



BEATuxedo®

BEA Tuxedo Domains コンポーネ ント

Copyright

Copyright © 2003 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, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

目次

このマニュアルについて

対象読者	vii
e-docs Web サイト	viii
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	x

1. Domains について

BEA Tuxedo の Domains コンポーネントとは	1-2
ドメイン間の相互運用性	1-4
ドメイン・ゲートウェイの種類	1-4
Domains コンフィギュレーションの例	1-6
ドメイン・ゲートウェイでサポートされる機能	1-7
ローカル・ドメインとリモート・ドメイン間での要求 / 応答型の通信	1-8
ローカル・ドメインとリモート・ドメイン間での会話型通信	1-10
リモート・ドメインでのメッセージのキュー登録	1-10
Domains の符号化および復号化操作	1-11
BEA Tuxedo Domains のアーキテクチャ	1-12
Domains コンフィギュレーション・ファイル	1-13
ドメイン・ゲートウェイ・サーバ	1-13
Domains 管理サーバ	1-14
Domains 管理ツール	1-16
Domains コンフィギュレーション・ファイル	1-19
DMCONFIG ファイルの場所	1-19
バイナリ形式の DMCONFIG ファイル	1-20
DMCONFIG ファイルのセクションの記述	1-20
DMCONFIG ファイル関連の新しい用語	1-29
Domains データ依存型ルーティングの指定	1-30
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 の指定	1-32

Domains コンポーネントによるトランザクション・タイムアウトの処理	1-32
Domains コンポーネントによるブロッキング・タイムアウトの処理	1-34
Domains の接続方針の指定	1-36
接続方針の設定方法	1-37
接続再試行プロセスの使用方法	1-39
接続方針によるリモート・サービスの可用性の判断	1-40
Domains のフェイルオーバーとフェイルバックの指定	1-42
Domains レベルのフェイルオーバーとフェイルバックの設定	1-42
Domains リンク・レベルのフェイルオーバーの設定	1-43
Domains のキープアライブの指定	1-44
TCP レベルのキープアライブ	1-45
Domains の TCP レベルのキープアライブの設定	1-46
アプリケーション・レベルのキープアライブ	1-48
Domains のアプリケーション・レベルのキープアライブの設定	1-49
旧リリースの BEA Tuxedo とのキープアライブの互換性	1-52
Domains 環境の設定	1-52
移行を考慮した Domains 環境の設定	1-55
DMADM サーバの移行	1-58
TDomain ゲートウェイ・グループの移行	1-58
個々のサーバ・プロセスをアクティブにする手段	1-59

2. ATMI Domains の計画とコンフィギュレーション

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画	2-2
オプション 1: 単一の BEA Tuxedo ドメインとしてアプリケーションを再設定する	2-7
オプション 2: アプリケーションを Domains コンフィギュレーションとして再設定する	2-13
creditapp Domains コンフィギュレーションの検証	2-23
Domains コンフィギュレーションの設定	2-29
サンプル Domains アプリケーション (simpapp) の設定	2-29
コンフィギュレーション・タスク	2-30
lapp の環境変数の設定	2-32
UBBCONFIG ファイルでの lapp の Domains 環境の定義	2-33
DMCONFIG ファイルでの lapp の Domains パラメータの定義	2-35
lapp のアプリケーションおよび Domains ゲートウェイ・コンフィギュ	

レーション・ファイルのコンパイル	2-39
rapp の環境変数の設定	2-40
UBBCONFIG ファイルでの rapp の Domains 環境の定義	2-41
DMCONFIG ファイルでの rapp の Domains パラメータの定義	2-43
rapp のアプリケーションおよび Domains ゲートウェイ・コンフィギュ レーション・ファイルのコンパイル	2-44
ドメイン間のデータの圧縮	2-46
リモート・ドメインへのサービス要求のルーティング	2-46
Domains コンフィギュレーションのセキュリティの設定	2-47
Domains のセキュリティ・メカニズム	2-48
Domains 認証のプリンシパル名の設定	2-49
Domains パスワード・セキュリティの設定	2-51
Domains アクセス制御リストの設定	2-59
リモート・ドメインの ACL 方針の設定	2-60
Domains リンク・レベル暗号化の設定	2-62
Domains コンフィギュレーションの接続の設定	2-63
クライアントの要求に応じた接続の要求 (ON_DEMAND 方針)	2-64
起動時の接続の要求 (ON_STARTUP 方針)	2-65
接続を受信メッセージのみに制限する (INCOMING_ONLY 方針)	2-66
ON_STARTUP のみでの接続再試行間隔の設定	2-66
最大再試行回数の設定	2-67
ドメイン間の接続方針のコーディング例	2-68
Domains コンフィギュレーションの接続の制御	2-69
ドメイン間の接続の確立	2-69
ドメイン間の接続の切断	2-70
接続状態の報告	2-70
Domains リンク・レベルのフェイルオーバーとキープアライブの設定	2-71

3. CORBA Domains の計画とコンフィギュレーション

CORBA Domains 環境の概要	3-2
シングル・ドメイン通信とマルチ・ドメイン通信	3-2
シングル・ドメイン通信	3-3
マルチ・ドメイン通信	3-4
CORBA Domains コンフィギュレーションの要素	3-5
コンフィギュレーション・ファイルの使い方	3-7

UBBCONFIG ファイル	3-8
DMCONFIG ファイル	3-9
factory_finder.ini ファイル	3-21
factory_finder.ini ファイルでの一意なファクトリ・オブジェクト識別子の指 定	3-26
factory_finder.ini ファイルの処理	3-27
CORBA Domains・コンフィギュレーションの種類.....	3-28
直接接続ドメイン	3-28
間接接続ドメイン	3-28
CORBA Domains・コンフィギュレーションの例	3-30
UBBCONFIG ファイルの例	3-30
DMCONFIG ファイルの例.....	3-35
factory_finder.ini ファイルの例	3-41

4. Domains の管理

Domains の実行時管理コマンドを使用する	4-1
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
ドメイン・ゲートウェイ・サーバを使用する	4-8
Domains 環境でのトランザクションの管理	4-8
Domains で TMS 機能を使用する	4-10
トランザクションで GTRID マッピングを使用する	4-13
ログ機能によるトランザクションのトラッキング	4-21
失敗したトランザクションの回復.....	4-24

このマニュアルについて

このマニュアルでは、BEA Tuxedo ATMI および CORBA 両方の環境における BEA Tuxedo Domains コンポーネントの設定と管理の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 Domains コンポーネントの概要を説明します。
- [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 Domains 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 Domains 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 Domains 環境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 システムについて理解してい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、e-docs Web サイトの BEA Tuxedo マニュアル・ページから入手できます。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 マニュアル・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 社の Web サイト (<http://www.adobe.co.jp/>) から無償でダウンロードできます。

関連情報

次の BEA Tuxedo ドキュメントには、BEA Tuxedo Domains コンポーネントに関連する情報が記載されています。

- 『製品の概要』

-
- 『BEA Tuxedo アプリケーションの設定』
 - 『BEA Tuxedo アプリケーション実行時の管理』
 - 『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 ATMI 環境および CORBA 環境の詳細については、『参考資料』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 8.1 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、<http://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。
等幅テキスト	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。 例： <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	コード内の重要な単語を示します。 例： <pre>void commit ()</pre>
等幅イタリック体	コード内の変数を示します。 例： <pre>String <i>expr</i></pre>

規則	項目
大文字	デバイス名、環境変数、および論理演算子を示します。 例： LPT1 SIGNON OR
{ }	構文の行で選択肢を示します。かっこは入力しません。
[]	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。 例： buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
	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。
...	コマンド行で次のいずれかを意味します。 <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 省略符号は入力しません。 例： buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
.	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。

1 Domains について

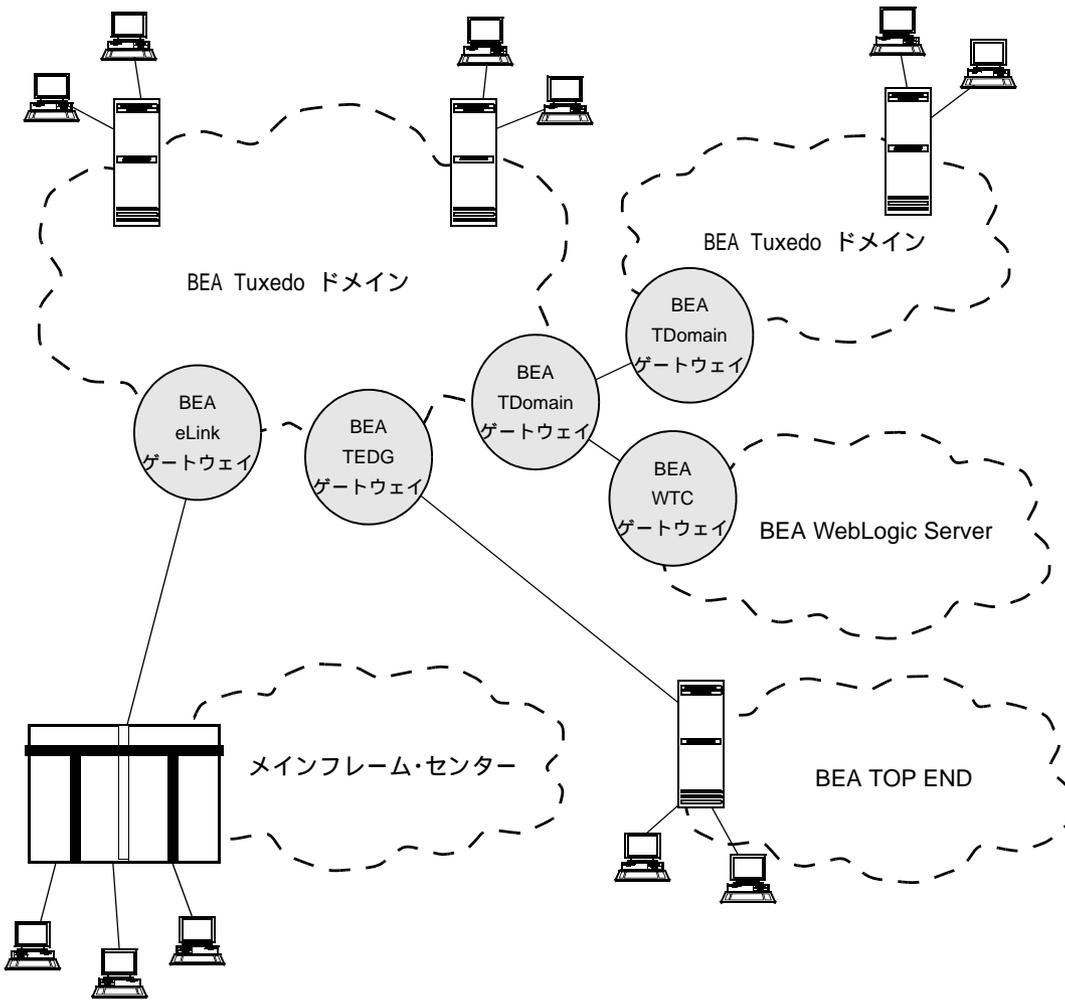
以下の節では、BEA Tuxedo Domains コンポーネントの概要を説明します。

- [BEA Tuxedo の Domains コンポーネントとは](#)
- [Domains コンフィギュレーションの例](#)
- [ドメイン・ゲートウェイでサポートされる機能](#)
- [BEA Tuxedo Domains のアーキテクチャ](#)
- [Domains コンフィギュレーション・ファイル](#)
- [Domains データ依存型ルーティングの指定](#)
- [Domains のトランザクション・タイムアウトとブロッキング・タイムアウトの指定](#)
- [Domains の接続方針の指定](#)
- [Domains のフェイルオーバーとフェイルバックの指定](#)
- [Domains のキープアライブの指定](#)
- [Domains 環境の設定](#)
- [移行を考慮した Domains 環境の設定](#)

BEA Tuxedo の Domains コンポーネントとは

企業のビジネスが拡大するに従って、機能、地理、機密度に基づいて分かれている、それぞれが管理上独立したアプリケーションにビジネス情報管理を取り入れる必要が出てきます。それらの独立したビジネス・アプリケーションはドメインと呼ばれ、情報を共有する必要があります。BEA Tuxedo Domains コンポーネントはビジネスのドメイン間で相互運用を実現するインフラストラクチャを提供し、それによって BEA Tuxedo クライアント / サーバ・モデルが複数のトランザクション処理 (TP) ドメインに拡張されます。次の図は、BEA Tuxedo Domains コンポーネントによって、どのようにして複数のドメインが結び付けられるのかを示しています。

図 1-1 BEA Tuxedo Domains コンポーネントを使用したドメイン間通信



ドメイン間の相互運用性

リモート・ドメインのサービスをローカル・ドメインのユーザが透過的に利用できるようにしたり、ローカル・ドメインのサービスをリモート・ドメインのユーザが利用できるようにすることで、BEA Tuxedo Domains コンポーネントは企業のビジネス・アプリケーションの間にある壁を取り払います。また、Domains コンポーネントを利用することで、BEA Tuxedo アプリケーションを実行する企業は、ほかのトランザクション処理 (TP) システム (BEA 社の WebLogic Server、BEA 社の TOP END、IBM/Transarc の Encina、IBM の CICS など) で動作するアプリケーションとの相互運用を実現してビジネスを拡張することができます。

企業ではビジネス・アプリケーションの性質をその名前の一部としてよく使用するので、アプリケーションの名前は「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 ドメインは、UBBCONFIG という 1 つのコンフィギュレーション・ファイルで制御される単一コンピュータまたは複数コンピュータのネットワークです。BEA Tuxedo のコンフィギュレーション・ファイルには、どのような名前でも付けられます。ただし、そのファイルの内容は『BEA Tuxedo のファイル形式とデータ記述方法』の UBBCONFIG(5) リファレンス・ページで指定された形式に準拠している必要があります。BEA Tuxedo ドメインは、1 つのユニットとして管理されます。

ドメイン・ゲートウェイの種類

BEA Tuxedo Domains コンポーネントは、各種のネットワークおよびドメインと通信するためのさまざまなゲートウェイを提供します。具体的には、Domains コンポーネントでは以下のドメイン・ゲートウェイが提供されます。

- BEA Tuxedo TDomain ゲートウェイ (GWTDOMAIN サービス・プロセスでインプリメント) ネットワーク・プロトコル TCP/IP 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 ドメイン間での相互運用性を実現します。WebLogic Tuxedo Connector (WTC) ゲートウェイ (BEA WebLogic Server コンポーネント) と関係することで、

BEA Tuxedo TDomain ゲートウェイは Tuxedo ドメインと WebLogic Server アプリケーションの間でも相互運用性を実現できます。

- BEA TOP END ドメイン・ゲートウェイ (TEDG) (GW_{TOPEND} サーバ・プロセスでインプリメント) ネットワーク・プロトコル TCP/IP 経由で BEA Tuxedo ドメインと TOP END システムの間の相互運用性を実現します。
- BEA eLink Adapter for Mainframe TCP ゲートウェイ (GW_{IDOMAIN} サーバ・プロセスでインプリメント) ネットワーク・プロトコル TCP/IP 経由で BEA Tuxedo ドメインと、IBM OS/390 顧客情報管理システム (CICS) および情報管理システム (IMS) の下で動作するアプリケーションの間の相互運用性を実現します。ゲートウェイでは非トランザクション・タスクのみサポートされます。
- BEA eLink Adapter for Mainframe SNA ゲートウェイ (GW_{SNAX} サーバ・プロセスでインプリメント) システム・ネットワーク体系 (SNA) 拡張プログラム間通信機能 (APPC) または共通プログラミング・インターフェイス・コミュニケーション (CPI-C) がサポートされるプラットフォーム (IBM OS/400、OS/390 CICS および IMS システム、および VSE/CICS) で動作する BEA Tuxedo ドメインとアプリケーション間の相互運用性を実現します。ゲートウェイでは複数の SNA ネットワークとの通信がサポートされます。
- BEA eLink Adapter for Mainframe OSI TP ゲートウェイ (GW_{OSITP} サーバ・プロセスでインプリメント) BEA Tuxedo ドメインと、開放型システム間相互接続 (OSI) トランザクション処理 (TP) 規格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) によって定義された分散トランザクション処理のためのプロトコルです。ゲートウェイでは、グローバル・トランザクションとさまざまな非トランザクション・タスクがサポートされます。

これ以降は、BEA TDomain ゲートウェイおよび BEA Tuxedo ドメイン間の通信を中心に説明します。WTC ゲートウェイについては、次のドキュメントを参照してください。

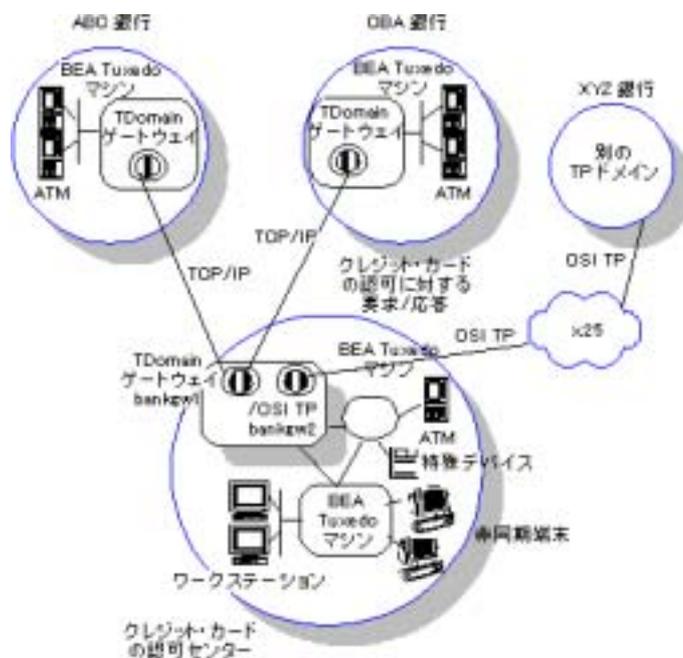
- 『製品の概要』
- 『WebLogic Tuxedo Connector』

TEDG ゲートウェイについては、『[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。BEA eLink のゲートウェイについては、『[BEA eLink Documentation](#)』を参照してください。

Domains コンフィギュレーションの例

次の図は、ドメインが4つのサンプル Domains コンフィギュレーションを示しています。そのうちの3つは BEA Tuxedo ドメインです。

図 1-2 銀行業務の Domains コンフィギュレーションの例



図の一番下にある BEA Tuxedo クレジット・カード認可センターには、bankgw1 という TDomain ゲートウェイ・グループと bankgw2 という OS/TP ゲートウェイ・グループの2つのゲートウェイ・グループがあります。bankgw1 は、ネットワーク・プロトコル TCP/IP を使用して2つのリモート

BEA Tuxedo 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、ネットワーク・プロトコル OSI TP 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。その要求は、グループ bankgw1 で動作している GWTDOMAIN というドメイン・ゲートウェイ・サーバ・プロセスによって受信されます。このゲートウェイは、ローカルで動作している別のサーバ・プロセス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。このサーバは、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、GWOSITP というドメイン・ゲートウェイ・サーバ・プロセスを通じて XYZ 銀行に残高照会を要求できます。

BEA Tuxedo Domains コンポーネントは、リモート・サービス (ほかのドメインのサービス) をローカル・サービスであるかのように宣言するドメイン・ゲートウェイ・サーバ・プロセスを通じてドメイン間通信を実現します。

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。

- マルチネットワークのサポート ゲートウェイは、TCP/IP、IPX/SPX、OSI 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、ゲートウェイは、そのリンク先であるネットワーク・ライブラリの機能に制限を受けます。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。たとえば、BEA Tuxedo TDomain ゲートウェイでは TCP/IP のみサポートされます。

- 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。
- トランザクション管理 - ゲートウェイにより、ATMI アプリケーション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。
- 複数のメッセージング・モデル - ゲートウェイは、次の ATMI メッセージング・モデルをサポートしています。既存の BEA Tuxedo アプリケーションを修正する必要はありません。
 - **要求 / 応答型モデル** - BEA Tuxedo システムを使用する ATMI アプリケーションは、ほかのドメインで動作するアプリケーションのサービスを要求できます。
 - **会話型モデル** - ATMI アプリケーションは、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。
 - **キューの処理モデル** - BEA Tuxedo システムを使用する ATMI アプリケーションは、ほかのドメインのキューにデータを保存できます。
- **型付きバッファのサポート** - ゲートウェイは、BEA Tuxedo ATMI 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。

ローカル・ドメインとリモート・ドメイン間での要求 / 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。論理的に 1 つのアプリケーション内で使用するよう制限されており、ドメイン間での使用はサポートされていない以下の BEA Tuxedo ATMI 関数を除いて、BEA Tuxedo アプリケーションはローカル・サービスの場合とまったく同じようにリモート・サービスを要求できます。

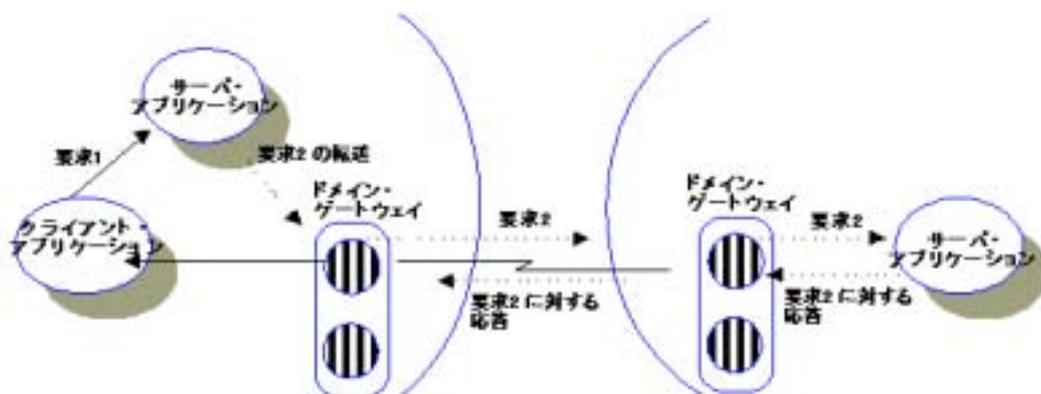
- `tpinit(3c)/tpterm(3c)` - BEA Tuxedo アプリケーションは、リモート・ドメインの環境にはアタッチされません。リモート・ドメインへのアクセスは、ドメイン・ゲートウェイを使用して行われます。したがって、

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)` Domains ではこれらの関数はサポートされていません。ドメイン・ゲートウェイで、ドメイン間の動的なサービスの宣言がサポートされていないからです。
- `tpnotify(3c)` および `tpbroadcast(3c)` Domains では、これらの関数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 1-3tpforward を使用してリモート・サービスに要求を送信する



BEA Tuxedo 要求 / 応答型モデルの詳細については、『BEA Tuxedo システム入門』の 2-13 ページの「要求 / 応答型通信」を参照してください。

ローカル・ドメインとリモート・ドメイン間での会話型通信

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会話型のモデルをサポートしています。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、会話型モデル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 `TPFAIL` 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。

BEA Tuxedo 会話型モデルの詳細については、『BEA Tuxedo システム入門』の [2-14 ページ](#) の「[会話型通信](#)」を参照してください。

リモート・ドメインでのメッセージのキュー登録

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キューの処理モデルをサポートしています。クライアントやサーバはどれも、リモート・ドメインのキューにメッセージまたはサービス要求を格納できます。格納された要求はすべて、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。

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。

BEA Tuxedo キューの処理モデルの詳細については、『BEA Tuxedo システム入門』の 2-15 ページの「メッセージ・キューイング通信」を参照してください。

Domains の符号化および復号化操作

ドメイン・ゲートウェイは、ドメイン・ゲートウェイ・サーバ・プロセスが実行されているリリースの BEA Tuxedo システム・ソフトウェアによってサポートされているすべての定義済み型付きバッファをサポートします。リリース 8.1 の BEA Tuxedo は、11 種の定義済みバッファ型をサポートしています。

BEA Tuxedo リリースでサポートされている各バッファ型には、プログラムの介入なしで初期化、メッセージの送受信、およびデータの符号化 / 復号化を行うために自動的に呼び出すことができるそれ独自のルーチン・セットがあります。それらのルーチンは、型付きバッファ・スイッチと呼ばれます。

BEA Tuxedo ATMI 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 (サービス要求と応答) を送信するために型付きバッファが使用されます。その名前のおりそれ自身についての情報 (メタデータ) が含まれている型付きバッファにより、アプリケーション・プログラマは、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、ワークステーション、ローカルの BEA Tuxedo マシン、およびリモート・ドメインから送信されるサービス要求を受信し、処理できます。ドメイン・ゲートウェイは、次の理由により、符号化されている受信したすべてのサービス要求を適切な型付きバッファ・スイッチを使用して復号化します。

- 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。したがって、ドメイン・ゲート

ウェイでは、データを適切にリモート・ドメインにルーティングするために符号化されたデータを復号化する必要があります。

- 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・ゲートウェイでは符号化されたデータを復号化してどのデータ形式が使用されているのかを判別する必要があります。

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。符号化 / 復号化をサポートする定義済みのバッファ型について、BEA Tuxedo システムではそれらの型を 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。

型付きバッファおよび符号化 / 復号化操作の詳細については、『BEA Tuxedo システム入門』の [2-27 ページ](#) の「[型付きバッファ](#)」を参照してください。

BEA Tuxedo Domains のアーキテクチャ

BEA Tuxedo Domains のアーキテクチャは、主に次の 4 つの要素で構成されています。

- Domains コンフィギュレーション・ファイル
- ドメイン・ゲートウェイ・サーバ
- Domains 管理サーバ
- Domains 管理ツール

Domains コンフィギュレーション・ファイル

Domains コンフィギュレーションは、BEA Tuxedo Domains コンポーネントを介して通信したりサービスを共有したりできる複数のドメイン（アプリケーション）の集合です。複数のドメインがどのように接続されるのか、およびどのサービスが互いに利用可能になるのかは、Domains コンフィギュレーション・ファイルで定義します。Domains コンフィギュレーションに関する各 BEA Tuxedo ドメインでは、それ専用の Domains コンフィギュレーション・ファイルが必要です。

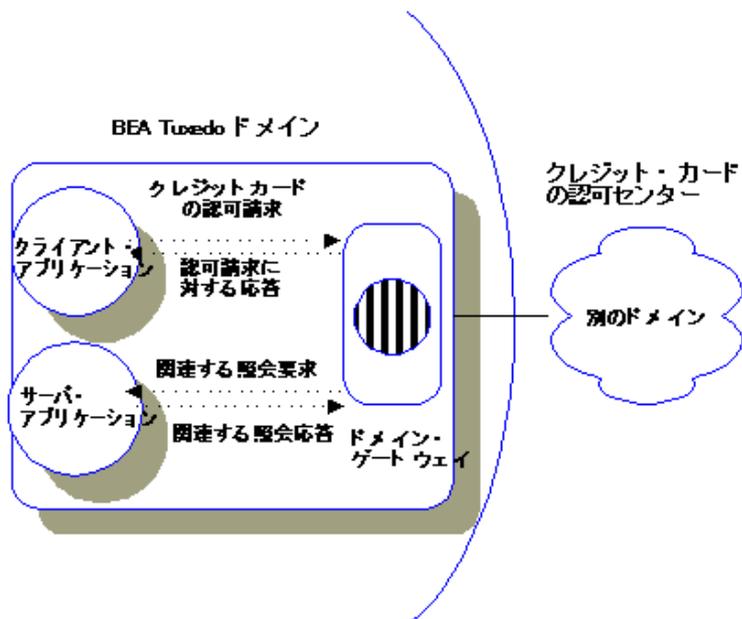
テキスト形式の Domains コンフィギュレーション・ファイルは `DMCONFIG` ファイルと呼ばれますが、任意の名前を付けることもできます。ただし、そのファイルの内容が『BEA Tuxedo のファイル形式とデータ記述方法』の `DMCONFIG(5)` リファレンス・ページで説明されている形式に準拠している場合に限ります。バイナリ形式の Domains コンフィギュレーション・ファイルは、`BDMCONFIG` と呼ばれます。`DMCONFIG` ファイルの詳細については、[1-19 ページの「Domains コンフィギュレーション・ファイル」](#)を参照してください。

ドメイン・ゲートウェイ・サーバ

BEA Tuxedo Domains コンポーネントは、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ドメイン・ゲートウェイ・プロセスを使用してマルチ・ドメインの相互運用性を実現します。そのドメイン・ゲートウェイ・プロセスは、アプリケーション・プログラマとアプリケーション・ユーザが両方とも別のドメインのサービスに透過的にアクセスできるようにする BEA Tuxedo 提供サーバです。

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 1-4 ゲートウェイを介した双方向通信

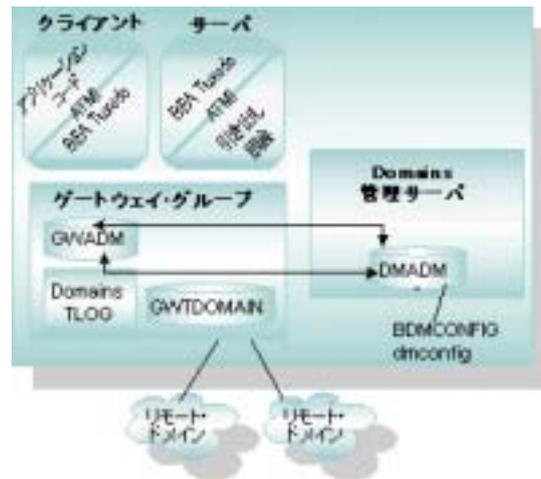


この図では、ドメイン・ゲートウェイがクレジット・カードの認可要求を処理し、別のドメインに送信する様子を示しています。このゲートウェイは、認可要求に対する応答も処理します。

Domains 管理サーバ

次の図は、Domains コンフィギュレーションを管理するために使用される BEA Tuxedo Domains 管理サーバを示しています。

図 1-5 Domains 管理サーバ



ドメイン・ゲートウェイ・グループは、前の図で示されているように、ゲートウェイ管理サーバ (GWADM)、ドメイン・ゲートウェイ・サーバ (GWTDOMAIN など)、および (オプションの) Domains トランザクション・ログ (TLOG) で構成されます。GWADM サーバは、ドメイン・ゲートウェイの実行時における管理を可能にします。BEA Tuxedo ドメインは、ドメイン・ゲートウェイ・グループを通じて 1 つ以上のリモート・ドメインと通信できます。

BEA Tuxedo ドメインで動作するすべてのドメイン・ゲートウェイ・グループと関連付けられているのは、Domains 管理サーバ (DMADM) です。Domains 管理サーバは、BEA Tuxedo Domains コンフィギュレーション・ファイル (BDMCONFIG) の実行時における管理を可能にします。

GWADM サーバ

GWADM (5) サーバは、DMADM サーバに登録して、対応するゲートウェイ・グループで使用されるコンフィギュレーション情報を取得します。GWADM は、DMADMIN サービスからの要求、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。DMADMIN サービスは、DMADM によって宣言された汎用管理サービスです。GWADM は、定期的に "I-am-alive" メッセージを DMADM サーバに送信します。

DMADM から応答がなければ、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、そのゲートウェイ・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。

GWADM の詳細については、4-1 ページの「[Domains の管理](#)」を参照してください。

DMADM サーバ

DMADM(5) サーバは、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 サーバの初期化プロシージャの一部として GWADM サーバによって要求されます。登録サービスは、要求元のゲートウェイ・グループが要求するコンフィギュレーション情報をダウンロードします。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、Domains 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。

DMADM の詳細については、4-1 ページの「[Domains の管理](#)」を参照してください。

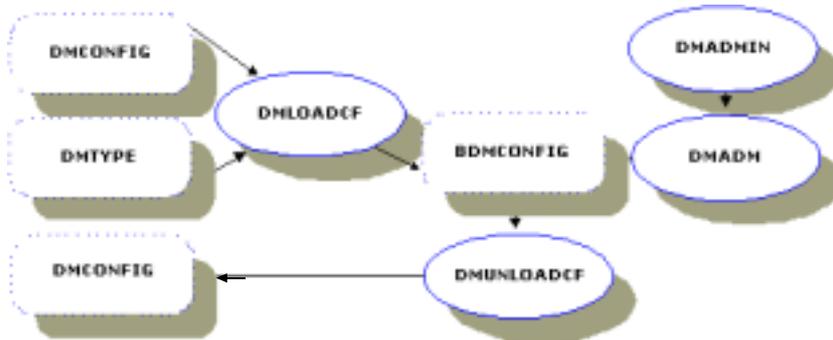
Domains 管理ツール

次の Domains 管理ツールは、Domains コンフィギュレーションの設定と管理のために BEA Tuxedo システムによって提供されます。

- `dmloadcf(1)` DMCONFIG ファイルを読み取り、構文をチェックし、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(1)` 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。
- `dmadmin(1)` BEA Tuxedo の管理者が Tuxedo ドメインの実行時に BDMCONFIG ファイルを更新できるようにします。

次の図は、Domains 管理ツールと Domains のテキスト形式およびバイナリ形式のコンフィギュレーション・ファイルの関係を示しています。dmadmin ユーティリティを使用する管理は、DMADM サーバで宣言される DMADMIN サービスを通じて行います。

図 1-6 Domains 管理ツールとファイルの関係



dmloadcf コマンド

dmloadcf(1) コマンドは、DMCONFIG ファイルを解析して、その情報を BDMCONFIG にロードします。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るデバイス・ファイルまたはシステム・ファイルの名前を示します。

dmloadcf 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・ドメインに必要なプロセス間通信 (IPC) 資源の量を計算できます。

dmloadcf コマンドは、DMTYPE ファイル (Windows の場合は `%TUXDIR%\udataobj\DMTYPE`、UNIX の場合は `$TUXDIR/udataobj/DMTYPE`) をチェックして、Domains 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・ゲートウェイ・タイプが有効であるかどうかを検証します。各タイプのドメイン・ゲートウェイには、DMTYPE ファイルでタグとして使用されるドメイン・タイプ指定子 (TDOMAIN、TOPEND、SNAX、OSITP、OSITPX) があります。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 ゲートウェイ用のエントリがあります。

