



BEATuxedo®

BEA Tuxedo コマ ンド・リファレンス

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

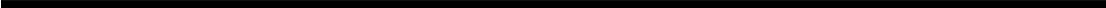
対象読者	vii
e-docs Web サイト	vii
マニュアルの印刷方法	viii
関連情報	viii
サポート情報	viii
表記上の規則	ix

セクション 1 - コマンド

BEA Tuxedo コマンドについて	1-6
blde_dce(1)	1-8
blde_dce(1)	1-9
buildclient(1)	1-10
buildobjclient(1)	1-15
buildobjserver(1)	1-20
buildserver(1)	1-25
buildtms(1)	1-32
buildwsh(1)	1-34
cobcc(1)	1-36
dadmin(1)	1-39
dmloadcf(1)	1-71
dmunloadcf(1)	1-75
gencat(1)	1-77
genicf(1)	1-80
idl(1)	1-82
idl2ir(1)	1-87
idltojava(1)	1-89
ir2idl(1)	1-94
irdel(1)	1-96
ISL(1)	1-97
m3idltojava(1)	1-108

mkfldhdr、mkfldhdr32(1)	1-112
mklanginfo(1)	1-114
qadmin(1)	1-117
rex(1)	1-145
tidl(1)	1-148
tlisten(1)	1-156
tadmin(1)	1-161
tboot(1)	1-182
tmconfig、wtmconfig(1)	1-191
tmipcrm(1)	1-204
tmloadcf(1)	1-207
tmshutdown(1)	1-211
tmunloadcf(1)	1-217
topendpasswd(1)	1-219
tpacladd(1)	1-220
tpaclcv(1)	1-222
tpacldel(1)	1-223
tpaclmod(1)	1-224
tpaddusr(1)	1-226
tpdelusr(1)	1-228
tpgrpadd(1)	1-230
tpgrpdel(1)	1-232
tpgrpmod(1)	1-233
tpmigldap(1)	1-235
tpmodusr(1)	1-237
tpusradd(1)	1-239
tpusrdel(1)	1-242
tpusrmod(1)	1-243
tuxadm(1)	1-245
tuxwsvr(1)	1-247
txrpt(1)	1-253
ud、wud(1)	1-255
uuidgen(1)	1-261
viewc、viewc32(1)	1-263
viewdis、viewdis32(1)	1-266

wlisten(1).....1-267



このマニュアルについて

このマニュアルでは、BEA Tuxedo ソフトウェアのコマンドについて説明します。リファレンス・ページはコマンド名のアルファベット順になっています。

対象読者

このマニュアルは、次のような読者を対象としています。

- BEA Tuxedo 環境でアプリケーションをコンフィギュレーションおよび管理するシステム管理者
- BEA Tuxedo 環境でアプリケーションをプログラミングするアプリケーション開発者

このマニュアルは、BEA Tuxedo プラットフォーム、および C 言語または COBOL 言語のプログラミングの知識があることを前提としています。

e-docs Web サイト

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

マニュアルの印刷方法

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

このマニュアルの PDF 版は、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 Documents ページの [PDF 版] ボタンをクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader をお持ちではない場合は、Adobe Web サイト (<http://www.adobe.co.jp/>) から無償でダウンロードできます。

関連情報

関連情報は各リファレンス・ページの「関連項目」に一覧表示されています。

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 を通して 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。
<i>イタリック 体</i>	強調またはマニュアルのタイトルを示します。

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

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

セクション 1 - コマンド

表 1BEA Tuxedo コマンド

名前	機能説明
BEA Tuxedo コマンドについて	BEA Tuxedo アプリケーションの設定および保守に使用するコマンドについての説明
bldc_dce(1)	OSF/DCE を介して呼び出される BEA Tuxedo システム・クライアントを作成
blds_dce(1)	OSF/DCE を呼び出す BEA Tuxedo システム・サーバを作成
buildclient(1)	BEA Tuxedo クライアント・モジュールを作成
buildobjclient(1)	CORBA クライアント・アプリケーションを構築
buildobjserver(1)	CORBA サーバ・アプリケーションを構築
buildserver(1)	BEA Tuxedo サーバ・ロード・モジュールを構築
buildtms(1)	トランザクション・マネージャ・サーバのロード・モジュールを構築
buildwsh(1)	カスタマイズされたワークステーション・ハンドラ・プロセスを構築
cobcc(1)	COBOL コンパイル・インターフェイス
dmadmin(1)	BEA Tuxedo ドメイン管理プログラムのコマンド・インタプリタ
dmloadcf(1)	DMCONFIG ファイルを構文解析し、バイナリの BDMCONFIG コンフィギュレーション・ファイルをロード
dmunloadcf(1)	BDMCONFIG ファイル (バイナリのドメイン・コンフィギュレーション・ファイル) をアンロード

セクション 1 - コマンド

表 1BEA Tuxedo コマンド (続き)

名前	機能説明
<code>gencat(1)</code>	フォーマットされたメッセージ・カタログを生成
<code>genicf(1)</code>	ICF ファイルを生成
<code>idl(1)</code>	Object Management Group (OMG) のインターフェイス定義語ファイルをコンパイルし、インターフェイスに必要なファイルを生成
<code>idl2ir(1)</code>	インターフェイス・リポジトリを作成し、そのインターフェイス・リポジトリにインターフェイス定義をロード
<code>idltojava(1)</code>	OMG によって定義される、IDL から Java へのマッピングに基づいて、IDL ファイルを Java ソース・コードにコンパイル
<code>ir2idl(1)</code>	インターフェイス・リポジトリの内容を表示
<code>irdel(1)</code>	指定されたオブジェクトをインターフェイス・リポジトリから削除
<code>ISL(1)</code>	IIOP を使用した、リモート BEA Tuxedo クライアントからの BEA Tuxedo オブジェクトへのアクセスを有効化
<code>m3idltojava(1)</code>	Object Management Group (OMG) のインターフェイス定義語 (IDL) ファイルをコンパイルし、Java でインプリメントするインターフェイス定義に必要なクライアント・スタブおよびサーバ・スケルトン・ファイルを生成
<code>mkfldhdr、 mkfldhdr32(1)</code>	フィールド・テーブルからヘッダ・ファイルを作成
<code>mklanginfo(1)</code>	ロケールの言語情報定数をコンパイル
<code>qmadmin(1)</code>	キュー・マネージャの管理プログラムのコマンド・インタプリタ
<code>rex(1)</code>	オフラインの正規表現コンパイラおよびテスタ
<code>tidl(1)</code>	インターフェイス定義言語コンパイラ
<code>tlisten(1)</code>	汎用リスナ・プログラム
<code>tmadmin(1)</code>	BEA Tuxedo の掲示板のコマンド・インタプリタ
<code>tmboot(1)</code>	BEA Tuxedo のコンフィギュレーションを起動

表 1BEA Tuxedo コマンド (続き)

名前	機能説明
<code>tmconfig</code> , <code>wtmconfig(1)</code>	ネイティブ・クライアントまたはワークステーション・クライアントとして実行中の BEA Tuxedo アプリケーションに関する情報を動的に更新および検索
<code>tmipcrm(1)</code>	ローカル・マシンで、BEA Tuxedo アプリケーションが割り当てた IPC リソースを削除
<code>tmloadcf(1)</code>	UBBCONFIG ファイル (テキスト形式のコンフィギュレーション・ファイル) を解析し、TUXCONFIG ファイル (バイナリのコンフィギュレーション・ファイル) をロード
<code>tmsshutdown(1)</code>	BEA Tuxedo サーバをシャットダウン
<code>tmunloadcf(1)</code>	TUXCONFIG ファイル (バイナリのコンフィギュレーション・ファイル) をアンロード
<code>topenpasswd(1)</code>	BEA Tuxedo システムの TOP END Domain Gateway 機能を使用するための BEA TOP END パスワードを追加
<code>tpacladd(1)</code>	アクセス制御リストの新しいエントリをシステムに追加
<code>tpaclcvt(1)</code>	BEA Tuxedo セキュリティ・データ・ファイルを変換
<code>tpacldel(1)</code>	アクセス制御リストのエントリを削除
<code>tpaclmod(1)</code>	システム上のアクセス制御リストのエントリを変更
<code>tpaddusr(1)</code>	BEA Tuxedo パスワード・ファイルを作成
<code>tpdelusr(1)</code>	BEA Tuxedo パスワード・ファイルからユーザを削除
<code>tpgrpadd(1)</code>	新しいグループをシステムに追加
<code>tpgrpdel(1)</code>	システムからグループを削除
<code>tpgrpmod(1)</code>	システム上のグループを変更
<code>tpmigldap(1)</code>	Tuxedo ユーザおよびグループを WebLogic Server に移行
<code>tpmodusr(1)</code>	BEA Tuxedo システム・パスワード・ファイルを保守

セクション 1 - コマンド

表 1BEA Tuxedo コマンド (続き)

名前	機能説明
<code>tpusradd(1)</code>	新しいプリンシパルをシステムに追加
<code>tpusrdel(1)</code>	システムからユーザを削除
<code>tpusrmod(1)</code>	システム上のユーザ情報を変更
<code>tuxadm(1)</code>	BEA Tuxedo Administration Console の CGI ゲートウェイ
<code>tuxwsvr(1)</code>	BEA Tuxedo Administration Console で使用するミニ・ウェブ・サーバ
<code>txrpt(1)</code>	BEA Tuxedo システム・サーバ / サービス・レポート・プログラム
<code>ud、wud(1)</code>	BEA Tuxedo ドライバ・プログラム
<code>uuidgen(1)</code>	UUID (Universal Unique Identifier) を生成
<code>viewc、viewc32(1)</code>	BEA Tuxedo VIEW を生成する <code>views</code> コンパイラ

表 1BEA Tuxedo コマンド (続き)

名前	機能説明
<code>viewdis</code> 、 <code>viewdis32(1)</code>	バイナリ VIEW ファイル対応の VIEW 逆アセンブラ
<code>wlisten(1)</code>	BEA Tuxedo Administration Console リスナ・プロセス

BEA Tuxedo コマンドについて

機能説明 『BEA Tuxedo コマンド・リファレンス』では、BEA Tuxedo ソフトウェアで提供されるシェル・レベルのコマンドについてアルファベット順に説明します。

リファレンス・ページのコマンド構文 特に指定がない限り、リファレンス・ページの形式で説明されているコマンドは、下記の構文に従ってオプションや引数を受け付け、次に説明するように解釈されます。

name [*-option ...*] [*cmdarg ...*]

name は実行可能ファイルの名前、*option* は *noargletter ...* または *argletter optarg [...]* のいずれかの文字列です。

option の前には常に `&dlq;-` が付きます。

noargletter

オプション引数を必要としない *option* を表す 1 文字の英字。“-”の後には、2 つ以上の *noargletter* オプションを付けることができます。

optarg

その前に指定されている *argletter* を満足させる文字列。*argletter* の後に複数の *optargs* を指定するときには、それぞれをカンマあるいは空白で区切らなければなりません。ただし、空白で区切る場合には、全体をカッコで囲む必要があります。

cmdarg

コマンドのオペランドを表すパス名 (またはその他のコマンド引数)。

-

(ダッシュ) 単独で使用された場合、追加引数を標準入力から指定することを意味します。

--

(2 つのダッシュ) この後に続く項目を下位プログラムの引数と見なします。

- [] *option* または *cmdarg* を囲み、それらが省略可能であることを示します。
- { } *or* 記号 (|) で区切った複数の *cmdargs* を囲み、対応する *option* の使用時にそれらの中からいずれかを選択しなければならないことを示します。
- ... *option* または *cmdarg* を複数指定できることを意味します。

bldc_dce(1)

- 名前** bldc_dce - OSF/DCE を介して呼び出すことができる BEA Tuxedo ATMI クライアントを作成
- 形式** bldc_dce [-o *output_file*] [-i *idl_options*] [-f *firstfiles*] [-l *lastfiles*] [*idl_file* . . .]
- 機能説明** bldc_dce は、すべての入力 IDL ソース・ファイルおよび関連する ACF ソース・ファイルを解析し、それらを C のソース・ファイルとオブジェクト・ファイル、および OSF/DCE ライブラリと組み合わせて、BEA Tuxedo ATMI クライアントを生成します。この生成された BEA Tuxedo ATMI クライアントは、DCE RPC (つまり DCE RPC クライアント) を介して呼び出すことができます。
- コマンド行引数には、入力 IDL ソース・ファイルと、IDL コンパイラの動作を制御するオプションを指定します。使用できるオプションは次のとおりです。
- o *output_file*
デフォルトのファイル名は a.out です。
 - i *idl_options*
IDL コンパイラに渡すオプションを指定します。C のコンパイル・システムに関連するオプションは、このプログラムによって自動的に提供されます。このオプションを使用すると、アプリケーションでマネージャ・エントリ・ポイント・ベクトル (Manager Entry Point Vector) を使用するための -no_mepv オプションを指定できます。
 - f *firstfiles*
BEA Tuxedo ATMI システム・ライブラリと OSF/DCE ライブラリより前に取り込んでコンパイルする、コンパイラ・オプション、C ソース・ファイル、および C オブジェクト・ファイルを指定します。
 - l *lastfiles*
BEA Tuxedo ATMI システム・ライブラリと OSF/DCE ライブラリの後に取り込んでコンパイルする C ライブラリを指定します。
- 関連項目** [tidl\(1\)](#)

blds_dce(1)

名前 blds_dce - OSF/DCE を呼び出す BEA Tuxedo ATMI サーバを作成

形式 blds_dce [-o *output_file*] [-i *idl_options*] [-f *firstfiles*]
[-l *lastfiles*] [-s *service*] [*idl_file* . . .]

機能説明 blds_dce は、すべての入力 IDL ソース・ファイルおよび関連 ACF ソース・ファイルを解析し、それらを C ソース・ファイルと C オブジェクト・ファイル、および OSF/DCE ライブラリと組み合わせて、DCE RPC 呼び出しを作成できる BEA Tuxedo ATMI サーバを生成します。このコマンドの主な使用目的は、BEA Tuxedo システムから OSF/DCE へのゲートウェイ・プロセスを作成することです。

コマンド行引数には、入力 IDL ソース・ファイルと、IDL コンパイラの動作を制御するオプションを指定します。使用できるオプションは次のとおりです。

-o *output_file*

デフォルトのファイル名は a.out です。

-i *idl_options*

IDL コンパイラに渡すオプションを指定します。C のコンパイル・システムに関連するオプションは、このプログラムによって自動的に提供されます。このオプションを使用すると、アプリケーションでマネージャ・エントリ・ポイント・ベクトル (Manager Entry Point Vector) を使用するための -no_mepv オプションを指定できます。

-f *firstfiles*

BEA Tuxedo ATMI システム・ライブラリと OSF/DCE ライブラリより前に取り込んでコンパイルする、コンパイラ・オプション、C ソース・ファイル、および C オブジェクト・ファイルを指定します。

-l *lastfiles*

BEA Tuxedo ATMI システム・ライブラリと OSF/DCE ライブラリの後に取り込んでコンパイルする C ライブラリを指定します。

-s *service*[,*service* . . .]

サーバで宣言するサービスを指定します。

関連項目 [tidl\(1\)](#)

buildclient(1)

名前	buildclient - BEA Tuxedo ATMI クライアント・モジュールを作成
形式	buildclient [-C] [-v] [{-r <i>rmname</i> -w }] [-o <i>name</i>] [-f <i>firstfiles</i>] [-l <i>lastfiles</i>]
機能説明	<p>buildclient は、BEA Tuxedo ATMI クライアント・モジュールを作成するときに使用します。このコマンドでは、-f および -l オプションで指定されるファイルと標準 BEA Tuxedo ATMI ライブラリを組み合わせ、1つのロード・モジュールを形成します。このロード・モジュールは、使用しているオペレーティング・システム用に定義されたデフォルトの C 言語コンパイラ・コマンドを使用し、buildclient によって作成されます。UNIX システム用のデフォルトの C 言語コンパイラ・コマンドは cc(1) コマンドです (UNIX システムのリファレンス・マニュアルを参照してください)。</p> <p>-v</p> <p>buildclient を冗長モードで機能させます。特に、cc コマンドをその標準出力に書き出します。</p> <p>-w</p> <p>ワークステーション・ライブラリを使用してクライアントを作成することを指定します。デフォルトの設定では、ネイティブ・モードおよびワークステーション・モード両方のライブラリがある場合には、ネイティブ・クライアントが作成されます。このオプションは、-r オプションとともに使用できません。</p> <p>-r <i>rmname</i></p> <p>クライアントに関連するリソース・マネージャを指定します。<i>rmname</i> は、\$TUXDIR/udataobj/RM にあるリソース・マネージャのテーブルにあるものでなければなりません。このファイルの各行は次のような形式になります。</p> <pre>rmname:rmstructure_name:library_names</pre> <p>詳細については、builddtms(1) を参照してください。<i>rmname</i> の値を使用することにより、\$TUXDIR/udataobj/RM にあるエントリは、リソース・マネージャに関連したライブラリを自動的に含み、トランザクション・マネージャとリソース・マネージャ間のインターフェイスを正しく設定するのに使用されます。他の値は、リソース・マネージャのテーブルに追加されているものを指定できます。-r オプションが指</p>

定されない場合のデフォルト設定では、クライアントはリソース・マネージャに関連しません。UBBCONFIG(5) のリファレンス・ページを参照してください。

- o 出力ロード・モジュールのファイル名を指定します。このオプションの指定がない場合は、a.out という名前になります。
- f buildclient のリンク段階の最初に、BEA Tuxedo ATMI ライブラリより前に取り込む 1 つ以上のユーザ・ファイルを指定します。複数のファイルを指定する場合は、各ファイル名を空白で区切り、リスト全体を引用符で囲まれなければなりません。このオプションは何回指定してもかまいません。コンパイラ・オプションおよび引数を含むように指定する場合は、以下に説明する CFLAGS および ALTFLAGS 環境変数を使用します。
- l buildclient のリンク段階の最後で、BEA Tuxedo ライブラリの後に取り込む 1 つ以上のユーザ・ファイルを指定します。複数のファイルを指定する場合は、各ファイル名を空白で区切り、リスト全体を引用符で囲まれなければなりません。このオプションは何回指定してもかまいません。
- C COBOL のコンパイルを指定します。

環境変数

TUXDIR

buildclient は、環境変数 TUXDIR を使用して、クライアント・プロセスのコンパイル時に使用する BEA Tuxedo ATMI ライブラリとインクルード・ファイルを見つけます。

CC

buildclient は通常、デフォルトの C 言語コンパイル・コマンドを使用してクライアント実行可能コードを生成します。デフォルトの C 言語コンパイル・コマンドは、サポートされている各オペレーティング・システムごとに定義されており、UNIX システムの場合は cc(1) です。代替コンパイラを指定できるようにするため、buildclient は環境変数 cc が存在するかどうかを調べます。CC が buildclient の環境に存在しない場合、またはこの環境変数が文字列 "" である場合、buildclient はデフォルトの C 言語コンパイラを使用します。環境内

に `CC` が存在する場合、実行されるコンパイラの名前が `CC` の値となります。

CFLAGS

環境変数 `CFLAGS` は、コンパイラ・コマンド行の一部として引き渡される引数のセットを指定するときに使用します。この変数は、コマンド行オプション `&ldq;-I"DIR"/include"` とともに `buildclient` により自動的に引き渡されます。`CFLAGS` が `buildclient` の環境に存在しない場合、またはこの環境変数が文字列 `"` である場合、`buildclient` はコンパイラ・コマンド行引数を追加しません。

ALTCC

`-c` オプションを使って COBOL のコンパイルを指定すると、`buildclient` は通常、BEA Tuxedo のシェル `cobcc` を使用して `cob` を呼び出し、クライアント実行可能コードを生成します。`buildclient` は、別のコンパイラを指定する `ALTCC` という環境変数の有無をチェックします。`ALTCC` が `buildclient` の環境に存在しない場合、またはこの環境変数が文字列 `"` である場合、`buildclient` は `cobcc` を使用します。環境に `ALTCC` が存在する場合は、その値をとって実行するコンパイラ・コマンドとします。

注記 Windows システムでは、環境変数 `ALTCC` および `ALTCFLAGS` は使用できません。これらを設定すると予期しない結果が生じます。まず COBOL コンパイラを使用してアプリケーションをコンパイルして、次に生成されたオブジェクト・ファイルを `buildclient(1)` コマンドに渡す必要があります。

ALTCFLAGS

環境変数 `ALTCFLAGS` には、`-c` オプションを指定した場合に、COBOL コンパイラ・コマンドの一部として渡す追加の引数を指定します。コマンド行オプションに対する追加です。

```
"-I${TUXDIR}/include"
```

このオプションは、`buildclient` によって自動的に渡されます。`.-c` オプションを使用する場合、コンパイラ・オプションやその引数を `buildclient -f` オプションで指定するとエラーが発生するため、`ALTCFLAGS` を使用する必要があります。設定しなかった場合は、上記の `CFLAGS` と同じ値に設定されます。

注記 ALTC 環境変数についての下の注記を参照してください。

COBOPT

環境変数 COBOPT には、`-c` オプションを指定した場合に、COBOL コンパイラが使用する追加の引数を指定します。

COBCPY

環境変数 COBCPY には、`-c` オプションを指定した場合に、COBOL コンパイラが使用する COBOL コピー・ファイルが存在するディレクトリを指定します。

LD_LIBRARY_PATH (UNIX システムの場合)

環境変数 LD_LIBRARY_PATH には、BEA Tuxedo システムの共有オブジェクトに加えて、COBOL コンパイラが使用する共有オブジェクトが存在するディレクトリを指定します。一部の UNIX システムでは、別の環境変数が必要となる場合もあります。HP-UX システムでは SHLIB_PATH 環境変数を使用します。AIX システムでは LIBPATH 環境変数を使用します。

LIB (Windows NT システム)

ライブラリを検索するディレクトリのリストを指定します。それぞれのディレクトリはセミコロン (;) で区切ります。

移植性

buildclient コンパイル・ツールは、以下のプラットフォームで使用できません。

- BEA Tuxedo ATMI サーバ環境がサポートされているプラットフォーム
- 32 ビット Windows オペレーティング・システムを実行するワークステーション・プラットフォーム

buildclient コマンド行で指定するファイル名は、お使いのオペレーティング・システムの構文と指定方法に従ってください。

使用例

```
CC=ncc CFLAGS="-I /APPDIR/include"; export CC CFLAGS
buildclient -o empclient -f emp.c -f "userlib1.a userlib2.a"
```

```
COBCPY=$TUXDIR/cobinclude
COBOPT="-C ANS85 -C ALIGN=8 -C NOIBMCOMP -C TRUNC=ANSI -C OSEXT=cbl"
COBDIR=/usr/lib/cobol LD_LIBRARY_PATH=$COBDIR/coblib:$TUXDIR/lib
export COBOPT COBCPY COBDIR LD_LIBRARY_PATH
buildclient -C -o empclient -f name.cbl -f "userlib1.a userlib2.a"
```

関連項目 [buildserver\(1\)](#)、[buildtms\(1\)](#)、[compilation\(5\)](#)
UNIX システムのリファレンス・マニュアルの [cc\(1\)](#)、[ld\(1\)](#)

buildobjclient(1)

名前 buildobjclient - CORBA クライアント・アプリケーションを構築

形式 buildobjclient [-v][-o name] [-f firstfile-syntax]
[-l lastfile-syntax] -P

機能説明 buildobjclient コマンドを使用して、CORBA クライアント・アプリケーションを構築できます。このコマンドは、`-f` および `-l` オプションで指定されるファイルと標準の CORBA ライブラリを組み合わせ、クライアント・アプリケーションを形成します。クライアント・アプリケーションは、使用しているオペレーティング・システム用に定義されたデフォルトの C++ 言語コンパイル・コマンドによって作成されます。

指定された `.c` ファイルおよび `.cpp` ファイルはすべて、使用しているオペレーティング・システムのコンパイル・システムを 1 回だけ呼び出してコンパイルされます。ユーザは、`CC` 環境変数にコンパイラ名を指定することにより、使用するコンパイラを指定できます。`CC` 環境変数が定義されていない場合に `buildobjclient` が呼び出されると、`.c` ファイルと `.cpp` ファイルをコンパイルするために、オペレーティング・システムのデフォルトの C++ 言語コンパイル・コマンドが呼び出されます。

ユーザは、環境変数 `CFLAGS` または `CPPFLAGS` を設定することにより、コンパイラに渡すオプションを指定できます。`CFLAGS` が定義されていない場合に `buildobjclient` が呼び出されると、`buildobjclient` コマンドは `CPPFLAGS` の値 (定義されている場合) を使用します。

オプション

`-v`

`buildobjclient` を冗長モードで機能させます。特に、コンパイル・コマンドをその標準出力に書き出します。

`-o name`

このコマンドで生成するクライアント・アプリケーションの名前を指定します。名前が指定されていない場合、アプリケーション・ファイルの名前は `client<.type>` になります。`type` は拡張子で、アプリケーションのオペレーティング・システムによって異なります。たとえば、UNIX システムでは `type` は付きませんが、Windows システムでは `type` は `.EXE` になります。

`-f firstfile-syntax`

`buildobjclient` コマンドのコンパイルおよびリンク段階で最初に取り込むファイルを指定します。指定されたファイルの後に CORBA ライブラリが取り込まれます。ファイルは、次の表に示すように 2 種類の方法で指定できます。

表 2 最初に取り込むファイルの指定方法

ファイル名の指定方法	説明
<code>-f firstfile</code>	1つのファイルを指定します。
<code>-f "file1.cpp file2.cpp file3.cpp ..."</code>	複数のファイルを指定する場合は、各ファイル名を空白で区切り、リスト全体を引用符で囲みます。

注記 名前に空白が含まれているファイルは指定できません。

注記 `-f` オプションは何度でも指定できます。

`-l lastfile-syntax`

`buildobjclient` コマンドのコンパイルおよびリンク段階で最後に取り込むファイルを指定します。指定されたファイルは、CORBA ライブラリの後に取り込まれます。ファイルは、次の表に示すように2種類の方法で指定できます。

表 3 最後に取り込むファイルの指定方法

ファイル名の指定方法	説明
<code>-l lastfile</code>	1つのファイルを指定します。
<code>-l "file1.cpp file2.cpp file3.cpp ..."</code>	複数のファイルを指定する場合は、各ファイル名を空白で区切り、リスト全体を引用符で囲みます。

注記 `-l` オプションは何度でも指定できます。

`-P`

適切な POA ライブラリをイメージ(つまり、クライアントをサーバとしても動作させるライブラリ)にリンクします。生成されたイメージはサーバとしても動作でき、オブジェクトを作成する際に `Callbacks` ラッパー・クラスを使用できます。ただし、生成された共同クライアント/サーバでは、BEA Tuxedo TP Framework の機能であるオブジェクトの状態管理およびトランザクション管理を使用することができま

せん。クライアントの生成時には、`-P` スイッチが IDL コンパイラに渡される必要があります。`buildobjserver` を使用すると、TP Framework が提供するすべてのサポートに対応したサーバを作成できます。デフォルトでは、サーバ・ライブラリへのリンクは行われません。つまり、デフォルトでは、クライアントのみが作成され、共同クライアント / サーバは作成されません。

`-h` または `-?`

`buildobjclient` コマンドの使用方法に関するヘルプを提供します。それ以外に行われるアクションはありません。

環境変数

`TUXDIR`

クライアント・アプリケーションのコンパイル時に使用する CORBA ライブラリおよびインクルード・ファイルを検索します。

`CC`

`.c` または `.cpp` ファイル拡張子を持つファイルのコンパイルに使用するコンパイラを指定します。この環境変数が定義されていない場合、`.c` および `.cpp` ファイルのコンパイルには、使用しているオペレーティング・システムのデフォルトの C++ 言語コンパイル・コマンドが呼び出されます。

`CFLAGS`

`.c` または `.cpp` ファイル拡張子を持つファイルのコンパイラ・コマンド行の一部として渡される引数を指定します。`CFLAGS` が `buildobjclient` コマンド環境に存在しない場合、`buildobjclient` コマンドは `CPPFLAGS` 環境変数があるかどうかを調べます。

`CPPFLAGS`

注記 `CFLAGS` 環境変数によって渡される引数は、`CPPFLAGS` 変数より優先されます。

`.c` または `.cpp` ファイル拡張子を持つファイルのコンパイラ・コマンド行の一部として渡される引数を指定します。

これは、`buildobjclient` コマンドに自動的に渡されるコマンド行オプション `"-I$(TUXDIR)/include"` (UNIX システムの場合) または `/I%TUXDIR%\include` (Windows システムの場合) に追加されます。`CPPFLAGS` が `buildobjclient` コマンド環境に存在しない場合、コンパイラ・コマンドは追加されません。

LD_LIBRARY_PATH (UNIX システムの場合)

コンパイラが使用する共有オブジェクト、および CORBA ソフトウェアで共有するオブジェクトが入っているディレクトリを指定します。それぞれのディレクトリはコロン (:) で区切ります。一部の UNIX システムでは、別の環境変数が必要となる場合もあります。HP-UX システムでは `SHLIB_PATH` 環境変数を使用します。AIX システムでは `LIBPATH` 環境変数を使用します。

LIB (Windows システムの場合)

ライブラリを検索するディレクトリのリストを指定します。それぞれのディレクトリはセミコロン (;) で区切ります。

移植性 `buildobjclient` コマンドは、クライアントのみの CORBA システムではサポートされません。

使用例 次の例では、Windows システム上で CORBA クライアント・アプリケーションを作成します。

```
set CPPFLAGS=-I%APPDIR%\include
buildobjclient -o empclient.exe -f emp_c.cpp -l userlib1.lib
```

次の例では、c シェルを使用する UNIX システム上で CORBA クライアント・アプリケーションを作成します。

```
setenv CPPFLAGS=$APPDIR/include
buildobjclient -o empclient -f emp_c.cpp -l userlib1.a
```

buildobjserver(1)

名前 buildobjserver - CORBA サーバ・アプリケーションを構築

形式 buildobjserver [-v] [-o name] [-f firstfile-syntax]
[-l lastfile-syntax] [-r rmname][-t]

機能説明 buildobjserver コマンドを使用して、CORBA サーバ・アプリケーションを構築できます。このコマンドは、メイン・ルーチンで `-f` および `-l` オプションによって指定されるファイルと標準の CORBA ライブラリを組み合わせ、サーバ・アプリケーションを形成します。サーバ・アプリケーションは、プラットフォームのデフォルトの C++ コンパイラによって作成されません。

指定された `.c` ファイルおよび `.cpp` ファイルはすべて、使用しているオペレーティング・システムのコンパイル・システムを 1 回だけ呼び出してコンパイルされます。ユーザは、`CC` 環境変数にコンパイラ名を指定することにより、使用するコンパイラを指定できます。`CC` 環境変数が定義されていない場合に `buildobjserver` が呼び出されると、`.c` および `.cpp` ファイルをコンパイルするために、オペレーティング・システムのデフォルトの C++ 言語コンパイル・コマンドが呼び出されます。

ユーザは、環境変数 `CFLAGS` または `CPPFLAGS` を設定することにより、コンパイラに渡すオプションを指定できます。`buildobjserver` の呼び出し時に、`CFLAGS` は定義されていないが、`CPPFLAGS` が定義されている場合、このコマンドは `CPPFLAGS` の値を使用します。

オプション `-v`

`buildobjserver` コマンドを冗長モードで機能させます。特に、コンパイル・コマンドを標準出力に書き出します。

`-o name`

このコマンドで生成するサーバ・アプリケーションの名前を指定します。名前が指定されない場合、アプリケーション・ファイルの名前は `server.type` になります。`type` は拡張子で、アプリケーションで使用中的オペレーティングシステムを示します。たとえば、アプリケーションが `server` を呼び出す場合は UNIX システム、アプリケーションが `server.EXE` を呼び出す場合は Windows NT システムです。

-f firstfile-syntax

`buildobjserver` コマンドのコンパイルおよびリンク段階で最初に (つまり、CORBA ライブラリの前に) 取り込むファイルを指定します。ファイルを指定する 2 種類の方法については、17 ページの「[最初に取り込むファイルの指定方法](#)」を参照してください。

-l lastfile-syntax

`buildobjserver` コマンドのコンパイルおよびリンク段階で最後に (つまり、CORBA ライブラリの後に) 取り込むファイルを指定します。ファイルを指定する 2 種類の方法については、17 ページの「[最後に取り込むファイルの指定方法](#)」を参照してください。

-r rmname

このサーバのリソース・マネージャを指定します。`rmname` は、`$TUXDIR/udataobj/RM` (UNIX システムの場合) または `%TUXDIR%\udataobj\RM` (Windows NT システムの場合) にあるリソース・マネージャ・テーブルに存在するものでなければなりません。このファイルの各エントリは、次のような形式になります。
`rmname:rmstructure_name:library_names`
`rmname` の値を使用することにより、`$TUXDIR/udataobj/RM` または `%TUXDIR%\udataobj\RM` にあるエントリは、リソース・マネージャに関連したライブラリを自動的に取り込み、トランザクション・マネージャとリソース・マネージャとの間にインターフェイスをセットアップします。値 `TUXEDO/SQL` は、BEA Tuxedo System/SQL リソース・マネージャのライブラリを取り込みます。他の値は、リソース・マネージャのテーブルに追加されているものを指定できます。`-r` オプションが指定されていない場合、デフォルトでヌル・リソース・マネージャが使用されます。

-h または -?

`buildobjserver` コマンドの実行時に役立つ情報が記載されたヘルプを表示します。それ以外に行われるアクションはありません。

-t

作成中の CORBA サーバ・アプリケーションでマルチスレッド処理を有効にします。このオプションを指定すると、`UBBCONFIG` ファイルの `MAXDISPATCHTHREADS` パラメータで 1 を超える値を指定することもできます。指定しない場合、CORBA サーバはシングルスレッドのアプリケーションとして動作します。

- 環境変数** TUXDIR
サーバ・アプリケーションのコンパイル時に使用する CORBA ライブラリおよびインクルード・ファイルを検索します。
- CC
-l または -f オプションで指定される、拡張子が .c または .cpp ファイルのコンパイルに使用するコンパイラを指定します。
- CFLAGS
.c または .cpp ファイル拡張子を持つファイルのコンパイラ・コマンド行の一部として渡される引数を指定します。CFLAGS が buildobjserver コマンド環境で使用できない場合、buildobjclient コマンドは CPPFLAGS 環境変数があるかどうかを調べます。
- CPPFLAGS
注記 CFLAGS 環境変数によって渡される引数は、CPPFLAGS 環境変数より優先されます。
- 拡張子が .c または .cpp のファイルのコンパイラ・コマンド行の一部として渡される引数を指定します。このオプションは、
-I\$TUXDIR/include コマンド行オプション (UNIX システム) または
/I%TUXDIR%\include コマンド行オプション (Windows NT システム)
に追加して使用し、buildobjserver コマンドによって自動的に渡されます。CPPFLAGS が buildobjserver コマンド環境で使用できない場合、コンパイラ・コマンドは追加されません。
- LD_LIBRARY_PATH (UNIX システムの場合)
コンパイラが使用する共用オブジェクト、および CORBA の共用オブジェクトが入っているディレクトリを一覧表示します。ディレクトリ名はコロン (:) で区切ります。一部の UNIX システムでは、別の環境変数が必要となる場合もあります。HP-UX システムでは SHLIB_PATH 環境変数を使用します。AIX システムでは LIBPATH 環境変数を使用します。
- LIB (Windows NT システム)
ライブラリにあるディレクトリを一覧表示します。ディレクトリ名はセミコロン (;) で区切ります。
- 移植性** buildobjserver コマンドは、クライアントのみの CORBA システムではサポートされません。

使用例 次の例では、`emp_s.cpp` および `emp_i.cpp` ファイルを使用して、UNIX システム上で CORBA サーバ・アプリケーションを作成します。

```
buildobjserver -r TUXEDO/SQL -o unobserved
                -f "emp_s.cpp emp_i.cpp"
```

次の例は、`buildobjserver` コマンドで環境変数 `CC` および `CFLAGS` を使用する方法を示しています。また、`-f` および `-lm` オプションを使用して、Bourne または Korn シェルで数学ライブラリをリンクする方法も示しています (UNIX システム)。

```
CFLAGS=-g CC=/bin/cc \
buildobjserver -r TUXEDO/SQL -o TLR -f TLR.o -f util.o -l -lm
```

次の例は、リソース・マネージャを指定せずに UNIX システム上で `buildobjserver` コマンドを使用する方法を示しています。

```
buildobjserver -o PRINTER -f PRINTER.o
```

RM ファイルの例 以下では、サポートされている各オペレーティング・システム・プラットフォーム用の RM ファイルの例を示します。

Windows NT

```
Oracle_XA:xaosw:C:\Orant\rdbms73\xa\xa73.lib
          C:\Orant\pro22\lib\msvc\sqllib18.lib
```

UNIX

```
Oracle_XA:xaosw:-L$ORACLE_HOME/rdbms/lib
                -L$ORACLE_HOME/precomp/lib -lc
                -L/home4/m01/app/oracle/product/7.3.2/lib -lsql -lclntsh
                -lsqlnet -lncr -lcommon -lgeneric -lepc -lnlsrtl3 -lc3v6
                -lcore3 -lsocket -lnsl -lm -ldl -lthread
```

Digital UNIX

```
Oracle_XA:xaosw:-L${ORACLE_HOME}/lib -lxa
                ${ORACLE_HOME}/lib/libsql.a -lsqlnet -lncr -lsqlnet
                ${ORACLE_HOME}/lib/libclient.a -lcommon -lgeneric -lsqlnet
                -lncr -lsqlnet ${ORACLE_HOME}/lib/libclient.a -lcommon
                -lgeneric -lepc -lepcpt -lnlsrtl3 -lc3v6 -lcore3
                -lnlsrtl3 -lcore3 -lnlsrtl3 -lm
```

AIX

セクション 1 - コマンド

```
Oracle_XA:xaosw:-L${ORACLE_HOME}/lib -lxa -lsql -lsqlnet  
-lncr -lclient -lcommon -lgeneric -lepc -lnlsrtl3 -lc3v6  
-lcore3 -lm -lld
```

HP-UX : Oracle 8.04

```
Oracle_XA:xaosw:-L${ORACLE_HOME}/lib -lclntsh
```

buildserver(1)

名前 `buildserver` - BEA Tuxedo ATMI のサーバのロード・モジュールを構築

形式 `buildserver [-C] [-s { @filename | service[,service . . .]
[:func] | :func }] [-v] [-o outfile] [-f firstfiles]
[-l lastfiles] [{-r|-g} rname] [-k] [-t]`

機能説明 `buildserver` は、BEA Tuxedo ATMI のサーバのロード・モジュールを構築するときを使用します。このコマンドは、`-f` および `-l` オプションで指定されるファイルと標準のサーバのメイン・ルーチンおよび標準 BEA Tuxedo ATMI ライブラリを組み合わせて、ロード・モジュールを作成します。このロード・モジュールは、`buildserver` が呼び出す `cc(1)` コマンドによって作成されます (UNIX システムのリファレンス・マニュアルの `cc(1)` を参照してください)。`buildserver` のオプションにはそれぞれ、次のような意味があります。

`-v`

`buildserver` を冗長モードで動作させます。特に、`cc` コマンドをその標準出力に書き出します。

`-o outfile`

出力されるロード・モジュールを収めるファイルの名前を指定します。このファイル名を指定しない場合、ロード・モジュールの名前は `SERVER` になります。

`-f firstfiles`

`buildserver` のコンパイルおよびリンク段階で最初に取り込む 1 つ以上のファイルを指定します。指定されたファイルが取り込まれた後で BEA Tuxedo ATMI ライブラリが取り込まれます。複数のファイルを指定する場合は、各ファイル名を空白で区切り、リスト全体を引用符で囲まれなければなりません。このオプションは何回指定してもかまいません。コンパイラ・オプションおよび引数を含むように指定する場合は、以下に説明する `CFLAGS` および `ALTCFLAGS` 環境変数を使用します。

`-l lastfiles`

`buildserver` のコンパイルおよびリンク段階で最後に取り込む 1 つ以上のファイルを指定します。指定されたファイルは、BEA Tuxedo ATMI ライブラリが取り込まれた後で取り込まれます。複数のファイ

ルを指定する場合は、各ファイル名を空白で区切り、リスト全体を引用符で囲まれなければなりません。このオプションは何回指定してもかまいません。

`-r rmname`

このサーバのリソース・マネージャを指定します。*rmname* は、`$TUXDIR/udataobj/RM` にあるリソース・マネージャのテーブルにあるものでなければなりません。このファイルの各行は次のような形式になります。

```
rmname:rmstructure_name:library_names
```

詳細については、[buildtms\(1\)](#) を参照してください。*rmname* の値を使用することにより、`$TUXDIR/udataobj/RM` にあるエントリは、リソース・マネージャに関連したライブラリを自動的に含み、トランザクション・マネージャとリソース・マネージャ間のインターフェイスを正しく設定するのに使用されます。他の値は、リソース・マネージャのテーブルに追加されているものを指定できます。`-r` オプションの指定がない場合、デフォルトの設定としてヌル・リソース・マネージャが使用されます。[UBBCONFIG\(5\)](#) のリファレンス・ページを参照してください。

`-s { @filename | service[,service...][:func] | :func }`

サーバのブート時に宣言できるサービスの名前を指定します。サービス名および暗黙のファンクション名は 15 文字以下でなければなりません。明示的関数名 (コロンの後に指定する名前) は、128 文字まで使用できます。この文字数より長い名前が指定された場合は、警告メッセージが表示されて短縮されます。[tmadmin\(1\)](#) または [TM_MIB\(5\)](#) によりファイルを取得した場合は、名前の最初の 15 文字だけが表示されます ([servopts\(5\)](#) を参照してください)。サービスに関連付けることのできるすべての関数を、このオプションで指定する必要があります。ほとんどの場合、サービスは同じ名前を持つ関数によって実行されます。つまり、*x* サービスは関数 *x* によって実行されます。たとえば、次のように指定すると、サービス *x*、*y*、および *z* を提供するサーバが構築されます。これらのサービスはそれぞれ同じ名前の関数によって処理されます。

```
-s x,y,z
```

その他のケースでは、異なる名前の関数でサービスが実行されることもあります。次のように指定すると、サービス *x*、*y*、および *z* を提供

するサーバが構築されます。これらのサービスはそれぞれ関数 `abc` によって処理されます。

```
-s x,y,z:abc
```

カンマとカンマの間に空白を入れてはいけません。関数名の前にはコロンを付けます。別のケースでは、実行時までサービス名が分からないことがあります。関連するサービスを持つ可能性のあるすべての関数を、`buildserver` で指定しなければなりません。サービス名がマップされている可能性のある関数を指定するには、関数名の前にコロンを付けます。たとえば、次のように指定すると、サービスが関連付けられている可能性のある関数 `pqr` によってサーバが構築されます。

`tpadvertise(3c)` は、関数 `pqr` にサービス名をマップするために使用されます。

```
-s :pqr
```

`-s` オプションでファイル名を指定するには、ファイル名の前に「@」文字を付けます。このファイルの各行は、`-s` オプションの引数と見なされます。このファイルには、コメントを入れることができます。コメント行の先頭には「#」文字を置きます。このファイルは、サービス名がマップされている可能性のある、サーバ中のすべての関数を指定するのに使用できます。

`-s` オプションは何回使用してもかまいません。「.」文字で始まるサービスはシステムで使用するよう予約されているため、`-s` オプションを指定してそのようなサービスをサーバに組み込もうとすると、`buildserver` は異常終了します。

-C

COBOL のコンパイルを指定します。

`buildserver` は通常、`cc` コマンドを使用して `a.out` を生成します。代替コンパイラを指定できるようにするため、`buildserver` はシェル変数 `CC` が存在するかどうかを調べます。`CC` が `buildserver` の環境にない場合、または変数の値がヌル文字列 "" である場合は、`buildserver` はコンパイラとして `cc` を使用します。環境内に `CC` が存在する場合、実行されるコンパイラの名前が `CC` の値となります。同様に、シェル変数 `CFLAGS` も、コンパイラへ渡す一連のパラメータを取り込むためにチェックされます。

-k

サーバの main スタブを保持します。buildserver は、サービス・テーブルなどのデータ構造と main() 関数を持つ main スタブを生成します。通常これは、サーバの構築時に、コンパイルの後で削除されます。このオプションは、ソース・ファイルを保持する必要があることを示します (ソース・ファイル名を表示するには、-v オプションを使用します)。

注記 このファイルの生成内容は、リリースによって変更される場合があります。このファイルで公開されているデータ構造およびインターフェイスを重視しないでください。このオプションは、構築の問題のデバッグを支援するためのものです。

-t

マルチスレッド処理を指定します。サーバをマルチスレッド操作で使用する場合は、必ずこのオプションを指定してください。コンフィギュレーション・ファイルの MAXDISPATCHTHREADS に 1 より大きな値が設定されている場合に、このオプションを指定せずにサーバを起動しようとすると、ユーザ・ログに警告メッセージが出力され、サーバはシングルスレッド操作に戻ります。

このオプションは、管理者がスレッド・セーフな方法でプログラミングされていないサーバをマルチスレッド・サーバとして起動するのを防止するために使用されます。

環境変数 TUXDIR

buildserver は、環境変数 TUXDIR を使用して、サーバ・プロセスのコンパイル時に使用する BEA Tuxedo ATMI ライブラリおよびインクルード・ファイルを見つけます。

CC

buildserver は通常、デフォルトの C 言語コンパイル・コマンドを使用してサーバ実行可能コードを生成します。デフォルトの C 言語コンパイル・コマンドは、サポートされているオペレーティング・システムごとに定義されており、UNIX システムの場合は cc(1) です。代替コンパイラを指定できるようにするため、buildserver は環境変数 CC が存在するかどうかを調べます。CC が buildserver の環境に存在しない場合、またはこの環境変数が文字列 "" である場合、buildserver はデフォルトの C 言語コンパイラを使用します。CC が環境に存在すれば、実行するコンパイラの名前としてその値が使用されます。

CFLAGS

環境変数 CFLAGS は、コンパイラ・コマンド行の一部として引き渡される引数のセットを指定するときに使用します。これは、`buildserver` に自動的に渡されるコマンド行オプション `"-I${TUXDIR}/include"` に追加されます。CFLAGS が `buildserver` の環境に存在しない場合、またはこの環境変数が文字列 "" である場合、`buildserver` はコンパイラ・コマンド行の引数を追加しません。

ALTCC

`-c` オプションを使って COBOL のコンパイルを指定すると、`buildserver` は通常、BEA Tuxedo のシェル `cobcc(1)` を使用します。すると、`cobcc` は `cob` を呼び出し、クライアント実行可能コードを生成します。`buildserver` は、別のコンパイラを指定する ALTCC という環境変数の有無をチェックします。ALTCC が `buildserver` の環境に存在しない場合、または文字列 "" である場合、`buildserver` は `cobcc` を使用します。環境に ALTCC が存在する場合は、その値をとって実行するコンパイラ・コマンドとします。

注記 Windows システムでは、環境変数 ALTCC および ALTCFLAGS は使用できません。これらを設定すると予期しない結果が生じます。まず COBOL コンパイラを使用してアプリケーションをコンパイルして、次に生成されたオブジェクト・ファイルを `buildserver(1)` コマンドに渡す必要があります。

ALTCFLAGS

環境変数 ALTCFLAGS には、`-c` オプションを指定した場合に、COBOL コンパイラ・コマンドの一部として渡す追加の引数を指定します。これは、`buildserver` に自動的に渡されるコマンド行オプション `"-I${TUXDIR}/include"` に追加されます。`-c` オプションを使用する場合、コンパイラ・オプションやその引数を `buildserver -f` オプションで指定すると、エラーが発生しますので、ALTCFLAGS を使用する必要があります。設定しなかった場合は、上記の CFLAGS と同じ値に設定されます。

注記 ALTCC 環境変数の下の注記を参照してください。

COBOPT

環境変数 COBOPT には、`-c` オプションを指定した場合に、COBOL コンパイラが使用する追加の引数を指定します。

COBCPY

環境変数 COBCPY には、`-c` オプションを指定した場合に、COBOL コンパイラが使用する COBOL コピー・ファイルが存在するディレクトリを指定します。

LD_LIBRARY_PATH (UNIX システムの場合)

環境変数 LD_LIBRARY_PATH には、BEA Tuxedo システムの共有オブジェクトに加えて、COBOL コンパイラが使用する共有オブジェクトが存在するディレクトリを指定します。一部の UNIX システムでは、別の環境変数が必要となる場合もあります。HP-UX システムでは SHLIB_PATH 環境変数を使用します。AIX システムでは LIBPATH 環境変数を使用します。

LIB (Windows NT システム)

ライブラリを検索するディレクトリのリストを指定します。それぞれのディレクトリはセミコロン (;) で区切ります。

互換性 以前のリリースでは、`sql` すなわち `database` の `genoption` を指定するのに `-g` オプションを使用できました。上位互換性を保持するため、このオプションは `-r` オプションの同義語になっています。

移植性 `buildserver` コンパイル・ツールは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで使用できます。RM XA ライブラリは、Windows プラットフォームではサポートされていません。

注意事項 コンパイル・システムによっては、`main()` へのコードの追加が必要になる場合があります。たとえば、C++ でコンストラクタを初期化したり、COBOL 用のライブラリを初期化するような場合です。サーバの `main()` のすべての変数宣言の直後で、すべての実行文の前に、アプリケーションのコードをインクルードするための一般的な機構が用意されています。これによって、アプリケーションは変数の宣言と文の実行を 1 ブロックのコードで行うことができます。アプリケーション `exit` は、次のように定義できます。
`#ifdef TMMAINEXIT #include "mainexit.h" #endif` この機能を使用するためには、環境変数 `ALTCFLAGS` (COBOL の場合) または `CFLAGS` (C の場合) に `"-DTMMAINEXIT"` を指定し、カレント・ディレクトリに `mainexit.h` を置く (または `-I` インクルード・オプションを使用して他のディレクトリからインクルードする) 必要があります。

たとえば、Micro Focus Cobol V3.2.x で PRN 番号の最後の数字が 11.03 より大きい場合、共有ライブラリの使用時には、main() のすべての COBOL ルーチンの中で（おそらく関数プロトタイプ宣言の後で）cobinit() を呼び出す必要があります。これを行うには、cobinit() の呼び出しが入った mainexit.h を作成し、上記の手順に従ってください。

使用例 次の例は、リソース・マネージャ (-r TUXEDO/SQL) ライブラリを buildserver コマンド行で指定する方法を示しています。

```
buildserver -r TUXEDO/SQL -s OPEN_ACCT -s CLOSE_ACCT -o ACCT
-f ACCT.o -f appinit.o -f util.o
```

次の例は、buildserver に変数 CC および CFLAGS 変数を与える方法、および -f を使用して CC 行への -lm オプションを指定して数学ライブラリをリンクする方法を示したものです。

```
CFLAGS=-g CC=/bin/cc buildserver -r TUXEDO/SQL -s DEPOSIT
-s WITHDRAWAL -s INQUIRY -o TLR -f TLR.o -f util.o -f -lm
```

次の例は、リソース・マネージャを指定せずに buildserver コマンドを使用する方法です。

```
buildserver -s PRINTER -o PRINTER -f PRINTER.o
```

次の例は、COBOL のコンパイルです。

```
COBCPY=$TUXDIR/cobinclude COBOPT="-C ANS85 -C ALIGN=8 -C NOIBMCOMP
-C TRUNC=ANSI -C OSEXT=cbl" COBDIR=/usr/lib/cobol
LD_LIBRARY_PATH=$COBDIR/coblib export COBOPT COBCPY COBDIR
LD_LIBRARY_PATH buildserver -C -r TUXEDO/SQL -s OPEN_ACCT
-s CLOSE_ACCT -o ACCT -f ACCT.o -f appinit.o -f util.o
```

関連項目 [builtdtms\(1\)](#)、[servopts\(5\)](#)、[UBBCONFIG\(5\)](#)

お使いのオペレーティング・システムのリファレンス・マニュアルで説明する C コンパイラとリンクに関する記述

buildtms(1)

名前	<code>buildtms</code> - トランザクション・マネージャ・サーバのロード・モジュールを構築
形式	<code>buildtms [-v] -o name -r rm_name</code>
機能説明	<code>buildtms</code> はトランザクション・マネージャのロード・モジュールを構築するときに使用します。

BEA Tuxedo システムではいくつかの TM サーバと一緒に提供されますが、分散トランザクション処理用に BEA Tuxedo システムとともに使用する新しいリソース・マネージャを提供することもできます。このようなリソース・マネージャは、X/Open の XA インターフェイスに準拠していなければなりません。リソース・マネージャ・ベンダは、次の 4 つの項目を提供しなければなりません。1) リソース・マネージャの名前、リソース・マネージャの機能を示すフラグ、および実際の XA の関数ポインタを収める `xa_switch_t` 型の構造体の名前、2) `xa_switch_t` 構造体の `name` 要素に格納されるリソース・マネージャの名前、3) XA インターフェイスのサービスを提供しソフトウェアをサポートするオブジェクト・ファイルの名前、そして 4) `UBBCONFIG` コンフィギュレーション・ファイルのパラメータ `OPENINFO` および `CLOSEINFO` に提供される情報文字列の形式、の 4 項目です。[UBBCONFIG\(5\)](#) を参照してください。

新しいリソース・マネージャを BEA Tuxedo システムに統合化する際には、ファイル `$TUXDIR/udataobj/RM` を更新して、そのリソース・マネージャに関する情報を組み込むようにしなければなりません。このファイルの形式は次のとおりです。

```
rm_name:rm_structure_name:library_names
```

ここで、`rm_name` はリソース・マネージャの名前、`rm_structure_name` は `xa_switch_t` 構造体の名前、および `library_names` はリソース・マネージャのオブジェクト・ファイルのリストです。これらの値の前後には空白類（タブやスペース）を入れることができ、`library_names` の中に埋め込んでかまいません。値の中にコロン（`:`）を含めることはできません。また、シャープ（`#`）で始まる行はコメントとして扱われ、無視されます。

新しいリソース・マネージャのトランザクション・マネージャは `buildtms` を使用して構築し、`$TUXDIR/bin` にインストールしなければなりません。
`buildtms` は `buildserver(1)` コマンドを使用して `a.out` を生成します。
`buildtms` コマンドには、以下のオプションがあります。

- v
`buildtms` を冗長モードで動作させます。具体的には、`buildserver` コマンドをその標準出力に出力し、`buildserver` に `-v` オプションを指定します。
- o *name*
出力されるロード・モジュールを収めるファイルの名前を指定します。
- r *rm_name*
このサーバのリソース・マネージャを指定します。この値 *rm_name* は、`$TUXDIR/udataobj/RM` にあるリソース・マネージャ・テーブルに登録されていなければなりません。*rm_name* の値に対応するエントリは、リソース・マネージャに対する正しいライブラリを自動的に組み込み、かつトランザクション・マネージャとリソース・マネージャとの間のインターフェイスを正しく設定 (`xa_switch_t` 構造体を使用して) するときを使用します。

`buildtms` は、`buildserver` コマンドを使用して `a.out` を生成します。
`buildserver` は、コンパイラおよびコンパイラ・フラグに関してそれぞれ `cc` および `CFLAGS` 環境変数を使用します (設定されていれば)。詳細については、`buildserver(1)` を参照してください。

移植性 `buildtms` は、BEA Tuxedo ATMI または CORBA サーバ環境をサポートするすべてのプラットフォームで、BEA Tuxedo システム提供のコンパイル・ツールとしてサポートされています。RM XA ライブラリは、Windows プラットフォームではサポートされていません。

使用例 `buildtms -o $TUXDIR/bin/TMS_XYZ -r XYZ/SQL # TMS for XYZ resource manager`

関連項目 `buildserver(1)`、`UBBCONFIG(5)`

buildwsh(1)

名前	<code>buildwsh</code> - カスタマイズされたワークステーション・ハンドラ・プロセスの構築
形式	<code>buildwsh [-v] [-o name] [-f files]</code>
機能説明	<p><code>buildwsh</code> は、カスタマイズされた BEA Tuxedo ATMI ワークステーション・ハンドラ・モジュールを作成するために使用されます。そのファイルには、アプリケーション・バッファ・タイプ・スイッチと必要なサポート・ルーチンだけを含むようにします。このコマンドは、<code>-f</code> オプションで指定されるファイルと、ワークステーション・ハンドラ・ロード・モジュールを形成するために必要な標準の BEA Tuxedo ATMI ライブラリを組み合わせます。このロード・モジュールは UNIX システムのリファレンス・マニュアルの <code>cc(1)</code> コマンドによって構築されます。この <code>cc</code> コマンドは、<code>buildwsh</code> によって実行されます。このコマンドのオプションは、次のとおりです。</p> <p><code>-v</code></p> <p><code>buildwsh</code> を冗長モードで動作させます。<code>cc</code> コマンドの実行結果が、標準出力へ書き込まれます。</p> <p><code>-o name</code></p> <p>出力されるワークステーション・ハンドラ・ロード・モジュールのファイル名を指定します。ここで指定した名前は、コンフィギュレーション・ファイル内の <code>SERVER</code> セクションにおける <code>WSL(5)</code> サーバの <code>-w WSHname</code> オプションで指定しなければなりません。指定しない場合、ロード・モジュールには、<code>WSH</code> という名前が付きます。</p> <p><code>-f firstfiles</code></p> <p><code>buildwsh</code> のコンパイル段階やリンク段階で取り込む 1 つ以上のユーザ・ファイルを指定します。ソース・ファイルは、<code>cc</code> コマンドまたは環境変数 <code>CC</code> によって指定されたコマンドによってコンパイルされます。ソース・ファイルのコンパイルによって生成されたオブジェクト・ファイルと、<code>-f</code> オプションのパラメータとして直接指定されたオブジェクト・ファイルは、基本となるワークステーション・ハンドラ・プロセスを構築するために必要なすべてのオブジェクト・ファイルの後で、かつ BEA Tuxedo ATMI ライブラリの前に取り込まれます。複数のファイルを指定する場合は、各ファイル名を空白で区切り、リスト全体を引用符で囲まなければならない。このオプションは、何回も指定することができます。</p>

buildwsh は通常、cc コマンドを使用して a.out を生成します。代替コンパイラを指定できるようにするために、buildwsh はシェル変数 CC が存在するかどうかを調べます。CC が buildwsh の環境に存在しない場合、またはこの変数が文字列 "" である場合、buildwsh は cc をコンパイラとして使用します。環境内に CC が存在する場合、実行されるコンパイラの名前が CC の値となります。同様に、シェル変数 CFLAGS も、コンパイラへ渡すべき一連のパラメータを取り込むためにチェックされます。

アプリケーションが共有ライブラリを使用する場合は、ここで説明したコンパイルとリンクの作業をする必要はありません。詳細については、『C 言語を使用した BEA Tuxedo アプリケーションのプログラミング』の「型付きバッファの管理」を参照してください。

移植性 buildwsh コンパイル・ツールは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで使用できます。

使用例

```
CC=ncc CFLAGS="-I $TUXDIR/include"; export CC CFLAGS buildwsh
-o APPWSH -f apptypsw.o
```

関連項目 [buildclient\(1\)](#), [WSL\(5\)](#)

UNIX システムのリファレンス・マニュアルの [cc\(1\)](#)、[ld\(1\)](#)

cobcc(1)

名前 cobcc - COBOL コンパイル・インターフェイス

形式 cobcc [option . . .] filename . . .

