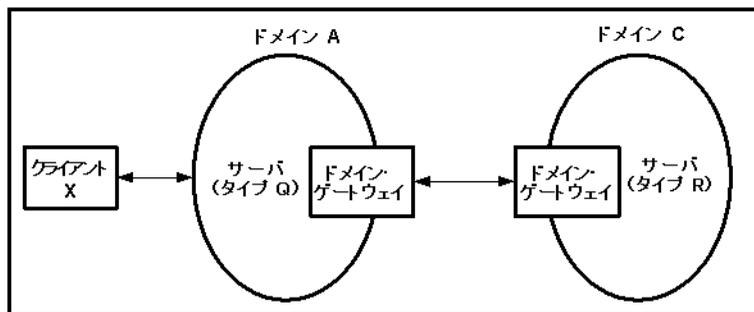
各企業では、このようなドメイン上のさまざまなアプリケーションを統合することが求められています。通常、1つのドメインを拡張して企業全体の業務をカバーすることは可能です。ただし、1つのドメイン内のマシン数やサービス数を大幅に増やすのは、現実的ではありません。シングル・ドメインでは、ドメイン全体を管理する必要があるため、コンフィギュレーションが急速に肥大化し、アプリケーションの開発やインプリメントより多くの労力がかかります。

したがって、ドメインを管理しやすいサイズに抑えるには、アプリケーションを複数のドメインに分散し、さらに、各ドメインのアプリケーションからほかのドメインのサービスにアクセスできるようにする必要があります。このようなドメイン間の通信機能を総称して「BEA Tuxedo ドメイン」と呼びます。

ドメイン間通信

次の図は、単純なマルチ・ドメイン・コンフィギュレーションを示しています。

図 3-1 マルチ・ドメイン・コンフィギュレーション



以下は、クライアント X とドメイン A 間のシングル・ドメイン通信の手順です。

1. クライアント X は、Bootstrap オブジェクトを使用してドメイン A に接続します。クライアント・アプリケーションは、Bootstrap オブジェクトを使用して FactoryFinder を検索し、その FactoryFinder を使用して、ファクトリにタイプ Q のオブジェクトを要求します (FactoryFinder の呼び出しは、ドメイン A の呼び出しになります)。
2. FactoryFinder からファクトリが返されると、クライアントはドメイン A でそのファクトリを呼び出します。
3. ファクトリは、タイプ Q のオブジェクト・リファレンス (Q1) を返します。
4. クライアントは、ドメイン A でオブジェクト Q1 を呼び出します。

注記 上の手順では、クライアントはオブジェクトの位置や、所属先のドメインを認識していません。クライアント・マシンは単純な構造で、必要なインフラストラクチャも少ないため (大部分はスタンドアロン)、クライアントをドメイン A に接続するための管理は比較的単純です。クライアントの主な管理タスクは BEA Tuxedo ドメインへの接続ですが、実際の管理タスクは、ドメイン A 内の ISL のアドレスを設定することです。

マルチ・ドメイン通信では、Q1 はドメイン C にあるオブジェクト R1 のサービスを要求するので、ドメインの境界をまたいで上の手順 1 ~ 4 に対応する処理を実行する必要があります。実際の手順は以下のようにになります。

1. オブジェクト Q1 は Bootstrap オブジェクトを使用して FactoryFinder を検索し、その FactoryFinder を使用して、ファクトリにタイプ R のオブジェクトを要求します。
2. FactoryFinder からドメイン C のファクトリに対するリファレンスが返されると、オブジェクト Q1 がそのファクトリを呼び出します。
3. ファクトリは、タイプ R のオブジェクト・リファレンス (R1) を返します。
4. オブジェクト Q1 がオブジェクト R1 を呼び出します。

注記 クライアント X で、オブジェクト Q1 がドメイン C のファクトリとオブジェクトを取得するには、何らかの管理タスクが必要です。図 3-1 に示すように、ドメイン間通信にはドメイン・ゲートウェイを使用します。ドメイン・ゲートウェイは、ドメイン内のシステム・サーバです。

システム・サーバは、ユーザ定義のサーバとは異なり、ネーム・サーバ、FactoryFinder、ISL などと共に BEA Tuxedo 製品の一部として提供されています。ドメイン・ゲートウェイは、ドメインの接続ポイントになるので、概念的には ISL と似ています。ただし、ISL と異なり、ドメイン・ゲートウェイは別のドメイン・ゲートウェイに接続し、それ自身がドメインの接続ポイントになります。つまり、ドメイン・ゲートウェイの役割は、別のドメイン・ゲートウェイに接続することです。したがって、2つのドメイン・ゲートウェイは相互に協調し合い、別のドメインに存在するオブジェクトの呼び出しが、適切なドメインに確実にルーティングされるようにします。

ドメイン・ゲートウェイがこのような役割を果たすには、適切なコンフィギュレーションが必要です。次の節では、このコンフィギュレーションについて説明します。

マルチ・ドメイン・コンフィギュレーションの要素

マルチ・ドメイン・コンフィギュレーションは、以下の要素で構成されます。

- BEA Tuxedo コンフィギュレーション・ファイル

UBBCONFIG ファイルでは、ドメイン名を指定し、ドメイン・ゲートウェイ・サーバのグループ・エントリとサービス・エントリを設定します。UBBCONFIG ファイルでは、ドメイン・ゲートウェイの属性は指定しません。これらの属性はすべて DMCONFIG ファイルで指定します。

- ドメイン・コンフィギュレーション・ファイル

ドメイン・コンフィギュレーション・ファイル (DMCONFIG) では、ローカル・ドメインに接続されるリモート・ドメインを指定します。DMCONFIG ファイルがないと接続が確立されません。

- FactoryFinder ドメイン・コンフィギュレーション・ファイル

1 つまたは複数のドメインに接続されたドメインごとに、1 つの FactoryFinder ドメイン・コンフィギュレーション・ファイル (`factory_finder.ini`) が必要です。ドメインが別のドメインに接続されていない場合、このファイルは必要ありません。

このファイルでは、ドメインの境界をまたいで検索するファクトリを指定します。`factory_finder.ini` ファイルと `DMCONFIG` ファイルには、同じ接続先ドメインの情報を指定し、どちらも同じ接続を実現をできるようにする必要があります。

- リモート・ドメインでのオブジェクトの呼び出し

BEA Tuxedo ドメインの主な機能は、CORBA ドメインのアプリケーションが、別のドメインのオブジェクトを呼び出す際に、クライアントまたはサーバ・アプリケーションがドメインを意識せずに処理できるようにすることです。ドメインの境界をまたぐこのような呼び出しを、アプリケーションがドメインの境界を意識せずに実行するには、さまざまなコンフィギュレーション情報が必要です。

リモート・ドメインのオブジェクト・リファレンスを呼び出すことができるかどうかは、ドメインごとに、`UBBCONFIG`、`DMCONFIG`、および `factory_finder.ini` の 3 つのコンフィギュレーション・ファイルが適切に設定され、これらの内 `DMCONFIG` と `factory_finder.ini` の 2 つのコンフィギュレーション・ファイルがドメイン間で対応しているかどうかによって決まります。ドメインの数が増えるほど、このような調整も複雑になります。

- リモート・ドメインのオブジェクト・リファレンス

オブジェクト・リファレンスを使用して、ローカル・ドメインまたはリモート・ドメインを指定できます。FactoryFinder がリモート・ドメイン内のファクトリに対するオブジェクト・リファレンスを返すと、通常リモート・ドメインの参照が発生します。また、ファクトリがそのリモート・ドメインのオブジェクト・リファレンスを作成して返しても、リモート・ドメインの参照が発生します (ただし、ファクトリのドメインに対するローカル参照になります)。

注記 アプリケーションは、オブジェクト・リファレンスのドメインを認識しません。つまり、アプリケーション側では、オブジェクト・リファレンスが参照するドメインを見つけられません。したがって、リモート・ドメインに対するオブジェクト・リファレンスの呼び出しは、アプリケーションに対して透過的に行われます。この透過性により、管理者は各ドメインのサービスを自由に設定し、複数のドメイン間にリソースを分散することができます。ドメインに関する情報がアプリケーションに存在する場合、コンフィギュレーションを変更すると、アプリケーションも同様に作成し直す必要があります。

- FactoryFinder

ローカル・ドメインのサーバから別のドメインのオブジェクト・リファレンスを取得する場合、アプリケーション側では、ローカル・ドメインのオブジェクト・リファレンスを取得する場合と同じ方法で FactoryFinder を使用します。アプリケーション側では、FactoryFinder が別のドメインのファクトリに対するリファレンスを返したことが認識されないため、同じ方法が使用されます。コンフィギュレーション・ファイルには、このことは示されません。

FactoryFinder またはファクトリによってオブジェクト・リファレンスが取得されると、そのオブジェクト・リファレンスを任意の転送先に渡すことができます。たとえば、ローカル・ドメインのオブジェクトに渡したり、クライアントに返したり、別のドメインに渡すことができます。

複数の CORBA ドメインを設定する

次の 3 つのコンフィギュレーション・ファイルを使用して複数のドメインを設定します。

- コンフィギュレーション・ファイル (UBBCONFIG)
- ドメイン・コンフィギュレーション・ファイル (DMCONFIG)
- FactoryFinder ドメイン・コンフィギュレーション・ファイル (factory_finder.ini)

コンフィギュレーション・ファイル

複数のドメインを設定するには、UBBCONFIG ファイルで以下のパラメータを指定する必要があります。

- ドメイン名
- ゲートウェイ・グループ
- ゲートウェイ・サービス

ドメイン名

シングル・ドメイン (スタンドアロン・ドメイン) ではこのパラメータは不要ですが、別のドメインに接続されたドメインでは、DOMAIN ID を指定する必要があります。このパラメータは、UBBCONFIG ファイルの RESOURCES セクションで次のように指定します。

```
DOMAIN ID = <domain-name>
```

<domain-name> は、1 ~ 13 文字の範囲で指定しなければなりません。次に例を示します。

```
DOMAIN ID = headquarters
```

<domain-name> は、関連する DMCONFIG ファイルの DM_REMOTE_SERVICES および DM_LOCAL_SERVICES セクションで参照される名前です。<domain-name> は次の形式で参照されます。

```
"//<domain-name>"
```

二重引用符も参照の一部として必要です。スラッシュ (//) は、この名前が BEA Tuxedo ATMI ドメインではなく BEA Tuxedo CORBA ドメインに適用されることを意味します。次に例を示します。

```
"//headquarters"
```

注記 企業内の各ドメインには、一意の <domain-name> を指定する必要があります。

ゲートウェイ・グループとゲートウェイ・サービス

その他のシステム・サービス同様、ゲートウェイにもグループ名とサービス名を指定する必要があります。たとえば、GROUPS セクションで次のように指定します。

```
LGWGRP      GRPNO=4      LMID=LDMO
```

この例では、LGWGRP はユーザが選択した名前です (ここでは「Local Gateway Group」の略と考えられます)。

ドメイン・ゲートウェイのサービス名は GWTDOMAIN で、ほかのグループ同様、サーバ・グループおよびサーバ ID と関連付ける必要があります。サービス名は、選択したサーバ・グループ名と関連付けられた SERVERS セクションで指定します。次に例を示します。

```
GWTDOMAIN SRVGRP=LGWGRP SRVID=1
```

これにより、ドメイン・ゲートウェイが使用されていること、および DMCONFIG ファイルに関連情報があることが BEA Tuxedo CORBA サーバに通知されます。

ドメイン・コンフィギュレーション・ファイル (DMCONFIG)

ドメインごとに 1 つの DMCNFIG ファイルが必要です。このファイルでは、ローカル・ドメイン (DMCNFIG ファイルが存在するドメイン) とリモート・ドメイン (それ以外のドメイン) の関係を指定します。DMCNFIG ファイルには、コア BEA Tuxedo ドメインと BEA Tuxedo CORBA ドメインに関する情報が格納されています。

以下の節では、BEA Tuxedo CORBA ドメインに適用される情報について説明します。DMCNFIG ファイルに関するほかの説明では、ローカル・ドメインとリモート・ドメイン間の通信は BEA Tuxedo ATMI サービスをベースにしていると記載されていますが、この概念は BEA Tuxedo CORBA 環境には適用されません。BEA Tuxedo CORBA 環境での「サービス」名は、BEA Tuxedo CORBA 要求を処理する別の BEA Tuxedo ドメインの名前になります。

DMCNFIG ファイルは 8 つのセクションで構成されますが、その内 DM_ROUTING は BEA Tuxedo CORBA 環境には適用されません。それ以外の 7 つのセクションは BEA Tuxedo CORBA 環境に適用されますが、BEA Tuxedo ATMI パラメータの大部分は使用されません。この 7 つのセクションとは、DM_RESOURCES、DM_LOCAL_DOMAINS、DM_REMOTE_DOMAINS、DM_LOCAL_SERVICES、DM_REMOTE_SERVICES、DM_ACCESS_CONTROL、および DM_TDOMAIN です。