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

`dmloadcf` の詳細については、『BEA Tuxedo コマンド・リファレンス』の [dmloadcf\(1\)](#) リファレンス・ページを参照してください。

dmunloadcf コマンド

`dmunloadcf(1)` コマンドは、`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て、標準出力に出力します。

`dmunloadcf` の詳細については、『BEA Tuxedo コマンド・リファレンス』の [dmunloadcf\(1\)](#) リファレンス・ページを参照してください。

dmadmin コマンド

`dmadmin(1)` コマンドは、BEA Tuxedo の管理者が Tuxedo の実行中にドメイン・ゲートウェイを設定、監視、およびチューニングできるようにします。このコマンドは、管理コマンドを変換して要求を `DMADMIN` サービス (`DMADM` サーバにより宣言される汎用管理サービス) に送信する管理コマンド・インタプリタとしても機能します。この結果、`DMADMIN` サービスは、`BDMCONFIG` 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。

`-c` オプションを指定して `dmadmin` を呼び出すと、`BDMCONFIG` ファイルが動的に更新されます。変更されるコンフィギュレーションに応じて、一部の更新はすぐに行われ、ほかの更新はその更新で影響を受けるものが新しく発生したときにのみ行われます。

`dmadmin` の詳細については、[4-1 ページの「Domains の管理」](#)を参照してください。

Domains コンフィギュレーション・ファイル

Domains コンフィギュレーションに關与する各 BEA Tuxedo ドメインには、ドメイン間パラメータの定義されるコンフィギュレーション・ファイルがあります。テキスト形式のコンフィギュレーション・ファイルは `DMCONFIG` と呼ばれますが、コンフィギュレーション・ファイルには任意の名前を付けることができます。ただし、そのファイルの内容が『BEA Tuxedo のファイル形式とデータ記述方法』の [DMCONFIG\(5\)](#) リファレンス・ページで説明されている形式に準拠している場合に限ります。典型的なコンフィギュレーション・ファイル名は、文字列 `dm` で始まり、その後にはファイル名 `dmconfig` の `config` のようなニーモニック文字列が続きます。

Domains コンフィギュレーションの管理者は、コンフィギュレーションに参加する BEA Tuxedo ドメインごとに別個の `DMCONFIG` ファイルを作成する必要があります。`DMCONFIG` ファイルは、任意のテキスト・エディタで作成および編集できます。

DMCONFIG ファイルの場所

Domains コンフィギュレーションに關与する BEA Tuxedo ドメインの `DMCONFIG` ファイルは、Tuxedo ドメインの `UBBCONFIG` ファイルで指定されているとおりに、Domains 管理サーバ `DMADM` が実行されるマシン上に配置されます。`DMADM` サーバは、Tuxedo ドメインのどのマシン（マスタ・マシン、非マスタ・マシン）でも実行できます。

注記 BEA Tuxedo ドメインのマスタ・マシンは、ドメインの `UBBCONFIG` ファイルが格納され、`UBBCONFIG` ファイルの `RESOURCES` セクションでマスタ・マシンとして指定されます。Tuxedo ドメインは、マスタ・マシンを使用して起動、停止、および管理します。

バイナリ形式の DMCONFIG ファイル

BDMCONFIG ファイルは、DMCONFIG ファイルのバイナリ形式です。dmloadcf コマンドを実行することで作成されます。このコマンドは、DMCONFIG を解析して、バイナリ形式の BDMCONFIG ファイルを BDMCONFIG 環境変数で示された位置にロードします。DMCONFIG と同じように、BDMCONFIG ファイルにも任意の名前を付けることができます。その実際の名前は、BDMCONFIG 環境変数で指定されたデバイス・ファイル名またはシステム・ファイル名です。BDMCONFIG 環境変数は、BDMCONFIG がロードされるデバイス・ファイル名またはシステム・ファイル名で終わる絶対パス名に設定する必要があります。

UBBCONFIG のバイナリ形式である TUXCONFIG ファイルと違って、BDMCONFIG ファイルは Tuxedo アプリケーションの起動時に Tuxedo ドメインのほかのどのマシンにも複製転送されません。BDMCONFIG ファイルを Tuxedo ドメインのほかのマシンにも配置するためには、そのドメインの管理者が手作業で配置する必要があります。

DMCONFIG ファイルのセクションの記述

DMCONFIG ファイルは、複数の指定セクションで構成されます。セクションは、アスタリスク (*) が先頭に付いた行から始まります。アスタリスク (*) の直後にはセクション名が表示されます。セクション名を指定するときにはアスタリスクが必要です。

使用可能なセクションは、

- DM_LOCAL (DM_LOCAL_DOMAINS と呼ばれる)
- DM_REMOTE (DM_REMOTE_DOMAINS と呼ばれる)
- DM_EXPORT (DM_LOCAL_SERVICES と呼ばれる)
- DM_IMPORT (DM_REMOTE_SERVICES と呼ばれる)
- DM_RESOURCES
- DM_ROUTING
- DM_ACCESS_CONTROL

- `DM_domtype` (`domtype` は `TDOMAIN`、`TOPEND`、`OSITP`、`OSITPX`、または `SNACRM + SNALINKS + SNASTACKS`)

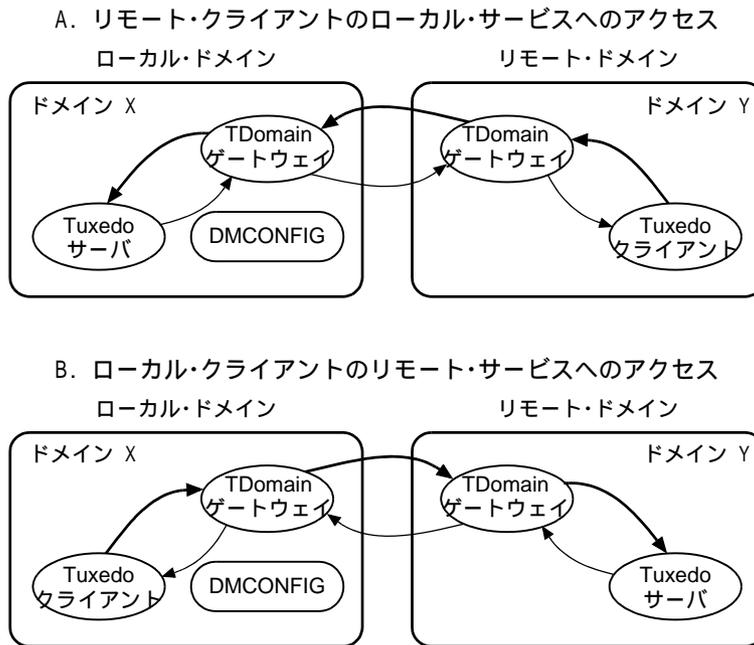
注記 `DM_LOCAL` セクションは、`DM_REMOTE` セクションの前に置く必要があります。

Domains コンフィギュレーションの管理者は、以上のセクションを以下の目的で使用します。

- リモート・ドメインのアプリケーション・クライアントがローカル・ドメインのサービスにアクセスするためのローカル・ドメイン・アクセス・ポイントを定義します。
- 各ローカル・ドメイン・アクセス・ポイントを通じて利用可能なローカル・サービスを指定します。
- ローカル・ドメインのアプリケーション・クライアントがリモート・ドメインのサービスにアクセスするためのリモート・ドメイン・アクセス・ポイントを定義します。
- 各リモート・ドメイン・アクセス・ポイントを通じて利用可能なリモート・サービスを指定します。
- ローカル・ドメイン・アクセス・ポイントおよびリモート・ドメイン・アクセス・ポイントを特定のドメイン・ゲートウェイ・グループ (`TDOMAIN`、`TOPEND` など) およびネットワーク・アドレスにマッピングします。

次の図は、いまここで説明している作業の単純な例です。

図 1-72 つの BEA Tuxedo ドメインで共有されるサービスを確定する 例



この例では、ドメイン X に作成された DMCONFIG ファイルを補うドメイン Y の DMCONFIG ファイルも作成する必要があります。つまり、ドメイン X の DMCONFIG ファイルのローカル・ドメイン・アクセス・ポイントがドメイン Y の DMCONFIG ファイルでリモート・ドメイン・アクセス・ポイントになり、ドメイン X の DMCONFIG ファイルのリモート・ドメイン・アクセス・ポイントがドメイン Y の DMCONFIG ファイルでローカル・ドメイン・アクセス・ポイントになります。例では、TDomain ゲートウェイ・サーバの使い方が例示されています。

次の表では、DMCONFIG ファイルの各セクションについて説明します。

表 1-1DMCONFIG ファイルのセクション (シート 1 / 6)

セクション	目的
DM_LOCAL (DM_LOCAL_DOMAINS と も呼ばれる)	<p>1 つ以上のローカル・ドメイン・アクセス・ポイント識別子 (ローカル・ドメインまたは LDOM と呼ばれる) を定義します。定義した各ローカル・ドメイン・アクセス・ポイント (論理名) について、このセクションでアクセス・ポイントのドメイン・ゲートウェイ・グループ (TDOMAIN など) を指定し、アクセス・ポイントを通じて利用できるローカル・サービスを DM_EXPORT セクションで指定します。ローカル・ドメイン・アクセス・ポイントを通じて利用可能なローカル・サービスは、1 つ以上のリモート・ドメインのクライアントから利用できます。</p> <p>このセクションでは、この BEA Tuxedo ドメインがリモート・ドメインと通信するために使用する各ゲートウェイ・グループ (TDOMAIN、TOPEND、SNAX、OSITP、OSITPX) について 1 つずつ、複数のローカル・ドメイン・アクセス・ポイントを定義できます。</p> <p>ローカル・ドメイン・アクセス・ポイントは、ゲートウェイ・グループごとに 1 つのみ定義できます。ドメイン・ゲートウェイ・グループは、GWADM サーバ・プロセスとドメイン・ゲートウェイ・サーバ・プロセス (GWTDOMAIN、TDomain ゲートウェイ・サーバなど) で構成されます。</p> <p>次に、ローカル・ドメイン・アクセス・ポイントのエントリ例を示します。</p> <pre>*DM_LOCAL LOCAL1 GWGRP=GWTGROUP TYPE=TDOMAIN ACCESSPOINTID="BA.CENTRAL01"</pre> <p>注記 ACCESSPOINTID パラメータの代わりに DOMAINID を使用することもできます。</p>

表 1-1DMCONFIG ファイルのセクション (シート 2 / 6)

セクション	目的
DM_REMOTE (DM_REMOTE_DOMAINS とも呼ばれる)	<p>1 つ以上のリモート・ドメイン・アクセス・ポイント識別子 (リモート・ドメインまたは RDOM と呼ばれる) を定義します。定義した各リモート・ドメイン・アクセス・ポイント (論理名) について、このセクションでアクセス・ポイントのドメイン・ゲートウェイ・グループ (TDOMAIN など) を指定し、アクセス・ポイントを通じて利用できるリモート・サービスを DM_IMPORT セクションで指定します。リモート・ドメイン・アクセス・ポイントを通じて利用可能なリモート・サービスは、ローカル・ドメインのクライアントから利用できます。</p> <p>このセクションでは、この BEA Tuxedo ドメインがリモート・ドメインと通信するために使用する各ゲートウェイ・グループ (TDOMAIN、TOPEND、SNAX、OSITP、OSITPX) について 1 つ以上、複数のリモート・ドメイン・アクセス・ポイントを定義できます。</p> <p>次に、リモート・ドメイン・アクセス・ポイントのエントリ例を示します。</p> <pre>*DM_REMOTE REMOT1 TYPE=TDOMAIN ACCESSPOINTID="BA.BANK01" REMOT2 TYPE=TDOMAIN ACCESSPOINTID="BA.BANK02"</pre> <p>注記 ACCESSPOINTID パラメータの代わりに DOMAINID を使用することもできます。</p>

表 1-1DMCONFIG ファイルのセクション (シート 3 / 6)

セクション	目的
DM_EXPORT (DM_LOCAL_SERVICES とも呼ばれる)	<p>DM_LOCAL セクションで定義されたローカル・ドメイン・アクセス・ポイントを通じて1つ以上のリモート・ドメインにエクスポートされるローカル・サービスを定義します。ローカル・ドメイン・アクセス・ポイントで指定されたサービスのみ、1つ以上のリモート・ドメインのクライアントから利用できます。つまり、このセクションでサービスを指定すると、ローカル・サービスへのリモート・クライアントのアクセスが制限されます。DM_EXPORT セクションがないか、あっても何も指定されていない場合は、ローカル・ドメインで宣言されたすべてのサービスがリモート・ドメインから利用可能になります。</p> <p>リモート・ドメインから利用可能になったローカル・サービスは、ローカル UBBCONFIG ファイルの SERVICES セクションからそのプロパティの多くを継承します。継承される特性として、LOAD、PRIO、AUTOTRAN、ROUTING、BUFTYPE、TRANTIME があります。</p> <p>次に、リモート・ドメインから利用可能になったローカル・サービスの例を示します。</p> <pre>*DM_EXPORT LTOLOWER LACCESSPOINT=LOCAL1 CONV=N RNAME="TOWER" ACL=branch</pre> <p>注記 LACCESSPOINT パラメータの代わりに LDOM を使用することもできます。</p>

1 Domains について

表 1-1DMCONFIG ファイルのセクション (シート 4 / 6)

セクション	目的
DM_IMPORT (DM_REMOTE_SERVICES とも呼ばれる)	<p>DM_REMOTE セクションで定義された 1 つ以上のリモート・ドメイン・アクセス・ポイントを通じてインポートされ、1 つ以上のローカル・ドメイン・アクセス・ポイントを通じてローカル・ドメインから利用可能なリモート・サービスを指定します。DM_IMPORT セクションがないか、あっても何も指定されていない場合、ローカル・ドメインからはどのリモート・サービスも利用できません。</p> <p>ローカル・ドメインから利用可能になったリモート BEA Tuxedo サービスは、リモート UBBCONFIG ファイルの SERVICES セクションからそのプロパティの多くを継承します。継承される特性として、LOAD、PRIO、AUTOTRAN、ROUTING、BUFTYPE、TRANTIME があります。</p> <p>次に、ローカル・ドメインから利用可能になったリモート・サービスの例を示します。</p> <pre>*DM_IMPORT RTOUPPER AUTOTRAN=N RACCESSPOINT=REMOT1 LACCESSPOINT=LOCAL1 CONV=N RNAME="TOUPPER"</pre> <p>注記 RACCESSPOINT パラメータの代わりに RDOM、LACCESSPOINT パラメータの代わりに LDOM を使用することもできます。</p>
DM_RESOURCES	<p>グローバルの Domains コンフィギュレーション情報、具体的にはユーザ指定のコンフィギュレーション・バージョン文字列を指定します。このセクションでは、VERSION=<i>string</i> が唯一のパラメータです。<i>string</i> は、現在の DMCONFIG ファイルのバージョン番号を入力できるフィールドです。このフィールドは、ソフトウェアのチェックを受けません。</p>
DM_ROUTING	<p>同じサービスを提供する複数のリモート・ドメインの 1 つにローカルのサービス要求をルーティングするためのデータ依存型ルーティング基準を指定します。詳細については、1-30 ページの「Domains データ依存型ルーティングの指定」を参照してください。</p>

表 1-1DMCONFIG ファイルのセクション (シート 5 / 6)

セクション	目的
DM_ACCESS_CONTROL	<p>1 つ以上のアクセス制御リスト (ACL) の名前を指定し、1 つ以上のリモート・ドメイン・アクセス・ポイントを指定された各 ACL 名に関連付けます。ACL=ACL_NAME を設定して DM_EXPORT セクションで ACL パラメータを使用すると、特定のローカル・ドメイン・アクセス・ポイントを通じてエクスポートされるローカル・サービスへのアクセスを ACL_NAME と関連付けられたリモート・ドメイン・アクセス・ポイントのみに制限できます。</p> <p>次に、ACL エントリの例を示します。</p> <pre>*DM_ACCESS_CONTROL branch ACLIST=REMOT1</pre>

表 1-1DMCONFIG ファイルのセクション (シート 6 / 6)

セクション	目的
DM_domtype	<p>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。現時点で、<i>domtype</i> の値としては、TDOMAIN、TOPEND、OSITP、OSITPX、または SNACRM + SNALINKS + SNASTACKS を指定できます。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。</p> <p>DM_TDOMAIN セクションでは、ローカルまたはリモート・ドメイン・アクセス・ポイントの TDomain 固有のネットワーク・コンフィギュレーションを定義します。1 つ以上の WebLogic Server アプリケーションと関連付けられた 1 つ以上のリモート・ドメイン・アクセス・ポイントのネットワーク・コンフィギュレーションを定義して、アプリケーションで Tuxedo ATMI サーバと WebLogic Server Enterprise JavaBean (EJB) サーバを結合することもできます。詳細については、『BEA Tuxedo の相互運用性』を参照してください。</p> <p>DM_TDOMAIN セクションでは、リモート・ドメインからローカル・サービスへの要求がローカル・ドメイン・アクセス・ポイントを通じて受け付けられる場合に、そのローカル・ドメイン・アクセス・ポイントごとにエントリが必要です。このセクションで指定された各ローカル・ドメイン・アクセス・ポイントについて、受信時接続のリスンに使用するネットワーク・アドレスを指定する必要があります。</p> <p>DM_TDOMAIN セクションでは、ローカル・ドメインからリモート・サービスへの要求がリモート・ドメイン・アクセス・ポイントを通じて受け付けられる場合に、そのリモート・ドメイン・アクセス・ポイントごとにエントリが必要です。このセクションで指定された各リモート・ドメイン・アクセス・ポイントについて、そのリモート・ドメイン・アクセス・ポイントへの接続時に使用する接続先ネットワーク・アドレスを指定する必要があります。</p> <p>Domains のリンク・レベルのフェイルオーバーが使用されている場合は、リモート・ドメイン・アクセス・ポイントに複数の接続先ネットワーク・アドレスを指定してゲートウェイのミラーリングをインプリメントできます。ゲートウェイのミラーリングの例については、1-43 ページの「Domains リンク・レベルのフェイルオーバーの設定」を参照してください。</p> <p>DM_TOPEND セクションについては、『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ DMCONFIG(5) で GWTOPEND(5) を参照してください。DM_OSITP、DM_OSITPX、DM_SNACRM、DM_SNALINKS、および DM_SNASTACKS の各セクションについては、『BEA eLink Documentation』を参照してください。</p>

DMCONFIG ファイルの詳細については、『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ [DMCONFIG\(5\)](#) および [DM_MIB\(5\)](#) を参照してください。

DMCONFIG ファイル関連の新しい用語

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。新しい用語は、[DMCONFIG\(5\)](#) のリファレンス・ページ、セクション名、パラメータ名、エラー・メッセージ、および [DM_MIB\(5\)](#) のリファレンス・ページ、クラス、およびエラー・メッセージで使用されています。

旧バージョンとの互換性のため、BEA Tuxedo 7.1 より以前に使用された 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 のリリース 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。次の表では、DMCNFIG ファイルの以前の用語と新しい用語を対比させています。

以前の用語		新しい用語	
セクション名	パラメータ名	セクション名	パラメータ名
DM_LOCAL_DOMAINS		DM_LOCAL	
DM_REMOTE_DOMAINS		DM_REMOTE	
	DOMAINID		ACCESSPOINTID
	MAXRDOM		MAXACCESSPOINT
	MAXRDTRAN		MAXRAPTRAN
DM_LOCAL_SERVICES		DM_EXPORT	
DM_REMOTE_SERVICES		DM_IMPORT	
	LDOM		LACCESSPOINT
	RDOM		RACCESSPOINT

BEA Tuxedo のリリース 7.1 以降の `dmunloadcf` コマンドでは、デフォルトで新しいドメイン関連の用語を使用する `DMCONFIG` ファイルが生成されます。以前のドメイン関連の用語を使用する `DMCONFIG` ファイルを出力するには、`-c` オプションを使用します。次に例を示します。

```
prompt> dmunloadcf -c > dmconfig_prev
```

Domains データ依存型ルーティングの指定

次のどのバッファ・タイプでも、`DMCONFIG` ファイルの `DM_ROUTING` セクションで、ローカルのサービス要求をリモート・ドメインにルーティングするためのデータ依存型ルーティング基準を指定できます。

- FML
- FML32
- VIEW
- VIEW32
- X_C_TYPE
- X_COMMON
- XML

次の例では、リモート・サービスの `TOUPPER` は `REMOT1` および `REMOT2` という 2 つのリモート・ドメイン・アクセス・ポイントを通じて利用でき、`TOUPPER` のデータ依存型ルーティング基準は `ACCOUNT` というルーティング基準テーブルで定義されています。例の `RTOUPPER1` と `RTOUPPER2` は、リモート・ドメインで予期される実際のサービス名 `TOUPPER` のエイリアスです。

```
*DM_IMPORT
RTOUPPER1  AUTOTRAN=N
           RACCESSPOINT=REMOT1
```

```

LACCESSPOINT=LOCAL1
CONV=N
RNAME="TOUPPER"
ROUTING=ACCOUNT
RTOUPPER2  AUTOTRAN=N
RACCESSPOINT=REMOT2
LACCESSPOINT=LOCAL1
CONV=N
RNAME="TOUPPER"
ROUTING=ACCOUNT

*DM_ROUTING
ACCOUNT    FIELD=branchid
           BUFTYPE="VIEW:account"
           RANGES="MIN-1000:REMOT1,1001-3000:REMOT2"

```

ACCOUNT ルーティング・テーブルに関して、VIEW と account はこのルーティング・テーブルが有効であるデータ・バッファのタイプとサブタイプで、branchid はルーティングが適用される VIEW データ・バッファのフィールドの名前です。branchid フィールドの有効な値は次のとおりです。

- REMOT1 アクセス・ポイントの場合、有効な値の範囲は REMOT1 と関連付けられたマシンで許可される最小値から 1000 までです。
- REMOT2 アクセス・ポイントの場合、有効な値の範囲は 1001 から 3000 までです。

TOUPPER サービス要求の branchid フィールドの値が MIN-1000 の範囲にある場合、サービス要求は REMOT1 アクセス・ポイントを通じてルーティングされます。TOUPPER サービス要求の branchid フィールドの値が 1001-3000 の範囲にある場合、サービス要求は REMOT2 アクセス・ポイントを通じてルーティングされます。

Domains のトランザクション・タイムアウトとブロッキング・タイムアウトの指定

BEA Tuxedo システムには、トランザクション・タイムアウト・メカニズムとブロッキング・タイムアウト・メカニズムの2つのタイムアウト・メカニズムがあります。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 ATMI 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、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、サービス要求に対する応答を ATMI 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。

プロセスがトランザクション・モードではない場合、BEA Tuxedo システムによってブロッキング・タイムアウトが実行されます。プロセスがトランザクション・モードである場合は、BEA Tuxedo システムによってトランザクション・タイムアウトが実行されますが、ブロッキング・タイムアウトは実行されません。後者の説明はドメイン内トランザクション (1 つの BEA Tuxedo ドメイン内で処理されるトランザクション) では当てはまりますが、ドメイン間トランザクションでは当てはまりません。ドメイン間トランザクションでは、プロセスがトランザクション・モードである場合、Domains ソフトウェアによってトランザクション・タイムアウトとブロッキング・タイムアウトの両方が実行されます。

Domains コンポーネントによるトランザクション・タイムアウトの処理

BEA Tuxedo のトランザクション・タイムアウト・メカニズムは、Domains コンポーネントにもそのまま適用されます。ドメイン・ゲートウェイはトランザクション・マネージャ・サーバ (TMS) 機能をインプリメントしているため、BEA Tuxedo BBL 管理プロセスで生成される TMS タイムアウト・

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。

DMCONFIG ファイルの `DM_EXPORT` セクションでリモート・ドメインから利用できるようにされたローカル・サービスは、ローカル `UBBCONFIG` ファイルの `SERVICES` セクションから次のトランザクション関連プロパティを継承します。

- `AUTOTRAN` `AUTOTRAN` がサービスに対して有効になっている状態で、まだトランザクションにないサービスに対するサービス要求が受信されると、ローカルの BEA Tuxedo システムによって自動的にそのサービスのトランザクションが開始されます。
- `TRANTIME` サービスに対して自動的に開始されたトランザクションのトランザクション・タイムアウト値 (秒単位)。トランザクションでこのタイムアウト値を過ぎると、そのトランザクションの影響を受ける BEA Tuxedo ノード (マシン) によって TMS タイムアウト・メッセージが生成されます。

同様に、DMCONFIG ファイルの `DM_IMPORT` セクションでローカル・ドメインから利用できるようにされたリモート BEA Tuxedo サービスは、リモートの `UBBCONFIG` ファイルの `SERVICES` セクションから `AUTOTRAN` プロパティと `TRANTIME` プロパティを継承します。トランザクションで `TRANTIME` タイムアウト値を過ぎると、そのトランザクションの影響を受ける BEA Tuxedo ノードによって TMS タイムアウト・メッセージが生成されます。

BEA Tuxedo のリリース 8.1 以降が動作するマシンで宣言されたサービスは、`UBBCONFIG` ファイルの `RESOURCES` セクションから `MAXTRANTIME` という追加的なトランザクション・タイムアウト・プロパティを継承します。`MAXTRANTIME` タイムアウト値が `TRANTIME` タイムアウト値またはトランザクションを開始する `tpbegin(3c)` の呼び出しで渡されたタイムアウト値より小さい場合、トランザクションのタイムアウトは `MAXTRANTIME` の値に削減されます。`MAXTRANTIME` は、BEA Tuxedo の 8.0 以前が動作しているマシンで開始されたトランザクションには関係ありません。ただし、BEA Tuxedo の 8.1 以降が動作しているマシンがそのトランザクションの影響を受ける場合、トランザクションのタイムアウト値はそのノードに設定された `MAXTRANTIME` 値に制限 (必要に応じ削減) されます。

Domains コンフィギュレーションでは、次のトランザクション処理シナリオが考えられます。

- ドメイン間トランザクションが `MAXTRANTIME` パラメータを理解しないノードに影響する場合、またはノードは `MAXTRANTIME` パラメータを理解するが、パラメータが設定されていない場合、トランザクションのタイムアウト値は `TRANTIME` で決まるか、トランザクションを開始した `tpbegin()` の呼び出しで渡されたタイムアウト値で決まります。
`TRANTIME` または `tpbegin()` タイムアウト値を過ぎると、トランザクションの影響を受けるすべての BEA Tuxedo ノード (トランザクションを開始したノードを含む) で TMS タイムアウト・メッセージが生成されます。
- ドメイン間トランザクションが `MAXTRANTIME` パラメータを理解するノードに影響する場合で、そのパラメータがそのノードに対して設定されている場合、トランザクションのタイムアウト値はそのノードの `MAXTRANTIME` 値に削減されます。

`TRANTIME` または `tpbegin()` タイムアウト値が `MAXTRANTIME` 以下である場合、トランザクション処理は前に説明したシナリオになります。

`TRANTIME` または `tpbegin()` タイムアウト値が `MAXTRANTIME` より大きい場合、トランザクションのタイムアウト値は影響を受けるノードによって `MAXTRANTIME` に削減されます。`MAXTRANTIME` タイムアウト値を過ぎると、影響を受けるノードで TMS タイムアウト・メッセージが生成されます。

`MAXTRANTIME` の詳細については、`UBBCONFIG(5)` の `RESOURCES` セクションの `MAXTRANTIME`、または `TM_MIB(5)` の `T_DOMAIN` クラスの `TA_MAXTRANTIME` を参照してください。

Domains コンポーネントによるブロッキング・タイムアウトの処理

BEA Tuxedo のブロッキング・タイムアウト・メカニズムでは、ローカル・マシンで動作する各 BEA Tuxedo クライアント・プロセスまたはサーバ・プロセスに割り当てられたレジストリ・スロット内の情報を使用します。レジ

ストリ・スロットは、各プロセスに1つです。レジストリ・スロット内の情報は、BLOCKTIME で指定された期間を過ぎてもブロックされたままの要求元を検出するためにローカルの BBL で使用されます。ドメイン・ゲートウェイ・プロセスは一度に複数のサービス要求を処理できるマルチタスク型サーバなので（複数のレジストリ・スロットが必要）、ドメイン・ゲートウェイではレジストリ・スロット・メカニズムを使用できません。Domains 環境でブロッキング・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がクリーンアップされます。

DMCONFIG ファイルの DM_LOCAL セクションでは、BLOCKTIME パラメータを使用してローカル・ドメイン・アクセス・ポイントのブロッキング・タイムアウトを設定できます。次に例を示します。

```
*DM_LOCAL
LOCAL1  GWGRP=GWTGROUP
        TYPE=TDOMAIN
        ACCESSPOINTID="BA.CENTRAL01"
        BLOCKTIME=30
```

BLOCKTIME パラメータでは、ATMI ブロッキング呼び出しがブロックされる最大待ち時間を指定します。この時間を過ぎると、タイムアウトになります。ブロッキング・タイムアウト状態は、影響を受けたサービス要求が失敗していることを意味します。

ブロッキング・タイムアウト値は、UBBCONFIG ファイルの RESOURCES セクションで指定された SCANUNIT パラメータの乗数です。値 SCANUNIT * BLOCKTIME は、SCANUNIT 以上、32,767 秒以下である必要があります。

BLOCKTIME が DCONFIG ファイルで指定されていない場合、デフォルトは UBBCONFIG ファイルの RESOURCES セクションで指定された BLOCKTIME パラメータの値に設定されます。BLOCKTIME パラメータが UBBCONFIG ファイルで指定されていない場合、デフォルトは SCANUNIT * BLOCKTIME が約 60 秒になるように設定されます。

トランザクションの期間が BLOCKTIME を過ぎると、ドメイン間トランザクションでブロッキング・タイムアウト状態が生じます。つまり、ドメイン間トランザクションについては、BLOCKTIME 値が UBBCONFIG ファイルの SERVICES セクションで指定された TRANTIME タイムアウト値、またはトランザクションを開始する tpbegin() の呼び出しで渡されたタイムアウト値より小さい場合、トランザクションのタイムアウトは BLOCKTIME の値に削減され

ます。その一方で、1つの BEA Tuxedo ドメイン内で処理されるドメイン内トランザクションの場合、TUXCONFIG ファイルの RESOURCES セクションで指定された BLOCKTIME 値はドメイン内トランザクションのタイムアウトに影響しません。

Domains の接続方針の指定

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1つ以上のリモート・ドメインへの接続条件を指定できます。

- ON_DEMAND (デフォルト) リモート・サービスへのクライアント要求または管理「接続」コマンドのいずれかによって要求されたときに接続します。この接続方針における接続の確立方法は次のとおりです。
 - クライアント要求
 - 手動による `dmadmin(1) connect` コマンドの実行
 - 受信時接続
- ON_STARTUP ゲートウェイ・サーバの初期化(起動)時に接続します。この接続方針における接続の確立方法は次のとおりです。
 - BEA Tuxedo アプリケーション起動時の自動接続
 - 手動による `dmadmin(1) connect` コマンドの実行
 - 受信時接続
- INCOMING_ONLY 受信時に接続しますが、自動的には行いません。この接続方針における接続の確立方法は次のとおりです。
 - 手動による `dmadmin(1) connect` コマンドの実行
 - 受信時接続

接続方針は、TDomain および TOP END ドメイン・ゲートウェイのみに適用されます。

接続方針の設定方法

DMCONFIG ファイルの DM_LOCAL セクションでは、CONNECTION_POLICY パラメータを使用してローカル・ドメイン・アクセス・ポイントの接続方針を設定します。次に例を示します。

```
*DM_LOCAL
LOCAL1  GWGRP=GWTGROUP
        TYPE=TDOMAIN
        ACCESSPOINTID="BA.CENTRAL01"
        BLOCKTIME=30
        CONNECTION_POLICY=ON_STARTUP
```

ローカル・ドメイン・アクセス・ポイントの接続方針を指定しない場合、そのアクセス・ポイントの接続方針はデフォルトで ON_DEMAND になります。

BEA Tuxedo リリース 8.1 以降が動作している TDomain ゲートウェイについては、DMCONFIG ファイルの DM_TDOMAIN セクションでローカルまたはリモート・ドメイン単位で接続方針を設定できます。次に例を示します。

```
*DM_LOCAL
LOCAL1  GWGRP=GWTGROUP
        TYPE=TDOMAIN
        ACCESSPOINTID="BA.CENTRAL01"
        BLOCKTIME=30

*DM_TDOMAIN
LOCAL1  NWADDR="//albany.acme.com:4051"
        CONNECTION_POLICY=ON_STARTUP
REMOT1  NWADDR="//newyork.acme.com:65431"
        CONNECTION_POLICY=ON_DEMAND
REMOT2  NWADDR="//philly.acme.com:65431"
```

リモート・ドメイン・アクセス・ポイントに指定された接続方針は、ローカル・ドメイン・アクセス・ポイントに指定された接続方針に優先します。このため、前の例の接続方針コンフィギュレーションは次のようになります。

```
LOCAL1 to REMOT1 — ON_DEMAND
LOCAL1 to REMOT2 — ON_STARTUP
```

BEA Tuxedo 8.1 以降では、DMCONFIG ファイルの DM_TDOMAIN セクションでローカル・ドメイン・アクセス・ポイントに次のいずれかの接続方針値を指定できます。