機能説明 cobcc は、COBOL コンパイラに対するインターフェイス・シェルとして使
用します。cobcc は、-c (COBOL) オプションを指定して [buildclient\(1\)](#) ま
たは [buildserver\(1\)](#) が実行された場合にはデフォルトで呼び出されます。
ALTCC 環境変数を指定すると無効になります。

以下に cobcc で指定可能なオプションを示します。これらのオプションを使
用するには、buildclient や buildserver を実行する際に、環境変数
ALTCFLAGS を cobcc が認識できるオプション文字列に設定します。各オプ
ションの詳細については、COBOL と C コンパイラのドキュメントを参照し
てください。

注記 Windows システムでは、環境変数 ALTCC および ALTCFLAGS は使用で
きません。これらを設定すると予期しない結果が生じます。まず
COBOL コンパイラを使用してアプリケーションをコンパイルし、次
に生成されたオブジェクト・ファイルを [buildclient\(1\)](#) または
[buildserver\(1\)](#) コマンドに渡す必要があります。

cc や cob とは異なり cobcc では、すべてのオプションはファイル名の前に置
かれなければなりません。

-c
リンク・フェイズを抑制します。つまりコンパイルは行われますが、
実行可能プログラムは作成されません。

-p -g -r -O
これらのオプションは直接 COBOL コンパイラに渡されます。

-l argument
オプションと引数は、直接 COBOL コンパイラに渡されます (両者を
分離する空白文字は除きます)。

-L argument
オプションと引数は、直接 COBOL コンパイラに渡されます (両者を
分離する 1 つの空白文字も含みます)。

- `-o output_file`
リンク・ステージからの出力である実行可能ファイルの名前を指定します。
- `-E -P -S`
COBOL コンパイラを介して C コンパイラに渡されます。またリンク・ステージを抑制します。
- `-A -C -H -f -G`
COBOL コンパイラを介して C コンパイラに渡されます。
- `-w`
COBOL と C コンパイラの警告メッセージを抑制します。
- `-D argument`
このオプションと引数は、COBOL コンパイラを介して C コンパイラに渡されます。C 言語でのマクロ定義に使用します。
- `{-T -Y -U -I -B -X -F -q} argument`
各オプションは引数として扱われます。オプションと引数は COBOL コンパイラを介して C コンパイラに渡されます。
- `-V -v`
各オプションは COBOL コンパイラと C コンパイラに渡されます。
- `-a -s`
各オプションはローダに渡されます。
- `-u argument`
このオプションと引数はローダに渡されます。
- `-W argument`
argument は、カンマで区切られた最大 3 個のフィールドで構成します。引数の最初の部分が `-p` または `-o` であれば、C コンパイラに渡されます。`-a` で始まる場合には、アセンブラに渡されます。`-l` で始まる場合にはローダに渡されます。`-c` で始まるのであれば、COBOL コンパイラに渡されます。それ以外の場合には、C コンパイラに渡されません。

オプションと引数、およびファイル名は該当するオプションと共に COBOL コンパイラに渡され、適切な情報が COBOL コンパイラ、C コンパイラ、あるいはローダで処理されます。COBOL コンパイラの名前は "cob" と想定されており、PATH で定義されています。

関連項目 [buildclient\(1\)](#)、[buildserver\(1\)](#)

UNIX システムのリファレンス・マニュアルの [cc\(1\)](#)

『Micro Focus COBOL/2 Operating Guide』 Micro Focus Ltd.

dmdadmin(1)

名前 dmdadmin - BEA Tuxedo Domains 管理プログラムのコマンド・インタプリタ

形式 dmdadmin [-c]

機能説明 dmdadmin は対話式のコマンド・インタプリタで、Domains コンフィギュレーションで呼び出される特定の BEA Tuxedo アプリケーション用に定義されたドメイン・ゲートウェイ・グループを管理するために使用します。ここでは、BEA Tuxedo Domains コンポーネントの TDomain ゲートウェイ、TOP END Domain Gateway (TEDG)、SNA Gateway (SNAX)、および OSI TP ゲートウェイで dmdadmin を使用する方法について説明します。BEA Tuxedo Domains コンポーネントについては、『[BEA Tuxedo Domains コンポーネント](#)』を参照してください。

dmdadmin には、管理モードとコンフィギュレーション・モードの2つのモードがあります。

dmdadmin をパラメータなしで呼び出すと管理モードになります。デフォルトのモードは管理モードです。このモードでは、dmdadmin をアクティブなアプリケーション内のアクティブなノード(ワークステーションを除く)上で実行できます。アプリケーション管理者は、このモードを使用して、アクティブなドメイン・ゲートウェイ・グループのパラメータを取得または変更できます。また、アプリケーション管理者は、このモードを使用して、特定のローカル・ドメイン・アクセス・ポイント用の Domains トランザクション・ログを作成、破棄、または再初期化できます。このケースでは、そのローカル・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・グループをアクティブにはなりません。また、dmdadmin は対応するゲートウェイ・グループに割り当てられたマシン上で実行する必要があります。

dmdadmin は、-c オプション付きで呼び出されるか、config サブコマンドが呼び出されたときにコンフィギュレーション・モードになります。アプリケーション管理者は、このモードを使用して Domains コンフィギュレーション・ファイル(BDMCONFIG)のバイナリ・バージョンのコンフィギュレーション情報を更新したり、新しいコンフィギュレーション情報を追加できます。

dmdadmin では、Domains 管理サーバ(DMADM)を使用して BDMCONFIG ファイルの管理を行い、ゲートウェイ管理サーバ(GWADM)を使用してアクティブな DOMAIN ゲートウェイ・グループ(Domains コンフィギュレーションで起動

される BEA Tuxedo アプリケーションにつき 1 つの `DMADM` プロセスが実行され、1 つのドメイン・ゲートウェイ・グループにつき 1 つの `GWADM` プロセスが実行されます) の再構成を行う必要があります。

管理モード・
コマンド

`dmadmin` が起動すると、プロンプト (`&dlq;>&drq;`) “式でコマン” きます。

コマンド [引数]

共通して出現するいくつかの引数には、`default` コマンドを使用してデフォルト値が与えられます。`default` コマンドによって設定されたパラメータを受け付けるコマンドは、`default` に値が設定されているかどうかを調べます。値が設定されていない場合は、エラー・メッセージが返されます。

一度設定したデフォルト値は、`default` コマンドで変更しない限り、セッション終了まで有効です。このデフォルト値は、コマンド行で明示的に値を入力して変更するか、値 * (アスタリスク) を入力して設定解除します。こうして変更した値は、コマンドの単一のインスタンスの間有効です。

`dmadmin` コマンドからの出力は、使用しているページ別表示コマンドに従ってページ別に表示されます (後述の `paginate` サブコマンドを参照してください)。

コマンドの入力形式には、完全な名前でも省略形でも使用でき、該当する引数はコマンドの後に付きます。この後の一覧では、完全な名前の後の丸かっこ内に省略形を示しています。角かっこ [] で囲まれた引数は省略可能です。中かっこ { } で囲まれた引数は相互に排他的な選択肢を示します。多くの場合、`local_domain_access_point_name` は必須の引数であり、これは `default` コマンドを使用して設定することができます。

以下のコマンドは、管理モードで使用できます。

```
advertise (adv) -d local_domain_access_point_name [{service}]
```

指定されたローカル・ドメイン・アクセス・ポイントが提供するすべてのリモート・サービス、または指定されたリモート・サービスを宣言します。

```
audit (audit) -d local_domain_access_point_name [{off | on}]
```

指定されたローカル・ドメイン・アクセス・ポイントに対して、監査追跡をアクティブ (`on`) にまたはアクティブ解除 (`off`) します。オプションが設定されていない場合は、現在の設定値が `on` と `off` で切り替わり、新しい設定が表示されます。初期設定は `off` です。

BEA Tuxedo 8.0 以降のソフトウェアでマルチ・ドメイン・トランザクションを作成した場合は、Domains トランザクション監査機能によって自動的に、リモート (親) アプリケーションからローカル (従属) ア

アプリケーションの監査ログに、ローカル GTRID に加えてグローバル・トランザクション ID (GTRID) が書き込まれます。

監査レコードには、プロセス ID、ローカル・ドメイン・アクセス・ポイント名、リモート・ドメイン・アクセス・ポイント名、サービス名、ローカル GTRID (トランザクション・モードの場合のみ)、親 GTRID (トランザクション・モードの場合のみ)、監査レコードの型 (文字列)、および現在のタイムスタンプの順に、コロンで区切った文字列フィールドが並びます。

`chbktime (chbt) -d local_domain_access_point_name -t bktime`

特定のローカル・ドメイン・アクセス・ポイントのブロッキング・タイムアウトを変更します。

`config (config)`

コンフィギュレーション・モードに入ります。このモードで発行されたコマンドは、この後で説明する [46 ページの「コンフィギュレーション・モードのコマンド」](#) で定義された規則に従います。

`connect (co) -d local_domain_access_point_name`

`[-R remote_domain_access_point_name]`

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します。ローカル・ドメイン・ゲートウェイに接続の再試行が設定されている場合は接続が失敗しても、自動接続再試行プロセスによって接続試行が繰り返されます。`-R` が指定されていない場合、このローカル・ドメイン・ゲートウェイに設定されているすべてのリモート・ドメイン・アクセス・ポイントに対してコマンドが適用されます。

`crdmllog (crdlog)[-d local_domain_access_point_name]`

現在のマシン (`dmadmin` が動作しているマシン) 上の指定されたローカル・ドメイン・アクセス・ポイントに対して、Domains トランザクション・ログを作成します。このコマンドでは、`DMCONFIG` ファイルで指定したパラメータが使用されます。指定のローカル・ドメイン・アクセス・ポイントに関連付けられているドメイン・ゲートウェイ・グループが現在のマシンでアクティブであるか、ログがすでに存在する場合は、このコマンドは失敗します。

`default (d) [-d local_domain_access_point_name]`

デフォルトのローカル・ドメイン・アクセス・ポイントにするための、対応する引数を設定します。デフォルトの設定は、引数として * (

アスタリスク)を指定すると設定を解除できます。default コマンドを引数なしで入力した場合は、現在のデフォルトの値が表示されます。

```
disconnect (dco) -d local_domain_access_point_name
[-R remote_domain_access_point_name]
```

ローカル・ドメイン・ゲートウェイとリモート・ドメイン・ゲートウェイ間の接続を切断します。接続再試行プロセスは開始されません。アクティブな接続がない場合でも、自動接続再試行プロセスが有効になっていると自動再試行プロセスは停止されます。-R が指定されていない場合、このローカル・ドメイン・ゲートウェイに設定されているすべてのリモート・ドメイン・アクセス・ポイントに対してコマンドが適用されます。

```
dsdmlog (dsdlg) -d local_domain_access_point_name [-y]
```

現在のマシン (dmadmin が動作しているマシン) 上の指定されたローカル・ドメイン・アクセス・ポイントに対して、Domains トランザクション・ログを破棄します。このローカル・ドメイン・アクセス・ポイントに Domains トランザクション・ログが定義されていないか、ローカル・ドメイン・アクセス・ポイントに関連付けられているドメイン・ゲートウェイ・グループがアクティブであるか、または未終了のトランザクション・レコードがログ内に存在している場合は、エラーが返されます。「未終了のトランザクション」とは、グローバル・トランザクションが既にコミットされているが、トランザクションの終了がまだ書き込まれていないトランザクションのことです。このコマンドは、-y オプションを指定していない場合に確認のためのプロンプトを表示します。

```
echo (e) [{off | on}]
```

on に設定すると、入力コマンド行をエコーします。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。初期設定は off です。

```
forgettrans (ft) -d local_domain_access_point_name [-t tran_id]
```

指定のローカル・ドメイン・アクセス・ポイントの1つまたはすべてのヒューリスティック・ログを破棄します。トランザクション識別子 tran_id が指定されている場合は、そのトランザクションのヒューリスティック・ログ・レコードのみが破棄されます。トランザクション識別子 tran_id は、printtrans コマンドか、ULOG ファイルから取得できます。

`help (h) [command]`

ヘルプ・メッセージを表示します。`command` を指定すると、そのコマンドについて、省略形、引数、および説明が表示されます。引数をすべて省略すると、すべてのコマンドの構文が表示されます。

`indmlog (indlg) -d local_domain_access_point_name [-y]`

現在のマシン (`dmadmin` が動作しているマシン) 上の指定されたローカル・ドメイン・アクセス・ポイントに対して、Domains トランザクション・ログを再初期化します。このローカル・ドメイン・アクセス・ポイントに `DMTLOG` が定義されていないか、ローカル・ドメイン・アクセス・ポイントに関連付けられているドメイン・ゲートウェイ・グループがアクティブであるか、または未終了のトランザクション・レコードがログ内に存在している場合は、エラーが返されます。「未終了のトランザクション」とは、グローバル・トランザクションが既にコミットされているが、トランザクションの終了がまだ書き込まれていないトランザクションのことです。このコマンドは、`-y` オプションを指定していない場合に確認のためのプロンプトを表示します。

`paginate (page) [{off | on}]`

出力をページ別表示します。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。標準入力または標準出力のどちらかが非端末デバイスでない限り、初期設定は `on` です。ページ別表示機能は、標準入力と標準出力のどちらも端末デバイスである場合のみ、`on` になります。シェル環境変数 `PAGER` を使用すると、ページ別表示出力に使用されるデフォルトのコマンドを無効化することができます。デフォルトのページ別表示コマンドは、ネイティブのオペレーティング・システム環境に固有のコマンドです。たとえば、UNIX システム・オペレーティング環境では、コマンド `pg` がデフォルトのコマンドです。

`passwd (passwd) [-r] local_domain_access_point_name
remote_domain_access_point_name`

指定し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の入力を管理者に求めます。`-r` オプションは、既存のパスワードと新しいパスワードをシステムが生成した新しいキーを使用して符号化することを指定します。パスワードは最大で 30 文字までに切り捨てられます。`passwd` は、TDomain ゲートウェイでのみサポートされていません。

```
printdomain (pd) -d local_domain_access_point_name
```

指定のローカル・ドメイン・アクセス・ポイントに関する情報を表示します。接続されたりリモート・ドメイン、ゲートウェイ・グループ・プロセスによって共有されているグローバル情報、およびドメイン・ゲートウェイ・タイプのインスタンスによって変化する追加情報などが表示されます。

```
printstats (pstats) -d local_domain_access_point_name
```

指定のローカル・ドメイン・アクセス・ポイントによって収集された統計情報および性能情報を表示します。表示される情報は、ドメイン・ゲートウェイのタイプによって変わります。

```
printtrans (pt) -d local_domain_access_point_name
```

指定のローカル・ドメイン・アクセス・ポイントに関するトランザクション情報を出力します。各トランザクション・レコードの出力には、次のようにコロンで区切った文字列フィールドが並びます。

```
process ID:local domain access point name:remote domain access
point name:service name:local GTRID:remote GTRID:record
type:timestamp
```

トランザクションがドメインに対してローカルに実行される場合は、リモート GTRID のフィールドは空になります。

```
quit (q)
```

セッションを終了します。

```
resume (res) -d local_domain_access_point_name [{ -all | service}]
```

指定のサービスの処理、または指定のローカル・ドメイン・アクセス・ポイントで処理するすべてのリモート・サービスの処理を再開します。

```
stats (stats) -d local_domain_access_point_name [{ off | on | reset }]
```

指定のローカル・ドメイン・アクセス・ポイントに関する統計収集を、アクティブ (on)、アクティブ解除 (off)、または再設定 (reset) します。オプションを何も指定しない場合は、現在の設定値が on と off で切り替わり、新しい設定が表示されます。初期設定は off です。

```
suspend (susp) -d local_domain_access_point_name [{ -all | service}]
```

指定のローカル・ドメイン・アクセス・ポイントの 1 つまたはすべてのリモート・サービスを中断します。

`topenpasswd (typasswd) [-r] local_domain_access_point_name`

管理者に対して、指定したローカル・ドメイン・アクセス・ポイントの新しいパスワードの入力を求めるプロンプトが表示されます。`-r` オプションは、既存のパスワードと新しいパスワードをシステムが生成した新しいキーを使用して符号化することを指定します。パスワードは 12 文字以内で入力します。

新しいパスワードは、セキュリティ機能が有効になっている BEA TOP END システムに RTQ メッセージを送信するときに使用されます。ユーザ ID として渡されるのは、ローカル・ドメイン・アクセス・ポイントの `ACCESSPOINTID (DOMAINID)` です。`topenpasswd` は、TOP END Domain Gateway でのみサポートされます。

`unadvertise (unadv) -d local_domain_access_point_name [{ -all | service}]`

指定のローカル・ドメイン・アクセス・ポイントの 1 つまたはすべてのリモート・サービスを宣言解除します。

`verbose (v) [{off|on}]`

冗長モードで出力を行います。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。初期設定は `off` です。

`! shellcommand`

シェルにエスケープして、`shellcommand` を実行します。

`!!`

直前のシェル・コマンドを再実行します。

`# [text]`

"#" で始まる行はコメント行で、無視されます。

`<CR>`

最後のコマンドを繰り返します。

コンフィギュレーション・モードのコマンド `dmadmin` コマンドは、`-c` オプション付きで実行するか、`config` サブコマンドを使用したときにコンフィギュレーション・モードになります。このモードで `dmadmin` を使用すると、`BDMCONFIG` ファイルの実行時の更新が可能になります。`dmadmin` は、追加あるいは取り出す入力フィールド値を含むバッファを管理し、各操作が完了した後で出力フィールド値およびステータスを表示します。ユーザは、利用可能なテキスト・エディタを使用して入力バッファを更新できます。

dmadmin はまず希望のセクションを求めるプロンプトを表示し、次に希望の操作を求めるプロンプトを表示します。

BDMCONFIG ファイルのセクションを求めるプロンプトは次のとおりです。

Section:

- | | |
|--------------------|-----------------------|
| 1) RESOURCES | 2) LOCAL_DOMAINS |
| 3) REMOTE_DOMAINS | 4) LOCAL_SERVICES |
| 5) REMOTE_SERVICES | 6) ROUTING |
| 7) ACCESS_CONTROL | 8) PASSWORDS |
| 9) TDOMAINS | 10) OSITPS |
| 11) SNADOMS | 12) LOCAL_REMOTE_USER |
| 13) REMOTE_USERS | 14) SNACRMS |
| 15) SNASTACKS | 16) SNALINKS |
| 18) TOPEND | 19) OSITPX |
| q) QUIT | |

Enter Section [1]:

デフォルトセクションの番号は、プロンプトの終わりに角かっこ内に表示されます。デフォルトの値を受け入れるときは、RETURN または ENTER を押します。別のセクションを選択するときは、その番号を入力して RETURN または ENTER を押します。

次に dmadmin は、希望の操作を求めるプロンプトを表示します。

Operations:

- | | |
|----------------|-----------|
| 1) FIRST | 2) NEXT |
| 3) RETRIEVE | 4) ADD |
| 5) UPDATE | 6) DELETE |
| 7) NEW_SECTION | 8) QUIT |

Enter Operation [1]:

デフォルトの操作番号がプロンプトの終わりの角かっこ内に表示されます。RETURN または ENTER を押してこのオプションを選択します。別の操作を選択するときは、その番号を入力して RETURN または ENTER を押します。

現在サポートされている操作は次のとおりです。

1. FIRST - 指定したセクションの先頭レコードを取り出します。キー・フィールドは必要ありません(入力バッファ内にある場合、キー・フィールドは無視されます)。
2. NEXT - 入力バッファ内のキー・フィールドに基づいて、指定したセクションから次のレコードを取り出します。

3. `RETRIEVE` - キー・フィールドで指定されたセクションから、指定されたレコードを取り出します。以下のフィールドの説明を参照してください。
4. `ADD` - 指定したセクションに、指定したレコードを追加します。フィールドが指定されていない場合（フィールドが不要である場合に限り）、`DMCONFIG(5)` で指定されたデフォルト値が使用されます。すべてのフィールドの現在の値は、出力バッファ内に返されます。この操作は、BEA Tuxedo 管理者だけが行えます。
5. `UPDATE` - 指定したセクションの入力バッファ内にある、指定したレコードを更新します。入力バッファ内で指定されていないフィールドは変更されません。すべてのフィールドの現在の値は、入力バッファ内に返されます。この操作は、BEA Tuxedo 管理者だけが行えます。
6. `DELETE` - 入力バッファで指定したレコードを、選択したセクションから削除します。この操作は、BEA Tuxedo 管理者しか実行できません。
7. `NEW SECTION` - 入力バッファをクリアします。すべてのフィールドが削除されます。この操作の後、`dmadmin` は即座にセクションの入力を求めます。
8. `QUIT` - プログラムを正常に終了します。`dmadmin` は終了します。プロンプトで値 `q` を入力してもプログラムを終了できます。

コンフィギュレーション操作を行う場合は、有効なユーザ識別子は、このプログラムが実行されるマシンの BEA Tuxedo 管理者ユーザ識別子 (UID) と一致している必要があります。レコードが更新または追加されると、`dmloadcf(1)` によって使用されるすべてのデフォルト値および検証が強制されます。

`dmadmin` は次に、入力バッファを編集するかどうかをたずねるプロンプトを表示します。

```
Enter editor to add/modify fields [n]?
```

ここで、`y` を入力すると、入力バッファは一時ファイルにコピーされ、テキスト・エディタが起動されます。どのエディタを使用するかは、環境変数 `EDITOR` で指定できます。デフォルトでは UNIX のテキスト・エディタの `ed` が使用されます。入力形式は、フィールド名とフィールド値のペアで、[49 ページの「構成入力フォーマット」](#) で記述しています。各 `DMCONFIG` セクションに関連するフィールド名は、以下のサブセクションの表にリストしま

す。フィールドおよび関連する範囲の指定方法、デフォルト値、制限などの指定方法については、[DMCONFIG\(5\)](#)、[DMCONFIG for GWTOPEND\(5\)](#)、および [DM_MIB\(5\)](#) を参照してください。ほとんどのケースでは、フィールド名は `DMCONFIG` ファイル内の `KEYWORD` と同じです (接頭辞として "TA_" が付いています)。ユーザが入力バッファの編集を終えると、`dmadmin` がそれを読み込みます。ある特定のフィールド名が複数行に現れる場合、最初のものだけが使用され、残りは無視されます。エラーが発生した場合は、構文エラーが表示され、問題を訂正するかどうかをたずねるプロンプトが表示されます。

```
Enter editor to correct?
```

問題を訂正しない場合 (応答 `n`) は、入力バッファにフィールドは含まれません。問題を訂正する場合は、エディタが再度実行されます。

最後に、操作を行うかどうかをたずねるプロンプトが表示されます。

```
Perform operation [y]?
```

操作が完了すると、`dmadmin` は戻り値を `Return value TAOK` 内に表示し、その後出力バッファ・フィールドが続きます。次に、プロセスが再開され、セクションを求めるプロンプトが表示されます。すべての出力バッファ・フィールドは、バッファがクリアされない限り、入力バッファ内で利用できます。

いつでもブレイクを入力して、セクションを求めるプロンプトでの対話を再開できます。

"QUIT" を選択すると、構成のテキストバージョンのバックアップを作成する許可を求めるプロンプトが表示されます。

```
Unload BDMCONFIG file into ASCII backup [y]?
```

バックアップを選択した場合、ファイル名の入力を求めるプロンプトが表示されます。

```
Backup filename [DMCONFIG]
```

成功時にはバックアップが作成されたことが示され、失敗時にはエラーが表示されます。

構成入力
フォーマット

入力パケットは、次のようにフォーマットされた行で構成されます。

```
fldname fldval
```

フィールド名は、1 つまたは複数のタブ (または空白) によってフィールド値と区別されます。

長いフィールド値は、1 つまたは複数のタブ (`dmadmin` に再度読み込まれたときに破棄されます) で始まる継続行を使用することで、次の行に継続できません。

改行文字だけで構成される空の行は無視されます。

表示できない文字をフィールド値に入力したり、タブからフィールド値を開始するには、バックスラッシュを使用し、その後に希望する文字を 2 文字の 16 進表現で使用します (UNIX システムのリファレンス・マニュアルの ASCII(5) を参照してください)。たとえば、スペースは `\20` と入力データに入力できます。バックスラッシュは 2 回書くことで 1 つのバックスラッシュを入力できます。`dmadmin` は、このフォーマットの入力をすべて認識しますが、最も役に立つのは表示できない文字の場合です。

構成の制限事項

次に動的 Domains を再構成する機能についての一般的な制限事項を示します。

- キー・フィールドの値 (次のセクションを参照) は変更できません。システムがダウンしているときに、コンフィギュレーション・ファイルを再ロードすれば、キー・フィールドを変更できます。
- 動的削除は、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・グループがアクティブ (動作中) のときは実行できません。

Domains 関連の新しい用語

BEA Tuxedo のリリース 7.1 以降では、Domains 用の MIB で、ローカル・ドメインとリモート・ドメインとの相互作用を記述するため、クラスと属性の用語が改善されています。新しい用語は、`DM_MIB` のリファレンス・ページ、クラス、エラー・メッセージ、および `DMCONFIG` のリファレンス・ページ、セクション名、パラメータ名、およびエラー・メッセージで使用されています。新しい用語は `dmadmin` のユーザ・インターフェイスには適用されていませんが、`dmadmin` は `DMCONFIG` の従来の用語と新しい用語の両方を認識します。

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

構成フィールド識別子/更新に関する制限事項

次のセクションは、各 DMCONFIG について、どのフィールド識別子が各 DMCONFIG フィールドに関連しているか、識別子のフィールド・タイプは何か、およびいつフィールドを更新できるかを説明しています。適用可能なフィールド値はすべて、取り出し操作によって返されます。レコードを追加できるフィールド、またはレコードを追加するために必要なフィールドについては、[DMCONFIG\(5\)](#)、[DMCONFIG for GWTOPEND\(5\)](#)、および [DM_MIB\(5\)](#) で説明します。以下にキーとして示すフィールドは、セクション内のレコードを一意に識別するために使用するキー・フィールドです。これらのキー・フィールドは、更新を行うとき入力バッファ内になければなりません。また、これらのキー・フィールドを動的に更新することはできません。「更新」列は、どのようなときにフィールドを更新できるかを示しています。可能な値は次のとおりです。

- Yes - いつでも更新できます。
- NoGW - ローカル・ドメイン・アクセス・ポイントに関連付けられたゲートウェイ・グループが実行中の間は動的に更新できません。
- No - ドメイン・ゲートウェイ・グループが1つでも実行中であれば動的に更新できません。

DM_LOCAL セクションの構成

次の表に、DM_LOCAL セクション (DM_LOCAL_DOMAINS セクション) のフィールド一覧を示します。dmadmin 操作のプロンプトでこのセクションにアクセスするには、2 (LOCAL_DOMAINS) を入力します。

表 4DM_LOCAL セクション

フィールド識別子	タイプ	更新	注意
TA_LDOME	文字列	NoGW	キー：ローカル・ドメイン・アクセス・ポイント名
TA_GWGRP	文字列	NoGW	
TA_TYPE	文字列	NoGW	形式：{TDOMAIN TOPEND SNAX OSITP OSITPX}
TA_DOMAINID	文字列	NoGW	新しい DMCONFIG 用語では、DOMAINID は ACCESSPOINTID
TA_AUDITLOG	文字列	NoGW	
TA_BLOCKTIME	数値	NoGW	
TA_CONNECTION_POLICY	文字列	NoGW	形式：{ON_DEMAND ON_STARTUP INCOMING_ONLY}
TA_MAXRETRY	数値	NoGW	TA_CONNECTION_POLICY に伴うパラメータ
TA_RETRY_INTERVAL	数値	NoGW	TA_CONNECTION_POLICY に伴うパラメータ
TA_DMTLOGDEV	文字列	NoGW	
TA_DMTLOGNAME	文字列	NoGW	
TA_DMTLOGSIZE	数値	NoGW	
TA_MAXRDTRAN	数値	NoGW	新しい DMCONFIG 用語では、MAXRDTRAN は MAXRAPTRAN
TA_MAXTRAN	数値	NoGW	

表 4DM_LOCAL セクション (続き)

フィールド識別子	タイプ	更新	注意
TA_MAXRDOM	数値	NoGW	OSITP にのみ適用可能 (新しい DMCONFIG 用語では、MAXRDOM は MAXACCESSPOINT)
TA_SECURITY	文字列	NoGW	TDOMAIN (TDomain) 形式 : {NONE APP_PW DM_PW} TOPEND (TEDG) 形式 : {NONE CLEAR SAFE PRIVATE} SNAX (SNA) 形式 : {NONE DM_USER_PW} OSITPX (OSI TP 4.x) 形式 : {NONE DM_PW}

DM_REMOT
E セクショ
ンの構成

次の表に、DM_REMOTE セクション (DM_REMOTE_DOMAINS セクション) のフィールドの一覧を示します。dmadmin 操作のプロンプトでこのセクションにアクセスするには、3 (REMOTE_DOMAINS) を入力します。

表 5DM_REMOTE セクション

フィールド識別子	タイプ	更新	注意
TA_RDOM	文字列	No	キー : リモート・ドメイン・アクセス・ポイント名
TA_TYPE	文字列	No	形式 : {TDOMAIN TOPEND SNAX OSITP OSITPX}
TA_DOMAINID	文字列	No	新しい DMCONFIG 用語では、DOMAINID は ACCESSPOINTID

セクション 1 - コマンド

DM_TDOMA IN セクションの構成 DM_TDOMAIN セクションには、TDOMAIN タイプ・ドメインが必要とするネットワーク・アドレッシング・パラメータが含まれています。次の表に、このセクションのフィールドの一覧を示します。

表 6DM_TDOMAIN セクション

フィールド識別子	タイプ	更新	注意
TA_LDOM または TA_RDOM	文字列	No/NoG W	キー：ローカルまたはリモート・ドメイン・アクセス・ポイント名
TA_NWADDR	文字列	No/NoG W	テキスト (ASCII) 形式 (NULL 文字を除く)
TA_NWDEVICE	文字列	No/NoG W	
TA_CONNECTION_POLICY * **	文字列	No/NoG W	形式：{LOCAL ON_DEMAND ON_STARTUP INCOMING_ONLY}
TA_MAXRETRY *	数値	No/NoG W	TA_CONNECTION_POLICY に伴うパラメータ
TA_RETRY_INTERVAL *	数値	No/NoG W	TA_CONNECTION_POLICY に伴うパラメータ
TA_TCPKEEPALIVE *	文字列	No/NoG W	形式：{LOCAL N Y} ドメイン TCP レベルのキープアライブ
TA_KEEPALIVE *	数値	No/NoG W	形式：-1 <= 2147483647 ミリ秒 (秒に丸められる) ドメインのアプリケーション・レベルのキープアライブ

* BEA Tuxedo リリース 8.1 以降で使用可能。
** DM_LOCAL セクションの同じパラメータより優先される。

表 6DM_TDOMAIN セクション (続き)

フィールド識別子	タイプ	更新	注意
TA_KEEPAWAIT *	数値	No/NoG W	形式 :0 <= 2147483647 ミ リ秒 (秒に丸められる) TA_KEEPAWAIT に伴うパ ラメータ

* BEA Tuxedo リリース 8.1 以降で使用可能。
** DM_LOCAL セクションの同じパラメータより優先される。

ローカル・ドメイン・アクセス・ポイント識別子 (TA_LDOMEIN) の場合、
TA_NWADDR フィールドおよび TA_NWDEVICE フィールドは、そのローカル・ド
メイン・アクセス・ポイントに関連付けられたゲートウェイ・グループが実
行中でないときに更新できます。

DM_TOPEN
D セクショ
ンの構成

DM_TOPEND セクションには、TOPEND タイプ・ドメインで必要となるネット
ワーク・アドレッシング・パラメータが含まれています。次の表に、このセ
クションのフィールドの一覧を示します。

表 7DM_TOPEND セクション

フィールド識別子	タイプ	更新	注意
TA_LDOMEIN または TA_RDOMEIN	文字列	No/NoG W	キー : ローカルまたはリ モート・ドメイン・アク セス・ポイント名
TA_NWADDR	文字列	No/NoG W	テキスト (ASCII) 形式 (NULL 文字を除く)
TA_NWDEVICE	文字列	No/NoG W	テキスト (ASCII) 形式 (NULL 文字を除く)
TA_TP_SYSTEM	文字列	No/NoG W	BEA TOP END システム 名

次の 2 つの条件が揃った場合、

- ドメイン識別子がローカル・ドメイン・アクセス・ポイント (TA_LDOM) である
- ローカル・ドメイン・アクセス・ポイントを表すゲートウェイ・グループが実行中でない (NoGW)

TA_NWADDR、TA_NWDEVICE、TA_TP_SYSTEM、および TA_TE_PWD フィールドを更新できます。ドメイン識別子がリモート・ドメイン・アクセス・ポイント (TA_RDOM) の場合、ゲートウェイ・グループの実行中 (No) に TA_NWADDR、TA_NWDEVICE、および TA_TP_SYSTEM フィールドは更新できません。TE_TE_PWD はローカル・ドメイン・アクセス・ポイント (TA_LDOM) のみに適用されます。

DM_OSITP セクションには、OSITP タイプ・ドメインが必要とする OSI TP 1.3 対応のネットワーク・アドレッシング・パラメータが含まれています。次の表に、このセクションのフィールドの一覧を示します。

表 8DM_OSITP セクション

フィールド識別子	タイプ	更新	注意
TA_LDOM または TA_RDOM	文字列	No/NoGW	キー：ローカルまたはリモート・ドメイン・アクセス・ポイント名
TA_APT	文字列	No/NoGW	
TA_AEQ	文字列	No/NoGW	
TA_AET	文字列	No/NoGW	
TA_ACN	文字列	No/NoGW	
TA_APID	文字列	No/NoGW	
TA_AEID	文字列	No/NoGW	
TA_PROFILE	文字列	No/NoGW	

ローカル・ドメイン・アクセス・ポイント識別子 (TA_LDOM) の場合、この表に記載されていないフィールドは、そのローカル・ドメイン・アクセス・ポイントに関連付けられたゲートウェイ・グループが実行中でないときに更新できません。

DM_OSITPX セクションには、OSITPX タイプ・ドメインが必要とする OSI TP 4.0 以降対応のネットワーク・アドレッシング・パラメータが含まれています。次の表に、このセクションのフィールドの一覧を示します。

注記 DM_OSITPX セクションは、BEA Tuxedo リリース 8.0 以降が動作していないと使用できません。

表 9DM_OSITPX セクション

フィールド識別子	タイプ	更新	注意
TA_LDOM または TA_RDOM	文字列	No/NoGW	キー：ローカルまたは リモート・ドメイン・ アクセス・ポイント 名
TA_AET	文字列	No/NoGW	
TA_NWADDR	文字列	No/NoGW	
TA_TSEL	文字列	No/NoGW	
TA_DNSRESOLUTION	文字列	No/NoGW	
TA_PSEL	文字列	No/NoGW	
TA_SSEL	文字列	No/NoGW	
TA_TAILORPATH	文字列	No/NoGW	
TA_MINENCRYPTBITS	数値	No/NoGW	
TA_MAXENCRYPTBITS	数値	No/NoGW	
TA_MULTIPLEXING	数値	No/NoGW	
TA_XATMIENCODING	文字列	No/NoGW	
TA_EXTENSIONS	文字列	No/NoGW	
TA_OPTIONS	文字列	No/NoGW	

ローカル・ドメイン・アクセス・ポイント識別子 (TA_LDOM) の場合、この表に記載されていないフィールドは、そのローカル・ドメイン・アクセス・ポイントに関連付けられたゲートウェイ・グループが実行中でないときに更新できます。

DM_EXPORT セクションの構成
 次の表に、DM_EXPORT (DM_LOCAL_SERVICES) セクションのフィールドの一覧を示します。dmadmin 操作のプロンプトでこのセクションにアクセスするには、4 (LOCAL_SERVICES) を入力します。

表 10DM_EXPORT セクション

フィールド識別子	タイプ	更新	注意
TA_SERVICENAME	文字列	No	キー: エクスポートするローカル・サービス名 (BEA Tuxedo CORBA 環境では、ローカル・ドメインのドメイン名はローカル UBBCONFIG ファイルで指定する。その際、service は <code>"/domain_name"</code> の形式で指定する)
TA_LDOM	文字列	Yes	キー: ローカル・ドメイン・アクセス・ポイント名 (新しい DMCONFIG 用語では、LDOM は LACCESSPOINT)
TA_ACLNAME	文字列	Yes	
TA_CONV	文字列	NoGW	形式: {Y N}
TA_RNAME	文字列	Yes	TDOMAIN、SNAX、OSITP、および OSITPX に適用可能
TA_TYPE	文字列	NoGW	形式: {SERVICE QSPACE QNAME} TOPEND にのみ適用可能
TA_TE_PRODUCT	文字列	NoGW	TOPEND にのみ適用可能
TA_TE_FUNCTION	文字列	NoGW	TOPEND にのみ適用可能
TA_TE_TARGET	文字列	NoGW	TOPEND にのみ適用可能
TA_TE_QUALIFIER	数値	NoGW	TOPEND にのみ適用可能

表 10DM_EXPORT セクション (続き)

フィールド識別子	タイプ	更新	注意
TA_TE_RTQGROUP	文字列	NoGW	TOPEND にのみ適用可能
TA_TE_RTQNAME	文字列	NoGW	TOPEND にのみ適用可能
TA_BUFTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能
TA_BUFSTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能 (BUFTYPE の「S」は「サブタイプ」を表す)
TA_OBUFTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能
TA_OBUFSSTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能 (OBUFSSTYPE の「S」は「サブタイプ」を表す)
TA_COUPLING	文字列	Yes	OSITPX にのみ適用可能
TA_INRECTYPE	文字列	Yes	OSITPX にのみ適用可能
TA_INRECSTYPE	文字列	Yes	OSITPX にのみ適用可能 (INRECSTYPE の「S」は「サブタイプ」を表す)
TA_OUTRECTYPE	文字列	Yes	OSITPX にのみ適用可能
TA_OUTRECSTYPE	文字列	Yes	OSITPX にのみ適用可能 (OUTRECSTYPE の「S」は「サブタイプ」を表す)

DM_IMPORT
T セクション
の構成

次の表に、DM_IMPORT セクション (DM_REMOTE_SERVICES セクション) のフィールドの一覧を示します。dadmin 操作のプロンプトでこのセクションにアクセスするには、5 (REMOTE_SERVICES) を入力します。

表 11DM_IMPORT セクション

フィールド識別子	タイプ	更新	注意
TA_SERVICENAME	文字列	No	キー：インポートするリモート・サービス名 (BEA Tuxedo CORBA 環境では、リモート・ドメインのドメイン名はリモート UBBCONFIG ファイルで指定する。その際、 <i>service</i> は " <i>//domain_name</i> " の形式で指定する)
TA_RDOM	文字列	No	キー：リモート・ドメイン・アクセス・ポイント名 (新しい DMCONFIG 用語では、RDOM は RACCESSPOINT)
TA_LDOM	文字列	No	キー：ローカル・ドメイン・アクセス・ポイント名 (新しい DMCONFIG 用語では、LDOM は LACCESSPOINT)
TA_CONV	文字列	NoGW	形式 : {Y N }
TA_LOAD	数値	Yes	
TA_RNAME	文字列	Yes	TDOMAIN、SNAX、OSITP、および OSITPX に適用可能
TA_TYPE	文字列	Yes	形式 : {SERVICE QSPACE QNAME} TOPEND にのみ適用可能
TA_ROUTINGNAME	文字列	Yes	
TA_TE_PRODUCT	文字列	Yes	TOPEND にのみ適用可能
TA_TE_FUNCTION	文字列	Yes	TOPEND にのみ適用可能

表 11DM_IMPORT セクション (続き)

フィールド識別子	タイプ	更新	注意
TA_TARGET	文字列	Yes	TOPEND にのみ適用可能
TA_TE_QUALIFIER	数値	Yes	TOPEND にのみ適用可能
TA_TE_RTQGROUP	文字列	Yes	TOPEND にのみ適用可能
TA_TE_RTQNAME	文字列	Yes	TOPEND にのみ適用可能
TA_BUFTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能
TA_BUFSTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能 (BUFTYPE の「S」は「サブタイプ」を表す)
TA_OBUFTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能
TA_OBUFSTYPE	文字列	Yes	TOPEND、SNAX、OSITP、および OSITPX に適用可能 (OBUFSTYPE の「S」は「サブタイプ」を表す)
TA_AUTOPREPARE	文字列	Yes	OSITPX にのみ適用可能
TA_INRECTYPE	文字列	Yes	OSITPX にのみ適用可能
TA_INRECSTYPE	文字列	Yes	OSITPX にのみ適用可能 (INRECSTYPE の「S」は「サブタイプ」を表す)
TA_OUTRECTYPE	文字列	Yes	OSITPX にのみ適用可能
TA_OUTRECSTYPE	文字列	Yes	OSITPX にのみ適用可能 (OUTRECSTYPE の「S」は「サブタイプ」を表す)
TA_TPSUTTYPE	文字列	Yes	OSITPX にのみ適用可能

表 11DM_IMPORT セクション (続き)

フィールド識別子	タイプ	更新	注意
TA_REMTPSUT	文字列	Yes	OSITPX にのみ適用可能

セクション 1 - コマンド

DM_ROUTING セクションの構成
次の表に、DM_ROUTING セクションのフィールドの一覧を示します。

表 12DM_ROUTING セクション

フィールド識別子	タイプ	更新	注意
TA_ROUTINGNAME	文字列	No	キー：ルーティング基準テーブルの名前
TA_FIELD	文字列	Yes	
TA_RANGE	文字列	Yes	
TA_BUFTYPE	文字列	Yes	

DM_ACCESS_CONTROL セクションの構成
次の表に、DM_ACCESS_CONTROL セクションのフィールドの一覧を示します。

表 13DM_ACCESS_CONTROL セクション

フィールド識別子	タイプ	更新	注意
TA_ACLNAME	文字列	No	キー：アクセス制御リストの名前
TA_RDOM	文字列	Yes	キー：リモート・ドメイン・アクセス・ポイント名

DM_PASSWORDS セクションの構成

次の表に、DM_PASSWORDS セクションのフィールドの一覧を示します。このセクションは、TDomain ゲートウェイにのみ適用されます。

表 14DM_PASSWORDS セクション

フィールド識別子	タイプ	更新	注意
TA_LDOM	文字列	No	キー：ローカル・ドメイン・アクセス・ポイント名

表 14DM_PASSWORDS セクション (続き)

フィールド識別子	タイプ	更新	注意
TA_RDOM	文字列	No	キー: リモート・ドメイン・アクセス・ポイント名
TA_LPWD	文字列	Yes	形式: {Y N U}
TA_RPWD	文字列	Yes	形式: {Y N U}

TA_LPWD および TA_RPWD は、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(あるいはその両方)について定義済みのパスワードがあるかどうかを示します。パスワードは表示されません。UPDATE 操作を選択した場合は、対応するフィールドの値を U に設定する必要があります。この場合、エコーがオフになり、対応するパスワードの入力を求めるプロンプトが表示されます。

コンフィ
ギュレー
ション・
モードでの
診断

dmadmin は、FML 型付きバッファを割り当てできない場合、ユーザの /etc/passwd エントリを判別できない場合、または環境変数 FIELDTBLS または FLDTBLDIR を再設定できない場合に失敗します。

各操作の完了後、dmadmin によって表示される戻り値は、要求された操作のステータスを示します。戻り値には 3 つのクラスがあります。

次の戻り値は、パーミッションに関する問題または BEA Tuxedo の通信エラーを示しています。戻り値は、操作が正常に完了しなかったことを示しています。

[TAEPERM]

呼び出し側プロセスで ADD、UPDATE、または DELETE のいずれかの操作が指定されましたが、BEA Tuxedo 管理者として実行されていません。更新操作は管理者(つまり、TUXCONFIG ファイルの RESOURCES セクションの UID 属性で指定されたユーザ)が実行する必要があります。

[TAESYSTEM]

BEA Tuxedo エラーが発生しました。エラーの正確な内容は userlog(3c) に書き込まれます。

[TAEOS]

オペレーティング・システムのエラーが発生しました。

[TAETIME]

ブロッキング・タイムアウトが発生しました。入力バッファは更新されないで、検索操作に使用される情報は何も返されません。更新操作のステータスは、更新されたレコードで取り出しを行うことでチェックできます。

次の戻り値は、操作を行うこと自体の問題を示します。一般的には、入力バッファ内のアプリケーション・データに関するセマンティクスの問題です。文字列フィールド `TA_STATUS` は出力バッファ内で設定され、問題を記述する短いテキストを含んでいます。文字列フィールド `TA_BADFLDNAME` は問題（エラーは1つのフィールドが原因で発生すると想定しています）の原因となる値を含むフィールドのフィールド名に設定されます。

[TAECONFIG]

BDMCONFIG ファイルの読み取り中にエラーが発生しました。

[TAEDUPLICATE]

重複するレコードを追加しようとした。

[TAEINCONSIS]

1 つまたは一連のフィールド値の指定が矛盾しています。

[TAENOTFOUND]

操作対象として指定されたレコードが見つかりませんでした。

[TAENOSPACE]

更新を行う操作が試行されましたが、BDMCONFIG ファイル内に十分なスペースがありませんでした。

[TAERANGE]

フィールド値が範囲外であるか、無効です。

[TAEREQUIRED]

フィールド値が必要ですが、設定されていません。

[TAESIZE]

文字列フィールドのフィールド値が長すぎます。

[TAEUPDATE]

実行できない更新を行おうとした。

次の戻り値は、操作が成功したことを示します。

[TAOK]

操作が成功しました。BDMCONFIG ファイルに対する更新は行われていません。

[TAUPDATED]

操作が成功しました。BDMCONFIG ファイルに対して更新が行われました。

dmunloadcf を使用して構成にエントリを表示するとき、オプションのフィールド値は、値が設定されていない(文字列の場合)か 0(整数の場合)の場合には表示されません。これらのフィールドは、dmadmin を使用するとき、出力バッファ内に常に表示されます。この方法で、管理者がエントリを取り出しやすくし、また以前に設定されていないフィールドを更新しやすくします。エントリには、フィールド名が含まれ、タブが後に続きますが、フィールド値はありません。

構成の例

次の例では、dmadmin を使用して新しいリモート・ドメイン・アクセス・ポイントを追加します。ed(1) がエディタとして使用されます。

```
$ EDITOR=ed dmadmin
> config
Sections:
    1) RESOURCES          2) LOCAL_DOMAINS
    3) REMOTE_DOMAINS    4) LOCAL_SERVICES
    5) REMOTE_SERVICES   6) ROUTING
    7) ACCESS_CONTROL    8) PASSWORDS
    9) TDOMAINS          10) OSITPS
   11) SNADOMS           12) LOCAL_REMOTE_USER
   13) REMOTE_USERS     14) SNACRMS
   15) SNASTACKS        16) SNALINKS
   18) TOPEND           19) OSITPX
    q) QUIT
Enter Section [1]:
Enter Section [1]:2
Operations:
    1) FIRST             2) NEXT
    3) RETRIEVE          4) ADD
    5) UPDATE            6) DELETE
    7) NEW_SECTION      8) QUIT
Enter Operation [1]: 4
Enter editor to add/modify fields [n]? y
a
TA_RDOM                B05
TA_DOMAINID            BA.BANK05
TA_TYPE                TDOMAIN
```

```
.
w
53
q
Perform operation [y]? <return>
Return value TAUPDATED
Buffer contents:
TA_OPERATION      4
TA_SECTION        2
TA_DOMAINID       BA.BANK05
TA_RDOM           B05
TA_TYPE           TDOMAIN
TA_STATUS          Update completed successfully
Operations:
    1) FIRST          2) NEXT
    3) RETRIEVE      4) ADD
    5) UPDATE        6) DELETE
    7) NEW_SECTION  8) QUIT
Enter Operation [4]:7
Section:
    1) RESOURCES      2) LOCAL_DOMAINS
    3) REMOTE_DOMAINS 4) LOCAL_SERVICES
    5) REMOTE_SERVICES 6) ROUTING
    7) ACCESS_CONTROL 8) PASSWORDS
    9) TDOMAINS      10) OSITPS
   11) SNADOMS       12) LOCAL_REMOTE_USER
   13) REMOTE_USERS  14) SNACRMS
   15) SNASTACKS     16) SNALINKS
   18) TOPEND        19) OSITPX
    q) QUIT
Enter Section [1]:9
Operations:
    1) FIRST          2) NEXT
    3) RETRIEVE      4) ADD
    5) UPDATE        6) DELETE
    7) NEW_SECTION  8) QUIT
Enter Operation [6]: 4
Enter editor to add/modify fields [n]? y
a
TA_RDOM           B05
TA_NWADDR         0x00020401c0066d05
TA_NWDEVICE       /dev/tcp
.
w
55
q
Perform operation [y]? <return>
Return value TAUPDATED
Buffer contents:
```

```

TA_OPERATION      4
TA_SECTION        8
TA_RDOM           B05
TA_NWADDR         0x00020401c0066d05
TA_NWDEVICE       /dev/tcp
TA_STATUS         Update completed successfully
Operations:
    1) FIRST          2) NEXT
    3) RETRIEVE       4) ADD
    5) UPDATE         6) DELETE
    7) NEW_SECTION   8) QUIT
Enter Operation [4]:8
> quit

```

dmadmin プログラムが終了します。

セキュリティ

dmadmin がアプリケーション管理者の UID を使用して実行された場合、許可を得たユーザであると想定され、セキュリティはバイパスされます。

dmadmin が別のユーザ ID で実行された場合、およびセキュリティ・オプションが TUXCONFIG ファイル内で使用可能になっている場合は、dmadmin プログラムを開始するための対応するアプリケーション・パスワードが必要です。標準入力端末である場合、dmadmin はエコーをオフの状態ユーザにパスワードの入力を求めます。標準入力端末でない場合は、パスワードは環境変数 APP_PW から取り出されます。この環境変数が指定されていないときに、アプリケーション・パスワードが必要な場合は、dmadmin を開始できません。

別のユーザ ID (管理者の UID 以外) で実行中の場合、制限されたセットのコマンドしか使用できません。

環境変数

dmadmin は環境変数 FIELDTBLS および FLDTBLDIR を再設定して、`${TUXDIR}/udataobj/dmadmin` フィールド・テーブルをピックアップします。このため、TUXDIR 環境変数は適切に設定する必要があります。

アプリケーションでセキュリティが必要なときに、dmadmin への標準入力が端末からではない場合、APP_PW 環境変数は対応するアプリケーション・パスワードに設定する必要があります。

TUXCONFIG 環境変数は、BEA Tuxedo コンフィギュレーション・ファイルのパス名に設定する必要があります。

一般的な診断

dmadmin コマンドをシステムがブートする前に入力すると、次のメッセージが表示されます。

No bulletin board exists. Only logging commands are available.

すると、`dmadmin` は対応するコマンドの入力を求めます。

正しくないアプリケーション・パスワードが入力されるか、環境を通してシェル・スクリプトで利用できない場合、ログ・メッセージが生成され、次のメッセージが表示されてコマンドが終了します。Invalid password entered.