次の節では、リスト 3-1 に示す DMCNFIG ファイルの例を説明します。

リスト 3-1 DMCNFIG ファイルの例

```
#
# BEA Tuxedo CORBA ドメイン・コンフィギュレーション・ファイル
#
*DM_RESOURCES
VERSION=Experimental8.9

*DM_LOCAL_DOMAINS
LDM GWGRP=LWGRP TYPE=TDOMAIN DOMAINID="MUTT"

*DM_REMOTE_DOMAINS
TDOM1 TYPE=TDOMAIN DOMAINID="JEFF"

*DM_TDOMAIN
LDM NWADDR="//MUTT:2507"
TDOM1 NWADDR="//JEFF:3186"

*DM_LOCAL_SERVICES
"/MUTT"
```

```
*DM_REMOTE_SERVICES
" //JEFF"      RDOM=TDOM1
```

DM_RESOURCES

DM_RESOURCES セクションには、VERSION フィールドしかありません。このフィールドはソフトウェアによってチェックされません。アプリケーションに関する情報をユーザが入力するために用意されたフィールドです。

```
*DM_RESOURCES
VERSION=Experimental8.9
```

DM_LOCAL_DOMAINS

DM_LOCAL_DOMAINS セクションでは、ローカル・ドメインに外部からアクセスする際のゲートウェイの属性を指定します。このセクションには、UBBCONFIG ファイルで指定されたゲートウェイ・グループごとにエントリがあり、これにより、ほかのドメインからローカル・ドメインへのアクセスを実現できます。各エントリでは、グループで実行中のドメイン・ゲートウェイ・プロセスに必要なパラメータを指定します。

エントリの形式は次のとおりです。

LDOM 必須パラメータ [オプション・パラメータ]

LDOM は、ローカル・ドメインのゲートウェイの参照に使用される識別子です。LDOM は、企業内のすべての LDOM と RDOM エントリで (相互接続されるすべてのドメイン間で) 一意でなければなりません。LDOM は、UBBCONFIG ファイルで指定された <domain-name> またはゲートウェイ・グループと同じではありません。LDOM は、DMCONFIG ファイル内でのみ使用される名前、UBBCONFIG ファイルが変更された場合の影響を限定します (UBBCONFIG ファイルの変更は、DMCONFIG のこの部分にのみ反映されます)。

以下は、必須パラメータです。

GWGRP = identifier

このパラメータは、このローカル・ドメインを表すゲートウェイ・サーバ・グループの名前 (UBBCONFIG ファイルで指定される名前) を指定します。

TYPE = TDOMAIN

TYPE パラメータは、BEA Tuxedo CORBA 環境のドメインを使用することを指定するために必要です。

```
DOMAINID = string
```

DOMAINID パラメータは、セキュリティ上の目的でローカル・ドメインを識別するために指定します。GWGRP のゲートウェイ・サーバ・グループは、セキュリティ・チェック時にこの文字列を使用します。UBBCONFIG ファイルの RESOURCES セクションにある <domain-name> と対応させる必要はありません。DOMAINID は、ローカル・ドメインとリモート・ドメインの両方にわたり、一意でなければなりません。string の値には、一連の文字 "BA.CENTRAL01" など) か、0x で始まる 16 進数 "0x0002FF98C0000B9D6" など) を指定できます。DOMAINID は、32 以下のオクテットで指定する必要があります。文字列を指定する場合は、32 文字以内で指定する必要があります (最後のヌルを含む)。

たとえば、次の例を考えます。

```
*DM_LOCAL_DOMAINS
LDM GWGRP=LWGRP TYPE=TDOMAIN DOMAINID="MUTT"
```

ここでは、LDM がローカル・ドメインへのアクセス・ポイントとして指定されています。LDM はサービス・グループ LWGRP (UBBCONFIG ファイルで指定) と関連付けられています。ゲートウェイが常にドメイン間のセキュリティ・チェックの対象になる場合、MUTT の名前で通過します。

オプション・パラメータは、ドメイン・ゲートウェイの操作で使用する資源と最大値 / 最小値を指定します。これらのパラメータの詳細については、『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の dmconfig(5) リファレンス・ページを参照してください。

DM_REMOTE_DOMAINS

DM_REMOTE_DOMAINS セクションでは、リモート・ドメインへのゲートウェイの属性を指定します。このセクションには、UBBCONFIG ファイルで定義された、リモート・ドメインに要求を送信するゲートウェイ・グループごとに 1 つのエントリがあります。各エントリでは、グループで実行中のドメイン・ゲートウェイ・プロセスに必要なパラメータを指定します。

エントリの形式は次のとおりです。

RDOM 必須パラメータ

RDOM は、リモート・ドメインへのアクセスを提供するゲートウェイの参照に使用される識別子です。RDOM は、企業内のすべての LDOM と RDOM エントリで（相互接続されるすべてのドメイン間で）一意でなければなりません。RDOM は、UBBCONFIG ファイルで指定された <domain-name> またはゲートウェイ・グループと同じではありません。RDOM は、DMCONFIG ファイル内でのみ使用される名前で、UBBCONFIG ファイルが変更された場合の影響を限定します（UBBCONFIG ファイルの変更は、DMCONFIG のこの部分にのみ反映されます）。

必須パラメータは以下のとおりです。

TYPE = TDOMAIN

TYPE パラメータは、BEA Tuxedo CORBA 環境のドメインを使用することを指定するために必要です。

DOMAINID = string

DOMAINID パラメータは、セキュリティ上の目的でリモート・ドメインを識別するために指定します。ゲートウェイは、セキュリティ・チェック時にこの文字列を使用します。DOMAINID は、UBBCONFIG ファイルの RESOURCES セクションにある <domain-name> と対応させる必要はありません。DOMAINID は、ローカル・ドメインとリモート・ドメインの両方にわたり、一意でなければなりません。string の値には、一連の文字 "BA.CENTRAL01" などが、"0x" で始まる 16 進数 "0x0002FF98C0000B9D6" など) を指定できます。DOMAINID は、32 以下のオクテットで指定する必要があります。文字列を指定する場合は、32 文字以内で指定する必要があります（最後のヌルを含む）。リモート・ドメインに関連付けられたエントリは、複数回指定できます。最初の指定（リモート・ドメインへの接続が最初に試行されときの接続先）は、一次アドレスと見なされます。一次エントリの NWADDR を使用してネットワーク接続が確立できない場合、2 番目のエントリに関連付けられた NWADDR が使用されます。（NWADDR は物理アドレスになります。DM_TDOMAIN を参照してください）。

たとえば、次の例を考えます。

```
*DM_REMOTE_DOMAINS
TDOM1 TYPE=TDOMAIN DOMAINID="JEFF"
```

TDOM1 がゲートウェイのアクセス・ポイント名として指定されています。ゲートウェイが、通信相手のゲートウェイとともに常にドメイン間セキュリティ・チェックの対象となる場合、ゲートウェイは通信相手が JEFF の名前で通過すると見なします。

DM_TDOMAIN

DM_TDOMAIN セクションでは、BEA Tuxedo CORBA ドメインをインプリメントするゲートウェイのネットワーク・アドレス情報を定義します。リモート・ドメインからの要求を受信するドメイン・ゲートウェイ、およびリモート・ドメインに要求を送信するドメイン・ゲートウェイごとに 1 つのエントリが必要です。

各エントリの形式は次のとおりです。

DOM 必須パラメータ [オプション・パラメータ]

DOM は、ローカル・ドメイン・アクセス・ポイント (DM_LOCAL_DOMAINS セクションの LDOM) またはリモート・ドメイン・アクセス・ポイント (DM_REMOTE_DOMAINS セクションの RDOM) を識別する値です。

以下は、必須パラメータです。

NWADDR = string

このパラメータを使用して、ローカル・ドメインまたはリモート・ドメインに関連するネットワーク・アドレスを指定します。ローカル・ドメインと関連がある場合は、NWADDR を使用して、ほかのドメインからの接続を受け付けます。リモート・ドメインと関連がある場合は、NWADDR を使用して接続を開始します。このパラメータは、プロセスが接続指示受け付けアドレスとして使用するネットワーク・アドレスを指定します。ドメイン・ゲートウェイに対する接続指示受け付けアドレスは、アプリケーションに参加しているほかのゲートウェイ・プロセスの通信手段となります。文字列の形式が "0xhex-digits" または "\\xhex-digits" の場合、偶数の有効な 16 進数を含める必要があります。これらの形式は、TCP/IP アドレスを含む文字配列に内部変換されます。このアドレスは、次の 2 つの形式のいずれかで指定することもできます。

```
//host.name:port_number"
"###.###.###:port_number"
```

最初の形式の場合、hostname は、アドレスがバインドされる時 (gethostbyname(3c) でローカル設定名前解決機能を使用) に TCP/IP ホスト・アドレスに解決されます。"###.###.###" はドットで区切った 10 進数の形式で、各 # は 0 から 255 までの 10 進数です。

Port_number は、0 ~ 65535 の 10 進数です (指定された文字列の 16 進表現)。次に例を示します。

```
*DM_TDOMAIN
  LDOM  NWADDR="//MUTT:2507"
  TDOM1 NWADDR="//JEFF:3186"
```

上の例では、最初のエントリでゲートウェイにドメイン・アクセス名 LDOM (UBBCONFIG で指定されたローカル・ゲートウェイ・グループ LGWGRP に対応) が指定されています。DM_LOCAL_DOMAINS で LDOM が指定されているので、このゲートウェイはほかのドメインからの要求を受け付けるように設定されていることがわかります。このゲートウェイはアドレス "//MUTT:2507" で要求をリスンします。同様に、2 番目のエントリでは、リモート・ドメインに要求を送信する際のドメイン・アクセス名 TDOM1 (DM_REMOTE_DOMAINS に対応) が指定されています。この場合、TDOM1 に関連付けられたゲートウェイが、アドレス "//JEFF:3186" に要求を送信します。

オプション・パラメータの詳細については、『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の dmconfig(5) リファレンス・ページを参照してください。

DM_REMOTE_SERVICES

DM_REMOTE_SERVICES セクションでは、リモート・ドメインのゲートウェイのその他の属性を指定します。各エントリの形式は次のとおりです。

```
service RDOM=<rdom-name>
    [LDOM=<ldom-name>]
    [TRAN_TIME=...]
```

service の形式は次のとおりです。

```
"//<domain-name>"
```

この <domain-name> は、UBBCONFIG ファイルの RESOURCES セクションにある <domain-name> と同じです。各エントリで rdom-name を指定し、オプションで ldom-name を指定します。ゲートウェイは、これらのエントリの属性を使用して、BEA Tuxedo CORBA ドメイン通信に必要なゲートウェイ・ペアを確立します。起動時に、ローカル・ドメインは rdom-name (DM_TDOMAIN セクションで指定されたアドレス) の属性を使用して、ほかのドメインのゲートウェイへの接続を確立します。セキュリティ機能を使用する場合は、相互認証のために rdom-name と ldom-name のその他の属性が使用されます。実行時には、BEA Tuxedo が要求を <domain-name> のドメインに送信することを決定します。この場合、rdom-name で指定されたゲートウェイを使用して、別のドメインに要求が送信されます。

大抵の場合、<domain-name> は rdom-name のアドレスで指定されたドメインの名前になります。この場合、要求が最終目的地のゲートウェイに到達すると、そのドメイン内で処理されます。次に例を示します。

```
*DM_REMOTE_SERVICES
    "//JEFF"      RDOM=TDOM1
```

この場合、ドメイン名 JEFF はアドレス "//JEFF:3186" にあります。このアドレスには、ドメイン名が JEFF に指定された UBBCONFIG ファイルがある場合とない場合があります。このファイルがある場合は、要求を直ちに処理できます。

指定した domain-name に対する要求を、ルーティングの中継点として機能する中間ドメインに送信するエントリを設定することもできます。

オプション・パラメータの `TRANTIME = integer` では、関連するサービスに対するトランザクションを自動的に開始するまでのデフォルトのタイムアウト値を秒単位で指定します。この値は、0 以上 2147483648 未満でなければなりません。デフォルトは 30 秒です。0 を指定すると、マシンの最大タイムアウト値に設定されます。

DM_LOCAL_SERVICES

DM_LOCAL_SERVICES セクションでは、外部からローカル・ドメインへの要求を受け付けるゲートウェイのその他の属性を指定します。

このセクション内にある各行の形式は次のとおりです。

```
service [LDM=<ldom-name>]
        [ACL=...]
```

service の形式は次のとおりです。

```
"//<domain-name>"
```

この <domain-name> は、UBBCONFIG ファイルの RESOURCES セクションにある <domain-name> と同じです。大抵の場合、この名前はゲートウェイが存在するドメインの名前になります。これは、この（ローカル）ドメインがほかのドメインから BEA Tuxedo CORBA 環境の要求を受け付けることを意味します。ローカル・ドメインがルーティングの中継点として機能する場合、異なるドメイン名の要求を受け付けるエントリを設定することもできますが、セキュリティ上必要な場合以外は不要です。