- ON_DEMAND
- ON_STARTUP
- INCOMING_ONLY

ローカル・ドメイン・アクセス・ポイントの接続方針を指定しない場合は、DMCONFIG ファイルの DM_LOCAL セクションで指定されたグローバル接続方針がデフォルトで使用されます。DM_TDOMAIN セクションで指定されたグローバル接続方針は、DM_LOCAL セクションで指定されたグローバル接続方針に優先します。

注記 DM_TDOMAIN セクションでグローバル接続方針を指定する場合は、DM_LOCAL セクションでグローバル接続方針を指定しないでください。

BEA Tuxedo 8.1 以降では、リモート・ドメイン・アクセス・ポイントの接続方針値も、DMCONFIG ファイルの DM_TDOMAIN セクションで次のいずれかから指定できます。

- LOCAL (デフォルト)
- ON_DEMAND
- ON_STARTUP
- INCOMING_ONLY

リモート・ドメイン・アクセス・ポイントに LOCAL を指定するか、接続方針を指定しないと、デフォルトでグローバル接続方針が使用されます。

リモート・ドメインの接続方針機能がないと、グローバル接続方針が ON_STARTUP の場合に、ローカル TDomain ゲートウェイはたとえ一部のリモート・ドメインが最初は使用されない場合でも起動時にすべてのリモート・ドメインに接続しようとします。このため、リモート・ドメインの数が多き場合は起動にかなりの時間がかかります。リモート・ドメインの接続方針機能を使用する場合は、グローバル接続方針の ON_STARTUP で、起動時に自動的に確立されないリモート・ドメインの接続を選択できます。

接続再試行プロセスの使用方法

CONNECTION_POLICY が ON_STARTUP に設定されている場合は、ほかの 2 つのパラメータを設定して、ローカル・ドメイン・ゲートウェイがリモート・ドメインへの接続を試行する回数を指定できます。デフォルトでは、ローカル・ドメイン・ゲートウェイは 60 秒おきに失敗した接続を再試行しますが、パラメータ MAXRETRY および RETRY_INTERVAL を使用してこの間隔に別の値を指定することもできます。

MAXRETRY パラメータでは、ドメイン・ゲートウェイがリモート・ドメインへの接続を試行する回数を指定します。最小値は 0、最大値は 2147483647 です。デフォルト設定は 2147483647 です。このパラメータを 0 に設定すると、接続再試行プロセスが無効になります。

RETRY_INTERVAL パラメータでは、リモート・ドメインへの接続を確立する自動的な試みの秒間隔を指定します。最小値は 0、最大値は 2147483647 です。デフォルト設定は 60 です。MAXRETRY パラメータが 0 に設定されている場合、RETRY_INTERVAL は設定できません。

例 1

```
*DM_LOCAL
LOCAL1 GWGRP=GWTGROUP
      TYPE=TDOMAIN
      ACCESSPOINTID="BA.CENTRAL01"
      BLOCKTIME=30
      CONNECTION_POLICY=ON_STARTUP
      MAXRETRY=5
      RETRY_INTERVAL=100
```

例 2 (BEA Tuxedo リリース 8.1 以降が動作する TDomain ゲートウェイでのみ可能)

```
*DM_LOCAL
LOCAL1 GWGRP=GWTGROUP
      TYPE=TDOMAIN
      ACCESSPOINTID="BA.CENTRAL01"
      BLOCKTIME=30

*DM_TDOMAIN
LOCAL1 NWADDR="//albany.acme.com:4051"
      CONNECTION_POLICY=ON_STARTUP
      MAXRETRY=5
```

```
RETRY_INTERVAL=100
REMOT1  NWADDR="//newyork.acme.com:65431"
CONNECTION_POLICY=ON_STARTUP
MAXRETRY=10
RETRY_INTERVAL=40
```

2 番目の例で、MAXRETRY および RETRY_INTERVAL の値 10 と 40 は、REMOT1 というリモート・ドメイン・アクセス・ポイントへの接続を確立するためにローカル TDomain ゲートウェイで 사용되는自動接続再試行の基準になります。

接続方針によるリモート・サービスの可用性の判断

接続方針を指定すると、それによって、リモート・ドメインからインポートされたサービスがドメイン・ゲートウェイによって BEA Tuxedo 掲示板でどのように宣言されるかが決まります。

- ON_DEMAND の場合、ローカル・ドメイン・ゲートウェイはリモート・ドメインからインポートされたサービスを継続的に宣言します。
- ON_STARTUP の場合、ローカル・ドメイン・ゲートウェイはリモート・ドメインからインポートされたサービスを、そのリモート・ドメインとの接続が存在する限り宣言します。
- INCOMING_ONLY の場合、ローカル・ドメイン・ゲートウェイは受信時接続で、または `dmadmin connect` コマンドの発行時にリモート・ドメインからインポートされたサービスを宣言します。

接続方針が ON_STARTUP または INCOMING_ONLY の場合 (ON_DEMAND は除く)、TDomain および TOP END ドメイン・ゲートウェイの機能である Dynamic Status はリモート・サービスの状態を調べます。リモート・サービスの状態は、ローカル・ドメイン・ゲートウェイとリモート・ドメイン・ゲートウェイのネットワーク接続の状態によって決まります。リモート・サービスは、そのサービスのあるドメインへの接続が正常に確立されるたびにローカル・ドメインで宣言されて利用可能になります。リモート・サービスは、そのサービスのあるドメインとの接続が切れるたびに中断されて利用不能になります。

ドメイン・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、そのサービスはドメイン・ゲートウェイによって BEA Tuxedo 掲示板で中断されます。

たとえば、DMCONFIG ファイルの DM_IMPORT 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。

```
*DM_IMPORT
RSVC AUTOTRAN=N
      RACCESSPOINT=REMOT1
      LACCESSPOINT=LOCAL1
RSVC AUTOTRAN=N
      RACCESSPOINT=REMOT2
      LACCESSPOINT=LOCAL1
```

REMOT1 および REMOT2 への接続が可能な場合、ドメイン・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。REMOT1 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 REMOT2 に送信します。REMOT1 および REMOT2 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。

関連項目

- [2-63 ページの「Domains コンフィギュレーションの接続の設定」](#)
- [2-69 ページの「Domains コンフィギュレーションの接続の制御」](#)

Domains のフェイルオーバーとフェイルバックの指定

DMCONFIG ファイルの DM_IMPORT セクションでは、Domains コンフィギュレーションの Domains レベルのフェイルオーバーおよびフェイルバック機能を設定できます。DMCONFIG ファイルの DM_TDOMAIN セクションでは、Domains コンフィギュレーションの Domains リンク・レベルのフェイルオーバー機能を設定できます。

Domains レベルのフェイルオーバーとフェイルバックの設定

Domains レベルのフェイルオーバーは、一次リモート・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。一次リモート・ドメインが回復したときには、そのドメインへのフェイルバックも行われます。

Domains レベルのフェイルオーバーとフェイルバックをサポートするには、特定のサービスを実行できるリモート・ドメイン・アクセス・ポイントのリストを指定します。次に例を示します。

```
*DM_IMPORT  
TOUPPER RACCESSPOINT="REMOT1,REMOT2,REMOT3"
```

この例で、TOUPPER サービスは REMOT1 (一次)、REMOT2、および REMOT3 の 3 つのリモート・ドメイン・アクセス・ポイントのいずれかで実行できます。REMOT1 が利用できなくなると、REMOT2 がフェイルオーバーに使用されます。REMOT1 と REMOT2 が両方とも使用できない場合は、REMOT3 がフェイルオーバーに使用されます。

サービスの代替リモート・ドメインを設定する必要がある場合は、ON_STARTUP または INCOMING_ONLY を CONNECTION_POLICY パラメータの値として指定します。接続方針として ON_DEMAND を指定すると、RACCESSPOINT パラメータで指定された代替リモート・ドメインに「フェイルオーバー」できません。

Domains レベルのフェイルバックは、一次リモート・ドメインへのネットワーク接続が次のいずれかの理由で再確立されたときに行われます。

- 自動接続再試行 (ON_STARTUP のみ)
- 受信時接続
- 手動による `dmadmin connect` コマンドの実行

Domains リンク・レベルのフェイルオーバーの設定

Domains のリンク・レベルのフェイルオーバーは、一次ネットワーク・リンクが失敗したときに二次ネットワーク・リンクが有効になるようにするメカニズムです。ただし、一次リンクが回復してもそのリンクへのフェイルバックは行われません。つまり、一次リンクが回復したときには、二次リンクを手動で無効にしてトラフィックを強制的に一次リンクに戻す必要がありません。

Domains のリンク・レベルのフェイルオーバーを設定するには、DMCONFIG ファイルの DM_TDOMAIN セクションでリモート・ドメイン・アクセス・ポイントの複数のエントリを指定します。次に例を示します。

```
*DM_TDOMAIN
REMOT1  NWADDR="//newyork.acme.com:65431"
REMOT1  NWADDR="//trenton.acme.com:65431"
```

最初のエントリは一次アドレスと見なされます。つまり、その NWADDR はリモート・ドメイン・アクセス・ポイントへの接続が試行されるときに試される最初のネットワーク・アドレスです。2 番目のエントリは二次アドレスと見なされます。つまり、その NWADDR は一次アドレスを使用して接続を確立できないときに試される 2 番目のネットワーク・アドレスです。

2 番目のエントリは、一次リモート・ゲートウェイのある BEA Tuxedo ドメインとは別の BEA Tuxedo ドメインにある二次リモート・ゲートウェイを指します。二次と一次のリモート・ゲートウェイは、その関連する `DMCONFIG` ファイルの `DM_LOCAL` セクションで同じ `ACCESSPOINTID` が定義されている必要があります。この処置は、ゲートウェイのミラーリングと呼ばれることがあります。この機能をトランザクションまたは会話で使用することはお勧めしません。また、一次リモート・ゲートウェイが使用可能なときに使用することもお勧めしません。

Domains のキープアライブの指定

Domains のキープアライブ (BEA Tuxedo リリース 8.1 以降が動作する TDomain ゲートウェイで利用可能) を利用すると、TDomain ゲートウェイの接続ごとに TCP レベルおよびアプリケーション・レベルでキープアライブ・プロトコルを有効化および設定できます。TCP レベルのキープアライブとアプリケーション・レベルのキープアライブは相互に排他的ではないので、両方のオプションを使用して Domains 接続を設定できます。

次の表は、Domains のキープアライブに関する主要な情報を提供します。

表 1-2 Domains のキープアライブについて

レベル	旧リリースの Tuxedo と相互運用できるか	個別タイマーがあるか	接続の失敗を迅速に検出できるか	ファイアウォールでのキープアライブ・イベントがあるか
TCP レベルのキープアライブ	はい	いいえ	はい*	はい
アプリケーション・レベルのキープアライブ	いいえ	はい	はい	はい

* TCP レベルのキープアライブで TDomain ゲートウェイ接続の失敗を迅速に検出するためには、時間間隔を短く設定する必要があります。ただし、その結果としてネットワークが TCP パケットで混雑する場合があります。

ほとんどの BEA Tuxedo Domains コンフィギュレーションはファイアウォールをまたがっており、ファイアウォールはアイドル接続がタイムアウトになる原因となります。Domains キープアライブは活動のない期間に BEA Tuxedo のドメイン間接続をオープンに維持するだけでなく、TDomain ゲートウェイで Domains 接続の失敗を迅速に検出できるようにします。現在、TDomain ゲートウェイは基盤の TCP スタックを通じて Domains 接続の失敗を検出しています。TCP スタックは、失敗の発生後 15 分以上経ってからその失敗を報告します。実際に何分かかかるかは、ローカルのオペレーティング・システムのコンフィギュレーションによって異なります。

TCP レベルのキープアライブ

キープアライブ機能は TCP の仕様に含まれていませんが、ほとんどのオペレーティング・システムでは TCP キープアライブ・タイマーが提供されます。TCP キープアライブ・タイマーを使用すると、TCP 接続の片側のサーバ・マシンでその接続のもう片側のクライアント・マシンがアクセス可能かどうかを検出できます。

TCP 接続を経由してサーバ・マシンに届くメッセージはすべて、TCP キープアライブ・タイマーをリセットします。キープアライブ・タイマーで事前定義済みの期間（通常は 2 時間）TCP 接続で活動がないと検出されると、タイマーが切れて、サーバ・マシンからセグメント検査パケットがクライアント・マシンに送信されます。接続がまだオープンであり、クライアント・マシンがまだ機能している場合は、クライアント・マシンから肯定応答がサーバ・マシンに送信されます。セグメント検査パケットを送信してから一定の期間肯定応答が届かない場合、サーバ・マシンは接続が切れていると判断して、その接続と関連付けられたリソースをすべて解放します。

接続がオープンでクライアント・マシンが機能しているかどうかを判断するだけでなく、TCP レベルのキープアライブはファイアウォールをまたがってアイドル接続をオープンに維持することもできます。接続で活動がない状態が定義済みの期間続いた後にセグメント検査パケットを自動的に送信することで、ファイアウォールのアイドル接続タイマーがタイムアウトになる前にリセットされ、それによって接続のオープンが維持されます。

オペレーティング・システムの TCP キープアライブ・タイマーの間隔は、通常は 2 時間に設定されます。この間隔は変更できますが、その変更によってマシンのすべての TCP 接続が影響を受けます。オペレーティング・システムの TCP キープアライブ間隔は、システム全体にかかわる値です。

Domains の TCP レベルのキープアライブの設定

Domains の BEA Tuxedo TCP レベル・キープアライブ・オプションは TCPKEEPALIVE という名前で、DMCONFIG ファイルの DM_TDOMAIN セクションにオプション・パラメータとして追加されています。このパラメータを使用すると、Domains の TCP レベルのキープアライブ・オプションをローカル・ドメインまたはリモート・ドメイン単位で有効にできます。

TCPKEEPALIVE の有効な値は次のとおりです。

- LOCAL (リモート・ドメイン・アクセス・ポイントでのみ有効)
- NO (キープアライブ無効)
- YES (キープアライブ有効)

Domains の TCP レベルのキープアライブ・オプションは、デフォルトでは無効です。Domains 接続で TCP レベルのキープアライブを有効にした場合、オペレーティング・システムの TCP キープアライブ・タイマーに設定されたシステム全体にかかわる値が、その接続のキープアライブ間隔として使用されます。

TCPKEEPALIVE の使い方を明確にするために、次の Domains TCP レベル・キープアライブ・コンフィギュレーションを考えてください。

```
*DM_TDOMAIN
LOCAL1  NWADDR="//albany.acme.com:4051"
        TCPKEEPALIVE=Y
REMOT1  NWADDR="//newyork.acme.com:65431"
REMOT2  NWADDR="//philly.acme.com:65431"
        TCPKEEPALIVE=NO
```

リモート・ドメイン・アクセス・ポイントに指定された TCP レベルのキープアライブ・コンフィギュレーションは、ローカル・ドメイン・アクセス・ポイントに指定された TCP レベルのキープアライブ・コンフィギュレーションに優先します。このため、前の例の TCP レベルのキープアライブ・コンフィギュレーションは次のようになります。

```
LOCAL1 から REMOT1  TCP レベルのキープアライブ有効
LOCAL1 から REMOT2  TCP レベルのキープアライブ無効
```

ローカル・ドメイン・アクセス・ポイントの場合、TCPKEEPALIVE パラメータに次のいずれかの値を指定できます。

- NO (デフォルト)
- YES

リモート・ドメイン・アクセス・ポイントの場合、TCPKEEPALIVE パラメータに次のいずれかの値を指定できます。

- LOCAL (デフォルト)
- NO
- YES

リモート・ドメイン・アクセス・ポイントで LOCAL を指定するか、コンフィギュレーションを指定しないと、デフォルトでローカルの TCP レベルのキープアライブ・コンフィギュレーションが使用されます。

注記 相互作用する 2 つの BEA Tuxedo ドメインのそれぞれで TCP レベルのキープアライブを有効にできます。ただし、両方のドメインで BEA Tuxedo 8.1 以降が動作している必要があります。

Domains 接続の接続方針が `ON_STARTUP` で、TCP 接続が TCP レベルのキープアライブの障害によりクローズしている場合は、自動接続再試行が行われます。接続の再試行が成功しない場合は、`dmadmin connect` コマンドを使用して接続を再確立する必要があります。`dmadmin connect` コマンドについては、[2-69 ページの「ドメイン間の接続の確立」](#)を参照してください。

アプリケーション・レベルのキープアライブ

オペレーティング・システムの TCP キープアライブについては、セグメント検査パケットが不必要に帯域幅を消費し、インターネット接続でユーザがパケット単位で支払うお金を浪費するという点で、一部の人は使用することに異を唱えています。また、アプリケーション層では次のことを判断すべきなので、キープアライブはトランスポート (TCP) 層ではなくアプリケーション層またはリンク層が適していると信じている人もいます。

- アプリケーションがメッセージの受信をかなり長い時間待っているかどうか
- TCP 接続がまだオープンであるか、および接続の反対側のマシンとアプリケーションがまだ機能しているかを判断するためにどのアクションを行うのか

誰がどのような意見を持っているかに関係なく、アプリケーション・レベルのキープアライブはキープアライブ・タイマーの間隔を接続ごとに設定できるという点では TCP レベルのキープアライブよりも優れています。TCP レベルのキープアライブの場合、タイマーの間隔はマシン単位で設定します。

アプリケーション・レベルのキープアライブを使用した場合、サーバ・アプリケーションはアプリケーション・キープアライブ・タイマーがタイムアウトになるたびにアプリケーション固有のキープアライブ・メッセージを送信します。キープアライブ・メッセージは通常はヘッダ情報のみで構成され、データは関連付けられていません。クライアント・アプリケーションは、肯定応答を送信してサーバ・アプリケーションに応答します。キープアライ

ブ・メッセージを送信してから事前定義済みの期間肯定応答が届かない場合、サーバ・アプリケーションは接続が切れていると判断して、その接続と関連付けられたリソースをすべて解放します。

接続がオープンでクライアント・アプリケーションが機能しているかどうかを判断するだけでなく、アプリケーション・レベルのキープアライブはファイアウォールをまたがってアイドル接続をオープンに維持することもできます。接続で活動がないことが定義済みの期間続いた後にキープアライブ・メッセージを自動的に送信することで、ファイアウォールのアイドル接続タイマーがタイムアウトになる前にリセットされ、それによって接続のオープンが維持されます。

Domains のアプリケーション・レベルのキープアライブの設定

Domains の BEA Tuxedo アプリケーション・レベル・キープアライブ・オプションは `KEEPALIVE` 名前です。このパラメータとそれに伴う `KEEPALIVEWAIT` というパラメータが、`DMCONFIG` ファイルの `DM_TDOMAIN` セクションのオプション・パラメータとして追加されています。これらのパラメータを使用すると、Domains のアプリケーション・レベルのキープアライブ・オプションをローカル・ドメインまたはリモート・ドメイン単位で設定できます。

`DMKEEPALIVE` パラメータでは、Domains 接続でトラフィックを受信することなくローカル TDomain ゲートウェイが待機する最長時間を指定します。この時間を過ぎると、ゲートウェイからアプリケーション・レベル・キープアライブ要求メッセージが送信されます。`DMKEEPALIVE` の有効な値は次のとおりです。

- -1 (リモート・ドメイン・アクセス・ポイントでのみ有効)。
- 0 (キープアライブ無効)。
- $1 \leq \text{値} \leq 2147483647$ (キープアライブ有効)。値はミリ秒単位ですが、現在は Domains ソフトウェアによって最近の秒に丸められます。

`DMKEEPALIVE` のデフォルト設定は 0 です。

DMKEEPALIVEWAIT パラメータでは、送信したキープアライブ・メッセージに対する肯定応答を受信することなくローカル TDomain ゲートウェイが待機する最長時間を指定します。この時間を過ぎると、ゲートウェイはリモート TDomain ゲートウェイとの接続が切れているものと判断し、その接続に関連付けられたすべてのリソースを解放します。DMKEEPALIVEWAIT の最小値は 0、最大値は 2147483647 ミリ秒で、現在は Domains ソフトウェアによって最近の秒に丸められます。DMKEEPALIVEWAIT のデフォルト設定は 0 です。

- DMKEEPALIVE が 0 (キープアライブ無効) の場合は、DMKEEPALIVEWAIT を設定しても効果はありません。
- DMKEEPALIVE が有効で、DMKEEPALIVEWAIT が DMKEEPALIVE より大きい値に設定されている場合、ローカル TDomain ゲートウェイは DMKEEPALIVEWAIT タイマーが切れるまでに複数のアプリケーション・レベル・キープアライブ・メッセージを送信します。この組み合わせの設定は可能です。
- DMKEEPALIVE が有効で、DMKEEPALIVEWAIT が 0 に設定されている場合、送信したキープアライブ・メッセージに対する肯定応答を受信することは重要ではありません。そのような肯定応答はすべて、ローカル TDomain ゲートウェイで無視されます。ローカル TDomain ゲートウェイは、DMKEEPALIVE タイマーがタイムアウトになるたびにキープアライブ・メッセージを送信し続けます。この組み合わせの設定を使用すると、ファイアウォールをまたがってアイドル接続をオープンに維持できます。

DMKEEPALIVE と DMKEEPALIVEWAIT の使い方を明確にするために、次の Domains アプリケーション・レベル・キープアライブ・コンフィギュレーションを考えてください。

```
*DM_TDOMAIN
LOCAL1  NWADDR="//albany.acme.com:4051"
        DMKEEPALIVE=1010
        DMKEEPALIVEWAIT=20
REMOT1  NWADDR="//newyork.acme.com:65431"
        DMKEEPALIVE=4000
        DMKEEPALIVEWAIT=3000
REMOT2  NWADDR="//philly.acme.com:65431"
        DMKEEPALIVE=-1
```

リモート・ドメイン・アクセス・ポイントに指定されたキープアライブ・コンフィギュレーションは、ローカル・ドメイン・アクセス・ポイントに指定されたキープアライブ・コンフィギュレーションに優先します。このため、前の例のアプリケーション・レベルのキープアライブ・コンフィギュレーションは次のようになります。

LOCAL1 から REMOT1 キープアライブ・タイマー = 4 秒、待機タイマー = 3 秒

LOCAL1 から REMOT2 キープアライブ・タイマー = 2 秒、待機タイマー = 1 秒

ローカル・ドメイン・アクセス・ポイントの場合、DMKEEPALIVE パラメータに次のいずれかの値を指定できます。

- 0 (デフォルト)
- $1 \leq \text{値} \leq 2147483647$ 。値はミリ秒単位ですが、現在は Domains ソフトウェアによって最近の秒に丸められます。

リモート・ドメイン・アクセス・ポイントの場合、DMKEEPALIVE パラメータに次のいずれかの値を指定できます。

- -1 (デフォルト)
- 0
- $1 \leq \text{値} \leq 2147483647$ 。値はミリ秒単位ですが、現在は Domains ソフトウェアによって最近の秒に丸められます。

リモート・ドメイン・アクセス・ポイントで -1 を指定するか、キープアライブ・コンフィギュレーションを指定しないと、デフォルトでローカルのアプリケーション・レベルのキープアライブ・コンフィギュレーションが使用されます。

注記 相互運用される 2 つの BEA Tuxedo ドメインのそれぞれで、アプリケーション・レベルのキープアライブを設定できます。待機間隔は同じでも別々でもかまいません。ただし、両方のドメインで BEA Tuxedo 8.1 以降が動作している必要があります。

Domains 接続の接続方針が ON_STARTUP であり、その接続でアプリケーション・レベルのキープアライブの障害が発生した場合は、自動的な接続再試行プロセスによって接続の再確立が行われます。接続再試行プロセスの詳細については、[1-39 ページの「接続再試行プロセスの使用法」](#)を参照してください。

旧リリースの BEA Tuxedo とのキープアライブの互換性

Domains の TCP レベルのキープアライブは、BEA Tuxedo 8.0 以前と互換性があります。Domains の TCP レベルのキープアライブはネットワーク・トランスポート (TCP) 層で実行されるので、TCP 接続の反対側の BEA Tuxedo ソフトウェアはどのリリースの BEA Tuxedo でもかまいません。

Domains のアプリケーション・レベルのキープアライブは、BEA Tuxedo 8.0 以前と互換性がありません。TCP 接続の反対側の BEA Tuxedo ソフトウェアは、BEA Tuxedo 8.1 以降でないとうアプリケーション・レベル・キープアライブ・メッセージを理解できません。旧リリースの BEA Tuxedo ソフトウェアが動作している TDomain ゲートウェイに接続した場合、TDomain ゲートウェイはアプリケーション・レベルのキープアライブ・メッセージを送信しません。代わりに、リモート・ドメインで旧リリースの BEA Tuxedo ソフトウェアが動作しており、Domains のアプリケーション・レベルのキープアライブがサポートされていないことを示す警告メッセージがローカルのユーザ・ログ (ULOG) に記録されます。

Domains 環境の設定

次のリストは、TDomain ゲートウェイ・タイプ用に Domains 環境を設定するためのタスクをまとめたものです。

1. テキスト・エディタで `UBBCONFIG` ファイルを編集し、Domains 管理サーバと TDomain ゲートウェイ・サーバを設定します。次に例を示します。

```
*GROUPS
DMADMGRP  LMID=SITE1  GRPNO=1
GWTGROUP  LMID=SITE2  GRPNO=2

*SERVERS
DMADM      SRVGRP=DMADMGRP
           SRVID=1001
           REPLYQ=N
```

```

                RESTART=Y
                GRACE=0
GWADM          SRVGRP=GWTGROUP
                SRVID=1002
                REPLYQ=N
                RESTART=Y
                GRACE=0
GWTDOMAIN     SRVGRP=GWTGROUP
                SRVID=1003
                RQADDR="GWTGROUP"
                REPLYQ=N
                RESTART=Y
                GRACE=0

```

注記 この例では、DMADM、GWADM、および GWTDOMAIN の各サーバで REPLYQ=N が指定されています。この設定は必須ではありません。必要に応じて REPLYQ=Y を指定して、それらのどのサーバにでも応答キューを指定できます。ただし、REPLYQ が N に設定されている場合は、パフォーマンスが向上することがあります。

TDomain ゲートウェイ・サーバとその関連する GWADM サーバは、BEA Tuxedo ドメインの同じマシンで実行する必要があります。DMADM サーバは、BEA Tuxedo ドメインのどのマシン（マスタ・マシン、非マスタ・マシン）でも実行できます。

2. `tmloadcf(1)` を実行して BEA Tuxedo コンフィギュレーションをロードします。`tmloadcf` コマンドを実行すると、`UBBCONFIG` が解析され、`TUXCONFIG` 変数が指す場所にバイナリ形式の `TUXCONFIG` ファイルがロードされます。
3. テキスト・エディタで `DMCONFIG` ファイルを編集し、TDomain ゲートウェイ・サーバ用に Domains 環境を設定します。次に例を示します。

```

*DM_LOCAL
LOCAL1        GWGRP=GWTGROUP
              TYPE=TDOMAIN
              ACCESSPOINTID="BA.CENTRAL01"
              BLOCKTIME=30
              CONNECTION_POLICY=ON_STARTUP
              MAXRETRY=5
              RETRY_INTERVAL=100

*DM_REMOTE
REMOT1        TYPE=TDOMAIN
              ACCESSPOINTID="BA.BANK01"

```

```
REMOT2      TYPE=TDOMAIN
            ACCESSPOINTID= "BA.BANK02"

*DM_EXPORT
LTOLOWER    LACCESSPOINT=LOCAL1
            CONV=N
            RNAME= "TOLOWER"

*DM_IMPORT
RTOUPPER    AUTOTRAN=N
            RACCESSPOINT=REMOT1
            LACCESSPOINT=LOCAL1
            CONV=N
            RNAME= "TOUPPER"

*DM_TDOMAIN
LOCAL1      NWADDR="//albany.acme.com:4051"
REMOT1      NWADDR="//newyork.acme.com:65431"
REMOT2      NWADDR="//philly.acme.com:65431"
```

DMCONFIG ファイルは、DMADM サーバと同じマシン上に配置する必要があります。

4. `tmloadcf(1)` を実行して Domains コンフィギュレーションをロードします。`dmloadcf` コマンドを実行すると、DMCONFIG が解析され、BDMCONFIG 変数が指す場所にバイナリ形式の BDMCONFIG ファイルがロードされます。
5. `tmboot(1)` を実行して BEA Tuxedo アプリケーション・サーバを起動します。`tmboot` コマンドは、すべての管理プロセスと環境変数 `TUXCONFIG` と `TUXOFFSET` に指定されている `TUXCONFIG` ファイルの `SERVERS` セクションにリストされているすべてのサーバを実行します。サーバは、`SERVERS` セクションにリストされている順序で起動します (`DMADM`、`GWADM`、そして `GWTDOMAIN` の順)。Domains サーバは、この順序で起動する必要があります。また、Domains サーバはアプリケーション・サーバの前に起動する必要があります。

Domains ATMI 環境を設定する詳しい例については、[2-1 ページの「ATMI 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。Domains CORBA 環境を設定する詳しい例については、[3-1 ページの「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。

移行を考慮した Domains 環境の設定

次に示すサンプルの `UBBCONFIG` ファイルと `DMCONFIG` ファイルを参照すると、Domains の移行を考慮して BEA Tuxedo アプリケーションを設定する方法がわかります。Domains の移行に特に重要なエントリは太字で示されています。

コード リスト 1-1 Domains の移行を考えて設定されたサンプル `UBBCONFIG` ファイル

```
*RESOURCES
IPCKEY      76666
MASTER     SITE1,SITE2
OPTIONS     LAN,MIGRATE
MODEL       MP
#
*MACHINES
mach1       LMID=SITE1
            TUXDIR="/home/rsmith/tuxroot"
            APPDIR="/home/rsmith/bankapp"
            TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
mach2       LMID=SITE2
            TUXDIR="/home/rsmith/tuxroot"
            APPDIR="/home/rsmith/bankapp"
            TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
mach3       LMID=SITE3
            TUXDIR="/home/rsmith/tuxroot"
            APPDIR="/home/rsmith/bankapp"
            TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
#
*GROUPS
DMADMGRP    LMID="SITE1,SITE3" GRPNO=1
GWTGROUP    LMID="SITE2,SITE3" GRPNO=2
.
.
.
*NETWORK
SITE1       NADDR="//albany.acme.com:4065"
            NLSADDR="//albany.acme.com:4068"
SITE2       NADDR="//auburn.acme.com:4065"
            NLSADDR="//auburn.acme.com:4068"
```

1 Domains について

```
SITE3      NADDR="//boston.acme.com:4065"
           NLSADDR="//boston.acme.com:4068"

#
*SERVERS
DMADM      SRVGRP=DMADMGRP
           SRVID=1001
           REPLYQ=N
           RESTART=Y
           GRACE=0
GWADM      SRVGRP=GWTGROUP
           SRVID=1002
           REPLYQ=N
           RESTART=Y
           GRACE=0
GWTDOMAIN SRVGRP=GWTGROUP
           SRVID=1003
           RQADDR="GWTGROUP"
           REPLYQ=N
           RESTART=Y
           GRACE=0
.
.
.
```

注記 この例では、DMADM、GWADM、および GWTDOMAIN の各サーバで REPLYQ=N が指定されています。この設定は必須ではありません。必要に応じて REPLYQ=Y を指定して、それらのどのサーバにでも応答キューを指定できます。ただし、REPLYQ が N に設定されている場合は、パフォーマンスが向上することがあります。

コード リスト 1-2 Domains の移行を考えて設定されたサンプル DMCONFIG ファイル

```
*DM_LOCAL
LOCAL1    GWGRP=GWTGROUP
           TYPE=TDOMAIN
           ACCESSPOINTID="BA.CENTRAL01"
           BLOCKTIME=30
           CONNECTION_POLICY=ON_STARTUP
           MAXRETRY=5
           RETRY_INTERVAL=100
```

```
*DM_REMOTE
REMOT1    TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK01"
REMOT2    TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK02"

*DM_EXPORT
LTOLOWER  LACCESSPOINT=LOCAL1
          CONV=N
          RNAME="TOLOWER"

*DM_IMPORT
RTOUPPER  AUTOTRAN=N
          RACCESSPOINT=REMOT1
          LACCESSPOINT=LOCAL1
          CONV=N
          RNAME="TOUPPER"

*DM_TDOMAIN
LOCAL1    NWADDR="//albany.acme.com:4051"
LOCAL1    NWADDR="//boston.acme.com:4051"
REMOT1    NWADDR="//newyork.acme.com:65431"
REMOT2    NWADDR="//philly.acme.com:65431"
```

サンプルのコンフィギュレーション・ファイルで、DMADM サーバと TDomain ゲートウェイ・グループ・サーバは SITE3 マシンに移行するように設定されています。DMADM の移行については、次のタスクを終えた後に管理者が SITE3 マシンで DMADM サーバ・プロセスをアクティブにします。

- SITE3 マシンで BDMCONFIG 環境変数を設定します。
- `dmloadcf(1)` コマンドを実行して、BDMCONFIG ファイルを SITE3 マシンにロードします。

TDomain ゲートウェイ・グループの移行については、管理者が SITE3 マシンで GWADM サーバ・プロセスと GWTDOMAIN サーバ・プロセスをアクティブにします。その時点で、LOCAL1 アクセス・ポイントと関連付けられたコンフィギュレーションと責任は、ネットワーク・アドレス

`boston.acme.com:4051` で接続要求をリスンする新しい GWTDOMAIN サーバ・プロセスによって処理されます。

注記 DMADM とドメイン・ゲートウェイ・グループは、同じマシンに移行する必要はありません。

DMADM サーバの移行

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. 新しいマシンで DMADM サーバ・プロセスをアクティブにします。詳細については、[1-59 ページの「個々のサーバ・プロセスをアクティブにする手段」](#)を参照してください。
3. 必要に応じて、BEA Tuxedo アプリケーションのすべてのドメイン・ゲートウェイ・グループを再起動します。詳細については、[1-59 ページの「個々のサーバ・プロセスをアクティブにする手段」](#)を参照してください。