- 相互運用性** `dmadmin` は、BEA Tuxedo リリース 5.0 以降にインストールする必要があります。リリース 5.0 ゲートウェイが存在する同じドメイン内の他のノードは、BEA Tuxedo リリース 4.1 またはそれ以降でもかまいません。
- 移植性** `dmadmin` 管理ツールは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで使用できます。
- 関連項目** [dmloadcf\(1\)](#)、[tmadmin\(1\)](#)、[topendpasswd\(1\)](#)、[DMADM\(5\)](#)、[DMCONFIG\(5\)](#)、[DMCONFIG for GWTOPEND\(5\)](#)
- 『BEA Tuxedo Domains コンポーネント』
- 『ATMI アプリケーションでの BEA Tuxedo TOP END Domain Gateway の使用』

dmloadcf(1)

名前 dmloadcf - DMCONFIG ファイルを解析し、バイナリの BDMCONFIG コンフィギュレーション・ファイルをロード

形式 dmloadcf [-c] [-n] [-y] [-b *blocks*] {*DMCONFIG_file* | - }

機能説明 dmloadcf は、ファイルまたは DMCONFIG 構文による標準入力を読み取り、構文をチェックして、オプションでバイナリの BDMCONFIG コンフィギュレーション・ファイルをロードします。BDMCONFIG 環境変数は、情報の格納先となる BDMCONFIG ファイルのパス名を指しています。

dmloadcf は、DMCONFIG ファイルの必要なセクションがないことを検出すると、エラー・メッセージをプリントします。入力ファイルを構文解析しているときに構文エラーを検出すると、dmloadcf は BDMCONFIG ファイルを更新せずに終了します。

dmloadcf を使用する場合は、\$TUXDIR/udataobj/DMTYPE ファイルが存在していなければなりません。このファイルは有効なドメイン・タイプを定義します。このファイルが存在しないと、dmloadcf は BDMCONFIG ファイルを更新せずに終了します。

dmloadcf を使用するユーザーの実効ユーザー識別子は、TUXCONFIG ファイルの RESOURCES セクションの UID と一致していなければなりません。

dmloadcf に -c オプションを付けると、プログラムはこの環境設定の各ローカル・ドメイン (ゲートウェイ・グループ) に必要な最小の IPC リソースをプリントします。BDMCONFIG ファイルは更新されません。

dmloadcf に -n オプションを付けると、プログラムはテキスト形式の DMCONFIG ファイルの構文チェックだけを行い、BDMCONFIG ファイルの更新は行いません。

構文チェックの後、dmloadcf は、環境変数 BDMCONFIG が参照するファイルが存在するかどうか、存在する場合はそのファイルが有効な BEA Tuxedo ファイルであるかどうか、および BDMCONFIG テーブルを含んでいるかどうかを調べます。これらの条件が真でなかった場合、そのユーザにはファイルを作成して初期化するよう、次のような指示が出されます。

```
Initialize BDMCONFIG file:path[y, q]?
```

path は、BDMCONFIG ファイルの完全なファイル名です。標準入力および標準出力が端末に送られない場合、またはコマンド行で *-y* オプションが指定されている場合は、このプロンプトは表示されません。このプロンプトに対して *&"&* または *&dl"rq;* 以外で答えると、*dmloadcf* はバイナリのコンフィギュレーション・ファイルを作成せずに終了します。

BDMCONFIG ファイルが正しく初期化されていない場合、Initialize BDMCONFIG file プロンプトの後に *y* を入力すると、*dmloadcf* は BEA Tuxedo ファイルシステムを作成し、BDMCONFIG テーブルを作成します。*-b* オプションをコマンド行で指定すると、BEA Tuxedo システムのファイルシステムを作成するときに、*-b* オプションの引数がデバイスのブロック数として使用されます。*-b* オプションの値が、新しい BDMCONFIG テーブルを収容できる大きさである場合、*dmloadcf* は指定された値を使用して新しいファイルシステムを作成します。それだけの大きさがない場合には、エラー・メッセージを出力して終了します。*-b* オプションを指定しないと、*dmloadcf* は、BDMCONFIG テーブルを収容できる大きさのファイルシステムを新たに作成します。ファイルシステムが既に存在している場合は、*-b* オプションが無視されます。BDMCONFIG が raw デバイス (初期化されていない) である場合には、*-b* オプションを使用することを強くお勧めします。この場合、*-b* オプションを使用して、raw デバイスのブロック数を設定しなければなりません。BDMCONFIG が通常の UNIX ファイル・システムである場合は、*-b* オプションを使用しないようお勧めします。

BDMCONFIG ファイルがすでに初期化されていることが分かったら、*dmloadcf* は、BDMCONFIG ファイルで記述されているローカル・ドメインが動作していないことを確認します。ローカル・ドメインが動作している場合は、*dmloadcf* がエラー・メッセージをプリントして終了します。ローカル・ドメインが動作していない場合、*dmloadcf* は次のプロンプトを表示して、BDMCONFIG ファイルを上書きするかどうかを確認します。

```
"Really overwrite BDMCONFIG file [y, q]?"
```

標準入力または標準出力が端末でない場合、または *-y* オプションをコマンド行で指定した場合、このプロンプトは表示されません。このプロンプトに対して *&dl"rq;* または *&dl"rq;* 以外で答えると、*dmloadcf* はファイルを上書きせずに終了します。

TUXCONFIG ファイルの RESOURCES セクションで SECURITY パラメータを指定すると、dmloadcf は標準入力をフラッシュし、端末のエコーをオフにして、次のプロンプトでアプリケーション・パスワードの入力を促します。Enter Application Password? パスワードは、8 文字以内の長さで入力できます。この SECURITY パラメータがオンになっているときは、(ファイル以外の)標準入力を介してテキスト形式の DMCONFIG ファイルをロードすることはできません。標準入力がない場合、すなわちパスワードの入力を促すプロンプトが(たとえば here ファイルを使って)表示されない場合、dmloadcf は環境変数 APP_PW にアクセスしてアプリケーション・パスワードを設定します。選択したパスワードが長すぎたり、環境変数 APP_PW を端末以外の標準入力に設定していない場合、dmloadcf はエラー・メッセージをプリントし、ログ・メッセージを生成し、BDMCONFIG ファイルをロードできません。

エラーが発生せず、チェックの結果がすべて条件を満たしていると、dmloadcf は DMCONFIG ファイルを BDMCONFIG ファイルにロードします。BDMCONFIG テーブル内に存在するすべての情報は上書きされます。

Domains 関連の新しい用語

BEA Tuxedo のリリース 7.1 以降では、Domains 用の MIB で、ローカル・ドメインとリモート・ドメインとの相互作用を記述するため、クラスと属性の用語が改善されています。新しい用語は、DM_MIB のリファレンス・ページ、クラス、エラー・メッセージ、および DMCONFIG のリファレンス・ページ、セクション名、パラメータ名、およびエラー・メッセージで使用されています。詳細については、DM_MIB(5) リファレンス・ページの「Domains 関連の新しい用語」を参照してください。

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 DMCONFIG 用語と、新しい 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 7.1 以降の dmloadcf では、DMCONFIG の両方のバージョンの用語を使用できます。デフォルトでは、dmunloadcf は、新しいドメイン関連の用語を使用する DMCONFIG ファイルを生成します。以前のドメイン関連の用語を使用する DMCONFIG ファイルを生成するには、dmunloadcf の -c オプションを使用します。

移植性

dmloadcf 管理ツールは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで使用できます。

環境変数

BDMCONFIG 環境変数は、BDMCONFIG ファイルを指していなければなりません。

使用例 次の例は、バイナリのコンフィギュレーション・ファイルを、テキスト形式ファイルの `bank.dmconfig` からロードする方法を示しています。ブロック数が 2000 の `BDMCONFIG` デバイスが作成（または初期化）されます。

```
dmloadcf -b 2000 bank.dmconfig
```

診断 入力にエラーが検出されると、エラーの原因となった行が、問題を指摘するメッセージとともに、標準エラーにプリントされます。`DMCONFIG` ファイルに構文エラーが見つかったり、システムが稼働中の場合は、`BDMCONFIG` ファイルの情報は更新されず、`dmloadcf` は終了コード 1 で終了します。

`dmloadcf` をアクティブなノード上で実行すると、次のエラー・メッセージが表示されます。

```
*** dmloadcf cannot run on an active node ***
```

`dmloadcf` を実行するユーザーの実効ユーザー識別子が、`TUXCONFIG` ファイルで指定された `UID` と一致しない場合は、次のエラー・メッセージが表示されます。

```
*** UID is not effective user ID ***
```

`dmloadcf` が正しく完了すると、終了コード 0 で終了します。`BDMCONFIG` ファイルが更新されると、このイベントを記録するための `userlog` メッセージが生成されます。

関連項目 [dmunloadcf\(1\)](#)、[DMCONFIG\(5\)](#)、[DMCONFIG for GWTOPEND\(5\)](#)、[UBBCONFIG\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

『BEA Tuxedo Domains コンポーネント』

『ATMI アプリケーションでの BEA Tuxedo TOP END Domain Gateway の使用』

dmunloadcf(1)

名前	dmunloadcf - バイナリの BDMCONFIG Domains コンフィギュレーション・ファイルをアンロード
形式	dmunloadcf [-c]
機能説明	<p>dmunloadcf は、BDMCONFIG コンフィギュレーション・ファイルをバイナリ表現からテキスト表現に変換します。この変換は、バイト配列が異なるマシン間で簡単にファイルをやり取りできるようにしたり、ファイルのコピーをコンパクトにかつ信頼性をもってバックアップするときに役立ちます。テキスト形式は、DMCONFIG(5) で説明されている形式と同じです。</p> <p>dmunloadcf は、BDMCONFIG 環境変数が参照する BDMCONFIG ファイルから値を読み取り、その値を標準出力に書き込みます。</p> <p>BEA Tuxedo リリース 7.1 以降では、dmunloadcf 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 ファイルを生成します。詳細については、次の「Domains 関連の新しい用語」を参照してください。以前のドメイン関連の用語を使用する DMCONFIG ファイルを生成するには、<code>-c</code> オプションを使用します。</p>
Domains 関連の新しい用語	<p>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、DM_MIB のリファレンス・ページ、クラス、エラー・メッセージ、および DMCONFIG のリファレンス・ページ、セクション名、パラメータ名、およびエラー・メッセージで使用されています。詳細については、DM_MIB(5) リファレンス・ページの「Domains 関連の新しい用語」を参照してください。</p> <p>旧バージョンとの互換性のため、BEA Tuxedo 7.1 より前に使用された DMCONFIG 用語と、新しい 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 7.1 以降の <code>dmloadcf</code> では、両方のバージョンの DMCONFIG 用語を使用できます。デフォルトでは、dmunloadcf は、新しいドメイン関連の用語を使用する DMCONFIG ファイルを生成します。以前のドメイン関連の用語を使用する DMCONFIG ファイルを生成するには、dmunloadcf の <code>-c</code> オプションを使用します。</p>
移植性	dmunloadcf コマンドは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。

使用例 次の例は、`/usr/tuxedo/BDMCONFIG` の環境設定を、`bdmconfig.backup` ファイルにアンロードする方法を示しています。

```
BDMCONFIG=/usr/tuxedo/BDMCONFIG dmunloadcf > bdmconfig.backup
```

診断 `dmunloadcf` は、`BDMCONFIG` 環境変数が指すファイルが存在しているかどうか、そのファイルが BEA Tuxedo システムの有効なファイルシステムであるかどうか、そしてそのファイルに `BDMCONFIG` テーブルが入っているかどうかをチェックします。これらの条件が 1 つでも満たされないと、`dmunloadcf` はエラー・メッセージをプリントして、エラー・コード 1 で終了します。`dmunloadcf` が正しく完了すると、終了コード 0 で終了します。

関連項目 [dmloadcf\(1\)](#)、[DMCONFIG\(5\)](#)、[DMCONFIG for GWTOPEND\(5\)](#)

『BEA Tuxedo Domains コンポーネント』

『ATMI アプリケーションでの BEA Tuxedo TOP END Domain Gateway の使用』

gencat(1)

名前 gencat - フォーマットされたメッセージ・カタログを生成

形式 gencat [-m] *catfile* *msgfile* . . .

機能説明 gencat ユーティリティは、メッセージ・テキストのソース・ファイル *msgfile* をフォーマットされたメッセージ・データベース *catfile* にマージします。このデータベースがなければ、データベース *catfile* が作成されず。*catfile* が存在する場合には、そのメッセージが新しい *catfile* に組み込まれます。セット番号とメッセージ番号が矛盾する場合には、*msgfile* に定義されている新しいメッセージ・テキストが *catfile* に現在格納されている古いメッセージ・テキストと置き換わります。gencat に入力されるメッセージ・テキストのソース・ファイルには、セット番号とメッセージ番号、あるいはメッセージ番号のみ (この場合、セット NL_SETD ([nl_types\(5\)](#) 参照) が想定されます) のいずれかを収めることができます。

メッセージ・テキストのソース・ファイルの形式は、次のように定義されています。ただし、メッセージ・テキストのソース行のフィールドは1つのASCIIの空白またはタブ文字で区切ります。その他のASCIIの空白やタブは、次のフィールドの一部とみなされます。

\$set *n* *comment*

ここで、*n* は次の \$set、\$delset またはファイル終わりまで下記メッセージのセット識別子を指定します。*n* は範囲 (1-`{NL_SETMAX}`) 内の数字でなければなりません。1つのソース・ファイル内のセット識別子は連続している必要はありません。セット識別子に続く文字列はすべて、注釈として扱われます。\$set ディレクティブの指定がメッセージ・テキスト・ソース・ファイルにない場合、すべてのメッセージはデフォルトのメッセージ・セット NL_SETD に格納されます。

\$delset *n* *comment*

既存のメッセージ・カタログからメッセージ・セット *n* を削除します。セット番号に続く文字列はすべて、注釈として扱われます (注: *n* が有効なセットでない場合は無視されます)。

\$ *comment*

ドル記号 (\$) に続いて ASCII の空白またはタブ文字がある行は、注釈として扱われます。

`m message_text`

`m` はメッセージ識別子で、これは範囲 (1-`{NL_MSGMAX}`) 内の番号です。このメッセージ・テキストの構文と、「注記」で説明する `-m` コマンド行オプションの構文とを混同しないようにしてください。メッセージ・テキストは、セット識別子が最後の `$set` ディレクティブによって指定され、かつメッセージ識別子が `m` である状態でメッセージ・カタログに格納されます。メッセージ・テキストが空で、ASCII の空白またはタブ・フィールド区切り文字があると、空の文字列がメッセージ・カタログに格納されます。メッセージ・ソース行にメッセージ番号があっても、フィールド区切り文字もメッセージ・テキストもない場合は、その番号を持つメッセージ (ある場合) がカタログから削除されます。メッセージ識別子は必ずしも連続している必要はありません。メッセージ・テキストの長さは、範囲 (0-`{NL_TEXTMAX}`) 内とします。

`$quote c`

この行は任意の引用文字 `c` を指定します。この文字は、後置きスペースやヌル (空) メッセージをメッセージ・ソース行で見分けやすいようにメッセージ・テキストを囲むときに使用できます。デフォルトの設定の場合、あるいは空の `$quote` ディレクティブが指定されている場合には、メッセージ・テキストに引用符を付けても認識されません。メッセージ・テキスト・ソース・ファイル内では、空の行は無視されます。テキスト文字列には、次の表に定義されている特殊な文字とエスケープ文字を含めることができます。

説明	記号	エスケープ・シーケンス
改行	NL(LF)	\n
水平タブ	HT	\t
垂直タブ	VT	\v
バックスペース	BS	\b
復帰	CR	\r
用紙送り	FF	\f
バックslash	\	\\

説明	記号	エスケープ・シーケンス
ビット・パターン	ddd	\ddd

エスケープ・シーケンス `\ddd` は、バックスラッシュとそれに続く 1、2 または 3 桁の 8 進数からなり、目的の文字の値を指定します。バックスラッシュに続く文字が指定されたものの 1 つでない場合には、バックスラッシュは無視されます。

バックスラッシュの次に ASCII の改行文字を使用して、文字列を次の行に継続させることもできます。たとえば、次の 2 行は 1 つのメッセージ文字列を表しています。

```
1 This line continues \
to the next line
```

上記のメッセージは下のメッセージと同じです。

```
1 This line continues to the next line
```

移植性 `gencat` は、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。

注意事項 このバージョンの `gencat` は、実行時に、`malloc` によって割り当てられた領域に読み取られるカタログを生成します。`gencat` のいくつかのバージョンで利用できる共用カタログはこのバージョンでは利用できません。一部のシステムでは、`malloc` で割り当てられたカタログの生成には `-m` オプションの指定が必要とされます。このオプションはコマンド行で指定することもできますが、その場合は何も機能しません。`malloc` で割り当てられたカタログがデフォルトの設定です。`-m` オプションは互換性を保持する目的でのみサポートされています。

このコマンドで生成されるカタログ・ファイルのサイズは 64K までに限定されています。それ以上のサイズになるような場合には、このコマンドからエラーが報告され、カタログ・ファイルは生成されません。

関連項目 [nl_types\(5\)](#)

genicf(1)

名前 genicf - インプリメンテーション・コンフィギュレーション・ファイル (ICF) を生成

形式 genicf [options] *idl-filename*...

機能説明 *idl-filename(s)* を指定すると、ICF ファイルが生成されます。この ICF ファイルは、コード生成プロセスに、インプリメンテーションの方針に関する追加情報、およびインプリメンテーションとインプリメントされるインターフェイスとの関係を示す情報を提供するファイルです。ICF ファイルを *idl* コマンドへの入力として指定すると、*idl* コマンドは ICF ファイルに指定されたインプリメンテーションとインターフェイスのペアに対してのみサーバ・コードを生成します。

生成される ICF ファイルのファイル名は、コマンド行で指定する最初の *idl-filename* と同じですが、拡張子は *.icf* になります。

間違った OMG IDL 構文で *idl-filename(s)* を指定すると、それに応じたエラーが返されます。

オプション `-D identifier=[definition]`

C++ プリプロセッサ・ディレクティブの `#define` と同じ働きをします。つまり、`-D` オプションは、定義ファイル内にある特定の識別子の代わりに使用するトークン文字列またはマクロを定義します。定義が指定されていない場合、`identifier` は 1 に定義されます。複数の `-D` オプションを指定することができます。`-D` オプションと `identifier` との間の空白はなくてもかまいません。

`-I pathname`

OMG IDL プリプロセッサ・ディレクティブ `#include` で指定するディレクトリのほか、インクルード・ファイルを検索するディレクトリを指定します。複数のディレクトリを指定する場合は、複数の `-I` オプションを指定します。

OMG IDL プリプロセッサ・ディレクティブ `#include` には、システム (`<a.idl>` など) およびユーザ ("`a.idl`" など) の 2 種類があります。UNIX システムでは、システム・ディレクティブ `#include` のパスは、`/usr/include` および `-I` オプションで指定される任意のディレクトリです。ユーザ・ディレクティブ `#include` のパスは、`#include` ディレクティブを含むファイルの場所と、その後続く、システム・ディレ

クティブ `#include` に指定されたパスで構成されます。Windows 2000 システムでは、システム・ディレクティブ `#include` とユーザ・ディレクティブ `#include` に違いはありません。

`-h` および `-?`

`genicf` コマンドの使用方法に関するヘルプを提供します。それ以外に行われるアクションはありません。

使用例 次のコマンドは、`emp.icf` ファイルを作成します。`genicf emp.idl`.

関連項目 [idl\(1\)](#)

idl(1)

名前	idl - Object Management Group (OMG) のインターフェイス定義言語ファイル (IDL) をコンパイルし、インターフェイスに必要なファイルを生成
形式	idl [-i] [-D <i>identifier</i> [= <i>value</i>]] [-I <i>pathname</i>][-h] [-P] [-T] <i>idl-filename</i> ... [<i>icf-filename</i> ...]
機能説明	<p>提供されている <i>idl-filename</i>() ファイルおよびオプションの <i>icf-filename</i>() ファイルを指定すると、idl コマンドは以下のファイルを生成します。</p> <p><i>idl-filename_c.cpp</i> クライアント・スタブ。ユーザ定義のデータ型の関数が埋め込まれます。</p> <p><i>idl-filename_c.h</i> インターフェイスのクラス定義。</p> <p><i>idl-filename_s.cpp</i> POA_skeleton クラスのインプリメンテーションを含むサーバ・スケルトン。</p> <p><i>idl-filename_s.h</i> POA_skeleton クラス定義。</p> <p><i>idl-filename_i.cpp</i> インプリメンテーションの例。このファイルは、<i>-i</i> オプションが指定されている場合にのみ生成されます。</p> <p><i>idl-filename_i.h</i> POA_skeleton クラスから継承される、インプリメンテーションの例のクラス定義。このファイルは、<i>-i</i> オプションが指定されている場合にのみ生成されます。</p> <p>注記 ICF ファイルが指定されている場合は、その ICF ファイル内の情報を基に、デフォルト値を変更するインターフェイス/インプリメンテーションに関する情報がコード・ジェネレータに渡されます。通常、インプリメンテーションの活性化方針およびトランザクション方針は ICF ファイルに指定されます。ICF ファイルが指定されていない場合は、OMG IDL ファイルで指定されるすべてのインターフェイスに対してデフォルトの方針が有効になり、すべてのインターフェイス</p>

に対してスケルトン・コードが生成されます。*icf-filename* ファイルを *idl* コマンドへの入力として指定すると、*icf-filename* に指定されたインプリメンテーション / インターフェイスのペアのみが、サーバの一部として生成されます。

IDL コンパイラは、生成するクライアント・スタブの情報を *filename_c.cpp* ファイルおよび *filename_c.h* ファイルに格納します。生成されたサーバ・スケルトンの情報は、*filename_s.cpp* ファイルおよび *filename_s.h* ファイルに格納されます。

IDL コンパイラは、既に生成されているクライアント・スタブ・ファイル (*filename_c.cpp* および *filename_c.h*)、および既に生成されているサーバ・スケルトン・ファイル (*filename_s.cpp* および *filename_s.h*) を上書きします。以前のバージョンのファイルはすべて破棄されます。

-i オプションを使用すると、IDL コンパイラはサンプルのインプリメンテーション・クラス定義ファイル (*filename_i.h*) を上書きします。以前のバージョンのファイルは破棄されます。サンプルのインプリメンテーション・ファイル (*filename_i.cpp*) も上書きされますが、コード保護ブロック内のコードは上書きされず、新しく生成されるファイルに復元されます。データが失われるのを避けるために、サンプルのインプリメンテーション・ファイル (*filename_i.h* および *filename_i.cpp*) を再生成する場合は、あらかじめそれらのファイルを安全な場所にコピーしておくことをお勧めします。

不明なオプションがこのコマンドに渡されると、そのオプションおよび使用方法を通知するメッセージがユーザに表示され、コンパイルは実行されません。

パラメータ *idl filename*
OMG IDL 文を含む 1 つまたは複数のファイルの名前。

オプション *-D identifier[=definition]*
C++ プリプロセッサ・ディレクティブの *#define* と同じ働きをします。つまり、*-D* オプションは、定義ファイル内にある特定の識別子の代わりに使用するトークン文字列またはマクロを定義します。*definition* が指定されていない場合、*identifier* は 1 に定義されます。複数の *-D* オプションを指定することができます。*-D* オプションと名前との間の空白はなくてもかまいません。

-I pathname
OMG IDL プリプロセッサ・ディレクティブ *#include* で指定するディレクトリのほか、インクルード・ファイルを検索するディレクトリを

指定します。複数のディレクトリを指定する場合は、複数の `-I` オプションを指定します。

OMG IDL プリプロセッサ・ディレクティブ `#include` には、システム (`<a.idl>` など) およびユーザ ("`a.idl`" など) の 2 種類があります。システム・ディレクティブ `#include` のパスは、システム・インクルード・ディレクトリおよび `-I` オプションで指定されるディレクトリで構成されます。ユーザ・ディレクティブ `#include` のパスは、`#include` ディレクティブを含むファイルの場所と、その後続く、システム・ディレクティブ `#include` に指定されたパスで構成されません。

デフォルトでは、`#include` ディレクティブで取り込むファイル内のテキストは、クライアント・コードおよびサーバ・コードには取り込まれません。

`-i`

`idl-filename_i.cpp` ファイルを生成します。これらのファイルには、OMG IDL ファイルで指定されたインターフェイスをインプリメントするインプリメンテーション用のサンプル・テンプレートが入っています。

注記 `idl` コマンドに `-i` オプションを使用してインプリメンテーション・ファイルを更新する手順は以下のとおりです。

1. インプリメンテーション・ファイルをバックアップします。
2. BEA ObjectBroker から BEA Tuxedo に移行する場合は、生成されているインプリメンテーション・ファイルでコード保護ブロックの区切り文字の `"OBB_PRESERVE_BEGIN"` および `"OBB_PRESERVE_END"` を `"M3_PRESERVE_BEGIN"` および `"M3_PRESERVE_END"` に変更します。
3. インクルード・ファイルをメソッド・インプリメンテーション・ファイル (`*_i.cpp`) に追加した場合は、そのファイルを編集して、取り込んだ要素を `INCLUDES` 保護ブロック内に移動します。
4. `idl` コマンドで `-i` オプションを使用して、編集したインプリメンテーション・ファイルを再生成します。

5. 再生成する前のインプリメンテーション定義ファイル(*_i.h)に修正を加えていた場合は、新しく生成された定義ファイルにその修正箇所を反映させます。修正箇所はコード保護ブロック内に配置します。これにより、以降更新を行ってもその修正箇所は自動的に保護されるようになります。BEA Tuxedo リリース 7.1 では、インプリメンテーションのコンストラクタ関数およびデストラクタ関数の文法が変更されているため注意が必要です。
6. メソッド・インプリメンテーション・ファイル(*_i.cpp)の保護ブロックの外側に、またはインプリメンテーションのコンストラクタ関数またはデストラクタ関数に修正を加えていた場合は、新しく生成されたファイルを編集して、その修正箇所を反映させます。修正箇所はコード保護ブロック内に配置します。これにより、以降更新を行ってもその修正箇所は自動的に保護されるようになります。

-P

TP フレームワークの代わりに POA を使用するサーバ・コードを生成します。このオプションを指定すると、スケルトン・クラスは TP フレームワークの `Tobj_ServantBase` クラスからは継承されず、`PortableServer::ServantBase` クラスから直接継承されます。デフォルトでは、スケルトン・クラスには TP フレームワークが使用されません。したがって、共同クライアント / サーバを開発する場合、これらのサーバは TP フレームワークを使用しないので、このスイッチを使用する必要があります。

サーバントの継承ツリー内に `Tobj_ServantBase` クラスがない場合、そのサーバントには `activate_object` メソッドおよび `deactivate_object` メソッドはありません。CORBA サーバの場合、これらのメソッドは、サーバントでのメソッド呼び出しの前に TP フレームワークによって呼び出されて、動的にサーバントを初期化して保存するために使用されます。CORBA 共同クライアント / サーバでは、ユーザが記述するコードで明示的にサーバントを作成して、サーバントの状態を初期化する必要があるため、`Tobj_ServantBase` 操作は不要です。-P オプションを指定した場合は、TP フレームワークが使用できないので、ICF ファイルが使用されることはありません。

-T

tie ベースのサーバント・コードを生成します。このコードでは、デレゲーションを使用して C++ インプリメンテーション・クラスのインスタンスをサーバントに結合できます。このオプションを使用すると、

継承によるスケルトンに関係しないクラスで CORBA オブジェクト操作をインプリメントできます。デフォルトでは、このオプションはオフに設定されます。

-h または -?

idl コマンドの使用方法に関するヘルプを提供します。それ以外に行われるアクションはありません。

使用例 `idl emp.idl`
 `idl emp.idl emp.icf`

関連項目 [genicf\(1\)](#)

idl2ir(1)

名前 idl2ir - インターフェイス・リポジトリを作成し、インターフェイス・リポジトリにインターフェイス定義をロード

形式 idl2ir [options] *definition-filename-list*

オプション 使用できるオプションは次のとおりです。

```
[-f repository-name] [-c]
[-D identifier[=definition]]
[-I pathname [-I pathname] [...]] [-N{i|e}]
```

機能説明 このコマンドを使用すると、インターフェイス・リポジトリを作成し、そのインターフェイス・リポジトリにインターフェイス定義をロードできます。リポジトリ・ファイルがない場合は作成されます。リポジトリ・ファイルが存在する場合は、指定されたインターフェイス定義がそのファイルにロードされ、その結果、ファイルが更新されます。

このコマンドを使用すると、新しいインターフェイス・リポジトリ・データベース・ファイルが作成されます。

パラメータ *definition-filename-list*

リポジトリ定義を含むファイルを指定します。指定されたファイルは1つの論理ファイルとして見なされ、一度の操作でロードされます。

-f repository-name

インターフェイス・リポジトリ・ファイルのファイル名。-f オプションを指定しない場合、UNIX システムではインターフェイス・リポジトリ・ファイルとして *repository.ifr*、Microsoft Windows 2000 システムでは *repository_1.ifr* が作成されます。

-c

新しいリポジトリを作成します。既存のリポジトリが存在する場合に、このオプションが指定されると、そのリポジトリは無視されて新しいリポジトリで置き換えられます。既存のリポジトリが存在する場合に、このオプションが指定されていないときは、既存のリポジトリが更新されます。

-D identifier[=definition]

プリプロセッサ・ディレクティブの #define と同じ働きをします。つまり、-D オプションは、定義ファイル内にある特定の識別子の代わりに使用するトークン文字列またはマクロを定義します。definition が指

定されていない場合、identifier は 1 に定義されます。複数の `-D` オプションを指定することができます。

`-I pathname`

OMG IDL プリプロセッサ・ディレクティブ `#include` で指定するディレクトリのほか、インクルード・ファイルを検索するディレクトリを指定します。

OMG IDL プリプロセッサ・ディレクティブ `#include` には、システム (`<a.idl>` など) およびユーザ ("`a.idl`" など) の 2 種類があります。システム・ディレクティブ `#include` のパスは、UNIX システムの `/usr/include` および `-I` オプションで指定されるディレクトリで構成されます。システム・ディレクティブ `#include` のパスは、Windows NT システムのローカル・ディレクトリおよび `-I` オプションで指定されるディレクトリで構成されます。

ユーザ・ディレクティブ `#include` のパスは、カレント・ディレクトリおよび `-I` オプションで指定されるディレクトリで構成されます。複数の `-I` オプションを指定できます。

注記 インターフェイス・リポジトリに対するサーバ・プロセスの実行中にインターフェイス・リポジトリにロードされる追加定義は、そのサーバ・プロセスが停止して再起動するまでは受け付けられません。

idltojava(1)

名前 idltojava - OMG によって定義された IDL から Java へのマッピングに基づき、IDL ファイルを Java ソース・コードにコンパイル

形式 idltojava [idltojava Command Flags] [idltojava Command Options] *filename* ...

m3idltojava [idltojava Command Flags] [idltojava Command Options] *filename* ...

idltojava をクライアント・サイドの IDL ファイルで実行する場合は、次のコマンドを使用します。

```
idltojava <flags> <options> <idl-files>
```

idltojava コマンドは、C++ プリプロセッサを必要とし、非推奨の名前の生成に使用されません。コマンド idltojava は、クライアント・サイド ORB に適切な Java コードを生成します。

注記 リモート共同クライアント/サーバとは、コールバック・オブジェクトとして使用されるサーバ・オブジェクトをインプリメントしたクライアントのことです。リモート共同クライアント/サーバのサーバ・ロールは、堅牢性の面で CORBA サーバのサーバ・ロールより大きく劣ります。クライアントおよびサーバのいずれにも、BEA Tuxedo システム管理コンポーネントおよびインフラストラクチャ・コンポーネント (tmadmin、JNDI 登録、ISL/ISH など) はありません。したがって、BEA Tuxedo ATMI システムのスケラビリティ属性および信頼性属性はありません。

m3idltojava をサーバ・サイドの IDL ファイルで実行する場合は、次のコマンドを使用します。

```
m3idltojava <flags> <options> <idl-files>
```

注記 BEA Tuxedo リリース 8.0 では、m3idltojava コンパイラの使用は非推奨になりました。BEA 社は、CORBA Java クライアントおよび CORBA Java 共同クライアント/サーバのクライアント・スタブの生成には、idltojava コンパイラを使用することをお勧めします。

サーバ・サイド ORB は、使用を避ける名前以外の名前を使用するために組み込まれます。コマンド `m3idltojava` は、使用を避ける名前以外の名前を使用して、サーバ・サイド ORB に適切な Java コードを生成します。

機能説明 `idltojava` コマンドは、IDL ソース・コードを Java ソース・コードにコンパイルします。コンパイル後、`javac` コンパイラを使用して、そのソース・コードを Java バイトコードにコンパイルします。

コマンド `idltojava` を使用して、IDL ソース・コードを、コールバックに使用できる汎用クライアント・スタブおよび汎用サーバ・スケルトンに変換します。コマンド `m3idltojava` は、IDL ソース・コードを汎用クライアント・スタブおよび CORBA サーバ・スケルトンに変換する際に使用します。

指定された IDL ファイルからの IDL 宣言は、OMG IDL から Java へのマッピングの指定に従って Java 宣言に変換されます。

BEA Tuxedo システムの CORBA 環境に提供される `idltojava` コンパイラはいくつかの点で機能拡張されており、Sun Microsystems, Inc が提供するオリジナル・バージョンのコンパイラにはない機能です。BEA Tuxedo システム固有の変更について以下にまとめます。

BEA Tuxedo システムの `idltojava` コンパイラの特長は以下のとおりです。

- フラグの機能およびデフォルト値が、Sun Microsystems, Inc. のマニュアルで説明されるものとは異なります。
- 新しい `#pragma` タグを備えています。形式は次のとおりです。`#pragma ID <name> <Repository_id>`
- 新しい `#pragma` タグを備えています。形式は次のとおりです。`#pragma version <name> <m.n>`
- `#pragma` 接頭辞が内部スコープで機能するように拡張されています。空白の接頭辞が復帰します。
- Boolean 型の区別子を持つ共用体を使用できます。
- 複雑な入れ子の内部で宣言できます。

オプション 注記 Sun Microsystems Inc. の `idltojava` コンパイラのマニュアルでは説明されていない、いくつかのオプションが追加されています。

オプション	説明
-j <i>javaDirectory</i>	生成される Java ファイルを、指定されたディレクトリ <i>javaDirectory</i> に書き込みます。このディレクトリは、-p オプションが指定されていても影響を受けません。
-J <i>filesFile</i>	idltojava で生成されるファイルの一覧を <i>filesFile</i> に書き込みます。
-p <i>package-name</i>	生成されたすべての Java を格納する外部パッケージの名前を指定します。これは #pragma javaPackage と同じです。 外部パッケージは取り込む必要があります。外部パッケージの取り込みはコンパイラによって自動的にには行われません。外部パッケージがなくても idltojava コンパイラは Java ファイルを生成しますが、*.java ファイルのコンパイルの際に Java コンパイラ・エラーが発生します。
以下のオプションは、C/C++ コンパイラ・オプション (cpp) と同等です。	
-I <i>directory</i>	IDL ファイルに #include で取り込むファイルを検索するディレクトリまたはパスを指定します。このオプションはプリプロセッサに渡されます。
-D <i>symbol</i>	IDL ファイルの前処理の際に定義するシンボルを指定します。このオプションはプリプロセッサに渡されます。
-U <i>symbol</i>	IDL ファイルの前処理の際に未定義にするシンボルを指定します。このオプションはプリプロセッサに渡されます。

コマンド・フラグは、ここで示すとおり指定するとオンになり、フラグの前に no- を付けるとオフになります。たとえば、C プリプロセッサが入力 IDL ファイルで実行されないようにするには、-fno-cpp と指定します。

次の表にすべてのフラグを示します。

フラグ	説明
-----	----

セクション 1 - コマンド

<code>-flist-flags</code>	すべての <code>-f</code> フラグの状態を出力します。このフラグのデフォルト値は <code>off</code> です。
<code>-flist -debug-flags</code>	デバッガ・フラグの一覧を表示します。
<code>-fcaseless</code>	キーワードおよび識別子の大文字と小文字を区別しないように指定します。このフラグのデフォルト値は <code>on</code> です。
<code>-fclient</code>	IDL ファイルをクライアント・サイドで生成します。このフラグのデフォルト値は <code>off</code> です。
<code>-fcpp</code>	IDL ソースを <code>idltocpp</code> コンパイラでコンパイルする前に C/C++ プリプロセッサで実行します。このフラグのデフォルト値は <code>on</code> です。
<code>-fignore-duplicates</code>	重複する定義を無視します。このオプションは、一度に複数の IDL ファイルをまとめてコンパイルする際に便利です。このフラグのデフォルト値は <code>off</code> です。
<code>-flist-options</code>	コマンド行で指定されるオプションの一覧を表示します。このフラグのデフォルト値は <code>off</code> です。
<code>-fmap-included-files</code>	プリプロセッサ・ディレクティブ <code>#include</code> で取り込む定義に対して <code>java</code> ファイルを生成します。このフラグのデフォルト値は <code>off</code> です。取り込んだ定義に対して <code>java</code> ファイルは生成されません。
<code>-fserver</code>	IDL ファイルをサーバ・サイドで生成します。このフラグのデフォルト値は <code>off</code> です。
<code>-fverbose</code>	コンパイル時にコンパイラからコメントを表示します。このフラグのデフォルト値は <code>off</code> です。
<code>-fversion</code>	コンパイラのバージョンおよびタイムスタンプを表示します。このフラグのデフォルト値は <code>off</code> です。
<code>-fwarn-pragma</code>	指定された <code>#pragma</code> が認識できない場合や指定に誤りがある場合に、警告メッセージを表示します。このフラグのデフォルト値は <code>on</code> です。
<code>-fwrite-files</code>	生成された <code>java</code> ファイルを書き込みます。このフラグのデフォルト値は <code>on</code> です。ファイルを実際には書き込まずにエラーがないかどうかを調べる場合は、 <code>-fno-write-files</code> と指定する必要があります。

注意 BEA Tuxedo システムの idltojava コンパイラが `#pragma` を処理する方法は、Sun Microsystems, Inc. の idltojava コンパイラでの処理方法とは少し異なります。

```
RepositoryPrefix="prefix"
```

IDL ファイルの先頭に `#pragma prefix "requested prefix"` という行を使用して、デフォルトのリポジトリの接頭辞を要求できます。

```
#pragma javaPackage "package"
```

1つのパッケージを呼び出して、デフォルトのパッケージをラップします。たとえば、IDL モジュールの `M` をコンパイルすると、通常は Java パッケージ `M` が作成されます。しかし、モジュール宣言の前に次のように指定すると、

```
#pragma javaPackage browser
```

パッケージ `M` は `browser` パッケージ内に作成されます。この `pragma` は、1つの IDL モジュール内の定義が複数の製品で使用される場合に便利です。コマンド行オプション `-p` を使用しても同じ結果が得られます。

```
#pragma ID scoped-name "IDL:<path>:<version>"
```

識別子のスコープ指定された名前のリポジトリ ID を指定します。この `pragma` は IDL ファイル内のどの位置にでも使用できます。この `pragma` を構造体や共用体など複雑な型の内部で使用する場合は、スコープ指定された名前の要素のみを指定します。スコープ指定された名前の形式は `outer_name::name::inner_name` です。リポジトリ ID の要素には、一連の識別子をスラッシュ (`/`) で区切って指定します。要素には、10進数 `MM.mm` を指定します。`MM` はメジャー・バージョン番号、`mm` はマイナ・バージョン番号です。

ir2idl(1)

名前	インターフェイス・リポジトリの内容を表示
形式	<code>ir2idl [options] [interface-name]</code>
オプション	使用できるオプションは次のとおりです。 <code>[-f repository-name] [-n]</code> <code>[-t interface-type] [-o filename]</code>
機能説明	このコマンドはインターフェイス・リポジトリの内容を表示します。-o オプションを使用して出力をファイルに送れば、リポジトリから OMG IDL ファイルを抽出できます。デフォルトでは、リポジトリ・ファイルは <code>repository.ifr</code> です。
パラメータ	<p><code>interface-name</code> 内容を表示するインターフェイスの名前。インターフェイス名を指定しない場合、リポジトリ内のすべてのインターフェイスが表示されません。</p> <p><code>-f repository-name</code> インターフェイス定義を検索するリポジトリの名前。-f オプションを指定しない場合、<code>repository.ifr</code> が使用されます。</p> <p><code>-n</code> 継承されたオブジェクトを出力に含めないように指定します。</p> <p><code>-t interface-type</code> 表示するオブジェクトのタイプを指定します。オブジェクトのタイプは、以下のキーワードのいずれかです。 Attribute Constant Exception Interface Method Module Operation Typedef</p> <p>このオプションを指定しない場合、デフォルトですべてのタイプが表示されます。</p>

-o *filename*

取り出した OMG IDL 文を書き込むファイルを指定します。デフォルトは標準出力です。

irdel(1)

名前 指定されたオブジェクトをインターフェイス・リポジトリから削除

形式 `irdel [-f repository-name] [-i id] object-name`

機能説明 このコマンドは、指定されたインターフェイスをリポジトリから削除します。削除できるのは、ほかのインターフェイスから参照されていないインターフェイスだけです。デフォルトでは、リポジトリ・ファイルは `repository.ifr` です。

パラメータ `-f repository-name`

オプション。インターフェイス・リポジトリを指定します。
`repository-name` には、インターフェイス・リポジトリのファイルを指定します。このオプションが指定されていない場合、デフォルトで `repository.ifr` が使用されます。

`-i id`

指定されたオブジェクトのリポジトリ `id`。 `id` は第 2 レベルのルックアップとして使用されます。この `id` と指定されたオブジェクトの `id` が一致しない場合、オブジェクトは削除されません。

`object-name`

リポジトリから削除するインターフェイスの名前。名前は、単純なオブジェクト名でもスコープ指定された名前でもかまいません。たとえば、`MOD1::INTERF2::OP3` と指定できます。この場合、操作 `OP3` はインターフェイス `INTERF2` 内にあり、そのインターフェイスはアプリケーション `MOD1` 内にあります。

ISL(1)

名前 IIOP を使用してリモート BEA Tuxedo クライアントから BEA Tuxedo オブジェクトにアクセスできるようにする

形式 ISL SRVGRP="identifier"
 SRVID="number"
 CLOPT="[-A] [servopts options] -- -n netaddr
 [-C {detect|warn|none}]
 [-d device]
 [-E principal_name]
 [-K {client|handler|both|none}]
 [-m minh]
 [-M maxh]
 [-T Client-timeout]
 [-x mpx-factor]
 [-H external-netaddr]
 #options for Outbound IIOP
 [-O]
 [-o outbound-max-connections]
 [-s Server-timeout]
 [-u out-mpx-users]
 #options for SSL
 [-a]
 [-R renegotiation-interval]
 [-S secure port]
 [-v {detect|warn|none}]
 [-z [0|40|56|128]]
 [-Z [0|40|56|128]]"

機能説明 IIOP サーバ・リスナ (ISL) は BEA Tuxedo が提供するサーバ・コマンドです。このコマンドを使用すると、IIOP を使用してリモートの BEA Tuxedo クライアントから BEA Tuxedo オブジェクトにアクセスできるようになります。アプリケーション管理者は、SERVERS セクションで IIOP サーバ・リスナをアプリケーション・サーバとして指定することで、アプリケーション・オブジェクトへのアクセスを可能にできます。関連するコマンド行オプションを使用して、IIOP サーバ・リスナおよび IIOP サーバ・ハンドラのパラメータを指定します。

位置指定、サーバ・グループ、サーバ ID、およびその他の汎用サーバ関連パラメータは、サーバ用の標準コンフィギュレーション・ファイルのメカニズムを使用して ISL に関連付けられます。ISL コマンド行オプションを使用してカスタマイズすることもできます。

各 ISL はアプリケーションの一部としてブートされるため、リモートの BEA Tuxedo クライアントが多数存在しても、1 つの周知のネットワーク・アドレスを使用すればアプリケーションに簡単にアクセスできます。IIOP サーバ・ハンドラは、着信時の負荷の状況に基づき、必要に応じて ISL によって動的に起動および停止されます。

共同クライアント / サーバの場合、リモートの共同クライアント / サーバ ORB が双方向の IIOP 接続をサポートしていれば、ISL はリモート共同クライアント / サーバに対するアウトバウンド呼び出しに同じインバウンド呼び出しを使用できます。また、ISL は、ISH に接続されていない共同クライアント / サーバ内のオブジェクトに対してアウトバウンド呼び出し (アウトバウンド IIOP) を行うこともできます。この機能は、`-o` オプションが指定されると有効になります。関連するコマンド行オプションを使用して、アウトバウンド IIOP サポートを構成することができます。

パラメータ `-A`

ブート時に ISL がすべてのサービスを提供するように指定します。これはデフォルト値ですが、システムが提供するサーバとアプリケーション・サーバとの違いを強調するために示してあります。アプリケーション・サーバはブート時に、利用可能なサービスのサブセットのみを提供できます。二重ダッシュ (`--`) は、ブート後に ISL に渡されるパラメータの開始位置を示します。

`CLOPT` パラメータでは、`CLOPT` 文字列の二重ダッシュ (`--`) の後に、以下のオプションを指定します。

`-n netaddr`

サーバ・リスナがリモート CORBA クライアントからの接続の受け付けに使用するネットワーク・アドレスを指定します。リモート・クライアントは、環境変数 (`TOBJADDR`) にこの値を設定しておくか、Bootstrap オブジェクトのコンストラクタで値を指定する必要があります。詳細については、『C++ Programming Reference』を参照してください。これは必須パラメータです。

TCP/IP アドレスは、次のいずれかの形式で指定する必要があります。

```
"/hostname:port_number"  
"/#.#.#.#:port_number"
```

最初の形式では、ドメインは、ローカル名の解決機能 (通常は DNS) を使用して `hostname` のアドレスを検出します。ローカル・マシンが

ホストでなければなりません。ローカル名の解決機能により、`hostname` はローカル・マシンのアドレスに間違いなく解決される必要があります。

注記 ホスト名は文字で始まる必要があります。

2 つ目の形式の "#.#.#.#" は、10 進数がドットで区切られたものです。ドット区切りの 10 進数形式では、それぞれの # に 0 ~ 255 の数字を指定します。このドット区切りの 10 進数は、ローカル・マシンの IP アドレスを表します。

いずれの形式でも、`port_number` には、ドメイン・プロセスが着信要求を受け付ける TCP ポート番号を指定します。`port_number` には、0 ~ 65,535 の数字または名前を指定します。`port_number` に指定する名前は、ローカル・マシンのネットワーク・サービス・データベース内に存在する名前ではありません。

注記 Java の `Tobj_Bootstrap` オブジェクトは、`port_number` を `short` 型として格納します。したがって、Java クライアントからの接続をサポートする場合には、`port_number` に指定する値は 0 ~ 32,767 になります。

注記 プログラムが `Bootstrap` オブジェクトのコンストラクタまたは `TOBJADDR` で指定するネットワーク・アドレスは、アプリケーションの `UBBCONFIG` ファイルのネットワーク・アドレスと正確に一致する必要があります。アドレスの形式や、大文字 / 小文字も識別されます。アドレスが一致しない場合、`Bootstrap` コンストラクタの呼び出しは失敗し、次のようなエラー・メッセージが表示されます。

```
ERROR:Unofficial connection from client at
<tcp/ip address>/<port-number>:
```

たとえば、ネットワーク・アドレスとして `//TRIXIE:3500` が `ISL` コマンド行オプション文字列で指定された場合に、`Bootstrap` オブジェクトのコンストラクタまたは `TOBJADDR` で `//192.12.4.6:3500` や `//trixie:3500` が指定されていると、接続試行は失敗します。

UNIX システムでは、ホスト・システム上で `uname -n` コマンドを使用して、大文字と小文字のどちらが使用されているかを判断し

ます。Windows NT システムでは、[コントロール パネル]の[ネットワーク]を開いて、大文字と小文字のどちらが使用されているかを判断できます。

注記 BEA Tuxedo システムのワークステーション・リスナ (WSL) とは異なり、ネットワーク・アドレスの形式には //host:port しか使用できません。BEA Tuxedo サーバではホスト名とポート番号が使用されることが理由です。そのため、ホスト名が 16 進数形式で表されることはなく、ホスト名はドット区切りの IP アドレス形式でのみサーバに渡されます。

[-a]

リモート・アプリケーションから SSL 接続を受け付ける際に証明書ベースの認証を有効にします。

[-C detect|warn|none]

非公式な方法で IIOP リスナ / ハンドラへの接続が確立される際の、IIOP リスナ / ハンドラの動作を指定します。デフォルト値は detect です。

CORBA クライアントから IIOP リスナ / ハンドラへの接続には Bootstrap オブジェクトを使用するのが正しい方法です。IOR から直接確立される接続は「非公式の」接続です。たとえば、クライアントが Bootstrap オブジェクトを使用して 1 つの IIOP リスナ / ハンドラに接続した後で、2 つ目の IIOP リスナ / ハンドラのホストとポートを含む IOR を不注意で使用して、IIOP リスナ / ハンドラに接続する場合がありますが、このようなケースは稀です。通常、クライアントが使用する IOR には、Bootstrap オブジェクトを使用してクライアントが接続している IIOP リスナ / ハンドラのホストおよびポートが含まれています。このような IOR を使用しても接続を新たに確立することはできません。

注記 非公式な接続を使用すると、トランザクションを使用するリモート・クライアント・アプリケーションで問題が発生する可能性があります。アプリケーションは、同じトランザクション内での公式および非公式な両方の接続で呼び出しが成功していると認識しますが、実際には、公式な接続での呼び出しにしか ACID (原子性、一貫性、独立性、持続性) は保証されません。

`detect` を指定すると、ISL/ISH は、非公式な接続に対して `NO_PERMISSION` 例外を発行します。`warn` を指定すると、ISL/ISH は非公式な接続が検出された場合に、ユーザ・ログ例外にメッセージを書き込みますが、例外は発行しません。`none` を指定すると、ISL/ISH は非公式な接続を無視します。

`[-d device]`

サーバ・リスナおよびそのサーバ・ハンドラでネットワーク・アクセスに使用するデバイス・ファイル名を指定します。ソケットなど、トランスポート・プロバイダによってはデバイス名を必要としない場合もあるため、これはオプション・パラメータです。ただし、TLI などのトランスポート・プロバイダではデバイス名が必要です。TLI の場合は、このオプションは必須です。このパラメータにデフォルト値はありません。このパラメータは Windows 2000 システムには適用されません。

`[-E principal_name]`

オプション・パラメータ。信頼性のある接続プールを確立するために必要なプリンシパルのアイデンティティを指定します。信頼性のある接続プールは、CORBA アプリケーションがユーザ認証を必要とする場合にのみ確立できます。

リモート・クライアント・アプリケーションが、信頼性のある接続プールの一部でない接続上で各要求のセキュリティ情報を送信しても、そのセキュリティ情報は無視されます。

`[-K {client|handler|both|none}]`

クライアントまたは ISH プロセスのいずれか、あるいはその両方で、ネットワーク・プロバイダの `KEEPALIVE` オプションをアクティブにします。このオプションを使用すると、プロトコル・スタック・レベルでアイドル接続の状態が頻繁にテストされるので、ネットワーク障害がすばやく検出され、その信頼性も向上します。この機能の利用度とタイムアウトのしきい値は、オペレーティング・システムの調整可能なパラメータによって決まります。

`client` を指定すると、このオプションはクライアントに対して設定されます。`handler` を指定すると、ISL に対して設定されます。`both` を指定すると、接続の両端に設定されます。デフォルト値は `none` です。この場合、接続の両端のいずれにも `KEEPALIVE` オプションは設定されません。

注記 `KEEPALIVE` の間隔はオペレーティング・システムのパラメータによって決まるので、この値を変更すると `KEEPALIVE` が有効になっているその他のアプリケーションにも影響します。多くのプラットフォームではデフォルト値は長めの 2 時間に設定されています。

このオプションを適用できないプラットフォームもあります。`KEEPALIVE` オプションを指定しても ISH のマシンで使用できない場合は、ユーザ・ログ警告メッセージが生成されます。`KEEPALIVE` が要求されていてもクライアントのマシンで使用できない場合は、その設定は無視されます。

`[-m minh]`

この ISL と共に使用できるハンドラの最小数を指定します。デフォルト値は 0 です。ISL はブートすると即座にこの最小数の ISH を移動し、管理者が ISL にシャットダウンを発行するまでは、ISH がこの最小値を下回ることはありません。このパラメータのデフォルト値は 0 で、有効な範囲は 0 ~ 255 です。

`[-M maxh]`

この ISL と共に使用できるハンドラの最大数を指定します。ハンドラは、リモート BEA Tuxedo クライアントからのシステム・アクセスの要求に応じて起動します。このパラメータのデフォルト値は、この ISL の多重係数 (小数点以下切り上げ) で論理マシンの `MAXWSCLIENTS` の設定値を割った値です。多重係数については、後述の `-x` オプションを参照してください。このパラメータの有効範囲は 1 ~ 4,096 です。この値は `minh` 以上でなければなりません。

`[-T Client-timeout]`

非アクティブ・クライアントのタイムアウト・オプションを指定します。非アクティブ・クライアントのタイムアウトとは、クライアントがアイドル状態を継続できる時間 (分単位) です。クライアントがこの時間内に要求を行わなかった場合、IIOP リスナ / ハンドラはクライアント接続を切断します。この引数を指定しない場合、または 0 が指定されている場合、タイムアウトは発生しません。

`[-x mpx-factor]`

オプション・パラメータ。各 ISH で使用する多重化のレベルを制御します。このパラメータの値には、各 ISH で同時にサポートできるリモート BEA Tuxedo クライアントの数を指定します。ISH は、新しいリモート BEA Tuxedo クライアントを処理するために、必要に応じて

新しいハンドラを起動します。この値は 1 ~ 4,096 でなければなりません。デフォルト値は 10 です。

[`-H external netaddr`]

ISL のクライアントに返されるインターオペラブル・オブジェクト・リファレンスでホストおよびポートとして設定する外部ネットワーク・アドレスを指定します。形式は、`ISL CLOPT -n netaddr` オプションと同じです。この機能は、IIOP、つまりリモート・クライアントがファイアウォールを越えて ISL に接続する必要がある場合に便利です。

[`-O`]

このオプション (大文字 O) を使用すると、ISH に接続されたクライアント内に存在しないオブジェクトへのアウトバウンド IIOP が有効になります。`-o` オプションは多少のリソースを必要とするので、デフォルトではアウトバウンド IIOP は使用しないように設定されます。

[`-o outbound-max-connections`]

このオプション (小文字 o) は、各 ISH で可能なアウトバウンド接続の最大数を指定します。このオプションは、実際には、この ISL の制御下にある単一の ISH で一度に同時にアクティブにできるアウトバウンド IIOP ソケットの数を制限します。

このオプションを指定する場合は、`-O` (大文字 O) オプションも指定する必要があります。このオプションの値は 0 より大きく 4,096 以下でなければなりません。また、このオプションの値 (`outbound-max-connections`) とハンドラの最大数をかけた数は、32,767 未満でなければなりません。デフォルト値は 20 です。

[`-R renegotiation-interval`]

ネゴシエーションを行う間隔を分単位で指定します。指定された時間おきにネゴシエーションが行われない場合、IIOP リスナ/ハンドラは、クライアントに対してインバウンド接続のセッションのネゴシエーションを行うように要求するか、アウトバウンド接続であれば実際に再度ネゴシエーションを行います。デフォルト値は 0 で、セッションのネゴシエーションは定期的には行われません。

[`-S secure-port`]

SSL プロトコルを使用した安全な接続の受け付けに IIOP リスナ/ハンドラが使用するポート番号を指定します。IIOP リスナ/ハンドラで安

全な接続のみを使用するように設定するには、`-s` および `-n` オプションで指定するポート番号に同じ値を設定します。

`[-s Server-timeout]`

サーバ・タイムアウトとは、リモート・サーバがアイドル状態を継続できる時間（分単位）です。リモート・サーバがこの時間内に要求を受け取らなかった場合、ISH はそのサーバへのアウトバウンド IOP 接続を切断します。その後で要求があれば、ISH はリモート・サーバに再接続します。このオプションは、不安定なサーバ・プラットフォームに使用できます。ISH はこの時間に達するまでは接続を切断しないが、この時間の経過後に接続が切断されることを保証しないという点でこれは最善の値です。このオプションを指定する場合は、`-o`（大文字 O）オプションも指定する必要があります。この値は 1 以上でなければなりません。このオプションが指定されていない場合は、デフォルトで 60（1 時間）が設定されます。

`[-u out-mpx-users]`

オプション・パラメータ。各 ISH で使用するアウトバウンド多重化のレベルを制御します。このオプションの値は、ISH での各アウトバウンド IOP 接続で同時にサポートできるアウトバウンド IOP ユーザ（ネイティブ・クライアントまたはサーバ）の数を指定します。ISH は、最大 `out-mpx-users` 個までの新しいユーザを処理するために、必要に応じて新しい ISH を起動します。このオプションを指定する場合は、`-O`（大文字 O）オプションも指定する必要があります。このオプションは 0（ゼロ）より大きく 1,024 以下でなければなりません。デフォルト値は 10 です。

`[-v {detect|warn|none}]`

BEA オブジェクト・リクエスト・ブローカ (ORB) によって開始されたアウトバウンド接続のピアのデジタル証明書が、Secure Sockets Layer (SSL) プロトコル・ハンドシェイクの一部として受信された場合に、IOP リスナ / ハンドラがどのように動作するかを指定します。検証は安全な接続を開始した側でのみ行われます。この検証によって、サーバのデジタル証明書内のドメイン名で指定された同じネットワーク・アドレスにピア・サーバが実際に存在することが確認されます。この検証は、技術的には SSL プロトコルの一部ではありませんが、Web ブラウザで行われるチェックに似ています。

`detect` を指定すると、BEA ORB は、接続確立に使用するオブジェクト・リファレンスで指定されたホストが、ピア・サーバのデジタル証明書で指定されたドメイン名と一致するかどうかを確認します。比較に失敗した場合、BEA ORB はピアの認証を拒否し、接続をドロップします。`detect` がデフォルト値です。

`warn` を指定すると、BEA ORB は、接続確立に使用するオブジェクト・リファレンスで指定されたホストが、ピア・サーバのデジタル証明書で指定されたドメイン名と一致するかどうかを確認します。比較に失敗した場合、BEA ORB はユーザ・ログにメッセージを書き込みますが、接続は継続して処理されます。

`none` を指定すると、BEA ORB はピアの検証を行わずに接続を継続して処理します。

`-v` パラメータは、SSL および LLE (リンク・レベル暗号化) のライセンスがインストールされている場合にのみ使用できます。

`[-z |0|40|56|128]`

クライアントと IIOP リスナ / ハンドラの間でネットワーク接続を確立する際に必要な最小レベルの暗号化を指定します。0 は暗号化が行われないことを示し、40、56、および 128 は暗号化キーの長さ (ビット単位) を指定します。この最小レベルの暗号化が満たされない場合、接続は確立されません。このオプションは、SSL および LLE (リンク・レベル暗号化) のライセンスがインストールされている場合にのみ使用できます。

`[-z |0|40|56|128]`

クライアントと IIOP リスナ / ハンドラの間でネットワーク接続を確立する際に使用できる最大レベルの暗号化を指定します。0 は暗号化が行われないことを示し、40、56、および 128 は暗号化キーの長さ (ビット単位) を指定します。デフォルトでは、ライセンスで指定された機能が使用されます。このオプションは、SSL および LLE (リンク・レベル暗号化) のライセンスがインストールされている場合にのみ使用できます。

移植性 UNIX および Microsoft Windows NT オペレーティング・システムでは、IIOP サーバ・リスナは BEA Tuxedo が提供するサーバとしてサポートされません。

セクション 1 - コマンド

相互運用性 ISL は、IIOP 準拠の ORB と共に運用できます。

リモート・オブジェクトのタイプおよび目的のアウトバウンド IIOP 設定によっては、追加のプログラミング作業が必要となる場合もあります。表 15 に、オブジェクトのタイプおよびアウトバウンド IIOP 設定別の要件を示します。

表 15 アウトバウンド IIOP を使用する際のプログラミング要件

オブジェクトのタイプ	非対称での要件	ペア接続での要件	双方向での要件
リモート共同クライアント/サーバ	ISL CLOPT -O オプションを設定する	Tobj_Bootstrap::register_callback_port メソッドを使用してコールバック・ポートを登録する	CORBA::ORB::create_policy メソッドを使用して POA で BiDirPolicy を設定する
外部 (非 CORBA) ORB	ISL CLOPT -O オプションを設定する	適用外	外部 ORB が POA および BiDirPolicy をサポートする場合は、CORBA::ORB::create_policy メソッドを使用して POA で BiDirPolicy を設定する
リモート・クライアント	リモート・クライアントはサーバではないため、アウトバウンド IIOP は使用できない		
ネイティブ共同クライアント/サーバ	アウトバウンド IIOP は使用されない		
ネイティブ・クライアント	アウトバウンド IIOP は使用されない		
ネットワーク・アドレス	ISL を実行するローカル・マシンが TCP/IP アドレッシングを使用している場合を考えます。ローカル・マシンの名前は backus.company.com、アドレスは 155.2.193.18 です。ISL はポート番号 2334 で要求を受け付けます。-1 オプションで指定されるアドレスは、次のとおりです。		

```
//155.2.193.18:2334
```

```
//backus.company.com:2334
```

使用例

```
*SERVERS
```

```
ISL SRVGRP="ISLGRP" SRVID=1002 RESTART=Y GRACE=0
```

```
CLOPT="-A -- -n //piglet:1900 -d /dev/tcp"
```

m3idltojava(1)

注記 BEA Tuxedo リリース 8.0 では、m3idltojava コンパイラの使用は非推奨になりました。BEA 社は、CORBA Java クライアントおよび CORBA Java 共同クライアント / サーバのクライアント・スタブの生成には、idltojava コンパイラを使用することをお勧めします。

名前 Object Management Group (OMG) のインターフェイス定義語 (IDL) ファイルをコンパイルし、Java でインプリメントするインターフェイス定義に必要なクライアント・スタブおよびサーバ・スケルトン・ファイルを生成します。このコマンドは、Java サーバ・アプリケーションを作成する場合にのみ使用します。

形式 m3idltojava [-p] [-j javaDirectory] [-Idirectory][-Dsymbol] [-Usymbol] [-foptions] idl-filename...