エクスポートされたサービスは、TUXCONFIG ファイルの SERVICES セクション内のエントリのサービスに対して指定された特性か、またはデフォルトの特性を継承します。継承される特性として、LOAD、PRIO、AUTOTRAN、ROUTING、BUFTYPE、TRANTIME があります。

オプション・パラメータの `ACL = identifier` では、アクセス制御リスト (ACL) の名前を指定します。ローカル・ドメインはこのリストを使用して、リモート・ドメインからこのサービスへの要求を制限します。ACL の名前は、DM_ACCESS_CONTROL セクションで定義します。このパラメータを指定しないと、このサービスに対する要求についてアクセス制御が実行されません。

たとえば、次の例を考えます。

```
*DM_LOCAL_SERVICES
"/MUTT"
```

このドメインで MUTT という名前のドメインに対する要求を受け付けるように指定されています。

DM_ACCESS_CONTROL

DM_ACCESS_CONTROL セクションでは、ローカル・ドメインで使用するアクセス制御リストを指定します。このセクション内にある各行の形式は次のとおりです。

ACL_NAME 必須パラメータ

ACL_NAME は、アクセス制御リストを識別するための (識別子の) 名前で、15 文字以下で指定する必要があります。

以下は、必須パラメータです。

```
ACLIST = identifier [,identifier]
```

ACLIST には、1 つ以上のリモート・ドメイン名 (RDOM) をカンマで区切って指定します。ワイルドカード文字 (*) を使用すると、DM_REMOTE_DOMAINS セクションで定義したすべてのリモート・ドメインがローカル・ドメインにアクセスできることを指定できます。

注記 factory_finder.ini ファイルと DMCONFIG ファイルの内容は対応していません。つまり、factory_finder.ini ファイルでファクトリにアクセス可能な別のドメインが宣言されている場合、DMCONFIG ファイルでもそのドメインにアクセスできるように設定されていなければなりません。

factory_finder.ini ファイル

管理者は、別の (リモート) /Domain に存在するファクトリ・オブジェクトで、現在の (ローカル) /Domain で使用可能なファクトリ・オブジェクトを識別する必要があります。これらのファクトリは、FactoryFinder ドメイン・コンフィギュレーション・ファイル (factory_finder.ini ファイル) で識別します。このファイルは ASCII 形式のファイルであり、テキスト・エディタを使用して作成したり、更新できます。

factory_finder.ini ファイルを使用して、ローカル・ドメインで使用可能なリモート CORBA ファクトリを識別することができます。

factory_finder.ini ファイルの形式は、以下に示すように、/Domains を記述する構文に準拠しています。

```
*DM_REMOTE_FACTORIES
  "local_factory_id.factory_kind"
  DOMAINID="domain_id"
  RNAME="remote_factory_id.factory_kind"
  ...
*DM_LOCAL_FACTORIES
  "factory_id.factory_kind"
  ...
```

以下は、CORBA ファクトリ・オブジェクトの構文例です。

```
*DM_REMOTE_FACTORIES
  "AccountFactory.FactoryKind"
  DOMAINID="MyAccountFactoryDomain"
  RNAME="MyAccountFactory.FactoryKind"
```

各項目の説明は次のとおりです。AccountFactory は、ローカル・ドメインの FactoryFinder にファクトリを登録する名前です。MyAccountFactoryDomain は、リモート・ドメインの名前です。MyAccountFactory は、リモート・ドメインの FactoryFinder にファクトリを登録する名前です。

プロセスが開始されると、マスタ NameManager が factory_finder.ini ファイルを読み込みます。マスタ NameManager の起動方法によって、factory_finder.ini ファイルのどの部分が処理されるかが異なります。マスタ NameManager がアプリケーションの起動処理の一部として初期化モードで起動された場合、ファイルの内容全体が処理されます。この場合、DM_REMOTE_FACTORIES セクションの内容が、インポートするファクトリ・オブジェクトに追加されるエントリになります。

これに対し、マスタ NameManager がプロセスの失敗によって再起動された場合、DM_LOCAL_FACTORIES セクションのみ読み出されます。別のドメインにエクスポートされるファクトリ・オブジェクトを制限する情報を再ロードするには、factory_finder.ini ファイルのこのセクションを再度読み出す必要があります。

注記 マスタ NameManager は、プロセスが開始された場合のみ

factory_finder.ini ファイルを読み出すので、たとえば、新しいドメインにインポートするファクトリ・オブジェクトを追加する必要がある場合などに、マスタ NameManager をシャットダウンせずに更新することはできません。

factory_finder.ini ファイルは、そのファイルが存在するドメインに適用されます。このファイルには、DM_REMOTE_FACTORIES と DM_LOCAL_FACTORIES の 2 つのセクションがありますが、どちらかのセクションが存在しない場合や、内容が空の場合もあります。

以下の節では、DM_REMOTE_FACTORIES セクションと DM_LOCAL_FACTORIES セクションの使用方法を説明します。

DM_REMOTE_FACTORIES

DM_REMOTE_FACTORIES セクションは、リモート・ドメインで使用可能なファクトリ・オブジェクトの内、ローカル・ドメインのアプリケーションで使用できるようにインポートされるファクトリ・オブジェクトの情報を提供します。このセクションには、リモート・ファクトリ・オブジェクトの識別子がリストされます。オブジェクト登録時に kind 値が "FactoryInterface" に設定された識別子は、このセクションにリストされなければなりません。たとえば、TP Framework によって識別子 Teller でドメイン "Norwest" に登録されたりモート・ファクトリ・オブジェクトのエントリは、次のように指定されます。

```
*DM_REMOTE_FACTORIES
  "Teller.FactoryInterface"
    DOMAINID="Norwest"
    RNAME="BankTeller.FactoryInterface"
```

RNAME が指定されていない場合、ファクトリ名で factory_kind を指定し、ファクトリ名を引用符で囲まなければなりません。これらの条件を満たさないと、NameManager が適切なファクトリを検出できません。エントリに factory_kind 値がない場合、デフォルト値の "FactoryInterface" が使用されます。以下の例では、ファクトリ・オブジェクトが識別子 Teller でドメイン "Norwest" に登録されています。RNAME が指定されていないので、factory_kind 値が指定され、ファクトリ名が引用符で囲まれていることに注意してください。

```
*DM_REMOTE_FACTORIES
  "Teller.FactoryInterface"
    DOMAINID="Norwest"
```

マルチ・ドメイン・コンフィギュレーションではファクトリの ID が競合する場合がありますので、ファクトリ ID と RNAME パラメータを使用して、ローカル・ドメインでリモート・ファクトリに別の ID または「エイリアス」を指定することができます。リスト 3-2 に、TP Framework によって識別子 BankTeller でドメイン "Norwest" に登録されたりモート・ファクトリの例を 2 つ示します。どちらの例でも、ローカル・ドメインではファクトリを Teller のエイリアスで使用できます。

リスト 3-2 リモート・ファクトリへのエイリアスの割り当て

```
#EXAMPLE 1:
*DM_REMOTE_FACTORIES
  Teller
    DOMAINID="Norwest"
    RNAME="BankTeller.FactoryInterface"
```

```
#EXAMPLE 2:
*DM_REMOTE_FACTORIES
  "Teller.FactoryInterface"
    DOMAINID="Norwest"
    RNAME="BankTeller.FactoryInterface"
```

同じリモート・ファクトリに複数のエイリアスを割り当てることもできます。リスト 3-3 の例では、リモート・ファクトリがローカル・ドメインに Teller と BankTeller の 2 つのエイリアスで登録されています。

リスト 3-3 リモート・ファクトリへの複数のエイリアスの割り当て

```
*DM_REMOTE_FACTORIES
  "Teller.FactoryInterface"
    DOMAINID="Norwest"
    RNAME="BankTeller.FactoryInterface"
  "BankTeller.FactoryInterface"
    DOMAINID="Norwest"
    RNAME="BankTeller.FactoryInterface"
```

使用上の注意: マルチドメイン・コンフィギュレーションでは、企業のドメイン全体でファクトリ・オブジェクトの識別子が一意でなければなりません。

マルチドメイン・コンフィギュレーションでは、2 つの異なるドメインに同じ `factory_id.factory_kind` 識別子 ("Teller.FactoryInterface" など) のファクトリ・オブジェクトが存在することはできません。

2 つのドメインで同じ識別子または名前が使用されている場合、ソフトウェアの動作は、BEA WebLogic Enterprise を使用して CORBA ドメイン環境を設定したかどうかによって異なります。

- BEA WebLogic Enterprise 5.1 より前のリリースでは、ドメインの最初のサーバがファクトリを登録しても、エラー・メッセージは生成されません。1 つのドメインに同じ名前の 2 つのファクトリが登録されると、マスタ NameManager が失敗します。
- BEA WebLogic Enterprise のリリース 5.1 以降または BEA Tuxedo のリリース 8.0 以降では、エラー・メッセージが生成されて ULOG に記録されます。

注記 BEA Tuxedo CORBA 環境のシングル・ドメイン・コンフィギュレーションでは、同じ名前の複数のファクトリ・オブジェクトを使用することができます。このようなコンフィギュレーションは、ロード・バランシングを導入する場合のみ使用できます。

ドメイン全体で識別子や名前が一意であることを確認し、このような問題を回避するには、以下に示す 2 つの方法があります。

1. 企業全体で一意的識別子を使用します。この方法では、すべての識別子をマスター・リストで管理する必要があります。
2. `factory_finder.ini` ファイルで `RNAME` パラメータを使用して、ローカル `NameManager` ではエイリアスを使用するように設定します (この場合、ローカル・クライアントもリモート・ファクトリ・オブジェクトへのアクセスにエイリアスを使用するように修正する必要があります)。リスト 3-2 は、`factory_finder.ini` ファイルで `RNAME` パラメータを使用してエイリアスを作成する例を示しています。

DM_LOCAL_FACTORIES

`DM_LOCAL_FACTORIES` セクションでは、ほかのドメインにエクスポート可能なローカル・ドメインのファクトリ・オブジェクトを指定します。このセクションは、以下の方法で使用できます。

- `factory_finder.ini` に `DM_LOCAL_FACTORIES` セクションが存在しない場合、または存在しても空の場合、ローカル・ドメインのすべてのファクトリ・オブジェクトは、リモート・ドメインで使用できます。これにより、管理者がローカル・ドメインのファクトリ・オブジェクトごとにエントリを設定しなくても、ローカル・ファクトリ・オブジェクトをリモート・ドメインで使用することができます。
- `factory_finder.ini` ファイルに `DM_LOCAL_FACTORIES` セクションが存在し、予約されたキーワードの "NONE" がある場合、そのローカル・ドメインのファクトリ・オブジェクトはリモート・ドメインで使用できません。これにより、管理者がローカル・ドメインのファクトリ・オブジェクトごとにエントリを設定しなくても、ファクトリ・オブジェクトへのアクセスを制限することができます。

ファクトリ・オブジェクト登録時に `kind` 値が "FactoryInterface" に設定された識別子または名前は、このセクションにリストされなければなりません。たとえば、TP Framework によって識別子 `Teller` で登録されたファクトリ・オブジェクトのエントリは、次のように指定されます。

```
*DM_LOCAL_FACTORIES
  "Teller.FactoryInterface"
```

NameManager で適切なファクトリ・オブジェクトを検索するには、`factory_kind` が指定されていなければなりません。`factory_kind` が指定されていないエントリには、デフォルト値の "FactoryInterface" が適用されます。これにより、CORBA NamingService を使用できます。

`factory_finder.ini` ファイルでは、"*DM_REMOTE_FACTORIES" で始まるセクションを挿入することにより、リモート・ドメインにエクスポート可能なファクトリの検索方法を指定できます。つまり、このセクションが存在することは、ローカル・ドメインがリモート・ドメインのファクトリを検索できるということを意味します。

ドメイン A の `factory_finder.ini` に挿入されるエントリの形式は次のとおりです。

```
*DM_REMOTE_FACTORIES
fA.FactoryInterface DOMAINID=B
```

これは、ドメイン A で識別子 fA のファクトリを検索するという要求が、ドメイン B の FactoryFinder によって実行できることを意味します。もちろん、2つのドメインの UBBCONFIG および DMCONFIG ファイルで、これらのドメイン間のドメイン・ゲートウェイが接続されるように設定されていなければなりません。

このエントリの別の形式を以下に示します。

```
CDE.FactoryInterface DOMAINID=B RNAME=fA.FactoryInterface
```