ドメイン・ゲートウェイ・グループを再起動しない場合、それらは動作を続けますが、DMADM が移行された後はドメイン・ゲートウェイ・グループに対するすべての MIB 要求が失敗します。

TDomain ゲートウェイ・グループの移行

トランザクションが Domains コンフィギュレーションで使用されている場合、TDomain 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。

TDomain 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
LOCAL1 NWADDR="//primary:port"
LOCAL1 NWADDR="//backup:port"
```

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. 新しいマシンで `GWADM` および `GWTDOMAIN` サーバ・プロセスをアクティブにします。詳細については、次のセクションを参照してください。

個々のサーバ・プロセスをアクティブにする手段

個々の BEA Tuxedo サーバ・プロセスは、次のいずれかの手段でアクティブにできます。

- BEA Tuxedo Administration Console
- コマンド `tmboot(1)` (`-s` コマンド行オプションを指定)
- MIB (`TM_MIB(5)`) API

アプリケーションの移行タスクについては、『BEA Tuxedo アプリケーション実行時の管理』の「[アプリケーションの移行](#)」を参照してください。

2 ATMI Domains の計画とコンフィギュレーション

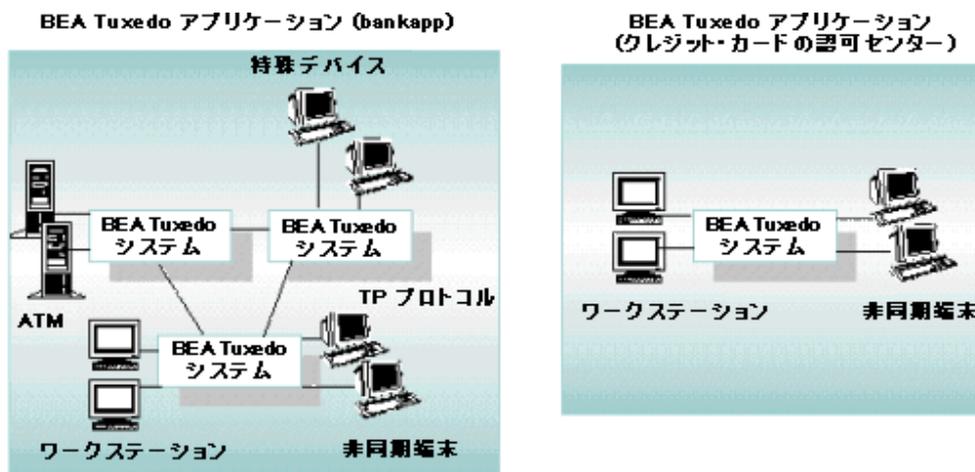
以下の節では、BEA Tuxedo ATMI Domains 環境のドメインの計画および設定の方法について説明します。

- [複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画](#)
- [creditapp Domains コンフィギュレーションの検証](#)
- [Domains コンフィギュレーションの設定](#)
- [Domains コンフィギュレーションのセキュリティの設定](#)
- [Domains コンフィギュレーションの接続の設定](#)
- [Domains コンフィギュレーションの接続の制御](#)
- [Domains リンク・レベルのフェイルオーバーとキープアライブの設定](#)

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画

次の図は、bankapp アプリケーションとクレジット・カード認可アプリケーションの2つの BEA Tuxedo アプリケーションを示しています。

図 2-12 つの BEA Tuxedo アプリケーション



bankapp アプリケーションは、各地の支店にある ATM を本店に接続します。クレジット・カード認可アプリケーションは、クレジット・カードを求める顧客の要求を処理します。銀行の支店長たちは、時間が経つにつれて、bankapp アプリケーションがクレジット・カード認可アプリケーションと直に通信できれば顧客へのサービスが向上すると考えるようになります。直接通信できれば、新しい口座を開く人に即座にクレジット・カードを発行できます。

bankapp アプリケーションは BEA Tuxedo に同梱のサンプル・アプリケーションであり、クレジット・カード認可アプリケーションは bankapp の仮想的な拡張機能です。bankapp アプリケーションのファイルは次の場所にあります。

- `tux_prod_dir\samples\atmi\bankapp` (Windows)
- `tux_prod_dir/samples/atmi/bankapp` (UNIX)

`tux_prod_dir` は、BEA Tuxedo がインストールされたディレクトリです。

次のリストは、`ubbmp` というファイルの内容を示しています。`ubbmp` は、`bankapp` アプリケーションのマルチ・マシン・バージョンの `UBBCONFIG` ファイルです。

コード リスト 2-1 bankapp アプリケーションの `ubbmp` コンフィギュレーション・ファイル

```
.
.
.
*RESOURCES
IPCKEY          80952
UID             <user id from id(1)>
GID             <group id from id(1)>
PERM           0660
MAXACCESSERS   40
MAXSERVERS     35
MAXSERVICES    75
MAXCONV        10
MAXGTT         20
MASTER         SITE1,SITE2
SCANUNIT       10
SANITYSCAN     12
BBLQUERY       30
BLOCKTIME      30
DBBLWAIT       6
OPTIONS        LAN,MIGRATE
MODEL          MP
LDBAL          Y
##SECURITY     ACL
##AUTHSVC      ". .AUTHSVC"
#
*MACHINES
<SITE1's uname> LMID=SITE1
                 TUXDIR="<TUXDIR1>"
                 APPDIR="<APPDIR1>"
                 ENVFILE="<APPDIR1>/ENVFILE"
                 TLOGDEVICE="<APPDIR1>/TLOG"
                 TLOGNAME=TLOG
                 TUXCONFIG="<APPDIR1>/tuxconfig"
```

2 ATMI Domains の計画とコンフィギュレーション

```
TYPE="<machine type1>"
ULOGPFX="<APPDIR1>/ULOG"
<SITE2's uname> LMID=SITE2
TUXDIR="<TUXDIR2>"
APPDIR="<APPDIR2>"
ENVFILE="<APPDIR2>/ENVFILE"
TLOGDEVICE="<APPDIR2>/TLOG"
TLOGNAME=TLOG
TUXCONFIG="<APPDIR2>/tuxconfig"
TYPE="<machine type2>"
ULOGPFX="<APPDIR2>/ULOG"

#
*GROUPS
#
# 認証サーバのグループ
#
##AUTHGRP          LMID=SITE1   GRPNO=101

#
# アプリケーション・キュー (/Q) サーバのグループ
#
##QGRP1            LMID=SITE1   GRPNO=102
##                TMSNAME=TMS_QM TMSCOUNT=2
##                OPENINFO="TUXEDO/QM:<APPDIR1>/qdevice:QSP_BANKAPP"

#
# アプリケーション・マネージャのサーバのグループ
#
##MGRGRP1          LMID=SITE1   GRPNO=103

#
# イベント・ブローカ・サーバのグループ
#
##EVBGRP1          LMID=SITE1   GRPNO=104

DEFAULT:           TMSNAME=TMS_SQL TMSCOUNT=2
BANKB1             LMID=SITE1   GRPNO=1
                  OPENINFO="TUXEDO/SQL:<APPDIR1>/bankd11:bankdb:readwrite"
BANKB2             LMID=SITE2   GRPNO=2
                  OPENINFO="TUXEDO/SQL:<APPDIR2>/bankd12: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>"

*SERVERS
```

2-4 『BEA Tuxedo Domains コンポーネント』

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画

```
#
# アプリケーション固有の認証を提供する TUXEDO System /T サーバ
# Ref. AUTHSVR(5).
#
##AUTHSVR          SRVGRP=AUTHGRP  SRVID=1  RESTART=Y  GRACE=0  MAXGEN=2
##                  CLOPT="-A"

#
# TUXEDO System /T メッセージ・キュー・マネージャ。tpenqueue(3) および
# tpdequeue(3) を呼び出すプログラムの代わりにメッセージを
# キューに登録したり、取り出したりするサーバ。TMQUEUE(5) を参照
#
##TMQUEUE          SRVGRP=QGRP1    SRVID=1  CONV=N  GRACE=0
##                  CLOPT="-s QSP_BANKAPP:TMQUEUE"

#
# TUXEDO System /T メッセージ転送サーバ。後で処理するように tpenqueue(3)
# で格納されたメッセージを転送する。TMQFORWARD(5) を参照
#
##TMQFORWARD       SRVGRP=QGRP1    SRVID=2  CONV=N  REPLYQ=N  GRACE=0
##                  CLOPT="-- -e -n -d -q Q_OPENACCT_LOG"

#
# TUXEDO System /T ユーザ・イベント・ブローカ。イベントがポストされたときに
# サブスクリाइブに通知してユーザ・イベントを管理する。TMUSREVT(5) を参照
#
##TMUSREVT         SRVGRP=EVBGRP1  SRVID=1  GRACE=3600
##                  ENVFILE="<APPDIR1>/TMUSREVT.ENV"
##                  CLOPT="-e tmusrevt.out -o tmusrevt.out -A --
##                  -f <APPDIR1>/tmusrevt.dat"
##                  SEQUENCE=11

#
# 特定のイベントをサブスクリाइブする TUXEDO アプリケーション・サーバ
#
##ACCTMGR SRVGRP=MGRGRP1          SRVID=1
##                  CLOPT="-A -o ACCTMGR.LOG -- -w 1000.00"
##                  SEQUENCE=12

DEFAULT: RESTART=Y    MAXGEN=5    REPLYQ=Y  CLOPT="-A"

TLR          SRVGRP=BANKB1    SRVID=1  RQADDR=tlr1
              CLOPT="-A -- -T 100 -e 1000.00"
TLR          SRVGRP=BANKB1    SRVID=2  RQADDR=tlr1
              CLOPT="-A -- -T 200 -e 1000.00"
TLR          SRVGRP=BANKB2    SRVID=3  RQADDR=tlr2
              CLOPT="-A -- -T 600 -e 1000.00"
TLR          SRVGRP=BANKB2    SRVID=4  RQADDR=tlr2
              CLOPT="-A -- -T 700 -e 1000.00"
```

2 ATMI Domains の計画とコンフィギュレーション

```
XFER          SRVGRP=BANKB1   SRVID=5
XFER          SRVGRP=BANKB2   SRVID=6
ACCT          SRVGRP=BANKB1   SRVID=7
ACCT          SRVGRP=BANKB2   SRVID=8
BAL           SRVGRP=BANKB1   SRVID=9
BAL           SRVGRP=BANKB2   SRVID=10
BTADD        SRVGRP=BANKB1   SRVID=11
BTADD        SRVGRP=BANKB2   SRVID=12
AUDITC       SRVGRP=BANKB1   SRVID=13   CONV=Y   MIN=1   MAX=10   RQADDR="auditc"
BALC         SRVGRP=BANKB1   SRVID=24
BALC         SRVGRP=BANKB2   SRVID=25
#
*SERVICES
DEFAULT:     LOAD=50   AUTOTRAN=Y   TRANTIME=30
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
BR_ADD       PRIO=20   ROUTING=BRANCH_ID
TLR_ADD      PRIO=20   ROUTING=BRANCH_ID
ABAL         PRIO=30   ROUTING=b_id
TBAL         PRIO=30   ROUTING=b_id
ABAL_BID     PRIO=30   ROUTING=b_id
TBAL_BID     PRIO=30   ROUTING=b_id
ABALC_BID    PRIO=30   ROUTING=b_id
TBALC_BID    PRIO=30   ROUTING=b_id
#
*ROUTING
ACCOUNT_ID   FIELD=ACCOUNT_ID
              BUFTYPE="FML"
              RANGES="10000-59999:BANKB1,
                    60000-109999:BANKB2"
BRANCH_ID    FIELD=BRANCH_ID
              BUFTYPE="FML"
              RANGES="1-5:BANKB1,
                    6-10:BANKB2"
b_id         FIELD=b_id
              BUFTYPE="VIEW:aud"
              RANGES="1-5:BANKB1,
                    6-10:BANKB2"
```

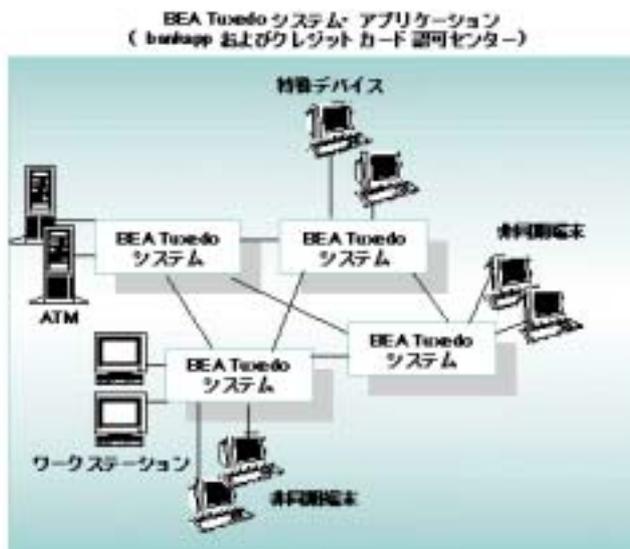
以下の節では、bankapp アプリケーションとクレジット・カード認可アプリケーションが直接通信できるように再設定する 2 通りの方法を示します。

- 2-7 ページの「オプション 1: 単一の BEA Tuxedo ドメインとしてアプリケーションを再設定する」
- 2-13 ページの「オプション 2: アプリケーションを Domains コンフィギュレーションとして再設定する」

オプション 1: 単一の BEA Tuxedo ドメインとしてアプリケーションを再設定する

1 つの方法は、次の図のように、bankapp アプリケーションとクレジット・カード認可アプリケーションを 1 つの BEA Tuxedo アプリケーション (ドメイン) に結合することです。

図-2-22 つの BEA Tuxedo システム・アプリケーションの結合



結合したアプリケーションの UBBCONFIG ファイルの作成

結合したアプリケーションの UBBCONFIG ファイルを作成するには、クレジット・カード認可アプリケーションの UBBCONFIG ファイルから次の情報を取り出し、それを bankapp アプリケーションの UBBCONFIG ファイルに追加します。

- クレジット・カード認可アプリケーションのマシン、ネットワーク、およびグループのエントリを UBBCONFIG ファイルに追加します。
- クレジット・カード認可アプリケーションのサーバ・エントリを UBBCONFIG ファイルに追加します。
- クレジット・カード認可アプリケーションのサービス・エントリを UBBCONFIG ファイルに追加します。

次のリストは、結合したアプリケーションの UBBCONFIG ファイル例を示しています。

コード リスト 2-2 結合したアプリケーションのサンプル UBBCONFIG ファイル

```
*RESOURCES
IPCKEY          76666
UID             0000
GID             000
PERM            0660
MAXACCESSERS   40
MAXSERVERS     35
MAXSERVICES    75
MAXCONV        10
MAXGTT         100
MASTER        SITE1,SITE2
SCANUNIT       10
SANITYSCAN     5
BBLQUERY       50
BLOCKTIME      2
DBBLWAIT       6
OPTIONS        LAN,MIGRATE
MODEL          MP
LDBAL          Y
#
*MACHINES
#
```

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画

```
# bankapp 部分のマシン
mach1      LMID=SITE1
           TUXDIR="/home/rsmith/tuxroot"
           APPDIR="/home/rsmith/bankapp"
           ENVFILE="/home/rsmith/bankapp/ENVFILE"
           TLOGDEVICE="/home/rsmith/bankapp/TLOG"
           TLOGNAME=TLOG
           TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
           TYPE="type1"
           ULOGPFX="/home/rsmith/bankapp/ULOG"
mach2      LMID=SITE2
           TUXDIR="/home/rsmith/tuxroot"
           APPDIR="/home/rsmith/bankapp"
           ENVFILE="/home/rsmith/bankapp/ENVFILE"
           TLOGDEVICE="/home/rsmith/bankapp/TLOG"
           TLOGNAME=TLOG
           TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
           TYPE="type2"
           ULOGPFX="/home/rsmith/bankapp/ULOG"
mach3      LMID=SITE3
           TUXDIR="/home/rsmith/tuxroot"
           APPDIR="/home/rsmith/bankapp"
           ENVFILE="/home/rsmith/bankapp/ENVFILE"
           TLOGDEVICE="/home/rsmith/bankapp/TLOG"
           TLOGNAME=TLOG
           TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
           TYPE="type2"
           ULOGPFX="/home/rsmith/bankapp/ULOG"

#
# クレジット・カード認可部分のマシン
sfexpz     LMID=SITE4
           TUXDIR="/home/rsmith/tuxroot"
           APPDIR="/home/rsmith/bankapp"
           ENVFILE="/home/rsmith/bankapp/ENVFILE"
           TLOGDEVICE="/home/rsmith/bankapp/TLOG"
           TLOGNAME=TLOG
           TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
           TYPE="type1"
           ULOGPFX="/home/rsmith/bankapp/ULOG"

#
*GROUPS
DEFAULT:  TMSNAME=TMS_SQL      TMSCOUNT=2
#
# bankapp 部分のグループ
BANKB1    LMID=SITE1           GRPNO=1
           OPENINFO="TUXEDO/SQL:/home/rsmith/bankapp/bankd11:bankdb:readwrite&dr
q;
BANKB2    LMID=SITE2           GRPNO=2
```

2 ATMI Domains の計画とコンフィギュレーション

```
OPENINFO="TUXEDO/SQL:/home/rsmith/bankapp/bankdl2:bankdb:readwrite&dr
q;
BANKB3    LMID=SITE3          GRPNO=3
OPENINFO="TUXEDO/SQL:/home/rsmith/bankapp/bankdl3:bankdb:readwrite&dr
q;
#
# クレジット・カード認可部分のグループ
CREDIT    LMID=SITE4          GRPNO=4
OPENINFO="TUXEDO/SQL:/home/rsmith/bankapp/crtdl1:bankdb:readwrite&drq
;
#
*NETWORK
#
# bankapp 部分のネットワーク接続
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>"
SITE3     NADDR="<network address of SITE3>"
          BRIDGE="<device of provider3>"
          NLSADDR="<network listener address of SITE3>"

#
# クレジット・カード認可部分のネットワーク接続
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 部分のサーバ
TLR      SRVGRP=BANKB1      SRVID=1          RQADDR=tlr1
          CLOPT="-A -- -T 100 -e 1000.00"
TLR      SRVGRP=BANKB1      SRVID=2          RQADDR=tlr1
          CLOPT="-A -- -T 200 -e 1000.00"
TLR      SRVGRP=BANKB2      SRVID=3          RQADDR=tlr2
          CLOPT="-A -- -T 600 -e 1000.00"
TLR      SRVGRP=BANKB2      SRVID=4          RQADDR=tlr2
          CLOPT="-A -- -T 700 -e 1000.00"
TLR      SRVGRP=BANKB3      SRVID=5          RQADDR=tlr3
          CLOPT="-A -- -T 800 -e 1000.00"
TLR      SRVGRP=BANKB3      SRVID=6          RQADDR=tlr3
          CLOPT="-A -- -T 900" -e 1000.00
XFER     SRVGRP=BANKB1      SRVID=7
```

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画

```
XFER      SRVGRP=BANKB2   SRVID=8
XFER      SRVGRP=BANKB3   SRVID=9
ACCT      SRVGRP=BANKB1   SRVID=10
ACCT      SRVGRP=BANKB2   SRVID=11
ACCT      SRVGRP=BANKB3   SRVID=12
BAL       SRVGRP=BANKB1   SRVID=13
BAL       SRVGRP=BANKB2   SRVID=14
BAL       SRVGRP=BANKB3   SRVID=15
BTADD     SRVGRP=BANKB1   SRVID=16
BTADD     SRVGRP=BANKB2   SRVID=17
BTADD     SRVGRP=BANKB3   SRVID=18
AUDITC    SRVGRP=BANKB1   SRVID=19      CONV=Y  MIN=1  MAX=10  RQADDR="audite"
BALC      SRVGRP=BANKB1   SRVID=20
BALC      SRVGRP=BANKB2   SRVID=21
BALC      SRVGRP=BANKB3   SRVID=22
#
# クレジット・カード認可部分のサーバ
TLRA      SRVGRP=CREDIT   SRVID=26
          CLOPT="-A -- -T 300"

ACCTA     SRVGRP=CREDIT   SRVID=27
CRDT      SRVGRP=CREDIT   SRVID=35
#
*SERVICES
DEFAULT:  LOAD=50  AUTOTRAN=Y  TRANTIME=30
#
# bankapp 部分のサービス
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
BR_ADD      PRIO=20  ROUTING=BRANCH_ID
TLR_ADD     PRIO=20  ROUTING=BRANCH_ID
ABAL        PRIO=30  ROUTING=b_id
TBAL        PRIO=30  ROUTING=b_id
ABAL_BID    PRIO=30  ROUTING=b_id
TBAL_BID    PRIO=30  ROUTING=b_id
ABALC_BID   PRIO=30  ROUTING=b_id
TBALC_BID   PRIO=30  ROUTING=b_id
#
# クレジット・カード認可部分のサービス
WITHDRAWALA PRIO=50
INQUIRYA    PRIO=50
OPENCA      PRIO=40
CLOSECA     PRIO=40
DEPOSITA    PRIO=50
```

2 ATMI Domains の計画とコンフィギュレーション

```
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="1-5:BANKB1,
                        6-10:BANKB2,
                        11-15:BANKB3"

b_id           FIELD=b_id
                BUFTYPE="VIEW:aud"
                RANGES="1-5:BANKB1,
                        6-10:BANKB2,
                        11-15:BANKB3"
```

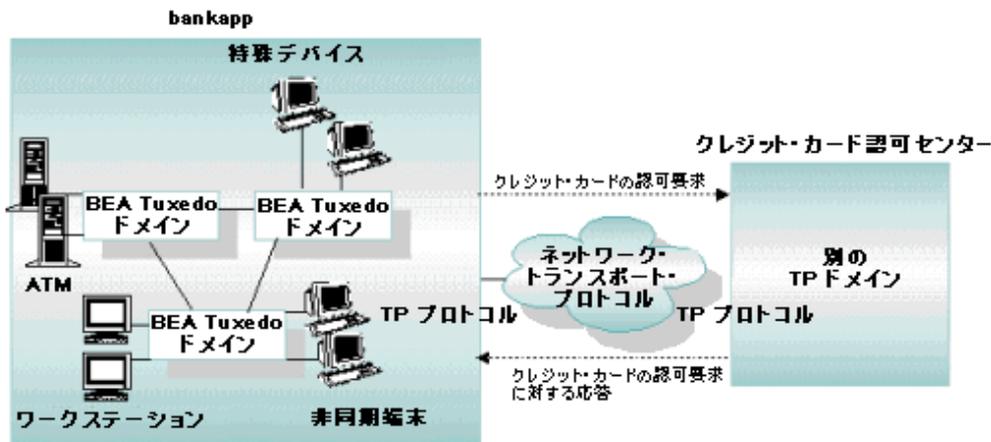
オプション 1 の制限

- 1つの大きなアプリケーションの管理は、2つの小さなアプリケーションの管理よりも複雑になることがあります。小さいアプリケーションはそれぞれに `UBBCONFIG` ファイルがあり、したがってそれ専用の管理インターフェイスを利用できます。
- ネットワーク接続されたアプリケーションの起動は、各サーバを個別に起動するのに時間がかかることと、ネットワーク経由で掲示板を複製転送しなければならないことのためにより多くの資源が消費される場合があります。より小さな別々のアプリケーションは同時に起動できます。

オプション 2: アプリケーションを Domains コンフィギュレーションとして再設定する

もう 1 つの方法は、次の図のように、bankapp アプリケーションとクレジット・カード認可アプリケーションを Domains コンフィギュレーションとして再設定することです。2 つのドメインは、それぞれ各ドメインで動作する 2 つの TDomain ゲートウェイ・サーバ・プロセスを通じて相互運用されます。

図 2-3 Domains コンフィギュレーション



bankapp アプリケーションとクレジット・カード認可アプリケーションの Domains コンフィギュレーションを作成するには、2 つの UBBCONFIG ファイル (各 BEA Tuxedo アプリケーションに 1 つ) および 2 つの DMCONFIG ファイル (各 BEA Tuxedo アプリケーションに 1 つ) を作成する必要があります。

Domains 環境の bankapp アプリケーションの UBBCONFIG ファイルの作成

Domains 環境の bankapp アプリケーションの UBBCONFIG ファイルを作成するには、2-8 ページの「結合したアプリケーションのサンプル UBBCONFIG ファイル」に示されている UBBCONFIG ファイルのコピーを土台として次の変更を行います。

- MACHINES セクションで、クレジット・カード認可アプリケーションのマシン・エントリを削除します。
- NETWORK セクションで、クレジット・カード認可アプリケーションのネットワーク・エントリを削除します。
- GROUPS セクションで、次の編集を行います。
 - クレジット・カード認可アプリケーションのグループ・エントリを削除します。
 - DMADM サーバのグループ・エントリを追加し、GWADM サーバと GWTDOMAIN サーバにまた別のグループ・エントリを追加します。
- SERVERS セクションで、次の編集を行います。
 - クレジット・カード認可アプリケーションのサーバ・エントリを削除します。
 - DMADM、GWADM、および GWTDOMAIN サーバのサーバ・エントリを追加します。
- SERVICES セクションで、クレジット・カード認可アプリケーションのサービス・エントリを削除します。

次のリストは、Domains 環境の bankapp アプリケーションの UBBCONFIG ファイル例を示しています。

コード リスト 2-3 Domains 環境の bankapp アプリケーションのサンプル UBBCONFIG ファイル

```
*RESOURCES
IPCKEY      76666
UID         0000
GID         000
```

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画

```
PERM                0660
MAXACCESSERS       40
MAXSERVERS         35
MAXSERVICES        75
MAXCONV            10
MAXGTT             100
MASTER            SITE1 , SITE2
SCANUNIT           10
SANITYSCAN         5
BBLQUERY           50
BLOCKTIME          2
DBBLWAIT           6
OPTIONS            LAN , MIGRATE
MODEL              MP
LDBAL              Y
MAXBUFTYPE         16
#
*MACHINES
mach1               LMID=SITE1
                   TUXDIR="/home/rsmith/tuxroot"
                   APPDIR="/home/rsmith/bankapp"
                   ENVFILE="/home/rsmith/bankapp/ENVFILE"
                   TLOGDEVICE="/home/rsmith/bankapp/TLOG"
                   TLOGNAME=TLOG
                   TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
                   TYPE="type1"
                   ULOGPFX="/home/rsmith/bankapp/ULOG"
mach2               LMID=SITE2
                   TUXDIR="/home/rsmith/tuxroot"
                   APPDIR="/home/rsmith/bankapp"
                   ENVFILE="/home/rsmith/bankapp/ENVFILE"
                   TLOGDEVICE="/home/rsmith/bankapp/TLOG"
                   TLOGNAME=TLOG
                   TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
                   TYPE="type2"
                   ULOGPFX="/home/rsmith/bankapp/ULOG"
mach3               LMID=SITE3
                   TUXDIR="/home/rsmith/tuxroot"
                   APPDIR="/home/rsmith/bankapp"
                   ENVFILE="/home/rsmith/bankapp/ENVFILE"
                   TLOGDEVICE="/home/rsmith/bankapp/TLOG"
                   TLOGNAME=TLOG
                   TUXCONFIG="/home/rsmith/bankapp/tuxconfig"
                   TYPE="type2"
                   ULOGPFX="/home/rsmith/bankapp/ULOG"
#
*GROUPS
DEFAULT: TMSNAME=TMS_SQL TMSCOUNT=2
#
```

2 ATMI Domains の計画とコンフィギュレーション

```
# bankapp のグループ
BANKB1 LMID=SITE1 GRPNO=1
OPENINFO="TUXEDO/SQL:/home/rsmith/bankapp/bankd11:bankdb:readwrite&dr
q;
BANKB2 LMID=SITE2 GRPNO=2
OPENINFO="TUXEDO/SQL:/home/rsmith/bankapp/bankd12:bankdb:readwrite&dr
q;
BANKB3 LMID=SITE3 GRPNO=3
OPENINFO="TUXEDO/SQL:/home/rsmith/bankapp/bankd13:bankdb:readwrite&dr
q;
#
# Domains のグループ
DMADMGRP LMID=SITE1 GRPNO=4
GWTGROUP LMID=SITE2 GRPNO=5
#
*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>"
SITE3 NADDR="<network address of SITE3>"
BRIDGE="<device of provider3>"
NLSADDR="<network listener address of SITE3>"
#
*SERVERS
DEFAULT: RESTART=Y MAXGEN=5 REPLYQ=Y CLOPT="-A"
#
# Domains のサーバ
DMADM SRVGRP=DMADMGRP
SRVID=1001
REPLYQ=N
RESTART=Y
GRACE=0
GWADM SRVGRP=GWTGROUP
SRVID=1002
REPLYQ=N
RESTART=Y
GRACE=0
GWTDOMAIN SRVGRP=GWTGROUP
SRVID=1003
RQADDR="GWTGROUP"
REPLYQ=N
RESTART=Y
GRACE=0
#
```

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画

```
# bankapp のサーバ
TLR      SRVGRP=BANKB1  SRVID=1      RQADDR=tlr1
         CLOPT="-A -- -T 100 -e 1000.00"
TLR      SRVGRP=BANKB1  SRVID=2      RQADDR=tlr1
         CLOPT="-A -- -T 200 -e 1000.00"
TLR      SRVGRP=BANKB2  SRVID=3      RQADDR=tlr2
         CLOPT="-A -- -T 600 -e 1000.00"
TLR      SRVGRP=BANKB2  SRVID=4      RQADDR=tlr2
         CLOPT="-A -- -T 700 -e 1000.00"
TLR      SRVGRP=BANKB3  SRVID=5      RQADDR=tlr3
         CLOPT="-A -- -T 800 -e 1000.00"
TLR      SRVGRP=BANKB3  SRVID=6      RQADDR=tlr3
         CLOPT="-A -- -T 900" -e 1000.00
XFER     SRVGRP=BANKB1  SRVID=7
XFER     SRVGRP=BANKB2  SRVID=8
XFER     SRVGRP=BANKB3  SRVID=9
ACCT     SRVGRP=BANKB1  SRVID=10
ACCT     SRVGRP=BANKB2  SRVID=11
ACCT     SRVGRP=BANKB3  SRVID=12
BAL      SRVGRP=BANKB1  SRVID=13
BAL      SRVGRP=BANKB2  SRVID=14
BAL      SRVGRP=BANKB3  SRVID=15
BTADD    SRVGRP=BANKB1  SRVID=16
BTADD    SRVGRP=BANKB2  SRVID=17
BTADD    SRVGRP=BANKB3  SRVID=18
AUDITC   SRVGRP=BANKB1  SRVID=19      CONV=Y  MIN=1  MAX=10  RQADDR="auditc"
BALC     SRVGRP=BANKB1  SRVID=20
BALC     SRVGRP=BANKB2  SRVID=21
BALC     SRVGRP=BANKB3  SRVID=22
#
*SERVICES
DEFAULT:  LOAD=50  AUTOTRAN=Y  TRANTIME=30
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
BR_ADD      PRIO=20  ROUTING=BRANCH_ID
TLR_ADD     PRIO=20  ROUTING=BRANCH_ID
ABAL        PRIO=30  ROUTING=b_id
TBAL        PRIO=30  ROUTING=b_id

ABAL_BID    PRIO=30  ROUTING=b_id
TBAL_BID    PRIO=30  ROUTING=b_id
ABALC_BID   PRIO=30  ROUTING=b_id
TBALC_BID   PRIO=30  ROUTING=b_id
#
```

2 ATMI Domains の計画とコンフィギュレーション

```
*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="1-5:BANKB1,
                        6-10:BANKB2,
                        11-15:BANKB3"

b_id            FIELD=b_id
                 BUFTYPE="VIEW:aud"
                 RANGES="1-5:BANKB1,
                        6-10:BANKB2,
                        11-15:BANKB3"
```

注記 この例では、DMADM、GWADM、および GWTDOMAIN の各サーバで REPLYQ=N が指定されています。この設定は必須ではありません。必要に応じて REPLYQ=Y を指定して、それらのどのサーバにでも応答キューを指定できます。ただし、REPLYQ が N に設定されている場合は、パフォーマンスが向上することがあります。

bankapp アプリケーションの DMCONFIG ファイルの作成

bankapp アプリケーションの DMCONFIG ファイルを作成することも必要です。次のリストはその例です。DMCONFIG ファイルのバイナリ形式 (BDMCONFIG) は、DMADM サーバと同じマシンに配置する必要があります。

コード リスト 2-4 bankapp アプリケーションのサンプル DMCONFIG ファイル

```
*DM_LOCAL
LOCAL1          GWGRP=GWTGROUP
                 TYPE=TDOMAIN
                 ACCESSPOINTID="BANK"
                 BLOCKTIME=10
                 CONNECTION_POLICY=ON_STARTUP
                 DMTLOGDEV="/home/rsmith/bankapp/DMTLOG"
                 AUDITLOG="/home/rsmith/bankapp/AUDITLOG"
```

```
#
*DM_REMOTE
REMOT1          TYPE=TDOMAIN
                ACCESSPOINTID="CREDIT.CARD"