機能説明 m3idltojava コマンドは、OMG IDL ソース・ファイルを Java ソース・コードにコンパイルします。コンパイル後、javac コンパイラを使用して、そのソース・コードを Java バイトコードにコンパイルします。指定された OMG IDL ファイルからの OMG IDL 宣言は、OMG IDL から Java へのマッピングに従って Java 宣言に変換されます。

提供されている idl-filename ファイルを指定すると、m3idltojava コマンドは、サーバ・アプリケーションの OMG IDL ファイルに定義される各インターフェイスに対して以下のファイルを生成します。

interface-name.java

OMG IDL ファイル内のインターフェイス定義の Java 版が格納されます。それぞれのインターフェイス・インプリメンテーションは、org.omg.CORBA.Object クラスに拡張されます。

*_interface-name*Stub.java

クライアント・スタブ・ファイルです。

*_interface-name*ImplBase.java

サーバ・スケルトン・ファイルです。これは、サーバ・アプリケーションのオブジェクト・インプリメンテーション・クラスによって拡張されます。

*interface-name*Helper.java

オブジェクトのヘルパー・クラスが格納されます。

`interface-nameHolder.java`

オブジェクトのホルダー・クラスが格納されます。

`m3idltojava` コンパイラは、クライアント・スタブおよびサーバ・スケルトン・ファイルを生成します。以前のバージョンのファイルはすべて上書きされます。

不明なオプションがこのコマンドに渡されると、そのオプションおよび使用方法を通知するメッセージがユーザに表示され、コンパイルは実行されません。

パラメータ

`idl-filename`

OMG IDL 文を含む 1 つまたは複数のファイルの名前を表します。

オプション

`-p package`

生成された Java クラスを、指定されたパッケージにその一部として含めます。コンパイラによって適切なディレクトリ階層が作成され、パッケージと対応するディレクトリ内にこれらのファイルが格納されます。`-j` オプションを指定すると、指定されたディレクトリの下に階層が作成されます。指定しない場合は、カレント・ディレクトリの下に階層が作成されます。OMG IDL ソース・ファイルで `#pragma javaPackage` を使用して、このオプションを変更することができます。

`-j javaDirectory`

生成された Java ファイルを、指定されたディレクトリに書き込みます。このディレクトリは、`-p` オプションが指定されていても影響を受けません。

`-Idirectory`

OMG IDL プリプロセッサ・ディレクティブ `#include` で指定するディレクトリのほか、インクルード・ファイルを検索するディレクトリを指定します。複数のディレクトリを指定する場合は、複数の `-I` オプションを指定します。

OMG IDL プリプロセッサ・ディレクティブ `#include` には、システム (`<a.idl>` など) およびユーザ (`"a.idl"` など) の 2 種類があります。システム・ディレクティブ `#include` のパスは、システム・インクルード・ディレクトリおよび `-I` オプションで指定されるディレクトリで構成されます。ユーザ・ディレクティブ `#include` のパスは、`#include` ディレクティブを含むファイルの場所と、その後続く、

システム・ディレクティブ `#include` に指定されたパスで構成されます。

デフォルトでは、`#include` ディレクティブで取り込むファイル内のテキストは、クライアント・コードおよびサーバ・コードには取り込まれません。

-dsymbol

OMG IDL ファイルの前処理の際に定義するシンボルを指定します。`m3idltojava` コマンドは、このシンボルをプリプロセッサに渡します。

-usymbol

OMG IDL ファイルの前処理の際に未定義にするシンボルを指定します。`m3idltojava` コマンドは、このシンボルをプリプロセッサに渡します。

-foptions

以下のオプションは、ここ示されるとおりに指定するとオンになり、文字列 `no-` を付けるとオフになります。たとえば、C プリプロセッサが入力 OMG IDL ファイルで実行されないようにするには、`-fno-cpp` と指定します。

-flist-flags

すべての `-f` フラグの状態を表示します。デフォルトでは、このオプションは無効です。

-fclient

クライアント・アプリケーション・ファイルを生成します。デフォルトでは、このオプションは有効です。

-fserver

サーバ・アプリケーション・ファイルを生成します。デフォルトでは、このオプションは有効です。

-fverbose

`m3idltojava` コマンドを冗長モードで機能させます。特に、コマンド出力をその標準出力に書き出します。デフォルトでは、このオプションは無効です。

-fversion

コンパイラのバージョンおよびタイムスタンプを出力します。デフォルトでは、このオプションは無効です。

使用例 次のコマンドは、Simple.idl に対してサーバ・アプリケーション・ファイルのみを生成します。

```
m3idltojava -fno-client Simple.idl
```

次のコマンドは、Simple.idl に対してクライアント・アプリケーション・ファイルのみを生成します。

```
m3idltojava -fno-server Simple.idl
```

mkfldhdr、mkfldhdr32(1)

名前	mkfldhdr、mkfldhdr32 - フィールド・テーブルからヘッダ・ファイルを作成
形式	mkfldhdr [-d outdir] [field_table...] mkfldhdr32 [-d outdir] [field_table...]
機能説明	<p>mkfldhdr は、各フィールド・テーブルを C プログラムにおける組み込みに適するヘッダ・ファイルに変換します。結果として得られたヘッダ・ファイルからは、フィールド名からフィールド ID への変換を行うための #define マクロが得られます。ヘッダ・ファイルの名前は、変換する各ファイルのファイル名に .h を付けることによって得られます。</p> <p>フィールド・テーブル名は、コマンド行で指定してもかまいません。また、各ファイルは対応するヘッダ・ファイルに変換されます。</p> <p>フィールド・テーブル名をコマンド行で指定しなかった場合には、プログラムは変換するフィールド・テーブルのリストとして FIELDTBLS 環境変数を、またファイルの検索に使用するディレクトリのリストとして FLDTBLDIR 環境変数を使用します。FIELDTBLS は、フィールド・テーブル・ファイル名の、カンマで区切られたリストを指定します。FIELDTBLS に値がない場合、(唯一の)フィールド・テーブル・ファイルの名前として fld.tbl が使用されます(この場合、結果として得られるヘッダ・ファイルは fld.tbl.h になります)。FLDTBLDIR 環境変数は、複数のディレクトリがコロンで区切られた並びであり、この中から名前が絶対パス名でないフィールド・テーブルが検索されます。フィールド・テーブルの検索は、UNIX システムが PATH 変数を使用して実行可能コマンドの検索を行う方法と非常によく似ています。FLDTBLDIR の定義がなければ、カレント・ディレクトリだけが検索されます。このため、コマンド行にフィールド・テーブル名の指定がなく、FIELDTBLS と FLDTBLDIR が設定されていない場合には、mkfldhdr はカレント・ディレクトリのフィールド・テーブル fld.tbl をヘッダ・ファイル fld.tbl.h に変換します。</p> <p>-d オプションは、出力ヘッダ・ファイルを現在の作業ディレクトリ以外のディレクトリに作成したいときに使用します。</p> <p>mkfldhdr32 は、32 ビットの FML で使用します。環境変数 FIELDTBLS32 および FLDTBLDIR32 を使用します。</p>

エラー フィールド・テーブルのロードに失敗した場合、あるいは出力ファイルを生成できなかった場合には、エラー・メッセージが出力されます。

使用例

```
FLDTBLDIR=/project/fldtbls
FIELDTBLS=maskftbl,DBftbl,miscftbl,
export FLDTBLDIR FIELDTBLS
```

この例では、ディレクトリ `/project/fldtbls` のファイル `maskftbl`、`DBftbl` および `miscftbl` を処理することにより、`mkfldhdr` は、`#include` ファイル `maskftbl.h`、`DBftbl.h` および `miscftbl.h` をカレント・ディレクトリに作成します。

上記の例のように環境変数を設定すると、コマンド `mkfldhdr -d$FLDTBLDIR` は同じ入力フィールド・テーブル・ファイルを処理して、同じ出力ファイルを生成しますが、それらのファイルは環境変数 `FLDTBLDIR` の値で示されるディレクトリに格納されます。

また、次のコマンド `mkfldhdr myfields` は、入力ファイル `myfields` を処理して、カレント・ディレクトリに `myfields.h` を生成します。

関連項目 「[FML 関数の紹介](#)」、[field_tables\(5\)](#)

mklanginfo(1)

名前 mklanginfo - ロケールの言語情報定数をコンパイル

形式 mklanginfo [*fname*]

機能説明 このプログラムは引数として指定されたファイルを取り込み、その入力を \$TUXDIR/locale/*xx*/LANGINFO に配置するのに適したファイルに変換します (*xx* は特定のロケール)。ファイルの引数が指定されていない場合は、標準入力を使用されます。この言語値は、[setlocale\(3c\)](#)、[strftime\(3c\)](#)、および [nl_langinfo\(3c\)](#) で使用されます。

mklanginfo は入力行を読み取ります。このとき、空白または `!` で始まる行は無視され、力行は次の' ればなりません。

```
<token> = "value"
```

トークン (*token*) と二重引用符値との間の文字は、トークンの後に空白があれば、二重引用符を除けば何でもかまいません。value がヌル文字列であると、その行は無視されます。そうでなければ、トークンは 1 から 48 までの任意の整数とするか、以下に示す文字列のいずれかとしなければなりません。

整数 文字列 値 1

DAY_1	週の第 1 日目 (例: Sunday)	2
DAY_2	週の第 2 日目 (例: Monday)	3
DAY_3	週の第 3 日目 (例: Tuesday)	4
DAY_4	週の第 4 日目 (例: Wednesday)	5
DAY_5	週の第 5 日目 (例: Thursday)	6
DAY_6	週の第 6 日目 (例: Friday)	7
DAY_7	週の第 7 日目 (例: Saturday)	8
ABDAY_1	週の第 1 日目の略称 (例: Sun)	9
ABDAY_2	週の第 2 日目の略称 (例: Mon)	10
ABDAY_3	週の第 3 日目の略称 (例: Tue)	11
ABDAY_4	週の第 4 日目の略称 (例: Wed)	12
ABDAY_5	週の第 5 日目の略称 (例: Thu)	13
ABDAY_6	週の第 6 日目の略称 (例: Fri)	14
ABDAY_7	週の第 7 日目の略称 (例: Sat)	15
MON_1	年の最初の月 (例: January)	16
MON_2	年の 2 番目の月 (例: February)	17
MON_3	年の 3 番目の月 (例: March)	18
MON_4	年の 4 番目の月 (例: April)	19

MON_5	年の 5 番目の月 (例: May) 20
MON_6	年の 6 番目の月 (例: June) 21
MON_7	年の 7 番目の月 (例: July) 22
MON_8	年の 8 番目の月 (例: August) 23
MON_9	年の 9 番目の月 (例: September) 24
MON_10	年の 10 番目の月 (例: October) 25
MON_11	年の 11 番目の月 (例: November) 26
MON_12	年の 12 番目の月 (例: December) 27
ABMON_1	年の最初の月の略称 (例: Jan) 28
ABMON_2	年の 2 番目の月の略称 (例: Feb) 29
ABMON_3	年の 3 番目の月の略称 (例: Mar) 30
ABMON_4	年の 4 番目の月の略称 (例: Apr) 31
ABMON_5	年の 5 番目の月の略称 (例: May) 32
ABMON_6	年の 6 番目の月の略称 (例: Jun) 33
ABMON_7	年の 7 番目の月の略称 (例: Jul) 34
ABMON_8	年の 8 番目の月の略称 (例: Aug) 35
ABMON_9	年の 9 番目の月の略称 (例: Sep) 36
ABMON_10	年の 10 番目の月の略称 (例: Oct) 37
ABMON_11	年の 11 番目の月の略称 (例: Nov) 38
ABMON_12	年の 12 番目の月の略称 (例: Dec) 39
RADIXCHAR	基数文字 (例: '.') 40
THOUSEP	1000 位の区切り文字 41
YESSTR	肯定応答文字 (例: yes) 42
NOSTR	否定応答文字 (例: no) 43
CRNCYSTR	通貨記号 44
D_T_FMT	日付・時刻のフォーマット文字列 (例: &dlq;%a%b%d%H:%M`rq;) 45
D_FMT	日付のフォーマット文字列 (例: &dlq;%m/%d/%y") 46
T_FMT	時刻のフォーマット文字列 (例: &dlq;H:%M:%S&" 47
AM_FMT	午前を表す接尾辞 (例: AM) 48
PM_FMT	午後を表す接尾辞 (例: PM)

入力行の順序は任意でかまいません (ある入力行が同じ値で複数回現れる場合は、その値の最後の行が使用されます)。

ファイルを処理した後、`mklanginfo` は上記の各言語情報定数ごとに文字列の名前と文字列値を上記の順序で標準エラー出力に出力します。指定されていない言語情報定数があれば、それに対応する値としてヌル文字列が使用されます。`nl_langinfo` はこうした未設定定数に対しては C ロケールのデフォルトの値 (U.S. English) を使用します。

コマンド行にファイル名が指定されていると、`mklanginfo` は「コンパイル」された出力を `fname.out` に出力します。指定されていない場合は標準出力に出力します。形式は、すべて NULL で終わる文字列値のリストです（改行文字は入りません）。

診断 ファイルの読み取りエラーや構文エラーがあると、エラー・メッセージが標準エラー出力に書き出された後、終了コード 1 でプログラムは終了します。成功時には、プログラムは終了コード 0 で終了します。

使用例 BEA Tuxedo システムのデフォルト値（ロケール C）は、`$TUXDIR/locale/C/lang.text` にあります。たとえば、UNIX システム・プラットフォームでフランス語を使用するには、次のように指定します。

```
mkdir $TUXDIR/locale/french
cd $TUXDIR/locale/french
cp $TUXDIR/locale/C/lang.text .
ed lang.text
... フランス語に翻訳
w
q
mklanginfo lang.text > LANGINFO
```

ファイル `$TUXDIR/locale/C/lang.text` - C ロケールのデフォルト値
`$TUXDIR/locale/C/LANGINFO` - C ロケールのコンパイル済みファイル
`$TUXDIR/locale/xx/LANGINFO` - xx ロケールのコンパイル済みファイル

注意事項 `mklanginfo` コマンドと、その実行結果として得られる `LANGINFO` ファイルは、`setlocale()`、`strftime()`、あるいは `nl_langinfo()` の BEA Tuxedo システムの互換関数を使用する場合にのみ必要となります。UNIX で提供される関数は、別のファイル・セットおよびファイル形式を使用します。

関連項目 [nl_langinfo\(3c\)](#)、[setlocale\(3c\)](#)、[strftime\(3c\)](#)、[langinfo\(5\)](#)

qmadmin(1)

名前 qmadmin - キュー・マネージャ管理プログラム

形式 [QMCONFIG=<device>] qmadmin [<device>]

機能説明 qmadmin は、次の一覧で示す各種コマンドを使用することにより、メッセージ・キューの作成、検査、および変更を行います。汎用デバイス・リスト (UDL) は、BEA Tuxedo ATMI システムが稼動するマシンに物理記憶空間をマップします。UDL のエントリは、キューとキュー・スペースのメッセージが格納される、ディスク空間を指定します。キュー・スペースのための汎用デバイス・リストが存在する (または存在することになる) デバイス (ファイル) 名は、コマンド行引数として指定することも、環境変数 QMCONFIG を介して指定することもできます。この両方で指定された場合には、コマンド・オプションが使用されます。

qmadmin はシステム付属のサーバとして、通常の初期化プロセスを行わないため、UBBCONFIG ファイルから ULOGPFX の値を参照しません。その結果、qmadmin コマンドによって生成されたすべてのログ・エントリが、現在の作業ディレクトリに書き込まれます。これに対処するには、ULOGPFX 環境変数をユーザ・ログのあるディレクトリのパス名に設定して、エクスポートします。

qmadmin は、大なり記号 > をプロンプトとして使用します。各引数は、空白 (タブまたはスペース、あるいはその両方) で区切って入力されます。空白を含む引数は、二重引用符で囲むことができます。二重引用符で囲まれた引数に二重引用符を含める場合は、引数内の二重引用符をバックスラッシュでエスケープしなければなりません。コマンド行でコマンド情報を指定しなかった場合には、情報の入力を促すプロンプトが表示されます。必須の引数を入力しないと、警告メッセージが表示され、プロンプトが再表示されます。オプション・パラメータに関する情報の入力を求めるプロンプトは表示されません。

コマンド入力用のプロンプトが表示されているときに、q または <CTRL-d> を入力すると、ユーザがプログラムを終了することができます。コマンドからの出力は、BREAK を押すことによって終了させることができ、これによりプログラムは次のコマンドの入力を要求するプロンプトを表示します。コマンド入力用のプロンプトが表示されているときにリターン・キーを押すと、直前に実行したコマンドを繰り返します (ただし、BREAK を行った後以外)。

一度リターン・キーを押してしまったコマンドを完全に取り消す方法はありません。BREAK を押しても、それはコマンドからの出力があればその出力を終了させるだけです。したがって、必ず正確にコマンドをタイプ入力してから、リターン・キーを押すようにしてください。

qmadmin コマンドからの出力は、使用しているページ別表示コマンドに従ってページ別表示されます (後述の `paginate` サブコマンドを参照)。

qmadmin を初めて入力するときは、オープンしているキュー・スペースはありません。キュー・スペースを作成するには `qspacecreate` を実行します。このキュー・スペースをオープンするには、`qopen` を実行します。

`qaborttrans`、`qclose`、`qchangeprio`、`qchangequeue`、`qchangetime`、`qchangeexptime`、`qcommittrans`、`qchange`、`qcreate`、`qdeletemsg`、`qinfo`、`qlist`、`qprinttrans`、および `qset` は、キュー・スペースがオープンしているときのみ実行できます。

次の表に、qmadmin コマンドを機能別に示します。

コマンド・タイプ	コマンド	目的
一般		
	<code>echo</code>	入力コマンド行をエコーします。
	<code>help</code>	ヘルプ・メッセージを表示します。
	<code>paginate</code>	ページ別に出力します。
	<code>quit</code>	セッションを終了します。
	<code>verbose</code>	冗長モードで出力を行います。
	<code>! shellcommand</code>	シェルにエスケープして、 <code>shellcommand</code> を実行します。
	<code>!!</code>	直前のシェル・コマンドを繰り返します。
	<code>#</code>	コメント行を指定します。
	<code><CR></code>	最後のコマンドを繰り返します。
キュー・スペース		

コマンド・タイプ	コマンド	目的
	chdl	汎用デバイス・リストのエントリ名を変更します。
	crdl	汎用デバイス・リストにエントリを作成します。
	dsdl	汎用デバイス・リストにあるエントリを破棄します。
	ipcrm	キュー・スペースに使用される IPC データ構造体を削除します。
	ipcs	キュー・スペースに使用される IPC データ構造体を一覧表示します。
	lidl	汎用デバイス・リストを表示します。
	livtoc	VTOC テーブルのすべてのエントリについて情報を表示します。
	qaddext	キュー・スペースの範囲を広げます。
	qclose	現在オープンしているキュー・スペースをクローズします。
	qopen	キュー・スペースの構造体をオープンして初期化します。
	qsize	キュー・スペースに必要な共用メモリのサイズを計算します。
	qspacechange	キュー・スペースのパラメータを変更します。
	qspacecreate	キューに入れられたメッセージ用のキュー・スペースを作成します。
	qspacedestroy	指定されたキュー・スペースを破棄します。
	qspacelist	キュー・スペースの作成パラメータを一覧表示します。
キュー		
	qchange	現在オープンされているキュー・スペースのキューを修正します。

セクション 1 - コマンド

コマンド・タイプ	コマンド	目的
	qcreate	現在オープンされているキュー・スペースにキューを作成します。
	qdestroy	名前付きのキューを破棄します。
	qinfo	関連付けられているキューまたはすべてのキューについて情報を一覧表示します。
メッセージ		
	qchangeexp	キューのメッセージの期限を変更します。
	qchangeprio	キューのメッセージの優先順位を変更します。
	qchangequeue	同じキュー・スペース内の別のキューにメッセージを移動します。
	qchangetime	キューのメッセージの実行時間を変更します。
	qdeletemsg	キューからメッセージを削除します。
	qlist	キューのメッセージを一覧表示します。
	qscan	ほかのコマンドで使用する選択基準を設定します。
	qset	ほかのコマンドで使用するキュー名を設定します。
トランザクション		
	qaborttrans	プリコミット・トランザクションをアボートします。
	qcommittrans	プリコミット・トランザクションをコミットします。
	qprinttrans	未処理のトランザクションのトランザクション・テーブル情報を表示します。

qadmin コマンド

コマンドの形式は、完全な名前でも省略形でも（省略形を使用できる場合。この後の一覧では、完全な名前の後の丸かっこ内に省略形を示しています）入力することができ、該当する引数はコマンドの後に付きます。角かっこ [] で囲まれている引数は省略可能です。中かっこ { } で囲まれている引数は相互に排他的な選択肢を示します。

`chdl [dlindex [newdevice]]`

汎用デバイス・リストのエントリ名を変更します。第 1 引数は汎用デバイス・リストの変更するデバイスのインデックスです (デバイス・インデックスは `lid1` で返されます)。第 1 引数をコマンド行で指定しなかった場合は、プログラムによって入力が必要されます。

第 2 引数は新しいデバイス名です。デバイス名がコマンド行に指定されていない場合、現在のデバイス名が表示され、新しいデバイス名の入力を求めるプロンプトが表示されます。デバイス名の長さは 64 文字以内に制限されています。デバイス名を変更すると、前のデバイス名を介してファイルやデータにアクセスすることはできなくなるので、このコマンドはよく注意して使用しなければなりません。

汎用デバイス・リスト (UDL) とボリューム一覧 (VTOC) の詳細については、『BEA Tuxedo アプリケーション実行時の管理』を参照してください。

`crdl [device [offset [size]]]`

汎用デバイス・リストにエントリを作成します。注意：デバイス・リストの先頭のエントリは、`QMCONFIG` が指すデバイスに対応していなければならない、またオフセット 0 を持たなければなりません。コマンド行で引数を指定しなかった場合には、プログラムはその引数の入力を要求します。

引数としては、デバイス名、空間の割り当てを開始できるブロック番号、および割り当てられる物理ページ数 (ディスク・セクタ) です。

指定したファイルに対して、複数の範囲で空間を割り当てることができます。たとえば、`/app/queues/myspace 0 500` と `/app/queues/myspace 1000 500` を割り当てることで、合計 1000 ブロックの割り当てを行い、500 から 999 までのブロックを未使用にすることができます。

先頭のデバイス・エントリで割り当てられるブロックのいくつかは、デバイス・リストおよび内容テーブルによって使用されます。最大 25 のエントリをデバイス・リストに作成できます。

`dSDL [-y] [dlindex]`

汎用デバイス・リストにあるエントリを破棄します。`dlindex` 引数は、汎用デバイス・リストから削除されるデバイスのデバイス・リストに

おけるインデックスです。これがコマンド行に指定されていない場合、その入力を求めるプロンプトが表示されます。ただし、エントリ 0 は、`vTOC` ファイルとその他のデバイス・リスト・エントリをすべて最初に除去した後でなければ除去できません (エントリ 0 には、デバイス・リストと目次が記録されていて、これを削除すると、これら 2 つのテーブルも破壊されてしまうためです)。`vTOC` ファイルは、関連するエントリを削除することによってのみ削除できます (たとえば、デバイスに存在するキュー・スペースを破壊することによって)。プログラムは、`-y` を指定していない限り、確認のためのプロンプトを表示します。

`echo` (e) [`off` | `on`]

`on` に設定すると、入力コマンド行をエコーします。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。初期設定は `off` です。

`help` (h) [`command` | `all`]

ヘルプ・メッセージを表示します。`command` を指定すると、そのコマンドについて、省略形、引数、および説明が表示されます。`all` を指定すると、すべてのコマンドに関する説明が表示されます。

引数がコマンド行に指定されていない場合、すべてのコマンドの構文が表示されます。

`ipcrm` [`-f`] [`-y`] [`queue_space_name`]

指定されたキュー・スペースに使用される IPC データ構造体を削除します。キュー・スペースの名前がコマンド行に指定されていない場合、その入力を求めるプロンプトが表示されます。指定したキュー・スペースが `qmadmin` でオープンされている場合は、クローズされません。`ipcrm` は、キュー・スペースが使用するすべての IPC リソースを認識しており、IPC リソースを削除する手段としては唯一の手段です。`qmadmin` では、IPC リソースが削除される前に、キュー・スペースにほかのプロセスが一切アタッチされていないことを保証します。`-f` オプションを指定すると、ほかのプロセスがアタッチされている IPC リソースでも強制的に削除することができます。`-f` オプションを指定している場合は、`-y` オプションが指定されていない限り、このコマンドは実行前に確認のためのプロンプトを表示します。コマンドが正常に完了すると、指定されたキュー・スペースのすべての一時的メッセージが完全に失われます。

`ipcs [queue_space_name]`

キュー・スペースに使用される IPC データ構造体があれば、それらの構造体を一覧表示します。キュー・スペースをオープンしているプロセスがなければ、構造体は一切使用されていません。キュー・スペースの名前がコマンド行に指定されていない場合、その入力を求めるプロンプトが表示されます。

`lidl [dlindex]`

汎用デバイス・リストを表示します。各デバイスごとに、インデックス、名前、先頭ブロック、およびデバイスのブロック数が一覧表示されます。冗長モードでは、未使用領域を示すマップが表示されます（未使用領域の先頭アドレスおよびサイズ）。`dlindex` を指定すると、デバイス・リストのそのエントリについての情報のみが表示されません。

`livtoc`

VTOC テーブルのすべてのエントリについて情報を表示します。各エントリについて表示される情報は、VTOC テーブルの名前、VTOC テーブルのあるデバイス、デバイスの先頭からの VTOC テーブルのオフセット、およびこのテーブルに割り当てられているページ数があります。VTOC には、最大 100 のエントリがあります。

`paginate (page) [{off|on}]`

出力をページ別表示します。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。標準入力または標準出力のどちらかが非端末デバイスでない限り、初期設定は on です。ページ別表示機能は、標準入力と標準出力のどちらも端末デバイスである場合のみ、on になります。

デフォルトのページ別表示コマンドは、ネイティブのオペレーティング・システム環境に固有のページャになります。たとえば、コマンド `pg` は、UNIX System オペレーティング環境におけるデフォルトのコマンドです。シェル環境変数 `PAGER` を使用すると、ページ別表示出力に使用されるデフォルトのコマンドを無効化することができます。

`qaborttrans (qabort) [-y] [tranindex]`

指定したトランザクション・インデックス `tranindex` に関連付けられている、プリコミットされているトランザクションをヒューリスティックにアボートします。トランザクション・インデックスをコマンド行で指定しなかった場合は、プログラムはその入力を要求しま

す。トランザクションが決定されることがわかっていて、その決定がコミットすることであった場合は、`qaborttrans` は失敗します。インデックスは、事前の `qprinttrans` コマンドの実行から取得されます。`-y` オプションが指定されていなければ、確認が要求されます。このコマンドの使用には、十分な注意が必要です。

`qaddext [queue_space_name [pages]]`

キュー・スペースにエクステントを追加します。キュー・スペースは、アクティブであってはなりません (キュー・スペースにはどんなプロセスもアタッチされてはなりません)。キュー・スペース名、およびキュー・スペースに割り当てる追加物理ページ数を、コマンド行で指定しなかった場合は、プログラムは、その入力を要求します。指定したキュー・スペースが `qmadmin` でオープンされている場合は、クローズされます。物理ページ数は 4 の倍数に最も近い値に切り下げられます。この説明と例については、`qspacecreate` を参照してください。スペースは、UDL で `QMCONFIG` デバイスに関連付けられて定義されているエクステントから割り当てられます。新しいキュー・スペースエクステントは、`VTOC` においてそれぞれ追加エントリを使用します (最大 100 のエントリが使用可能です)。キュー・マネージャは、このエクステントに名前を指定して、このエクステントの迅速な識別とキュー・スペースへの関連付けができるようにします。コマンドが正常に完了すると、指定されたキュー・スペースのすべての一時的メッセージが完全に失われます。

`qchange [queue_name [out-of-order [retries [delay [high [low [cmd]]]]]]`
`[-d persist|nonpersist] [-n nhigh,nlow,ncmd]`
`[-e default_relative_expiration_time]`

現在オープンしているキュー・スペースで、キューを修正します。引数はコマンド行で指定します。指定しないと、プログラムにより入力が必要になります。引数には、キューの名前、順序を無視したキュー登録を許可するかどうか (許可しない、キューの先頭に登録、指定した `msgid` の前に登録)、再試行の回数、各再試行の間の遅延時間 (秒)、しきい値コマンドを実行するための上限値と下限値、および永続的メッセージに対するしきい値コマンドそのものを指定できます。

`out-of-order` の値は、`none`、`top`、または `msgid` です。`top` と `msgid` の両方を、カンマで区切って指定することができます。

しきい値を使用すると、永続的メッセージについてしきい値に達したときに、コマンドが自動的に実行されます。上限は、コマンドを実行

する時期を指定します。上限に達した場合は、コマンドを再度実行する前に下限に戻しておく必要があります。たとえば、上限と下限がそれぞれ 100 メッセージ、50 メッセージの場合、キュー内のメッセージが 100 になるとコマンドが実行されます。コマンドはキュー中のメッセージが 50 に減り、再度 100 になるまで実行されません。キューの容量の指定は、キューが使用するバイト数またはブロック数（接尾辞 "b" または "B" を後に付けた数字）、キューが使用するキュー・スペースに対する割合（"%" を後に付けた数字）、またはキューの全メッセージ数（"m" を後に付けた数字）を使用して行えます。上下のしきい値のタイプは、同じでなければなりません。下限しきい値にタイプを指定するかどうかは省略可能ですが、指定する場合は、上限しきい値のタイプと一致させなければなりません。メッセージ (m) 接尾辞は、永続的メッセージと一時的メッセージの両方に適用されます。その他のしきい値の接尾辞は、永続的メッセージにのみ適用されます。-n オプションを使用して、一時的メッセージのしきい値を指定します。コマンド行で指定する場合、空白文字を含むしきい値コマンドは二重引用符で囲みます。再試行回数 (retry) は、メッセージをキューから取り出してトランザクションをロールバックし、再びキューにメッセージを戻す、という処理を実行できる最大回数を示します。再試行までの遅延時間も指定できます。再試行回数に達すると、メッセージはキュー領域に定義されるエラー・キューに移動されます。エラー・キューが定義されていない場合、メッセージは削除されます。キューに対するキュー順序の値は変更できません。優先順位の低いメッセージは、他に優先順位の高いメッセージがある場合でも、10 のメッセージがたまるごとにキューから取り出されます。

-d オプションは、キューのデフォルトの配信方針を指定します。-d オプションに有効な値は、`persist` と `nonpersist` です。デフォルトの配信方針が `persist` の場合、明示的に配信モードが指定されずにキューに登録されているメッセージは、永続的（ディスク・ベース）配信方式で配信されます。配信方針が `nonpersist` の場合は、明示的に配信モードが指定されずにキューに登録されているメッセージは、一時的（メモリ内）配信方式で配信されます。-d オプションが指定されない場合、情報の入力を求めるプロンプトは表示されず、デフォルトの配信方針は変更されません。デフォルトの配信方針が変更されても、既にキューに入っているメッセージについては、サービスの配信基準は変更されません。現在キュー・スペース内にあるメッセージに対する応答キューを変更する場合、キューのデフォルトの配信方針の変更によって、サービスの応答基準が変更されることはありません。

メモリ領域の不足やフラグメント化のために一時的メッセージをキューに登録できない場合、メッセージ用の永続ストレージが十分にあってキューへの登録操作は失敗します。ディスクの不足やフラグメント化のために永続的メッセージをキューに登録できない場合、メッセージ用の非永続ストレージが十分にあってキューへの登録操作は失敗します。

キュー・スペースの中の一時的メッセージ用に確保するメモリ量をゼロ (0) にすると、一時的メッセージ用の領域は確保されません。一時的メッセージ用のメモリ領域の指定の詳細については、`qspacecreate` と `qspacechange` を参照してください。この場合、一時的メッセージをキューに登録しようとしても失敗します。これは、サービスの配信基準が指定されていないメッセージを、デフォルトの配信方針が `nonpersist` に設定されたキューに登録するときにも当てはまります。

`-n` オプションは、コマンドを自動実行する際のしきい値を指定します。非永続ストレージ領域の値がしきい値に達すると、コマンドが自動的に実行されます。`nhigh` は、コマンド `ncmd` を実行するタイミングを指定します。`nhigh` が上限に達したときは、コマンドを再度実行する前に `nlow` を下限に戻しておく必要があります。`-n` オプションを指定する場合は、`nhigh`、`nlow`、および `ncmd` を共に指定する必要があります。これらを指定していないと、コマンドは失敗します。`ncmd` の値は、空文字列として指定することもできます。`-n` オプションを指定していなくても、情報の入力を求めるプロンプトは表示されません。

メモリ容量 (キュー内の非永続的データの量) のしきい値は、バイト (b)、ブロック (B)、またはパーセンテージ (数字の後に % を付ける) のいずれかのタイプで指定できます。`nhigh` と `nlow` のしきい値は、同じタイプにする必要があります。たとえば、`nhigh` が 100% に設定されている場合、`nlow` もパーセンテージで指定しなければなりません。`nlow` のしきい値のタイプは省略可能です。`-n` オプションが指定されていない場合、一時的メッセージ用のデフォルトのしきい値は変更されません。`ncmd` に空白を使用する場合は、空白を二重引用符で囲む必要があります。

しきい値 `[...[high[low[cmd]]]...]` の接尾辞 `m` は、永続的メッセージと一時的メッセージを含むキュー内にあるすべてのメッセージに適用されるため、`nhigh` と `nlow` は使用できません。接尾辞 `-m` が指定され

ていないしきい値 [...[*high*[*low*[*cmd*]]...] は、永続的 (ディスク・ベース) メッセージにのみ適用されます。

`-e default_relative_expiration_time` オプションは、キューに登録されるメッセージのうち、期限が明示的に指定されていないメッセージに対して期限を設定します。期限は、相対時間にすることも `none` にすることもできます。期限に達したとき、メッセージがキューから取り出されていない場合や管理時に削除されていない場合には、キューからメッセージが削除され、メッセージに関連付けられているすべてのリソースが解放されて、統計情報が更新されます。有効期限がメッセージの使用可能時間より前の場合、いずれかの時間を変更して使用可能時間が有効期限より前になるようにしない限り、キューからメッセージを取り出すことはできません。また、これらのメッセージがキューからの取り出しの対象になったことがなくても、有効期限が切れるとキューから削除されます。トランザクション中にメッセージの期限が切れてもトランザクションは失敗しません。トランザクション内でキューへの登録、またはキューからの取り出し中に有効期限が切れたメッセージは、トランザクションが終了した時点でキューから削除されます。メッセージの期限が切れても通知されません。

`-e` オプションが指定されない場合、キューのデフォルトの有効期限は変更されません。キューの期限を `qchange` を使用して変更する場合、既にキューに入っているメッセージは変更の対象外になります。`-e` オプションが指定されていなくても、この入力を求めるプロンプトは表示されません。

相対時間の `default_relative_expiration_time` の形式は `+seconds` です。`seconds` は、操作をキュー・マネージャが正常に処理してから有効期限に達するまでの経過秒数です。ゼロ (0) を設定すると、すぐに期限に達します。`default_relative_expiration_time` の値には、文字列 `none` を設定することもできます。文字列 `none` を指定した場合、キューへの登録時に明示的に期限が指定されていないメッセージは、明示的に期限を割り当てない限り、期限切れになりません。

`qchangeexp (qce) -y [newtime]`

キューにあるメッセージの有効期限を変更します。期限に達したとき、メッセージがキューから取り出されていない場合や管理時に削除されていない場合には、キューからメッセージが削除され、メッセージに関連付けられているすべてのリソースが解放されて、統計情報が更新されます。有効期限の時間がメッセージの使用可能時間より前の

場合、使用可能時間が有効期限の切れる時間より前になるようにいずれかの時間を変更しない限り、メッセージをキューから取り出すことはできません。また、これらのメッセージがキューからの取り出しの対象になったことがなくても、有効期限が切れるとキューから削除されます。トランザクション中にメッセージの期限が切れてもトランザクションは失敗しません。トランザクション内でキューへの登録、またはキューからの取り出し中に有効期限が切れたメッセージは、トランザクションが終了した時点でキューから削除されます。メッセージの期限が切れても通知されません。

有効期限を設定するキューは、`qset` コマンドを使って選択します。更新するメッセージを制限する選択基準は、`qscan` コマンドを使って設定します。選択基準が何も設定されていない場合、キューにあるすべてのメッセージが変更されます。デフォルトでは、期限を設定する前に確認が要求されます。`-y` オプションを指定すると確認は行われません。`newtime` の値には、現在時刻からの相対時間、絶対時間、または `none` を指定できます。`newtime` の値がコマンド行に指定されていない場合、その入力を求めるプロンプトが表示されます。

キューに登録されるメッセージのうち、メッセージ期限をサポートしていないバージョンの BEA Tuxedo ATMI システムによって登録されたものについては、値の変更を行うキュー・マネージャがメッセージ期限をサポートしていても、期限を設定できるように変更することはできません。このような BEA Tuxedo ATMI システムのバージョンによってキューに登録されたメッセージが `qchangeexp` の対象に含まれている場合、選択されたメッセージのうちのいくつかがこの制限により変更されなかったことを知らせるエラー・メッセージが表示されます。

相対時間による期限は、キュー・マネージャのプロセスで要求を受け取った時点からの相対時間です。相対時間の `newtime` の形式は `+seconds` です。`seconds` は、操作をキュー・マネージャが正常に処理してから有効期限に達するまでの経過秒数です。`seconds` がゼロ (0) に設定されている場合、メッセージはすぐに期限切れになります。絶対時間による期限は、キュー・マネージャ・プロセスが存在するマシンの時間によって決まります。絶対時間の `newtime` の形式は `YY[MM[DD[HH[MM[SS]]]]]` です。これについては、`qscan` の項で説明します。`newtime` の値には、文字列 `none` を設定することもできます。この場合、対象となるメッセージが期限切れになることはありません。

`qchangeprio (qcp) [-y] [newpriority]`

キューのメッセージの優先順位を変更できます。対象となるキューは、`qset` コマンドを使用して設定され、更新されるメッセージを制限する選択基準は、`qscan` コマンドを使用して設定されます。

選択基準が何も設定されていない場合、キューにあるすべてのメッセージが変更されます。`-y` オプションを指定していない限り、変更が行われる前に確認を求められます。`qlist` コマンドを実行して、どのメッセージが変更されるのかを調べることをお勧めします（入力ミスを少なくするためです）。`newpriority` の値は、メッセージが処理のために転送される際に使用される、新しい優先順位を指定します。優先順位の範囲は、1 以上 100 以下の値でなければなりません。コマンド行で指定していないと、入力するよう求められます。

`qchangequeue (qcq) [-y] [newqueue]`

同じキュー・スペース内の別のキューにメッセージを移動できます。メッセージの移動元のキューは、`qset` コマンドを使用して設定され、移動するメッセージを制限する選択基準は、`qscan` コマンドを使用して設定されます。選択基準が何も設定されていない場合、キューにあるすべてのメッセージが移動されます。`-y` オプションを指定していない限り、変更が行われる前に確認を求められます。`qlist` コマンドを実行して、どのメッセージが移動されるのかを調べることをお勧めします（入力ミスを少なくするためです）。`newqueue` の値は、メッセージの移動先となる新しいキュー名を指定します。`newqueue` がコマンド行に指定されていない場合、その入力を求めるプロンプトが表示されます。メッセージのサービス配信基準が、`newqueue` のデフォルトの配信方針と一致するように変更されることはありません。

期限付きのメッセージが移動された場合、移動元では相対時間による期限が指定されていたとしても、移動先のキューでの絶対時間による期限になります。

`qchangetime (qct) [-y] [newtime]`

キューにあるメッセージの使用可能時間を変更できます。キューの指定には、`qset` コマンドを使用します。更新するメッセージを制限する選択基準は、`qscan` コマンドを使用して設定します。

選択基準が何も設定されていない場合、キューにあるすべてのメッセージが変更されます。`-y` オプションを指定していない限り、変更が行われる前に確認を求められます。`qlist` コマンドを実行して、どの

メッセージが変更されるのかを調べることをお勧めします (入力ミス
を少なくするためです)。*newtime* の値は、現在時刻からの相対時間
にすることも、絶対時間にすることもできます。コマンド行で指定し
ていないと、入力するよう求められます。相対時間の *onetime* の形式
は、*+seconds* です。*seconds* は現在からメッセージが実行されるまで
の秒数です (0 の場合は即時実行を意味します)。絶対時間の *newtime*
の形式は *YY[MM[DD[HH[MM[SS]]]]]* です。これについては、*qscan* の項で
説明します。

qclose

現在オープンしているキュー・スペースをクローズします。コマンド
が正常に完了すると、指定されたキュー・スペースのすべての一時的
メッセージが完全に失われます。

qcommittrans (qcommit) [-y] [tranindex]

指定したトランザクション・インデックス *tranindex* に関連付けられ
ているプリコミットされているトランザクションをヒューリスティック
にコミットします。トランザクション・インデックスをコマンド行
で指定しなかった場合は、プログラムはその入力を要求します。トラン
ザクションが決定されることがわかっている、その決定がアポート
することであった場合は、*qcommittrans* は失敗します。インデック
スは、事前の *qprinttrans* コマンドの実行から取得されます。*-y* オプ
ションが指定されていなければ、確認が要求されます。このコマンド
の使用には、十分な注意が必要です。

qcreate (qcr) [queue_name [qorder [out-of-order [retries [delay [high [low [cmd]]]]]]]] [-d persist|nonpersist] [-n nhigh,nlow,ncmd] [-e default_relative_expiration_time]

現在オープンされているキュー・スペースにキューを作成します。引
数はコマンド行で指定します。指定しないと、プログラムにより入力
が求められます。引数には、キューの名前、キューの順序 (FIFO か
LIFO、期限、優先順位、時間)、順序を無視したキュー登録を許可す
るかどうか (許可しない、キューの先頭に登録、指定した *msgid* の前
に登録)、再試行の回数、各再試行の間の遅延時間 (秒)、しきい値コ
マンドを実行するための上限値と下限値、および永続的メッセージに
対するしきい値コマンドそのものを指定できます。

キューの順序 (*qorder*) の値は、*fifo*、*lifo*、*priority*、*expiration*、
および *time* です。キューの順序を指定する際には、最も重要なソー
ト値を最初に、次に重要なソート値をその後、というようにして指

定します。fifo や lifo は、最も重要度の低い(または唯一の)ソート値としてのみ指定できます。fifo も lifo も指定しない場合のデフォルトの設定は fifo になり、これは、ほかのソート基準の指定範囲内で適用されます。expiration が指定されている場合、期限が指定されていないメッセージは、期限付きメッセージがすべてキューから取り出された後に取り出されます。複数のソート値を、カンマで区切って指定することができます。out-of-order の値は、none、top、または msgid です。top と msgid の両方を、カンマで区切って指定することができます。

しきい値を使用すると、永続的メッセージについてしきい値に達したときに、コマンドを自動的に実行できます。上限は、コマンドを実行する時期を指定します。上限に達した場合は、コマンドを再度実行する前に下限に戻しておく必要があります。たとえば、この制限値が 100 メッセージと 50 メッセージである場合は、キューのメッセージ数が 100 になったときにコマンドが実行されます。そして、そのキューが排出されてメッセージ数が 50 を下回り、メッセージが再度増えて 100 に達するまではコマンドは再実行されません。

キューの容量の指定は、キューが使用するバイト数またはブロック数(接尾辞 "b" または "B" を後に付けた数字)、キューが使用するキュー・スペースに対する割合("%" を後に付けた数字)、またはキューの全メッセージ数("m" を後に付けた数字)を使用して行えます。上下のしきい値のタイプは、同じでなければなりません。メッセージ(m)接尾辞は、永続的メッセージと一時的メッセージの両方に適用されます。その他のしきい値の接尾辞は、永続的メッセージのみ適用されます。-n オプションを使用して、一時的メッセージのしきい値を指定します。下限しきい値にタイプを指定するかどうかは省略可能ですが、指定する場合は、上限しきい値のタイプと一致させなければなりません。コマンド行で指定する場合、空白文字を含むしきい値コマンドは二重引用符で囲みます。

再試行回数(retry)は、メッセージをキューから取り出してトランザクションをロールバックし、再びキューにメッセージを戻す、という処理を実行できる最大回数を示します。再試行までの遅延時間も指定できます。再試行回数に達すると、メッセージはキュー領域に定義されるエラー・キューに移動されます。エラー・キューが定義されていない場合、メッセージは削除されます。優先順位の低いメッセージは、

他に優先順位の高いメッセージがある場合でも、10 のメッセージがたまるごとにキューから取り出されます。

`-d` オプションは、キューのデフォルトの配信方針を指定します。`-d` オプションに有効な値は、`persist` と `nonpersist` です。デフォルトの配信方針が `persist` の場合、明示的に配信モードが指定されずにキューに登録されているメッセージは、永続的 (ディスク・ベース) 配信方式で配信されます。配信方針が `nonpersist` の場合は、明示的に配信モードが指定されずにキューに登録されているメッセージは、一時的 (メモリ内) 配信方式で配信されます。`-d` オプションが指定されない場合には、情報の入力を求めるプロンプトは表示されず、デフォルトの配信方針は `persist` です。デフォルトの配信方針が変更されても、既にキューに入っているメッセージについては、サービスの配信基準は変更されません。

メモリ領域の不足やフラグメント化のために一時的メッセージをキューに登録できない場合、メッセージ用の永続ストレージが十分にあってキューへの登録操作は失敗します。ディスクの不足やフラグメント化のために永続的メッセージをキューに登録できない場合、メッセージ用の非永続ストレージが十分にあってキューへの登録操作は失敗します。

キュー・スペースの中の一時的メッセージ用に確保するメモリ量をゼロ (0) にすると、一時的メッセージ用の領域は確保されません。一時的メッセージ用のメモリ領域の指定の詳細については、`qspacecreate` と `qspacechange` を参照してください。この場合、一時的メッセージをキューに登録しようとしても失敗します。これは、サービスの配信基準が指定されていないメッセージを、デフォルトの配信方針が `nonpersist` に設定されたキューに登録するときにも当てはまります。

`-n` オプションは、コマンドを自動実行する際のしきい値を指定します。非永続ストレージ領域の値がしきい値に達すると、コマンドが自動的に実行されます。`nhigh` は、コマンド `ncmd` を実行するタイミングを指定します。`nhigh` が上限に達したときは、コマンドを再度実行する前に `nlow` を下限に戻しておく必要があります。`-n` オプションを指定する場合は、`nhigh`、`nlow`、および `ncmd` を共に指定する必要があります。これらを指定していないと、コマンドは失敗します。`ncmd` の値は、空文字列として指定することもできます。`-n` オプションを指定していなくても、情報の入力を求めるプロンプトは表示されません。

メモリ容量 (キュー内の非永続的データの量) のしきい値は、バイト (b)、ブロック (B)、またはパーセンテージ (数字の後に % を付ける) のいずれかのタイプで指定できます。 *nhigh* と *nlow* のしきい値は、同じタイプにする必要があります。たとえば、*nhigh* が 100% に設定されている場合、*nlow* もパーセンテージで指定しなければなりません。*nlow* のしきい値のタイプは省略可能です。 *-n* オプションが指定されていない場合、*nhigh* には 100%、*nlow* には 0% がそれぞれデフォルト値として使用され、*ncmd* は " " に設定されます。*ncmd* に空白を使用する場合は、空白を二重引用符で囲む必要があります。

しきい値 [... [*high*[*low*[*cmd*]]] ...] の接尾辞 *m* は、永続的メッセージと一時的メッセージを含むキュー内にあるすべてのメッセージに適用されるため、*nhigh* と *nlow* は使用できません。接尾辞 *-m* が指定されていないしきい値 [... [*high*[*low*[*cmd*]]] ...] は、永続的 (ディスク・ベース) メッセージにのみ適用されます。

-e default_relative_expiration_time オプションは、キューに登録されるメッセージのうち、期限が明示的に指定されていないメッセージに対して期限を設定します。期限は、相対時間にするのも *none* にすることもできます。期限に達したとき、メッセージがキューから取り出されていない場合や管理時に削除されていない場合には、キューからメッセージが削除され、メッセージに関連付けられているすべてのリソースが解放されて、統計情報が更新されます。有効期限がメッセージの使用可能時間より前の場合、いずれかの時間を変更して使用可能時間が有効期限より前になるようにしない限り、キューからメッセージを取り出すことはできません。また、これらのメッセージがキューからの取り出しの対象になったことがなくても、有効期限が切れるとキューから削除されます。トランザクション中にメッセージの期限が切れてもトランザクションは失敗しません。トランザクション内でキューへの登録、またはキューからの取り出し中に有効期限が切れたメッセージは、トランザクションが終了した時点でキューから削除されます。メッセージの期限が切れても通知されません。

-e オプションが指定されない場合、キューのデフォルトの期限は *none* に設定されます。キューの期限を *qchange* を使用して変更する場合、既にキューに入っているメッセージは変更の対象外になります。*-e* オプションが指定されていなくても、この入力を求めるプロンプトは表示されません。

相対時間の `default_relative_expiration_time` の形式は `+seconds` です。 `seconds` は、操作をキュー・マネージャが正常に処理してからの有効期限に達するまでの経過秒数です。ゼロ (0) を設定すると、すぐに期限に達します。 `default_relative_expiration_time` の値には、文字列 `none` を設定することもできます。文字列 `none` を指定した場合、キューへの登録時に明示的に期限が指定されていないメッセージは、明示的に期限を割り当てない限り、期限切れになりません。

`qdeletemsg (qdltm) [-y]`

キューからメッセージを削除できます。キューの指定には、`qset` コマンドを使用します。削除するメッセージを制限する選択基準は、`qscan` コマンドを使って設定します。選択基準が何も設定されていない場合、キューにあるすべてのメッセージが削除されます。削除が行われる前に、確認が要求されます。 `qlist` コマンドを実行して、どのメッセージが削除されるのかを調べることをお勧めします (入力ミスを少なくするためです)。 `-y` オプションが指定されていない限り、このコマンドは確認のためのプロンプトを表示します。

`qdestroy (qds) [{ -p | -f }] [-y] [queue_name]`

指定されたキューを破棄します。デフォルトでは、キューに要求が存在していたり、キュー・スペースにプロセスがアタッチされていたりすると、エラーが返されます。 `-p` オプションを指定すると、キュー・スペースにアタッチされているプロセスがなければ、キューからあらゆるメッセージを「パージ」して破棄することができます。 `-f` オプションを指定すると、メッセージまたはプロセスがキュー・スペースにアタッチされている場合でも、「強制的に」キューを削除できます。このとき、メッセージがその時点でトランザクションに関連していた場合は、このコマンドは異常終了し、`userlog` にエラーが書き込まれます。このコマンドは、 `-y` オプションを指定していない場合に確認のためのプロンプトを表示します。

`qinfo [queue_name]`

関連付けられているキューまたはすべてのキューについて情報を一覧表示します。このコマンドは、指定されたキューのメッセージ数、または引数を指定していない場合はすべてのキューのメッセージ数、キューに関連付けられている永続的メッセージと一時的メッセージの両方が使用している領域の量、永続的および一時的に送られるメッセージの数、指定されたキューのメッセージの総数、および永続的メッセージと一時的メッセージが使用している領域の量を表示します。冗長モードでは、このコマンドは、各キューのキュー作成パラ

メータ、キューのデフォルトの期限（設定されている場合）、順序付け基準、およびキューのデフォルトの配信方針も表示します。

`qlist (ql)`

キューのメッセージを一覧表示します。キューの指定には、`qset` コマンドを使用します。表示するメッセージを制限する選択基準は、`qscan` コマンドを使って設定します。設定基準が何も設定されていない場合は、キューにあるすべてのメッセージが表示されます。

選択した各メッセージについて、メッセージ識別子が出力されるほか、メッセージの優先順位、これまでに試みた再試行の回数、メッセージの長さ、サービスのデリバリ基準、応答に対するサービスの基準、および期限（設定されている場合）が出力されます。メッセージに可用時間が関連付けられている場合、またはトランザクションのロールバックのためにメッセージに再試行時間がスケジューリングされている場合は、その可用時間が出力されます。関連識別子が存在し、`verbose` モードがオンの場合は、関連識別子が表示されます。

`qopen [queue_space_name]`

指定したキュー・スペースの内部構造体をオープンし、初期化します。コマンド行でキュー・スペースを指定しなかった場合は、プログラムがその入力を要求します。キュー・スペースが既に `qadmin` でオープンされている場合は、クローズされます。

`qprinttrans (qpt)`

現在未終了のトランザクションのトランザクション・テーブル情報を表示します。トランザクション識別子、インデックス、トランザクション・ステータスが表示されます。インデックスは、`qaborttrans` や `qcommittrans` でトランザクションのアボートまたはコミットを行う際に使用されるものです。

`qscan [{ [-t time1[-time2]] [-p priority1[-priority2]] [-m msgid] [-i corrid][[-d delivery_mode] [-e time1[-time2]] | none]}`

コマンド `qchangeprio`、`qchangequeue`、`qchangetime`、`qdeletemsg`、および `qlist` に使用する選択基準を設定します。引数 `none` は、選択基準がないことを示します。この場合は、すべてのメッセージが対象になります。引数なしでこのコマンドを実行すると、現在の選択基準値が表示されます。コマンド行オプションで値の範囲を指定する場合（`-t`、`-e`、`-p` など）、値の範囲に空白を入れることはできません。`-t` オプションは、時間の値または時間の範囲を示すために使用できます。

time1 と *time2* の形式は YY[MM[DD[HH[MM[SS]]]]] です。それぞれ、年、月、日、分、秒を指定します。この日時の値から省略した単位は、デフォルト値として、可能な範囲内の最小値が使用されます。たとえば、&d“02&d”すると &dlq;7“000000&drq;”ます。年の値を 00 ~ 37 とすると 2000 ~ 2037 と見なされ、70 ~ 99 とすると 1970 ~ 1999 と見なされ 38 ~ 69 とすると無効になります。-p オプションは、優先順位の値、または優先順位の範囲を示すために使用できます。優先順位の値は、1 以上 100 以下の範囲になります。-m オプションは、メッセージ識別子の値を示すために使用できます。メッセージ識別子の値は、キューにメッセージが登録されるときにシステムがメッセージに割り当てた値です。メッセージ識別子は、1 つのキュー内で一意の値を持ち、この値の長さは 32 文字以内です。32 文字未満の値は、右側に NULL (0x0) が埋められます。バックスラッシュおよび表示不可能な文字 (スペース、改行、タブなどの空白文字) を入力するときは、バックスラッシュの後にその文字を表す 2 桁の 16 進値を付けた値を入力しなければなりません。たとえば、スペースは \20 なので &dlq;o\20world”と入力します。-i オプションは、メッセージに関連付けられている関連識別子の値を示すために使用できます。この識別子の値は、アプリケーションによって割り当てられ、キューに登録されたメッセージと一緒に格納され、応答メッセージまたはエラー・メッセージの返信と一緒に格納するために渡されて、アプリケーションが特定の要求に対する応答を識別できるようにします。この値の長さは 32 文字以内です。32 文字未満の値は、右側に NULL (0x0) が埋められます。バックスラッシュおよび表示不可能な文字 (スペース、改行、タブなどの空白文字も含む) を入力するときは、バックスラッシュの後にその文字を表す 2 桁の 16 進値を付けた値を入力しなければなりません (たとえば、スペースは \20 なので、"my\20ID\20value" のようにします)。

-d *delivery_mode* オプションの有効な値は、*persist* と *nonpersist* です。このオプションは、*qscan* で選択されるメッセージの配信モードを指定します。これにより、オペレータは配信方式に基づいた操作ができます。