これは、ドメイン A で識別子 "CDE" のファクトリを検索するという要求が、ドメイン B の FactoryFinder で ID fA を使用して実行されることを意味します。エイリアスを呼び出す場合もあります。

注記 ファクトリ ID の最後には ".FactoryInterface" が必要です。このマニュアルのサンプル・コンフィギュレーションではこれが省略されていますが、実際のファイルには必要です。

`factory_finder.ini` ファイルの詳細については、『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `factory_finder.ini` ファイルの説明を参照してください。

ローカル・ファクトリ

各ドメインでは、ほかのドメインからアクセスできるファクトリを指定することができます。この場合、`*DM_LOCAL_FACTORIES` で始まるセクションで指定します。

`factory_finder.ini` ファイルが存在しない場合、存在してもこのセクションがない場合、または空の場合、すべてのローカル・ファクトリはリモート・ドメインからアクセスできます。このセクションが存在し、キーワード `None` が指定されている場合、そのローカル・ドメインのファクトリはエクスポートできません。つまり、リモート `FactoryFinder` で検索できるファクトリはありません。このセクションが存在する場合、リモート・ドメインで使用可能なファクトリのリストを指定できます。たとえば、次の例を考えます。

```
*DM_LOCAL_FACTORIES
fA.FactoryInterface
fB.FactoryInterface
```

上の例では、ファクトリ `fA` と `fB` がほかのドメインから検索できるように設定されています。このセクションにリストされていないその他のファクトリは検索できません。リモート・ファクトリとは異なり、ローカル・ファクトリでエイリアスを使用することはできません。

注記 `factory_finder.ini` ファイルと `DMCONFIG` ファイルの内容は対応していません。つまり、`factory_finder.ini` ファイルでファクトリにアクセス可能な別のドメインが宣言されている場合、`DMCONFIG` ファイルでもそのドメインにアクセスできるように設定されていなければなりません。

CORBA ドメイン・コンフィギュレーションの種類

マルチ・ドメイン機能を使用する場合、直接接続ドメインと間接接続ドメインという2種類のコンフィギュレーションが可能です。どちらのコンフィギュレーションも、管理者がドメイン・コンフィギュレーション・ファイルの `DMCONFIG` を使用して設定できます。

直接接続ドメイン

企業内の各ドメインには、ほかのドメインへのゲートウェイを個別に設定することができます。このようなコンフィギュレーションを行うと、ターゲット・ドメインに直接要求を送信できるので、遅延時間を最小限に抑えることができます。ドメインの数が少ない場合は、このように何通りもの接続を作成するコンフィギュレーションが有効ですが、新しいドメインごとに2つの新しいゲートウェイが必要になります。ドメイン数が増えた場合、管理者は別のコンフィギュレーション、つまり、要求の送信速度ではなく、ドメイン接続の管理のしやすさを優先するコンフィギュレーションを検討する必要があります。このような場合に有効なのが、ドメインを間接的に接続するコンフィギュレーションです。

間接接続ドメイン

このコンフィギュレーションでは、管理者がまずトラフィックのパターンを分析する必要があります。データをやり取りする頻度が低いドメインは、ゲートウェイ削除の対象になります。ただし、ゲートウェイを削除した後も、ほかのドメインとデータをやり取りする経路を残しておく必要があります。このような場合に使用されるのが、目的のドメインに直接アクセスできる中間ドメインを経由して要求を送信する方法です。たとえば、A、B、Cの3つのドメインがあるとします。ドメインAとB、ドメインBとCは直接接続されていますが、ドメインAとCは直接接続されていません(図3-2参照)。ドメインAとCが通信するには、中継点としてドメインBを使用しなければなりません。したがって、ドメインAのDMCONFIGファイルでは、ドメインB経由でドメインCに接続するよう指定し、ドメインCのDMCONFIGファイルでも、ドメインB経由でドメインAに接続するよう指定する必要があります。接続形態は以下のようになります。

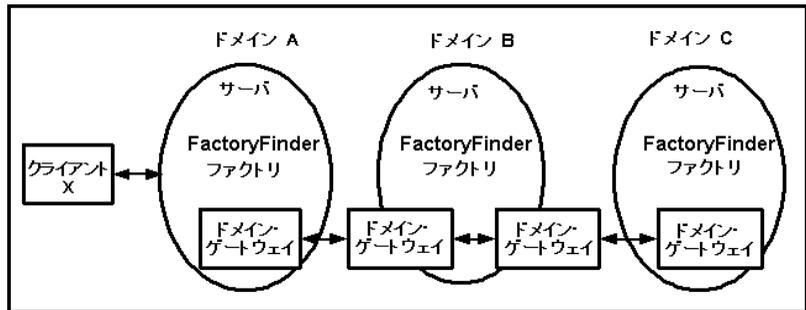
```
ドメイン      A      <->      B      <->      C
ゲートウェイ  GAB  GBA      GBC  GCB
```

ドメインAにはゲートウェイ・プロセスGAB(AからBへのゲートウェイ)があり、これによってドメインBに接続します。ドメインAのDMCONFIGファイルでは、GABがドメインBとC間のゲートウェイとして機能するように指定されています。ドメインCのDMCONFIGファイルでも同様に、GCBがBとAに接続されるように設定されています。ドメインBのDMCONFIGファイルには、Aに接続するゲートウェイ・プロセス(GBA)と、Cに接続するゲートウェイ・プロセス(GBC)の2つが設定されています。このような方法を間接接続と呼びます。

この間接続では、A のサーバが C のオブジェクトに対する要求を呼び出す場合、BEA Tuxedo CORBA サーバはその要求をゲートウェイ GAB に送信できることを認識しています。BEA Tuxedo ゲートウェイは、送信先の B のゲートウェイが、その要求を処理することはできず、中継点として受け付けることを認識していません。要求がドメイン B に送信されると、GBC を通して C にルーティングされ、ドメイン C で要求が処理されます。このように、この方法では要求の処理に 1 ホップ追加されます。

ドメイン B の 2 つのゲートウェイを 1 つにして、B でホップが追加されないようにすることもできます。この場合、ドメイン B での処理は同じですが、1 つのゲートウェイ・プロセスですべて処理されます。

図 3-2 間接続ドメイン



例：複数の CORBA ドメインの設定

次の節は、直接接続ドメインの設定例を示します。

注記 これらの例は設定方法を示すためのものであり、実際に使用する場合、システム環境に応じて、環境変数 APPDIR、TUXCONFIG、および TUXDIR を変更しなければなりません。また、角かっこ (< と >) で囲まれている部分 (<App Server Name> など) は適切な情報に読み替え、実際には角かっこを外します。

UBBCONFIG ファイルの例

リスト 3-4、リスト 3-5、および リスト 3-6 では、直接接続された 3 つのドメイン、Here、There、および Yonder の UBBCONFIG ファイルを示します。

注記 これらのファイルを使用するには、*host* をローカル・マシンの名前に置き換える必要があります。

リスト 3-4 Here ドメインの UBBCONFIG ファイル

```
#
#   Copyright (c) 1999 BEA Systems, Inc.
#   All rights reserved
#
#
# RESOURCES
#
*RESOURCES
  IPCKEY          123312
  DOMAINID      HereD
  MASTER        LAPP
  MODEL         SHM
  LDBAL         N

#
# MACHINES
#
*MACHINES
  <host>
    LMID=LAPP
    APPDIR="/tst1/wle4.2/test_dom/t07:
            /tst1/wle4.2/dec_unix/wlemdomai"
    TUXCONFIG="/tst1/wle4.2/test_dom/tuxconfig"
    TUXDIR="/lclobb/lc"
    MAXWSCLIENTS=10

#
# GROUPS
#
*GROUPS
  DEFAULT:      LMID=LAPP
  ICEGRP        GRPNO=11 OPENINFO=NONE
  GROUP1       GRPNO=21 OPENINFO=NONE
  LDMGRP        GRPNO=3
  LGWGRP        GRPNO=4
```

```

#
# SERVERS
#
*SERVERS
    DEFAULT:    CLOPT="-A"
    DMADM        SRVGRP=LDMGRP SRVID=1
    GWADM        SRVGRP=LGWGRP SRVID=1
    GWTDOMAIN   SRVGRP=LGWGRP SRVID=2
    TMSYSEVT    SRVGRP=ICEGRP SRVID=1
    TMFFNAME    SRVGRP=ICEGRP SRVID=2
                    CLOPT="-A -- -N -M -f <FF ini file for Here>"
    TMFFNAME    SRVGRP=ICEGRP SRVID=3 CLOPT="-A -- -N"
    TMFFNAME    SRVGRP=ICEGRP SRVID=4 CLOPT="-A -- -F"
    <App Server Name>    SRVGRP=GROUP1 SRVID=2
    ISL         SRVGRP=GROUP1 SRVID=1
                    CLOPT="-A -- -d /dev/tcp -n //<host>:<port>"

#
# SERVICES
#
*SERVICES

```

リスト 3-5 There ドメインの UBBCONFIG ファイル

```

#
#      Copyright (c) 1999 BEA Systems, Inc.
#      All rights reserved
#
# RESOURCES
#
*RESOURCES
    IPCKEY      133445
    DOMAINID    ThereD
    MASTER      LAPP1
    MODEL       SHM
    LDBAL       N

#
# MACHINES
#
*MACHINES
    <host>
        LMID=LAPP1
        APPDIR="D:\test_dom\t07;D:\Iceberg\qa\orb\bld\wlemdomain"
        TUXCONFIG="D:\test_dom\tuxconfig"
        TUXDIR="D:\Iceberg"
        MAXWSCLIENTS=10

#

```

```

# GROUPS
#
*GROUPS
    DEFAULT    LMID=LAPP1
    ICEGRP     GRPNO=11  OPENINFO=NONE
    GROUP1     GRPNO=21  OPENINFO=NONE
    LDMGRP     GRPNO=3
    LGWGRP     GRPNO=4

#
# SERVERS
#
*SERVERS
    DEFAULT:   CLOPT="-A"
    DMADM      SRVGRP=LDMGRP SRVID=1
    GWADM      SRVGRP=LGWGRP SRVID=1
    GWTDOMAIN  SRVGRP=LGWGRP SRVID=2
    TMSYSEV    SRVGRP=ICEGRP SRVID=1
    TMFFNAME   SRVGRP=ICEGRP SRVID=2
                CLOPT="-A -- -N -M -f <FF ini file for There>"
    TMFFNAME   SRVGRP=ICEGRP SRVID=3 CLOPT="-A -- -N"
    TMFFNAME   SRVGRP=ICEGRP SRVID=4 CLOPT="-A -- -F"
    <App Server Name>  SRVGRP=GROUP1 SRVID=2
    ISL        SRVGRP=GROUP1 SRVID=1
                CLOPT="-A -- -d /dev/tcp -n //<host>:<port>"

#
# SERVICES
#
*SERVICES

```

リスト 3-6 Yonder ドメインの UBBCONFIG ファイル

```

#      Copyright (c) 1999 BEA Systems, Inc.
#      All rights reserved
#
# RESOURCES
#
*RESOURCES
    IPCKEY     123334
    DOMAINID   YonderD
    MASTER     LAPP
    MODEL      SHM
    LDBAL      N

#
# MACHINES
#
*MACHINES

```

```

<host>
    LMID=LAPP
    APPDIR="/tst1/wle4.2/test_dom/t07p:
                /tst1/wle4.2/<host3>/wlemdomain"
    TUXCONFIG="/tst1/wle4.2/test_dom/<host3>/tuxconfig"
    TUXDIR="/lclobb/lc"
    MAXWSCLIENTS=10

#
# GROUPS
#
*GROUPS
    DEFAULT:    LMID=LAPP
    ICEGRP      GRPNO=11 OPENINFO=NONE
    GROUP1     GRPNO=21 OPENINFO=NONE
    LDMGRP      GRPNO=3
    LGWGRP      GRPNO=4

#
# SERVERS
#
*SERVERS
    DEFAULT:    CLOPT="-A"
    DMADM       SRVGRP=LDMGRP SRVID=1
    GWADM       SRVGRP=LGWGRP SRVID=1
    GWTDOMAIN   SRVGRP=LGWGRP SRVID=2
    TMSYSEVT    SRVGRP=ICEGRP SRVID=1
    TMFFNAME    SRVGRP=ICEGRP SRVID=2
                CLOPT="-A -- -N -M"
    TMFFNAME    SRVGRP=ICEGRP SRVID=3 CLOPT="-A -- -N"
    TMFFNAME    SRVGRP=ICEGRP SRVID=4 CLOPT="-A -- -F"
    <App Server Name> SRVGRP=GROUP1 SRVID=2
    ISL         SRVGRP=GROUP1 SRVID=1
                CLOPT="-A -- -d /dev/tcp -n //<host>:<port>"

#
# SERVICES
#
*SERVICES

```

DMCONFIG ファイルの例

リスト 3-7、リスト 3-8、および リスト 3-9 では、直接接続された 3 つのドメイン、Here、There、および Yonder の DMCONFIG ファイルを示します。