#
# このサンプル DMCONFIG ファイルのように、DM_EXPORT セクションがない
# 場合は、ローカル・ドメインで宣言されたすべてのサービスが
# リモート・ドメインから利用できる。このため、次の bankapp サービス
# はクレジット・カード認可アプリケーションから利用できる。
#
# WITHDRAWAL
# DEPOSIT
# TRANSFER
# INQUIRY
# CLOSE_ACCT
# OPEN_ACCT
# BR_ADD
# TLR_ADD
# ABAL
# TBAL
# ABAL_BID
# TBAL_BID
# ABALC_BID
# TBALC_BID
#
*DM_IMPORT
WITHDRAWALA    RACCESSPOINT=REMOT1
                LACCESSPOINT=LOCAL1
INQUIRYA       RACCESSPOINT=REMOT1
                LACCESSPOINT=LOCAL1
OPENCA         RACCESSPOINT=REMOT1
                LACCESSPOINT=LOCAL1
CLOSECA        RACCESSPOINT=REMOT1
                LACCESSPOINT=LOCAL1
DEPOSITA       RACCESSPOINT=REMOT1
                LACCESSPOINT=LOCAL1
OPEN_ACCT2     RACCESSPOINT=REMOT1
                LACCESSPOINT=LOCAL1
OPENC          RACCESSPOINT=REMOT1
                LACCESSPOINT=LOCAL1
#
*DM_TDOMAIN
LOCAL1         NWADDR="albany.acme.com:4051"
REMOT1         NWADDR="newyork.acme.com:65431"
```

Domains 環境のクレジット・カード認可アプリケーションの UBBCONFIG ファイルの作成

Domains 環境のクレジット・カード認可アプリケーションの UBBCONFIG ファイルを作成するには、クレジット・カード認可アプリケーションの UBBCONFIG ファイルに次の変更を加えます。

- GROUPS セクションで、DMADM サーバのグループ・エントリを追加し、GWADM サーバと GWTDOMAIN サーバにまた別のグループ・エントリを追加します。
- SERVERS セクションで、DMADM、GWADM、および GWTDOMAIN サーバのサーバ・エントリを追加します。

次のリストは、Domains 環境のクレジット・カード認可アプリケーションの UBBCONFIG ファイル例を示しています。

コード リスト 2-5 Domains 環境のクレジット・カード認可アプリケーションのサンプル UBBCONFIG ファイル

```
*RESOURCES
IPCKEY          76666
UID             0000
GID             000
PERM            0660
MAXACCESSERS   40
MAXSERVERS     35
MAXSERVICES    75
MAXCONV        10
MAXGTT         100
MASTER        SITE1
SCANUNIT       10
MODEL          SHM
LDBAL          Y
#
*MACHINES
sfexpz         LMID=SITE1
               TUXDIR="/home/rsmith/tuxroot"
               APPDIR="/home/rsmith/creditapp"
               ENVFILE="/home/rsmith/creditapp/ENVFILE"
               TLOGDEVICE="/home/rsmith/creditapp/TLOG"
               TLOGNAME=TLOG
               TUXCONFIG="/home/rsmith/creditapp/tuxconfig"
               TYPE="type1"
```

複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画

```
        ULOGPFX="/home/rsmith/creditapp/ULOG"
#
*GROUPS
DEFAULT:  TMSNAME=TMS_SQL  TMSCOUNT=2
#
# クレジット・カード認可のグループ
CREDIT    LMID=SITE1      GRPNO=1
           OPENINFO="TUXEDO/SQL:/home/rsmith/creditapp/crtdll1:bankdb:readwrite&d
rqi;
#
# Domains のグループ
DMADMGRP  LMID=SITE1      GRPNO=2
GWTGROUP  LMID=SITE1      GRPNO=3
#
*SERVERS
DEFAULT:  RESTART=Y          MAXGEN=5  REPLYQ=Y  CLOPT="-A"
#
# Domains のサーバ
DMADM     SRVGRP=DMADMGRP
           SRVID=50
           REPLYQ=N
           RESTART=Y
           GRACE=0
GWADM     SRVGRP=GWTGROUP
           SRVID=60
           REPLYQ=N
           RESTART=Y
           GRACE=0
GWTDOMAIN SRVGRP=GWTGROUP
           SRVID=70
           RQADDR="GWTGROUP"
           REPLYQ=N
           RESTART=Y
           GRACE=0
#
# クレジット・カード認可のサーバ
TLRA      SRVGRP=CREDIT   SRVID=1
           CLOPT="-A -- -T 600"
ACCTA     SRVGRP=CREDIT   SRVID=2
CRDT      SRVGRP=CREDIT   SRVID=3
#
*SERVICES
DEFAULT:  LOAD=50  AUTOTRAN=Y  TRANTIME=30
# クレジット・カード認可のサービス
WITHDRAWALA  PRIO=50
INQUIRYA     PRIO=50
OPENCA       PRIO=40
CLOSECA      PRIO=40
DEPOSITA     PRIO=50
```

2 ATMI Domains の計画とコンフィギュレーション

```
OPEN_ACCT2      PRIO=40
OPEN            PRIO=40
```

注記 この例では、DMADM、GWADM、および GWTDOMAIN の各サーバで REPLYQ=N が指定されています。この設定は必須ではありません。必要に応じて REPLYQ=Y を指定して、それらのどのサーバにでも応答キューを指定できます。ただし、REPLYQ が N に設定されている場合は、パフォーマンスが向上することがあります。

クレジット・カード認可アプリケーションの DMCONFIG ファイルの作成

クレジット・カード認可アプリケーションの DMCONFIG ファイルを作成することも必要です。次のリストはその例です。

コード リスト 2-6 クレジット・カード認可アプリケーションのサンプル DMCONFIG ファイル

```
*DM_LOCAL
LOCAL1      GWGRP=GWTGROUP
            TYPE=TDOMAIN
            ACCESSPOINTID="CREDIT.CARD"
            BLOCKTIME=8
            DMTLOGDEV="/home/rsmith/creditapp/DMTLOG"
            AUDITLOG="/home/rsmith/creditapp/AUDITLOG"

#
*DM_REMOTE
REMOT1      TYPE=TDOMAIN
            ACCESSPOINTID="BANK"

#
# このサンプル DMCONFIG ファイルのように、DM_EXPORT セクションがない
# 場合は、ローカル・ドメインで宣言されたすべてのサービスが
# リモート・ドメインから利用できる。このため、次のクレジット・カード認可サービ
# ス
# は bankapp アプリケーションから利用できる。
#
# WITHDRAWALA
# INQUIRYA
# OPENCA
# CLOSECA
# DEPOSITA
```

```
# OPEN_ACCT2
# OPENC
#
*DM_IMPORT
WITHDRAWAL  RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
DEPOSIT     RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
TRANSFER    RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
INQUIRY     RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
CLOSE_ACCT  RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
OPEN_ACCT   RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
BR_ADD      RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
TLR_ADD     RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
ABAL        RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
TBAL        RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
ABALC_BID   RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
TBALC_BID   RACCESSPOINT=REMOT1
             LACCESSPOINT=LOCAL1
#
*DM_TDOMAIN
LOCAL1      NWADDR="newyork.acme.com:65431"
REMOT1      NWADDR="albany.acme.com:4051"
```

creditapp Domains コンフィギュレーションの検証

creditapp アプリケーションは、4つのマシンをまたがるサンプルの Domains コンフィギュレーションです。実質的に、creditapp アプリケーションは bankapp アプリケーションとクレジット・カード認可アプリケーション

ションが直に通信できるように再設定するもう1つの方法です (2-2 ページの「複数の BEA Tuxedo アプリケーションに基づく Domains の構築計画」を参照)。この方法では、bankapp アプリケーションとクレジット・カード認可アプリケーションが、TDomain ゲートウェイ・サーバ・プロセスを使用して相互運用される4つの BEA Tuxedo ドメイン (マシンごとに1つ) として再設定されます。

creditapp アプリケーションは、BEA Tuxedo に同梱されています。そのファイルの場所は次のとおりです。

- `tux_prod_dir\samples\atmi\creditapp` (Windows)
- `tux_prod_dir/samples/atmi/creditapp` (UNIX)

`tux_prod_dir` は、BEA Tuxedo がインストールされたディレクトリです。

creditapp アプリケーションの Domains コンフィギュレーションでは、4つの UBBCONFIG ファイル (各 BEA Tuxedo ドメインに1つ) と4つの DMCONFIG ファイル (各 BEA Tuxedo ドメインに1つ) が必要です。4つの UBBCONFIG ファイルの名前は `ubbdom1 ~ ubbdom4`、4つの DMCONFIG ファイルの名前は `domcon1 ~ domcon4` です。それらのファイルは、creditapp ディレクトリにあります。

次のリストは、`ubbdom1` コンフィギュレーション・ファイルの内容を示しています。SERVERS セクションでは、3つの TDomain ゲートウェイ・グループが設定されています。それらの TDomain ゲートウェイ・グループは、このドメインが Domains コンフィギュレーションのほかの3つのドメインと通信するために使用します。

コードリスト 2-7 creditapp アプリケーションの ubbdom1 コンフィギュレーション・ファイル

```
.
.
.
*RESOURCES
IPCKEY      80952
UID         <user id from id(1)>
GID         <group id from id(1)>
PERM       0660
MAXACCESSERS 40
MAXSERVERS 35
MAXSERVICES 75
MAXCONV    10
MASTER     SITE1
MODEL       SHM
LDBAL      Y
MAXGTT     100
MAXBUFTYPE 16
SCANUNIT   10
SANITYSCAN 5
DBBLWAIT   6
BBLQUERY   50
BLOCKTIME  2
#
#
*MACHINES
<SITE1's uname> LMID=SITE1
                 TUXDIR="<TUXDIR1>"
                 APPDIR="<APPDIR1>"
                 ENVFILE="<APPDIR1>/ENVFILE"
                 TLOGDEVICE="<APPDIR1>/TLOG"
                 TLOGNAME=TLOG
                 TUXCONFIG="<APPDIR1>/tuxconfig"
                 ULOGPFX="<APPDIR1>/ULOG"
                 TYPE="<machine typel>"

#
#
*GROUPS
DEFAULT:        LMID=SITE1
BANKB1          GRPNO=1      TMSNAME=TMS_SQLTMSCOUNT=2
                 OPENINFO="TUXEDO/SQL:<APPDIR1>/crdtdl1:bankdb:readwrite"
BANKB2          GRPNO=2
BANKB3          GRPNO=3
BANKB4          GRPNO=4
#
```

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"
DMADM             SRVGRP=BANKB2     SRVID=32
GWADM             SRVGRP=BANKB2     SRVID=30
GWTDOMAIN        SRVGRP=BANKB2     SRVID=31
GWADM             SRVGRP=BANKB3     SRVID=24
GWTDOMAIN        SRVGRP=BANKB3     SRVID=25
GWADM             SRVGRP=BANKB4     SRVID=20
GWTDOMAIN        SRVGRP=BANKB4     SRVID=21
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
```

次のリストは、domcon1 Domains コンフィギュレーション・ファイルの内容を示しています。DM_LOCAL セクション (DM_LOCAL_DOMAINS セクションとも呼ばれる) では、3 つの TDomain ゲートウェイ・グループが設定されています。それらの TDomain ゲートウェイ・グループは、このドメインが Domains コンフィギュレーションのほかの 3 つのドメインと通信するために使用します。ここで示されている domcon1 の内容は、[1-29 ページの「DMCONFIG ファイル関連の新しい用語」](#)で説明した新しい Domains 用語を使って更新されています。

コードリスト 2-8 creditapp アプリケーションの domcon1 Domains コンフィギュレーション・ファイル

```
.
.
.
*DM_RESOURCES
#
VERSION=U22
#
#
#
*DM_LOCAL
#
QDOM1      GWGRP=BANKB2
           TYPE=TDOMAIN
           ACCESSPOINTID="QDOM1"
           BLOCKTIME=10
           MAXACCESSPOINT=89
           DMTLOGDEV="<APPDIR1>/DMTLOG"
           AUDITLOG="<APPDIR1>/AUDITLOG"
           DMTLOGNAME="DMTLOG_TDOM1"

QDOM2      GWGRP=BANKB3
           TYPE=TDOMAIN
           ACCESSPOINTID="QDOM2"
           BLOCKTIME=10
           MAXACCESSPOINT=89
           DMTLOGDEV="<APPDIR1>/DMTLOG"
           AUDITLOG="<APPDIR1>/AUDITLOG"
           DMTLOGNAME="DMTLOG_TDOM2"

QDOM3      GWGRP=BANKB4
           TYPE=TDOMAIN
           ACCESSPOINTID="QDOM3"
           BLOCKTIME=10
           MAXACCESSPOINT=89
           DMTLOGDEV="<APPDIR1>/DMTLOG"
           AUDITLOG="<APPDIR1>/AUDITLOG"
           DMTLOGNAME="DMTLOG_TDOM3"

#
#
*DM_REMOTE
#
TDOM1      TYPE=TDOMAIN
           ACCESSPOINTID="TDOM1"
```

2 ATMI Domains の計画とコンフィギュレーション

```
TDOM2          TYPE=TDOMAIN
                ACCESSPOINTID="TDOM2"

TDOM3          TYPE=TDOMAIN
                ACCESSPOINTID="TDOM3"

#
#
*DM_TDOMAIN
#
TDOM1          NWADDR="<network address of SITE2>"
                NWDEVICE="<device of provider2>"

TDOM2          NWADDR="<network address of SITE3>"
                NWDEVICE="<device of provider3>"

TDOM3          NWADDR="<network address of SITE4>"
                NWDEVICE="<device of provider4>"

QDOM1          NWADDR="<network address of SITE1>"
                NWDEVICE="<device of provider1>"

QDOM2          NWADDR="<network address of SITE1A>"
                NWDEVICE="<device of provider1>"

QDOM3          NWADDR="<network address of SITE1B>"
                NWDEVICE="<device of provider1>"

#
#
*DM_EXPORT
#
WITHDRAWALA
INQUIRYA
OPENCA
CLOSECA
```

creditapp アプリケーションを実行する場合は、まず creditapp ディレクトリにある README ファイルを読んでください。README ファイルでは、RUNME.sh という UNIX シェル・スクリプトを使用して creditapp アプリケーションを実行する方法が説明されています。Windows システムで creditapp アプリケーションを実行する場合は、README ファイルで基本的な設定情報を読んでから、Windows 環境で設定タスクを行います。Windows 上での BEA Tuxedo の使い方の詳細については、『[Windows NT での BEA Tuxedo システムの使用](#)』を参照してください。

Domains コンフィギュレーションの設定

Domains 環境を設定する場合、Domains 管理者は Domains コンフィギュレーションにかかわるほかのドメイン (リモート・ドメイン) について BEA Tuxedo ドメインが知る必要のあるすべての情報を指定する必要があります。この情報には、リモート・ドメインにエクスポートされるローカル・サービス、リモート・ドメインからインポートされるサービス、およびリモート・ドメインにアクセスするためのアドレッシング・パラメータとセキュリティ・パラメータが含まれます。この情報は、Domains コンフィギュレーションにかかわる各ドメインの `UBBCONFIG` および `DMCONFIG` コンフィギュレーション・ファイルで定義されます。

以下の節で説明する Domains の例は、BEA Tuxedo に同梱されている `simpapp` アプリケーションに基づいています。このサンプル・アプリケーションは次の場所にあります。

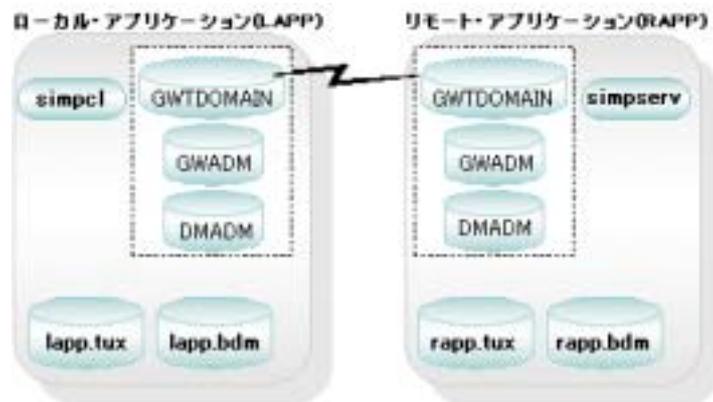
- `tux_prod_dir\samples\atmi\simpapp` (Windows)
- `tux_prod_dir/samples/atmi/simpapp` (UNIX)

`tux_prod_dir` は、BEA Tuxedo がインストールされたディレクトリです。

サンプル Domains アプリケーション (simpapp) の設定

次の図で示されている Domains の例は、2 つの BEA Tuxedo ドメインで構成されています。1 つは `lapp` (`simpapp` に基づくローカル・アプリケーション)、もう 1 つは `rapp` (`simpapp` に基づくリモート・アプリケーション) です。`lapp` アプリケーションは、`rapp` アプリケーションで利用可能な `TOUPPER` というサービスにそのクライアントがアクセスできるように設定されています。

図 2-4simpapp 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・タスク

`lapp` アプリケーションと `rapp` アプリケーションを設定するには、次のタスクを行う必要があります。



lapp の環境変数の設定

lapp アプリケーションを正常に設定するためには、次の環境変数を設定する必要があります。

- TUXDIR このマシン上の BEA Tuxedo システムのルート・ディレクトリへの絶対パス名。 *tux_prod_dir* として表されることもあります。
- APPDIR このマシン上の lapp アプリケーションのルート・ディレクトリへの絶対パス名。
- TUXCONFIG このマシン上で lapp のバイナリ形式のアプリケーション・コンフィギュレーション・ファイルが置かれているデバイスまたはファイルの絶対パス名。
- BDMCONFIG このマシン上で lapp のバイナリ形式の Domains コンフィギュレーション・ファイルが置かれているデバイスまたはファイルの絶対パス名。
- PATH %TUXDIR%\bin (Windows) または \$TUXDIR/bin (UNIX) を指定する必要があります。
- LD_LIBRARY_PATH (UNIX のみ) このマシンでロードする必要がある動的に読み込み可能なライブラリのリスト (\$TUXDIR/lib が必須)。HP-UX (HP 9000) では、LD_LIBRARY_PATH の代わりに SHLIB_PATH を使用します。

Windows の例

```
prompt> set TUXDIR=C:\bea\tuxedo
prompt> set APPDIR=C:\home\lapp
prompt> set TUXCONFIG=C:\home\lapp\lapp.tux
prompt> set BDMCONFIG=C:\home\lapp\lapp.bdm
prompt> set PATH=%APPDIR%;%TUXDIR%\bin;%PATH%
```

注記 Windows は、PATH 変数に設定されたパスを使って、動的に読み込み可能な必須ライブラリ・ファイルにアクセスします。

UNIX の例

```

prompt> TUXDIR=/home/rsmith/boa/tuxedo
prompt> APPDIR=/home/rsmith/lapp
prompt> TUXCONFIG=/home/rsmith/lapp/lapp.tux
prompt> BDMCONFIG=/home/rsmith/lapp/lapp.bdm
prompt> PATH=$APPDIR:$TUXDIR/bin:/bin:$PATH
prompt> LD_LIBRARY_PATH=$APPDIR:$TUXDIR/lib:/lib:/usr/lib:
        $LD_LIBRARY_PATH
prompt> export TUXDIR APPDIR TUXCONFIG BDMCONFIG PATH LD_LIBRARY_PATH

```

UBBCONFIG ファイルでの lapp の Domains 環境の定義

lapp.ubb (lapp アプリケーションのテキスト形式のコンフィギュレーション・ファイル) では、必須パラメータのみ定義します。ほかのパラメータにはデフォルト設定を使用します。次のリストは、lapp.ubb の内容を示しています。

コード リスト 2-9 lapp.ubb コンフィギュレーション・ファイル

```

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

*MACHINES
giselle

                LMID=LAPP
                TUXDIR="/home/rsmith/tuxedo"
                APPDIR="/home/rsmith/lapp"
                TUXCONFIG="/home/rsmith/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
.
.
.
```

注記 この UBBCONFIG ファイルでは、REPLYQ=N が DMADM、GWADM、および GWTDOMAIN の各サーバで指定されています。この設定は必須ではありません。必要に応じて REPLYQ=Y を指定して、それらのどのサーバにでも応答キューを指定できます。ただし、REPLYQ が N に設定されている場合は、パフォーマンスが向上することがあります。

サーバ・グループの定義

lapp.ubb では、次のサーバ・グループが定義されています。

- LDMGRP Domains 管理サーバ (DMADM) が含まれます。
- LGWGRP ゲートウェイ管理サーバ (GWADM) と TDomain ゲートウェイ・サーバ (GWTDOMAIN) が含まれます。

サーバの定義

- DMADM Domains 管理サーバは、バイナリ形式の Domains コンフィギュレーション・ファイル (BDMCONFIG) の Domains コンフィギュレーション情報を実行時に修正します。DMADM は、登録済みのゲートウェイ・グループをリストします。Domains コンフィギュレーションに参加する BEA Tuxedo ドメインで実行できる 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、DMADM サーバから Domains コンフィギュレーション情報を取得します。また、ゲートウェイ

イ・グループを管理したり、トランザクション・ログを記録したりもします。

- `GWTDOMAIN` TDomain ゲートウェイ・サーバはリモート BEA Tuxedo ドメインとの相互のアクセスを可能にし、複数の BEA Tuxedo ドメインの相互運用を実現します。TDomain ゲートウェイでエクスポートおよびインポートされるローカル・サービスとリモート・サービスについては、Domains コンフィギュレーション・ファイル (`DMCONFIG`) を参照してください。

DMCONFIG ファイルでの lapp の Domains パラメータの定義

`lapp.dom` (`lapp` Domains のテキスト形式のコンフィギュレーション・ファイル) では、必須パラメータのみ定義します。オプション・パラメータにはデフォルト設定を使用します。次のリストは、`lapp.dom` ファイルの内容を示しています。

コード リスト 2-10 lapp.dom Domains コンフィギュレーション・ファイル

```
#
# lapp.dom
#
*DM_LOCAL
LAPP          GWGRP=LGWRP
              TYPE=TDOMAIN
              ACCESSPOINTID="111111"

*DM_REMOTE
RAPP          TYPE=TDOMAIN
              ACCESSPOINTID="222222"

*DM_EXPORT

*DM_IMPORT
TOUPPER

*DM_TDOMAIN
LAPP          NWADDR="//giselle:5000"
RAPP          NWADDR="//juliet:5000"
```

DM_LOCAL セクションの定義

DM_LOCAL セクションでは、ローカル・ドメイン・アクセス・ポイント、それらに関連するドメイン・ゲートウェイ・グループ、およびそれらの特性を指定します。ローカル・ドメイン・アクセス・ポイントは、ドメイン・ゲートウェイ・グループごとに1つのみです。

lapp.dom ファイルでは、唯一のローカル・ドメイン・アクセス・ポイント LAPP を指定し、その LAPP アクセス・ポイントの次のプロパティを定義します。

- GWGRP の値は LGWRP で、これは lapp.ubb ファイルで指定されたドメイン・ゲートウェイ・サーバ・グループの名前です。
- TYPE の値 TDOMAIN は、lapp アプリケーションがローカルの TDomain ゲートウェイ・サーバを通じて rapp アプリケーショ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。TYPE のほかの値には、TOPEND (BEA TOP END ゲートウェイ)、IDOMAIN (BEA eLink Adapter for Mainframe ゲートウェイ)、SNAX

(BEA eLink Adapter for Mainframe SNA ゲートウェイ)、および OSITP/OSITPX (BEA eLink Adapter for Mainframe OSI TP ゲートウェイ) があります。

- ACCESSPOINTID では、ローカル・ドメイン・アクセス・ポイントの名前を指定します。この識別子は、Domains コンフィギュレーションにかかわるすべてのドメインの中で一意である必要があります。

DM_REMOTE セクションの定義

DM_REMOTE セクションでは、リモート・ドメイン・アクセス・ポイントとその特性を指定します。リモート・ドメイン・アクセス・ポイントは、各ドメイン・ゲートウェイ・グループに1つでも複数でもかまいません。

lapp.dom ファイルでは、唯一のリモート・ドメイン・アクセス・ポイント RAPP を指定し、その RAPP アクセス・ポイントの次のプロパティを定義します。

- TYPE の値 TDOMAIN は、lapp アプリケーションがローカルの TDomain ゲートウェイ・サーバを通じて rapp アプリケーションと通信することを示します。
- ACCESSPOINTID では、リモート・ドメイン・アクセス・ポイントの名前を指定します。この識別子は、Domains コンフィギュレーションにかかわるすべてのドメインの中で一意である必要があります。

DM_EXPORT セクションの定義

DM_EXPORT セクションでは、ローカル・ドメイン・アクセス・ポイントを通じて1つ以上のリモート・ドメインにエクスポートされるサービスの情報を指定します。このセクションがないか、あっても何も指定されていない場合は、ローカル・ドメインで宣言されたすべてのサービスが、DM_REMOTE セクションで定義されたアクセス・ドメインと関連付けられたリモート・ドメインから利用可能になります。

lapp.dom ファイルで指定されているように、LAPP アクセス・ポイントを通じて rapp アプリケーションから利用できる lapp サービスはありません。

DM_IMPORT セクションの定義

DM_IMPORT セクションでは、1 つ以上のリモート・ドメイン・アクセス・ポイントを通じてインポートされ、1 つ以上のローカル・ドメイン・アクセス・ポイントを通じてローカル・ドメインから利用できるようになるサービスの情報を指定します。このセクションがないか、あっても何も指定されていない場合、ローカル・ドメインからはどのリモート・サービスも利用できません。

lapp.dom ファイルで指定されているように、TOUPPER という rapp サービスが lapp アプリケーションから利用できます。

DM_TDOMAIN セクションの定義

DM_TDOMAIN セクションでは、BEA Tuxedo Domains コンポーネントに必要なアドレッシング情報を定義します。コンフィギュレーション・ファイルの LOCAL セクションおよび REMOTE セクションで指定された各ドメイン・アクセス・ポイントは、DM_TDOMAIN セクションのエントリとなります。

各ローカル・ドメイン・アクセス・ポイントのエントリには NWADDR 値が関連付けられます。この値は、ローカル・ドメインが1 つ以上のリモート・ドメインからの接続を受け付けるネットワーク・アドレスを指定します。

各リモート・ドメイン・アクセス・ポイントのエントリには NWADDR 値が関連付けられます。この値は、ローカル・ドメインがリモート・ドメインへの接続を試みるネットワーク・アドレスを指定します。

lapp.dom ファイルで指定されているように、lapp アプリケーションはネットワーク・アドレス giselle:5000 で受信時接続要求をリスンします。giselle は lapp アプリケーションが動作しているマシンの名前、5000 はリスン・ポートです。これも lapp.dom で指定されていますが、lapp アプリケーションが rapp アプリケーションに接続を試みる際には、ネットワーク・アドレス juliet:5000 が使用されます。juliet は rapp アプリケーションが動作しているマシンの名前、5000 は接続先ポートです。

lapp のアプリケーションおよび Domains ゲートウェイ・コンフィギュレーション・ファイルのコンパイル

lapp.ubb アプリケーション・コンフィギュレーション・ファイルには、lapp アプリケーションの起動に必要な情報が格納されます。このファイルは、`tmloadcf(1)` を実行してバイナリ・データ・ファイルにコンパイルします。

lapp.dom Domains コンフィギュレーション・ファイルには、リモートの rapp TDomain ゲートウェイと通信するためにローカルの lapp TDomain ゲートウェイで使用される情報が格納されます。このファイルは、`dmloadcf(1)` を実行してバイナリ・データ・ファイルにコンパイルします。

両方のコンフィギュレーション・ファイルのコンパイルで、次のセッション例を参考にしてください。

Windows:

```
prompt> cd C:\home\lapp
prompt> set TUXCONFIG=C:\home\lapp\lapp.tux
prompt> tmloadcf -y lapp.ubb
prompt> set BDMCONFIG=C:\home\lapp\lapp.bdm
prompt> dmloadcf -y lapp.dom
```

UNIX:

```
prompt> cd /home/rsmith/lapp
prompt> TUXCONFIG=/home/rsmith/lapp/lapp.tux
prompt> export TUXCONFIG
prompt> tmloadcf -y lapp.ubb
prompt> BDMCONFIG=/home/rsmith/lapp/lapp.bdm
prompt> export BDMCONFIG
prompt> dmloadcf -y lapp.dom
```

lapp アプリケーションと rapp アプリケーションの両方を作成したら、それぞれのマシンで次のように `tmboot(1)` コマンドを実行してアプリケーションを起動します。

```
prompt> tmboot -y
```

アプリケーションを起動する順序は関係ありません。4-1 ページの「Domains の管理」の説明に従って、`dmadmin(1)` を使用してアプリケーションを監視します。アプリケーションが両方とも起動したら、`rapp` アプリケーションで提供される TOUPPER サービスを `lapp` アプリケーションのクライアントから呼び出すことができます。

rapp の環境変数の設定

`rapp` アプリケーションを正常に設定するためには、次の環境変数を設定する必要があります。

- `TUXDIR` このマシン上の BEA Tuxedo システムのルート・ディレクトリへの絶対パス名。 `tux_prod_dir` として表されることもあります。
- `APPDIR` このマシン上の `rapp` アプリケーションのルート・ディレクトリへの絶対パス名。
- `TUXCONFIG` このマシン上で `rapp` のバイナリ形式のアプリケーション・コンフィギュレーション・ファイルが置かれているデバイスまたはファイルの絶対パス名。
- `BDMCONFIG` このマシン上で `rapp` のバイナリ形式の Domains コンフィギュレーション・ファイルが置かれているデバイスまたはファイルの絶対パス名。
- `PATH` `%TUXDIR%\bin` (Windows) または `$TUXDIR/bin` (UNIX) を指定する必要があります。
- `LD_LIBRARY_PATH` (UNIX のみ) このマシンでロードする必要のある動的に読み込み可能なライブラリのリスト (`$TUXDIR/lib` が必須)。HP-UX (HP 9000) では、`LD_LIBRARY_PATH` の代わりに `SHLIB_PATH` を使用します。

Windows の例

```
prompt> set TUXDIR=C:\bea\tuxedo
prompt> set APPDIR=C:\home\rapp
prompt> set TUXCONFIG=C:\home\rapp\rapp.tux
prompt> set BDMCONFIG=C:\home\rapp\rapp.bdm
prompt> set PATH=%APPDIR%;%TUXDIR%\bin;%PATH%
```

注記 Windows は、`PATH` 変数に設定されたパスを使って、動的に読み込み可能な必須ライブラリ・ファイルにアクセスします。

UNIX の例

```
prompt> TUXDIR=/home/rsmith/bea/tuxedo
prompt> APPDIR=/home/rsmith/rapp
prompt> TUXCONFIG=/home/rsmith/rapp/rapp.tux
prompt> BDMCONFIG=/home/rsmith/rapp/rapp.bdm
prompt> PATH=$APPDIR:$TUXDIR/bin:/bin:$PATH
prompt> LD_LIBRARY_PATH=$APPDIR:$TUXDIR/lib:/lib:/usr/lib:
    $LD_LIBRARY_PATH
prompt> export TUXDIR APPDIR TUXCONFIG BDMCONFIG PATH LD_LIBRARY_PATH
```

UBBCONFIG ファイルでの rapp の Domains 環境の定義

`rapp.ubb` (`rapp` アプリケーションのテキスト形式のコンフィギュレーション・ファイル) では、必須パラメータのみ定義します。ほかのパラメータにはデフォルト設定を使用します。次のリストは、`rapp.ubb` ファイルの内容を示しています。

コード リスト 2-11 rapp.ubb 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
#
*RESOURCES
IPCKEY          222222
MASTER         RAPP
MODEL          SHM

*MACHINES
juliet

                LMID=RAPP
                TUXDIR="/home/rsmith/bea/tuxedo"
                APPDIR="/home/rsmith/rapp"
                TUXCONFIG="/home/rsmith/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
simplserv       SRVGRP=APPGRP SRVID=1
.
.
.
*SERVICES
TOUPPER
.
.
.
```

注記 この UBBCONFIG ファイルでは、REPLYQ=N が DMADM、GWADM、および GWTDOMAIN の各サーバで指定されています。この設定は必須ではありません。必要に応じて REPLYQ=Y を指定して、それらのどのサーバにでも応答キューを指定できます。ただし、REPLYQ が N に設定されている場合は、パフォーマンスが向上することがあります。

rapp.ubb では、次のサーバ・グループが定義されています。

- RDMGRP Domains サーバ DMADM が含まれます。
- RGWGRP Domains サーバ GWADM と GWTDOMAIN が含まれます。
- APPGRP アプリケーション・サーバ `simpserver` が含まれます。

`simpserver` サーバは、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。

DMCONFIG ファイルでの `rapp` の Domains パラメータの定義

`rapp.dom` (`rapp` Domains のテキスト形式のコンフィギュレーション・ファイル) では、必須パラメータのみ定義します。ほかのパラメータにはデフォルト設定を使用します。次のリストは、`rapp.dom` ファイルの内容を示しています。

コード リスト 2-12 `rapp.dom` Domains コンフィギュレーション・ファイル

```
# rapp.dom
#
*DM_LOCAL
RAPP          GWGRP=RGWGRP
              TYPE=TDOMAIN
              ACCESSPOINTID="222222"

*DM_REMOTE
LAPP          TYPE=TDOMAIN
              ACCESSPOINTID="111111"