-e オプションは、期限または期限の範囲を示すために使用できます。*time1* と *time2* の形式は、-t オプションの *time1* と *time2* の形式と同じです。

qset [queue_name]

コマンド qchangeprio、qchangequeue、qchangetime、qdeletemsg、および qlist に使用するキュー名を設定します。引数なしでこのコマンドを実行すると、現在のキュー名が表示されます。

qsize [pages [queues [transactions [processes [messages]]]]]
 [-A actions] [-H handles] [-C cursors] [-O owners] [-Q tmp_queues]
 [-f filter_memory] [-n nonpersistent_msg_memory[b,B]]
 [-o overflow_memory]

pages、queues、(同時) transactions、processes、および(キュー内) messages で指定したサイズを使用して、キュー・スペースに必要な共有メモリのサイズを計算します。コマンド行で値を指定しなかった場合は、プログラムはそれらの入力を要求します。必要となるシステム・セマフォの数も出力されます。上記以外のオプションで有効な値については、qspacecreate で説明しています。

qspacechange (qspch) [queue_space_name [ipckey [trans [procs [messages [errorq [inityn [blocking]]]]]]]] [-A actions] [-H handles]
 [-C cursors] [-O owners] [-Q tmp_queues] [-f filter_memory]
 [-n nonpersistent_msg_memory[b,B]] [-o overflow_memory]

キュー・スペースのためのパラメータを変更します。キュー・スペースはアクティブであってはなりません。つまり、キュー・スペースにはどんなプロセスもアタッチされてはなりません。必要な情報がコマンド行に指定されていない場合、その入力を求めるプロンプトが表示されます。有効な値については、このページの qspacecreate で説明しています。指定したキュー・スペースが qmadmin でオープンされている場合はクローズされます。新しいエクステントを追加する場合には、qaddext を使用します。キューの数は変更できません。

qspacecreate (qspc) [queue_space_name [ipckey [pages [queues [trans [procs [messages [errorq [inityn [blocking]]]]]]]]]] [-A actions]
 [-n nonpersistent_msg_memory[b,B]] [-o overflow_memory]

キュー・メッセージ用のキュー・スペースを作成します。情報がコマンド行に指定されていない場合、その入力を求めるプロンプトが表示されます。その情報には、キュー・スペース名、共用メモリ・セグメントとセマフォの ipckey、このキュー・スペースに割り当てる物理ページ数、キューの数、同時実行可能なトランザクションの数、このキュー・スペースに同時にアタッチできるプロセスの数、一度にキューに登録できるメッセージの数、このキュー・スペースのエラー・キュー名、このキュー・スペースの新しいエクステントでペー

ジを初期化するかどうか、およびキュー・スペースの初期化やウォーム・スタート・ディスクの入出力を行なうためのブロッキング要因があります。

物理ページ数は切り下げられ、4の倍数に最も近い値のページ数になります。たとえば、必要なページ数が50の場合は48ページ分のメモリが割り当てが行なわれ、必要なページ数が52の場合は52ページ分のメモリ割り当てが行なわれます。エラー・キューは、最大再試行回数に達したメッセージを保持するために使用されます。メッセージは、初めに登録されていたキューからエラー・キューに移動されず、このキューが確実に排出されることについての責任は、管理者にあります。

割り当てられた物理ページ数は、キュー・スペースのオーバーヘッドを保持できるだけの十分な大きさでなければなりません（各キューにつき1ページのほかもう1ページ）。初期化のオプションとして'y'または&sl'srq;が指定された場合は、キュー・スペースを保持するために使用する領域が初期化され、このコマンドの実行にしばらく時間がかかります。冗長モードでは、キュー・スペースの初期化が5%完了するごとに、標準出力にピリオド(.)が出力されます。初期化のオプションをオンにしていないが対象デバイスが文字型特殊デバイスでない場合には、エクステントとして指定したサイズに達していなければ、ファイルは初期化されます。つまり、指定された領域を割り当てることができるようにファイルが拡大されます。

キュー・スペースの作成中およびウォーム・スタート（キュー・スペースの再起動）の際にブロックを読み書きする場合、入出力操作のサイズは、ブロッキング化係数によって指定されたディスク・ページ・サイズの倍数で行なわれます。

-A *actions* オプションは、キューイング・サービス・コンポーネントで同時処理できる追加操作の数を指定します。ブロッキング操作の発生時に追加操作を利用できる場合、ブロッキング操作は条件を満たす状態になるまで保留されるように設定されます。ブロッキング操作が保留されると、ほかの操作要求を処理できます。ブロッキング操作が完了すると、その操作に関連する操作は続く操作でも実行できるようになります。ブロッキング操作が要求された時点ですぐに条件を満たす状態にならず、利用可能な操作もない場合には、操作は失敗します。システムでは、キュー・スペースにアタッチ可能なプロセスの数と同じ数だけ操作が予約されているため、それぞれのキュー・マネー

ジャ・プロセスは少なくとも1つブロッキング操作を所有できます。システムによって予約されているブロッキング操作の数を超える場合、管理者は予約数より多くの追加ブロッキング操作に対応できるようにシステムを設定することができます。-A *actions* オプションが指定されていない場合、デフォルト値はゼロに設定されます。-A オプションが指定されていなくても、その入力を求めるプロンプトは表示されません。

-n *nonpersistent_msg_memory* オプションは、キュー・スペースのすべてのキューに対して、一時的メッセージ用に共有メモリに確保する領域のサイズを指定します。サイズはバイト単位 (b) またはブロック単位 (B) で指定できます。ブロック・サイズはディスク・ブロック・サイズと同じです。[bB] 接尾辞はオプションであり、指定されていない場合、デフォルト値はブロックに設定されます。-n オプションが指定されていない場合、メモリ・サイズのデフォルト値はゼロ (0) に設定されます。-n オプションが指定されていなくても、その入力を求めるプロンプトは表示されません。

nonpersistent_msg_memory にバイト単位 (b) で値が指定されている場合、この値はページあたりのバイト数 (ページ・サイズはディスク・ページ・サイズと同じ) で割って最も近い整数に切り下げられ、その数だけのページが割り当てられます。たとえば、ページ・サイズを1024 バイト (1KB) として考えると、要求された値が2000b の場合は1ページ分 (1024 バイト) のメモリ割り当てが行われ、要求された値が2048b の場合は2ページ分 (2048 バイト) のメモリ割り当てが行われます。ページあたりのバイト数より少ない値を要求すると、0ページ分 (0 バイト) が割り当てられます。

nonpersistent_msg_memory にブロック単位 (B) で値が指定されている場合、1メモリ・ブロックが1メモリ・ページと同じであるとすると、ページ数と同じ数だけの割り当てが行われます。たとえば、要求された値が50B の場合、50ページ分のメモリ割り当てが行われます。

キュー・スペースの *nonpersistent_msg_memory* がゼロ (0) の場合、一時的メッセージ用の空間は確保されません。この場合、一時的メッセージをキューに登録しようとしても失敗します。永続ストレージと非永続ストレージとの互換性はありません。メモリ領域の不足やフラグメント化のために一時的メッセージをキューに登録できない場合、メッセージ用の永続ストレージが十分にあっていてもキューへの登録操作

は失敗します。ディスクの不足やフラグメント化のために永続的メッセージをキューに登録できない場合、メッセージ用の非永続ストレージが十分にあってキューへの登録操作は失敗します。

`-o overflow_memory` オプションは、割り当て済みの共用メモリ・リソースの一部またはすべてが不足する負荷のピーク状況に対応するために、共用メモリに確保するメモリ領域のサイズを指定します。メモリ・サイズはバイト単位で指定します。追加したオブジェクトは、この追加メモリから先着順で割り当てられます。追加メモリで作成されたオブジェクトを閉じるか破棄すると、次に共用メモリ・リソース不足が発生するときに備えてメモリは解放されます。 `-o overflow_memory` オプションが指定されていない場合、デフォルト値は 0 に設定されます。 `-o` オプションが指定されていなくても、その入力を求めるプロンプトは表示されません。この追加メモリ領域では、設定数より多くのオブジェクトを生成できますが、特定のオブジェクトに対していつでも使用できるとは限りません。現在このメモリ領域を使用できるのは、アクション、ハンドル、カーソル、オーナ、一時キュー、タイマー、およびフィルタだけです。

`qspacedestroy (qspds) [-f] [-y] [queue_space_name]`

指定のキュー・スペースを破棄します。コマンド行で指定していないと、入力するよう求められます。指定したキュー・スペースが `qmadmin` でオープンされている場合は、クローズされます。デフォルトでは、このキュー・スペースにプロセスがアタッチされている場合、またはキュー・スペースのいずれかのキューに要求が存在する場合にはエラーが返されます。要求が入っているキューの破棄については、`qdestroy` コマンドを参照してください。 `-f` オプションを指定すると、キューにメッセージがあったり、キュー・スペースにプロセスがアタッチされている場合でも、すべてのキューを「強制的に」削除することができます。このコマンドは、`-y` オプションを指定していない場合に確認のためのプロンプトを表示します。コマンドが正常に完了すると、指定されたキュー・スペースの一時的メッセージはすべて失われます。

`(qspl) [queue_space_name]`

キュー・スペースの作成パラメータを一覧表示します。コマンド行でキュー・スペースを指定しなかった場合は、入力を求めるプロンプトが表示されます。キュー・スペース名を入力しなければ、現在オープンされているキュー・スペースのパラメータが表示されます。キュー・スペースがオープンされていなく、値の入力も行なわない場

合はエラーが発生します。そのキュー・スペースについての値 (qspacecreate でキュー・スペースを作成する際の設定されたもの、または qspacechange で最後に変更されたもの) を表示するほかに、キュー・スペースのエクステントすべてについてサイズが表示されません。また、システムによって予約されているメモリ量や、設定されている共用メモリの総量も表示されます。共用メモリ・リソース用に割り当てられるメモリ量が、バイト単位 (b) で要求したときの量と一致しない場合もあります。説明と例については、qspacecreate の `-n nonpersistent_msg_memory` オプションを参照してください。

quit (q)

セッションを終了します。

verbose (v) [{off | on}]

冗長モードで出力を行います。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。初期設定は off です。

! *shellcommand*

シェルにエスケープし、*shellcommand* を実行します。

!!

直前のシェル・コマンドを再実行します。

[*text*]

で始まる行はコメント行で、無視されます。

<CR>

最後のコマンドを繰り返します。

例 キューの設定手順の例を次に示します。

```
$ QMCONFIG=/dev/rawfs qmadmin
qmadmin - Copyright (c) 1987 ATT; 1991 USL. All rights reserved.
QMCONFIG=/dev/rawfs
# キュー・スペースを配置できるデバイスのリストを作成します。
# 2 つのデバイスを、それぞれ 80,000 ブロック
# および 600 ブロックと指定します。
# 注記 : 1 つ目のデバイスは実際にデバイス・リストを持ちます。
#
# 1 つ目のデバイスを raw スライスに作成します。
#
> crdl /dev/rawfs 0 80000
```

セクション 1 - コマンド

```
Created device /dev/rawfs, offset 0, size 80000 on /dev/rawfs
#
# もう 1 つのデバイスを UNIX ファイルに作成します。
#
> crdl /home/queues/FS 0 600
Created device /home/queues/FS, offset 0, size 600 on /dev/rawfs
#
# デバイス・リストを表示する場合
#
> v Verbose mode is now on

> lidl
universal device index.0:
    name:/dev/rawfs
    start:0
    size:20000
    free space map(1 entry used 47 available):
        size[1]:79974 addr[1]:26
universal device index.1:
    name:/home/queues/FS
    start:0
    size:600
    free space map(1 entry used 47 available):
        size[1]:600 addr[1]:0

#
# キュー・スペースを作成します。
#
> qspacecreate
Queue space name:myqueuespace
IPC Key for queue space:42000
Size of queue space in disk pages:50000
Number of queues in queue space:30
Number of concurrent transactions in queue space:20
Number of concurrent processes in queue space:30
Number of messages in queue space:20000
Error queue name:ERRORQ
Initialize extents (y, n [default=n]):y
Blocking factor [default=16]:16
.....
#
# キュー・スペースをオープンします。
#
> qopen myqueuespace
#

# キュー・スペースのデフォルト設定をキューに適用します。
> qcreate
```

```

Queue name:service1
queue order (priority, time, fifo, lifo):fifo
out-of-ordering enqueueing (top, msgid, [default=none]):top,msgid
retries [default=0]:1
retry delay in seconds [default=0]:30
High limit for queue capacity warning (b for bytes used, B for blocks used,
% for percent used, m for messages [default=100%]):100m
Reset (low) limit for queue capacity warning [default=0m]:50
queue capacity command:/usr/app/bin/mailadmin myqueuespace service1
#
# このプログラムを終了します。
#
> q

```

セキュリティ キューの管理者は BEA Tuxedo 管理者と同じでなければなりません。キューが存在するデバイスは、管理者が所有者でなければならず、`qadmin` は、キューの管理者としてのみ実行できます。キューに割り当てられた IPC リソースはすべて、キューの管理者によって所有され、モード 0600 で作成されます。

移植性 `qadmin` は、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。

Windows 標準 I/O `qchange ...Queue capacity` コマンドなど、`qadmin()` セッションで設定したコマンドを実行するために、Windows の `CreateProcess()` 関数は、子プロセスを `DETACHED PROCESS` として生成します。このようなプロセスには、標準入出力のための関連コンソールがありません。したがって、たとえば、標準コマンド・プロンプトのコマンド行構文を使って `dir` や `date` などの組み込み構文を実行する `qchange ... Queue capacity` を設定した後で、標準出力をファイルにパイプまたはリダイレクトすると、コマンド実行が完了した時点でそのファイルは空になります。

この問題を解決する方法として、たとえば、`qchange ...Queue capacity command` を実行するために、`date /t > x.out` コマンドでファイルの `date` 情報を取得します。この処理を対話的に行うには、次のような手順で実行します。

```

qadmin
> qopen yourQspace
> qchange yourQname
> go through all the setups... the threshold queue capacity warning,
and so on
> "Queue capacity command:"cmd /c date /t > x.out

```

セクション 1 - コマンド

この処理を `yourFile.cmd` などのコマンド・ファイルから行う場合は、`date /t > x.out` コマンドを `yourFile.cmd` に追加してから、次のような手順を実行します。

```
qmadmin
> qopen yourQspace
> qchange yourQname
> go through all the setups... the threshold queue capacity warning,
  and so on
> "Queue capacity command:" yourFile.cmd
```

関連項目 『BEA Tuxedo アプリケーション実行時の管理』

rex(1)

名前 `rex` - オフラインの正規表現コンパイラおよびテスト

形式 コンパイル:

```
rex pattern_file C_file
```

テスト:

```
rex pattern [file . . . ]
```

機能説明 引数なしで起動する場合、`rex` は標準入力から正規表現を読み取り、初期化した文字配列を標準出力に書き込みます。通常、出力は C プログラムにインクルードされます。これにより、実行時間およびプログラムのサイズの両方を節約できます。コマンド `rex` は、標準入力 (通常は入力ファイルから切り換えられる) の正規表現をコンパイルし、出力を標準出力 (通常は出力ファイルに切り換える) に書き込みます。

入力ファイルにはいくつかのパターンがあります。各形式は次のようになります。

```
name string [string . . . ]
```

ここでの `name` は、出力配列に使用する C の名前です。`string` は正規表現で、二重引用符で囲まれます。複数の `string` が `name` の後に続く場合、それらは 1 つの `string` に連結されます。複数の `string` は、厳密にはフォーマット上の都合です。パターンに二重引用符がある場合は、その前にはバックslashが必要で

出力は、C プログラムにインクルードされるかまたはコンパイルされ、後でロードされます。`rex` 出力を使用する C プログラムでは、`rematch(abc,line,0)` は、名前が `abc` である正規表現を `line` に適用します。

次は、サンプルの入力ファイルです。

```
cname "[a-zA-Z_][a-(3c)-Z0-9_]*"
tn      "\\(((0-9){3})$0\\)"
        "([0-9]{3})$1"
        "-"
        "([0-9]{4})$2"
```

これに対応する出力は、次のようになります。

```
/* pattern: "[a-zA-Z][a-zA-Z0-9]*" */
char cname[] = {
040,0,0206,012,0,0210,0141,0172,0210,0101,0132,0137,
... };

/* pattern: "\\((([0-9]{3})$0\\)(([0-9]{3})$1-([0-9]{4})$2" */
char tn[] = {
063,0,050,0202,0225,013,0,03,0206,06,0,0210,060,071,
... };
```

rex に 1 つまたは複数の引数を指定して起動することにより、テストするデータをパターンと突き合わせることができます。最初の引数は、残りの引数で指定される名前のファイルの各行に適用されるパターン（正規表現）として扱われます。ファイル名の引数が指定されない場合、標準入力が使われます。特殊なファイル名 - は、標準入力を指す引数として使われます。

一致するテキストがある場合、一致したものを含む行が出力され、行の一致した部分に下線が引かれます。さらに、指定されたサブ・パターン用に抽出したすべてのテキストが、別の行に出力されます。

次にコマンド rex の使用例を示します。

```
rex '(^| )([0-9]+)$0(|$)'
```

入力は次のとおりです。

```
... or 200 programmers in one week.
This sentence has 3 errors.
I need 12 bad men.
```

出力は次のとおりです。

```
... or 200 programmers in one week.
-----
$0 = `200'

This sentence has 3 errors.
-----
$0 = `3'

I need 12 bad men.
-----
$0 = `12'
```

診断 エラーは、ファイルのオープン・エラー、引数エラーなどです。

関連項目 [tsubscribe\(3c\)](#) の正規表現の情報

tidl(1)

名前 `tidl` - インターフェイス定義言語 (IDL) コンパイラ

形式 `tidl [option] . . . filename [option]. . .`

機能説明 `tidl` は、入力 IDL と関連する ACF ソース・ファイルを解析し、オプションとしてヘッダ・ファイル、クライアント・スタブ、サーバ・スタブ、および補助ファイルを作成します。作成されたソース・コードは、Classic C、ANSI C、または C++ 用のコンパイラでコンパイル可能です。

コマンド行引数には、入力 IDL ソース・ファイルと、IDL コンパイラの動作を制御するオプションを指定します。以下にオプションを示します。

`-client type`

作成するクライアント側のファイルを指定します。 `type` の値は次のとおりです。

`all`

クライアント・スタブ・ファイルと補助ファイルを作成します。 `-client` オプションが指定されない場合には、これがデフォルトになります。

`stub`

クライアント・スタブ・ファイルのみを作成します。

`aux`

クライアント補助ファイルのみを作成します。現時点では補助ファイルは作成されず、このオプションの影響はありません。

`none`

クライアント・ファイルは作成されません。

`-server type`

作成するサーバ側のファイルを指定します。 `type` の値は次のとおりです。

`all`

サーバ・スタブ・ファイルと補助ファイルを作成します。 `-server` オプションが指定されていない場合には、これがデフォルトになります。

`stub`

サーバ・スタブ・ファイルのみを作成します。

aux

サーバ補助ファイルのみを作成します。現時点では補助ファイルは作成されず、このオプションの影響はありません。

none

サーバ・ファイルは作成されません。

`-cstub filename`

クライアント・スタブ・ファイルのファイル名を指定します。ファイル名に `.c` 拡張子が付かない場合には、IDL コンパイラが拡張子を付加します。デフォルトのクライアント・スタブ名 (`-cstub` が指定されない場合) は、`_cstub.c` が付加された IDL ソース・ファイルのベース名 (ディレクトリやピリオドの後の接尾語が付かない単純なファイル名) です。関連するクライアント・スタブ・オブジェクト・ファイルには、クライアント・スタブ・ファイルの名前に `.c` ではなく `.o` が追加されます。

`-sstub filename`

サーバ・スタブ・ファイルのファイル名を指定します。ファイル名に `.c` 拡張子が付かない場合には、IDL コンパイラが拡張子を付加します。デフォルトのサーバ・スタブ名 (`-sstub` が指定されない場合) は、`_sstub.c` が付加された IDL ソース・ファイルのベース名 (ディレクトリやピリオドの後の接尾語が付かない単純なファイル名) です。関連するサーバ・スタブ・オブジェクト・ファイルは、サーバ・スタブ・ファイルの名前に `.c` ではなく `.o` が付加されます。

`-caux filename`

クライアント補助ファイルのファイル名を指定します。ファイル名に `.c` 拡張子が付かない場合には、IDL コンパイラが拡張子を付加します。デフォルトのクライアント補助名 (`-caux` が指定されない場合) は、`_caux.c` が付加された IDL ソース・ファイルのベース名 (ディレクトリやピリオドの後の接尾語が付かない単純なファイル名) です。関連するクライアント補助オブジェクト・ファイルには、クライアント補助ファイルの名前に `.c` ではなく `.o` が付加されます。

`-saux filename`

サーバ補助ファイルのファイル名を指定します。ファイル名に `.c` 拡張子が付かない場合には、IDL コンパイラが拡張子を付加します。デフォルトのサーバ補助名 (`-saux` が指定されない場合) は、`_saux.c` が付加された IDL ソース・ファイルのベース名 (ディレクトリやピリオド

ドの後の接尾語が付かない単純なファイル名)です。関連するサーバ補助オブジェクト・ファイルには、サーバ補助ファイルの名前に `.c` ではなく `.o` が付加されます。

`-header filename`

作成されるヘッダ・ファイルのファイル名を指定します。デフォルトのヘッダ・ファイル名 (`-header` が指定されない場合) は、`.h` が付加された IDL ソース・ファイルのベース名 (ディレクトリやピリオドの後の接尾語が付かない単純なファイル名) です。

`-out directory`

出力ファイルが作成されるディレクトリを指定します。デフォルト (`-out` が指定されない場合) では、ファイルは現在の作業ディレクトリに作成されます。

`-keep type`

どのファイル・タイプを保持するか指定します。デフォルトでは、IDL コンパイラは C ソース・ファイル (クライアント・スタブなど) を作成し、C コンパイラを使用してオブジェクト・ファイルを作り、C のソース・ファイルを削除します。保持できるファイル・タイプは次のとおりです。

`none`

いずれのファイルも作成しないか、C コンパイラを呼び出しません。

`c_source`

C ソース・ファイルのみを保存して、C コンパイラを呼び出しません。

`object`

オブジェクト・ファイルのみを保存して、C ソース・ファイルを削除します。これはデフォルトです。

`all`

C ソース・ファイルとオブジェクト・ファイルの両方を保存します。

`-I directory`

インポートされた IDL ファイルとインクルード・ファイルを検索するディレクトリを指定します。`-I` の後の空白は任意です。`-I` オプション

ンを複数回指定して、複数のディレクトリを指定できます。デフォルトでは、まず現在の作業ディレクトリを検索し、次に `-I` オプションで指定されたディレクトリを指定の順序で検索し、最後にシステムの IDL ディレクトリ (`$TUXDIR/include`) を検索します。この順序は、インクルード・ディレクトリを C プリプロセッサと C コンパイラに渡す場合も同様です。ファイルが複数のディレクトリに存在する場合には、上記の検索順で最初に見付かったファイルが使用されます。

`-no_def_idir`

`-I` オプションと一緒に使用しない場合には、インポート・ファイルとインクルード・ファイルを現在の作業ディレクトリのみで検索するようにします。1 つまたは複数の `-I` オプションと共に使用する場合には、`-I` ディレクトリのみを検索します。現在の作業ディレクトリやシステムの IDL ディレクトリは検索されません。

`-cpp_cmd "cmd"`

ソース・ファイルを拡張するために呼び出す C プリプロセッサ・コマンドを指定します。

`-no_cpp`

C プリプロセッサがソース・ファイルの拡張を行わないようにします。これは、ソース・ファイルにはプリプロセッサ・ディレクティブ (マクロ置換や `#include` など) を含めないことを意味します。

`-cpp_opt "opt"`

C プリプロセッサに渡される追加オプションを指定します。デフォルト値はヌル文字列です。IDL コンパイラは、`-cpp_cmd`、`-cpp_opt`、`-D`、および `-U` 引数 (指定順)、`-I` 引数 (指定順)、およびソース・ファイル名 (IDL または ACF ファイル名) で構成するコマンド行を実行します。

`-D name[=def]`

名前を定義し、オプションとして C プリプロセッサに渡される値も定義します。`-D` オプションを複数回指定することで、複数のシンボルを定義できます。`-D` の後の空白は任意です。

`-U name`

C プリプロセッサのために定義した名前を取り消します。`-U` オプションを複数回指定することで、複数のシンボルを定義を取り消すことができます。`-U` の後の空白は任意です。

- `-cc_cmd "cmd"`
オブジェクト・ファイルを作成するための C コンパイラ・コマンドを指定します。デフォルト (`-cc_cmd` を指定しない場合) は、"`cc -c`" です。
- `-cc_opt "opt"`
追加の C コンパイラ・オプションを指定します。デフォルト (`-cc_opt` を指定しない場合) はヌル文字列です。IDL コンパイラは、`-cc_cmd`、`-cc_opt`、および `-I` 引数 (指定順) およびソース・ファイル名 (スタブ・ファイルまたは補助ファイル名前) で構成するコマンド行を実行します。
- `-syntax_check`
出力ファイルは作成せずに、入力ファイルの構文エラーを調べます。
- `-no_warn`
コンパイラからの警告メッセージが出力されないようにします。
- `-confirm`
ソース・ファイルをのコンパイルは行わず、(明示的に、または暗黙的に) 選択した IDL コンパイラ・オプションを表示します。`-v` オプションと共に使用すると、`-confirm` なしの場合に実行する処理を表示し、実際の処理は行いません。たとえば、入力ファイルの解析、出力ファイルの作成、コンパイル時のメッセージを表示します。
- `-v`
冗長モードを指定します。行われている処理についてのメッセージ (入力ファイルの解析、出力ファイルの作成、コンパイルなど) が標準エラー出力に表示されます。
- `-version`
IDL コンパイラのバージョン番号を表示します。
- `-stdin`
ファイルからではなく、標準入力から IDL ソース入力を獲得します。デフォルトのファイル名は、入力 IDL ソース・ファイルが `a.idl` と命名されているものとして作成されます。たとえば、デフォルトのクライアント・スタブ・ファイルの名前は `a_cstub.c` になります。
- `-cepv`
クライアント・エントリ・ポイント・ベクトル (CEPV) を作成します。デフォルトではクライアント・スタブ・モジュール内の関数は、イン

ターフェイス定義でのオペレーション名に命名されます。ただし、これは複数バージョンのインターフェイス、同じオペレーション名を持つインターフェイス、または同じクライアント・プログラム（オペレーション名が複数回定義される）にリンクされるローカルとリモートの同一関数には許可されません。

`-cepv` オプションを指定すると、関数名はクライアント・スタブに対してローカルに宣言され、クライアント・エントリ・ポイント・ベクトル（関数ポインタの配列）が、`interface_vmajor_minor_c_epv` という名前でクライアント・スタブ内に（グローバルに）定義されます。`interface` はインターフェイス名で、`major` はメジャー・バージョン番号、`minor` はマイナー・バージョン番号です。インターフェイス・オペレーションは、CEPV 内のアドレスを使用して間接的に呼び出す必要があります。

`-no_mepv`

マネージャ（サーバ）・エントリ・ポイント・ベクトル（MEPV）を作成しません。デフォルトでは、サーバ内のアプリケーション関数は、インターフェイス定義のオペレーション名に命名されるものと想定します。ただし、これは複数バージョンのインターフェイス、同じオペレーション名を持つインターフェイス、または同じサーバ・プログラム（オペレーションが複数回指定される）にリンクされるローカルおよびリモートの同一関数には許可されません。通常、マネージャ・エントリ・ポイント・ベクトル（関数ポインタの配列）は、

`interface_vmajor_minor_s_epv` という名前でサーバ・スタブ内に（グローバルに）定義されます。`interface` はインターフェイス名、`major` はメジャー・バージョン番号、`minor` はマイナー・バージョン番号で、オペレーション名で初期化されます。マネージャ・エントリ・ポイント・ベクトルは、アプリケーション・サービス関数の呼び出しに使用します。`-no_mepv` オプションを指定すると、MEPV はサーバ・スタブ内には作成されず、アプリケーションで構造体を設定する必要があります。この方法では、アプリケーションでは、エントリ・ポイントで使用する任意の名前に設定することができます。

`-error all`

追加のエラー・チェックを指定します。デフォルトでは、IDL コンパイラはエラーが 50 回検出されると終了します。

`-port level`
移植性検査のレベルを指定します。次のレベルがサポートされていません。

`case`
2つの識別子に大文字と小文字の区別がない場合には、警告メッセージを出力します。

`none`
追加の移植性検査を行いません。これがデフォルトの設定です。

`-no_enum_lit`
列挙リテラルがスタブ・ファイルに作成されないようにします。デフォルトでは列挙リテラルが作成されます。

`-use_const`
`#define` 定義の代わりに ANSI C `const` 宣言が定数値に使用されます。

IDL ソース・ファイルとすべてのインポート IDL ファイルでは、コンパイラは IDL ソース・ファイルのベース名に接尾語 `.acf` が付いた関連する ACF を検索します。検索するディレクトリは C プリプロセッサで指定されたものと同じ (前述の `-I` と `-no_def_idir` を参照) で、ACF ファイルは IDL ソース・ファイル名で指定されたディレクトリで検索されます。

注意事項 IDL のファイル名 `tbase.idl` は、IDL コンパイラのために予約されています。

使用例 IDL ソース・ファイル `math1.idl` の例を次に示します。

```
[uuid(2048A080-0B0F-14F8-26E0-930269220000)]
interface math1
{
import "math2.idl";

long add_op([in] long first1, [in] long second);
long sub_op([in] long first1, [in] long second);
}
```

次に、サンプルの ACF ソース・ファイル `math1.acf` を示します。

```
[auto_handle]
interface math1
{
include "stdio";
```

```
[code] add_op([fault_status,comm_status] result);  
}
```

次のコマンド行は、out/math1.h と共に、クライアント側のファイル out/math1_cs.c と out/math1_cs.o のみを作成する（補助ファイルは不要） math1.idl をコンパイルします。IDL コンパイラは、カレント・ディレクトリ、app サブディレクトリ、\$TUXDIR/include の順に math2.idl（除算と乗算を行う）を検索します。

```
tidl math1.idl -Iapp -client all -server none -keep all  
-cstub math1_cs -out app
```

関連項目 [uuidgen\(1\)](#)

tlisten(1)

名前 tlisten - 汎用リスナ・プロセス

形式 tlisten [-d *device*] -l *nlsaddr* [-u {*uid-#* | *uid-name*}] [-z *bits*] [-Z *bits*]

機能説明 tlisten は、BEA Tuxedo ATMI アプリケーション・プロセッサ上のデーモン・プロセスとして動作するネットワーク独立型のリスナ・プロセスで、ほかの BEA Tuxedo ATMI プロセス (tmbboot(1) など) に対してリモート・サービス接続を行います。tlisten は次のようなコマンド行オプションを使用します。

-d *device*

ネットワーク・デバイスの絶対パス名。リリース 6.4 以降では、このパラメータは省略できます。それ以前のリリースでは、基本となるネットワーク・プロバイダから要求される場合に使用してください。

-l *nlsaddr*

プロセスが接続をリスンするネットワーク・アドレス。TCP/IP アドレスは、次のいずれかの形式で指定できます。

"//hostname:port_number"

"//#.##.##.##:port_number"

最初の形式では、tlisten は、ローカル名の解決機能 (通常 DNS) を利用して *hostname* のアドレスを見付けます。*hostname* はローカル・マシンでなければなりません。ローカル名解決機能によって *hostname* をローカル・マシンのアドレスに明確に解決します。

2 番目の例の文字列 *#.##.##.##* はドットで区切った 10 進数の形式です。ドット区切りの 10 進数形式では、各 # には 0 ~ 255 の数字を使用します。このドット区切りの 10 進数は、ローカル・マシンの IP アドレスを表します。どちらの形式の場合も、*port_number* は tlisten プロセスが入力要求に対してリスンする TCP ポート番号です。*port_number* は 0 ~ 65535 の数値または名前です。

注記 一部のポート番号は、お使いのシステムで使用されるトランスポート・プロトコル (TCP/IP など) のために予約されている場合があります。予約されているポート番号を確認するには、トランスポート・プロトコルのマニュアルを調べてください。

`port_number` が名前の場合は、ローカル・マシン上でネットワーク・サービス・データベースの中からその名前を見つけなければなりません。先頭に `&dlq;"q;` を付けて、そのアドレスを 16 進形式で指定できます。`&dlq;"q;` の後の各文字は、0 ~ 9 の数字か、A ~ F のアルファベット（大文字と小文字は区別しない）です。16 進数の形式は、IPX/SPX や TCP/IP のような任意のバイナリ・ネットワーク・アドレスに使うことができます。アドレスはまた、任意の文字列として指定することもできます。値は、コンフィギュレーション・ファイルの中の NETWORK セクションの NLSADDR パラメータに指定された値と同じでなければなりません。

`-u {uid-# | uid-name}`

`tlisten` は、指定されたユーザとして動作します。このオプションにより、`root` によるシステムの初期化処理の一部として `tlisten` を起動できます。`tlisten` の実行ユーザが `root` である場合には、このオプションは必須です。したがって、`tlisten` プロセスは `root` が実行することは可能ですが、`root` としては動作しません。`tlisten` コマンドを `root` 以外のユーザが使用する場合には、`-u` オプションは必要はありません。`root` 以外のユーザは `-u` オプションを設定することはできませんが、このオプションにはそのユーザ自身のユーザ ID にしか設定できないので、あまり意味はありません。プロセッサ上での `tlisten` プロセスは、使用するアプリケーション管理ユーザ ID が同じである BEA Tuxedo ATMI アプリケーションすべてをサポートできます。

`-z [0 | 40 | 56 | 128]`

BEA Tuxedo の管理プロセスと `tlisten` との間のネットワーク・リンクを確立する場合の、最小レベルの暗号化を指定します。0 は暗号化が行われないことを示し、40、56、および 128 は暗号化キーの長さをビット単位で指定します。ここで指定する最小レベルの暗号化が満たされない場合、リンクの確立は失敗します。デフォルト値は 0 です。このオプションを使用できるのは、国際版または米国 / カナダ版の BEA Tuxedo セキュリティのライセンスがインストールされている場合に限られます。

注記 リンク・レベルの暗号化の値 40 は、下位互換性を維持するために提供されています。

-z [0 | 40 | 56 | 128]

BEA Tuxedo の管理プロセスと `tlisten` との間にネットワーク・リンクを確立する場合の、最大レベルの暗号化を指定します。0 は暗号化が行われないことを示し、40、56、および 128 は暗号化キーの長さをビット単位で指定します。デフォルト値は 128 です。このオプションを使用できるのは、国際版または米国 / カナダ版の BEA Tuxedo セキュリティのライセンスがインストールされている場合に限りません。

注記 リンク・レベルの暗号化の値 40 は、下位互換性を維持するために提供されています。

`tlisten` プロセスは、ほとんどのサービス要求を認証します。`tlisten` は、パスワードの一覧ファイルを読み取ります。サービスを要求するプロセスは、そのファイル内にあるパスワードを少なくとも 1 つ提示しなければなりません。`APPDIR` 環境変数が設定された場合、`APPDIR/.adm/tlisten.pw` という名前のファイルからパスワードが取得されます。このファイルが見付からない場合、システムは `TUXDIR/udataobj/tlisten.pw` という名前のファイルを検索します。このファイルは BEA Tuxedo ATMI システムのインストール時に作成されます。長さがゼロまたはパスワード・ファイルが見付からない場合は、パスワードのチェックは行われません。こうしたセキュリティで保護されないモードでの起動時は、`tlisten` および `tlisten` に接続しているプロセスにより警告メッセージがユーザ・ログに書き込まれます。

`tmbboot` などのように、`tlisten` からサービスを要求するプロセスは、認証の際に使用するパスワードをプロセスが稼働しているマシンのファイルから検索します。このとき、`tlisten` と同じ方法でパスワード・ファイルを検索します。

- 環境変数
- `tlisten` コマンドを呼び出す前に、`TUXDIR` を設定しエクスポートしておく必要があります。
 - 共有オブジェクトを使用する SVR4 アプリケーションの場合も、環境変数 `LD_LIBRARY_PATH` を `TUXDIR/lib` に設定してから `tlisten` プロセスを起動する必要があります。一部の UNIX システムでは、別の環境変数が必要となる場合もあります。HP-UX システムでは `SHLIB_PATH` 環境変数を使用します。AIX システムでは `LIBPATH` 環境変数を使用します。
 - `APPDIR` は、`tlisten` パスワード・ファイルの位置を検索するために設定します。

- ULOGPFX は、ログ・メッセージを格納しているファイルを送信するのに使用できます。

注記 インストールの間に管理パスワード・ファイルが作成されます。BEA Tuxedo ATMI システムでは、必要に応じて次のディレクトリでこのファイルを検索します。検索順序は次のとおりです。

- APPDIR/.adm/tlisten.pw
- TUXDIR/udataobj/tlisten.pw

管理パスワード・ファイルが確実に見付かるように、APPDIR または TUXDIR 環境変数を必ず設定してください。

リンクレベルの暗号化	リンクレベル暗号化機能が <code>tlisten</code> と <code>tmboot</code> などの要求プロセスとの間で動作している場合は、リンクレベル暗号化を取り決め、アクティブにしてから認証します。
プロセスの終了	<code>tlisten</code> プロセスは通常、SIGTERM シグナルを送ることによってしか終了できません。
推奨される 使用法	システムの起動時、アプリケーションごとに1つの <code>tlisten</code> プロセスを起動することをお勧めします。 <code>tlisten</code> を呼び出す前に、環境変数 <code>TUXDIR</code> および <code>APPDIR</code> を設定しておく必要があります。 <code>tlisten</code> プロセスを起動する別の方法として、手動による起動もできます。 <code>tlisten</code> プロセスをアプリケーション管理者が起動する場合、 <code>-u</code> オプションは省略できます。同じネットワーク・アドレスを使用して <code>tlisten</code> コマンドを重複して呼び出すと、コマンドは自動的に終了し、該当するメッセージが記録されます。
ネットワーク・アドレス	<code>tlisten</code> を実行するローカル・マシンが TCP/IP アドレス指定機能を使用しており、アドレスが <code>155.2.193.18</code> 、マシン名が <code>backus.company.com</code> であるとします。さらに、 <code>tlisten</code> はポート番号 <code>2334</code> で要求を受け取るとします。このポート番号 <code>2334</code> が <code>bankapp-nlsaddr</code> という名前のネットワーク・サービス・データベースに追加されているとします。 <code>-l</code> オプションで指定するアドレスは、次に示す方法で表現できます。 //155.2.193.18:bankapp-nlsaddr //155.2.193.18:2334 //backus.company.com:bankapp-nlsaddr //backus.company.com:2334 0x0002091E9B02C112

最後の表現は 16 進形式です。0002 は TCP/IP アドレスの先頭部分、091E は 16 進数に変換されたポート番号 2334 です。その後、IP アドレス 155.2.193.12 の各要素は 16 進数に変換されています。したがって、155 は 9B、2 は 02 のようになります。

STARLAN ネットワークの場合は、通常、推奨アドレス `uname.tlisten` が一意の名前になります。

Windows コントロール・パネル・アプレットを介してリモート Windows マシン上で `tlisten` プロセスを起動するためには、そのリモート・マシンに対する管理者権限が必要です。

関連項目 [UBBCONFIG\(5\)](#)

tmadmin(1)

名前 tmadmin - BEA Tuxedo 掲示板コマンド・インタプリタ

形式 tmadmin [-r] [-c] [-v]

機能説明 tmadmin は、以下に説明するコマンドをもち、単一プロセッサ、マルチプロセッサ、あるいはネットワーク環境において掲示板や関連するエンティティ (クライアント、サーバ、サービス等のそれぞれの情報) の検査および修正を行うときに使用します。TUXCONFIG および TUXOFFSET 環境変数を使用して、BEA Tuxedo のコンフィギュレーション・ファイルがロードされた場所とオフセットを決定します。

-c オプションを使用して tmadmin を呼び出すと、コンフィギュレーション・モードになります。有効なコマンドは、default、echo、help、quit、verbose、livtoc、crdl、lidl、dsdl、indl、および dumplog です。

tmadmin は、非アクティブ状態のノードも含め、どのノードでもこのモードで呼び出せません。tmadmin が実行中の BBL を介してアプリケーションに管理者プロセスまたはクライアントとして結合できる場合には、そのノードがアクティブであると見なされます。

-r オプションを使用すると、tmadmin は、管理者としてではなくクライアントとして掲示板に参加します。つまり、読み取り専用モードを要求します。これは、管理者プロセス用スロットを非占有状態にしておく場合に有用です。

注記 -r オプションを使用すると、このオプションなしで tmadmin を実行した際に取得される情報の一部しか得られないことに注意してください。特に、tmadmin -r を実行した場合、リモート・サイトで動作しているサーバの負荷値は取得できません。

管理プロセスになれるのは、一度に 1 つの tmadmin プロセスのみです。BEA Tuxedo 管理者以外のユーザが -r オプションを指定し、セキュリティがオンになっている場合、ユーザはパスワードを求められます。

-v オプションを使用して tmadmin を実行すると、BEA Tuxedo のバージョン番号とライセンス番号が表示されます。情報を出力した後、tmadmin は終了します。その他 2 つのオプションは、-v オプションと共に使用しても無視され、-v オプションによって要求される情報だけが表示されます。

通常、`tmadmin` はアクティブ・アプリケーション内であれば、任意のアクティブ・ノード上で実行できます。分断されているアクティブ・ノード上で実行する場合、使用できるコマンドはローカル掲示板への読み取り専用のアクセスを行うものに限られます。これらのコマンドには、`bbbs`、`bbparms`、`bbstat`、`default`、`dump`、`dumplog`、`echo`、`help`、`interfaceparms`、`printactiveobject`、`printclient`、`printinterface`、`printfactory`、`printnet`、`printqueue`、`printroute`、`printserver`、`printservice`、`printtrans`、`printgroup`、`reconnect`、`quit`、`serverparms`、`serviceparms`、および `verbose` があります。これらのほか、構成コマンドも使用できます。この分断ノードが MASTER のバックアップ・ノードである場合 (コンフィギュレーション・ファイルの `RESOURCES` セクションで MASTER パラメータの 2 つめのエントリとして指定)、`master` コマンドを使用して、分断アプリケーションのこの部分の MASTER にこのノードを設定することも可能です。

アプリケーションがアクティブ状態でない場合、`tmadmin` は MASTER プロセッサ上でしか実行できません。このモードでは、すべてのコンフィギュレーション・モード・コマンドと共に、`TLOG` コマンド (`crlog`、`dslog`、および `inlog`) と `boot` も利用できます。

`tmadmin` が呼び出された後は、次に示す構文に従って、プロンプト (`>`) からコマンドを入力してください。

```
command [arguments]
```

共通して出現するいくつかの引数には、`default` コマンドを使用してデフォルト値が与えられます。`default` コマンドで設定されたパラメータを受け付けるコマンドは、`default` をチェックして、値が設定されているかどうかを確認します。値が設定されていない場合は、エラー・メッセージが返されません。

ネットワーク環境またはマルチプロセッサ環境では、`default machine` を設定して単一の掲示板にアクセスできます。このマシン (`machine`) には、`UBBCONFIG` ファイルの `MACHINES` セクションで記述される論理 `machine ID` (`LMID`) を指定します。`default machine` に `all` を設定すると、すべての掲示板にアクセスできます。`machine` に `DBBL` を設定すると、特別掲示板が指定されます。`default machine` はプロンプトの一部として表示されます。たとえば、プロンプト `MASTER>` の一部として表示されます。

default コマンドによって *machine* が設定されていない場合、DBBL が使用されます。SHM モードではローカル BBL が使用されます。

コマンド行で指定する *machine* 値は、通常は default の設定 (printserver など) から取得できます。ただし、TLOG など、コマンドによっては TUXCONFIG で検出されるデバイスに対して実行されるので注意が必要です。この場合、default に DBBL または all を設定するとエラーになります。logstart などのコマンドでは、コマンド行で *machine* の値を指定する必要があります。この場合、*machine* 値は -m オプションの引数で指定しません。

一度設定したデフォルト値は、default コマンドで変更しない限り、セッション終了まで有効です。このデフォルト値は、コマンド行で明示的に値を入力して変更するか、&ldq;*&drq“ して設定解除します。こうして変更した値は、コマンドの単一のインスタンスの間有効です。

tmadmin コマンドの出力は、ページ別表示コマンドを指定することでページ別に表示できます。後述の *paginate* サブコマンドの説明を参照してください。

詳細情報または要約情報を出力するコマンドがあります。verbose コマンドは、デフォルトの出力レベル設定に使用します。ただし、各コマンド (boot、shutdown、および config を除く) には、そのコマンドのみ詳細または要約情報を出力するための -v または -t オプションがあります。要約モードで出力すると、一部の情報 (LMID または GROUP 名、サービス名、サーバ名など) は省略されることがあります。省略された場合は、値の末尾にプラス記号 (+) が付加されます。情報をすべて表示するには、詳細モードでコマンドを再入力します。

tmadmin の
コマンド

コマンドは、そのフルネームあるいは (かっこ内に示される) 省略名のいずれかで入力し、適切な引数が続きます。角かっこ [] で囲まれた引数は省略可能です。中かっこ { } で囲まれた引数は相互に排他的な選択肢を示します。該当するデフォルト値が default コマンドで設定されている場合は、そのコマンド・オプションをコマンド行で指定する必要がないことに注意してください。{ } で囲まれたオプションのグループに続く ... は、コマンド行で 1 つ以上 (少なくとも 1 つ必要) のオプションを指定できることを示します。

aborttrans (abort) [-yes] [-g *groupname*] *tranindex*

コマンド行またはデフォルトにより *groupname* が指定されると、指定サーバ・グループの指定されたトランザクション・インデックス

(*tranindex*) に対応するトランザクションをアボートさせます。このオプションの指定がない場合は、トランザクションのコーディネータにグローバル・トランザクションのアボートを通知します。トランザクション既かが決定されており、その決定がコミットである場合は、`aborttrans` は失敗します。インデックスには、`printtrans` コマンドを前回実行したときの結果が使用されます。トランザクションを完全に終了させるには、そのトランザクションに関わっているすべてのグループに対して `printtrans` および `aborttrans` を実行する必要があります。このコマンドの使用には、十分な注意が必要です。

```
advertise (adv) {-q qaddress [ -g groupname ]  
[-i srvid] | -g groupname -i srvid} service[:func]
```

指示されたサービスに対してサービス・テーブルを作成します。*service* は関数 *func* にマッピングすることができます。*qaddress* が指定されていない場合は、*groupname* と *srvid* を指定し、サーバを一意に識別する必要があります。この *service* を MSSQ セットに追加する場合には、そのセットのすべてのサーバでこのサービスを宣言します。MSSQ セットの一部のサーバしかそのサービスを宣言できない場合は、その宣言全体が無効です。文字 (.) で始まるサービスはシステム・サーバで使用するために予約されているため、アプリケーション・サーバでは宣言できません。

```
bbclean (bbc) machine
```

マシン *machine* 上の掲示板のすべてのアクセス・プロセス、および DBBL の完全性をチェックします。`bbclean` は、停止したサーバを適切に削除し、それが再起動可能であれば再起動します。また、どのプロセスからも利用されていないリソースがあれば、それらも削除します。最後に `bbclean` は、各 BBL の状態を DBBL にチェックさせます。SCANUNIT の時間内に応答しない BBL は、分断された掲示板としてマークされます。特殊掲示板のみをクリーンアップするには、*machine* に DBBL を指定します。SHM モードでは、`bbclean` は BBL を再起動します (失敗した場合)。*machine* パラメータは任意です。

```
bbparms (bbp)
```

掲示板のパラメータの要約を表示します。たとえば、サーバの最大数、オブジェクトの最大数、インターフェイスの最大数、サービスの最大数を出力します。

bbsread (bbs) *machine*

マシン *machine* 上の掲示板の IPC リソースのリストを表示します。
SHM モードでは、*machine* パラメータは任意です。リモート・マシン
からの情報は使用できません。

bbstats (bbs)

掲示板に関する統計情報の要約を表示します。shmstats も参照してく
ださい。

boot (b) [*options*]

このコマンドは tmbboot() コマンドと同じです。オプションの説明お
よび使用上の制限事項については、tmbboot(1) を参照してください。

broadcast (bcst) [-m *machine*] [-u *usrname*] [-c *cltname*] [*text*]

任意通知型メッセージをすべての選択クライアントに送信します。送
信されるメッセージは STRING 型の型付きバッファで、*text* で指定さ
れたデータが入ります。ただし、*text* は 80 文字を超えないようにし
てください。*text* に複数の単語を指定する場合には、“*text text*” の
ように全体を引用符で囲む必要があります。パラメータが指定されて
いない (かつデフォルト値がない) 場合、その識別子にワイルドカー
ド値が指定されたものと解釈されます。

changeload (chl) [-m *machine*] {-q *qaddress* [-g *groupname*]
[-i *srvid*] -s *service* | -g *groupname* -i *srvid* -s *service* | -I *interface*
[-g *groupname*]} *newload*

指定サービスまたはインターフェイスのロード・ファクタを *newload*
に変更します。*qaddress* が指定されない場合、サーバを一意に識別す
るために *groupname* と *srvid* 必ず指定する必要があります。CORBA
環境では、*interface* を指定します。*machine* が all であるか、設定
されていない場合には、全マシンに対して変更がなされます。それ以
外の場合は、ローカルの変更は指定した *machine* でのみ有効となりま
す。ローカルの修正は、引き続きグローバル (またはローカル) の変
更によって上書きされます。

changepriority (chp) [-m *machine*] {-q *qaddress* [-g *groupname*]
[-i *srvid*] -s *service* | -g *groupname* -i *srvid* -s *service* | -I *interface*
[-g *groupname*]} *newpri*

指定サービスまたはインターフェイスがキューから取り出す時の優先
順位を *newpri* に変更します。*qaddress* が指定されない場合、サーバ

を一意に識別するために *groupname* と *srvid* 必ず指定する必要があります。CORBA 環境では、*interface* を指定します。*machine* が *all* であるか、設定されていない場合には、全マシンに対して変更がなされます。それ以外の場合は、ローカルの変更は指定した *machine* でのみ有効となります。ローカルの修正は、引き続きグローバル (またはローカル) の変更によって上書きされます。

`changetrace (chtr) [-m machine] [-g groupname] [-i srvid] newspec`
 現在実行しているプロセスの実行時トレース動作を *newspec* に変更します (*newspec* の構文については、[tmtrace\(5\)](#) を参照してください)。実行中の特定のサーバ・プロセスのトレース仕様を変更するには、`-g` および `-i` オプションを指定します。特定のグループで実行中のサーバ・プロセスの構成を変更するには、`-i` オプションを指定せずに `-g` オプションのみを指定します。特定のマシンで実行中のすべてのクライアントおよびサーバ・プロセスの構成を変更するには、`-m` オプションを指定します。`-g`、`-i`、および `-m` のいずれのオプションも指定しなかった場合は、デフォルトのマシンのすべての非管理のプロセスが対象となります。このコマンドは、現在実行中でないクライアントやサーバ、および /WS クライアントの動作には影響を与えません。

`changetrtime (chtt) [-m machine] {-q qaddress [-g groupname] -
 [-i srvid] -s service | -g groupname -i srvid -s service | -I interface
 [-g groupname] } newtlim`
 指定サービスまたはインターフェイスのトランザクション・タイムアウト値を *newtlim* に変更します。*qaddress* が指定されない場合、サーバを一意に識別するために *groupname* と *srvid* を必ず指定する必要があります。CORBA 環境では、*interface* を指定します。*machine* が *all* であるか、設定されていない場合には、全マシンに対して変更がなされます。それ以外の場合は、ローカルの変更は指定した *machine* でのみ有効となります。ローカルの修正は、引き続きグローバル (またはローカル) の変更によって上書きされます。

`committrans (commit) [-yes] -g groupname tranindex`
 指定トランザクション・インデックス *tranindex* に対応するトランザクションを指定サーバ・グループでコミットします。トランザクションが指定サーバ・グループでプリコミットされていなかった場合、あるいはトランザクションが「アボートのみ」として認識されている場合は、`committrans` は異常終了します。インデックスには、`printtrans` コマンドを前回実行したときの結果が使用されます。

`-yes` オプションを指定せずにこのコマンドを実行すると、処理を進める前に確認を求められます。このコマンドの使用には、十分な注意が必要です。

`config (conf)`

このコマンドは `tmconfig` コマンドと同じです。使用方法については、[tmconfig](#)、[wtmconfig\(1\)](#) を参照してください。

`crdl -b blocks -z config -o configoffset [-O newdefoffset] [newdevice]`

汎用デバイス・リストにエントリを作成します。`blocks` は、デバイスに割り当てる物理ブロック数を指定します。デフォルトの `blocks` 値は、1000 ブロックに初期化されます。`configoffset` は、割り当て開始空間のブロック番号を指定します。`-o` オプションを指定せず、デフォルト値も設定されていない場合は、環境変数 `FSOFFSET` の値が使用されます。`FSOFFSET` が設定されていない場合、デフォルトで 0 になります。`config` 引数には、最初のデバイス (デバイス・リストを格納している) を指定します。これは、/ で始まる絶対パス名でなければなりません。`-z` オプションを指定せず、デフォルト値も設定されていない場合は、`FSCONFIG` 環境変数に指定されているパスが使用されます。`crdl` コマンドへの `newdevice` 引数を指定する場合、この引数には作成するデバイスを指定します。これは、/ で始まる絶対パス名でなければなりません。このパラメータの指定を省略すると、`newdevice` はデフォルトで構成デバイスになります。`newdefoffset` は `newdevice` の先頭へのオフセットを指定します。デフォルトの `-o` オプション (大文字の O) と共に指定されていない場合、デフォルトで 0 (ゼロ) が設定されます。

`crlog (crlg) -m machine`

指定またはデフォルトの `machine` (“DBBL” または “all&dr” 定できません) に対して DTP トランザクション・ログを作成します。コンフィギュレーション・ファイル `TLOG` がそのマシン用に定義されていなければ、エラーになります。このコマンドは、`TUXCONFIG` ファイルを参照して、`TLOG` を格納している BEA Tuxedo ファイルシステム、そのファイルシステム内での `TLOG` の名前、オフセット、およびサイズを判別します ([UBBCONFIG\(5\)](#) を参照)。

```
default (d) [-g groupname] [-i srvId] [-m machine] [-u usrname] [-c cltname]
[-q qaddress] [-s service] [-b blocks] [-o offset] [-z config] [-a { 0|1|2}]
[-I interface] [-B objectid] [-r routingname]
```

対応する引数をデフォルトのグループ名、サーバ ID、マシン、ユーザ名、クライアント名、キュー、サービス名、デバイス・ブロック、デバイス・オフセット、あるいは UDL 構成デバイス・パス (/ で始まる絶対パス名でなければなりません) などに設定します。-a オプションについては、`printservice` を参照してください。CORBA 環境では、対応する引数に、デフォルトのオブジェクトのインターフェイス名、オブジェクト ID、またはファクトリベースのルーティング名を設定することもできます。-B を使用してオブジェクト ID パラメータが指定されている場合、マシン引数 (-m) を指定する必要もあります。すべてのデフォルト値は、引数として * を指定することにより設定を解除できます。`machine` に設定されている場合、以降の検索を特殊掲示板から行わせるには、`machine` に DBBL を設定する必要があります。`machine` の設定解除 (-m *) は DBBL の設定と等値です。default コマンドを引数なしで入力した場合は、現在のデフォルトの値が表示されません。

```
dsdl [-yes] [-z config] [-o offset] dlindex
```

汎用デバイス・リストにあるエントリを破棄します。`dlindex` 引数は、汎用デバイス・リストから削除されるデバイスのデバイス・リストにおけるインデックスです。ただし、エントリ 0 は vTOC ファイルとその他のデバイス・リスト・エントリをすべて最初に破棄した後でなければ破棄できません (エントリ 0 には、デバイス・リストと目次が記録されていて、これを破棄すると、これら 2 つのテーブルも破棄されてしまうためです)。`config` 引数には、汎用デバイス・リストを格納するデバイスを指定します。/ で始まる絶対パス名でなければなりません。-z オプションを指定せず、デフォルト値も設定されていない場合は、`FSCONFIG` 環境変数に指定されているパスが使用されます。`offset` オプションを使用すると、`config` へのオフセットを指定できます。-o オプションを指定せず、デフォルト値も設定されていない場合は、環境変数 `FSOFFSET` の値が使用されます。`FSOFFSET` が設定されていない場合、デフォルトで 0 になります。-yes オプションを指定せずに、このコマンドを実行すると、処理を進める前に確認を求められます。

`dslog (dslg) [-yes] -m machine`

指定またはデフォルトの *machine* (“DBBL” または “all&dr” 定できません) の DTP トランザクション・ログを破棄します。TLOG がマシンに定義されていない場合、アプリケーションが非アクティブでない場合、あるいは未終了のトランザクション・レコードがログに残っている場合には、エラーが返されます。未終了のトランザクションとは、グローバル・トランザクションがコミットされたが、まだトランザクションの終わりが書き込まれていないという意味です。このコマンドは、TUXCONFIG ファイルを参照して、TLOG を収めている BEA Tuxedo ファイルシステム、およびそのファイルシステム内の TLOG の名前を判別します。-yes オプションを指定せずにこのコマンドを実行すると、処理を進める前に確認を求められます。