注記 リスト 3-7 をマルチドメイン・コンフィギュレーションで使用するには、Here ドメインの *host1*、There ドメインの *host2*、および Yonder ドメインの *host3* を、それぞれローカル・マシンの名前と置き換える必要があります。

リスト 3-7 3 ドメイン・コンフィギュレーションにおける Here ドメインのローカル・マシン用 DMCONFIG ファイル

```
#
#Copyright (c) 1999 BEA Systems, Inc.
#All rights reserved
#
#
# Tuxedo DOMAIN CONFIGURATION FILE
#
*DM_RESOURCES
    VERSION=U22

#
# DM_LOCAL_DOMAINS
#
*DM_LOCAL_DOMAINS
    LDOM1  GWGRP=LGWGRP  TYPE=TDOMAIN  DOMAINID="HereG"

#
# DM_REMOTE_DOMAINS
#
*DM_REMOTE_DOMAINS
    TDOM1    TYPE=TDOMAIN  DOMAINID="ThereG"
    TDOM2    TYPE=TDOMAIN  DOMAINID="YonderG"

#
# DM_TDOMAIN
#
*DM_TDOMAIN
    LDOM1  NWADDR="//<host1>:<tcpport>"
    TDOM1  NWADDR="//<host2>:<tcpport>"
    TDOM2  NWADDR="//<host3>:<tcpport>"

#
# DM_LOCAL_SERVICES
#
*DM_LOCAL_SERVICES
    "//HereD"

#
# DM_REMOTE_SERVICES
#
*DM_REMOTE_SERVICES
    "//ThereD      "RDOM=TDOM1
    "//YonderD     "RDOM=TDOM2
```

注記 リスト 3-8 をマルチドメイン・コンフィギュレーションで使用するには、There ドメインの *host1*、Here ドメインの *host2*、および Yonder ドメインの *host3* を、それぞれローカル・マシンの名前と置き換える必要があります。

リスト 3-8 3 ドメイン・コンフィギュレーションにおける There ドメインの DMCONFIG ファイル

```
#
#Copyright (c) 1999 BEA Systems, Inc.
#    All rights reserved
#
#
# Tuxedo DOMAIN CONFIGURATION FILE
#
*DM_RESOURCES
    VERSION=U22

#
# DM_LOCAL_DOMAINS
#
*DM_LOCAL_DOMAINS
    LDOM1  GWGRP=LGWGRP  TYPE=TDOMAIN  DOMAINID="ThereG"

#
# DM_REMOTE_DOMAINS
#
*DM_REMOTE_DOMAINS
    TDOM1   TYPE=TDOMAIN  DOMAINID="HereG"
    TDOM2   TYPE=TDOMAIN  DOMAINID="YonderG"

#
# DM_TDOMAIN
#
*DM_TDOMAIN
    LDOM1  NWADDR="//<host1>:<tcpport>"
    TDOM1  NWADDR="//<host2>:<tcpport>"
    TDOM2  NWADDR="//<host3>:<tcpport>"

#
# DM_LOCAL_SERVICES
#
*DM_LOCAL_SERVICES
    "//ThereD"

#
# DM_REMOTE_SERVICES
#
*DM_REMOTE_SERVICES
```

```
//HereD      "RDOM=TDOM1
//YonderD    "RDOM=TDOM2
```

注記 リスト 3-9 をマルチドメイン・コンフィギュレーションで使用するには、Yonder ドメインの *host1*、Here ドメインの *host2*、および There ドメインの *host3* を、それぞれローカル・マシンの名前と置き換える必要があります。

リスト 3-9 3 ドメイン・コンフィギュレーションにおける Yonder ドメインの DMCONFIG ファイル

```
#
#Copyright (c) 1999 BEA Systems, Inc.
#   All rights reserved
#
#
# Tuxedo DOMAIN CONFIGURATION FILE
#
*DM_RESOURCES
    VERSION=U22

#
# DM_LOCAL_DOMAINS
#
*DM_LOCAL_DOMAINS
    LDOM1  GWGRP=LWGRP  TYPE=TDOMAIN  DOMAINID="YonderG"

#
# DM_REMOTE_DOMAINS
#
*DM_REMOTE_DOMAINS
    TDOM1   TYPE=TDOMAIN  DOMAINID="HereG"
    TDOM2   TYPE=TDOMAIN  DOMAINID="ThereG"

#
# DM_TDOMAIN
#
*DM_TDOMAIN
    LDOM1  NWADDR="//<host1>:<tcpport>"
    TDOM1  NWADDR="//<host2>:<tcpport>"
    TDOM2  NWADDR="//<host3>:<tcpport>"

#
# DM_LOCAL_SERVICES
#
```

```
*DM_LOCAL_SERVICES
  "//YonderG"
#
# DM_REMOTE_SERVICES
#
*DM_REMOTE_SERVICES
  "//HereD      "RDOM=TDOM1
  "//ThereD     "RDOM=TDOM2
```

factory_finder.ini ファイルの例

この節では、Here ドメインと There ドメインの factory_finder.ini ファイルの例を示します。Yonder ドメインには factory_finder.ini ファイルは必要ありません。

リスト 3-10 Here ローカル・ドメインの factory_finder.ini ファイル

```
#Copyright (c) 1999 BEA Systems, Inc.
#All rights reserved
#
# ドメイン "Here" の FactoryFinder 初期化ファイル
# "Here" はローカルドメインです
#
# DM_LOCAL_FACTORIES
#
*DM_LOCAL_FACTORIES
  "AFactory.FactoryInterface"
#
# DM_REMOTE_FACTORIES
#
*DM_REMOTE_FACTORIES
  "AFacYonder.FactoryInterface"
  DOMAINID="YonderD"
  RNAME="AFactory.FactoryInterface"

  "BFactory.FactoryInterface"
  DOMAINID="YonderD"
```

リスト 3-11 There リモート・ドメインの factory_finder.ini ファイル

```
#
#Copyright (c) 1999 BEA Systems, Inc.
#All rights reserved
#
# ドメイン "There" の FactoryFinder 初期化ファイル
# "There" はリモート・ドメインです
#
# DM_LOCAL_FACTORIES
#
*DM_LOCAL_FACTORIES
    "AFactory.FactoryInterface"
#
# DM_REMOTE_FACTORIES
#
*DM_REMOTE_FACTORIES
    "AFacYonder.FactoryInterface"
DOMAINID="YonderD"
RNAME="AFactory.FactoryInterface"
    "BFactory.FactoryInterface"
DOMAINID="YonderD"
```

4 Domains の管理

ここでは、次の内容について説明します。

- Domains の実行時管理コマンドを使用する
- 管理インターフェイス `dmadmin(1)` を使用する
- Domains 管理サーバ `DMADM(5)` を使用する
- ゲートウェイ管理サーバ `GWADM(5)` を使用する
- ゲートウェイ・プロセスを使用する
- Domains 環境でのトランザクションの管理

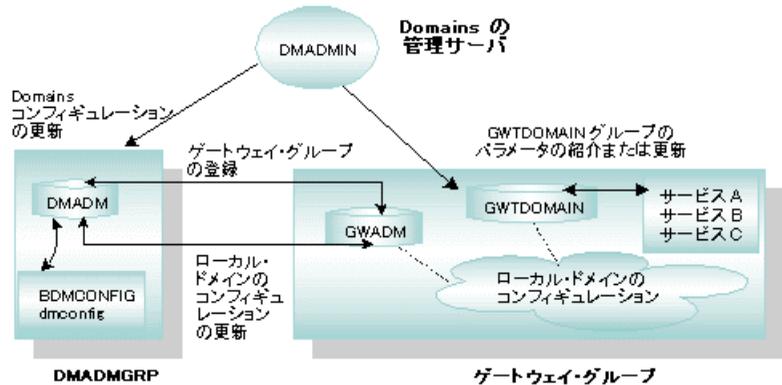
Domains の実行時管理コマンドを使用する

Domains コンポーネントを既存の BEA Tuxedo アプリケーションと統合するには、ドメイン・ゲートウェイ・グループとゲートウェイ・サーバのエントリを `TUXCONFIG` ファイルに追加します。実行中の BEA Tuxedo アプリケーションにマルチ・ドメイン・コンフィギュレーションを追加するには、`tmconfig(1)` (`tmconfig`、`wtmconfig(1)` を参照) または `tmadmin(1)` コマンドを使用します。`tmadmin` を使用すると、Domain ゲートウェイ・グループや個々のゲートウェイの掲示板で使用できる情報をリストすることもできます。

Domains 環境を設定し、統合すると、Domains ソフトウェアに用意されている一連の管理ツールを使用して、Domains 環境を動的に管理できます。たとえば、アプリケーションのどこからでもアクセスできるサービスをリスト化したり、そのリストを変更することができます。Domains ソフトウェアには、BEA Tuxedo のプログラミング・インターフェイス (ATMI) の機能に加え、拡張された機能が備わっているため、クライアントはドメイン内のあらゆるサービスを呼び出すことができます。この機能により、プログラマは、アプリケーション・コードを変更せずに、アプリケーションを拡張したり、分割できます。

次の図は、Domains の管理サブシステムにおける管理コマンドと管理サーバの関係を示します。

図 4-1 Domains の実行時の管理



Domains には、次の管理コマンドが用意されています。

- `dmadmin(1)` 一般的な管理サービス用のコマンドです。管理者は、BEA Tuxedo アプリケーションの実行中に、ドメイン・ゲートウェイ・グループを動的に設定、監視、および調整したり、Domains のコンフィギュレーション・ファイル (BDMCONFIG) を更新できます。このコマンドは、管理コマンドをサービス要求に変換し、そのサービス要求を DMADMIN サービス (DMADM サーバによって宣言された一般的な管理サービス) に送信する、フロント・エンド・プロセスとして機能します。DMADMIN サービスは、DMADM サーバに組み込まれている、検証、検索、または更新を行う機能呼び出して、BDMCONFIG ファイルを管理します。
- `DMADM(5)` ゲートウェイ・グループの管理サーバ用のコマンドです。Domains のコンフィギュレーションを更新するための管理プロセスを提供します。このサーバは、`dmadmin` コマンドに対するバックエンドとして機能し、ゲートウェイ・グループに対して登録サービスを提供します。この登録サービスは、GWADM サーバの初期化処理で要求されます。登録サービスにより、ゲートウェイ・グループが要求するコンフィギュレーション情報がダウンロードされます。DMADM サーバは、登録済みのゲートウェイ・グループのリストを管理し、コンフィギュレーション・ファイルが変更されると、変更内容をリスト内のゲートウェイ・グループに伝播します。

- GWADM(5) ゲートウェイ・プロセス用のコマンドです。GWADM サーバは、DMADM サーバに登録し、対応するゲートウェイ・グループが使用するコンフィギュレーション情報を取得します。GWADM は、DMADM からの問い合わせ、つまり、対応するゲートウェイ・グループの実行時オプションでの統計情報や変更に対する問い合わせを受け付けます。GWADM サーバは、「I-am-alive」メッセージを定期的に DMADM サーバに送信します。DMADM サーバから応答が返されない場合、GWADM サーバは再度登録を行います。このメカニズムにより、GWADM サーバは、そのグループの Domains コンフィギュレーションに関する最新の情報を常に保持できます。
- GWTDOMAIN(5) ゲートウェイ・プロセス GWTDOMAIN 用のコマンドです。リモート・ゲートウェイ・プロセスへの接続を実現し、BEA Tuxedo のドメイン間でメッセージの送受信を処理します。クライアントおよびサーバは、GWTDOMAIN プロセスを介して BEA Tuxedo のドメイン間でメッセージを送受信します。

注記 GWTDOMAIN 以外のゲートウェイでは、GWTDOMAIN 以外の実行可能ファイルを使用する必要があります。詳細については、BEA eLink Adapter for Mainframe のマニュアル、および『ATMI アプリケーションでの BEA Tuxedo TOP END Domain Gateway の使用』を参照してください。

- BDMCONFIG バイナリ形式の Domains のコンフィギュレーション・ファイルです。このファイルに含まれているコンフィギュレーション・パラメータが BEA Tuxedo ソフトウェアによって解釈されると、実行可能なアプリケーションが生成されます。

注記 TUXCONFIG ファイルの SERVERS セクションで GWADM サーバが定義されている場合に、CLOPT パラメータを使用してゲートウェイ・グループを起動すると、ゲートウェイ・パラメータを指定することもできます

DMADM と Domain ゲートウェイ・グループの移行方法

DMADM は移行することができます。DMADM を新しいマシンに移行するには、次の手順に従います。

1. DMCONFIG を新しいマシンにコピーし、`dmloadcf` を実行します。
2. すべてのドメイン・ゲートウェイ・グループ、たとえば、GWADM とドメイン・ゲートウェイ (GWTDOMAIN) をシャットダウンします。

注記 ドメイン・ゲートウェイ・グループは、シャットダウンされない限り動作し続けます。ただし、DMADM が移行されると、ドメイン・ゲートウェイ・グループに対するすべての MIB 要求は失敗します。