*DM_EXPORT
TOUPPER

*DM_IMPORT

*DM_TDOMAIN
RAPP          NWADDR="//juliet:5000"
LAPP          NWADDR="//giselle:5000"
```

rapp.dom Domains コンフィギュレーション・ファイルは lapp.dom Domains コンフィギュレーション・ファイルと似ていますが、エクスポートおよびインポートされるサービスは異なります。具体的には、rapp.dom ファイルでは rapp アプリケーションの次の Domains コンフィギュレーションが定義されます。

- RAPP というローカル・ドメイン・アクセス・ポイントと LAPP というリモート・ドメイン・アクセス・ポイントを指定します。両方のアクセス・ポイントとも、RGWGRP という TDomain ゲートウェイ・サーバ・グループと関連付けられます。
- TOUPPER という rapp サービスが lapp アプリケーションから利用できるように指定します。
- どの lapp サービスも rapp アプリケーションから利用できないように指定します。
- rapp アプリケーションがネットワーク・アドレス juliet:5000 で受信時接続要求をリスンするように指定します。juliet は rapp アプリケーションが動作しているマシンの名前、5000 はリスン・ポートです。
- rapp アプリケーションが lapp アプリケーションに接続を試みるときに、ネットワーク・アドレス giselle:5000 が使用されるように指定します。giselle は lapp アプリケーションが動作しているマシンの名前、5000 は接続先ポートです。

rapp のアプリケーションおよび Domains ゲートウェイ・コンフィギュレーション・ファイルのコンパイル

rapp.ubb アプリケーション・コンフィギュレーション・ファイルには、rapp アプリケーションの起動に必要な情報が格納されます。このファイルは、`tmloadcf(1)` を実行してバイナリ・データ・ファイルにコンパイルします。

rapp.dom Domains コンフィギュレーション・ファイルには、リモートの lapp TDomain ゲートウェイと通信するためにローカルの rapp TDomain ゲートウェイで使用される情報が格納されます。このファイルは、[dmloadcf\(1\)](#) を実行してバイナリ・データ・ファイルにコンパイルします。

両方のコンフィギュレーション・ファイルのコンパイルで、次のセッション例を参考にしてください。

Windows:

```
prompt> cd C:\home\rapp
prompt> set TUXCONFIG=C:\home\rapp\rapp.tux
prompt> tmloadcf -y rapp.ubb
prompt> set BDMCONFIG=C:\home\rapp\rapp.bdm
prompt> dmloadcf -y rapp.dom
```

UNIX:

```
prompt> cd /home/rsmith/rapp
prompt> TUXCONFIG=/home/rsmith/rapp/rapp.tux
prompt> export TUXCONFIG
prompt> tmloadcf -y rapp.ubb
prompt> BDMCONFIG=/home/rsmith/rapp/rapp.bdm
prompt> export BDMCONFIG
prompt> dmloadcf -y rapp.dom
```

rapp アプリケーションと lapp アプリケーションの両方を作成したら、それぞれのマシンで次のように [tmboot\(1\)](#) コマンドを実行してアプリケーションを起動します。

```
prompt> tmboot -y
```

アプリケーションを起動する順序は関係ありません。[4-1 ページの「Domains の管理」](#)の説明に従って、[dmadmin\(1\)](#) を使用してアプリケーションを監視します。アプリケーションが両方とも起動したら、rapp アプリケーションで提供される TOUPPER サービスを lapp アプリケーションのクライアントから呼び出すことができます。

関連項目

- [1-19 ページの「Domains コンフィギュレーション・ファイル」](#)
- [2-46 ページの「ドメイン間のデータの圧縮」](#)
- [2-46 ページの「リモート・ドメインへのサービス要求のルーティング」](#)
- 『BEA Tuxedo のファイル形式とデータ記述方法』の [UBBCONFIG\(5\)](#)
- 『BEA Tuxedo のファイル形式とデータ記述方法』の [DMCONFIG\(5\)](#)

ドメイン間のデータの圧縮

ドメイン間を送信されるデータは、高速化を目的として圧縮できます。圧縮を設定するには、DMCONFIG ファイルの DM_TDOMAIN セクションで Cmplimit パラメータを設定します。このパラメータ (リモート・ドメイン・アクセス・ポイントのみで有効) では、リモート・ドメインにデータを送信するときの圧縮しきい値を指定します。最小値は 0、最大値は 2147483647 です。デフォルト設定は 2147483647 です。指定されたサイズより大きいアプリケーション・バッファは圧縮されます。

Cmplimit パラメータ設定の詳細については、『BEA Tuxedo アプリケーション実行時の管理』の「[ネットワーク・データの圧縮](#)」を参照してください。

リモート・ドメインへのサービス要求のルーティング

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にドメイン・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 セクションで指定します。FML、FML32、VIEW、VIEW32、X_C_TYPE、X_COMMON、および XML の型付きバッファがサポートされています。

Domains コンフィギュレーションにかかわるドメインのルーティング・テーブルを作成するには、DMCONFIG ファイルの DM_ROUTING セクションで次の情報を指定します。

- ルーティング・エントリが有効なバッファ・タイプ
- ルーティング・エントリおよびルーティング・フィールドの名前
- ルーティング・フィールドの範囲と関連付けられたリモート・ドメイン名

Domains のデータ依存型ルーティング・コンフィギュレーションの例については、[1-30 ページの「Domains データ依存型ルーティングの指定」](#)を参照してください。Domains のデータ依存型ルーティングの詳細については、『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ [DMCONFIG\(5\)](#) の DM_ROUTING セクションを参照してください。

Domains コンフィギュレーションのセキュリティの設定

BEA Tuxedo ATMI 環境では、Domains コンフィギュレーションに次の基本的なセキュリティ機能を利用できます。

- 認証 接続を確立するときにローカル・ドメインおよびリモート・ドメインの ID を検証します。
- 認可 アクセス制御リスト (ACL) を利用してリモート・クライアントのローカル・サービスへのアクセスを制限します。
- リンク・レベルの暗号化 ドメイン間通信のプライバシー性を維持します。

Domains コンフィギュレーションで利用できるセキュリティ機能と個々の BEA Tuxedo アプリケーションで利用できるセキュリティ機能は比較的依存性はありませんが、互換性があります。BEA Tuxedo アプリケーションで利用できるセキュリティ機能については、『[BEA Tuxedo のセキュリティ機能](#)』を参照してください。

Domains のセキュリティ・メカニズム

BEA Tuxedo Domains コンポーネントは、次のセキュリティ・メカニズムを提供します。

- **Domains 認証** ローカル・ドメインとリモート・ドメインが互いに接続するときに相互に認証できる手段を提供します。DMCONFIG ファイルの DM_LOCAL セクションおよび DM_REMOTE セクションの CONNECTION_PRINCIPAL_NAME パラメータでローカル・ドメインおよびリモート・ドメインの ID (プリンシパル名) を指定します。

また、ローカル・ドメインとリモート・ドメインではお互いへの接続を試みる際に 3 レベルのパスワード・セキュリティのいずれかを使用できます。パスワード・セキュリティのレベルは、DMCONFIG ファイルの DM_LOCAL セクションの SECURITY パラメータを使用してローカル・ドメイン単位で設定します。
- **Domains ローカル・ドメイン・アクセス** ローカル・サービスをリモート・ドメインに制限します。サービスがリモート・ドメインにエクスポートされていない場合、そのサービスはリモート・ドメインから利用できません。サービスをエクスポートするには、DMCONFIG ファイルの DM_EXPORT セクションにそのサービスのエントリを配置します。
- **Domains アクセス制御リスト (ACL)** ローカル・ドメインのサービスの利用を特定のリモート・ドメインのみに制限します。DMCONFIG ファイルの DM_ACCESS_CONTROL セクションで ACL 名を作成して、その ACL 名を DMCONFIG ファイルの EXPORT セクションでサービスに適用します。
- **Domains ACL 方針** リモート・ドメインの ACL 方針を制御します。リモート・ドメインのローカルまたはグローバルの ACL 方針は、DMCONFIG

ファイルの `DM_REMOTE` セクションの `ACL_POLICY` パラメータで設定します。

- Domains リンク・レベルの暗号化 通信するドメイン・ゲートウェイ間のデータ秘密性を確保します。TDomain ゲートウェイのリンク・レベルの暗号化は、`DMCONFIG` ファイルの `DM_TDOMAIN` セクションの `MINENCRYPTBITS` パラメータと `MAXENCRYPTBITS` パラメータを使用して設定します。

Domains 認証のプリンシパル名の設定

『BEA Tuxedo のセキュリティ機能』の [2-27 ページの「ドメイン間のリンクの確立」](#)で説明されているように、ローカルの TDomain ゲートウェイでは、ローカル・ドメインとリモート・ドメインの両方が知っており、その結果としてドメインが相互に接続しようとしたときにリモート・ドメインでローカル・ドメインを認証できる ID (プリンシパル名) が必要です。同じように、リモートの TDomain ゲートウェイでは、リモート・ドメインとローカル・ドメインの両方が知っており、その結果としてドメインが相互に接続しようとしたときにローカル・ドメインでリモート・ドメインを認証できる ID (プリンシパル名) が必要です。また、ローカル TDomain ゲートウェイではその割り当てられたプリンシパル名を使用して接続の設定時に必要なセキュリティ・クリデンシャルを取得します。

ローカル TDomain ゲートウェイでは、プリンシパル名がもう 1 つ必要です。その 2 つめのプリンシパル名は、[2-60 ページの「リモート・ドメインの ACL 方針の設定」](#)で説明されているローカルのアクセス制御リスト (ACL) 方針を適用するために必要なセキュリティ・クリデンシャルを取得するために使用します。

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、リリース 7.1 以降の BEA Tuxedo アプリケーションで動作する TDomain ゲートウェイのプリンシパル名を指定します。

- UBBCONFIG の SEC_PRINCIPAL_NAME (文字列)

BEA Tuxedo 7.1 以降が動作するアプリケーションで認証用に使用されるセキュリティ・プリンシパル名を指定します。このパラメータの最大文字数は、文字列の最後を表す NULL 文字列を除いて 511 文字です。このパラメータで指定されたプリンシパル名は、このアプリケーションで動作する 1 つ以上のシステム・プロセス (TDomain ゲートウェイ GWTDOMAIN プロセスを含む) の ID になります。

アプリケーションの起動時に、アプリケーション内のそれぞれの TDomain ゲートウェイ・プロセスは、認証プラグインを呼び出して、SEC_PRINCIPAL_NAME で指定されたセキュリティ・プリンシパル名のセキュリティ・クリデンシャルを取得します。TDomain ゲートウェイは、SEC_PRINCIPAL_NAME パラメータで指定されたプリンシパル名で、これらのクリデンシャルを取得します。

- DMCONFIG の DM_LOCAL セクションの CONNECTION_PRINCIPAL_NAME (文字列)

接続プリンシパル名識別子を指定します。このプリンシパル名は、リモート・ドメインに接続するときこのローカル・ドメイン・アクセス・ポイントと関連付けられたドメイン・ゲートウェイの ID を検証するために使用します。このパラメータは、BEA Tuxedo 7.1 以降が動作する TDOMAIN タイプのドメイン・ゲートウェイのみに適用されます。

CONNECTION_PRINCIPAL_NAME パラメータの最大文字数は、文字列の最後を表す NULL 文字列を除いて 511 文字です。このパラメータを指定しない場合は、ローカル・ドメイン・アクセス・ポイントの ACCESSPOINTID 文字列がデフォルト値になります。

デフォルトの認証プラグインで、このローカル・ドメイン・アクセス・ポイントの CONNECTION_PRINCIPAL_NAME に値を割り当てる場合、その値は、このローカル・ドメイン・アクセス・ポイントの ACCESSPOINTID パラメータの値と同じでなければなりません。これらの値が一致しないと、ローカル TDomain ゲートウェイ・プロセスが起動せず、次の userlog(3c) メッセージが生成されます。ERROR: 証明書を取得できません。

- DMCONFIG の DM_REMOTE セクションの CONNECTION_PRINCIPAL_NAME (文字列)

接続プリンシパル名識別子を指定します。このプリンシパル名は、ローカル・ドメインに接続するときこのリモート・ドメイン・アクセス・ポイントの ID を検証するために使用します。このパラメータは、BEA Tuxedo 7.1 以降が動作する TDOMAIN タイプのドメイン・ゲートウェイのみに適用されます。

CONNECTION_PRINCIPAL_NAME パラメータの最大文字数は、文字列の最後を表す NULL 文字列を除いて 511 文字です。このパラメータを指定しない場合は、リモート・ドメイン・アクセス・ポイントの ACCESSPOINTID 文字列がデフォルト値になります。

デフォルトの認証プラグインで、このリモート・ドメイン・アクセス・ポイントの CONNECTION_PRINCIPAL_NAME に値を割り当てる場合、その値は、このリモート・ドメイン・アクセス・ポイントの ACCESSPOINTID パラメータの値と同じでなければなりません。これらの値が一致しないと、ローカル TDomain ゲートウェイとリモート TDomain ゲートウェイを接続しようとしても失敗し、次の `userlog(3c)` メッセージが生成されません。ERROR: ドメイン `domain_name` の管理用キーを初期化できません。

次の例で、DMCONFIG ファイルの CONNECTION_PRINCIPAL_NAME ID は、LOCAL1 アクセス・ポイントおよび REMOT1 アクセス・ポイントを通じて接続を確立するときに使用されます。

```
*DM_LOCAL
LOCAL1  GWGRP=bankg1
        TYPE=TDOMAIN
        ACCESSPOINTID="BA.CENTRAL01"
        CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"

*DM_REMOTE
REMOT1  TYPE=TDOMAIN
        ACCESSPOINTID="BA.BANK01"
        CONNECTION_PRINCIPAL_NAME="BA.BANK01"
```

Domains パスワード・セキュリティの設定

ドメイン・ゲートウェイでは、リモート・ドメインから要求された受信時接続を認証できます。アプリケーション管理者は、リモート・ドメインからの受信時接続でいつセキュリティを適用するのかを定義できます。

管理者は、DMCONFIG ファイルの DM_LOCAL セクションの SECURITY パラメータを設定して、特定のローカル・ドメインで使用されるセキュリティのレベルを指定できます。パスワード・セキュリティには次の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) リモート・ドメインからの受信時接続が認証されません。
- アプリケーション・パスワード (APP_PW オプション) リモート・ドメインからの受信時接続が TUXCONFIG ファイルで定義されたアプリケーション・パスワードで認証されます。アプリケーション・パスワードは、UBBCONFIG ファイルでは定義されません。BEA Tuxedo アプリケーション・パスワードは、`tmloadcf(1)` で管理されます。このコマンドは、TUXCONFIG ファイルで SECURITY オプションが有効な場合にパスワードを要求します。パスワードは、TUXCONFIG ファイルと一緒に自動的にコンフィギュレーションのほかのマシンに複製転送されます。パスワードは、`tmadmin` コマンドを使用して動的に更新できます。
- Domains パスワード (DM_PW オプション) ローカル・ドメインとリモート・ドメインの接続は、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。DM_PASSWORDS セクションは、DMCONFIG ファイルでは定義されません。パスワードは、`dmloadcf` が実行された後に、DM_MIB(5) または `dmadmin(1)` コマンドの `passwd` サブコマンドを使用して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。各エントリでは、ローカル・ドメインへのアクセスでリモート・ドメインによって使用されるパスワードと、リモート・ドメインへのアクセスでローカル・ドメインによって使用されるパスワードが設定されます。

TUXCONFIG ファイルで SECURITY パラメータが NONE に設定されている場合や、パラメータが設定されていない場合でも、Domains コンフィギュレーションでは、TDomain ゲートウェイによる DM_PW レベルのセキュリティの適用が要求されることがあります。DM_PW オプションが選択されている場合、各リモート・ドメインではパスワードを BDMCONFIG ファイルの DM_PASSWORDS セクションで定義する必要があります。つまり、パスワードがないと、受信時接続は TDomain ゲートウェイによって拒否されます。

DM_MIB(5) を使用した Domains パスワード (DM_PW) の設定

DM_MIB を使用すると、Domains パスワード (DM_PW) を設定できます。DM_MIB の T_DM_PASSWORDS クラスは、TDOMAIN タイプのローカルおよびリモート・アクセス・ポイントを介したドメイン間認証のコンフィギュレーション情報を表します。T_DM_PASSWORDS クラスには、リモート・ドメインごとに次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 で識別) とリモート・ドメイン・アクセス・ポイント (TA_DMRACCESSPOINT で識別) の接続を認証するためのローカル・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 で識別) とローカル・ドメイン・アクセス・ポイント (TA_DMLACCESSPOINT で識別) の接続を認証するためのリモート・パスワード

注記 パスワードは、暗号化された形式で安全に格納されます。

MIB 管理要求のフォーマットと MIB 管理応答の解釈については、『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ [DM_MIB\(5\)](#) を参照してください。

dmadmin コマンドを使用した Domains パスワード (DM_PW) の設定

dmadmin コマンドを使用しても、Domains パスワード (DM_PW) を設定できません。

```
prompt> dmadmin
passwd [-r] local_domain_access_point_name
        remote_domain_access_point_name
```

dmadmin コマンドは、指定されたローカルおよびリモート・ドメイン・アクセス・ポイントの新しいパスワードを要求します。dmadmin(1)の詳細については、[4-1 ページの「Domains の管理」](#)を参照してください。

ドメイン間のパスワード・セキュリティのコーディング例

DMCONFIG ファイルの DM_LOCAL 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、接続は失敗します。

例 1 セキュリティを NONE に設定する

ローカル・ドメインで SECURITY が NONE に設定されている場合、受信時接続は認証されません。SECURITY が NONE に設定されているローカル・ドメインでも SECURITY が DM_PW に設定されているリモート・ドメインに接続できますが、そのような接続の確立前に、DM_MIB(5) または dmadmin passwd コマンドを使用して両サイドでパスワードを定義する必要があります。

コード リスト 2-13 アプリケーションと Domains の両方でセキュリティを NONE に設定する

```
LOCAL1: SECURITY in UBBCONFIG set to NONE
        SECURITY in DMCONFIG set to NONE

REMOT1: SECURITY in UBBCONFIG set to NONE
        SECURITY in DMCONFIG set to DM_PW
```

この例では、LOCAL1 ではセキュリティが適用されていませんが、REMOT1 では DM_PW セキュリティが適用されています。イニシエータ (LOCAL1) 側では、UBBCONFIG と DMCONFIG の関連属性は次のように設定されます。

```
UBBCONFIG
*RESOURCES
SECURITY NONE

DMCONFIG
*DM_LOCAL
LOCAL1    GWGRP=bankg1
          TYPE=TDOMAIN
          ACCESSPOINTID="BA.CENTRAL01"
          CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
          SECURITY=NONE

*DM_REMOTE
REMOT1    TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK01"
          CONNECTION_PRINCIPAL_NAME="BA.BANK01"
```

応答 (REMOT1) 側では、UBBCONFIG と DMCONFIG の関連属性は次のように設定されます。

```
UBBCONFIG
*RESOURCES
SECURITY NONE

DMCONFIG
*DM_LOCAL
REMOT1    GWGRP=bankg2
          TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK01"
          CONNECTION_PRINCIPAL_NAME="BA.BANK01"
          SECURITY=DM_PW

*DM_REMOTE
LOCAL1    TYPE=TDOMAIN
          ACCESSPOINTID="BA.CENTRAL01"
          CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
```

TUXCONFIG ファイルと BDMCONFIG ファイルで必須属性が設定された後に、LOCAL1 と REMOT1 でアプリケーションを起動します。

```
LOCAL1:
dmadmin
passwd LOCAL1 REMOT1
Enter Local Domain Password:fool
```

2 ATMI Domains の計画とコンフィギュレーション

```
Reenter Local Domain Password:foo1
Enter Remote Domain Password:foo2
Reenter Remote Domain Password:foo2
```

REMOT1:

```
dmadmin
passwd REMOT1 LOCAL1
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が作成されたら、接続を確立して、リモート・ドメインでサービスを呼び出すことができます。

コード リスト 2-14 アプリケーション・セキュリティを NONE、Domains セキュリティを DM_PW に設定する

イニシエータ (LOCAL1) 側では、UBBCONFIG と DMCONFIG の関連属性は次のように設定されます。

```
UBBCONFIG
*RESOURCES
SECURITY NONE

DMCONFIG
*DM_LOCAL
LOCAL1 GWGRP=bankg1
TYPE=TDOMAIN
ACCESSPOINTID="BA.CENTRAL01"
CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
SECURITY=DM_PW

*DM_REMOTE
REMOT1 TYPE=TDOMAIN
ACCESSPOINTID="BA.BANK01"
CONNECTION_PRINCIPAL_NAME="BA.BANK01"
```

応答 (REMOT1) 側では、UBBCONFIG と DMCONFIG の関連属性は次のように設定されます。

```
UBBCONFIG
*RESOURCES
SECURITY NONE

DMCONFIG
*DM_LOCAL
```

```
REMOT1    GWGRP=bankg2
          TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK01"
          CONNECTION_PRINCIPAL_NAME="BA.BANK01"
          SECURITY=DM_PW

*DM_REMOTE
LOCAL1    TYPE=TDOMAIN
          ACCESSPOINTID="BA.CENTRAL01"
          CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
```

TUXCONFIG ファイルと BDMCONFIG ファイルで必須属性が設定された後に、LOCAL1 と REMOT1 でアプリケーションを起動します。

```
LOCAL1:
dmadmin
passwd LOCAL1 REMOT1
Enter Local Domain Password:foo1
Reenter Local Domain Password:foo1
Enter Remote Domain Password:foo2
Reenter Remote Domain Password:foo2

REMOT1:
dmadmin
passwd REMOT1 LOCAL1
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が作成されたら、接続を確立して、リモート・ドメインでサービスを呼び出すことができます。

例 2 セキュリティを APP_PW に設定する

UBBCONFIG の SECURITY パラメータが APP_PW 以上に設定されている場合、DMCONFIG の SECURITY は NONE、APP_PW、または DM_PW に設定できます。1 つの DMCONFIG ファイルでドメインの複数のビューを定義できるので (ローカル・ドメインの定義ごとに 1 ビュー)、それらの各ビューに異なるタイプのセキュリティ・メカニズムを割り当てることができます。

注記 DMCONFIG でローカル・ドメイン・アクセス・ポイントの SECURITY が APP_PW に設定されている場合、UBBCONFIG の SECURITY は APP_PW 以上に設定する必要があります。

コード リスト 2-15 アプリケーションと Domains の両方でセキュリティを APP_PW に設定する

```
LOCAL1: SECURITY in UBBCONFIG set to APP_PW
        SECURITY in DMCONFIG set to APP_PW

REMOT1: SECURITY in UBBCONFIG set to APP_PW
        SECURITY in DMCONFIG set to APP_PW
```

この例では、LOCAL1 と REMOT1 の両方で APP_PW セキュリティが適用されます。

イニシエータ (LOCAL1) 側では、UBBCONFIG と DMCONFIG の関連属性は次のように設定されます。

```
UBBCONFIG
*RESOURCES
SECURITY APP_PW

DMCONFIG
*DM_LOCAL
LOCAL1    GWGRP=bankg1
          TYPE=TDOMAIN
          ACCESSPOINTID="BA.CENTRAL01"
          CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
          SECURITY=APP_PW

*DM_REMOTE
REMOT1    TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK01"
          CONNECTION_PRINCIPAL_NAME="BA.BANK01"
```

応答 (REMOT1) 側では、UBBCONFIG と DMCONFIG の関連属性は次のように設定されます。

```
UBBCONFIG
*RESOURCES
SECURITY APP_PW
```

```
DMCONFIG
*DM_LOCAL
REMOT1    GWGRP=bankg2
          TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK01"
          CONNECTION_PRINCIPAL_NAME="BA.BANK01"
          SECURITY=APP_PW

*DM_REMOTE
LOCAL1    TYPE=TDOMAIN
          ACCESSPOINTID="BA.CENTRAL01"
          CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
```

TUXCONFIG ファイルと BDMCONFIG ファイルが作成された後に、LOCAL1 と REMOT1 でアプリケーションを起動します。

Domains アクセス制御リストの設定

DMCONFIG ファイルの DM_ACCESS_CONTROL セクションで Domains アクセス制御リスト (ACL) を設定するには、ACL の名前とその ACL 名に関連付けられたリモート・ドメイン・アクセス・ポイントを指定します。その手順は次の表にまとめられています。

ドメイン ACL フィールド	説明
Domains ACL 名	この ACL の名前。 1 ~ 30 文字の文字列で構成された有効な名前。表示可能である必要があり、コロン、シャープ記号、または改行文字は使用できない。 例 : ACLGRP1
リモート・ドメイン・リスト	このアクセス制御リストでアクセス権が与えられるリモート・ドメインのリスト。 このフィールドの有効な値は、カンマで区切られた 1 つ以上のリモート・ドメイン名の集合。 例 : REMDOM1 , REMDOM2 , REMDOM3

ACL を作成したら、DMCONFIG ファイルの DM_EXPORT セクションの ACL パラメータを使用して、特定のローカル・ドメイン・アクセス・ポイントを通じ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 ACL 名と関連付けられたリモート・ドメイン・アクセス・ポイントのみに制限します (例: ACL=ACLGRP1)。

リモート・ドメインの ACL 方針の設定

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 以降が動作するリモート・ドメインの ACL 方針を設定および制御します。これらのパラメータは、DMCONFIG ファイルの DM_REMOTE セクションで設定します。

■ ACL_POLICY (LOCAL | GLOBAL)

このリモート・ドメイン・アクセス・ポイントのアクセス制御リスト (ACL) 方針を指定します。このパラメータは、BEA Tuxedo 7.1 以降が動作するタイプ TDOMAIN のドメイン・ゲートウェイおよび BEA Tuxedo 8.0 以降が動作するタイプ OSITPX のドメイン・ゲートウェイのみに適用されます。

LOCAL の場合は、リモート・ドメインから受信したサービス要求のクレデンシャル (ID) が、ローカル・ドメインによって、このリモート・ドメイン・アクセス・ポイントの LOCAL_PRINCIPAL_NAME パラメータで指定されたプリンシパル名に置換されます。GLOBAL の場合、リモート・サービス要求と一緒に受信したクレデンシャルはローカル・ドメインによって置換されません。リモート・サービス要求と一緒にクレデンシャルを受信していない場合、ローカル・ドメインはそのサービス要求をローカル・サービスにそのまま転送します (通常は失敗する)。デフォルトは LOCAL です。

■ LOCAL_PRINCIPAL_NAME (文字列)

このリモート・ドメイン・アクセス・ポイントの ACL_POLICY パラメータが LOCAL に設定されているか、デフォルトのままの場合に、リモート・ドメインから受信したサービス要求にローカル・ドメインが割り当てるローカル・プリンシパル名識別子 (クレデンシャル)。このパラメータは、BEA Tuxedo 7.1 以降が動作するタイプ TDOMAIN のドメイン・ゲート

ウェイおよび BEA Tuxedo 8.0 以降が動作するタイプ `OSITPX` のドメイン・ゲートウェイのみに適用されます。

`LOCAL_PRINCIPAL_NAME` パラメータの最大文字数は、文字列の最後を表す `NULL` 文字列を除いて 511 文字です。このパラメータを指定しない場合は、リモート・ドメイン・アクセス・ポイントの `ACCESSPOINTID` 文字列がデフォルト値になります。

■ `CREDENTIAL_POLICY` (`LOCAL` | `GLOBAL`)

このリモート・ドメイン・アクセス・ポイントのクリデンシャル方針を指定します。このパラメータは、BEA Tuxedo 8.0 以降が動作するタイプ `TDOMAIN` のドメイン・ゲートウェイのみに適用されます。

`LOCAL` の場合は、このリモート・ドメイン・アクセス・ポイントに対するローカル・サービス要求からクリデンシャル (ID) がローカル・ドメインによって削除されます。`GLOBAL` の場合、このリモート・ドメイン・アクセス・ポイントに対するローカル・サービス要求からクリデンシャルは削除されません。デフォルトは `LOCAL` です。

`CREDENTIAL_POLICY` パラメータは、リモート・ドメインに送信する前にローカル・ドメインがローカル・サービス要求からクリデンシャルを削除するかどうかを指定します。`ACL_POLICY` パラメータは、リモート・ドメインから受信したサービス要求のクリデンシャルが、`LOCAL_PRINCIPAL_NAME` パラメータで指定されたプリンシパル名に置換されるかどうかを指定します。

次の例では、`REMOT1` アクセス・ポイントの接続に対し、`DMCONFIG` ファイルで `ACL` がグローバルに設定されています。つまり、`LOCAL1` アクセス・ポイントのドメイン・ゲートウェイは `REMOT1` アクセス・ポイントからのクライアント要求を変更なしで受け渡します。`ACL` がグローバルに設定されている場合、`REMOT1` アクセス・ポイントの `LOCAL_PRINCIPAL_NAME` エントリは無視されます。また、`CREDENTIAL_POLICY=GLOBAL` であるため、`LOCAL1` アクセス・ポイントのドメイン・ゲートウェイは `REMOT1` アクセス・ポイントに対するローカル・サービス要求からクリデンシャルを削除しません。

```
*DM_LOCAL
LOCAL1  GWGRP=bankg1
        TYPE=TDOMAIN
        ACCESSPOINTID="BA.CENTRAL01"
        CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
        SECURITY=DM_PW

*DM_REMOTE
REMOT1  TYPE=TDOMAIN
        ACCESSPOINTID="BA.BANK01"
        CONNECTION_PRINCIPAL_NAME="BA.BANK01"
        ACL_POLICY=GLOBAL
        CREDENTIAL_POLICY=GLOBAL
        LOCAL_PRINCIPAL_NAME="BA.BANK01.BOB"
```

Domains リンク・レベル暗号化の設定

Domains リンク・レベルの暗号化 (LLE) は、ローカル・ドメイン・ゲートウェイをリモート・ドメイン・ゲートウェイに接続するネットワーク・リンク上で送受信されるメッセージの秘密性を保護します。リンク・レベルの暗号化によるセキュリティには、0 ビット (暗号化なし)、56 ビット (国際版)、128 ビット (米国およびカナダ版) の 3 つのレベルがあります。

ドメイン・ゲートウェイのリンクで Domains リンク・レベルの暗号化を設定するには、次の手順に従います。

1. テキスト・エディタで DMCONFIG ファイルを開き、DM_TDOMAIN セクションに次の行を追加します。

```
*DM_TDOMAIN
LOCAL1  NWADDR="newyork.acme.com:65431"
        MINENCRYPTBITS=min
        MAXENCRYPTBITS=max