`dump (du) filename`

現在の掲示板をファイル *filename* にダンプします。

`dumptlog (dl) -z config [-o offset] [-n name] [-g groupname] filename`

TLOG のテキスト・バージョンを、指定された *filename* にダンプします。TLOG は指定の *config* および *offset* にあり、指定の *name* が使用されます。-n オプションを指定せず、デフォルト値も設定されていない場合は、&dq;drq; が使用されます。*config* 引数には、汎用デバイス・リストを格納するデバイスを指定します。/ で始まる絶対パス名でなければなりません。-z オプションを指定せず、デフォルト値も設定されていない場合は、FSCONFIG 環境変数に指定されているパスが使用されます。-o *offset* オプションを使用すると、*config* のオフセットを指定できます。-o オプションを指定せず、デフォルト値も設定されていない場合は、環境変数 FSOFFSET の値が使用されます。FSOFFSET が設定されていない場合、デフォルトで 0 になります。*groupname* の指定があれば、そのグループがコーディネータとなっているトランザクションのログ・レコードだけがダンプされます。

`echo (e) [{off | on}]`

on に設定すると、入力コマンド行をエコーします。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。初期設定は off です。

`help (h) [{command | all}]`

ヘルプ・メッセージを表示します。*command* が指定されている場合は、そのコマンドの略称、引数、および説明が表示されます。all を指定

すると、すべてのコマンドの説明が表示されます。引数をすべて省略すると、すべてのコマンドの構文が表示されます。

`initdl (indl) [-yes] -z config [-o offset] dlindex`

デバイス・リストのデバイスを再初期化します。引数 *dlindex* は、再初期化するデバイスに対応する、汎用デバイス・リストのデバイス・インデックスです。指定デバイスの領域はすべて解放されます。そのため、デバイスに格納されているファイルなどが上書きされる可能性があります。このコマンドの使用には注意が必要です。-yes オプションを指定せずにこのコマンドを実行すると、処理を進める前に確認を求められます。config 引数には、汎用デバイス・リストを格納するデバイスを指定します。/ で始まる絶対パス名でなければなりません。-z オプションを指定せず、デフォルト値も設定されていない場合は、FSCONFIG 環境変数に指定されているパスが使用されます。-o offset オプションを使用すると、config のオフセットを指定できます。-o オプションを指定せず、デフォルト値も設定されていない場合は、環境変数 FSOFFSET の値が使用されます。FSOFFSET が設定されていない場合、デフォルトで 0 になります。

`inlog [-yes] -m machine`

指定またはデフォルトの *machine* (“DBBL” または “all&dr” 定できません) の DTP トランザクション・ログを再初期化します。そのマシンに TLOG が定義されていない場合、またはアプリケーションが非アクティブでない場合は、エラーが返されます。未処理のトランザクションが TLOG に残っている場合は、これらのトランザクションに参加しているリソース・マネージャ間で、データの一貫性が損なわれている可能性があります。リソース・マネージャが、ローカル・トランザクションを正しくコミットせずに、ローカル・トランザクションをアボートさせてしまっている可能性があるからです。このコマンドは、TUXCONFIG ファイルを参照して、TLOG を収めている BEA Tuxedo ファイルシステム、およびそのファイルシステム内の TLOG の名前を判別します。-yes オプションを指定せずにこのコマンドを実行すると、処理を進める前に確認を求められます。

`interfaceparms (ifp) -g groupname -I interface`

特定のオブジェクト・インターフェイスに関する情報を出力します。インターフェイス名、負荷、優先順位、タイムアウト、トランザクションタイム・アウト値などが出力されます。引数の *groupname* と

interface は一意でなければなりません。このコマンドは、CORBA 環境でのみ使用されます。

`lidl -z config [-o offset] [dlindex]`

汎用デバイス・リストを表示します。各デバイスごとに、名前、開始ブロック、およびブロック数が表示されます。冗長モードでは、フリー・スペースを示すマップが表示されます(開始アドレスとサイズ)。*dlindex* を指定すると、デバイス・リストのそのエントリについての情報のみが表示されます。*config* 引数には、汎用デバイス・リストを格納するデバイスを指定します。/ で始まる絶対パス名でなければなりません。*-z* オプションを指定せず、デフォルト値も設定されていない場合は、`FSCONFIG` 環境変数に指定されているパスが使用されます。*-o offset* オプションを使用すると、*config* のオフセットを指定できます。*-o* オプションを指定せず、デフォルト値も設定されていない場合は、環境変数 `FSOFFSET` の値が使用されます。`FSOFFSET` が設定されていない場合、デフォルトで 0 になります。

`livtoc -z config [-o offset]`

VTOC テーブルのすべてのエントリについて情報を表示します。各エントリについて表示される情報は、VTOC テーブルの名前、VTOC テーブルのあるデバイス、デバイスの先頭からの VTOC テーブルのオフセット、およびこのテーブルに割り当てられているページ数があります。*config* 引数には、汎用デバイス・リストを格納するデバイスを指定します。/ で始まる絶対パス名でなければなりません。*-z* オプションを指定せず、デフォルト値も設定されていない場合は、`FSCONFIG` 環境変数に指定されているパスが使用されます。*-o offset* オプションを使用すると、*config* のオフセットを指定できます。*-o* オプションを指定せず、デフォルト値も設定されていない場合は、環境変数 `FSOFFSET` の値が使用されます。`FSOFFSET` が設定されていない場合、デフォルトで 0 になります。

`loadtlog -m machine filename`

`dumpltlog` によって生成された TLOG のテキスト・バージョンを、指定されたファイル *filename* から指定されたまたは `default` の *machine* (“DBBL” または `&dl“&dr”` 定不可) の既存の TLOG に読み込みます。

`logstart machine`

指定マシン *machine* 上で TLOG 情報のウォーム・スタートを強制実行します。このコマンドは通常、`loadtlog` の実行後、サーバ・グループの移行の際のディスク再配置に続いて行うようにしてください。

`master (m) [-yes]`

分断時に `master` がバックアップ・ノードで動作していると、このバックアップ・ノードはマスタ・ノードに変わり、DBBL が起動されて管理プロセスを引き継ぎます。バックアップ・ノードがマスタとして動作しているときに、`master` をマスタ・ノード上で実行すると、その DBBL はマスタ・ノードに移行し、バックアップ・ノードはその時点でマスタ・ノードとしての働きを停止します。`-yes` オプションを指定せずにこのコマンドを実行すると、処理を進める前に確認を求められます。

`migrategroup (migg) [-cancel] group_name`

`migrategroup` コマンドはサーバ・グループ名を引数とします。コンフィギュレーション・ファイルに `MIGRATE` オプションとそのグループの代替位置の指定があれば、*group_name* に属するすべてのサーバは代替位置に移行します。移行の際には、次のコマンドを実行してサーバをシャットダウンする必要があります。

```
shutdown -R -g groupname
```

`-R` オプションは、移行できるように掲示板のサーバ名を保持するために使用されます。`shutdown -R` の後、移行を取り消す場合には、次のコマンドを使用します。

```
migrategroup -cancel groupname
```

`-cancel` オプションは、掲示板からサーバ名を削除するときに使用します。

`migratemach (migm) [-cancel] machine`

指定マシン *machine* 上で動作しているすべてのサーバが代替位置に移行します。移行の際には、次のコマンドを実行してサーバをシャットダウンする必要があります。

```
shutdown -R -l machine
```

`migratemachine` コマンドを使用する場合には、*machine* 上のすべてのサーバ・グループは同じ代替位置を持つ必要があります。そうでない

場合は `migrategroup` を使用してください。Domains ゲートウェイ・サーバを含む LMID の移行は、これらのゲートウェイ・サーバを代替 LMID に移行することを意味します。-cancel オプションを指定すると、進行中の移行が取り消されます。つまり、`tmshutdown -R` コマンドによってサーバはシャットダウンされていますが、移行されていない状態です。

`paginate (page) [{off | on}]`

出力をページ別表示します。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。標準入力または標準出力のどちらかが非端末デバイスでない限り、初期設定は on です。tty 標準入力と標準出力が共に端末デバイスの場合にのみ on になります。

デフォルトのページ別表示コマンドは、ネイティブのオペレーティング・システム環境によって異なります。たとえば、UNIX オペレーティングシステム環境では、`pg` がデフォルトのコマンドです。シェル環境変数 `PAGER` を使用すると、ページ別表示出力に使用されるデフォルトのコマンドを無効化することができます。

`passwd`

アプリケーションのセキュリティを確保するため、管理者にアプリケーション・パスワードの入力を求めます。

`pclean (pcl) machine`

`pclean` を使用すると、まず、指定の `machine` に対して強制的に `bbclean` を行い、再起動またはクリーンアップの必要なサーバすべてを再起動またはクリーンアップします。`machine` が分断されている場合は、マシン `machine` 上で稼動するよう指定されているプロセスとサービスに対応するエントリが、非分断の掲示板からすべて削除されます。`machine` が分断されていない場合には再起動またはクリーンアップできないプロセスまたはサービスはすべて削除されます。

`printclient (pclt) [-m machine] [-u username] [-c cltname]`

指定された一連のクライアント・プロセスに関する情報を表示します。引数が指定されておらず、デフォルトも定義されていない場合は、すべてのクライアントに関する情報が出力されます。マルチコンテキスト・クライアントで `printclient` を実行すると、各コンテキストについて個別にエントリが生成されます。-m、-u、および -c オプションやデフォルト値の設定を使用して、マシン、ユーザ名、または

クライアント名の任意の組み合わせに限定して情報を得ることができません。

`printconn (pc) [-m machine]`

指定マシンが関与する会話型接続に関する情報を表示します。-m オプションやデフォルト値を使用することにより、情報を指定したマシンに対する接続に関するものに限定することができません。*machine* の値に `&"ll&"` または `&dlq"&drq"` すると、すべてのマシンの情報が出力されます。

`printactiveobject (pao) [-B objectid] [-m machine]`

ドメイン内でアクティブなオブジェクトに関する情報を表示します。オブジェクト ID、インターフェイス名、サービス名、プログラム名、グループ ID、プロセス ID、リファレンス・カウントなどが出力されます。このコマンドでは、オブジェクト ID およびマシン ID はオプション・パラメータです。オブジェクト ID が指定されていない場合、すべてのアクティブなオブジェクトの情報が出力されます。マシン ID が指定されていない場合、コマンドを発行したマシン上でアクティブなすべてのオブジェクトの情報が出力されます。128 文字を超えるオブジェクト ID は、40 文字の英数字から成るハッシュ値として表示されます。

`printfactory (pf)`

ファクトリ・ファインダに登録されているオブジェクト・ファクトリに関する情報を出力します。インターフェイス名、インターフェイスのファクトリ識別子、現在のファクトリ状態の属性などが出力されます。このコマンドには引数はありません。このコマンドは、CORBA 環境でのみ使用されます。

`printgroup (pg) [-m machine] [-g groupname]`

サーバ・グループ・テーブル情報を表示します。デフォルトでは、すべてのグループの情報を表示します。-g および -m オプションやデフォルト値を使用して、特定のグループやマシンに関する情報を出力できます。サーバ・グループ名、サーバ・グループ番号、一次 / 代替 LMID、および現在の位置などの情報が得られます。

`printinterface (pif) [-m machine] [-g groupname] [-I interface]`

指定されたオブジェクト・インターフェイスに関する情報を出力します。インターフェイス名、キュー名、グループ ID、マシン ID、ルーティング名、インターフェイスから送られた要求の数などが出力され

ます。このコマンドでは、マシン名、グループ名、およびインターフェイス名はオプション・パラメータです。マシン名を指定すると、インターフェイスに対してアクティブなオブジェクトの数が出力されます。マシン名が指定されていない場合は、アクティブなオブジェクトに関する情報を取得できないことを示すハイフン (-) が出力されます。このコマンドは、CORBA 環境でのみ使用されます。

`printnet (pnw) [mach_list]`

ネットワーク接続に関する情報を出力します。デフォルトでは、すべてのマシンの情報を表示します。`printnet` コマンドでは、複数のマシン (LMID) をカンマで区切ったリストを引数として指定できます。このようなりストを指定した場合は、指定マシンが関係するネットワーク接続情報が表示されます。各マシンについて、分断されているかどうかが表示されます。分断されていない場合、そのマシンが接続される LMID とメッセージの送受信回数を表す情報が表示されます。

`printqueue (pq) [qaddress]`

アプリケーションおよび管理サーバの指定キューに関する情報を表示します。デフォルトでは、すべてのキューに関する情報が表示されます。`qaddress` コマンド行またはデフォルト値は指定したキューに関する情報を制限するために使用されます。出力される情報には、キューが存在するサーバ名、マシンの名前があります。

`printroute (pr) [-r routingname]`

ファクトリ・ベースのルーティング定義に関する情報を出力します。ルーティング名、ルーティング・タイプ、ルーティング・フィールド、ルーティング範囲などが出力されます。`routingname` が指定されていない場合は、既存のすべてのルートが表示されます。このコマンドは、BEA Tuxedo のデータ依存型ルーティングおよび CORBA のファクトリ・ベース・ルーティングの両方のルートを出力します。ルーティング・タイプの出力フィールドでは、ファクトリ・ベース・ルーティングのエントリは `FACTORY` に、データ依存型ルーティングのエントリは `SERVER` に表示されます。冗長モードで、データ依存型ルーティングのエントリに関する情報が要求された場合には、バッファ・タイプとフィールド・タイプも出力されます。このコマンドは、CORBA 環境でのみ使用されます。

`printserver (psr) [-m machine] [-g groupname] [-i srvid] [-q qaddress]`

アプリケーションおよび管理サーバに関する情報を表示します。`-q`、`-m`、`-g` および `-i` オプションを使用すれば、キュー、マシン、グルー

プあるいはサーバの任意の組み合わせに限定して情報を得ることができます。マルチコンテキスト・サーバで `printserver` を実行すると、そのサーバのすべてのコンテキストについて 1 つのエントリが表示されます。

```
printservice (psc) [-m machine] [-g groupname] [-i srvid] [-a { 0 | 1 | 2 }]
[-q qaddress] [-s service]
```

アプリケーションおよび管理サービスに関する情報を表示します。
`-q`、`-m`、`-g`、`-i` および `-s` オプションを使用することで、キュー、マシン、グループ、サーバ、またはサーバの任意の組み合わせに限定して情報を得ることができます。`-a` オプションを使用すると、サービスのクラスを選択できます。`-a0` と指定すると、アプリケーション・サービスだけを表示します。`-a1` と指定すると、アプリケーション・サービスに加えて、アプリケーションから呼び出せるシステム・サービスも表示します。`-a2` と指定すると、上記の 2 つに加えて、BEA Tuxedo から呼び出せるシステム・サービスも表示します。

```
printtrans (pt) [-g groupname] [-m machine]
```

指定マシンまたはデフォルトのマシンに関するグローバル・トランザクション・テーブル情報を表示します。`machine` が `&"11"` または `&d"BL"` の場合、アプリケーションのすべての非分断マシンすべてのトランザクション・テーブルからの情報がマージされます。コマンド行またはデフォルトの `groupname` 値によって、指定グループが参加している (コーディネータを含む) トランザクションに限定して情報を得ることができます。

簡略モードでは、トランザクション識別子、`aborttrans` または `committrans` でトランザクションをアボートまたはコミットするとき使用するインデックス、トランザクション状態、参加プロセスの数などが表示されます。

一方、冗長モードでは、トランザクション・タイムアウト情報や、コーディネータの識別情報などを含むサーバ・グループ名、ステータスなどの参加プロセスに関する情報も表示されます。

```
quit (q)
```

セッションを終了します。

```
reconnect (rco) non-partitioned_machine1 partitioned_machine2
```

非分断マシンから分断マシンへの新たな接続を行います。reconnect は、非分断マシンから分断マシンに新たな接続を強制します。接続が既にアクティブであれば、再接続する前にクローズされます。この場合、処理中のメッセージが失われて、トランザクションがタイムアウトすることがあります。マシンやネットワーク接続をダウンすることも可能ですが、ネットワーク・インターフェイス・ドライバは、BRIDGE に対しエラーを通知することなく、要求を受け付け、バッファに格納します。この場合、再接続は失敗し、リモート・マシンがアクセスできないことを BRIDGE が認識するようにします。ほとんどの場合、ネットワーク障害が解決すれば BRIDGE は自動的に再接続を行い、reconnect による手作業での再接続は必要ありません。

```
resume (res) {-q qaddress | -g groupname | -i srvid | -s service | -I interface} ...
```

指定されたサービスを再開 (中断解除) します。-q、-g、-s、-I、および -i オプションは、キュー、グループ、サービス、インターフェイス (CORBA 環境でのみ)、およびサーバ (少なくともこれらのオプションの 1 つが指定されるかデフォルトの値を持たなければなりません) の組み合わせにより再開されたサービスを制限するために使用されます。したがって、次のコマンド行を使用すると、アドレス servq8 のキューに宣言されているすべてのサービスを簡単に再開できます。

```
> resume -q servq8
```

中断されていたサービスが再開されると、そのサービスを提供するサーバが、そのサービス (および提供するその他の再開サービス) の候補サーバとして選択されます。複数のサーバが 1 つのキューから読み取りを行うと、そのキューから読み取りを行うすべてのサーバに、ある特定のサービスの状態が反映されます。

```
serverparms (srp) -g groupname -i srvid
```

あるグループの *groupname* および *srvid* で指定されたサーバに関連するパラメータを表示します。

```
serviceparms (scp) -g groupname -i srvid -s service
```

groupname、*srvid*、および *service* によって指定されたサービスに関連するパラメータを表示します。

`shmstats (sstats) [ex | app]`

コンフィギュレーション・ファイルで `MODEL SHM` を指定すると、`shmstats` を使用してより正確な統計情報を得ることができます。引数なしで入力すると、`shmstats` は掲示板構造体内の `bbparms.options` メンバの `TMACCSTATS` フラグの現在の設定を返します。これにより、収集している統計情報が正確なものか、大まかなものかが判別できません。`ex` オプションを指定すると、`shmstats` は `TMACCSTATS` フラグをオンにして掲示板をロックします。またサーバ・テーブル、キュー・テーブル、およびサービス・テーブル・エントリのカウンタをゼロにリセットします。

`shutdown (stop) [options]`

このコマンドは、`tmshutdown()` コマンドと同様です。`tmshutdown` のオプションを使用すると、停止させるサーバを選択することができます。オプションの説明および使用上の制限事項については、[tmshutdown\(1\)](#) を参照してください。

`suspend (suspend) { -q qaddress | -g groupname | -i srvid | -s service | -I interface } ...`

サービスを中断します。`-q`、`-g`、`-s`、`-I`、および `-i` オプションを使用すれば、キュー、グループ、サービス、インターフェイス (CORBA 環境のみ)、またはサーバの任意の組み合わせに限定してサービスを中断できます。ただし、これらのオプションを少なくとも1つ指定するか、オプションにデフォルト値が含まれている必要があります。したがって、次のコマンドを使用すれば、アドレス `servq8` のキューに宣言されているすべてのサービスを簡単に中断できます。

```
> suspend -q servq8
```

サービスが中断されると、そのサービスを提供するサーバはそのサービスの候補サーバとしては選択されなくなります。ただし、このサーバは、それが提供するほかのサービスについては引き続き選択されることができます。中断されたサービスに関してキューに登録されている要求は、そのキューの内容がなくなるまで処理されます。複数のサーバが1つのキューから読み取りを行うと、そのキューから読み取りを行うすべてのサーバに、ある特定のサービスの状態が反映されません。

```
unadvertise (unadv) {-q qaddress [-g groupname] [-i srvid] | -g groupname
-i srvid} service
```

指定された *service* についてサービス・テーブルのエントリを除去します。*qaddress* が指定されていない場合は、*groupname* と *srvid* を指定し、サーバを一意に識別する必要があります。キューまたはそのキューにある特定のサーバを指定しても結果は同じです。複数サーバの単一キュー (MSSQ) のセットからこの *service* を削除すると、*service* の宣言は、キューから読み込まれるすべてのサーバから削除されます。

```
verbose (v) [{off | on}]
```

冗長モードで出力を行います。オプションが指定されていない場合は、設定可能な状態に現在の設定が切り替えられて、新しい設定が表示されます。初期設定は *off* です。各のコマンドで *-v* および *-t* を使用すると、一時的に現在の設定を無効にできます。

```
! shellcommand
```

シェルにエスケープし、*shellcommand* を実行します。

```
!!
```

直前のシェル・コマンドを再実行します。

```
# [text]
```

"#" で始まる行はコメント行で、無視されます。

```
<CR>
```

最後のコマンドを繰り返します。

セキュリティ 管理者として *tmadmin* を実行する場合、セキュリティ機構は通りません。既にアプリケーション管理者のログイン ID であることが確認済みであるためです。

tmadmin をアプリケーション管理者以外のユーザとして実行できる唯一のケースは、そのアプリケーションをクライアントとしてアクセスするよう *-r* オプションが使用されている場合です。このようなユーザが *-r* オプションを指定して *tmadmin* を呼び出した場合、そのアプリケーションに対してセキュリティ機構が働き、アプリケーションのデータにアクセスするにはそのアプリケーションのパスワードが必要になります。標準入力端末である場合、*tmadmin* はユーザにパスワードの入力を求めてきます。なお、パスワードの入力時には、その文字は画面上には表示されません。標準入力端末

末でない場合には、このパスワードは環境変数 `APP_PW` から取得されます。この環境変数の指定がなく、アプリケーション・パスワードが必要な場合には、`tmadmin` は異常終了します。

環境変数 `tmadmin` は、`-r` オプションの指定がある場合、またはこれをアプリケーション管理者として登録できない場合、アプリケーション・クライアントとして働きます。このような場合、標準入力端末でないときには、保護アプリケーションのアプリケーション・パスワードを `APP_PW` に設定しなければなりません。

診断 システムのブート前に `tmadmin` コマンドを入力すると、次のようなメッセージが表示されます。

```
No bulletin board exists. Entering boot mode
```

```
>
```

この後、`tmadmin` は `boot` コマンドの入力を待ちます。

`-c` オプションを指定せずに、`MASTER` 以外の非アクティブなノードから `tmadmin` コマンドが入力されると、次のメッセージが表示され、コマンドは終了します。

```
Cannot enter boot mode on non-master node.
```

間違ったアプリケーション・パスワードが入力されたり、入力したパスワードが環境を通してシェル・スクリプトで利用できない場合には、ログ・メッセージが生成され、次のメッセージが表示されてコマンドが終了します。

```
Invalid password entered.
```

相互運用性 `tmadmin` は、相互操作可能なアクティブ・アプリケーション内であれば任意のノード上で実行可能です。ただし、利用可能なコマンドやコマンド行引数は、`tmadmin` が動作しているノードに対応するリリースで `tmadmin` から利用できるものに限定されています。たとえば、`broadcast`、`passwd`、および `printclient` などのコマンドはリリース 4.1 のノードでは利用できません。

移植性 `tmadmin` は、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。

注意事項 *machine* オプションは、ネットワーク化されていない単一プロセッサ環境では何の効果も得られません。

関連項目 [tmboot\(1\)](#)、[tmloadcf\(1\)](#)、[tmshutdown\(1\)](#)、[compilation\(5\)](#)、[UBBCONFIG\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

tmboot(1)

名前	tmboot - BEA Tuxedo コンフィギュレーションを起動
形式	tmboot [-l <i>lmid</i>] [-g <i>grpname</i>] [-i <i>srvid</i>] [-s <i>aout</i>] [-o <i>sequence</i>] [-S] [-A] [-b] [-B <i>lmid</i>] [-T <i>grpname</i>] [-e <i>command</i>] [-w] [-y] [-g] [-n] [-c] [-M] [-d1]
機能説明	<p>tmboot は、オプションの指定に従って BEA Tuxedo アプリケーション全体あるいはその一部を起動します。tmboot は、掲示板の管理者 (コンフィギュレーション・ファイルの UID で指示されている) あるいは root しか呼び出せません。また、tmboot コマンドは、コンフィギュレーション・ファイルの RESOURCES セクションで MASTER として指定されているマシン、あるいは MASTER として機能するバックアップ・マシン (つまり、tmadmin(1) の master コマンドにより DBBL がすでに稼働状態である) 上でしか呼び出せません。ただし、これは -b オプションを指定した場合を除きます。この場合は、MASTER と指定しなくても、バックアップ・マシンからシステムを起動できます。</p> <p>オプションなしの場合、tmboot はすべての管理プロセスと環境変数 TUXCONFIG と TUXOFFSET に指定されているコンフィギュレーション・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。MODEL が MP の場合、RESOURCES セクションの MASTER パラメータで指定されたマシン上で DBBL 管理サーバが起動します。管理サーバ (BBL) は、MACHINES セクションにリストされている各マシン上で起動されます。GROUPS セクションに記述されている各グループごとに、TMS サーバは各エントリの TMSNAME および TMSCOUNT パラメータに基づいて起動されます。また、すべての管理サーバが起動した後、SERVERS セクションに記述されているサーバが起動します。あるグループに属する TMS あるいはゲートウェイがブートした後、そのグループの最初のアプリケーション・サーバがブートします。必要に応じて、TUXCONFIG ファイルが遠隔マシンに伝搬されます。tmboot は通常、ブートされたプロセスがその初期化を完了するまで (つまり、tpsvrinit() が完了するまで)、次のプロセスのブートには移りません。</p> <p>ゲートウェイ・サーバの起動時には、ゲートウェイの管理サービスが宣言され、そのゲートウェイの CLOPT パラメータを基に外部サービスを表すアプリケーション・サービスが宣言されます。外部サーバの考え方をとる場合、これらのサーバはこの時点でゲートウェイによりブートされます。</p>

LMID をブートすると、その LMID 上の全グループがブートされることになります。

アプリケーション・サーバは、SEQUENCE パラメータで指定された順序、あるいはコンフィギュレーション・ファイル内のサーバ・エントリの記述順にブートされます (UBBCONFIG(5) の説明を参照)。コンフィギュレーション・ファイルの SERVERS セクションに記述されている複数のサーバが同じ SEQUENCE パラメータを持つ場合、tmboot はこれらのサーバを同時にブートし、これらのすべての初期化が完了するまで次の処理には進みません。SERVERS セクションの各エントリは MIN と MAX の 2 つのパラメータをもつことがあります。tmboot は、-i オプションが指定されていない場合は、MIN 個のアプリケーション・サーバをブートします (サーバ・エントリに MIN の指定がなければ、デフォルトの値として 1 が使用されます)。ただし、-i オプションを指定した場合には、個々のサーバを MAX で指定された数だけ個別にブートすることができます。

サーバを起動できなかった場合には、中央イベント・ログに診断メッセージが書き込まれます。-q が指定されていない場合、診断メッセージは標準出力にも出力されます。この後 tmboot は処理を継続します。ただし、異常終了したプロセスが BBL の場合には、その BBL に依存しているサーバは自動的に無視されます。また、異常終了したプロセスが DBBL であると、tmboot はコンフィギュレーション・ファイルに記述されている残りのサーバについては無視します。サーバが代替 LMID をコンフィギュアしており、一次マシンを起動できない場合、tmboot は代替マシンでサーバを起動します。起動に成功した場合は、DBBL にメッセージを送信して TUXCONFIG のサーバ・グループのセクションを更新します。

SERVERS セクションに記述されたサーバの場合、tmboot は CLOPT、SEQUENCE、SRVGRP、および SRVID だけを使用します。これらを総称して、サーバのブート・パラメータと言います。サーバはブートすると、コンフィギュレーション・ファイルの内容を読み取り、その実行時パラメータを見付けます (パラメータの詳細については、UBBCONFIG(5) を参照してください)。

管理サーバおよびアプリケーション・サーバはすべて、APPDIR を現在の作業ディレクトリとしてブートされます。APPDIR の値は、該当サーバがブートされるマシンに対応するコンフィギュレーション・ファイル内の MACHINES セクションで指定します。

サーバの実行可能プログラムの検索パスは `APPDIR`、`TUXDIR/bin`、`/bin`、`/usr/bin`、および `MACHINE` の `ENVFILE` に指定されている `PATH` です。この検索パスは、サーバの絶対パス名が指定されていない場合にのみ使用されます。この検索パスの設定は、サーバの `ENVFILE` に指定する値によって変更することはできません。

サーバがブートすると、変数 `TUXDIR`、`TUXCONFIG`、`TUXOFFSET` および `APPDIR` は、そのマシンに対してコンフィギュレーション・ファイルに指定されている値を持つように、環境に設定されます。また、変数 `LD_LIBRARY_PATH` がすべてのサーバ環境に設定されます。デフォルト値は

`$APPDIR:$TUXDIR/lib:/lib:/usr/lib:lib` で、`lib` はマシンの `ENVFILE` での最初の `LD_LIBRARY_PATH=` 行の値です。`ENVFILE` の構文の詳細については、[UBBCONFIG\(5\)](#) を参照してください。一部の UNIX システムでは、別の環境変数が必要となる場合もあります。HP-UX システムでは `SHLIB_PATH` 環境変数を使用します。AIX システムでは `LIBPATH` 環境変数を使用します。

サーバの `ULOGPFX` もまた、ブート時に、コンフィギュレーション・ファイルの該当マシンに対するパラメータに基づいて設定されます。パラメータの指定がないと、デフォルトの値として `$APPDIR/ULOG` が使用されます。

これらの処理はすべて、アプリケーションの初期化関数 `tpsvrinit()` が呼び出される前に行われます。

`tmbboot` のコマンド行オプションの多くは、システムのブート時の内容を制限するものであり、システムの一部のブートに使用することもできます。以下のオプションがサポートされています。

`-l lmid`

`LMID` パラメータが `lmid` である各グループの場合、そのグループに関連するすべての `TMS` とゲートウェイ・サーバがブートされ、そのグループに関連して `SERVERS` セクションに記述されているすべてのサーバが実行されます。

`-g grpname`

`SRVGRP` パラメータが `grpname` であるグループの全 `TMS` およびゲートウェイ・サーバが起動され、続いてそのグループに関連して `SERVERS` セクションに記述されているすべてのサーバが起動されます。`TMS` サーバは、グループ・エントリの `TMSNAME` および `TMSCOUNT` パラメータに基づいて起動されます。

- i *srvid*
SERVERS セクションに記述されているサーバで、SRVID パラメータが *srvid* である全サーバが起動されます。
- s *aout*
SERVERS セクションのサーバで、名前が *aout* である全サーバが実行されます。このオプションは、TMS およびゲートウェイ・サーバをブートするときに使用することもできます。通常、このオプションは -g オプションと一緒に使用します。
- o *sequence*
SERVERS セクションのサーバで、SEQUENCE パラメータが *sequence* であるサーバがすべて実行されます。
- S
SERVERS セクションに記述されている全サーバが実行されます。
- A
MACHINES セクションに記述されているマシンの全管理サーバが実行されます。このオプションは、DBBL とすべての BBL と BRIDGE プロセスが正しい順序で起動されるようにするときに使用します (-M オプションの項も参照)。
- b
BACKUP のマシンからシステムを起動します (BACKUP のマシンを MASTER にする必要はありません)。
- B *lmid*
論理名が *lmid* であるプロセッサ上で BBL が起動されます。
- M
このオプションはマスタ・マシン上の管理サーバを起動するときに使用します。MODEL が MP の場合、RESOURCES セクションの MASTER パラメータで指定されたマシン上で DBBL 管理サーバが起動します。また、BBL が MASTER マシン上で起動され、LAN オプションと NETWORK エントリがコンフィギュレーション・ファイルに指定されている場合には、BRIDGE も起動されます。
- dl
コマンド行オプションを標準出力に出力します。sdb を使用してアプリケーション・サービスのデバッグを行うのに便利です。

-T *grpname*

SRVGRP パラメータが *grpname* であるグループ内の全 TMS サーバが起動されます (そのグループに関連する TMSNAME および TMSCOUNT パラメータに基づく)。このオプションによる TMS の起動は、TMS サーバ名 (-s オプション) およびグループ名 (-g) に基づいてブートする場合と同じです。

-e *command*

プロセスがブートに失敗した場合、*command* が実行されます。*command* は、SHELL 環境変数で指定されたコマンド・インタプリタが解釈できるものであれば、任意のプログラム、スクリプト、または一連のコマンドでかまいません。これにより、ブート手順で大きな障害が発生する前に処理を打ち切ることができます。*command* に空白を含める場合には、その文字列全体を引用符で囲む必要があります。このコマンドは、サーバをブートしようとするマシン上ではなく、*tmboot* が実行されるマシン上で実行します。

注記 Windows 2000 システムでリダイレクトまたはパイプを選択する場合は、以下のいずれかの方法を使用してください。

- リダイレクトまたはパイプを、コマンド・ファイルまたはスクリプト内から行います。
- キュー・マネージャ管理プログラムからリダイレクトするには、コマンドの前に *cmd* を付けます。次に例を示します。

```
cmd /c ipconfig > out.txt
```
- バイナリ実行可能ファイルの作成を選択する場合は、Windows API 関数の `AllocConsole()` を使用して、バイナリ実行可能ファイル内にコンソールを割り当てます。

-w

先に初期化段階に入ったサーバの処理の完了を待たずに、次のサーバを起動するように *tmboot* に通知します。このオプションの使用には注意が必要です。BBL は有効な DBBL が存在することを前提としており、通常のサーバはそれらが置かれているプロセッサ上で BBL が稼働していることを必要とします。これらの条件は、サーバが同期的に起動されなければ保証できません。このオプションを使用すると、通常はサーバが順に起動されるときにとられる待ち時間がなくなります。

- y
すべての管理プロセスおよびサーバ・プロセスをブートすべきかどうかを確認するプロンプトに対して `yes` の応答を想定します (このプロンプトは、制限を課すオプションを一切つけずにコマンドが入力されたときにのみ表示されます)。
- q
標準出力に実行シーケンスを表示しません。-y が指定されているものと想定します。
- n
実行シーケンスを表示しますが、実行はされません。
- c
この構成で必要とされる最小限の IPC リソースを表示します。
- l、-g、-i、-o、-s の各オプションを組み合わせで使用した場合、指定されたすべての条件を満たすサーバだけがブートされます。-l、-g、-s、-T の各オプションを使用すると、TMS サーバがブートされます。また、-l、-g、-s はそれぞれ、ゲートウェイ・サーバをブートします。-l、-g、-i、-o、-s、-s の各オプションはアプリケーション・サーバに適用されます。BBL が該当マシンに存在しないと、アプリケーション・サーバをブートするオプションを指定しても、ブートできません。-A、-M および -B オプションは、管理プロセスにのみ適用されます。

標準入力、標準出力、標準エラー出力の各ファイル記述子は、ブートされたサーバについてはすべてクローズされます。

- 相互運用性 `tmboot` は、相互運用可能なアプリケーション内の最上位のリリースが提供されているマスタ・ノード上で実行しなければなりません。`tmboot` は、最新リリースをサポートしないサイトの /WS リスナなどの管理サーバのブートを行うことにつながるコンフィギュレーション・ファイル内の条件を検出し報告します。
- 移植性 `tmboot` は、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。

環境変数	インストールの間に管理パスワード・ファイルが作成されます。必要に応じて、BEA Tuxedo システムはこのファイルを次のディレクトリから検索します。検索順序は次のとおりです。APPDIR/.adm/tlisten.pw、TUXDIR/udataobj/tlisten.pw。パスワード・ファイルが見付かるように、環境変数 APPDIR または TUXDIR またはその両方を必ず設定してください。
リンク・レベルの暗号化	リンク・レベルの暗号化機能が <code>tmboot</code> と <code>tlisten</code> の間で稼動している場合には、リンク・レベル暗号化はメッセージが認証されるプロセスを保護するために最初に調整され、また活性化されます。
診断	<p>TUXCONFIG として、存在しないファイルが設定されると、次に示す 2 つの致命的なエラー・メッセージが表示されます。</p> <ul style="list-style-type: none">■ error processing configuration file■ configuration file not found <p><code>tmboot</code> はサーバのブートに失敗すると、終了コード 1 を返して終了します。後でユーザ・ログを調べるにより、エラーの詳細を知ることができます。成功した場合は、<code>tmboot</code> は終了コード 0 を返して終了します。</p> <p><code>tmboot</code> が、アクティブでない非マスタ・ノード上で実行されると、次のような致命的エラー・メッセージが表示されます。</p> <pre>tmboot cannot run on a non-master node.</pre> <p><code>tmboot</code> が、実際のマスタ・ノードでないアクティブなノード上で実行されると、次のような致命的エラー・メッセージが表示されます。</p> <pre>tmboot cannot run on a non acting-master node in an active application.</pre> <p>複数の TUXCONFIG ファイルで同じ IPCKEY が使用されていると、<code>tmboot</code> は異常終了して次のメッセージを表示します。</p> <pre>Configuration file parameter has been changed since last tmboot</pre> <p>LAN 以外の構成で MACHINES セクションに複数のノード名があると、次のような致命的エラー・メッセージが表示されます。</p> <pre>Multiple nodes not allowed in MACHINES for non-LAN application.</pre>

tlisten が LAN アプリケーションの MASTER マシン上で実行していない場合、警告メッセージが表示されます。この場合 `tmadmin(1)` は、リモート・マシン上で管理モードで実行することはできません。読み取り専用の操作のみに制限されます。これは、バックアップ・サイトが障害の発生後にマスター・サイトをリポートできないことを意味します。

使用例 CS0 および CS1 というローカル名のマシン上にあるサーバだけを起動するには、次のようにします。

```
tmboot -l CS0 -l CS1
```

DBG1 という名前のグループに属するサーバ CREDEB だけを起動するには、次のようにします。

```
tmboot -g DBG1 -s CREDEB1
```

ローカル名が PE8 のマシン上の BBL、および PE8 の位置にあるすべてのサーバをブートするには、次のようにします。

```
tmboot -B PE8 -l PE8
```

その構成に必要とされる最小限の IPC リソースを表示するには、次のようにします。

```
tmboot -c
```

次に、`-c` オプションを使用した場合の出力例を示します。

```

Ipc sizing (minimum BEA Tuxedo values only) ...
                               Fixed Minimums Per Processor
SHMMIN:1
SHMALL:1
SEMAP:  SEMMNI

```

Node	Variable Minimums Per Processor						
	SEMUME, SEMMNU, SEMMNS	SEMMSL	A *	SEMMNI	MSGMNI	MSGMAP	SHMMAX *
-----	-----	-----	-----	-----	-----	-----	-----
sfpup	60	1	60	A + 1	10	20	76K
sfsup	63	5	63	A + 1	11	22	76K

where 1 = A = 8.

1 プロセッサ当たりのアプリケーション・クライアントの数を各 `MSGMNI` 値に追加します。`MSGMAP` は `MSGMNI` の 2 倍にします。`SHMMIN` は常に 1 にセットします。

IPC の必要最小限の条件は、使用マシンに設定したパラメータと比べて検討することができます。これらのパラメータの変更方法については、お使いのマシンのシステム管理者マニュアルを参照してください。-y オプションを使用した場合、その結果得られる表示は上記の例とは若干異なります。

注意事項 `tmboot` コマンドは、ハングアップ・シグナル (SIGHUP) を無視します。ブート時にシグナルが検出された場合でも、プロセスは続行します。

-c オプションを使用した場合に表示される最小限の IPC リソースは、指定されたコンフィギュレーション・ファイルに記述されている構成にのみ適用されます。リソース・マネージャまたはその他の BEA Tuxedo 構成で必要となる IPC リソースについては、計算上は考慮されません。

関連項目 [tmadmin\(1\)](#)、[tmloadcf\(1\)](#)、[tmshutdown\(1\)](#)、[UBBCONFIG\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

tmconfig、wtmconfig(1)

名前 tmconfig、wtmconfig - ネイティブ・クライアントまたはワークステーション・クライアントとして実行中の BEA Tuxedo アプリケーションに関する情報を動的に更新および検索

形式 tmconfig
wtmconfig

機能説明 tmconfig は対話型プログラムであり、BEA Tuxedo アプリケーションの動作中に、コンフィギュレーション・ファイル・パラメータまたは MIB 属性の更新、および TUXCONFIG セクション部へのレコードの追加を行います。tmconfig は、追加、変更、または検索対象の入力フィールド値があるバッファを管理します。操作が完了すると、tmconfig は出力フィールド値と状態を表示します。ユーザは、利用可能なテキスト・エディタを使用して入力バッファを更新できます。

tmconfig は BEA Tuxedo ネイティブ・クライアント、wtmconfig はワークステーション・クライアントです。これについては、tmadmin/printclient コマンド・シーケンスの出力で確認できます。アプリケーションが SECURITY 機能を使用している場合、tmconfig は起動時にアプリケーション・パスワードの入力を求めるプロンプトを表示します。

注記 tmconfig は BEA Tuxedo ネイティブ・クライアントであり、wtmconfig は BEA Tuxedo ワークステーション・クライアントであるという以外に双方の違いはなく、wtmconfig と tmconfig は共に同じ機能を備えているため、このリファレンス・ページでは tmconfig を中心に説明していきます。ここで説明する機能は wtmconfig も備えていると考えてかまいません。

tmconfig は最初に、目的のセクションを指定するよう求め、続いてそれに対する操作の内容を指定するよう求めます。セクションを求めるプロンプトは次のとおりです。

```
Section: 1) RESOURCES, 2) MACHINES, 3) GROUPS 4) SERVERS 5) SERVICES
6) NETWORK 7) ROUTING q) QUIT 9) WSL 10) NETGROUPS 11) NETMAPS
12) INTERFACES [1]:
```

プロンプト末尾の大括弧の中にはデフォルトのセクションが表示されます。

続いて、目的の操作内容を求めるプロンプトが次のように表示されます。

Operation: 1) FIRST 2) NEXT 3) RETRIEVE 4) ADD 5) UPDATE
6) CLEAR BUFFER 7) QUIT [1]:

プロンプト末尾にデフォルトの操作内容が表示されます。デフォルトの操作を選択する場合は、Return キーを押します。ほかの操作を選択する場合は、該当する番号を入力して Return キーを押します。

現在、以下の操作がサポートされています。

1. **FIRST** - 指定したセクションの先頭レコードを取り出します。キー・フィールドは必要ありません入力バッファ内にある場合、キー・フィールドは無視されます。
2. **NEXT** - 入力バッファ内のキー・フィールドに基づいて、指定したセクションから次のレコードを取り出します。
3. **RETRIEVE** - キー・フィールドで指定したセクションから、指定したレコードを取り出します。
4. **ADD** - 指定したセクションに、指定したレコードを追加します。指定されていないフィールド(必須でない限り)があれば、[UBBCONFIG\(5\)](#) に指定されているデフォルトの値が使用されます。すべてのフィールドの現在の値は、出力バッファ内に返されます。この操作は、BEA Tuxedo システム管理者しか実行できません。
5. **UPDATE** - 指定したセクションの入力バッファ内にある、指定したレコードを更新します。入力バッファ内で指定されていないフィールドは変更されません。すべてのフィールドの現在の値は、入力バッファ内に返されます。この操作は、BEA Tuxedo 管理者しか実行できません。
6. **CLEAR BUFFER** - 入力バッファをクリアします。すべてのフィールドが削除されます。この操作の後、`tmconfig` は直ちにセクションの指定を求めるプロンプトを出します。
7. **QUIT** - プログラムを正常に終了します。クライアントは終了します。プロンプトで値 `q` を入力してもプログラムを終了できます。

管理者の操作を行う場合、実効ユーザ識別子が、このプログラムを実行するマシンの BEA Tuxedo 管理者ユーザ識別子 (UID) と同じでなければなりません。レコードの更新や追加を行う場合には、すべてのデフォルトの値と [tmloadcf\(1\)](#) が使用する妥当性検査機能が有効になります。

次に `tmconfig` は、入力バッファを編集するかどうかを確認するプロンプトを表示します。

```
Enter editor to add/modify fields [n]?
```

ここで、`y` を入力すると、入力バッファは一時ファイルにコピーされ、テキスト・エディタが起動されます。どのエディタを使用するかは、環境変数 `EDITOR` によって決めることができます。デフォルト値は `ed` です。入力時の形式は「フィールド名」/「フィールド値」です。これについては、以下の「入力形式」の節で説明します。各 `UBBCONFIG` セクションに対応するフィールド名を、以下のサブセクションの表に示します。フィールドの意味とそれらに対応する指定範囲、デフォルト値、制約事項などについては、[UBBCONFIG\(5\)](#) に説明があります。許容値は、8 進数ではなく 10 進数で指定します。ほとんどの場合、フィールド名は `UBBCONFIG` ファイルの `KEYWORD` と同じですが、`"TA_"` が先頭に付きます。

ユーザが入力バッファの編集を完了すると、`tmconfig` はその内容を読み取ります。特定のフィールド名についての行が複数ある場合、最初のオカレンスで使用され、ほかのオカレンスは無視されます。何らかのエラーが発生すると、構文エラー・メッセージが表示され、その問題を訂正するかどうかを確認するプロンプトが表示されます。

```
Enter editor to correct?
```

問題を訂正しなかった場合（上記プロンプトに対して `n` と応答）は、入力バッファにはフィールドが入りません。問題を訂正する場合は、エディタが再度実行されます。

最後に、操作を実行するかどうかを確認するプロンプトが表示されます。

```
Perform operation [y]?
```

操作が完了すると、`tmconfig` は `Return value TAOK` のように戻り値を表示し、続いて出力バッファのフィールドを表示します。次に、プロセスが再開され、セクションを求めるプロンプトが表示されます。すべての出力バッファ・フィールドは、バッファがクリアされない限り、入力バッファ内で利用できます。

いつでも `ブレーク` を入力して、セクションを求めるプロンプトでの対話を再開できます。

QUIT を選択すると、コンフィギュレーション・ファイルのバックアップ (テキスト形式) を作成するかどうかを確認するプロンプトが表示されます。

```
Unload TUXCONFIG file into ASCII backup [y]?
```

バックアップの作成を選択すると、ファイル名の入力を求めるプロンプトが表示されます。

```
Backup filename [UBBCONFIG]?
```

無事にバックアップが完了すると、`tmconfig` バックアップ・コピーが作成されたことを示すメッセージを表示します。失敗した場合はエラー・メッセージを表示します。

入力形式 入力パッケージは、次のようにフォーマットされた行で構成されます。

```
fldname fldval
```

フィールド名とフィールド値は 1 つ以上のタブで区切ります。

長いフィールド値の場合、継続行に 1 つ以上のタブを付けることによって、次の行にわたるようにできます (これらのタブは、`tmconfig` に読み戻される時に取り除かれます)。

改行文字だけで構成される空の行は無視されます。

表示できない文字をフィールド値に入力したり、タブからフィールド値を開始するには、バックスラッシュを使用し、その後に希望する文字を 2 文字の 16 進表現で使用します (UNIX System リファレンス・マニュアルの ASCII(5) を参照してください)。たとえば、入力データに空白を入力するには、`\20` と入力します。バックスラッシュは 2 回書くことで 1 つのバックスラッシュを入力できます。`tmconfig` はこの形式で入力された文字をすべて認識しますが、16 進形式を使用する最大の利点は表示不能文字を入力できることです。

制限事項 動的な再構成機能については、次のような一般的な制限事項があります。

1. キー・フィールドの値 (次のセクションを参照) は変更できません。
キー・フィールドがエディタ・バッファ内で変更され操作が終了すると、別のレコードがキー・フィールドの新しい値に基づいて変更されます。システムがダウンしているときに、コンフィギュレーション・ファイルを再ロードすれば、キー・フィールドを変更できます。

2. LMID レベルのフィールドは、LMID がブートされている間には変更できません。同様に、GROUP レベルのフィールドは GROUP がブートされている間には変更できません。
3. 多くの RESOURCES パラメータは実行中のシステムでは変更できません。
4. 動的削除機能はサポートされていません。削除はオフラインで行います。
5. 間違ったセクションでパラメータを更新しようとした場合、たとえば、RESOURCES セクションにいるときに、MACHINES セクションの ENVFILE パラメータを更新しようとした場合、tmconfig が TAOK を返して操作が正常終了したように見えますが、アンロードされた UBBCONFIG ファイルは変更されていません。

TMCONFIG
、
UBBCONFIG、
および
MIBS 間の関係

以前の BEA Tuxedo システムのリリースでは、すべてのアプリケーション構成は、アプリケーションのすべてのコンフィギュレーション・パラメータが定義された UBBCONFIG ファイルというテキスト・ファイルを編集することで実現していました。その後のリリースでは、tmloadcf(1) コマンドを実行することにより、UBBCONFIG ファイルを TUXCONFIG というバイナリ・ファイルにコンパイルするようになりました。さらに新しくなったシステムでは、各種 TUXCONFIG パラメータの動的更新（アクティブなシステムの更新）をサポートする tmconfig コマンドが導入されるようになりました。

システム 6 以降では、BEA Tuxedo のリソースをクラスや属性に再定義する BEA Tuxedo 管理情報ベース (MIB) が導入されました。MIB の導入と同時に、BEA Tuxedo システムは、管理者（またはユーザ）がアプリケーションの属性にプログラムでアクセスし、変更することを可能にする管理 API も提供しました。

1 つの例外を除いて、この『BEA Tuxedo コマンド・リファレンス』の [tmconfig](#)、[wtmconfig\(1\)](#) では、MIB の各種クラスについて簡単にしか触れていません。その例外とは Network クラスです。tmconfig() では、このクラスについて詳しく説明します。その他のセクションの詳細については、[TM_MIB\(5\)](#) を参照してください。

アトリビュート
フィールド
の更新時期
と更新者

従来の tmconfig テーブルには、フィールドの更新が可能かどうかを示す値を入れる列がありました。これについては MIB のリファレンス・ページで説明していますが、形式について多少注意が必要です。[MIB\(5\)](#) のパーミッションの説明を参照してください。MIB テーブルのパーミッションの列は、ファイルへのアクセス制限における読み出し、書き込み、および実行の許可

に似ていますが、この列では単純なファイル・パーミッションより多くの情報と制御を指定することができます。たとえば、MIB テーブルのパーミッション列に値を指定することにより、システム稼動時にフィールドの変更が可能かどうかを示すことができます。

tmconfig を使用する前には [MIB\(5\)](#) をよくお読みください。

RESOURCES セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_DOMAIN クラスの説明を参照してください

注意 このセクションについては、ADD 操作は使用できません。このセクションにはレコードが 1 つしかないので、RETRIEVE 操作は FIRST 操作と同じです (キー・フィールドは不要)。NEXT 操作は常に「レコードなし」が返されます。

TA_LDBAL、TA_CMTRET、および TA_SYSTEM_ACCESS に対する変更は、それ以後ブートされる新しいクライアントとサーバに対してのみ適用されます。NO_OVERRIDE の指定があり、指定アクセス・タイプと一致しない (PROTECTED または FASTPATH) サーバ・エントリがある場合、TA_SYSTEM_ACCESS は変更できません。TA_NOTIFY と TA_AUTHSVC に対する変更は、それ以後起動される新しいクライアントに対してのみ適用されます。

上記のパラメータ以外の更新は、アンロードされたテキスト形式のバックアップ・ファイルには反映されません。

MACHINES セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_MACHINE クラスの説明を参照してください。

注意 RESOURCES セクションの OPTIONS に "LAN" が指定されていない限り、マシンを追加することはできません。

上記のパラメータ以外の更新は、アンロードされたテキスト形式のバックアップ・ファイルには反映されません。

GROUPS セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_GROUP クラスの説明を参照してください。

SERVERS セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_SERVER クラスの説明を参照してください。

注意 `SERVERS` セクションで変更されたパラメータは、次回、関連するサーバがブート（再開ではなく）された時点で有効になります。複数のサーバが `MSSQ` に定義されている場合（`TA_RQADDR`）、これらのサーバでは同じサービスがブートされます。たとえば、`TA_CLOPT` あるいは `ENVFILE` に対する変更は、それらが現在ブートされているサーバと一致しないよう、ブートされるサービスには影響しないようにしなければなりません。`TA_MAX` に変更があった場合、そのサーバ・セットのサーバが 1 つまたはそれ以上ブートされるまで、新しいサーバ識別子に対応する会話サーバは自動的に生成されません。

SERVICES セクション このセクションの属性については、[TM_MIB\(5\)](#) の `T_SERVICE` および `T_SVCGRP` クラスの説明を参照してください。

注意 `SERVICES` セクションで変更されたパラメータは、次回、そのサービスを提供するサーバがブートされる時点（再開ではなく）で有効になります。`TA_ROUTINGNAME` の更新が可能なのは、`TA_SRVGRP` フィールドに値が指定されていない場合、またはそのフィールドに `NULL` が指定されている場合だけです。この場合、`TA_ROUTINGNAME` 属性は、一致するすべての `SERVICES` エントリで同時に更新されます。`TA_ROUTINGNAME` は `SERVICES` セクションの `ROUTING` に対応しています。

上記のパラメータ以外の更新は、アンロードされたテキスト形式のバックアップ・ファイルには反映されません。

NETWORK セクション 次の表に、`NETWORK` セクションのフィールド一覧を示します。

NETWORK セクション

フィールド識別子	フィールド・タイプ	更新	注意
<code>TA_LMID</code>	文字列	No	キー
<code>TA_BRIDGE</code>	文字列	Sys	
<code>TA_NADDR</code>	文字列	Sys	テキスト (ASCII) 形式 (NULL 文字を除く)

セクション 1 - コマンド

NETWORK セクション (続き)

フィールド識別子	フィールド・タイプ	更新	注意
TA_NLSADDR	文字列	Sys	テキスト (ASCII) 形式 (NULL 文字を除く)

注意 関連する LMID がブートされている場合には、レコードを追加することはできません。

RESOURCES セクションの OPTIONS に LAN がない限り、NETWORK セクションに対する操作は何もできません。

上記のパラメータ以外の更新は、アンロードされたテキスト形式のバックアップ・ファイルには反映されません。

ROUTING セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_ROUTING クラスの説明を参照してください。

注意 *ROUTING セクションは、システムが稼働状態にあるときには更新したり追加したりすることはできません。掲示板のサイズを制御する、RESOURCES セクションの 3 つのパラメータ MAXDRT、MAXRFT、および MAXRTDATA が増加可能に設定されている場合、ROUTING セクションの新しいエントリを追加できます。

WSL セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_WSL クラスを参照してください。

注意 ワークステーション・リスナ・サーバの CLOPT の更新は、SERVER セクションを通じて行うこともできますが、T_WSL クラスを使用してください。

NETGROUP S セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_WSL クラスを参照してください。

NETMAPS セクション このセクションの属性については、[TM_MIB\(5\)](#) の T_NETMAP クラスを参照してください。

INTERFACE S セクション このセクションの属性については、[TM_MIB\(5\)](#) の `T_INTERFACE` クラスの説明を参照してください。

注意 `T_INTERFACE` クラスは、ドメインおよびサーバ・グループの両方のレベルで CORBA インターフェイスのコンフィギュレーション属性と実行時属性を表します。特定のサーバ・グループへの処理の分散化を実現する、ファクトリ・ベース・ルーティングをインプリメントする場合を除き、CORBA インターフェイスの必須パラメータはありません。

セキュリティ `tmconfig` は保護アプリケーションで実行する場合、そのアプリケーションを利用するときアプリケーション・パスワードが必要とされます。標準入力が端末であると、`tmconfig` はユーザにパスワードの入力を求めてきます。ユーザが入力するパスワード文字列は画面には表示されません。標準入力が端末でない場合には、このパスワードは環境変数 `APP_PW` から取得されます。この環境変数の指定がなく、アプリケーション・パスワードが必要な場合には、`tmconfig` は異常終了します。

ワークステーション・クライアント /WS クライアントの場合、このコマンドは `wtmconfig` です `UPDATE` および `ADD` コマンドはサポートされていません (`TAEPerm` が返されます)。

環境変数 `tmconfig` は環境変数 `FIELDTBLS` および `FLDTBLDIR` を設定しなおして、`${TUXDIR}/udataobj/tpadmin` フィールド・テーブルを取り出します。`TUXDIR` を正しく設定しておかなければなりません。

`APP_PW` は、標準入力が端末でない場合、保護アプリケーションのアプリケーション・パスワードに設定しなければなりません。

クライアントを BEA Tuxedo アプリケーションに結合する前に、いくつかの環境変数を設定しておく必要があります。

- `tmconfig` に環境変数 `TUXCONFIG` を設定します。
- `wtmconfig` に環境変数 `WSNADDR` を設定します。BEA Tuxedo システムで使用するネットワーク・プロトコルによっては、`WSDEVICE` の設定が必要な場合もあります。使用するネットワーク・プロトコルは、アプリケーションが稼動するプラットフォームによって異なります。お使いのプラットフォームで使用されるネットワーク・プロトコルについては、

『BEA Tuxedo システムのインストール』の付録 A「プラットフォーム・データシート」を参照してください。

診断 `tmconfig` は、型付きバッファを割り当てることができない場合、該当ユーザに対する `/etc/passwd` エントリを判別できない場合、クライアント・プロセスになれない場合、入力バッファ編集用に `/tmp` に一時ファイルを作成できない場合、あるいは環境変数 `FIELDTBLS` または `FLDTBLDIR` を設定し直すことができない場合には異常終了します。