3. DMADM グループを新しいマシンに移行します。

ドメイン・ゲートウェイ・グループは移行することができます。ただし、トランザクションを使用している場合、ドメイン・ゲートウェイ・グループは、同じ種類のマシン間でのみ移行できます。ドメイン・ゲートウェイ・グループを移行するには、次の手順に従います。

1. DMCONFIG ファイルの DM_TDOMAIN セクションに、次の形式で複数の受信アドレスを追加します。

```
*DM_TDOMAIN
LDM NWADDR="//primary:port"
LDM NWADDR="//backup:port"
```

注記 サードパーティによる IP のフェイルオーバー対応機能が有効な場合、この手順は不要です。

2. トランザクションを使用している場合は、Domains のトランザクション・ログを手動でバックアップ・マシンにコピーする必要があります。
3. 手順 1 で指定した 2 つのネットワーク・アドレスが、リモート・ドメインの DMCONFIG ファイルに含まれていることを確認します。
4. ドメイン・ゲートウェイ・グループを新しいマシンに移行します。

管理インターフェイス dmadmin(1) を使用する

dmadmin は、DMADM サーバ用の GWADM サーバへの管理インターフェイスです。これらの 2 つのサーバ間の通信は、FML の型付きバッファを使用して実行されます。管理者は、dmadmin コマンドを使用して、次の作業を実行できます。

- BDMCONFIG ファイルに格納された情報と特定の BEA Tuxedo アプリケーションで実行している異なるゲートウェイ・グループの情報を対話形式で管理できます。
- 統計情報やゲートウェイ・グループが収集したほかの情報を取得できます。
- ゲートウェイ・グループのパラメータを変更できます。
- BDMCONFIG ファイルに情報を追加したり、情報を更新できます。

注記 実行時に BDMCONFIG ファイルから削除できるのは、アクティブなゲートウェイ・グループとは関係のない情報だけです。

関連項目

- 『BEA Tuxedo コマンド・リファレンス』の `dmadmin(1)`

Domains 管理サーバ DMADM(5) を使用する

Domains の管理サーバである DMADM(5) は、BEA Tuxedo に組み込まれているサーバであり、以下の機能を実行します。

- `BDMCONFIG` ファイルを実行時に管理します。
- `BDMCONFIG` ファイルを管理します。
- 登録済みのゲートウェイ・グループをリストします。
- 実行時のコンフィギュレーションの変更を、登録済みのゲートウェイ・グループに伝播します。

DMADM サーバは、次の 2 つのサービスを宣言します。

- `DMADMIN`。DMADMIN サーバと `GWADM` サーバによって使用されます。
- `DMADM_svrid`。SRVID は、サービス用のサーバ ID です。登録済みの `GWADM` サーバは、特定の管理機能 (ゲートウェイ・グループのコンフィギュレーション情報をリフレッシュしたり、`GWADM` が登録されたままであることを通知する機能など) に対して `DMADM_svrid` を使用します。

DMADM サーバは、グループ内で実行しているサーバ (`DMADMGRP` など) として `TUXCONFIG` ファイルの `SERVERS` セクションで定義されている必要があります。このグループでは、DMADM サーバのインスタンスは 1 つしか存在できません。また、応答キューは指定できず、`REPLYQ=N` と指定する必要があります。

関連項目

- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の `DMADM(5)`

ゲートウェイ管理サーバ GWADM(5) を使用する

ゲートウェイ管理サーバ GWADM(5) は、BEA Tuxedo に組み込まれているサーバであり、Domains のゲートウェイ・グループ用の管理機能を提供します。GWADM サーバの主な機能は、以下のとおりです。

- DMADM サーバから Domains のコンフィギュレーション情報を取得したり、`dmadmin` からの問い合わせを受け付けます。GWADM サーバは、DMADM サーバに登録することにより、ゲートウェイ・グループのコンフィギュレーション情報を取得します。次に、GWADM サーバは、この情報を共用メモリに格納し、ゲートウェイでこのコンフィギュレーションを利用できるようにします。
- ゲートウェイ・グループの管理機能を提供します。たとえば、実行時の統計情報について `dmadmin` からの問い合わせを受け付けたり、ゲートウェイ・グループの実行時パラメータを変更します。
- ゲートウェイ・グループ用のトランザクション・ログ機能を提供します。GWADM サーバは、共用メモリに格納されている情報を読み込み、どのトランザクション・ログを記録する必要があるかを判断します。GWADM サーバは、起動時にログをスキャンして、トランザクションの回復が必要かどうかを調べます。回復が必要であれば、共用メモリ内のトランザクション情報を回復します。ゲートウェイ・サーバは、共用メモリの情報をスキャンして、対応するトランザクションを回復します。回復手順は、ゲートウェイ・グループで受信される新規の入力要求や出力要求とは、非同期的に行われます。

GWADM サーバは、ローカル・ドメイン名 (BDMCONFIG での `LDOM` キーワードの値) に基づいて、サービス名を宣言します。`dmadmin` コマンドは、このサービスを使用して、アクティブなすべてのゲートウェイ・グループまたは特定のゲートウェイ・グループから、情報を取り出します。

GWADM サーバは、`TUXCONFIG` ファイルの `SERVERS` セクションで定義しておく必要があります。グループに関連するゲートウェイが使用する `MSSQ` セットの一部にこのサーバを指定することはできません。また、応答キューを指定することはできません。`REPLYQ=N` を指定してください。また、ゲートウェイ・グループ内で最初に起動されるサーバでなければなりません。つまり、`SEQUENCE` で番号が指定されているか、またはゲートウェイ・サーバより先に定義されていなければなりません。

GWADM サーバには DMADM サーバが必要です。具体的には DMADM サーバを起動してから、GWADM を起動する必要があります。

GWADM サーバは、ゲートウェイ・グループに必要な共用メモリを作成し、DMADM サーバから受け取る情報をコンフィギュレーション・テーブル内に設定しなければなりません。GWADM サーバは `shmget` で `IPC_PRIVATE` を使用し、掲示板のレジストリ・エントリにある `shmid` フィールドに返された `ipckey` を格納します。ゲートウェイは、GWADM レジストリ・エントリを取得し、`shmid` フィールドをチェックすることにより、`ipckey` を取得できます。

関連項目

- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の GWADM(5)

ゲートウェイ・プロセスを使用する

ゲートウェイ・プロセスは、リモート・ゲートウェイ・プロセスへの接続を実現し、1 つ以上のリモート・ゲートウェイと同時に通信することができます。ゲートウェイは、BEA Tuxedo アプリケーションにインポートされたサービスを宣言し、アプリケーションによってエクスポートされたローカル・サービスへのアクセスを制御します。アプリケーションのエクスポートされたサービスとインポートされたサービスは、Domains コンフィギュレーション・ファイル (`DMCONFIG`) で定義します。ドメイン・ゲートウェイ・グループを動的に構成、監視、および調整するには、`dmadmin` を使用します。

関連項目

- 第 1 章の 5 ページ「ドメイン・ゲートウェイの種類」

Domains 環境でのトランザクションの管理

アプリケーション・プログラマは、トランザクションでリモート・サービスの実行を要求することができます。また、リモート・ドメインのユーザが、トランザクションでローカル・サービスの実行を要求することもできます。Domains は、リモート・トランザクションをローカル・トランザクションにマッピングしたり、これらのトランザクションが正しく終了（コミットまたはロールバック）されるように調整する役割を果たします。

BEA Tuxedo のシステム・アーキテクチャでは、トランザクション・マネージャ・サーバ (TMS: Transaction Manager Server) という別個のプロセスにより、特定のグループにアクセスするトランザクション・ブランチのコミットや回復が調整されます。ただし、Domains 環境で、受信したトランザクションのコミット操作を処理するには、ゲートウェイから TMS サーバに別途メッセージを送信しなければなりません。

Domains のアーキテクチャを単純化し、送信メッセージの数を抑えるため、TMS コードは、ゲートウェイ・コードと統合されています。これで、ドメイン・ゲートウェイは、BEA Tuxedo システムで使用されるトランザクション・プロトコルを処理できます。BEA Tuxedo のトランザクション・プロトコルを使用するには、ゲートウェイで TMS サービスが宣言されていなければなりません。この宣言は、最初のゲートウェイの起動時に実行されます。いったん TMS サービスが宣言されると、ゲートウェイ・グループ宛てのトランザクション・コントロール・メッセージは、すべてゲートウェイのキューに登録されます。

Domains のゲートウェイ・グループは、TUXCONFIG ファイルで定義されており、TMSNAME、TMSCOUNT、OPENINFO、CLOSEINFO の 4 つのパラメータは指定されません。これらのパラメータは、XA 対応のリソース・マネージャを使用するゲートウェイ・グループにのみ適用されるためです。Domains ゲートウェイは、このリソース・マネージャを使用しません。

ドメインのコミット・プロトコルは厳密な階層構造になっています。トランザクション・ツリーを水平に構成することはできません。上位ドメインが認識するのは、すぐ下の下位ドメインだけであり、各ドメインがトランザクション・ツリー全体を把握しているわけではないためです。ツリーを水平に構成すると、トランザクションに参加しているすべてのドメインにルート・ドメインを完全に接続することも必要になります。

トランザクション管理機能

ドメイン・ゲートウェイには、トランザクションを管理するための 4 つの機能が用意されています。これらの機能については、以下の節を参照してください。

- 第 4 章の 9 ページ「Domains で TMS 機能を使用する」
- 第 4 章の 12 ページ「トランザクションで GTRID マッピングを使用する」
- 第 4 章の 20 ページ「ログ機能によるトランザクションのトラッキング」
- 第 4 章の 22 ページ「失敗したトランザクションの回復」

Domains で TMS 機能を使用する

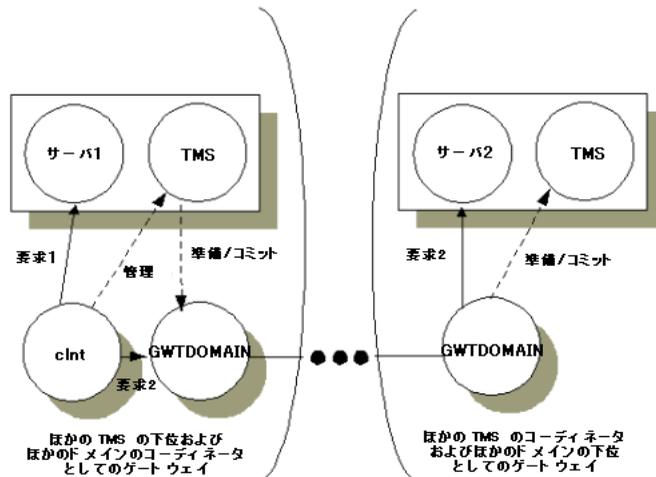
BEA Tuxedo システムの TMS は、X/Open XA 準拠のリソース・マネージャを使用するサーバ・グループと暗黙的に関連付けられた特別なサーバです。TMS サーバは、分散型の 2 フェーズ・コミット・プロトコルに伴うアプリケーション・サーバでの遅延を解消します。つまり、TMS は、TMS サービスに対して特殊なサービス要求を発行することにより、トランザクションのコミットを調整します。このサービスは、すべての TMS サーバで提供されます。

一方、Domains 環境では、GWTDOMAIN ゲートウェイは XA 準拠のリソース・マネージャと関連付けられていません。X/Open のトランザクション処理委員会 (TPWG: Transaction Processing Working Group) は、高度な XA インターフェイスを提案しましたが、このインターフェイスは、非同期性が高く、非ブロッキング・モデル型のゲートウェイには不適切なため、BEA Tuxedo システムでは使用されていません。Domains ゲートウェイは、専用の TMS サーバは使用しませんが、トランザクション・マネージャ・サーバ (TMS: Transaction Manager Servers) と同等の機能、つまり、ドメイン間で実行されるトランザクションの 2 フェーズ・コミットを調整します。