REMOT1  NWADDR="albany.acme.com:4051"
        MINENCRYPTBITS=min
        MAXENCRYPTBITS=max
```

2. `dmloadcf(1)` を実行してコンフィギュレーションをロードします。
`dmloadcf` コマンドを実行すると、DMCONFIG が解析され、BDMCONFIG 変数が指す場所にバイナリ形式の BDMCONFIG ファイルがロードされます。

上の例では、`tmboot(1)` を実行するとアプリケーションが起動します。各ドメイン・ゲートウェイは `BDMCONFIG` ファイルを読み込んで `MINENCRYPTBITS` および `MAXENCRYPTBITS` などのさまざまなパラメータにアクセスし、ローカル・ドメインおよびリモート・ドメインにそれらのパラメータを複製転送します。ローカル・ドメインがリモート・ドメインとのネットワーク・リンクを確立するとき、これらの2つのドメインは、双方でサポートされるキーの最大サイズが一致するまでキー・サイズの調整を行います。

Domains コンフィギュレーションの接続の設定

ローカル・ドメイン・ゲートウェイがリモート・ドメインに対して接続を試行するときの条件を指定できます。条件を指定するには、`DMCONFIG` ファイルの `DM_LOCAL` セクションの `CONNECTION_POLICY` パラメータに値を割り当てます。次の接続方針のいずれかを選択できます。

- ローカル・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`)。
- ブート時に接続する (`ON_STARTUP`)。
- 受信時に接続するが、自動的には行わない (`INCOMING_ONLY`)。

BEA Tuxedo リリース 8.1 以降では、`DMCONFIG` ファイルの `DM_TDOMAIN` セクションでリモート・ドメイン単位で接続方針を定義することもできます。詳細については、[1-37 ページの「接続方針の設定方法」](#)を参照してください。

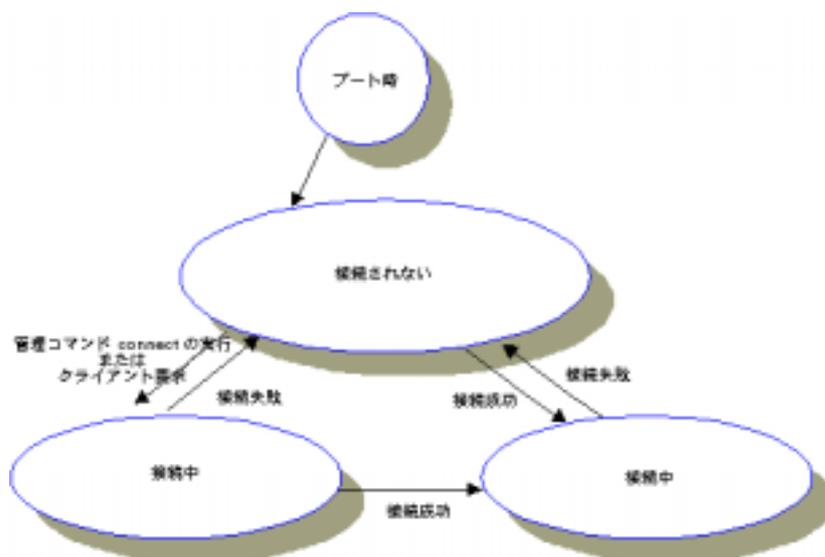
`ON_STARTUP` および `INCOMING_ONLY` の接続方針では、Dynamic Status が呼び出されます。Dynamic Status ([1-40 ページの「接続方針によるリモート・サービスの可用性の判断」](#)を参照) は、リモート・サービスの状態を調べて報告する BEA Tuxedo Domains の機能です。

クライアントの要求に応じた接続の要求 (ON_DEMAND 方針)

接続方針が ON_DEMAND (CONNECTION_POLICY=ON_DEMAND) の場合は、ローカル・クライアントがリモート・サービスを要求したか、管理コマンド `dmadmin connect` が実行されたときにだけ接続が試行されます。ON_DEMAND は、接続方針のデフォルト設定です。

次の図は、接続方針が ON_DEMAND であるドメイン・ゲートウェイによってどのように接続が試行されて確立されるのかを示しています。

図 2-5ON_DEMAND 方針で行われる接続

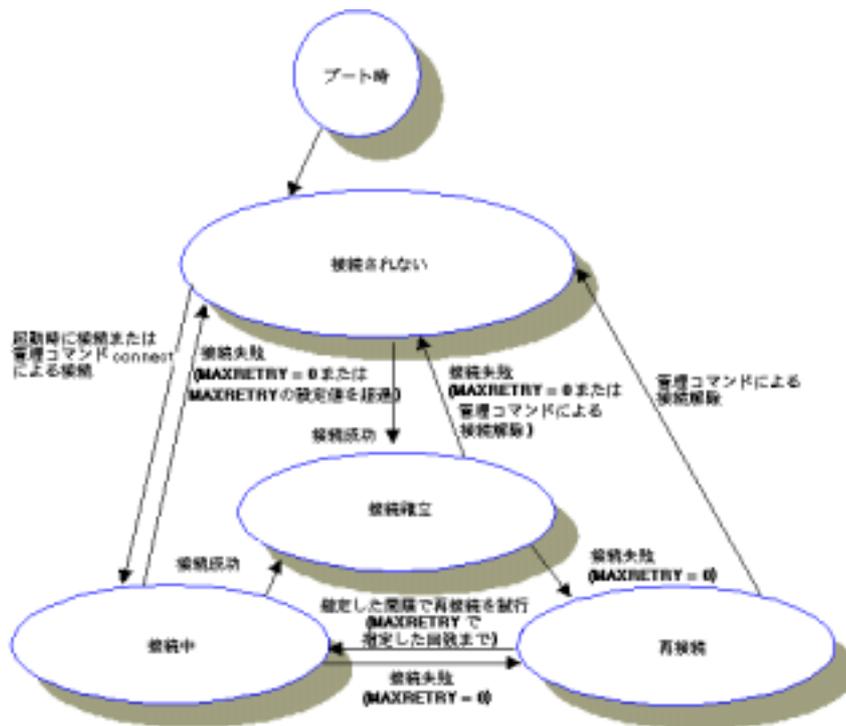


起動時の接続の要求 (ON_STARTUP 方針)

接続方針が ON_STARTUP (CONNECTION_POLICY=ON_STARTUP) の場合、ドメイン・ゲートウェイはドメイン・ゲートウェイ・サーバが初期化されたときにリモート・ドメインとの接続を試行します。デフォルトの ON_STARTUP 接続方針では、60 秒おきに失敗した接続が再試行されますが、この間隔に別の値を指定することもできます (2-66 ページの「ON_STARTUP のみでの接続再試行間隔の設定」を参照)。

次の図は、接続方針が ON_STARTUP であるドメイン・ゲートウェイによってどのように接続が試行されて確立されるのかを示しています。

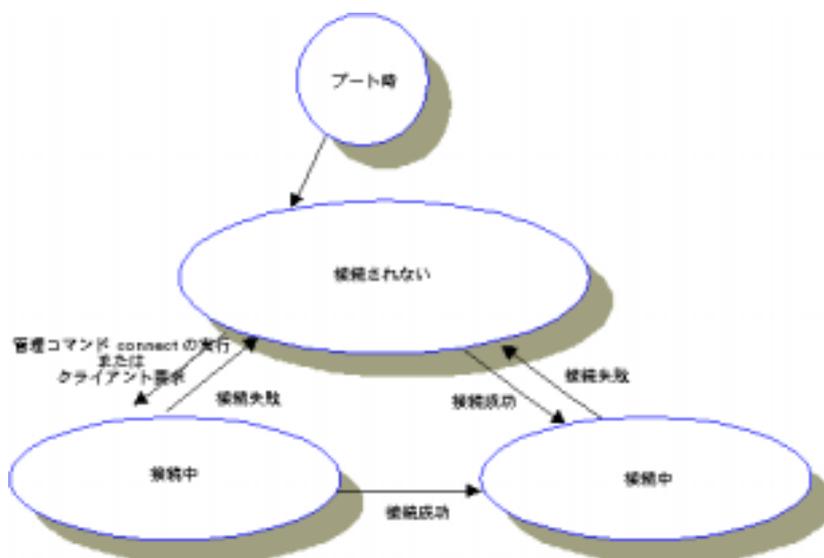
図 2-6ON_STARTUP 方針で行われる接続



接続を受信メッセージのみに制限する (INCOMING_ONLY 方針)

接続方針が INCOMING_ONLY (CONNECTION_POLICY=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行しません。次の図は、接続方針が INCOMING_ONLY であるドメイン・ゲートウェイによってどのように接続が試行されて確立されるのかを示しています。

図 2-7 INCOMING_ONLY 方針 (受信時接続のみ受け付ける) で行われる接続



ON_STARTUP のみでの接続再試行間隔の設定

CONNECTION_POLICY パラメータが ON_STARTUP に設定されている場合は、自動接続再試行プロセスを利用できます。接続再試行プロセスは、リモート・ドメインへの失敗した接続をドメイン・ゲートウェイで自動的に再試行できるようにします。管理者は、自動的な接続の試行頻度を管理できます。その

ためには、ゲートウェイが接続の確立を再び試みるまでの間隔 (秒単位) を指定します。再試行間隔は、次のように `RETRY_INTERVAL` パラメータを設定して指定できます。

```
RETRY_INTERVAL=number_of_seconds
```

0 ~ 2147483647 秒の値を指定できます。接続方針が `ON_STARTUP` で、`RETRY_INTERVAL` パラメータの値が指定されていない場合は、デフォルトの 60 秒が使用されます。

`RETRY_INTERVAL` パラメータは、接続方針が `ON_STARTUP` の場合のみ有効です。ほかの接続方針 (`ON_DEMAND` および `INCOMING_ONLY`) の場合、接続再試行プロセスは利用できません。

最大再試行回数の設定

ドメイン・ゲートウェイがリモート・ドメインへの接続を試行する最大の回数を指定するには、`MAXRETRY` パラメータに値を割り当てます。このパラメータの最小値は 0 で、デフォルトである最大値は `MAXLONG` パラメータの値 (2147483647) です。

- `MAXRETRY=0` のように設定すると、接続再試行プロセスは無効になります。ローカル・ドメイン・ゲートウェイは、リモート・ドメイン・ゲートウェイに自動的に接続しません。
- `MAXRETRY=number` を設定すると、ゲートウェイは指定された回数だけ接続の確立を試みます。
- `MAXRETRY=MAXLONG` を設定すると (デフォルト)、接続再試行プロセスは接続が確立されるまで繰り返されます (最高で 2147483647 回)。

`MAXRETRY` パラメータは、接続方針が `ON_STARTUP` の場合のみ有効です。ほかの接続方針 (`ON_DEMAND` および `INCOMING_ONLY`) の場合、接続再試行プロセスは利用できません。

次の表は、`MAXRETRY` および `RETRY_INTERVAL` がどのように自動接続再試行プロセスに影響するのかを示しています。

表 0-1MAXRETRY および RETRY_INTERVAL パラメータの設定例

設定	結果
CONNECTION_POLICY=ON_STARTUP MAXRETRY=3 RETRY_INTERVAL=30	ローカル・ドメイン・ゲートウェイは 30 秒間隔で最大 3 回接続を試みます。
CONNECTION_POLICY=ON_STARTUP MAXRETRY=0	ローカル・ドメイン・ゲートウェイは初期化時に接続を試行しますが、最初の試行が失敗した場合でも再試行はしません。
CONNECTION_POLICY=ON_STARTUP RETRY_INTERVAL=30	ドメイン・ゲートウェイは、接続が確立されるまで 30 秒間隔で接続を試行します。

ドメイン間の接続方針のコーディング例

Domains コンフィギュレーションにかかわるドメインは互いに独立して機能するので、接続方針のどのような組み合わせでも Domains コンフィギュレーションで使用できます。ただし、接続方針はどのような組み合わせでも実際に使えるわけではありません。たとえば、ほとんどの場合では、相互運用される 2 つのドメインそれぞれで接続方針を ON_STARTUP に設定してもあまり意味はありません。

次のコンフィギュレーション例は、接続方針の実際的な組み合わせです。この例で、LOCAL1 はローカルの DMCONFIG ファイルで ON_STARTUP に設定されていて、REMOT1 はリモートの DMCONFIG ファイルで INCOMING_ONLY に設定されています。

ローカルの DMCONFIG ファイル :

```
*DM_LOCAL
LOCAL1    GWGRP=bankg1
          TYPE=TDOMAIN
          CONNECTION_POLICY=ON_STARTUP
          MAXRETRY=5
          RETRY_INTERVAL=100

*DM_REMOTE
REMOT1    TYPE=TDOMAIN
          ACCESSPOINTID="BA.BANK01"
```

リモートの DMCONFIG ファイル :

```
*DM_LOCAL
REMOT1 GWGRP=bankg2
      TYPE=TDOMAIN
      ACCESSPOINTID="BA.BANK01"
      CONNECTION_POLICY=INCOMING_ONLY

*DM_REMOTE
LOCAL1 TYPE=TDOMAIN
      ACCESSPOINTID="BA.CENTRAL01"
      CONNECTION_PRINCIPAL_NAME="BA.CENTRAL01"
```

Domains コンフィギュレーションの接続の制御

管理者は、ドメイン間で確立する必要のあ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・ドメインの接続を切断することもできます。

ドメイン間の接続の確立

ローカル・ドメイン・ゲートウェイとリモート・ドメインの接続を確立するには、次のように、connect (co) サブコマンドを使用して dmadmin コマンドを実行します。

```
prompt> dmadmin co -d local_domain_access_point_name
```

デフォルトでは、接続は指定されたローカル・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・ドメインの間で確立されます。1つのリモート・ドメインのみと接続を確立する場合は、次のように、-R オプションを使用したコマンド行でそのドメインを指定します。

```
prompt> dmadmin co -d local_domain_access_point_name
      -R remote_domain_access_point_name
```

接続方針が ON_STARTUP、接続再試行プロセスが有効な状態で接続が失敗した場合は、接続再試行プロセスを通じて接続が繰り返し試行されます。

ドメイン間の接続の切断

ゲートウェイが自動接続再試行プロセスを通じて接続を再確立しない状態で、ローカル・ゲートウェイとリモート・ドメインの接続を切断するには、次のように、`disconnect (dco)` サブコマンドを使用して `dmadmin` コマンドを実行します。

```
prompt> dmadmin dco -d local_domain_access_point_name
```

デフォルトでは、ローカル・ドメインに対して設定されたすべてのリモート・ドメインの接続が切断されます。1つのリモート・ドメインのみと接続を切断する場合は、次のように、`-R` オプションを使用したコマンド行でそのドメインを指定します。

```
prompt> dmadmin dco -d local_domain_access_point_name  
-R remote_domain_access_point_name
```

自動接続再試行プロセスは、コマンドの実行時にアクティブな接続があるかどうかに関係なくこのコマンドで停止されます。

接続状態の報告

`dmadmin printdomain` コマンドを使用すると、接続状態および再試行されている接続についてレポートを生成できます。`connect` コマンドは、接続が成功したかどうかを報告します。`printdomain` コマンドは、指定されたローカル・ドメインについての情報を出力します。その情報には、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、接続しようとしているリモート・ドメインのリストが含まれます。

次の例は、`printdomain` コマンドが（その短縮形 `pd` で）`LOCAL1` というローカル・ドメイン・アクセス・ポイントに対して発行される `dmadmin` セッションを示しています。

```
prompt> dmadmin  
dmadmin - Copyright ...  
.  
.  
.  
pd -d LOCAL1  
Local domain :LOCAL1
```

```
Connected domains:
Domainid: REMOT1
Disconnected domains being retried:
Domainid: REMOT2

dco -d LOCAL1 -R REMOT1
Operation completed successfully. Use printdomain(pd) to obtain
results.

dco -d LOCAL1 -R REMOT2
Operation completed successfully. Use printdomain(pd) to obtain
results.

co -d LOCAL1 -R REMOT1
Operation completed successfully. Use printdomain(pd) to obtain
results.

pd -d LOCAL1
Local domain :LOCAL1
  Connected domains:
    Domainid: REMOT1
```

この例では、例自体がシンプルになるように、リモート・ドメイン・アクセス・ポイントの名前 (REMOT1、REMOT2) とそれらの DOMAINID (ACCESSPOINTID) 名 (REMOT1、REMOT2) が DMCONFIG ファイルの DM_REMOTE セクションで定義されているとおりに同じになっています。

Domains リンク・レベルのフェイルオーバーとキープアライブの設定

Domains のリンク・レベルのフェイルオーバーは、一次リンクが失敗したときに代替ネットワーク・リンクが有効になるようにするメカニズムです。Domains キープアライブは、ファイアウォールをまたがるドメイン間接続を活動のない期間オープンに維持するとともに、接続の障害を迅速に検出できるようにするメカニズムです。Domains キープアライブは、BEA Tuxedo リリース 8.1 以降で利用できます。

Domains リンク・レベルのフェイルオーバーについては、1-43 ページの「[Domains リンク・レベルのフェイルオーバーの設定](#)」を参照してください。
Domains キープアライブについては、1-44 ページの「[Domains のキープアライブの指定](#)」を参照してください。

3 CORBA Domains の 計画とコンフィギュ レーション

以下の節では、BEA Tuxedo CORBA Domains 環境のドメインの計画および設定の方法について説明します。

- [CORBA Domains 環境の概要](#)
- [シングル・ドメイン通信とマルチ・ドメイン通信](#)
- [CORBA Domains コンフィギュレーションの要素](#)
- [コンフィギュレーション・ファイルの使い方](#)
- [factory_finder.ini ファイルでの一意なファクトリ・オブジェクト識別子の指定](#)
- [factory_finder.ini ファイルの処理](#)
- [CORBA Domains・コンフィギュレーションの種類](#)
- [CORBA Domains・コンフィギュレーションの例](#)

CORBA Domains 環境の概要

BEA Tuxedo Domains コンフィギュレーションは、1-2 ページの「[BEA Tuxedo の Domains コンポーネントとは](#)」で説明されているようにコア ATMI ドメイン環境を拡張したものです。BEA Tuxedo ドメイン（ビジネス・アプリケーション）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。ドメインを意識するのは管理者のみです。

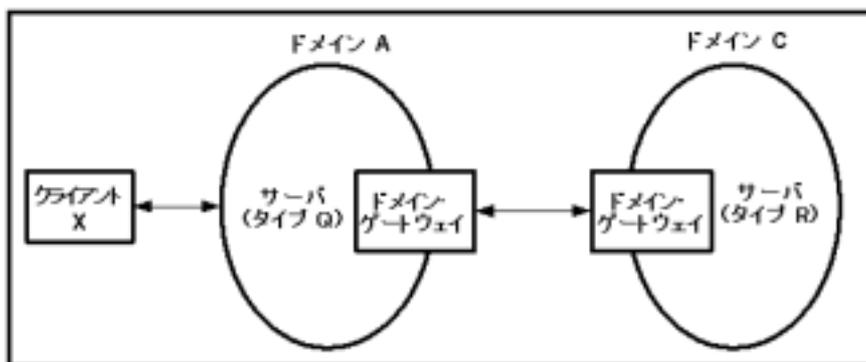
BEA Tuxedo Domains コンフィギュレーションで、管理者はドメインのどのサービスがコンフィギュレーション内のほかのドメインから利用できるのかを設定できます。このため、CORBA の観点から見ると、BEA Tuxedo Domains コンポーネントは BEA Tuxedo CORBA アプリケーションが相互に作用してリソースを共有するための手段であると言えます。CORBA クライアントと参加アプリケーションは、Domains コンフィギュレーションについて何も意識する必要はありません。知っておく必要があるのは、どのファクトリ・オブジェクトが利用可能で、どうしたらそのオブジェクトにアクセスできるのかということだけです。

ドメイン間のこの透過性により、管理者は各ドメインのサービスを設定し、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。

シングル・ドメイン通信とマルチ・ドメイン通信

次の図は、2 つの BEA Tuxedo CORBA アプリケーションで構成されたシングルな Domains コンフィギュレーションを示しています。

図 3-12 つの CORBA アプリケーションから成る Domains コンフィギュレーション



以降のシングル・ドメインとマルチ・ドメインに関する説明は、このシンプルな Domains コンフィギュレーションに基づいています。

シングル・ドメイン通信

以下は、シンプル Domains コンフィギュレーションにおける CORBA 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。

この手順を通じて、クライアントはオブジェクトがどこにあるのか、つまりどのドメインにあるのかについて何も知りません。

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

マルチ・ドメイン通信の場合、シンプル Domains コンフィギュレーションの 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。

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。シンプル Domains コンフィギュレーションに示すように、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。

CORBA Domains コンフィギュレーションの要素

BEA Tuxedo CORBA Domains ・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 ファイルと呼ばれるこのテキスト・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、次に説明する Domains コンフィギュレーション・ファイルで指定します。

BEA Tuxedo のコンフィギュレーション・ファイルには、どのような名前でも付けられます。ただし、そのファイルの内容は『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ `UBBCONFIG(5)` で説明されている形式に準拠している必要があります。

■ Domains コンフィギュレーション・ファイル

DMCONFIG ファイルと呼ばれるこのテキスト・ファイルでは、このドメイン(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。Domains コンフィギュレーションに参加するドメインごとに、1つの DMCONFIG ファイルが必要です。ドメインが別のドメインに接続しない場合、DMCONFIG ファイルは必要ありません。

Domains コンフィギュレーション・ファイルには、どのような名前でも付けられます。ただし、そのファイルの内容は『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ `DMCONFIG(5)` で説明されている形式に準拠している必要があります。

■ FactoryFinder Domains コンフィギュレーション・ファイル

factory_finder.ini と呼ばれるこのテキスト・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。CORBA Domains コンフィギュレーションに参加するドメインごとに、1つの factory_finder.ini ファイルが必要です。ドメインが別のドメインに接続しない場合、factory_finder.ini ファイルは必要ありません。

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。

FactoryFinder Domains コンフィギュレーション・ファイルには、どのような名前でも付けられます。ただし、そのファイルの内容は『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ factory_finder.ini(5) で説明されている形式に準拠している必要があります。

■ リモート・ドメインでのオブジェクトの呼び出し

CORBA の観点から見ると、BEA Tuxedo Domains コンポーネントの中心的機能は、CORBA ドメインのアプリケーションが、別の CORBA 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

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。

■ 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、クライアントに戻したり、別のドメインに渡すことができます。

コンフィギュレーション・ファイルの使い方

CORBA Domains コンフィギュレーションの設定には、次の 3 つのコンフィギュレーション・ファイルを使用します。

- UBBCONFIG BEA Tuxedo コンフィギュレーション・ファイル
- DMCONFIG - Domains コンフィギュレーション・ファイル
- factory_finder.ini FactoryFinder Domains コンフィギュレーション・ファイル

CORBA Domains コンフィギュレーションの各ドメインでは、これら 3 つのファイルのセットが必要です。管理者は、コンフィギュレーション・ファイルの各セット内、およびコンフィギュレーション・ファイルのセット間でコンフィギュレーションを調整する必要があります。Domains コンフィギュレーションでドメインの数が増えると、コンフィギュレーションを調整する手間も増えることになります。

UBBCONFIG 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・サーバ

ドメイン名

シングル BEA Tuxedo 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、DOMAINID 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。

```
DOMAINID 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。

```
DOMAINID headquarters
```

domain_name は、関連する DMCONFIG ファイルの DM_EXPORT および DM_IMPORT 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"
```

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。

ゲートウェイのグループとサーバ名

ほかのすべての BEA Tuxedo システム・サーバと同じように、ゲートウェイのグループとサーバ名を指定する必要があります。たとえば、UBBCONFIG ファイルの GROUPS セクションで次のように指定します。

```
LGWGRP      LMID=LDOM      GRPNO=4
```

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）。

BEA Tuxedo ドメイン・ゲートウェイ (TDomain ゲートウェイ) のサーバ名は 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。GWTDOMAIN 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
```

このエントリにより、TDomain 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 アプリケーションに通知されます。

DMCONFIG ファイル

Domains コンフィギュレーションに参加する各 BEA Tuxedo ドメインでは、それ専用の DMCONFIG ファイルが必要です。DMCONFIG ファイルでは、ローカル・ドメイン (DMCONFIG ファイルのあるドメイン) と 1 つ以上のリモート・ドメイン (ローカル・ドメインが相互運用されるドメイン) の関係を指定します。

DMCONFIG ファイルのほとんどのドキュメントでは、サービスを共有するための BEA Tuxedo ATMI ドメインの設定が中心的に説明されていますが、それは BEA Tuxedo CORBA 環境には適当ではありません。BEA Tuxedo CORBA Domains 環境では、「サービス」は BEA Tuxedo CORBA 要求にサービスできる BEA Tuxedo ドメインの名前でしかありません。

3 CORBA Domains の計画とコンフィギュレーション

CORBA Domains 環境には、DMCONFIG ファイルの次の 7 セクションが適用されます。

- DM_LOCAL (DM_LOCAL_DOMAINS と呼ばれる)
- DM_REMOTE (DM_REMOTE_DOMAINS と呼ばれる)
- DM_EXPORT (DM_LOCAL_SERVICES と呼ばれる)
- DM_IMPORT (DM_REMOTE_SERVICES と呼ばれる)
- DM_RESOURCES
- DM_ACCESS_CONTROL
- DM_TDOMAIN

注記 DM_LOCAL セクションは、DM_REMOTE セクションの前に置く必要があります。

これら 7 つのセクションの多くのパラメータは ATMI 用のパラメータなので、CORBA Domains 環境の設定には関係ありません。

以降の説明は、次のリストで示されているサンプル DCONFIG ファイルに基づいています。

コード リスト 3-1 BEA Tuxedo CORBA Domains 環境のサンプル
DMCONFIG ファイル

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

*DM_LOCAL
LDM    GWGRP=LWGRP  TYPE=TDOMAIN  ACCESSPOINTID="MUTT"

*DM_REMOTE
TDM1   TYPE=TDOMAIN  ACCESSPOINTID="JEFF"

*DM_EXPORT
"//MUTT"

*DM_IMPORT
"//JEFF"    RACCESSPOINT=TDM1

*DM_TDOMAIN
LDM    NWADDR="//sanfran.kmart.com:2507"
TDM1   NWADDR="//sanhose.kmart.com:3186"
```

注記 このリストの ACCESSPOINTID パラメータは DOMAINID パラメータで置換でき、RACCESSPOINT パラメータは RDOM パラメータで置換できます。このリストでは、新しい DMCNFIG 用語が使われています。

DM_RESOURCES

DM_RESOURCES セクションでは、グローバルの Domains コンフィギュレーション情報、具体的にはユーザ指定のコンフィギュレーション・バージョン文字列を指定します。このセクションでは、VERSION=*string* が唯一のパラメータです。*string* は、現在の DMCONFIG ファイルのバージョン番号を入力できるフィールドです。このフィールドは、ソフトウェアのチェックを受けません。

サンプルの DMCONFIG ファイルでは、VERSION パラメータが次のように Experimental8.9 に設定されています。

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

DM_LOCAL

DM_LOCAL セクション (DM_LOCAL_DOMAINS セクション) では、1 つ以上のローカル・ドメイン・アクセス・ポイント (論理名) を定義します。定義した各ローカル・ドメイン・アクセス・ポイントについて、このセクションでアクセス・ポイントのドメイン・ゲートウェイ・グループ (TDOMAIN など) を指定します。また、CORBA 環境用に、アクセス・ポイントを通じてアクセスできるローカル BEA Tuxedo ドメインの *domain_name* を DM_EXPORT セクションで指定します。ローカル・ドメインは、1 つ以上のリモート BEA Tuxedo ドメインの CORBA クライアントからローカル・ドメイン・アクセス・ポイントを通じてアクセス可能になります。

DM_LOCAL セクションでは、UBBCONFIG ファイルで定義されたドメイン・ゲートウェイ・グループごとに 1 つだけ必ずエントリが必要です。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。

DM_LOCAL セクションのエントリの形式は次のとおりです。

```
LocalAccessPoint required_parameters [optional_parameters]
```

LocalAccessPoint は、UBBCONFIG ファイルで定義されたゲートウェイ・グループを表すために選択したローカル・ドメイン・アクセス・ポイント識別子 (論理名) です。ローカル・ドメイン・アクセス・ポイントは、UBBCONFIG ファイルで指定された *domain_name* またはゲートウェイ・グループと同じ名

前ではありません。ローカル・ドメイン・アクセス・ポイントは `DMCONFIG` ファイル内でのみ使用される名前であり、`UBBCONFIG` ファイルが変更された場合の影響を限定します。`UBBCONFIG` の変更はローカル・ドメイン・アクセス・ポイントに定義されたパラメータのみに影響し、`DMCONFIG` ファイルを通じて使用されるローカル・ドメイン・アクセス・ポイントの論理名には影響しません。

以下は、必須パラメータです。

`GWGRP = identifier`

このパラメータでは、このローカル・ドメイン・アクセス・ポイントに関連付けられたドメイン・ゲートウェイ・サーバ・グループの名前 (`UBBCONFIG` ファイルの `GROUPS` セクションで指定された名前) を指定します。

`TYPE = TDOMAIN`

`TYPE` パラメータは、BEA Tuxedo CORBA 環境の `TDomain` ゲートウェイを使用することを指定するために必要です。

`ACCESSPOINTID = string`

`ACCESSPOINTID (DOMAINID)` パラメータは、リモート・ドメインとの接続を設定する際のセキュリティを目的として、このローカル・ドメイン・アクセス・ポイントと関連付けられたゲートウェイ・グループを識別するために使用します。`GWGRP` パラメータで指定されたゲートウェイ・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。`ACCESSPOINTID` 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。`ACCESSPOINTID` 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 `NULL` を含む)。

たとえば、次の例を考えます。

```
*DM_LOCAL
LDM GWGRP=LGWGRP TYPE=TDOMAIN ACCESSPOINTID="MUTT"
```

この例では、LDM が、サーバ・グループ名 LGWGRP (UBBCONFIG で指定) のローカル TDomain ゲートウェイ・グループと関連付けられたローカル・ドメイン・アクセス・ポイントとして識別されます。ドメイン・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。

注記 ドメイン・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で、かつローカル・ドメイン・アクセス・ポイントに CONNECTION_PRINCIPAL_NAME パラメータが指定されている場合、ゲートウェイはそのパラメータで指定された名前で通過します。

DM_LOCAL セクションのオプション・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。以上のパラメータの詳細については、『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ DMCONFIG(5) を参照してください。

DM_REMOTE

DM_REMOTE セクション (DM_REMOTE_DOMAINS セクション) では、1 つ以上のリモート・ドメイン・アクセス・ポイント (論理名) を定義します。定義した各リモート・ドメイン・アクセス・ポイントについて、このセクションでアクセス・ポイントのドメイン・ゲートウェイ・グループ (TDOMAIN など) を指定します。また、CORBA 環境用に、アクセス・ポイントを通じてアクセスできるリモート BEA Tuxedo ドメインの *domain_name* を DM_IMPORT セクションで指定します。リモート・ドメインは、ローカル・ドメインの CORBA クライアントからリモート・ドメイン・アクセス・ポイントを通じてアクセス可能になります。

このセクションでは、複数のリモート・ドメイン・アクセス・ポイントを定義できます。リモート・ドメインと通信するためにこの BEA Tuxedo ドメインで使用される各ドメイン・ゲートウェイ・グループについて 1 つ以上を定義できます。

DM_REMOTE セクションのエントリの形式は次のとおりです。

RemoteAccessPoint required_parameters

RemoteAccessPoint は、UBBCONFIG ファイルで定義された特定のゲートウェイ・グループから特定のリモート・ドメインにアクセスできるようにするために選択したリモート・ドメイン・アクセス・ポイント識別子 (論理名) で

す。リモート・ドメイン・アクセス・ポイントは、ローカル・ドメインまたはリモート・ドメインの UBBCONFIG ファイルで指定された *domain_name* またはゲートウェイ・グループと同じ名前ではありません。リモート・ドメイン・アクセス・ポイントは DMCONFIG 内でのみ使用される名前であり、UBBCONFIG が変更された場合の影響を限定します。UBBCONFIG の変更はリモート・ドメイン・アクセス・ポイントに定義されたパラメータのみに影響し、DMCONFIG ファイルを通じて使用されるリモート・ドメイン・アクセス・ポイントの論理名には影響しません。

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 TDomain ゲートウェイを使用することを指定するために必要です。

ACCESSPOINTID = *string*

ACCESSPOINTID (DOMAINID) パラメータは、リモート・ドメインとの接続を設定する際のセキュリティを目的として、このリモート・ドメイン・アクセス・ポイントと関連付けられたリモート・ドメインを識別するために使用します。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。ACCESSPOINTID 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。ACCESSPOINTID 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.BANK01" など) か、0x で始まる 16 進数 ("0x0002FF98C0000B9D6" など) を指定できます。ACCESSPOINTID 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 NULL を含む)。

たとえば、次の例を考えます。

```
*DM_REMOTE
TDOM1 TYPE=TDOMAIN ACCESSPOINTID="JEFF"
```

この例では、TDOM1 が、ローカル TDomain ゲートウェイ・グループと関連付けられたリモート・ドメイン・アクセス・ポイント名として識別されます。ドメイン・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。

注記 ドメイン・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で、かつリモート・ドメイン・アクセス・ポイントに `CONNECTION_PRINCIPAL_NAME` パラメータが指定されている場合、ゲートウェイは通信相手がそのパラメータで指定された名前で見ると見なします。

DM_EXPORT

`DM_EXPORT` セクション (`DM_LOCAL_SERVICES` セクション) では、CORBA 環境において、`DM_LOCAL` セクションで定義されたローカル・ドメイン・アクセス・ポイントを通じてエクスポートされる BEA Tuxedo ドメインの `domain_name` を指定します。ローカル・ドメイン・アクセス・ポイントに指定された BEA Tuxedo ドメインは、1 つ以上のリモート BEA Tuxedo ドメインの CORBA クライアントからアクセスできます。`DM_EXPORT` セクションは、CORBA Domains コンフィギュレーションでは必須です。

`DM_EXPORT` セクションのエントリの形式は次のとおりです。

```
service [LACCESSPOINT=local access point name]
        [ACL=...]
```

`service` の形式は次のとおりです。

```
"//domain_name"
```

この `domain_name` は、ローカル `UBBCONFIG` ファイルの `RESOURCES` セクションの `DOMAINID` パラメータに割り当てられた名前です。`DM_EXPORT` セクションでこの名前を入力すると、ローカル・ドメインがリモート・ドメインからの CORBA 要求を受け付けます。また、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のために、ローカル・ドメインのドメイン名以外のドメイン名に対する要求を受け付ける `service` エントリを指定することもできます。

オプション・パラメータの `ACL` 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 BEA Tuxedo CORBA ドメインからのローカル・ドメインへの要求を制限します。ACL の名前は、`DMCONFIG` ファイルの `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、ローカル・ドメインに対するリモート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_EXPORT
"//MUTT"
```

この場合は、名前が `MUTT` のローカル・ドメインが、`DM_REMOTE` セクションで定義されたリモート・ドメイン・アクセス・ポイントを通じてリモート CORBA 要求を受け付けます。

DM_IMPORT

`DM_IMPORT` セクション (`DM_REMOTE_SERVICES` セクション) では、CORBA 環境において、`DM_REMOTE` セクションで定義されたリモート・ドメイン・アクセス・ポイントを通じてインポートされる BEA Tuxedo ドメインの `domain_name` を指定します。リモート・ドメイン・アクセス・ポイントに指定された BEA Tuxedo ドメインは、ローカル・ドメインの CORBA クライアントからアクセスできます。`DM_IMPORT` セクションは、CORBA Domains コンフィギュレーションでは必須です。

`DM_IMPORT` セクションのエントリの形式は次のとおりです。

```
service [RACCESSPOINT=remote domain access point]
        [LACCESSPOINT=local domain access point]
        [TRANTIME=...]
```

`service` の形式は次のとおりです。

```
"//domain_name"
```

この `domain_name` は、リモート `UBBCONFIG` ファイルの `RESOURCES` セクションの `DOMAINID` パラメータに割り当てられた名前です。`DM_IMPORT` セクションでこの名前を入力すると、リモート・ドメインがローカル・ドメインからの CORBA 要求を受け付けます。また、リモート・ドメインがルーティングの中継点として機能する場合のために、リモート・ドメインのドメイン名以外のドメイン名に対する要求を受け付ける `service` エントリを指定することもできます。

たとえば、次の例を考えます。

```
*DM_IMPORT
"//JEFF" RACCESSPOINT=TDOM1
```

この場合は、名前が `JEFF` で、リモート・ドメイン・アクセス・ポイント `TDOM1` と関連付けられているリモート・ドメインが、`DM_LOCAL` セクションで定義されたローカル・ドメイン・アクセス・ポイントを通じて CORBA 要求を受け付けます。

DM_ACCESS_CONTROL

`DM_ACCESS_CONTROL` セクションでは、1 つ以上のアクセス制御リスト (ACL) の名前を指定し、1 つ以上のリモート・ドメイン・アクセス・ポイントを指定された各 ACL 名に関連付けます。 `ACL=ACL_NAME` を設定して `DM_EXPORT` セクションで `ACL` パラメータを使用すると、特定のローカル・ドメイン・アクセス・ポイントを通じてエクスポートされるローカル・ドメインへのアクセスを `ACL_NAME` と関連付けられたリモート・ドメイン・アクセス・ポイントのみに制限できます。

`DM_ACCESS_CONTROL` セクションのエントリの形式は次のとおりです。

ACL_NAME required_parameters

`ACL_NAME` は、アクセス制御リストを指定するために使用される *identifier* です。最大 15 文字まで使用できます。

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・アクセス・ポイント名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` セクションで定義したすべてのリモート・ドメイン・アクセス・ポイントがローカル・ドメインにアクセスできることを指定できます。