各操作が完了した後、`tmconfig` から出力される戻り値には、要求された操作に関する状況が示されます。戻り値には 3 つのクラスがあります。

次の戻り値は、パーミッションに関する問題または BEA Tuxedo の通信エラーを示しています。戻り値は、操作が正常に完了しなかったことを示しています。

[TAEPERM]

呼び出しプロセスは `TA_UPDATE` または `TA_ADD opcode` を指定しましたが、BEA Tuxedo 管理者として起動されていません。

[TAESYSTEM]

BEA Tuxedo エラーが発生しました。エラーの正確な内容は `userlog(3c)` に書き込まれます。

[TAEOS]

オペレーティング・システムのエラーが発生しました。

[TAETIME]

ブロッキング・タイムアウトが発生しました。入力バッファは更新されないため、取り出し操作では情報は返されません。更新操作のステータスは、更新されたレコードで取り出しを行うことでチェックできます。

次の戻り値は、操作を行うこと自体の問題を示します。一般的には、入力バッファ内のアプリケーション・データに関するセマンティクスの問題です。文字列フィールド `TA_STATUS` が出力バッファに設定され、その問題を示します。また、文字列フィールド `TA_BADFLDNAME` が、その問題の原因となった値を収めているフィールドのフィールド名に設定されます（エラーが 1 つのフィールドに限定されると想定した場合）。

[TAERANGE]

フィールド値が範囲外であるか、無効です。

[TAEINCONSIS]

フィールド値あるいは一連のフィールド値の指定に一貫性がありません(たとえば、既存の RQADDR 値を別の SRVGRP と SERVERNAME に指定した場合など)。

[TAECONFIG]

TUXCONFIG ファイルの読み取り中にエラーが発生しました。

[TAEDUPLICATE]

重複するレコードを追加しようとした。

[TAENOTFOUND]

操作対象として指定されたレコードが見つかりませんでした。

[TAEREQUIRED]

フィールド値が必要ですが、指定されていません。

[TAEISIZE]

文字列フィールドのフィールド値が長すぎます。

[TAEUPDATE]

実行できない更新を行おうとした。

[TAENOSPACE]

更新を行おうとしたが、TUXCONFIG ファイルまたは掲示板に十分な容量がありませんでした。

次の戻り値は、少なくとも MASTER サイトでは操作が成功したことを示しません。

[TAOK]

操作が成功しました。TUXCONFIG ファイルまたは掲示板には更新は行われませんでした。

[TAUPDATED]

操作が成功しました。TUXCONFIG ファイルまたは掲示板に更新が行われました。

[TAPARTIAL]

MASTER サイトでは操作は成功しましたが、1 つまたはそれ以上の非 MASTER サイトで失敗がありました。それらの非 MASTER サイトには、無効または分断のマークが付けられます。詳細については、管理者ガイドを参照してください。

相互運用性 BEA Tuxedo システム・リリース 4.0 または 4.1 ノードが起動されている状態では、UPDATE および ADD 操作は実行できません。これらの操作を行うには、前もってこれらのノードをシャットダウンしておかなければなりません。リブートすると、変更内容が有効になります。

TMUNLOAD CF の互換性 `tmunloadcf(1)` を使用して構成内のエントリを表示する場合、フィールドの値が設定されていないとき (文字列の場合)、フィールド値が 0 のとき (整数の場合)、あるいはフィールドのデフォルト値に一致する場合、それらのフィールド値は表示されません。これらのフィールドは、`tmconfig` を使用する際に常に出力バッファに存在します。この方法で、管理者がエントリを取り出しやすくし、また以前に設定されていないフィールドを更新しやすくします。エントリには、フィールド名が含まれ、タブが後に続きますが、フィールド値はありません。

使用例 次に挙げる例では、`tmconfig` を使用して、ワークステーション・リスナ・サーバ上に指定されたネットワーク・アドレスを訂正しています。このアドレスはサーバ・セクションの最初のエントリになっていることがあります。この例で使用されているエディタは `ed(1)` です。

```
$ EDITOR=ed tmconfig
```

```
Section:1) RESOURCES, 2) MACHINES, 3) GROUPS 4) SERVERS 5)SERVICES 6) NETWORK
7) ROUTING q) QUIT 9) WSL 10) NETGROUPS 11) NETMAPS 12) INTERFACES [1]:4
```

```
Operation: 1) FIRST 2) NEXT 3) RETRIEVE 4) ADD 5) UPDATE
```

```
6) CLEAR BUFFER 7) QUIT [1]: 1
```

```
Enter editor to add/modify fields [n]? <return>
```

```
Perform operation [y]? <return>
```

```
Return value TAOK
```

```
Buffer contents:
```

```
TA_OPERATION      4
TA_SECTION        3
TA_SRVID          2
TA_MIN            1
TA_MAX            1
TA_RQPERM        432
TA_RPPERM        432
TA_MAXGEN         1
```

```
TA_GRACE          86400
TA_STATUS         Operation completed successfully
TA_SRVGRP        WDBG
TA_SERVERNAME    WSL
TA_CLOPT         -A -- -d/dev/tcp -M4 -m2 -x5 -n0x0002fe19c00b6d6b
TA_CONV          N
TA_REPLYQ        N
TA_RESTART       N
Section:1) RESOURCES, 2) MACHINES, 3) GROUPS 4) SERVERS 5)SERVICES 6) NETWORK
7) ROUTING q) QUIT 9) WSL 10) NETGROUPS 11) NETMAPS 12) INTERFACES [4]: <return>
Operation: 1) FIRST 2) NEXT 3) RETRIEVE 4) ADD 5) UPDATE
6) CLEAR BUFFER 7) QUIT [1]: 5
Enter editor to add/modify fields [n]? y
240
/CLOPT/s/6d6b/690E/p
TA_CLOPT         -A -- -d/dev/tcp -M4 -m2 -x5 -n0x0002fe19c00b690E
w
240
q
Perform operation [y]? <return>
Return value TAUPDATED
Buffer contents:
TA_OPERATION     1
TA_SECTION       3
TA_SRVID         2
TA_MIN           1
TA_MAX           1
TA_RQPERM        432
TA_RPPERM        432
TA_MAXGEN        1
TA_GRACE         86400
TA_STATUS        Update completed successfully
TA_SRVGRP        WDBG
TA_SERVERNAME    WSL
TA_CLOPT         -A -- -d/dev/tcp -M4 -m2 -x5 -n0x0002fe19c00b690E
TA_CONV          N
TA_REPLYQ        N
TA_RESTART       N
Section:1) RESOURCES, 2) MACHINES, 3) GROUPS 4) SERVERS 5)SERVICES 6) NETWORK
7) ROUTING q) QUIT 9) WSL 10) NETGROUPS 11) NETMAPS 12) INTERFACES [1]: q
Unload TUXCONFIG file into ASCII backup [y]? <return>
Backup filename [UBBCONFIG]? <return>
Configuration backed up in UBBCONFIG
$ # boot the changed server
$ tmbboot -s WSL -i 2
```

関連項目 [tmbboot\(1\)](#)、[tmloadcf\(1\)](#)、[userlog\(3c\)](#)、[TM_MIB\(5\)](#)、[UBBCONFIG\(5\)](#)

tmipcrm(1)

名前 tmipcrm - ローカル・マシンで BEA Tuxedo ATMI アプリケーションが割り当てた IPC リソースを削除

形式 tmipcrm [-y] [-n] [TUXCONFIG_file]

機能説明 tmipcrm は、BEA Tuxedo ATMI アプリケーションによって割り当てられる、共用メモリ、メッセージ・キュー、セマフォなどの IPC リソースをクリーン・アップします。このコマンドは、通常、シャットダウンの異常終了などのエラー状況の発生後に実行されます。正常な状態では、割り当てられたすべての IPC リソースは、BEA Tuxedo ATMI システムのシャットダウン時にクリーン・アップされます。BEA Tuxedo ATMI のコア・システムとワークステーション・コンポーネントによって使用される IPC リソースも削除されます。

tmipcrm はローカル・サーバ・マシン上でのみ機能するので、BEA Tuxedo コンフィギュレーション内のリモート・マシン上にある IPC リソースはクリーン・アップされません。TUXCONFIG ファイルの名前は、TUXCONFIG 環境変数の値として指定するか、またはコマンド行で指定する必要があります。TUXCONFIG ファイルは必須で、読み取り可能である必要があります。

このコマンドを正常に実行できるのは、適切なパーミッションを持つユーザまたは管理者だけです。このコマンドは、掲示板 (BB) を格納する共用メモリ・セグメントにアタッチ可能であることを前提としており、掲示板に格納されて参照される IPC リソースを削除しようとしています。システム上のその他の条件によって削除が失敗することもあります。このような失敗が検出された場合には通知されます。

以下のオプションがサポートされています。

-y
すべてのプロンプトに「はい」と答えることを意味します。

-n
IPC リソースを削除せず、stdout に IPC リソースの一覧を表示して終了します。

TUXCONFIG_file
TUXCONFIG ファイルの完全なパス名。指定されていない場合は、TUXCONFIG 環境変数の値がデフォルト値になります。

診断 TUXCONFIG ファイルが見つからない場合は、致命的なエラーが発生し、次のメッセージが表示されます。

```
Cannot open tuxconfig file
```

TUXCONFIG ファイルが読み取れない場合は、致命的なエラーが発生し、次のメッセージが表示されます。

```
Execute permission denied, not application administrator
```

掲示板の共用メモリへのアタッチが失敗した場合は、致命的なエラーが発生し、次のメッセージが表示されます。

```
Cannot attach to BB!
```

使用例 このコマンドは、通常は会話モードで実行して、必要に応じて質問に対する応答を求めるプロンプトを表示します。-y オプションを指定すると、tmipcrm は応答を求めるプロンプトを表示しませんが、すべての質問に対して「はい」と答えたと見なされます。-n オプションを指定すると、tpipcrm は stdout に IPC リソースを表示するだけで終了し、IPC リソースは削除されません。

次の例は、このコマンドの一般的な使用方法を示しています。

```
$ tmipcrm /home/user/apps/tuxconfig
Looking for IPC resources in TUXCONFIG file
/home/user/apps/tuxconfig
The following IPC resources were found:

Message Queues:
0x2345
0x3456
...

Semaphores:
0x34567
0x45678
...

Shared Memory:
0x45678
0x56789
...

Remove these IPC resources (y/n)? y
Removing IPC resources ... done!
```

次のコード例では、ローカル・マシン上の BEA Tuxedo ATMI アプリケーションで使用される IPC リソースの一覧を、`ipclist` というファイルに出力します。

```
tmipcrm -n /home/user/apps/tuxconfig >ipclist
```

tmloadcf(1)

名前 tmloadcf - UBBCONFIG ファイルを解析し、バイナリ版の TUXCONFIG コンフィギュレーション・ファイルをロード

形式 tmloadcf [-n] [-y] [-c] [-b *blocks*] {UBBCONFIG_file | -}

機能説明 tmloadcf は、UBBCONFIG の構文に従った形式のファイルあるいは標準入力を読み取り、その構文をチェックし、指定に応じてバイナリ版の TUXCONFIG コンフィギュレーション・ファイルをロードします。環境変数 TUXCONFIG と TUXOFFSET (省略可能) は、それぞれ該当情報を格納する TUXCONFIG ファイルとオフセットを示します。tmloadcf は、UBBCONFIG ファイルの RESOURCES セクションに定義されている MASTER マシン上でしか実行できません。ただし、-c あるいは -n オプションが指定されている場合を除きます。

注記 TUXCONFIG 環境変数に指定するパス名は、UBBCONFIG ファイルの MACHINES セクションの TUXCONFIG パラメータに指定されているパス名と大文字と小文字の違いも含め、完全に一致していなければなりません。パス名が一致していない場合、tmloadcf(1) は正常に実行されません。

tmloadcf は、UBBCONFIG ファイルのセクションで抜けているものを検出すると、警告メッセージを表示します。ただし、LAN OPTION の指定のないコンフィギュレーションにおける NETWORK セクションの抜け (UBBCONFIG(5) を参照)、または ROUTING セクションの抜けは除きます。入力ファイルの解析中に構文エラーを検出すると、tmloadcf は TUXCONFIG ファイルへの更新を行わずに終了します。

tmloadcf を実行するユーザの実効ユーザ識別子は、UBBCONFIG ファイルの RESOURCES セクションに記述されている UID と同じでなければなりません (指定がある場合)。

tmloadcf にオプション -c を付けると、プログラムはこの構成に必要とされる最小の IPC リソースを表示します。プロセッサごとに異なるリソース要求条件が、構成に含まれる各プロセッサごとに表示されます。TUXCONFIG ファイルは更新されません。

tmloadcf で `-n` オプションを使用すると、プログラムは `UBBCONFIG` (コンフィグレーション・ファイルのテキスト・バージョン) の構文チェックだけを行い、`TUXCONFIG` ファイルの実際の更新は行いません。

構文チェックの後、`tmloadcf` は、`TUXCONFIG` が参照するファイルが存在するかどうか、存在する場合はそのファイルが有効な BEA Tuxedo システムのファイルシステムかどうか、および `TUXCONFIG` テーブルを含んでいるかどうかを調べます。これらの条件が満たされていない場合、`tmloadcf` は、`TUXCONFIG` の作成および初期化を行うかどうかを確認するプロンプトを表示します。

```
Initialize TUXCONFIG file:path [y, q]?
```

標準入力あるいは標準出力が端末でない場合、あるいはコマンド行で `-y` オプションが指定された場合には、このプロンプトは出されません。"y" または "Y" 以外の応答をすると、`tmloadcf` はコンフィギュレーション・ファイルを作成せずに終了します。

`TUXCONFIG` ファイルが正しく初期化されていない場合、処理を先に進めるよう指示すると、`tmloadcf` は BEA Tuxedo システムのファイルシステムを作成し、`TUXCONFIG` テーブルを作成します。コマンド行に `-b` オプションを指定すると、BEA Tuxedo システムのファイルシステムの作成時にそのデバイスのブロック数を引数として使用します。`-b` オプションの値が、新しい `TUXCONFIG` テーブルを収容できる大きさである場合、`tmloadcf` は指定された値を使用して新しいファイルシステムを作成します。それだけの大きさが無い場合には、エラー・メッセージを出力して終了します。`-b` オプションを指定しないと、`tmloadcf` は、`TUXCONFIG` テーブルを収容できる大きさのファイルシステムを新たに作成します。ファイルシステムが既に存在している場合は、`-b` オプションが無視されます。

`TUXCONFIG` が raw 型装置 (初期化されていない) であり、その raw 型装置上のブロック数を設定する必要がある場合には、`-b` オプションをできるだけ使用するようにしてください。`TUXCONFIG` が通常の UNIX ファイルである場合には、`-b` オプションは使用しないようにしてください。

`TUXCONFIG` ファイルが既に初期化されていると判断される場合、`tmloadcf` はその `TUXCONFIG` ファイルに記述されているシステムが稼働状態にないことを確認します。そのシステムが稼働状態にあると、`tmloadcf` はエラー・メッセージを出して終了します。

システムが非稼働状態にあり、TUXCONFIG ファイルが既に存在している場合は、ファイルを上書きするかどうかを確認するプロンプトが表示されます。

```
Really overwrite TUXCONFIG file [y, q]?
```

標準入力あるいは標準出力が端末でない場合、あるいはコマンド行で `-y` オプションが指定された場合には、このプロンプトは出されません。"y" または "q" 以外の応答を行うと、`tmloadcf` は終了して、ファイルの書き換えは行われません。

コンフィギュレーション・ファイルの `RESOURCES` セクションに `SECURITY` パラメータが指定されていると、`tmloadcf` は標準入力をフラッシュし、端末エコー機能をオフにして、ユーザにアプリケーション・パスワードの入力を求めます。

```
Enter Application Password?
Reenter Application Password?
```

パスワードは、30 文字以内の長さで入力できます。ファイルではなく標準入力を介して `UBBCONFIG` (コンフィギュレーション・ファイルのテキスト・バージョン) をロードするオプションは、`SECURITY` パラメータがオンのときには使用できません。標準入力がない場合、つまり、ユーザにパスワードの入力を求めることができない場合 (たとえば、`here` ファイルの場合)、アプリケーション・パスワードの設定のために環境変数 `APP_PW` が参照されます。`APP_PW` 環境変数が設定されておらず、標準入力がない場合には、`tmloadcf` はエラー・メッセージを出し、ログ・メッセージを生成して、`TUXCONFIG` ファイルのロードに失敗します。

エラーがないと仮定して、すべてのチェックの結果が正常であった場合、`tmloadcf` は `UBBCONFIG` ファイルを `TUXCONFIG` ファイルにロードします。このとき、`TUXCONFIG` テーブルにある既存の情報はすべて書き換えられます。

ロード中に丸められて、アンロードされたときに一致しない値もありますので、注意してください。そのような値には `MAXRFT` および `MAXRTDATA` がありますが、それだけではありません。

相互運用性 `tmloadcf` はマスタ・ノード上で実行しなければなりません。相互運用アプリケーションでは、マスタ・ノードは最新のリリースでなければなりません。

移植性 `tmloadcf` は、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。

環境変数 環境変数 `APP_PW` は、`SECURITY` パラメータが指定され、かつ標準入力として端末以外のものを使用して `tmloadcf` を実行するアプリケーションに対して設定しなければなりません。

使用例 `UBBCONFIG` ファイル `BB.shm` からコンフィギュレーション・ファイルをロードするには、次のようにデバイスを 2000 ブロックに初期化します。

```
tmloadcf -b2000 -y BB.shm
```

診断 入力にエラーが検出されると、エラーの原因となった行が、問題を指摘するメッセージと共に、標準エラーに出力されます。また、`UBBCONFIG` ファイルに構文エラーがあったり、システムがその時点で稼働中であると、`TUXCONFIG` ファイル内の情報はいっさい更新されず、`tmloadcf` は終了して、終了コード 1 を返します。

`UBBCONFIG` ファイルに指定されている UID と一致しない実効ユーザ識別子を持つユーザが `tmloadcf` を実行すると、次のようなエラー・メッセージが表示されます。

```
*** UID is not effective user ID ***
```

`tmloadcf` を非マスタ・ノード上で実行すると、次のようなエラー・メッセージが表示されます。

```
tmloadcf cannot run on a non-master node.
```

`tmloadcf` をアクティブ・ノード上で実行すると、次のようなエラー・メッセージが表示されます。

```
tmloadcf cannot run on an active node.
```

`tmloadcf` は正常に処理を完了すると、終了コード 0 を返して終了します。`TUXCONFIG` ファイルが更新された場合には、`userlog` メッセージが生成されてこのイベントが記録されます。

関連項目 [tmunloadcf\(1\)](#)、[UBBCONFIG\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

tmshutdown(1)

名前 tmshutdown - BEA Tuxedo サーバをシャットダウン

形式 tmshutdown [*options*]

機能説明 tmshutdown は、サーバの実行を停止させたり、コンフィギュレーション・ファイルに記述されているサービスの宣言を除去したりします。tmshutdown コマンドを呼び出せるのは、掲示板の管理者 (コンフィギュレーション・ファイルの UID パラメータで指定されている) または root だけです。また、tmshutdown は、コンフィギュレーション・ファイルの RESOURCES セクションで MASTER として指定されているマシン、あるいは MASTER として働くバックアップ・マシン (つまり、[tmadmin\(1\)](#) の master コマンドにより DBBL が既に稼働状態である) 上でしか呼び出せません。ただし、分断プロセッサに対して -P オプションが使用されている場合を除きます (下記参照)。

オプションなしの場合、tmshutdown はすべての管理サーバ、TMS、ゲートウェイ・サーバ、および TUXCONFIG 環境変数に指定されているコンフィギュレーション・ファイルの SERVERS セクションに記述されているサーバを停止させ、それらの関連 IPC リソースを除去します。各グループごとに、SERVERS セクションの全サーバ、関連するゲートウェイ・サーバ (外部グループの)、および TMS サーバがこの順序でシャットダウンされます。管理サーバは最後にシャットダウンされます。

SEQUENCE パラメータの指定がないアプリケーション・サーバは、コンフィギュレーション・ファイル内のサーバ・エントリの順序とは逆に最初にシャットダウンされ、SEQUENCE パラメータの指定されたアプリケーション・サーバはその後、順序番号の大きいものから順にシャットダウンされます。コンフィギュレーション・ファイルの SERVERS セクションの複数のサーバに同じ SEQUENCE パラメータが設定されている場合、tmshutdown はこれらのサーバを同時にシャットダウンします。SERVERS セクションの各エントリには、任意に MIN および MAX パラメータを指定することができます。tmshutdown は、-i オプションが指定されていない限り、各サーバ・エントリごとにそのサーバの全オカレンス (MAX 個まで) をシャットダウンします。-i オプションを使用すると、個々のオカレンスをシャットダウンできます。

あるサーバをシャットダウンできない場合、あるいはサービス宣言を除去できない場合、中央のイベント・ログに診断が書き出されます ([userlog\(3c\)](#) を参照)。次に、すべてのオプションについて個々に説明します。

-l *lmid*

関連する LMID パラメータが *lmid* である各グループごとに、該当グループに対応する SERVERS セクションの全サーバがまずシャットダウンされ、続いてそのグループに属する TMS とゲートウェイ・サーバがシャットダウンされます。

-g *grpname*

指定されたグループに属する SERVERS セクションのすべてのサーバ（つまり、SRVGRP パラメータが *grpname* に設定されている）がシャットダウンされ、続いてそのグループのすべて TMS とゲートウェイ・サーバがシャットダウンされます。TMS サーバのシャットダウンは、そのグループ・エントリの TMSNAME および TMSCOUNT パラメータの設定に基づいて行われます。外部グループの場合、HOST セクションの関連エントリに対応するゲートウェイ・サーバのシャットダウンは、パラメータ GATENAME と GATECOUNT に基づいて行われます。ゲートウェイをシャットダウンすると、そのプロセスが停止されると共に、その管理サービスと宣言されたすべての外部サービスの宣言が取り消されます。

-i *srvid*

SRVID パラメータが *srvid* である SERVERS セクションの全サーバがシャットダウンされます。30,000 を超える SRVID を入力しないでください。これらの数字はシステム・プロセス (TMS やゲートウェイ・プロセス) を表し、それらのシャットダウンには -l または -g オプションを使用しなければならないためです。

-s *aout*

SERVERS セクションのサーバのうち、名前が *aout* のサーバがすべてシャットダウンされます。このオプションは、TMS およびゲートウェイ・サーバのシャットダウンにも使用できます。

-o *sequence*

SERVERS セクションのサーバのうち、SEQUENCE パラメータが *sequence* に設定されているサーバがすべてシャットダウンされます。

-S

SERVERS セクションのサーバがすべてシャットダウンされます。

-A

管理サーバがすべてシャットダウンされます。

-M

マスタ・マシン上で管理サーバがシャットダウンされます。MASTER マシン上の BBL がシャットダウンされ、コンフィギュレーション・ファイルに LAN と NETWORK エントリが指定されていれば、BRIDGE がシャットダウンされます。MODEL が MP であると、DBBL 管理サーバがシャットダウンされます。

-B *lmid*

論理名が *lmid* であるプロセッサ上の BBL がシャットダウンされます。

-T *grpname*

SRVGRP パラメータが *grpname* に設定されたサーバ・グループ内のすべての TMS サーバが、そのサーバ・グループ・エントリに対応する TMSNAME および TMSCOUNT パラメータに基づいてシャットダウンされます。

-w *delay*

すべてのサーバを直ちに中断し、*delay* 秒だけシャットダウンの確認を待機した後、SIGTERM および SIGKILL シグナルをサーバに送って強制的に終了させます。

SIGKILL シグナルは捕捉できないため、その時点でプロセスによって実行されるコードには無関係に、シグナルを受信したプロセスはすべて直ちに終了されます。停止されるプロセスが掲示板を更新中だった場合には、掲示板に構造上の損傷を与えるおそれがあります。

注記 -w オプションが適用されるサーバは、UNIX System のシグナル SIGTERM を捕捉できません。

注記 SIGKILL シグナルの受信に基づいてサーバがシャットダウンされる場合、エントリは掲示板に残ります。BBL がシャットダウンされる場合、これらのエントリが検出され、BBL はシャットダウンしません。システムのシャットダウンを完了するには、再度 `tmshutdown` コマンドを使用しなければならない場合もあります。

-k {TERM | KILL}

`tmshutdown` は選択したサーバを直ちに中断し、これらを順序立てて (TERM) または強制的に (KILL) シャットダウンします。

SIGKILL シグナルは捕捉できないため、その時点でプロセスによって実行されるコードには無関係に、シグナルを受信したプロセスはすべて

て直ちに終了されます。停止されるプロセスが掲示板を更新中だった場合には、掲示板に構造上の損傷を与えるおそれがあります。

注記 このオプションは、UNIX シグナルの SIGTERM と SIGKILL をサポートするシステムでは、これらにマップされます。デフォルトでは、SIGTERM は BEA Tuxedo サーバで順序立てたシャットダウンを開始します。SIGTERM をアプリケーションでリセットすると、BEA Tuxedo システムがサーバをシャットダウンできなくなる場合があります。

注記 SIGKILL シグナルの受信に基づいてサーバがシャットダウンされる場合、エントリは掲示板に残ります。BBL がシャットダウンされる場合、これらのエントリが検出され、BBL はシャットダウンしません。システムのシャットダウンを完了するには、再度 `tmshutdown` コマンドを使用しなければならない場合もあります。

-y

全管理プロセスおよびサーバ・プロセスをシャットダウンするかどうかを確認するプロンプトに対して `yes` の応答がなされたものと見なされます (このプロンプトは、制限を課すオプションを一切つけずにコマンドが入力されたときにのみ表示されます)。

-y オプションを指定すると、すべてのサービスの宣言が掲示板から直ちに取り消され、以降サービス呼び出しができなくなります。

-q

標準出力に実行シーケンスを表示しません。-y が指定されているものと想定します。

-n

実行シーケンスを表示しますが、実行はされません。

-R

移行操作の場合のみ、元のプロセッサ上のサーバがシャットダウンされ、別のプロセッサへの移行準備としての掲示板エントリの削除は行われません。-R オプションは、-l あるいは -g オプションと一緒に使用しなければなりません (例: `tmshutdown -l lmid -R`)。また、コンフィギュレーション・ファイルの *RESOURCES セクションに MIGRATE オプションを指定しなければなりません。

-c

クライアントがまだアタッチされていても、BBL がシャットダウンされます。

-H *lmid*

単一プロセッサ上では、指定された *lmid* に対応するノード上の全管理サーバおよびアプリケーション・サーバがシャットダウンされます。一方、マルチプロセッサ (たとえば 3B4000) 上では、PE が 1 つしか指定されていない場合でも、すべての PE がシャットダウンされます。

-P *lmid*

このオプションを指定すると、`tmshutdown` は指定 *lmid* 上の掲示板にアタッチし、この *lmid* が該当アプリケーションの他の部分から分断されていることを確認してから (つまり、DBBL にはアクセスしません)、すべての管理サーバとアプリケーション・サーバをシャットダウンします。このコマンドは、コンフィギュレーション・ファイルの MACHINES セクションに指定されている *lmid* に対応するプロセッサ上で実行しなければなりません。

-l、-g、-s および -T オプションを使用すると、TMS サーバがシャットダウンされます。また、-l、-g および -s オプションを使用すると、ゲートウェイ・サーバがシャットダウンされます。-l、-g、-i、-s、-o および -S オプションはアプリケーション・サーバに適用されます。-A、-M および -B オプションは、管理プロセスにのみ適用されます。-l、-g、-i、-o および -s オプションを組み合わせると、指定された修飾条件をすべて満たすサーバのみがシャットダウンされます。

サーバのシャットダウン時にグローバル・トランザクションが稼働しているといったような分散トランザクション処理機能を使用する場合には、プリコミット後のコミットが記録される場所まで処理が進んでいなかったトランザクションは、アポートします。一方、コミット点まで進んでいたトランザクションは、サーバ (たとえば、TMS) が再度ブートされた時点で処理が完了します。

相互運用性

`tmshutdown` はマスタ・ノード上で実行しなければなりません。相互運用アプリケーションでは、マスタ・ノードは最新のリリースでなければなりません。`tmshutdown` は、リリース 4.1 サイトでリリース 4.2 管理サーバをシャットダウンさせることにつながるコンフィギュレーション・ファイル内の条件を検出し報告します。

- 移植性** `tmshutdown` は、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。
- 診断** `tmshutdown` は、サーバのシャットダウンに失敗した場合や致命的エラーが発生した場合には、終了コード 1 を返して終了します。詳細については、ユーザ・ログを調べてください。正常に実行できた場合には、終了コード 0 で終了します。
- `tmshutdown` が、マスタ・ノードでないアクティブ・ノード上で実行されると、次のような致命的エラー・メッセージが表示されます。
- ```
tmshutdown cannot run on a non acting-master node in an active application.
```
- プロセスをシャットダウンすることでアクティブ・プロセスが DBBL から分断されると、次のような致命的エラー・メッセージが表示されます。
- ```
cannot shutdown, causes partitioning.
```
- サーバが異常終了した場合には、次のようなメッセージが出されます。
- ```
CMDTUX_CAT:947 Cannot shutdown server GRPID
```
- 使用例** システム全体をシャットダウンして、BEA Tuxedo の IPC リソースをすべて削除するには、次のコマンドを実行します。30 秒経過しても確認が得られないと強制的に実行されます。
- ```
tmshutdown -w 30
```
- `lmid` の値が CS1 に設定されているマシン上のサーバだけをシャットダウンするには、次のコマンドを入力します。
- ```
tmshutdown -l CS1
```
- `-l` オプションは `SERVERS` セクションにリストされているサーバのみにアクションを制限するものであるため、CS1 上の BBL はシャットダウンされません。
- 注意事項** `tmshutdown` コマンドは、ハングアップ・シグナル (SIGHUP) を無視します。シャットダウン中にシグナルが検出されても、プロセスは継続します。
- 関連項目** [tmadmin\(1\)](#)、[tmboot\(1\)](#)、[UBBCONFIG\(5\)](#)
- 『BEA Tuxedo アプリケーション実行時の管理』

## tmunloadcf(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tmunloadcf - バイナリ版の TUXCONFIG コンフィギュレーション・ファイルをアンロード                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 形式   | tmunloadcf                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 機能説明 | <p>tmunloadcf は、TUXCONFIG コンフィギュレーション・ファイルをバイナリ表現からテキスト形式に変換します。この変換処理は、バイト・オーダが異なるマシン間でファイルを簡潔に移送したり、ファイルのバックアップ・コピーを簡潔に作成して信頼性を高める場合に役立ちます。テキスト形式については、UBBCONFIG(5) で説明します。</p> <p>tmunloadcf は、TUXCONFIG および TUXOFFSET 環境変数が示す TUXCONFIG ファイルから値を読み取り、それらを標準出力に書き出します。</p> <p>BEA Tuxedo リリース 7.1 で起動すると、複数のリソースに対してパスワードを使用できます。たとえば、リソース・マネージャの OPENINFO 文字列にパスワードを指定できます。パスワードを含む TUXCONFIG コンフィギュレーション・ファイルに対して tmunloadcf が実行された場合、出力には暗号化されたパスワードが表示されます。この暗号化されたパスワードは、1 度だけシステムにアップロードできます。</p> <p>環境設定中に四捨五入され、tmloadcf の呼び出しまたは TMIB インターフェイスによって設定した値と一致しない値もありますので、注意してください。そのような値には MAXRFT および MAXRTDATA がありますが、それだけではありません。</p> |
| 移植性  | tmunloadcf は、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 使用例  | <p>次のコマンドを入力すると、/usr/tuxedo/tuxconfig の構成がファイル tconfig.backup にアンロードされます。</p> <pre>TUXCONFIG=/usr/tuxedo/tuxconfig tmunloadcf &gt; tconfig.backup</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 診断   | tmunloadcf は、TUXCONFIG および TUXOFFSET 環境変数が参照するファイルが存在するかどうか、存在する場合はそのファイルが有効な BEA Tuxedo システムのファイルシステムであるかどうか、および TUXCONFIG テーブルを含んでいるかどうかを調べます。これらの条件のいずれかが満たされない場                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

合、`tmunloadcf` はエラー・メッセージを出し、終了コード 1 を返して終了します。処理が成功すると、`tmunloadcf` は終了コード 0 を返して終了します。

関連項目 [tmloadcf\(1\)](#)、[UBBCONFIG\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

## topendpasswd(1)

名前 topendpasswd - BEA Tuxedo ATMI システムの TOP END Domain Gateway  
機能で使用するパスワードを追加

形式 dmadmin  
> topendpasswd [-r] local\_domain\_access\_point\_name

機能説明 topendpasswd は、[dmadmin\(1\)](#) のサブコマンドとしてだけ実行できます。

このサブコマンドは、指定したローカル・ドメインの BEA TOP END で使用する新しいパスワードの入力を求めるプロンプトを管理者に対して表示します。-r オプションは、既存のパスワードと新しいパスワードをシステムが生成した新しいキーを使用して符号化することを指定します。パスワードは 12 文字以内です。このパスワードは、セキュリティ機能が有効になっている BEA TOP END システムに RTQ メッセージを送信する際に使用されます。渡されるユーザ ID は、ローカル・ドメインの DOMAINID です。

このサブコマンドを実行するには、事前に BEA Tuxedo Administration Console または [tmloadcf\(1\)](#) および [dmloadcf\(1\)](#) コマンドのいずれかを使用して、アプリケーションを設定しておく必要があります。アクティブなノード上であれば dmadmin を実行できます。

診断 dmadmin topendpasswd サブコマンドは、正常に完了すると、戻りコード 0 を返して終了します。

使用例 dmadmin  
>topendpasswd ldom  
Enter Local Domain Password:  
Re-enter Local Domain Password:

関連項目 [dmadmin\(1\)](#)

tpacladd(1)

名前 tpacladd - 新しいアクセス制御リストをシステムに追加

形式 tpacladd [-g GID[,GID .. .]] [-t type] name

機能説明 tpacladd を実行すると、BEA Tuxedo ATMI のセキュリティ・データ・ファイルにアクセス制御リスト (ACL) の新しいエントリが追加されます。この情報は、サービス、イベント、アプリケーションの各キューに対する BEA Tuxedo ATMI のアクセス制御に使用されます。このコマンドを実行する前に、SECURITY を USER\_AUTH、ACL、または MANDATORY\_ACL に設定した BEA Tuxedo のコンフィギュレーションを作成する必要があります。

以下のオプションが使用できます。

-g GID,...

1 つまたは複数の既存のグループの整数識別子またはグループ名の文字列のリストを指定します。このオプションは、指定したオブジェクトにどのグループがアクセスできるかを示します。このオプションを指定しないと、グループなしでエントリが追加されます。

-t type

オブジェクトのタイプを指定します。指定できるタイプは、ENQ、DEQ、SERVICE、または POSTEVENT のいずれかです。デフォルトは "SERVICE" です。

name

アクセス権を与えるサービス、イベント、またはアプリケーション・キューの名前を、表示可能な文字の重複しない文字列で指定します。コロン (:)、シャープ (#)、または改行文字 (\n) を使うことはできません。

このコマンドを実行する前に、グラフィカル・ユーザ・インターフェイスまたは [tmloadcf\(1\)](#) のいずれかを使用してアプリケーションを環境設定し、TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する必要があります。アプリケーションがアクティブでない場合は、tpacladd をコンフィギュレーションの MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。

移植性 このコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。

診断 `tpacladd` コマンドは正しく完了すると、終了コード 0 で終了します。

関連項目 [tpacldel\(1\)](#)、[tpaclmod\(1\)](#)、[tpgrpadd\(1\)](#)、[tpgrpdel\(1\)](#)、[tpgrpmod\(1\)](#)、[AUTHSVR\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

### tpaclcvt(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpaclcvt - BEA Tuxedo ATMI のセキュリティ・データ・ファイルを変換する                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 形式   | tpaclcvt [-u <i>userfile</i> ] [-g <i>groupfile</i> ]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 機能説明 | <p>tpaclcvt は、任意のバージョンの AUTHSVR (BEA Tuxedo リリース 5.0 で利用できるバージョン) で使用される既存のユーザ・ファイル調べ、これを BEA Tuxedo リリース 6.0 で使用する形式に変換します。また、<code>/etc/group</code> または類似のファイルに基づいてグループ・ファイルを生成し、<code>/etc/passwd</code> ファイルを変換します。</p> <p>以下のオプションが使用できます。</p> <p><code>-u <i>userfile</i></code><br/>System 5 のユーザ・ファイルを指定します。このオプションを指定しないと、ユーザ・ファイルは変換されません。</p> <p><code>-g <i>groupfile</i></code><br/>グループ・ファイル (通常は <code>/etc/group</code>) を指定します。このオプションを指定しないと、グループ・ファイルは変換されません。</p> <p>このコマンドを実行する前に、グラフィカル・ユーザ・インターフェイスまたは <code>tmloadcf(1)</code> のいずれかを使用してアプリケーションを環境設定し、TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定します。アプリケーションがアクティブでない場合は、tpaclcvt をコンフィギュレーションの MASTER 上で実行する必要があります。</p> |
| 移植性  | このコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 関連項目 | <a href="#">tpgrpadd(1)</a> 、 <a href="#">tpusradd(1)</a> 、 <a href="#">AUTHSVR(5)</a><br>『BEA Tuxedo アプリケーション実行時の管理』                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## tpacldel(1)

名前 tpacldel - アクセス制御リストのエントリを削除

形式 tpacldel [-t *type*] *name*

機能説明 tpacldel を実行すると、BEA Tuxedo ATMI のセキュリティ・データ・ファイルから既存のアクセス制御リスト (ACL) のエントリが削除されます。このコマンドを実行する前に、SECURITY を USER\_AUTH、ACL、または MANDATORY\_ACL に設定した BEA Tuxedo のコンフィギュレーションを作成する必要があります。

以下のオプションが使用できます。

-t *type*

オブジェクトのタイプを指定します。指定できるタイプは、ENQ、DEQ、SERVICE、または POSTEVENT のいずれかです。デフォルトのタイプは "SERVICE" です。

*name*

削除する既存の ACL のエントリを指定します。

このコマンドを実行する前に、グラフィカル・ユーザ・インターフェイスまたは [tmloadcf\(1\)](#) のいずれかを使用してアプリケーションを環境設定し、TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する必要があります。アプリケーションがアクティブでない場合は、tpacldel をコンフィギュレーションの MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。

移植性 このコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。

診断 tpacldel コマンドは正しく完了すると、終了コード 0 で終了します。

関連項目 [tpacladd\(1\)](#)、[tpaclmod\(1\)](#)、[AUTHSVR\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

### tpaclmod(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpaclmod - システム上のアクセス制御リストを変更                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 形式   | tpaclmod [-g <i>GID</i> [, <i>GID</i> ...]][-t <i>type</i> ] <i>name</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 機能説明 | <p>tpaclmod を実行すると、BEA Tuxedo のセキュリティ・データ・ファイル内のアクセス制御リスト (ACL) のエントリが変更されて、グループ識別子リストが置き換えられます。この情報は、サービス、イベント、アプリケーションの各キューに対する BEA Tuxedo ATMI のアクセス制御に使用されます。このコマンドを実行する前に、SECURITY を USER_AUTH、ACL、または MANDATORY_ACL に設定した BEA Tuxedo のコンフィギュレーションを作成する必要があります。</p> <p>以下のオプションが使用できます。</p> <p>-g <i>GID</i>,...</p> <p>1 つまたは複数の既存のグループの整数識別子またはグループ名の文字列のリストを指定します。このオプションは、指定したオブジェクトにどのグループがアクセスできるかを示します。このオプションを指定しないと、エントリはグループを持たないように変更されます。</p> <p>-t <i>type</i></p> <p>オブジェクトのタイプを指定します。指定できるタイプは、ENQ、DEQ、SERVICE、または POSTEVENT のいずれかです。デフォルトは "SERVICE" です。</p> <p><i>name</i></p> <p>既存の ACL の名前を指定します。</p> <p>このコマンドを実行する前に、グラフィカル・ユーザ・インターフェイスまたは <a href="#">tmloadcf(1)</a> のいずれかを使用してアプリケーションを環境設定し、TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する必要があります。アプリケーションがアクティブでない場合は、tpaclmod を環境設定の MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。</p> |
| 移植性  | このコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 診断   | tpaclmod コマンドは正しく完了すると、終了コード 0 で終了します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

関連項目 [tpacladd\(1\)](#)、[tpacldel\(1\)](#)、[AUTHSVR\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

## tpaddusr(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpaddusr - BEA Tuxedo のパスワード・ファイルを作成                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 形式   | tpaddusr <i>username file</i> [ <i>cltname</i> [ <i>UID</i> ]]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 機能説明 | このコマンドを使用すると、アプリケーション管理者は、BEA Tuxedo の AUTHSVR(5) サーバでの使用に適した UNIX システム形式のパスワード・ファイルを作成できます。tpaddusr は、ユーザの <i>username</i> をパスワード・ファイル <i>file</i> に追加します ( <i>file</i> に <code>/etc/passwd</code> を指定することはできません)。管理者は、追加したユーザに割り当てるパスワードを入力するように要求されます。必要であれば、0600 のアクセス権で <i>file</i> が作成されます。 <i>cltname</i> は指定すると、パスワード・エントリでの追加の修飾子を示します。 <i>username</i> または <i>cltname</i> 、またはそれらの両方は、'*' 文字として指定することができ、AUTHSVR(5) ではワイルドカードと見なされます。 <i>UID</i> は指定すると、ユーザの認証が成功した場合に返されるユーザの識別番号を示します。指定がない場合には、 <i>cltname</i> と <i>UID</i> はそれぞれ '*' と -1 がデフォルト値となります。 |
| 注意事項 | <i>cltname</i> の値 <code>tpsysadm</code> と <code>tpsysop</code> は、認証要求を処理する際に AUTHSVR(5) により特殊な処理が行われます。これらの <i>cltname</i> 値は、パスワード・ファイル内のワイルドカードの <i>cltname</i> 指定と照合されることはありません。<br><br>さらに、パスワード・ファイルへの追加順序に関係なく、ワイルドカードのエントリは、明示的に指定された値の後で処理されます。認証要求は、最初に一致するパスワード・ファイルのエントリに対してのみ認証されます。                                                                                                                                                                                                                                                                      |
| 移植性  | このコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 互換性  | このコマンドは、ユーザを <code>SECURITY_USER_AUTH</code> に環境設定するために使用します。 <code>SECURITY_ACL</code> または <code>MANDATORY_ACL</code> との互換性を保つためには (これらのセキュリティ・レベルに移行できるようにするためにも)、次の制限に従う必要があります。 <ul style="list-style-type: none"> <li>■ ユーザ名は一意でなければならず、ワイルドカードを使用することはできない。</li> <li>■ ユーザ識別子は一意でなければならない。ユーザ識別子は 0 より大きく 128K 未満でなければならない。</li> </ul>                                                                                                                                                                                                                          |

- ファイル名は \$APPDIR/tpusr でなければならない。

[tpusradd\(1\)](#) コマンドでは、これらの制限が自動的に適用されます。

**使用例** 以下の一連のコマンド入力は、簡単なパスワード・ファイルの構成方法を示しています。

```
$ # 1. Add username foo with wildcard cltname and no UID
$ tpaddusr foo /home/tuxapp/pwfile
$ # 2. Add username foo with cltname bar and UID 100
$ tpaddusr foo /home/tuxapp/pwfile bar 100
$ # 3. Add username foo with tpsysadm cltname and no UID
$ tpaddusr foo /home/tuxapp/pwfile tpsysadm
$ # 4. Add wildcard username with tpsysop cltname and no UID
$ tpaddusr '*' /home/tuxapp/pwfile tpsysop
$ # 5. Add wildcard username with wildcard cltname and no UID
$ tpaddusr '*' /home/tuxapp/pwfile '*'
```

次の表は、アプリケーションに対する種々のアクセス要求を認証するのに使用するパスワード・ファイル・エントリ(上記の番号で示されます)を示します。N/A は、パスワード・ファイルに一致するエントリが存在しないため、要求が許可されないことを示します。

| Username | Cltname    | Password | Entry |
|----------|------------|----------|-------|
| "foo"    | "bar"      |          | 2     |
| "foo"    | " "        |          | 1     |
| "foo"    | "tpsysadm" |          | 3     |
| "foo"    | "tpsysop"  |          | 4     |
| "guest"  | "tpsysop"  |          | 4     |
| "guest"  | "bar"      |          | 5     |
| "guest"  | "tpsysadm" |          | N/A   |

次に、上記で作成されたパスワード・ファイル用の AUTHSVR インスタンスのための \*SERVERS セクション例を示します。

```
AUTHSVR SRVGRP=G SRVID=1 RESTART=Y GRACE=0 MAXGEN=2 CLOPT="-A -- -f
/home/tuxapp/pwfile"
```

**関連項目** [tpdelusr\(1\)](#)、[tpmodusr\(1\)](#)、[tpusradd\(1\)](#)、[tpusrdel\(1\)](#)、[tpusrmod\(1\)](#)、[AUTHSVR\(5\)](#)

## tpdelusr(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpdelusr - BEA Tuxedo のパスワード・ファイルからユーザを削除                                                                                                                                                                                                                                                                                                                                                                                               |
| 形式   | tpdelusr <i>username file [cltname]</i>                                                                                                                                                                                                                                                                                                                                                                                                 |
| 機能説明 | このコマンドにより、アプリケーション管理者は、BEA Tuxedo の AUTHSVR(5) サーバでの使用に適した UNIX システム形式のパスワード・ファイルを管理できます。tpdelusr は、指定された <i>username</i> と <i>cltname</i> の組み合わせのパスワード・ファイル・エントリを削除するために使用します。 <i>file</i> に /etc/passwd を指定することはできません。 <i>cltname</i> は指定しなければ、デフォルトで <code>&amp;slq;*&amp;srq;</code> されます。 <i>username</i> または <i>cltname</i> 、もしくはこれらの両方に指定されたワイルドカードは、パスワード・ファイル内の対応するワイルドカード・エントリに対してのみ一致します。これらはすべての一致するエントリに対して拡張されるものではありません。 |
| 注意事項 | <p><i>cltname</i> の値 <code>tpsysadm</code> と <code>tpsysop</code> は、認証要求を処理する際に AUTHSVR(5) により特殊な処理が行われます。これらの <i>cltname</i> 値は、パスワード・ファイル内のワイルドカードの <i>cltname</i> 指定と照合されることはありません。</p> <p>さらに、パスワード・ファイルへの追加順序に関係なく、ワイルドカードのエントリは、明示的に指定された値の後で処理されます。認証要求は、最初に一致するパスワード・ファイルのエントリに対してのみ認証されます。</p>                                                                                                                                 |
| 移植性  | このコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                   |
| 互換性  | <p>このコマンドは、ユーザを SECURITY_USER_AUTH に環境設定するために使用します。SECURITY_ACL または MANDATORY_ACL との互換性を保つためには (これらのセキュリティ・レベルに移行できるようにするためにも)、次の制限に従う必要があります。</p> <ul style="list-style-type: none"> <li>■ ユーザ名は一意でなければならず、ワイルドカードを使用することはできない。</li> <li>■ ユーザ識別子は一意で、0 より大きく 128K 未満でなければならない。</li> <li>■ ファイル名は \$APPDIR/tpusr でなければならない。</li> </ul> <p><code>tpusrdel(1)</code> コマンドでは、これらの制限が自動的に適用されます。</p>                                       |

関連項目 [tpaddusr\(1\)](#)、[tpmodusr\(1\)](#)、[tpusradd\(1\)](#)、[tpusrdel\(1\)](#)、[tpusrmod\(1\)](#)、[AUTHSVR\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

### tpgrpadd(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpgrpadd - システムに新しいグループを追加                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 形式   | tpgrpadd [-g <i>GID</i> ] <i>grpname</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 機能説明 | <p>tpgrpadd は、BEA Tuxedo セキュリティ・データ・ファイルに適切なエントリを追加することにより、システム上で新しいグループ定義を作成します。この情報は、AUTHSVR(5) サーバによる BEA Tuxedo システムの認証や、アクセス制御に使用されます。このコマンドを実行する前に、SECURITY を USER_AUTH、ACL、または MANDATORY_ACL に設定した BEA Tuxedo のコンフィギュレーションを作成する必要があります。</p> <p>以下のオプションが使用できます。</p> <p><b>-g <i>GID</i></b><br/>           新しいグループのグループ識別子を指定します。このグループ識別子は、16K より小さい正の 10 進数でなければなりません。<i>GID</i> のデフォルトは、1 以上の次の使用可能な (一意な) 識別子です。グループ識別子 0 は、&amp;dl“er&amp;dr”ープ用に予約されています。</p> <p><b><i>grpname</i></b><br/>           新しいグループの名前を、表示可能な文字の重複しない文字列で指定します。シャープ (#)、カンマ (,)、コロン (:)、または改行文字 (\n) を使うことはできません。</p> <p>このコマンドを実行する前に、グラフィカル・ユーザ・インターフェイスまたは <a href="#">tmloadcf(1)</a> のいずれかを使用してアプリケーションを環境設定し、TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する必要があります。アプリケーションがアクティブでない場合、tpgrpadd はコンフィギュレーションの MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。</p> |
| 移植性  | このコマンドは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 診断   | tpgrpadd コマンドは正しく完了すると、終了コード 0 で終了します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 関連項目 | <a href="#">tpgrpdel(1)</a> 、 <a href="#">tpgrpmod(1)</a> 、 <a href="#">tpusradd(1)</a> 、 <a href="#">tpusrdel(1)</a> 、 <a href="#">tpusrmod(1)</a> 、 <a href="#">AUTHSVR(5)</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

『BEA Tuxedo アプリケーション実行時の管理』

### tpgrpdel(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpgrpdel - システムからグループを削除                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 形式   | tpgrpdel <i>grpname</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 機能説明 | <p>tpgrpdel は、BEA Tuxedo セキュリティ・データ・ファイルから対応するグループのエントリを削除することによって、システムからグループの定義を削除します。ただし、ユーザ・ファイルからグループ ID は削除されません。このコマンドを実行する前に、SECURITY を USER_AUTH、ACL、または MANDATORY_ACL に設定した BEA Tuxedo のコンフィギュレーションを作成する必要があります。</p> <p>以下のオプションが使用できます。</p> <p><i>grpname</i><br/>削除する既存のグループの名前を指定します。</p> <p>このコマンドを実行する前に、グラフィカル・ユーザ・インターフェイスまたは <a href="#">tmloadcf(1)</a> のいずれかを使用してアプリケーションを環境設定し、TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する必要があります。アプリケーションがアクティブでない場合、tpgrpdel を環境設定の MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。</p> |
| 移植性  | このコマンドは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 診断   | tpgrpdel コマンドは正しく完了すると、終了コード 0 で終了します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 関連項目 | <a href="#">tpgrpadd(1)</a> 、 <a href="#">tpgrpmod(1)</a> 、 <a href="#">tpusradd(1)</a> 、 <a href="#">tpusrdel(1)</a> 、 <a href="#">tpusrmod(1)</a> 、 <a href="#">AUTHSVR(5)</a>                                                                                                                                                                                                                                                                                                                                                                               |
|      | 『BEA Tuxedo アプリケーション実行時の管理』                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

## tpgrpmod(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpgrpmod - システム上のグループを変更                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 形式   | tpgrpmod [-g <i>GID</i> ] [-n <i>name</i> ] <i>grpname</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 機能説明 | <p>tpgrpmod は、BEA Tuxedo セキュリティ・データ・ファイルの適切なエントリを変更することによって、指定したグループの定義を変更します。このコマンドを実行する前に、SECURITY を USER_AUTH、ACL、または MANDATORY_ACL に設定した BEA Tuxedo のコンフィギュレーションを作成する必要があります。</p> <p>以下のオプションが使用できます。</p> <p>-g <i>GID</i><br/>新しいグループのグループ識別子を指定します。このグループ識別子は、16K より小さい正の 10 進数でなければなりません。グループ識別子 0 は、&amp;ldq;r&amp;drq” プ用に予約されています。</p> <p>-n <i>name</i><br/>新しいグループの名前を、表示可能な文字の重複しない文字列で指定します。カンマ (,)、コロン (:)、または改行文字 (\n) を使うことはできません。</p> <p><i>grpname</i><br/>変更するグループの現在の名前を指定します。</p> <p>このコマンドを実行する前に、グラフィカル・ユーザ・インターフェイスまたは <a href="#">tmloadcf(1)</a> のいずれかを使用してアプリケーションを環境設定し、TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する必要があります。アプリケーションがアクティブでない場合、tpgrpmod を環境設定の MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。</p> |
| 移植性  | このコマンドは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 診断   | tpgrpmod コマンドは正しく完了すると、終了コード 0 で終了します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 関連項目 | <a href="#">tpgrpadd(1)</a> 、 <a href="#">tpgrpdel(1)</a> 、 <a href="#">tpusradd(1)</a> 、 <a href="#">tpusrdel(1)</a> 、 <a href="#">tpusrmod(1)</a> 、 <a href="#">AUTHSVR(5)</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

『BEA Tuxedo アプリケーション実行時の管理』

## tpmigldap(1)

名前 tpmigldap - Tuxedo ユーザおよびグループを WebLogic Server に移行

形式 tpmigldap [-h *hostname*] [-p *port*] [-d *wls\_domain*] [-r *wls\_realm*]  
[-f *user\_password*] [-b *bind\_DN*] [[-w *ldap\_adm\_password*]|  
[-c]] [-u *tpusr*] [-g *tpgrp*] [-i *UID-kw*] [-e *GID-kw*]

機能説明 tpmigldap を実行すると、WebLogic Server のデフォルトのセキュリティ・データベースに Tuxedo のユーザおよびグループを追加できます。このコマンドを実行する前に、SECURITY を USER\_AUTH、ACL、または MANDATORY\_ACL に設定した Tuxedo のコンフィギュレーションを作成する必要はありません。

以下のオプションが使用できます。

-h *hostname*

WebLogic Server が配置されている hostname。

-p *port*

WebLogic Administration Console のポート番号を指定します。

-d *wls\_domain*

WebLogic Server ドメイン名を指定します。

-r *wls\_realm*

WebLogic Server セキュリティ・レルム名を指定します。

-f *user\_password*

WebLogic Server に移行する全ユーザのデフォルトのパスワード設定。

-b *bind\_DN*

バインド DN (通常、WebLogic Server 組み込み LDAP サーバの管理者の DN) を指定します。

-w *ldap\_adm\_password*

WebLogic Server 組み込み LDAP サーバの管理者のパスワードを指定します。

-c

LDAP 管理者のパスワードの入力を求めるようにします。

-u *tpusr*

*tpusr* ファイルのパス名を指定します。

## セクション 1 - コマンド

---

`-g tpgrp`

`tpgrp` ファイルのパス名を指定します。

`-i UID-kw`

WebLogic Server ユーザ情報で Tuxedo UID を識別するためのキーワードを指定します。

`-e GID-kw`

WebLogic Server ユーザ情報で Tuxedo GID を識別するためのキーワードを指定します。

**移植性** `tpmigldap` コマンドは、Tuxedo System/T リリース 8.1 以降が動作している非 /WS サイト上でのみ使用できます。

**診断** `tpmigldap` コマンドは正しく完了すると、終了コード 0 で終了します。

**使用例** `$tpmigldap -h proton -c -d wlsdomain -r wlsrealm -b cn=Admin`

## tpmodusr(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpmodusr - BEA Tuxedo のシステム・パスワード・ファイルを保守                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 形式   | tpmodusr <i>usrname file [cltname]</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 機能説明 | <p>このコマンドを使用して、アプリケーション管理者は、BEA Tuxedo システムの <a href="#">AUTHSVR(5)</a> サーバでの使用に適した UNIX システム形式のパスワード・ファイルを管理できます。このコマンドを実行する前に、<code>SECURITY</code> を <code>USER_AUTH</code>、<code>ACL</code>、または <code>MANDATORY_ACL</code> に設定した BEA Tuxedo のコンフィギュレーションを作成する必要があります。</p> <p>tpmodusr は、パスワード・ファイル <i>file</i> 内の指定したユーザのパスワードを変更するのに使用します (<i>file</i> に <code>/etc/passwd</code> を指定することはできません)。管理者は、ユーザに関連付ける新しいパスワードの入力を求められません。パスワードを指定しない場合、<i>cltname</i> にはデフォルト値として <code>*</code> が設定されます。<i>usrname</i> または <i>cltname</i>、もしくはこれらの両方に指定されたワイルドカードは、パスワード・ファイル内の対応するワイルドカード・エントリに対してのみ一致します。これらはすべての一致するエントリに対して拡張されるものではありません。</p> |
| 注意事項 | <p><i>cltname</i> の値 <code>tpsyzadm</code> と <code>tpsyzop</code> は、認証要求を処理する際に <a href="#">AUTHSVR(5)</a> により特殊な処理が行われます。これらの <i>cltname</i> 値は、パスワード・ファイル内のワイルドカードの <i>cltname</i> 指定と照合されることはありません。</p> <p>さらに、パスワード・ファイルへの追加順序に関係なく、ワイルドカードのエントリは、明示的に指定された値の後で処理されます。認証要求は、最初に一致するパスワード・ファイルのエントリに対してのみ認証されます。</p>                                                                                                                                                                                                                                                                                                                            |
| 移植性  | このコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 互換性  | <p>このコマンドは、ユーザを <code>SECURITY USER_AUTH</code> に環境設定するために使用します。<code>SECURITY ACL</code> または <code>MANDATORY_ACL</code> との互換性を保つためには (これらのセキュリティ・レベルに移行できるようにするためにも)、次の制限に従う必要があります。</p> <ul style="list-style-type: none"> <li>■ ユーザ名は一意でなければならず、ワイルドカードを使用することはできない。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                 |