ゲートウェイによる Domains 間でのトランザクションの調整方法

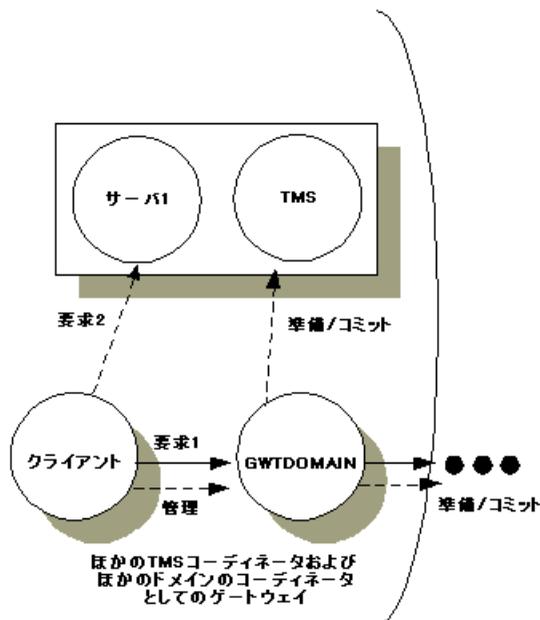
1. ドメイン・ゲートウェイは、TMS サービスを宣言し、そのサービスに関連するすべての操作を行います。このサービスに送信されたメッセージは、適切なゲートウェイ・グループによって使用されるキューに登録され、ゲートウェイはグループに対応付けられたトランザクションを管理します。
2. ゲートウェイは、ドメイン内の別のグループによって調整されるトランザクションの下位として位置付けることもできます。この場合、ゲートウェイは、ほかのリモート・ドメインで実行されるトランザクションよりも上位になります。リモート・ドメインによって調整されるトランザクションの下位として働く場合、ゲートウェイは、トランザクションがアクセスするローカル・ドメインのすべてのグループのためのコーディネータとして働きます。次の図では、下位とコーディネータの両方として機能しているゲートウェイを示します。

図 4-2 別の Domain グループの下位ゲートウェイまたはコーディネータとしてのゲートウェイ



3. このゲートウェイは、ドメイン内のトランザクションのコーディネータとして、特定のクライアントに対するトランザクションのコミットを管理します。その様子を次の図に示します。

図 4-3 ゲートウェイによって管理されるクライアントのコミット



4. ゲートウェイは、AUTOTRAN 機能を使用して、転送サービスを使用する特定のクライアントまたはサーバのためにトランザクションのコミットを管理します。この組み合わせが使用された場合には、転送チェーンの最後のサーバ (Domains ゲートウェイ) がコミットを発行し、トランザクションのコーディネータになります (ドメイン・ゲートウェイは、常に転送チェーンの最後のサーバとして機能します)。
5. ゲートウェイは、AUTOTRAN 機能で指定されたリモート・サービスに対してトランザクションを自動的に開始および終了します。この機能は、アプリケーション管理者がリモート・サービスとのネットワーク・コミュニケーションの信頼性を強化したいときに必要です。管理者がこの機能を指定するには、対応するリモート・サービス定義中のパラメータ AUTOTRAN に Y を設定します。

詳細については、『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の DMCONFIG(5) の DM_REMOTE_SERVICES セクションを参照してください。

6. ゲートウェイは、BEA Tuxedo システムのトランザクション・プロトコルを、リモート・ドメインとの相互運用に使用されるネットワーク・トランザクション・プロトコルにマッピングします。実際のマッピング方法は、使用する Domains のインスタンスによって決まります。つまり、TDomains、SNA、または OSI TP です。

トランザクションで GTRID マッピングを使用する

BEA Tuxedo システムのトランザクション・ツリーは、2 レベルで構成されています。ルートには、グローバル・トランザクションを調整するゲートウェイ・グループがあり、トランザクションはブランチに含まれます。各ゲートウェイ・グループは、ほかのグループからは独立して、グローバル・トランザクションの一部を実行します。したがって、各グループは、暗黙的にトランザクション・ブランチを定義します。BEA Tuxedo システムは、トランザクション・マネージャ・サーバ (TMS: Transaction Manager Servers) を使用して、各ブランチの完了を確認しながら、グローバル・トランザクションの完了を調整します。

GTRID は、グローバル・トランザクションの識別子です。GTRID マッピングでは、ドメインの境界をまたぐトランザクション・ツリーの構築方法を定義します。GTRID を指定するには、コンフィギュレーション・ファイルの RESOURCES セクションの MAXGTT パラメータを使用します。

密結合関係と疎結合関係の定義

X/Open DTP モデルのトランザクション・マネージャ・サーバ (TMS) は、リソース・マネージャ (RM) との関係を表すトランザクション・ツリーを構築できます。関係は「密結合」または「疎結合」で定義し、XA インターフェイスで使用されるトランザクション識別子 (XID) が使用されます。

「疎結合関係」では、1つのグローバル・トランザクションに参加するすべてのプロセスで同じトランザクション識別子 (XID) が使用され、同じ RM へのアクセスが行われます。この関係が確立されていると、プロセス間のデータ共有性を最大化できます。つまり、XA 準拠の RM では、同じ XID のプロセスによって使用されるリソースがロックを共有すると見なされます。BEA Tuxedo システムでは、「グループ」の概念に基づいて疎結合関係を実現します。つまり、指定されたグローバル・トランザクションの代わりに、グループがすべての作業を行い、それらの作業は同じトランザクション・プランチに設定されます。グループが行ったすべてのプロセスには、同じ XID が指定されます。

「疎結合関係」の TMS は、グローバル・トランザクションに参加する各作業に対して、トランザクション・プランチを生成します。RM は、各トランザクション・プランチを別々に処理します。トランザクション・プランチ間では、データやロックは共有されません。トランザクション・プランチ間でデッドロックが発生すると、グローバル・トランザクションがロールバックされます。BEA Tuxedo アプリケーションでは、1つのグローバル・トランザクションにさまざまなグループが参加している場合、グループごとに別のトランザクション・プランチが定義され、疎結合関係が確立されます。

Domains をまたがるグローバル・トランザクション

単独の BEA Tuxedo アプリケーション内のグローバル・トランザクションと、ドメインをまたがるグローバル・トランザクションとは、いくつかの違いがあります。最大の相違点は、Domains のフレームワークでは、トランザクション・ツリーの構成を 2 レベルに下げることができないことです。これには、次の 2 つの理由があります。

- トランザクションに、ルート・ドメイン (トランザクションを制御するドメイン) で認識できる数以上のドメインが含まれる場合があるため、トランザクション・ツリーの構造が完全に認識されません。
- トランザクション・ツリーが 2 レベルで構成されている場合、ルート・ドメインは、トランザクションのすべてのドメインに直接接続されていなければなりません。

つまり、ドメインをまたがるコミット・プロトコルは、階層型でなければなりません。ループ・バックされたサービス要求も、トランザクション・ツリーでは新しいプランチとして定義されます。

注記 ループ・バック要求とは、別のドメインに送信された後で、送信元のドメインに返される要求です。たとえば、ドメイン A がドメイン B のサービスを要求したとします。ドメイン B のサービスは、ドメイン A の別のサービスを要求します。トランザクション・ツリーには、ネットワーク・レベルで 2 つのブランチがあります。つまり、ドメイン A からドメイン B への要求を示すブランチ b1 と、ドメイン B からドメイン A への要求を示すブランチ b2 です。ドメイン A は、ドメイン B からコミットの指示を受けるまで、ブランチ b2 の作業をコミットすることはできません。

ドメインをまたがるグローバル・トランザクションのトランザクション・ツリー構造は、対応する Domains のインスタンスが使用する分散トランザクション処理プロトコルにも依存します。たとえば、OSI TP プロトコルのダイアログ (OSI TP でのサービス要求) は、それぞれ別のトランザクション・ブランチに関連付けられています。BEA Tuxedo システムでは、OSI TP インスタンスがサービス要求でダイアログを使用するため、各サービス要求は別のトランザクション・ブランチにマッピングされます。XAP-TP インターフェイスは、このマッピングを隠し、ユーザ定義の識別子 (BEA Tuxedo のインプリメンテーションでは GTRID) を使用して OSI TP サブツリー全体を参照するようなメカニズムを提供します。GTRID は、トランザクション・ツリーの構築方法を XAP-TP に指示します。つまり、指定された OSI TP トランザクションに含めるダイアログを指定します。したがって、BEA Tuxedo からは、OSI TP サブツリー全体が 1 つのトランザクション・ブランチとして管理されているように見えます。

しかし、この特徴は、ルート・ドメインから下位ドメインに送信されるサービス要求にのみ適用されます。逆方向で送信されるサービス要求には適用されません。OSI TP のインスタンスは、続いて疎結合関係をインプリメントします。受信したサービス要求は、新しい BEA Tuxedo グローバル・トランザクションにマッピングされます。

TDomain のインスタンスは、密結合関係を実現することにより、GTRID のマッピングを最適化しようとします。TDomain では、同じグローバル・トランザクションから発行された複数のサービス要求は、同じネットワーク・トランザクション・ブランチにマッピングされます。したがって、受信したサービス要求は、1 つの BEA Tuxedo トランザクションにマッピングされます。ただし、ドメイン間通信の階層構造とドメイン間の関係を示すトランザクション・ツリーは保持される必要があります。

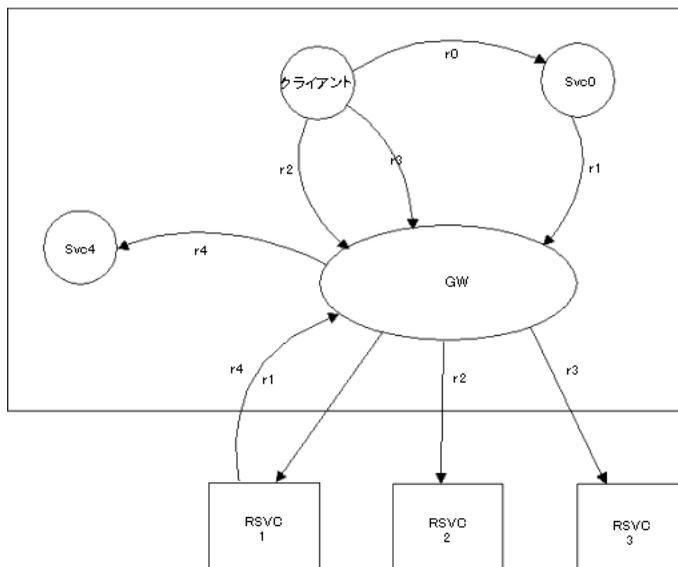
TDomain による最適化処理は、単一のドメインに対してのみ適用されます。トランザクションに複数のドメインが関係する場合、ネットワーク・トランザクション・ツリーには、ドメイン間の通信ごとに少なくとも 1 つのブランチが必要です。したがって、ドメインにまたがるネットワーク・トランザクション・ツリーは、疎結合のままになります。トランザクション・ブランチの数は、トランザクション内のドメイン数になります。これは、すべてのブランチが同じリソース・マネージャのインスタンスにアクセスしている場合も同じです。

ドメイン・ゲートウェイ・グループは、ドメイン間のトランザクションに対して異なるトランザクション・ブランチを生成するため、疎結合関係をインプリメントします。

ローカル要求とリモート要求を生成するサービス要求グラフの例

次の図は、ローカル・サービスに対する要求 (r0) およびリモート・サービスに対する 2 つの要求 (r2 および r3) を発行するクライアントを示す、サービス要求のグラフです。r0 は、ローカル・サービス (Svc0) に送信され、別のリモート・サービス要求 (r1) を生成します。r1 はリモート・サービス Rsvc1 に送信され、Rsvc1 は、ループ・バック・サービス要求 r4 をローカル・サービス Svc4 に送信します。Svc0 と Svc4 は、別々のグループ (G0 と G4) で実行されます。ドメイン・ゲートウェイは、ほかのグループ (GW) 内で実行され、リモート・サービス Rsvc1、Rsvc2、および Rsvc3 は別のドメイン (ドメイン B) で実行されます。

図 4-4 サービス要求グラフ



BEA eLink OSI TP と BEA Tuxedo Domains のトランザクション・ツリー

次の 2 つの図は、BEA eLink OSI TP のトランザクション・ツリーと、BEA Tuxedo Domains のトランザクション・ツリーを示します。これらの図では、ドメイン A とドメイン B が BEA Tuxedo システムのアプリケーションであることを想定しています。