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 TDomain 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。`DM_TDOMAIN` セクションでは次の要素を必ず指定します。

- ローカル・ドメイン・アクセス・ポイントごとに1エントリ(リモート・ドメインからの CORBA 要求がそのアクセス・ポイントを通じて受け付けられる場合)
- リモート・ドメイン・アクセス・ポイントごとに1エントリ(ローカル・ドメインからリモート・ドメインへの CORBA 要求がそのアクセス・ポイントを通じて受け付けられる場合)

DM_TDOMAIN セクションでは、1つ以上の WebLogic Server アプリケーションと関連付けられた1つ以上のリモート・ドメイン・アクセス・ポイントのコンフィギュレーションを定義して、アプリケーションで Tuxedo CORBA サーバと WebLogic Server Enterprise JavaBean (EJB) サーバを結合することもできます。詳細については、『BEA Tuxedo の相互運用性』の [2-1 ページの「BEA WebLogic Server との相互運用性」](#) を参照してください。

DM_TDOMAIN セクションのエントリの形式は次のとおりです。

```
AccessPoint required_parameters [optional_parameters]
```

AccessPoint は、DM_LOCAL セクションのローカル・ドメイン・アクセス・ポイントまたは DM_REMOTE セクションのリモート・ドメイン・アクセス・ポイントを識別するために使用される識別子値です。

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・アクセス・ポイントまたはリモート・ドメイン・アクセス・ポイントに関連するネットワーク・アドレスを指定します。関連しているのがローカル・ドメイン・アクセス・ポイントである場合、ネットワーク・アドレスはローカル・ドメイン・ゲートウェイでリモート・ドメインからの接続要求をリスンするために使用されます。関連しているのがリモート・ドメイン・アクセス・ポイントである場合、ネットワーク・アドレスはローカル・ドメイン・ゲートウェイでリモート・ドメインへの接続を開始するために使用されます。

string 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。

```
//hostname: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)。

たとえば、次の例を考えます。

```
*DM_TDOMAIN  
LDOM NWADDR="//sanfran.kmart.com:2507"  
TDOM1 NWADDR="//sanhose.kmart.com:3186"
```

この例では、ゲートウェイ・グループ LGWGRP に属する (LDOM アクセス・ポイントの DM_LOCAL セクションで指定) TDomain ゲートウェイが、リモート・ドメインからの接続要求をアドレス "//sanfran.kmart.com:2507" でリスンするように設定されています。また、TDomain ゲートウェイは、TDOM1 アクセス・ポイントと関連付けられたリモート・ドメインに要求を送信する際には "//sanhose.kmart.com:3186" への接続を開始するように設定されています。

DM_TDOMAIN セクションのオプション・パラメータについては、『BEA Tuxedo のファイル形式とデータ記述方法』のリファレンス・ページ DMCONFIG(5) を参照してください。

factory_finder.ini ファイル

factory_finder.ini ファイルでは、ローカル・ドメインで使用できるリモート・ファクトリ・オブジェクトを特定します。また、リモート・ドメインで使用できるローカル・ファクトリ・オブジェクトも特定します。

factory_finder.ini ファイルには、DM_REMOTE_FACTORIES および DM_LOCAL_FACTORIES という2つのセクションがあります。次の例で明らかに、factory_finder.ini ファイルの形式は DMCONFIG ファイルで使われる構文に基づいています。

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

*DM_LOCAL_FACTORIES
    "factory_id.factory_kind"
    ...
```

次の例は、CORBA ファクトリ・オブジェクトの構文を示しています。

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

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。

注記 Domains コンフィギュレーションに参加する2つのCORBAドメインで、*factory_id.factory_kind* 識別子が同じファクトリ・オブジェクトを持つことはできません。詳細については、3-26 ページの「[factory_finder.ini ファイルでの一意なファクトリ・オブジェクト識別子の指定](#)」を参照してください。

DM_REMOTE_FACTORIES

DM_REMOTE_FACTORIES セクションでは、ローカル・ドメインからアクセスできる（ローカル・ドメインにインポートされる）リモート・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` に登録されたりリモート・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。*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。

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"
```

Domains 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るので、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。次のリストは、TP Framework によって識別子 `BankTeller` でドメイン `Norwest` に登録されたリモート・ファクトリの2つの例を示しています。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。

コードリスト 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。次のリストの例では、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ます。

コードリスト 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"
```

DM_LOCAL_FACTORIES

DM_LOCAL_FACTORIES セクションでは、リモート・ドメインからアクセスできる（リモート・ドメインにエクスポートされる）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・オブジェクトがリモート・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。`NONE` キーワードを使用すると、管理者はローカル・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、DM_LOCAL_FACTORIES 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。

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
```

このエントリは、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
```

このエントリは、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。この別の形式はエイリアスと呼ぶ場合もあります。

注記 ファクトリ ID の最後には .FactoryInterface が必要です。わかりやすくするために、テスト・コンフィギュレーションについての説明では .FactoryInterface が省略されていますが、factory_finder.ini ファイルでは必ず必要です。

関連項目

- 『BEA Tuxedo のファイル形式とデータ記述方法』の [UBBCONFIG\(5\)](#)
- 『BEA Tuxedo のファイル形式とデータ記述方法』の [DMCONFIG\(5\)](#)
- 『BEA Tuxedo のファイル形式とデータ記述方法』の [factory_finder.ini\(5\)](#)

factory_finder.ini ファイルでの一意な ファクトリ・オブジェクト識別子の指定

シングル・ドメイン・コンフィギュレーションでは、ロード・バランシングを実現するために複数のファクトリ・オブジェクトが同じ名前であってもかまいません。ただし、Domains コンフィギュレーションでは、2つのドメインが同じ *factory_id.factory_kind* 識別子のファクトリ・オブジェクトを持つことはできません。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 Domains 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 5.1 以降または BEA Tuxedo 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。

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。

- 企業全体で一意的な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・リストで管理する必要があります。
- *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。この方法では、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。[3-23 ページの「リモート・ファクトリへのエイリアスの割り当て」](#)は、*RNAME* パラメータを使用してエイリアスを作成する *factory_finder.ini* ファイルの例を示しています。

factory_finder.ini ファイルの処理

起動時に、マスタ NameManager は `factory_finder.ini` ファイルを読み込みます。マスタ NameManager の起動される状況によって、`factory_finder.ini` ファイルがすべて読み込まれるか、または一部のみ読み込まれるかが決まります。

- マスタ NameManager プロセスが CORBA アプリケーションを起動する過程で起動された場合は (初期化モード)、`factory_finder.ini` ファイルの全内容が読み込まれます。`factory_finder.ini` ファイルの `DM_REMOTE_FACTORIES` セクションに追加された新しいファクトリ・オブジェクトはすべて、ローカルの BEA Tuxedo アプリケーションに知らされます。
- マスタ NameManager プロセスがプロセス障害の結果として再起動された場合は、`factory_finder.ini` ファイルの `DM_LOCAL_FACTORIES` セクションのみ読み込まれます。`DM_REMOTE_FACTORIES` セクションに追加された新しいファクトリ・オブジェクトは、ローカルの BEA Tuxedo アプリケーションには知らされません。

`factory_finder.ini` ファイルの `DM_REMOTE_FACTORIES` セクションにファクトリ・オブジェクトのある新しいドメインを追加するときには、マスタ NameManager を終了して再起動する必要があります。NameManager の詳細については、『BEA Tuxedo のファイル形式とデータ記述方法』の [TMFFNAME \(5\)](#) を参照してください。

CORBA Domains ・ コンフィギュレーションの種類

BEA Tuxedo Domains コンポーネントを使用して複数の BEA Tuxedo CORBA ドメインを接続する場合は、2 種類のコンフィギュレーションを設定できます。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 DMCONFIG ファイルを使用して設定できます。

直接接続ドメイン

Domains コンフィギュレーションのすべてのドメインは、Domains コンフィギュレーションのほかのすべてのドメインとゲートウェイ接続（直接接続）できます。ドメインが直接接続されている場合、要求は目的のドメインに直接届きます。

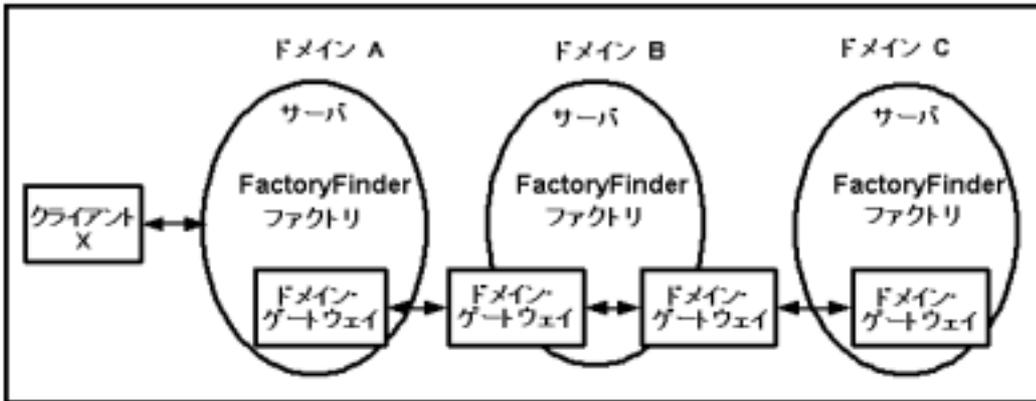
直接接続の Domains コンフィギュレーション（「n 方向」コンフィギュレーション）はドメインの数が少ない場合は合理的ですが、コンフィギュレーションに新しいドメインが追加されるたびに 2 つ、4 つ、またはそれ以上の新しいゲートウェイが必要になります。ある時点で、間接接続ドメインを設定することで、送信速度ではなくドメイン接続の管理しやすさを優先しようと考えようになります。

間接接続ドメイン

まずは、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、次の図の3つのドメイン A、B、および C を考えてください。

図 3-2 間接接続ドメイン



ドメイン A とドメイン B は直接接続されており、ドメイン B とドメイン C も直接接続されていますが、ドメイン A とドメイン C は直接には接続されていません。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、ドメイン B に接続するゲートウェイ・プロセス GAB (A から B のゲートウェイ) があります。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。このコンフィギュレーションを間接接続と呼びます。

この間接接続では、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。

CORBA Domains ・ コンフィギュレーションの例

以下の例は、直接接続された CORBA ドメインを設定する方法を示しています。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

次の 3 つのリストは、直接接続された 3 つのドメイン (Here、There、および Yonder) の UBBCONFIG ファイルを示しています。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。

コード 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
    TMMFFNAME    SRVGRP=ICEGRP SRVID=2
                CLOPT="-A -- -N -M -f <FF ini file for Here>"
    TMMFFNAME    SRVGRP=ICEGRP SRVID=3 CLOPT="-A -- -N"
    TMMFFNAME    SRVGRP=ICEGRP SRVID=4 CLOPT="-A -- -F"
```

3 CORBA Domains の計画とコンフィギュレーション

```
<App Server Name> SRVGRP=GROUP1 SRVID=2
ISL SRVGRP=GROUP1 SRVID=1
CLOPT="-A -- -d /dev/tcp -n //<host>:<port>"

#
# SERVICES
#
*SERVICES

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
```

コード リスト 3-5 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

次の3つのリストは、直接接続された3つのドメイン (Here、There、および Yonder) の DMCNFIG ファイルを示しています。これらのリストを Domains 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。

コード リスト 3-6 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 Domains コンフィギュレーション・ファイル
#
*DM_RESOURCES

    VERSION=U22

#
# DM_LOCAL
#
*DM_LOCAL

    LDOM1   GWGRP=LWGRP   TYPE=TDOMAIN   ACCESSPOINTID="HereG"

#
# DM_REMOTE
#
*DM_REMOTE

    TDOM1   TYPE=TDOMAIN   ACCESSPOINTID="ThereG"
    TDOM2   TYPE=TDOMAIN   ACCESSPOINTID="YonderG"

#
# DM_TDOMAIN
#
*DM_TDOMAIN

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

#
# DM_EXPORT
#
*DM_EXPORT
    "//HereD"

#
# DM_IMPORT
#
*DM_IMPORT
```

```
"/ThereD"  RACCESSPOINT=TDOM1  
"/YonderD" RACCESSPOINT=TDOM2
```

次のリストを Domains 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。

コード 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 There ドメインの DMCONFIG ファイル

```
#
# Copyright (c) 1999 BEA Systems, Inc.
#   All rights reserved
#
#
# Tuxedo Domains コンフィギュレーション・ファイル
#
*DM_RESOURCES

    VERSION=U22

#
# DM_LOCAL
#
*DM_LOCAL

    LDOM1   GWGRP=LWGRP   TYPE=TDOMAIN   ACCESSPOINTID="ThereG"

#
# DM_REMOTE
#
*DM_REMOTE

    TDOM1   TYPE=TDOMAIN   ACCESSPOINTID="HereG"
    TDOM2   TYPE=TDOMAIN   ACCESSPOINTID="YonderG"

#
# DM_TDOMAIN
#
*DM_TDOMAIN

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

#
# DM_EXPORT
#
*DM_EXPORT
    "//ThereD"

#
# DM_IMPORT
#
*DM_IMPORT
```

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

次のリストを Domains 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。

コード 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 Yonder ドメインの DMCONFIG ファイル

```
#
# Copyright (c) 1999 BEA Systems, Inc.
#   All rights reserved
#
#
# Tuxedo Domains コンフィギュレーション・ファイル
#
*DM_RESOURCES

    VERSION=U22

#
# DM_LOCAL
#
*DM_LOCAL

    LDOM1   GWGRP=LWGRP   TYPE=TDOMAIN   ACCESSPOINTID="YonderG"

#
# DM_REMOTE
#
*DM_REMOTE

    TDOM1   TYPE=TDOMAIN   ACCESSPOINTID="HereG"
    TDOM2   TYPE=TDOMAIN   ACCESSPOINTID="ThereG"

#
# DM_TDOMAIN
#
*DM_TDOMAIN

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

#
# DM_EXPORT
#
*DM_EXPORT
    "//YonderG"

#
# DM_IMPORT
#
*DM_IMPORT

    "//HereD"   RACCESSPOINT=TDOM1
    "//ThereD"  RACCESSPOINT=TDOM2
```

factory_finder.ini ファイルの例

以下の2つのリストは、Here ドメインと There ドメインの factory_finder.ini ファイルを示しています。Yonder ドメインには factory_finder.ini ファイルは必要ありません。

コード リスト 3-9 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
# これはローカル・ドメイン
#
# 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"
```

コード リスト 3-10 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
# これはリモート・ドメイン
#
# 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

以下の節では、BEA Tuxedo Domains 環境を管理する方法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 `dadmin(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
- ドメイン・ゲートウェイ・サーバを使用する
- Domains 環境でのトランザクションの管理

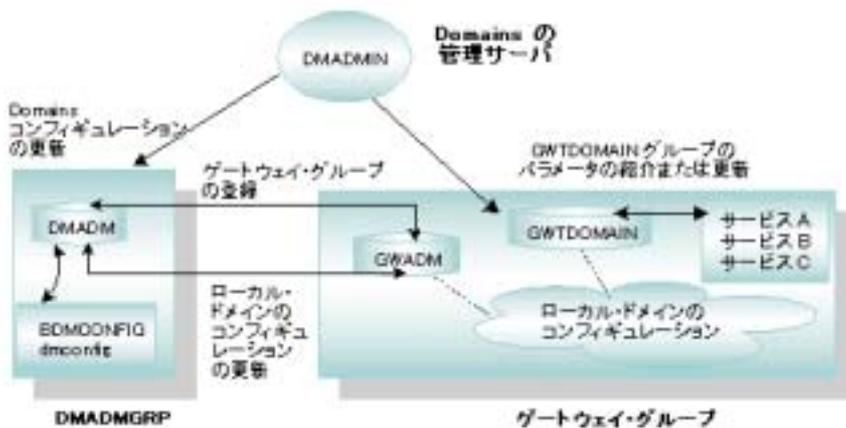
Domains の実行時管理コマンドを使用する

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 Domains コンフィギュレーションを追加するには、`tmconfig(1)` または `tmadmin(1)` コマンドを使用します。`tmadmin` を使用すると、ドメイン・ゲートウェイ・グループや個々のドメイン・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 コンポーネント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



BEA Tuxedo Domains コンポーネントには、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、管理コマンドを変換したり、サービス要求を DMADMIN サービスに送信したりするフロントエンド・プロセッサとして機能します。DMADMIN サービスは、DMADM サーバにより宣言される汎用管理サービスです。

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)** - Domains 管理サーバです。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、ドメイン・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。登録サービスは、要求元のドメイン・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をダウンロードします。`DMADM` サーバは、登録済みのドメイン・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ドメイン・ゲートウェイ・グループに伝播します。
- **GWADM(5)** - ゲートウェイ管理サーバです。`DMADM` サーバに登録され、対応するドメイン・ゲートウェイ・グループに使用され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、対応するドメイン・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)** - TDomain ゲートウェイ・サーバです。複数の BEA Tuxedo ドメイン間で相互運用性を実現します。WebLogic Tuxedo Connector (WTC) ゲートウェイ (BEA WebLogic Server コンポーネント) と関係することで、BEA Tuxedo TDomain ゲートウェイは Tuxedo ドメインと WebLogic Server アプリケーションの間でも相互運用性を実現できます。

注記 `GWTDOMAIN` 以外のドメイン・ゲートウェイのタイプについては、[『ATMI アプリケーションでの BEA Tuxedo TOP END Domain Gateway の使用』](#) および [『BEA eLink Documentation』](#) を参照してください。
- **BDMCONFIG** - Domains コンフィギュレーション・ファイルのバイナリ形式です。`TUXCONFIG` ファイルおよび `factory_finder.ini` ファイル (CORBA のみ) と一緒に、Domains コンフィギュレーションを作成する

ために BEA Tuxedo ソフトウェアで必要なすべてのコンフィギュレーション・パラメータを格納します。

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ドメイン・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。

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ドメイン・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。
- 統計情報やドメイン・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。
- ドメイン・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、アクティブなドメイン・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。
- 登録済みのドメイン・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、登録済みのドメイン・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` コマンドと `GWADM` サーバによって使用されます。
- `DMADM_svrid`。 `svrid` 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 (ドメイン・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。このグループには、DMADM サーバのインスタンスが 1 つのみ必要です。

関連項目

- 『BEA Tuxedo のファイル形式とデータ記述方法』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、ドメイン・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、ドメイン・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。
- ドメイン・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、ドメイン・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。
- ドメイン・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、ドメイン・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、GWADM サーバが属するドメイン・ゲートウェイ・グループに関連付けられたローカル・ドメイン・アクセス・ポイントの名前 (BDMCONFIG ファイルの `DM_LOCAL` セクションで指定) に基づいてサーバ名を宣言します。`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ドメイン・ゲートウェイ・グループまたは特定のドメイン・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` の一部として指定することはできません。また、ドメイン・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、ドメイン・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` を取得できません。

関連項目

- 『BEA Tuxedo のファイル形式とデータ記述方法』の [GWADM\(5\)](#)

ドメイン・ゲートウェイ・サーバを使用する

ドメイン・ゲートウェイ・サーバは、リモート・ドメイン・ゲートウェイ・サーバへの接続を実現し、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

- [1-4 ページの「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、ドメイン・ゲートウェイ・グループで 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、ドメイン・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。

ドメイン・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。ドメイン・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。

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。

- [4-10 ページの「Domains で TMS 機能を使用する」](#)
- [4-13 ページの「トランザクションで GTRID マッピングを使用する」](#)
- [4-21 ページの「ログ機能によるトランザクションのトラッキング」](#)
- [4-24 ページの「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

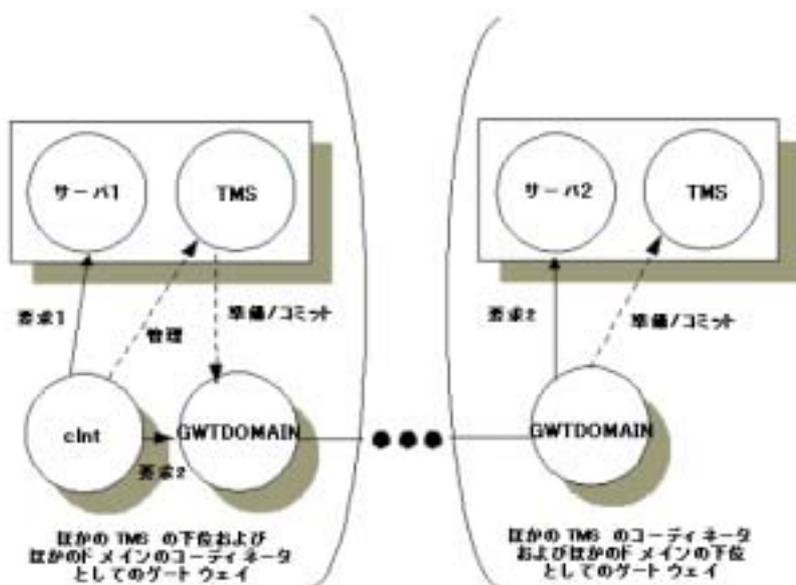
BEA Tuxedo システムの TMS は、X/Open XA 準拠のリソース・マネージャを使用するサーバ・グループと暗黙的に関連付けられた特別なサーバです。TMS サーバは、分散型の 2 フェーズ・コミット・プロトコルに伴うアプリケーション・サーバでの遅延を解消します。つまり、TMS サーバは、TMS サービスに対して特殊なサービス要求を発行することにより、トランザクションのコミットを調整します。このサービスは、すべての TMS サーバで提供されます。

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。ドメイン・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。

ドメイン・ゲートウェイは、次のようにしてドメインをまたがるトランザクションを調整します。

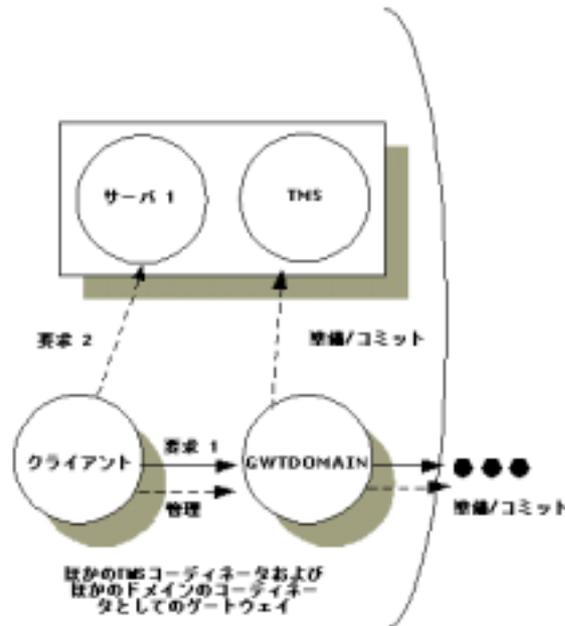
1. ドメイン・ゲートウェイは、TMS サービスを宣言し、そのサービスに関連するすべての操作を行います。このサービスに送信されたメッセージは、適切なドメイン・ゲートウェイ・グループによって使用されるキューに登録され、ゲートウェイはグループに対応付けられたトランザクションを管理します。
2. ゲートウェイは、ドメイン内の別のグループによって調整されるトランザクションの下位として位置付けることもできます。この場合、ゲートウェイは、ほかのリモート・ドメインで実行されるトランザクションよりも上位になります。リモート・ドメインによって調整されるトランザクションの下位として働く場合、ゲートウェイは、トラン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 別のドメイン・ゲートウェイ・グループの下位ドメイン・ゲートウェイまたはコーディネータとしてのドメイン・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 ドメイン・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(ドメイン・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 機能で指定されたりモート・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・サービス定義の中のパラメータ AUTOTRAN に Y を設定します。

詳細については、『BEA Tuxedo のファイル形式とデータ記述方法』の [DMCONFIG\(5\)](#) の `DM_IMPORT` 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、使用するドメイン・ゲートウェイのインスタンスによって決まります (TDomain、TOP END、SNA、または OSI TP)。

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ドメイン・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、TMS サーバを使用して、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、BEA Tuxedo 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。

「密結合関係」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」の概念に基づいて密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

ブ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・ブ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・ブランチを生成します。RM は、各トランザクション・ブランチを別々に処理します。トランザクション・ブランチ間では、データやロックは共有されません。トランザクション・ブ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・ブランチが定義され、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では、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・ツリーでは新しいブ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、対応するドメイン・ゲートウェイ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、ユーザー定義の識別子を使用して 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 (BEA Tuxedo のインプリメンテーションでは、この識別子は GTRID になります)。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ています。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

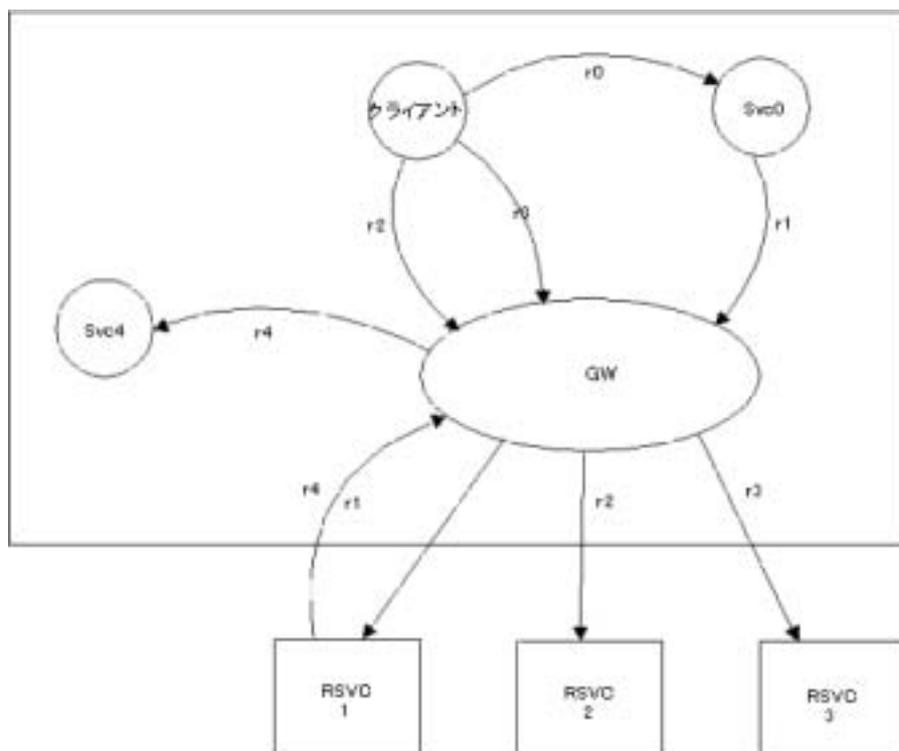
の数は、トランザクション内のドメイン数になります。これは、すべてのブランチが同じリソース・マネージャのインスタンスにアクセスしている場合も同じです。

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 (r_0) およびリモート・サービスに対する 2 つの要求 (r_2 および r_3) 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。 r_0 は、ローカル・サービス (svc_0) に送信され、別のリモート・サービス要求 (r_1) を生成します。 r_1 はリモート・サービス $Rsvc_1$ に送信され、 $Rsvc_1$ は、ループバック・サービス要求 r_4 をローカル・サービス svc_4 に送信します。 svc_0 と svc_4 は、別々のグループ (G_0 と G_4) 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 $Rsvc_1$ 、 $Rsvc_2$ 、および $Rsvc_3$ 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 ドメイン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

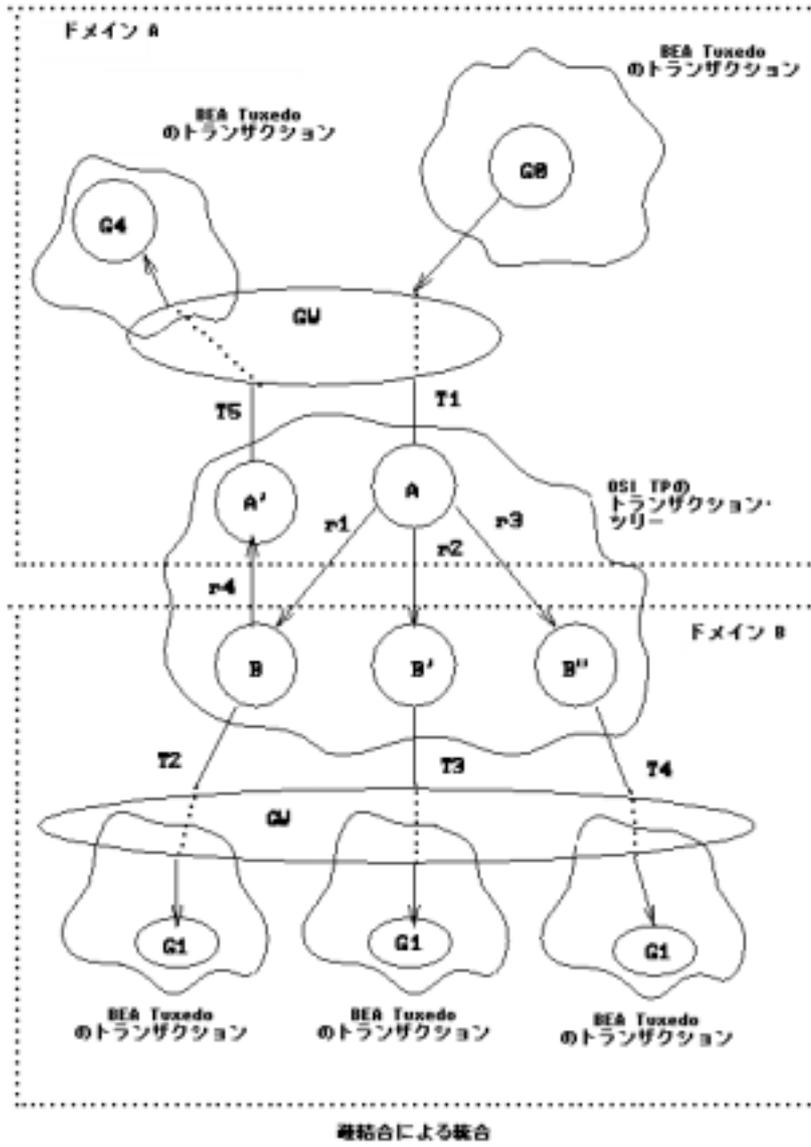
び r_3 の 3 つの要求は、それぞれ別の OSI TP ダイアログにマッピングされ、次に 1 つの OSI TP のトランザクション・ブランチにマッピングされます。ただし、OSI TP は、XAP-TP の機能を使用して、一意な識別子 (T_1) で OSI TP トランザクション全体を参照し、 r_1 、 r_2 、および r_3 の 3 つの要求にも使用します。OSI TP トランザクション識別子を作成し、対応する OSI TP トランザクション・ツリーを構築するかどうかは、XAP-TP に依存します。一般的な Domains ソフトウェアでは、 r_1 、 r_2 、および r_3 の 3 つの要求の T_1 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 r_1 、 r_2 、および r_3 は、3 つの異なる BEA Tuxedo トランザクション (T_2 、 T_3 、および T_4) にマッピングされます。グラフでは、Domain B のドメイン・ゲートウェイ・グループ GW が、グループ G_1 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 r_4 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 T_2 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。 r_4 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 T_5 を生成します。トランザクション・グラフでは、ドメイン A のドメイン・ゲートウェイ・グループ GW が、グループ G_4 を調整する様子を示します。

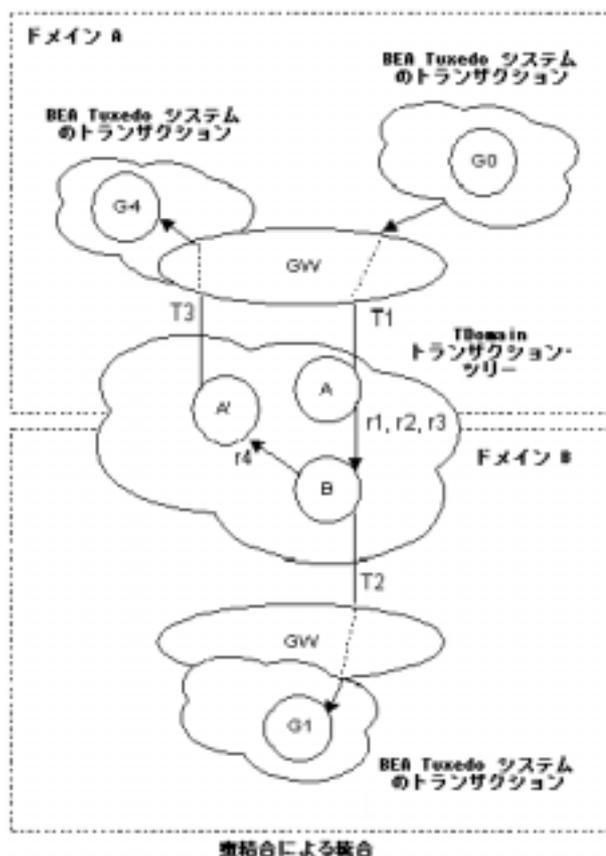
これらのマッピング関係により、OSI TP プロトコルの階層構造は強化されます。ただし、これらのマッピング関係は疎結合であるため、トランザクション内でデッドロックが発生する可能性が高まります。たとえば、グループ G_1 が示す RM には、3 つの BEA Tuxedo トランザクションがアクセスします。

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 のドメイン・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 r_4 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。 r_4 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 のドメイン・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。
- ドメイン・ゲートウェイの各インスタンス (TDomains、TOP END、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` 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、『BEA Tuxedo のファイル形式とデータ記述方法』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 (`dmadmin`) 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 `dmadmin(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、ドメイン・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、ドメイン・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すべてが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。