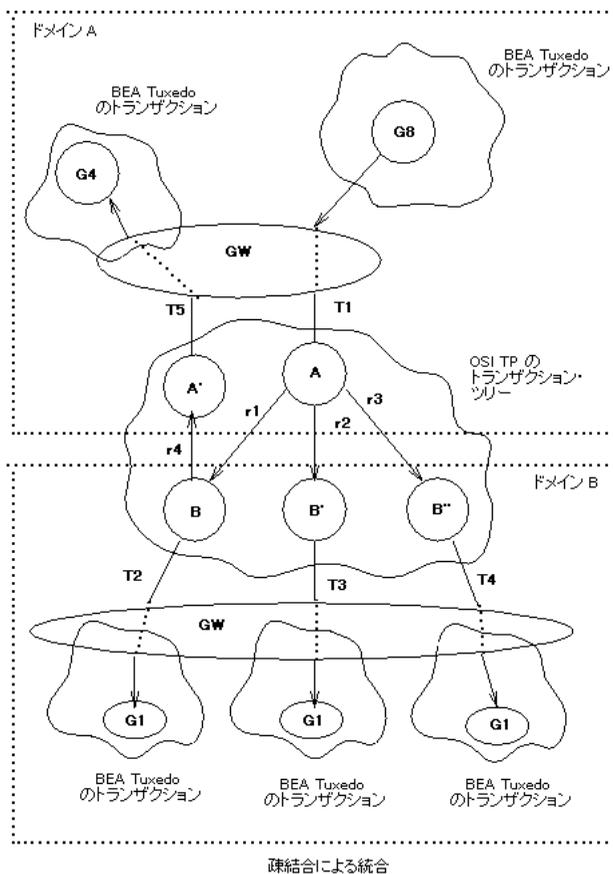
BEA eLink OSI TP は、OSI TP プロトコルを使用しているため、疎結合です。このインスタンスのトランザクション・ツリーでは、ドメイン A (クライアントが開始したグローバル・トランザクションを調整) 内にグループ G0 が示されています。グループ G0 は、グループ GW を調整します。r1、r2、および r3 の 3 つの要求は、それぞれ別の OSI TP ダイアログにマッピングされ、したがって別の OSI TP のトランザクション・ブランチにマッピングされます。ただし、OSI TP は、XAP-TP の機能を使用して、一意な識別子 (T1) で OSI TP トランザクション全体を参照し、r1、r2、および r3 の 3 つの要求にも使用します。OSI TP トランザクション識別子を作成し、対応する OSI TP トランザクション・ツリーを構築するかどうかは、XAP-TP に依存します。一般的な Domains ソフトウェアでは、r1、r2、および r3 の 3 つの要求の T1 識別子へのマッピングが、必ず実行される唯一の処理です。

Domain B では、新しい BEA Tuxedo トランザクションには新しいトランザクション・ブランチをマッピングする、という OSI TP の規則が使用されます。したがって、OSI TP トランザクション・ブランチである r1、r2、および r3 は、3 つの異なる BEA Tuxedo トランザクション (T2、T3、および T4) にマッピングされます。グラフでは、Domain B のゲートウェイ・グループ GW が、グループ G1 での 3 つの BEA Tuxedo トランザクションを調整します。

ループ・バックのサービス要求 r4 は、トランザクション・ツリーに別のブランチを生成します。OSI TP は、この要求を識別子 T2 にマッピングしますが、XAP-TP は、トランザクション・ツリーに新しいブランチを生成します。r4 の場合は、B から A' のブランチです。生成されたブランチは、ドメイン A の新しいトランザクション・ブランチであるため、ゲートウェイは新しい BEA Tuxedo トランザクションに対する新しいマッピング T5 を生成します。トランザクション・グラフでは、ドメイン A のゲートウェイ・グループ GW が、グループ G4 を調整する様子を示します。

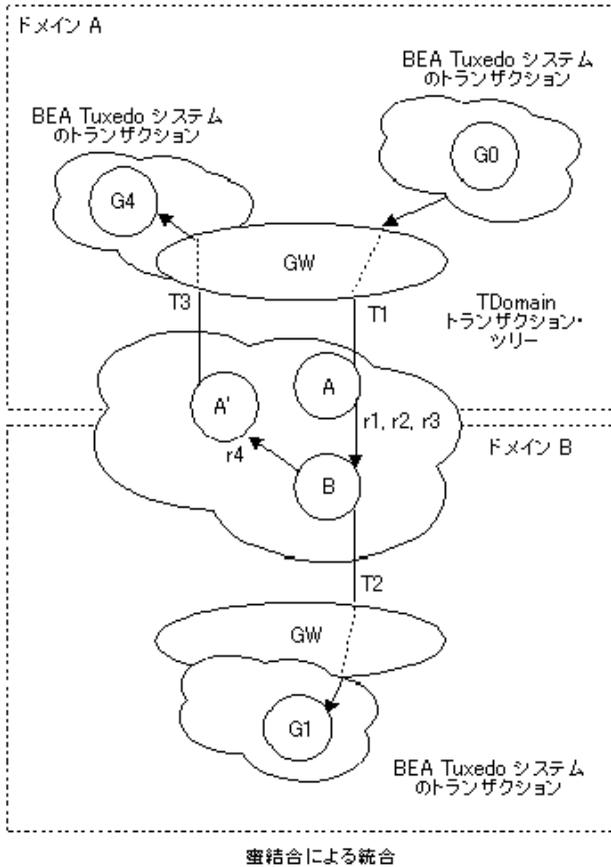
これらのマッピング関係により、OSI TP プロトコルの階層構造は強化されます。ただし、これらのマッピング関係は疎結合であるため、トランザクション内でデッドロックが発生する可能性が高まります。たとえば、グループ G1 が示す RM には、3 つの BEA Tuxedo トランザクションがアクセスします。

図 4-5 BEA eLink OSI TP 環境のトランザクション・ツリー



TDomain のインスタンスは、密結合関係を使用してドメインを統合する、つまり、2つのドメイン間の処理に必要なトランザクション・ブランチの数を減らすことにより、このデッドロックを解決します。次の図は、この関係を表すトランザクション・ツリーを示しています。

図 4-6 TDomain 環境のトランザクション・ツリー



ここでも、ゲートウェイでは、BEA Tuxedo システムのトランザクションとネットワーク・トランザクションのマッピングを行わなければなりません。また、ドメイン間の階層構造も維持されていなければなりません。この図では、 $r1$ 、 $r2$ 、および $r3$ の 3 つの要求が、1 つの TDomain トランザクション・ブランチにマッピングされています。したがって、ドメイン B で生成される必要があるのは、1 つの BEA Tuxedo システムのトランザクションだけです。T2 は、このマッピングを表します。グラフでは、ドメイン B のゲートウェイ・グループ G_W が、グループ G_1 を調整する様子を示します。要求 $r4$ は、Domain B の識別子 T2 にマッピングされますが、TDomain はトランザクション・ツリーに新しいブランチを生成します。 $r4$ の場合は、B から A' のブランチです。生成されたブランチは、ドメイン A の新しいトランザクション・ブランチであるため、ゲートウェイは新しい BEA Tuxedo トランザクションに対する新しいマッピング T3 を生成します。グラフでは、ドメイン A のゲートウェイ・グループ G_W が、グループ G_4 も調整する様子を示します。このマッピング関係により、ドメイン間通信の階層構造は強化されます。グループ G_4 は、グループ G_1 より先にコミットすることはできません。

Domains でのトランザクション管理のまとめ

Domains でのトランザクション管理は、次のようにまとめることができます。

- ゲートウェイは、BEA Tuxedo システムのトランザクションからネットワーク・トランザクションへのマッピングを実現します。新しいマッピングは、BEA Tuxedo システムのトランザクションごと、および着信したネットワーク・トランザクション・ブランチごとに作成されます。
- Domains の各インスタンス (TDomains、SNA、または OSI TP) は、個々のネットワーク・トランザクション・ツリーの表現を扱います。すべてのインスタンスでは、ドメイン間の通信の階層性が認められます。

ログ機能によるトランザクションのトラッキング

ログ機能は、2 フェーズ・コミット・プロトコルの進行をトラッキングするために使用します。ログの情報から、ネットワーク障害やマシンのクラッシュが発生したときに、トランザクションが完了したかどうかを確認できます。

ドメインをまたがるトランザクションが確実に処理されるようにするため、ドメイン・ゲートウェイでは、ローカル識別子とリモート識別子のマッピングが記録されます。このマッピング情報に加え、Domains のトランザクション管理機能により、異なるコミット・プロトコル・フェーズで決定された処理と、トランザクションに関連するリモート・ドメインの情報が記録されます。OSI TP の場合、XAP-TP インターフェイスにより、OSI TP プロトコル・マシンの回復に必要な情報が記録されます。blob (バイナリ・ラージ・オブジェクト) と呼ばれるこの情報は、コミット情報と同じログ・レコードに記録され、回復を行うときに使用されます。

Domains のログ・レコードの構造は、BEA Tuxedo システムの TLOG の構造とは異なります。TLOG レコードのサイズは決まっており、単一のページに格納されています。一方、Domains のログ・レコードのサイズは可変であり、レコードの大きさによっては、複数のページが必要な場合もあります。Domains のログ・メカニズムである DMTLOG では、さまざまなサイズのログ・レコードを格納できます。

TMS がドメイン・ゲートウェイ・グループより上位の場合は、コミットの調整のために BEA Tuxedo の TLOG が必要です。

ログ機能のしくみ

ログは、GWADM 管理サーバによって記録されます。ログへの書き込みは、GWTDOMAIN プロセスによって要求されますが、実際の書き込みは、GWADM プロセスによって実行されます。

各ドメイン・ゲートウェイ・グループには、DMTLOG というログ・ファイルを作成する必要があります。DMTLOG ファイルは、DMCONFIG ファイルの DM_LOCAL_DOMAINS セクションで定義されます。DMTLOG ファイルを作成するには、DMTLOGDEV パラメータにエントリを追加します。

```
DMTLOGDEV=string
```

string はログ・ファイルの名前です。さらに、次の 2 つのオプション・パラメータのどちらか、または両方を設定できます。

- DMTLOGNAME=*identifier*
- DMTLOGSIZE=*numeric*

詳細については、『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』の DMCONFIG(5) を参照してください。

管理者は、実行時管理ユーティリティ (dadmin) を使用して DMTLOG を作成することもできます。詳細については、『BEA Tuxedo コマンド・リファレンス』の `dadmin(1)` を参照してください。

ドメイン・ゲートウェイ・グループの起動時に、DMTLOG が作成されていないと、ゲートウェイ・サーバは、BDMCONFIG ファイルの情報に基づき、ログを自動的に作成します。

BDMCONFIG ファイルでログ・デバイスが指定されない限り、ドメイン・ゲートウェイ・グループは、要求をトランザクション・モードで実行できず、ゲートウェイ・グループは TMS サービスを提供できません。

コミット・プロトコルを調整するため、Domains ゲートウェイでは、次の 2 つのログ・レコードが必要です。

- レディ・レコード トランザクション・ツリーのリーフ、または中間ノードとして機能するファイルです。ゲートウェイによって作成されます。このファイルには、トランザクションに関連する上位および下位のリモート・ドメインについての情報が記録されます。レディ・レコードは、レコードを記録するドメイン・ゲートウェイ・グループの下位のものがすべて準備できたことを示します。
- コミット・レコード トランザクションがコミットされたことを記録します。ドメイン・ゲートウェイは、特定のトランザクション・ツリーのコーディネータとして、コミット・レコードを作成します。

トランザクションがすべてのマシンでコミットされると、そのトランザクションのログは削除されます。

OSI TP プロトコルを使用する場合は、次の 2 つのヒューリスティックなレコードが記録されます。

- ログ・ヒューリスティック・レコード ドメインで行われたヒューリスティックな決定の詳細を記録します。この情報は、適切なトランザクションの結果が上位ドメインに通知されるまで保持されます。
- ログ破損レコード トランザクション・ブランチの状態、つまり、ヒューリスティック・ハザード (下位ドメインのトランザクション・ブランチの結果が不明) またはヒューリスティック・ミックス (トランザクション・サブツリーの結果が混在) のいずれかを示します。これは、`tmadmin(1)` を実行して作成します。

ヒューリスティックなログ・レコードは、管理者によって明示的に削除されない限り、削除されません。この特性は、クラッシュ時の回復処理で正しい情報を取得し、管理者に対して診断情報を提供するために必要です。

管理者は、`forgettran` コマンド (`tmadmin(1)` で実行) を使用して、不要になったヒューリスティック・レコードを削除することができます。

失敗したトランザクションの回復

ドメイン・ゲートウェイ・グループが起動すると、ゲートウェイ・サーバは、`DMTLOG` を自動的にウォームスタートします。ウォームスタートでは、ログがスキャンされ、未完了のトランザクションがあるかどうかをチェックされます。未完了のトランザクションが見つかったら、そのトランザクションを処理するアクションが実行されます。

OSI TP では、`DMTLOG` 内の *blob* にあるトランザクション・レコードが、ネットワーク・アクセス・モジュールに渡されます。渡された *blob* は、内部状態を再構築し、失敗した接続を回復するために使用されます。

ドメイン・ゲートウェイ・グループがローカル TMS の下位であり、ヒューリスティックな決定が行われた場合、TMS は、最終的に決定された処理を示す `TMS_STATUS` メッセージを生成します。

- ゲートウェイが失敗した場合は、再起動後にゲートウェイ自体がクリーンアップされます (ホット・スタート)。ゲートウェイは、関連する未処理のトランザクションをすべてロールバックします。
- 第 1 フェーズのコミットが完了していない場合に通信回線エラーが発生すると、ゲートウェイは、その接続に関連するトランザクションをロールバックします。
- OSI TP Domains が使用されている場合に、トランザクションが第 2 フェーズのコミットに失敗すると、XAP-TP によって回復処理が行われます。