- ユーザ識別子は一意でなければならない。ユーザ識別子は 0 より大きく 128K 未満でなければならない。
- ファイル名は `$APPDIR/tpusr` でなければならない。

`tpusrmod(1)` コマンドでは、これらの制限が自動的に適用されます。

関連項目 `tpaddusr(1)`、`tpdelusr(1)`、`tpusradd(1)`、`tpusrdel(1)`、`tpusrmod(1)`、`AUTHSVR(5)`

『BEA Tuxedo アプリケーション実行時の管理』

## tpusradd(1)

名前 tpusradd - 新しいプリンシパルをシステムに追加

形式 tpusradd [-u *UID* ] [-g *GID*] [-c *client\_name*] *usrname*

機能説明 tpusradd を実行すると、BEA Tuxedo セキュリティ・データ・ファイルに新しいプリンシパル (ユーザまたはドメイン) のエントリが追加されます。この情報は、AUTHSVR(5) サーバでユーザ単位の認証に使用されます。

このコマンドを実行するには、以下の準備が必要です。

- グラフィカル・ユーザ・インターフェイスまたは [tmloadcf\(1\)](#) のいずれかを使用して、アプリケーションを環境設定する。
- TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する。
- SECURITY を USER\_AUTH、ACL、または MANDATORY\_ACL に設定する。

アプリケーションがアクティブでない場合、tpusradd をコンフィギュレーションの MASTER 上で実行する必要があります。アクティブな場合は、このコマンドはどのアクティブなモードでも実行できます。

このコマンドで作成されたシステム・ファイル・エントリは 1 行の制限が 512 文字です。いくつもの (複数の) オプションに長い引数を指定するとこの制限を超えることがあります。

以下のオプションが使用できます。

-u *UID*

ユーザ識別番号を指定します。*UID* は、16K より小さい正の 10 進数でなければなりません。*UID* は、アプリケーションの既存の識別子のリスト中で一意でなければなりません。*UID* のデフォルトは、1 以上の次の使用可能な (一意な) 識別子です。

-g *GID*

既存のグループの整数識別子またはグループ名の文字列を指定します。このオプションによって、新しいユーザがどのグループに属するかを定義します。デフォルトは、&ldquo;other&rdquo; (識別子 0) ”c

*client\_name*

ユーザに関連付けるクライアント名を、表示可能な文字列で指定します。このオプションを指定した場合、通常は、関連付けられたユーザ

の役割を表し、ユーザのエントリに関する付加的な情報となります。コロン (:) または改行文字 (\n) を使うことはできません。このオプションを指定しないと、デフォルトはワイルドカード &slq;\*’ になり、どのようなクライ ‘ 指定しても認証が成功します。

### username

ユーザの新しいログイン名を、表示可能な文字列で指定します。コロン (:), シャープ (#), または改行文字 (\n) を使うことはできません。ユーザ名は、アプリケーション用の既存の識別子のリスト中で一意でなければなりません。

管理者は、追加したユーザに割り当てるパスワードを入力するように要求されます。

ユーザ単位の認証や管理者のパーミッションの設定の詳細については、[AUTHSVR\(5\)](#) を参照してください。

**移植性** このコマンドは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。

**診断** tpusradd コマンドは正しく完了すると、終了コード 0 で終了します。

**使用例** 以下の一連のコマンドは、簡単なユーザ・ファイルの作成例を示します。

```
$ # 1. Add username foo with cltname bar and UID 100
$ tpusradd -u 100 -c bar foo
$ # 2. Add username foo with tpsysadm cltname and no UID
$ tpusradd -c tpsysadm foo
```

次の表は、アプリケーションへの様々なアクセス要求の認証に使用されるユーザ・エントリ (上に書かれている数字で示されます) と、関連付けられる UID/GID を示します。N/A は、ユーザ・ファイルに一致するエントリが存在しないため、要求が許可されないことを示します。

| Username | Cltname    | Password | Entry | Uid | Gid  |
|----------|------------|----------|-------|-----|------|
| "foo"    | "bar"      | 2        |       | 100 | 0    |
| "foo"    | " "        | 1        |       | 1   | 0    |
| "foo"    | "tpsysadm" | 3        |       | 0   | 8192 |
| "guest"  | "tpsysadm" | N/A      |       | N/A | N/A  |

次に示す例は、上で作成したユーザ・ファイルを処理する AUTHSVR のインスタンスの &dlq“ERS&drq” ユーザ・エントリです。

```
AUTHSVR SRVGRP=G SRVID=1 RESTART=Y GRACE=0 MAXGEN=2 CLOPT="-A"
```

関連項目 [tpgrpadd\(1\)](#)、[tpgrpdel\(1\)](#)、[tpgrpmod\(1\)](#)、[tpusrdel\(1\)](#)、[tpusrmod\(1\)](#)、[AUTHSVR\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

### tpusrdel(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | tpusrdel - システムからユーザを削除                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 形式   | tpusrdel <i>usrname</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 機能説明 | <p>tpusrdel コマンドは、システムからプリンシパル(ユーザ名またはドメイン名)の定義を削除します。このコマンドは、指定されたユーザの定義を削除します。<i>usrname</i> には、削除する既存のユーザの名前を指定します。</p> <p>このコマンドを実行するには、以下の準備が必要です。</p> <ul style="list-style-type: none"><li>■ グラフィカル・ユーザ・インターフェイスまたは <a href="#">tmloadcf(1)</a> のいずれかを使用して、アプリケーションを環境設定する。</li><li>■ TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する。</li><li>■ SECURITY を USER_AUTH、ACL、または MANDATORY_ACL に設定する。</li></ul> <p>アプリケーションがアクティブでない場合、tpusradd はコンフィギュレーションの MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。</p> |
| 移植性  | このコマンドは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 診断   | tpusrdel コマンドは正しく完了すると、終了コード 0 で終了します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 関連項目 | <a href="#">tpgrpadd(1)</a> 、 <a href="#">tpgrpdel(1)</a> 、 <a href="#">tpgrpmod(1)</a> 、 <a href="#">tpusradd(1)</a> 、 <a href="#">tpusrmod(1)</a>                                                                                                                                                                                                                                                                                                                                                                                |

## tpusrmod(1)

名前 tpusrmod - システムのユーザ情報を変更

形式 tpusrmod [-u *UID* ] [-g *GID*] [-c *client\_name*] [-l *new\_login*] [-p] *usrname*

機能説明 tpusrmod を実行すると、BEA Tuxedo セキュリティ・データ・ファイルのプリンシパル(ユーザまたはドメイン)のエントリが変更されます。この情報は、AUTHSVR(5) サーバでユーザ単位の認証に使用されます。

このコマンドを実行するには、以下の準備が必要です。

- グラフィカル・ユーザ・インターフェイスまたは [tmloadcf\(1\)](#) のいずれかを使用して、アプリケーションを環境設定する。
- TUXCONFIG ファイルを指すように TUXCONFIG 環境変数を設定する。
- SECURITY を USER\_AUTH、ACL、または MANDATORY\_ACL に設定する。

アプリケーションがアクティブでない場合、tpusradd はコンフィギュレーションの MASTER 上で実行する必要があります。アプリケーションがアクティブな場合は、このコマンドはどのアクティブなノードでも実行できます。

このコマンドで作成されたシステム・ファイル・エントリは 1 行の制限が 512 文字です。いくつもの(複数の)オプションに長い引数を指定するとこの制限を超えることがあります。

以下のオプションが使用できます。

-u *UID*

新しいユーザ識別番号を指定します。*UID* は、16K より小さい正の 10 進数でなければなりません。*UID* は、アプリケーションの既存の識別子のリスト中で一意でなければなりません。

-g *GID*

既存のグループの整数識別子またはグループ名の文字列を指定します。このオプションによって、ユーザが属するグループを再定義します。

`-c client_name`

ユーザに対するクライアント名を、表示可能な文字の重複しない文字列で指定します。コロン (:) または改行文字 (\n) を使うことはできません。

`-l new_login`

ユーザの新しいログイン名を、表示可能な文字列で指定します。コロン (:)、シャープ (#)、または改行文字 (\n) を使うことはできません。ユーザ名は、アプリケーション用の既存の識別子のリスト中で一意でなければなりません。また、このオプションを指定すると、`-p` オプションによってパスワードを再設定することになります。

`-p`

`tpusrmod` は、指定したユーザのパスワードを変更します。管理者は、そのユーザのための新しいパスワードを入力するように要求されます。

`usrname`

変更する既存のユーザの名前を、表示可能な文字列で指定します。

ユーザ単位の認証や管理者のパーミッションの設定の詳細については、[AUTHSVR\(5\)](#) を参照してください。

**移植性** このコマンドは、BEA Tuxedo サーバ環境がサポートされるプラットフォームで実行できます。

**診断** `tpusrmod` コマンドは正しく完了すると、終了コード 0 で終了します。

**関連項目** [tpgrpadd\(1\)](#)、[tpgrpdel\(1\)](#)、[tpgrpmod\(1\)](#)、[tpusradd\(1\)](#)、[tpusrdel\(1\)](#)、[AUTHSVR\(5\)](#)

『BEA Tuxedo アプリケーション実行時の管理』

## tuxadm(1)

名前 tuxadm - BEA Tuxedo Administration Console CGI ゲートウェイ

形式 `http://cgi-bin/tuxadm[TUXDIR=tuxedo_directory | INIFILE=initialization_file][other_parameters]`

機能説明 tuxadm は、ブラウザから Administration Console を初期化する際に使用される、コモン・ゲートウェイ・インターフェイス (CGI) プロセスです。「形式」に示されているように、このプログラムは 1 つのロケーションとして、または Web ブラウザからの URL としてのみ使用できます。通常、標準コマンド行プロンプトからは実行されません。tuxadm は、CGI プログラムが通常行うように、引数リストの構文解析に `QUERY_STRING` 環境変数を使用します。

tuxadm は引数を解析し、Administration Console 初期化ファイルを見つけます。TUXDIR パラメータが存在する場合、初期化ファイルはデフォルトで `$TUXDIR/udataobj/webgui/webgui.ini` に設定されます。INIFILE オプションが存在する場合は、パラメータの値は初期化ファイルへの絶対パスになります。その他のパラメータが存在する場合があります。

初期化ファイルの値を無効にするために追加のパラメータを使用することもできます。初期化ファイルのパラメータの完全なリストは、`wlisten` マニュアル・ページを参照してください。実際の初期化ファイルで許される値と一貫性のある上書きがされない限り、tuxadm プロセスによって `ENCRYPTBITS` パラメータが上書きされることはありません。

tuxadm の通常の動作は、Administration Console アプレットを起動する Web ページを作成するための HTML コマンドを、標準出力に生成することです。Web ページの一般的なフォーマットは、初期化ファイル内の `TEMPLATE` パラメータによって制御されます。初期化ファイルに含まれる HTML コマンドには、特殊文字列 `%APPLET%` を持つものがあり、この文字列の位置に Administration Console アプレットを埋め込みます。初期化ファイルからその他のパラメータ (`CODEBASE`、`WIDTH`、`HEIGHT` など) を使用することにより、Administration Console のインスタンスの作成に必要なすべてのパラメータを含む、正しい `APPLET` タグを生成できます。

エラー 異常終了した場合、tuxadm はエラー・メッセージを含む HTML コードを生成します。CGI プログラムの操作の性質上、tuxadm からエラー・コードが返ることはありません。

## セクション 1 - コマンド

---

関連項目 [tuxsvr\(1\)](#)、[wlisten\(1\)](#)

## tuxwsvr(1)

名前 tuxwsvr - BEA Tuxedo Administration Console で使用するミニ・ウェブ・サーバ

形式 tuxwsvr -l nlsaddr [-d device] [-L logfile] [-F]  
-i initialization\_file

機能説明 tuxwsvr は World Wide Web サーバ・プロセスです。これを使用すれば、顧客は BEA Tuxedo Administration Console プロセスが動作するマシン上に商用 Web サーバまたはパブリック・ドメイン Web サーバを設置していなくても、BEA Tuxedo Administration Console プロセスをサポートすることができます。tuxwsvr は、特にその他の指定がない限り、起動後はバックグラウンドで動作し、マシンがシャットダウンするか、オペレーティング・システム・コマンドを使用して tuxwsvr プロセスが強制終了されるまで実行を継続しません。

tuxwsvr は、BEA Tuxedo Administration Console のサポートに必要なすべての機能を含んでいますが、事前分岐プロセス、サーバ側の HTML インクルード (.shtml ファイル)、デフォルトのディレクトリ・インデックス、https 接続など、商用 Web サーバ用の多数の機能は含まれていません。ただし、BEA Tuxedo Administration Console は独自の暗号化プロトコルを実装しているため、https 接続がなくても安全なモードで実行することができます。性能上の理由から、一般的な Web サーバでは、受信された要求に対する DNS 逆ルックアップは実行されません。

tuxwsvr で使用されるコマンド行オプションには、以下のものがあります。

-l nlsaddr

プロセスが接続をリスンするネットワーク・アドレス。TCP/IP アドレスは次の形式で指定します。

```
"//hostname:port_number"
```

```
"//#.#.#.#:port_number"
```

最初の形式では、tuxwsvr は、ローカル名の解決機能 (通常 DNS) を使用して *hostname* のアドレスを見つけます。*hostname* はローカル・マシンでなければなりません。ローカル名解決機能によって *hostname* をローカル・マシンのアドレスに明確に解決します。

2 番目の例は、ドットで区切った 10 進数の形式 (#.#.#.#) です。ドット区切りの 10 進数形式では、各 # には 0 ~ 255 の数字を使用します。このドット区切りの 10 進数は、ローカル・マシンの IP アドレスを表します。

どちらの形式も *port\_number* は *tlisten* プロセスが入力要求に対してリスンする TCP ポート番号です。*port\_number* は 0 から 65535 までの数字または名前のどちらかです。*port\_number* が名前の場合は、ローカル・マシン上でネットワーク・サービス・データベースの中からその名前を見つけなければなりません。先頭に &ldquo;0x” を付けて、その “16 進数形式で指定できます。&ldquo;0x” の後の各文 “ ~ 9 の数字か、A ~ F までの英字 (大文字・小文字に関係なく) です。16 進数の形式は、IPX/SPX や TCP/IP のような任意のバイナリ・ネットワーク・アドレスに使うことができます。アドレスはまた、任意の文字列として指定することもできます。たとえば、文字列で指定したアドレスは STARLAN ネットワークで使用されます。

**-d device**

ネットワーク・デバイスの絶対パス名。リリース 6.4 以降では、このパラメータはオプションです。それ以前のリリースでは、*tcp* などの基本ネットワーク・プロバイダから要求される場合に使用します。

**-L logfile**

*tuxwsvr* が Web 要求とエラー・メッセージのログ記録用に使用するファイル名の接頭辞。実際のログファイルの名前は、月、日、年を表す 7 文字の文字列 *.mmddyy* をこの接頭辞に追加したものになります。このオプションを指定しない場合、Web サーバのログ・ファイルの接頭辞はカレント・ディレクトリの *WB* です。*tuxwsvr* プロセスを実行する各日の最初のログ・メッセージは新規ファイルに書き込まれます。

**-F**

*tuxwsvr* をバックグラウンドではなくフォアグラウンドで実行するよう指定します。このオプションは、主にテストとデバッグを行う際に便利です。*tuxwsvr* プロセスは、特にその他の指定がない限り、自動的にバックグラウンドで実行されます。コマンド行で後にアンパサンド (&) を付ける必要はありません。

`-i initialization_file`

初期化ファイルは、すべての `tuxwsvr` コマンド行で指定する必要があります。そのために使用するコマンド行オプションが `-i` です。次の節では、初期化ファイルの形式について説明します。

初期化ファイルの形式

初期化ファイルには、Web サーバで必要となるディレクトリへのマッピングが含まれます。また、いくつかのコメント行が含まれる場合もあります。コメント行の先頭には `#` 記号が付きます。コメント行以外の行は、空白で区切られた 3 つのフィールドで構成されます。

表 16 初期化ファイルの形式

| フィールド | 内容                                                                         |
|-------|----------------------------------------------------------------------------|
| 1     | HTML または CGI。この行で記述されたディレクトリ内のファイルの種類 (HTML ファイルまたは実行可能な CGI プログラム) を示します。 |
| 2     | パスの接頭辞。特定の要求が複数の接頭辞に一致する場合は、最初に一致した接頭辞が選択されます。                             |
| 3     | パスの接頭辞 (フィールド 2) のマッピング先となるディレクトリまたはファイル。                                  |

初期化ファイルでコメント行以外の最後の行には、接頭辞 `&slq;/` が必要です。これよりも前の行 `&slq;/` が付いて、警告メッセージが生成されます。

初期化ファイルを変更する際の注意

初期化ファイルは、起動時に 1 回読み取られます。したがって、このファイルに何らかの変更を加えた場合は、変更を有効にする前に `tuxwsvr` を停止して再起動する必要があります。

UNIX システムの初期化ファイルの例

次に UNIX システムの初期化ファイルの例を示します。

```
CGI /cgi-bin /home/tuxedo/udataobj/webgui/cgi-bin
CGI /webgui /home/tuxedo/udataobj/webgui/cgi-bin
HTML /java /home/tuxedo/udataobj/webgui/java
HTML /doc /home/tuxedo/doc
HTML / /home/tuxedo/udataobj/webgui
```

次のようなマシンのポート 8080 で Web サーバが稼動しているとします。

```
tuxmach.acme.com
```

次のいずれかの URL への要求を入力します。

```
http://tuxmach.acme.com:8080/cgi-bin/tuxadm?TUXDIR=/home/tuxedo
```

```
http://tuxmach.acme.com:8080/webgui/tuxadm?TUXDIR=/home/tuxedo
```

この要求により、以下の 2 つの処理が行われます。

- 次のプログラムが起動されます。

```
/home/tuxedo/udataobj/webgui/tuxadm
```

- World Wide Web CGI 仕様に記述されているように、環境変数 `QUERY_STRING` がプログラムの `TUXDIR=/home/tuxedo` に設定されます。

初期化ファイルの CGI ディレクトリの値として `$TUXDIR/bin` を指定することはお勧めできません。このように指定すると、Web ユーザが BEA Tuxedo の実行可能ファイルを起動してしまう可能性があります。ただし、このようなユーザによって、`tuxadm` 以外の実行可能ファイルの結果が参照されることはありません。これらの実行可能ファイルは CGI プログラムとして記述されていないためです。

また、前の例では、最初の HTML 行は、2 つ目の HTML 行でサブディレクトリの `/java` を同じパスにマッピングしているため重複しています。この最初の行を含める理由は、ユーザによっては HTML ドキュメントを格納している場所とは別の場所に Java クラス・ファイルを配置することがあるためです。

Windows の  
初期化ファ  
イルの例

次に Windows システムの初期化ファイルの例を示します。

```
HTML /tuxedo/webgui D:\\tuxedo\\htmldocs
CGI /cgi-bin C:\\cgi-bin
HTML /java D:\\tuxedo\\udataobj\\webgui\\java
HTML / D:\\tuxedo\\udataobj\\webgui
```

マシン `ntsvr1` のポート 80 で Web サーバが稼動しているとします。次の URL を入力します。

```
http://ntsvr1/tuxedo/webgui/page1.html
```

次のファイルが検索されます。

D:\\tuxedo\\htmldocs\\page1.html

おそらくこのファイルは、Administration Console を起動する、顧客によって作成されたページです。

**プロセスの終了** `tuxwsvr` プロセスを正常に終了する唯一の方法は、プロセスに SIGTERM シグナルを送ることです。

**推奨される使用方法** `tuxwsvr` プロセスは、商用 Web サーバを設置していない顧客向けに、BEA Tuxedo 管理 GUI 用の Web サーバとして提供されています。UNIX システムの場合は、次の形式のコマンド行を UNIX の初期化スクリプトに追加して、Web サーバが自動的に起動するようにすることをお勧めします。

```
TUXDIR=tuxdir_path_name $TUXDIR/bin/tuxwsvr -l nlsaddr -i
initialization_file
```

`tuxdir_path_name` は、BEA Tuxedo システム・ソフトウェアの場所への絶対パス名を表します。`nlsaddr` は、この `tuxwsvr` プロセスによって使用される、ネットワーク上のアドレスです。

`tuxwsvr` プロセスを起動する別の方法として、上記の推奨コマンド行を使用して手動で起動する方法があります。また、`cron` ジョブを使用して定期的（毎日またはそれより頻繁）に `tuxwsvr` プロセスを起動する方法もあります。同じネットワーク・アドレスを使用して `tuxwsvr` コマンドを重複して呼び出すと、コマンドは自動的に終了し、状況に応じてメッセージが記録されません。

**ネットワーク・アドレス** アプリケーション管理者が `tuxwsvr` プロセスにネットワーク・アドレスを指定する場合、指定されたネットワーク上で一意なネットワーク・アドレスを指定しなければならないという制約が 1 つだけあります。STARLAN ネットワークの場合は、推奨するアドレス `uname.tuxwsvr` が通常一意の名前となります。TCP/IP の場合は、アプリケーション管理者が選択した一意なポートとローカル・マシンのノード識別子をペアにした `0x0002ppppnnnnnnnnnn` の形式のアドレスになります。特定のマシン (`pppp`) で使用する一意なポート値は、ネットワークとマシンの組み合わせを考慮してユーザ間で取り決める必要があります。低いポート番号はシステム関連のサービスで使用されることが多いため、一般には高いポート番号が有効です。ノード・フィールド (`nnnnnnnn`) に適切な値は、次の手順を実行して `/etc/hosts` ファイルから見つけることができます。

1. `uname -n` を入力します。

`node_name` が返されます。

2. `grep node_name /etc/hosts` を入力します。

`182.11.108.107 node_name` が返されます。

3. ドット表記を 8 桁の 16 進数表記に変換します。

ネットワー  
ク・アドレ  
スの例

たとえば、`tuxwsvr` を実行しているローカル・マシンで TCP/IP アドレス指定機能を使用している場合を考えます。マシン名は `backus.company.com` で、アドレスは `155.2.193.18` です。`tuxwsvr` はポート番号 `2334` で要求を受け取ります。また、ポート番号 `2334` は、`bankapp-tuxwsvr` という名前のネットワーク・サービス・データベースに追加されているとします。`-l` オプションで指定するアドレスは次に示す方法で表現できます。

- `//155.2.193.18:bankapp-tuxwsvr`
- `//155.2.193.18:2334`
- `//backus.company.com:bankapp-tuxwsvr`
- `//backus.company.com:2334`
- `0x0002091E9B02C112`

最後に示す行は、アドレスを 16 進数形式で表したものです。`0002` は TCP/IP アドレスの先頭部分、`091E` はポート番号 `2334` を 16 進数に変換したもの、および `9B02C112` は IP アドレス `155.2.193.18` を 16 進数に変換したものです。最後の交換部分については、`155` は `9B`、`2` は `02` のように変換されています。

STARLAN ネットワークの場合は、通常、推奨アドレス `uname.tuxwsvr` が一意の名前になります。

関連項目 [tuxadm\(1\)](#)、[wlisten\(1\)](#)

## txrpt(1)

名前 txrpt - BEA Tuxedo ATMI システム・サーバ/サービス・レポート・プログラム

形式 txrpt [-t] [-n *names*] [-d *mm/dd*] [-s *time*] [-e *time*]

機能説明 txrpt は BEA Tuxedo ATMI システム・サーバの標準エラー出力を分析して、そのサーバ内でのサービス処理時間の要約を示します。その報告は、扱われた時間においてディスパッチされた回数および各サービスの平均所要時間(秒)を示します。txrpt はその入力を標準入力または入力として切り換えられた標準エラー出力ファイルからの入力を読み込みます。標準エラー出力ファイルは、[servopts\(5\)](#) の選択肢の中から `-r` オプションを使って呼び出されたサーバによって作成されます。`-e servopts` オプションを付けて指定することにより、ファイルに名前を付けることができます。複数のファイルを、txrpt 用に 1 つの入力ストリームに連結することができます。txrpt の各オプションには、次の意味があります。

`-t`

サービスの合計使用時間の最も長いものから順に出力レポートを生成します。指定がなければ、このレポートはサービスの合計呼び出し回数によって配列されます。

`-n names`

*names* で指定されたサービスに対するレポートだけを生成します。ここで、*names* は複数のサービス名をカンマで区切ったリストです。

`-d mm/dd`

サービス要求への報告を指定された月 (*mm*) と日 (*dd*) に限定します。デフォルト設定は、現在の日付です。

`-s time`

指定時刻 (*time*) 以降に行われた呼び出しのレポートを生成します。*time* の形式は `hr[:min[:sec]]` です。

`-e time`

指定時刻 (*time*) 以前に終了した呼び出しのレポートを生成します。*time* の形式は、`-s` フラグと同じです。

txrpt が生成するレポートは 1 日単位のものです。入力ファイルに複数日のレコードがある場合、`-d` オプションは、レポートする日付を制御します。

## セクション 1 - コマンド

---

**注意事項** サーバが `txrpt` を介して分析統計情報を収集する場合、`ULOGDEBUG` 変数を "y" には設定しないようにしてください。ファイル内のデバッグ・メッセージを `txrpt` が間違っって解釈してしまいます。

**使用例** 次に示すコマンド行を実行します。

```
txrpt -nSVC1 -d10/15 -s11:01 -e14:18 newr
```

作成されるレポートは次のようになります。

```
START AFTER: Thu Oct 15 11:01:00 1992
END BEFORE: Thu Oct 15 14:18:00 1992
SERVICE SUMMARY REPORT
```

| SVCNAME | 11a-12n | 13p-14p | 14p-15p | TOTALS  |
|---------|---------|---------|---------|---------|
|         | Num/Avg | Num/Avg | Num/Avg | Num/Avg |
| -----   | -----   | -----   | -----   | -----   |
| SVC1    | 2/0.25  | 3/0.25  | 1/0.96  | 6/0.37  |
| -----   | -----   | -----   | -----   | -----   |
| TOTALS  | 2/0.25  | 3/0.25  | 1/0.96  | 6/0.37  |

上記の例は、`svc1` が指定された時間内に全部で 6 回要求されたことを示しています。要求を処理する時間は、平均 0.37 秒でした。

**関連項目** [servopts\(5\)](#)

## ud、wud(1)

名前 ud、wud - BEA Tuxedo ATMI ドライバ・プログラム

形式 ud [-p] [-ddelay] [-eerror\_limit] [-r] [-ssleeptime] [-ttimeout]  
 [-n] [-u {n | u | j}] [-Uusrname] [-Ccltname] [-Sbufferize]  
 ud32 [options]  
 wud [options]  
 wud32 [options]

機能説明 ud は、`Fextread()` を使用して標準入力から入力パケットを読み取ります。詳細については、[Fextread](#)、[Fextread32\(3fml\)](#) を参照してください。このパケットには、サービス名を識別するフィールドが含まれていなければなりません。入力パケットは、FML フィールド化バッファ (FBFR) に送られてから該当サービスに送られます。FBFR を受け取るサービスがレコードをデータベースに追加するものである場合、大量のフィールド化データを BEA Tuxedo ATMI システムが認識しているデータベースに入力するために ud を使用できます。

入力パケットの行の先頭を示すフラグ (「入力形式」を参照) を使用することにより、ud を BEA Tuxedo ATMI サービスのテストに利用できます。

デフォルトの設定では、FBFR をサービスに送った後、ud は FBFR が返されることを期待します。送信および応答 FBFR は ud の標準出力に出力されます。エラー・メッセージは標準エラー出力に返されます。

ud32 は、FBFR32 タイプの FML32 バッファを使用します。

wud および wud32 はワークステーション・ライブラリを利用して構築されたバージョンの ud と ud32 です。/WS だけをサポートしているサイトでは、wud および wud32 コマンドだけが提供されています。

オプション ud は次のようなオプションをサポートしています。

- p 送信されたまたは返されたフィールド化バッファの表示を行いません。
- d 各要求の遅延応答を期待します。delay は、タイムアウトまでの最大遅延時間を秒単位で指定します。タイムアウトになると、stderr にエラー・メッセージが出力されます。ud が指定された遅延時間内に以前

の要求に対する応答メッセージを受け取った場合には、それらのメッセージは遅延 RTN パケットとして示されます。このため、遅延時間内に複数の応答パケットを受け取ることも可能です。wud 用の `-d` オプションは、DOS オペレーティング・システム上では動作しません。

`-e error_limit`

エラー回数が、`error_limit` で指定されている制限回数を上回ると、ud は、要求に対する処理を停止します。制限が指定されない場合のデフォルト設定は 25 です。

`-r`

ud は、サーバからの応答メッセージを期待しません。

`-s sleeptime`

入力バッファの送信と送信の間、スリープ状態になります。`sleeptime` は、スリープ時間を秒単位で指定します。

`-t timeout`

ud は、トランザクション・モードで要求を送ります。`timeout` は、トランザクションがタイムアウトするまでの時間 (秒) です。オプション `-d delay` と `-r` (応答なし) は、`-t` オプションと一緒に使用することはできません。

`-u {n|u|j}`

新しいパケットを読み取る前に要求バッファの内容を修正する方法を指定します。`n` オプションは、バッファを初期化し直すことを表します (新しいバッファとして扱う)。`u` オプションは、`Fupdate()` を使用してバッファを応答バッファで更新することを指定します。`j` オプションは、`Fojoin()` を使用して応答バッファを要求バッファに結合することを指定します。詳細については、`Fupdate`、`Fupdate32(3fml)` と `Fojoin`、`Fojoin32(3fml)` を参照してください。

`-n`

各パケットを読み取る前にバッファを再初期化します。つまり、各バッファを新しいバッファとして扱います。このオプションは `-un` と同等の働きをしますが、互換性を維持する目的で用意されています。

`-U usurname`

`usurname` は、アプリケーションに参加するときのユーザ名として使用します。

`-S buffersize`

デフォルトのバッファサイズが十分に大きくない場合、この `-s` オプションはその容量を拡大するのに使用できます。`buffersize` の値には、`MAXLONG` までの任意の数字を指定できます。

オプション `-d delay` と `-r` は相互に排他的です。

入力形式 入力パッケージは下記のような形式の行で構成されます。

`[flag]fldname fldval`

`flag` は省略可能です。`flag` を指定しなかった場合には、`fldname` で指定され、値 `fldval` をもつフィールドの新しいオカレンスがフィールド化バッファに追加されます。`flag` を指定する場合は、次のいずれかとします。

+

FBFR 内の `fldname` のオカレンス 0 を `fldval` に変更します。

-

`fldname` のオカレンス 0 を FBFR から削除します。タブ文字は必須で、`fldval` は無視されます。

=

`fldname` の値が変更されます。この場合、`fldval` はフィールド名を指定し、その値が `fldname` で指定されるフィールドに割り当てられます。

#

この行はコメントとして扱われ、無視されます。

`fldname` がリテラル値 `SRVCNM` であると、`fldval` は FBFR を受け取るサービスの名前となります。

長いフィールド値は、継続行の先頭にタブを挿入することにより、次の行にまたがって入力できます。

改行文字だけで構成される行は入力の終わりを示し、そのパッケージを `ud` に送ります。

入力パッケージが文字 `n` と改行文字で構成される行で始まる場合、FBFR は初期化し直されます。FBFR の再初期化は、コマンド行に `-un` オプションを使用することですべてのパッケージに指定することができます。

表示不能文字を入力パケットに入力するには、エスケープ文字の後に目的の文字を表す 16 進表現を入力します。詳細については、UNIX リファレンス・マニュアルの `ascii(5)` を参照してください。シェルからエスケープ文字を保護するには、追加のバックスラッシュが必要です。たとえば、スペースは `\20` と入力データに入力できます。`ud` はこの形式の全入力を認識しますが、その最大の利点は非表示文字を入力する際です。

- 処理モデル** 当初、`ud` はフィールド化バッファをその標準入力から読み取って、`fldname` が `SRVCNM` である行の `fldval` によって名前が与えられているサービスにその入力を送ります。`-r` オプションが選択されていない限り、`ud` は応答フィールド化バッファが送られるまで待機します。応答が得られると、`ud` は次のフィールド化バッファを標準入力から読み取ります。`ud` は、このようにするために、返されるバッファを現在のバッファとして保持します。つまり、2 番目のフィールド化バッファを形成する標準入力上の行が、返されたバッファへの追加情報として扱われます。つまり、デフォルトのアクションでは、`ud` は一連の入力行によってその内容が追加される現在のバッファを維持します。これらの入力行は、空白行で区切られます。コマンド行で `-un` オプションを指定するか、あるいは入力セットの先頭行として英字 `n` を唯一の文字とする行を含めることにより、`ud` に現在のバッファを廃棄する（つまり、その `FBFR` 構造を初期化し直す）よう指示できます。また、`-uu` オプション（`Fupdate` を使用）あるいは `-uj` オプション（`Fojoin` を使用）を指定することにより、`ud` に応答バッファの内容を要求バッファにマージさせることができます。
- セキュリティ** `ud` を保護アプリケーションで実行する場合、そのアプリケーションを利用するためにはアプリケーション・パスワードが必要です。標準入力端末である場合、`ud` はユーザにパスワードの入力を求めてきます。このとき、ユーザが入力パスワードは画面には表示されません。ただし、`ud` は標準入力からの一括入力を受け付けるので、標準入力は一般的に端末ではなくファイルとなっています。このケースでは、パスワードは環境変数 `APP_PW` から取り出されます。この環境変数が指定されていないときに、アプリケーション・パスワードが必要であると、`ud` は異常終了します。
- 移植性** これらのコマンドは、BEA Tuxedo ATMI サーバ環境がサポートされるプラットフォームで実行できます。

**環境変数** `FLDTBLDIR` および `FIELDTBLS` を設定してエクスポートする必要があります。  
`FLDTBLDIR` のディレクトリのリストには、`$TUXDIR/udataobj` が含まれていなければなりません。`FIELDTBLS` には、フィールド・テーブルの1つとして `Usysflds` が含まれていなければなりません。

標準入力端末でない場合、保護アプリケーションでは `APP_PW` をアプリケーション・パスワードに設定しなければなりません。`TPIDATA` は、標準入力が端末でない場合、認証サーバで保護アプリケーションにアプリケーションを結合するのに必要なそのアプリケーション特定のデータに設定しなければなりません。

また、ワークステーションからアクセスする場合には、`WSNADDR`、`WSDEVICE` および `WSTYPE` (省略可) を設定しなければなりません。クライアント・プロセス用に環境変数を設定する方法についての詳細は、[compilation\(5\)](#) を参照してください。

**診断** `ud` は、クライアント・プロセスになれないとき、必要な `FBFR` を作成できないとき、あるいは UNIX システムのエラーを検出したときには異常終了します。また、入力パケットのストリーム処理時に 25 個を超えるエラーを検出すると異常終了します。これらのエラーとしては、構文エラー、サービス名の欠落、トランザクションの開始またはコミットのエラー、入力 `FBFR` 送信時あるいは応答 `FBFR` 受信時のタイムアウトまたはエラーなどがあります。

**注意事項** 入力フィールドの最後のフィールド化バッファは空白行で終わってください。

#### 使用例

```
$ud <EOF>
SRVCNM BUY
CLIENT J. Jones
ADDR 21 Valley Road
STOCK AAA
SHARES 100
<CR>
+SRVCNM SELL
+STOCK XXX
+SHARES 300
STOCK YYY
SHARES 150
<CR>
n
SRVCNM BUY
CLIENT T. Smith
ADDR 1 Main Street
STOCK BBB
```

## セクション 1 - コマンド

---

```
SHARES 175
<CR>
+SRVCNM SELL
+STOCK ZZZ
+SHARES 100
<CR>
EOF
$
```

この例で、`ud` はまず、フィールド化バッファをサービス `BUY` に送り、`CLIENT` フィールドを `J. Jones` に、`ADDR` フィールドを `21 Valley Road` に、`STOCK` フィールドを `AAA` に、そして `SHARES` フィールドを `100` に設定します。

フィールド化バッファが `BUY` サービスから返されると、`ud` は、次の行セットを使用して `SRVCNM` を `SELL` に、`STOCK` を `XXX` に、`SHARES` を `300` に変更します。また、`STOCK` フィールドの追加オカレンス値を `YYY` で、`SHARES` フィールドの追加オカレンス値を `150` で作成します。このフィールド化バッファはその後、`SELL` サービス (`SRVCNM` フィールドの新しい値) に送られます。

`SELL` が応答フィールド化バッファを返してくると、`ud` は次の行セットを唯一の文字 `n` で構成される行で開始することにより、そのバッファを廃棄します。次に `ud` は、全く新しい入力パケットを、`SRVCNM` を `BUY` に、`CLIENT` を値 `T. Smith` に設定して構築します。

関連項目 [Fextread](#)、[Fextread32\(3fml\)](#)、[compilation\(5\)](#)

UNIX System V リファレンス・マニュアルの [ascii\(5\)](#)

『C 言語を使用した BEA Tuxedo アプリケーションのプログラミング』

『FML を使用した BEA Tuxedo アプリケーションのプログラミング』

『BEA Tuxedo アプリケーション実行時の管理』

## uuidgen(1)

名前 uuidgen - Universal Unique Identifier (UUID) の作成

形式 uuidgen [-o filename] [{-i | -n number}] [-v] [-h] [-?]

機能説明 uuidgen は、デフォルトで標準出力に Universal Unique Identifier (UUID) を出力します。UUID は、IDL インターフェイス定義を一意に識別するのに使用されます。UUID 文字列の形式は、8 桁の 16 進数、その後ダッシュ、ダッシュで区切られた 3 つのグループから成る 4 桁の 16 進数、さらにその後ダッシュと 12 桁の 16 進数が続く形式です (後述の「使用例」を参照)。

uuidgen(1) には次のオプションがあります。

-i

UUID 文字列 (形式については「使用例」を参照) を含む、IDL ファイルのテンプレートを作成します。このオプションは -n オプションと一緒に使用できません。

-n number

指定した数の UUID 文字列を作成します。このオプションは -i オプションと一緒に使用できません。

-o filename

指定したファイルに出力をリダイレクトします。

-v

uuidgen(1) のバージョン番号を表示しますが、UUID 文字列は作成しません。

-h または -?

uuidgen(1) のヘルプ情報を表示します。

ネットワーク・アドレス UUID の作成には、48 ビットの IEEE 802 アドレスを利用する必要があります。このアドレスはすべての環境で利用できるものではなく、またアドレス決定方法に移植性がある訳でもありませんが、BEA Tuxedo システムの uuidgen では複数の手法が利用できます。

- NADDR 環境変数が次の形式の値に設定されていると、

num.num.num.num

インターネット形式のアドレスと解釈され、それに応じて変換されます。

- そうでない場合、`WSNADDR` 環境変数が次の形式の値に設定されていると、  
`0xxxxxxxxxxxxxxxxxxx`  
ワークステーションで使用される 16 進数のネットワーク・アドレスと解釈されます。
- 上記のどちらでもなく、また DOS を使用するのであれば、マシンの `uname` を使用して `/etc/hosts` ファイル内でマシン・エントリを探し、インターネット形式のアドレスを取得します。
- 上記以外の場合、警告メッセージが表示され、`00.00.00.00` が使用されます。

上記のいずれの場合でも、32 ビットのアドレスが形成され、(48 ビットの) 残りのアドレスは `00.00` として処理されます。

**診断** 無効なコマンド行オプションが指定された場合、または出力ファイルを開くできなかった場合には、`uuidgen` はゼロ以外の終了コードで終了します。不正なネットワーク・アドレスが与えられたり、`00.00.00.00` が使用された場合には、警告メッセージが表示されます。

**使用例** UUID 文字列を生成するには、次のコマンドを入力します。

```
uuidgen
23C67E00-71B6-11C9-9DFC-08002B0ECEF1
```

インターフェイス定義を生成するために IDL テンプレートを作成するには、次のコマンドを入力します。

```
uuidgen -i
[uuid(B5F8DB80-3CCA-14F8-1E78-930269370000)]
interface INTERFACE
{
}
```

2 つの UUID 文字列を生成するには、次のコマンドを入力します。

```
uuidgen -n 2
C0B37080-3CCA-14F8-265F-930269370000
C0B37081-3CCA-14F8-2CDB-930269370000
```

**関連項目** [tidl\(1\)](#)

## viewc、viewc32(1)

名前 viewc、viewc32 - BEA Tuxedo ATMI VIEW を生成する VIEW コンパイラ

形式 viewc [-n] [-d *viewdir*] [-C] *viewfile* [*viewfile* . . . ]  
viewc32 [-n] [-d *viewdir*] [-C] *viewfile* [*viewfile* . . . ]

機能説明 viewc は VIEW コンパイラ・プログラムです。ソース VIEW ファイルを取り込み、以下のファイルを作成します。

- 実行時に解釈されて、FML バッファと C 構造体との間で実際のマッピングを行うバイナリ・ファイル。
- 1 つ以上のヘッダ・ファイル。
- COBOL コピー・ファイル ( 必要な場合 )。viewc の実行時には、C コンパイラが使用可能でなければなりません。

viewc32 は 32 ビット FML で使用されます。環境変数 `FLDDBLS32` および `FLDDBLDIR32` を使用します。

*viewfile* は、VIEW ソース記述を収めているファイルです。複数の *viewfile* を viewc コマンド行に指定できます。ただし、同じ VIEW 名が複数の *viewfile* に使用されていない場合に限りです。

デフォルトの設定では、*viewfile* 内のすべての VIEW がコンパイルされ、複数のファイルが生成されます。すなわち、VIEW オブジェクト・ファイル ( 接尾辞は `.v` ) とヘッダ・ファイル ( 接尾辞は `.h` ) です。オブジェクト・ファイルの名前は、`-d` によって代わりのディレクトリが指定されない限り、*viewfile.v* です。また、ヘッダ・ファイルはカレント・ディレクトリに作成されます。

`-c` オプションを指定すると、*viewfile* で定義している VIEW 1 つにつき 1 つの COBOL コピー・ファイルが作成されます。これらのコピー・ファイルはカレント・ディレクトリに作成されます。

viewc によるコンパイル時には、*viewfile* に指定されているフィールド識別子とフィールド名を、フィールド・テーブル・ファイルから得られる情報と対比し、マッピング情報をオブジェクト・ファイルに格納します。したがって、VIEW が FML バッファに基づいている場合には、環境変数 `FLDDBLS` と `FLDDBLDIR` を設定およびエクスポートして、関連するフィールド・テーブ

ル・ファイルを指すようにすることが重要です。FIELDTBLS と FLDTBLDIR の詳細については、『FML を使用した BEA Tuxedo アプリケーションのプログラミング』と『C 言語を使用した BEA Tuxedo アプリケーションのプログラミング』を参照してください。

viewc コンパイラがフィールド名とそのフィールド識別子の対応をとれなかった場合 (環境変数が正しく設定されていなかったり、フィールド・テーブル・ファイルにフィールド名が含まれていなかった場合)、警告メッセージ Field not found が表示されます。

-n オプションを使用すれば、FML バッファにマッピングされていない C 構造体に対応する VIEW 記述ファイルを作成することができます。『C 言語を使用した BEA Tuxedo アプリケーションのプログラミング』では、このような独立した VIEW 記述ファイルの作成および使用方法について説明していません。

viewc は、次のオプションを解釈します。

-n  
FML バッファにマッピングされていない C 構造体に対応する VIEW 記述ファイルをコンパイルするときに使用します。このオプションを使用すると、VIEW コンパイラは FML 情報を検索しません。

-d viewdir  
VIEW オブジェクト・ファイルをカレント・ディレクトリ以外で作成するために指定します。

-C  
COBOL コピー・ファイルを作成するよう指定します。

環境変数 CC

viewc は、通常、デフォルトの C 言語コンパイル・コマンドを使用してクライアントの実行可能ファイルを生成します。デフォルトの C 言語コンパイル・コマンドは、サポートされているオペレーティング・システムごとに定義されており、UNIX システムの場合は cc(1) です。別のコンパイラを指定可能にするため、viewc は環境変数 CC の有無を調べます。CC が viewc の環境に存在しない場合、または文字列 "" である場合、viewc はデフォルトの C 言語コンパイラを使用します。環境内に CC が存在する場合、実行されるコンパイラの名前が CC の値となります。

## CFLAGS

環境変数 `CFLAGS` は、コンパイラ・コマンド行の一部として引き渡される引数のセットを指定するときに使用します。`CFLAGS` が `viewc` の環境に存在しない場合、または文字列 "" である場合、`buildclient` はコンパイラ・コマンド行の引数を追加しません。

**移植性** 出力 VIEW ファイルは、マシンおよびコンパイラに依存するバイナリ・ファイルです。あるマシン上で特定のコンパイラを使用してビューを生成した場合、その VIEW ファイルを別のタイプのマシン上で使用することはできません。また、パディングやパッキングなど、生成される構造体オフセットが異なるコンパイラで使用することもできません。

以下の追加オプションが認識されます。

`-c { m | b }`

使用する C コンパイル・システムを指定します。Microsoft C コンパイラでは `m` です。Microsoft C コンパイラがこのオプションの省略値です。`-c` オプションは Windows でのみサポートされます。

`-1 filename`

パス 1 を実行し、作成されるバッチ・ファイルに `filename.bat` という名前を付けます。このファイルが作られた後、パス 2 を実行する前に `filename.bat` を実行する必要があります。パス 1 とパス 2 を使用すると、コンパイルできる VIEW のサイズを拡大できます。

`-2 filename`

パス 1 の出力を使用して、処理を完了するためにパス 2 を実行するように指定します。

**関連項目** 「[FML 関数の紹介](#)」

『C 言語を使用した BEA Tuxedo アプリケーションのプログラミング』

### viewdis、viewdis32(1)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 名前   | viewdis、viewdis32 - バイナリ VIEW ファイル対応の VIEW 逆アセンブラ                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 形式   | viewdis viewobjfile .. . viewdis32 viewobjfile . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 機能説明 | <p>viewdis は VIEW コンパイラが生成した VIEW オブジェクト・ファイルを逆アセンブルし、VIEW ファイル形式で VIEW 情報を表示します。また、対応する構造体メンバのオフセットも表示します。</p> <p>1 つまたは複数の <i>viewobjfiles</i> ( 接尾辞は <i>.v</i> ) をコマンド行に指定できます。デフォルトの設定では、カレント・ディレクトリの <i>viewobjfile</i> が逆アセンブルされます。viewobjfile がない場合、エラー・メッセージが表示されます。</p> <p><i>viewobjfile</i> にある情報は、VIEW ファイルにあるフィールド識別子とフィールド名の整合により、フィールド・テーブル・ファイルの情報と共に取得されたため、環境変数 <code>FIELDTBLS</code> および <code>FLDTBLDIR</code> の設定およびエクスポートを行うことが重要になります。</p> <p>viewdis の出力は元の VIEW 記述と同じように見え、主にコンパイル後のオブジェクト VIEW 記述が正しいかどうかを検証するときに使用します。</p> <p>viewdis32 は、32 ビットの FML に使用します。環境変数 <code>FIELDTBLS32</code> および <code>FLDTBLDIR32</code> を使用します。</p> |
| 関連項目 | <p><a href="#">viewc、viewc32(1)</a></p> <p>『FML を使用した BEA Tuxedo アプリケーションのプログラミング』</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

## wlisten(1)

名前 wlisten - BEA Tuxedo Administration Console リスナ・プロセス

形式 wlisten [-i *initialization\_file*]

機能説明 wlisten は、Administration Console アプレットからの着信接続を受信して、Administration Console ゲートウェイ・プロセス (*wgated*) を開始するリスナ・プロセスです。wlisten のオプションはすべて、-i オプションで指定する初期化ファイルから受け継がれます。-i オプションが指定されていない場合は、`$TUXDIR/udataobj/webgui/webgui.ini` がデフォルトの初期化ファイルになります。初期化ファイルで使用できる形式とパラメータについては後で説明します。デフォルトの初期化ファイルは、システムのインストール時に生成されます。

wlisten は、初期化ファイルに `FOREGROUND=Y` パラメータが含まれていない限り、起動後はバックグラウンドで動作し、マシンがシャットダウンされるか、wlisten プロセスがオペレーティング・システム・コマンドによって強制終了されるまで実行を継続します。

wlisten では、次のコマンド行オプションを使用します。

-i *initialization\_file*

Administration Console セッションで使用するパラメータに対して指定された *initialization\_file* を、wlisten で使用するよう指定します。初期化ファイルの形式については後で説明します。初期化ファイルのパラメータのほとんどは、BEA Tuxedo システムのインストール時に妥当な値に設定されます。このオプションがコマンド行に指定されていない場合、デフォルトの初期化ファイルの場所は `$TUXDIR/udataobj/webgui/webgui.ini` になります。

初期化ファイル -i オプションで指定する初期化ファイルにはパラメータが含まれており、アプレット、wlisten プロセス、およびゲートウェイ・プロセスはこれらのパラメータを使用して、Administration Console の接続と以降の操作に必要な設定情報を調整することができます。

初期化ファイルのパラメータのほとんどは、BEA Tuxedo システムのインストール時に設定されます。また、Administration Console の実行中、ユーザ入力に応じて、ほかのパラメータが自動的に追加されることもあります。たとえば、ドメインに接続すると、コンソールはそのドメインのリストを初期化

ファイルに追加します。次回からは、プルダウン式 [Domain] メニューを使用すると、そのドメインがリストされるようになります。このため、初期化ファイルを明示的に編集していないにもかかわらず、ファイルに行が追加されたり変更されることがあります。

初期化ファイルは、コメント行（空白行または # 文字が先頭にある行）とキーワード行から構成されます。キーワード行の形式は `keyword=value` です。以下に、有効なキーワードとその値を示します。

`TUXDIR=directory`

BEA Tuxedo ソフトウェアがインストールされるディレクトリ。このパラメータにはデフォルト値はなく、値を割り当てる必要があります。-i オプションが `wlisten` に指定されていない場合は、`TUXDIR` を環境内で設定する必要があり、通常は初期化ファイルに指定されている値に設定します。

`NADDR=network_address`

`wlisten` で使用するネットワーク・アドレスを指定します。このパラメータにはデフォルト値はなく、値を割り当てる必要があります。ネットワーク・アドレスの形式は、`tlisten` およびその他の BEA Tuxedo コマンドで使用する形式と同じです。詳細については、後述の「ネットワーク・アドレス」を参照してください。

`DEVICE=device`

`wlisten` で使用するネットワーク・デバイスを指定します。この変数はオプションです。バージョン 6.4 以前のリリースでは、ネットワーク・デバイスが選択されていないことを示す空文字列がデフォルト値に設定されています。Microsoft Windows などの一部のシステムでは、この設定が適切です。ここでは、`tlisten` の `-d` オプションに使用する値と同じ値を使用します。UNIX システムによっては、`/dev/tcp` を指定する場合もあります。この値を割り当てるかどうかは、オペレーティング・システムによって異なります。

`FOREGROUND=[Y | N]`

`wlisten` をフォアグラウンドで実行する必要があるかどうかを指定します。デフォルト値は `N` で、`wlisten` は自動的にバックグラウンドで実行されます。このオプションは、テストとデバッグのみを目的として使用します。

`WIDTH=pixels` と `HEIGHT=pixels`

アプレットの幅と高さをそれぞれ指定します。この領域は、セキュリティ機能が有効になっている場合にパスワード入力を求めるプロンプトに使用されます。デフォルト値はそれぞれ 400 と 150 です。

`FRAMEWIDTH=pixels` と `FRAMEHEIGHT=pixels`

管理タスクを実行するメイン・アプレット・ウィンドウの幅と高さをそれぞれ指定します。デフォルト値はそれぞれ 750 と 550 です。

`ENCRYPTBITS=[0 | 40]`

ゲートウェイおよびアプレット接続で使用する暗号化モードを設定します。デフォルト値は 0 で、暗号化を使用しません。40 オプションを選択すると、40 ビット RC4 暗号化が使用されます。この場合、`tlisten` パスワード・ファイルが必要で、暗号化キーを交換するために認証を行う必要があります。

`DOCBASE=document_root`

BEA Tuxedo Administration Console ヘルプ・ファイルがあるドキュメント・ベースを指定します。このパラメータは BEA Tuxedo システムのインストール時に設定され、通常は以後この値を変更しません。このパラメータにデフォルト値はなく、初期化ファイルで値を割り当てる必要があります。

`CODEBASE=applet_root`

BEA Tuxedo Administration Console アプレット・ファイルがあるコード・ベースの URL を指定します。このパラメータは BEA Tuxedo システムのインストール時に設定され、通常は以後この値を変更しません。このパラメータにデフォルト値はなく、初期化ファイルで値を割り当てる必要があります。

`SNAPDIR=snapshot_directory`

ユーザ・ログ・スナップショット・ファイルとイベント・ログ・スナップショット・ファイルが格納されるサーバ・ディレクトリ・パスを指定します。SNAPDIR の値は、URL ではなく絶対パス名です。このパラメータは BEA Tuxedo システムのインストール時に設定され、通常は以後この値を変更しません。このパラメータにデフォルト値はなく、初期化ファイルで値を割り当てる必要があります。

`SNAPBASE=http_root`

ユーザ・ログ・スナップショット・ファイルとイベント・ログ・スナップショット・ファイルが格納される URL ベースを指定します。

SNAPBASE の値は、絶対パス名ではなく URL です。このパラメータは BEA Tuxedo システムのインストール時に設定され、通常は以後この値を変更しません。このパラメータにデフォルト値はなく、初期化ファイルで値を割り当てる必要があります。

TEMPLATE=*template\_path*

起動時に Administration Console アプレットをユーザに配布するために使用するテンプレート・ファイルのパス名を指定します。テンプレート・ファイルには、Administration Console アプレットを埋め込むファイル内の位置に、文字列 %APPLET% を単独で含める必要があります。ファイルの残りの部分は、通常は命令、ロゴ、または Administration Console 管理者が使用するその他の情報を含む、標準の HTML 形式ファイルでなければなりません。デフォルトのパス名は \$TUXDIR/udataobj/webgui/webgui.html です。

INIFILE=*init\_file*

アプレットが使用する初期化ファイルの絶対パスを指定します。通常はこの初期化ファイルが使用されますが、ゲートウェイ・プロセスによって使用されるもの以外の初期化ファイルをアプレット・ユーザが使用することも技術的には可能です。ただし、別の初期化ファイルを使用することはお勧めできません。2 つの初期化ファイルを使用する場合は、互いのファイル間で矛盾が生じないようにする必要があります。たとえば、NADDR と CODEBASE パラメータ、各種ディレクトリ・パラメータなどは同じ値に設定する必要があり、ENCRYPTBITS パラメータの値は 2 つのファイル間で一致している必要があります。このように、アプリケーションで 2 つのファイルを使用すると、1 つだけしか使用しない場合よりエラーが発生しやすくなります。

FLDTBLDIR32=*field\_table\_dir* と FIELDTBLS32=*field\_tables*

Administration Console で使用するフィールド・テーブル・ディレクトリおよび値をそれぞれ指定します。これらのパラメータは、BEA Tuxedo システムのインストール・プログラムによって適切な値に設定されます。通常は、以後これらの値は変更しません。

**終了方法** `wlisten` プロセスを正常に終了する唯一の方法は、プロセスに SIGTERM シグナルを送ることです。

**推奨される  
使用法** ■ Administration Console リスナを確実に自動起動するには、UNIX システムの初期化スクリプトに次の形式でコマンド行を追加することをお勧めします。

```
$TUXDIR/bin/wlisten -i initialization_file
```

- `wlisten` プロセスを手動で開始するには、システム・プロンプトの後に上記のコマンド行を入力します。
- インストール中に管理パスワード・ファイルが作成されます。管理パスワードの検索を確実にを行うには、必要に応じて、BEA Tuxedo システムはこのファイルを次のディレクトリから検索します。検索順序は次のとおりです。

```
APPDIR/.adm/tlisten.pw TUXDIR/udataobj/tlisten.pw
```

管理パスワードファイルが確実に見つかるように、必ず `APPDIR` または `TUXDIR` の環境変数を設定してください。

ネットワー  
ク・アドレ  
ス

たとえば、`wlisten` を実行しているローカル・マシンで TCP/IP アドレス指定機能を使用している場合を考えます。マシン名は `backus.company.com` で、アドレスは `155.2.193.18` です。

また、`wlisten` はポート番号 `2334` で要求を受け取るとします。

注記 一部のポート番号は、お使いのシステムで使用されるトランスポート・プロトコル (TCP/IP など) のために予約されている場合があります。予約されているポート番号を確認するには、トランスポート・プロトコルのマニュアルを調べてください。

また、このポート番号 `2334` が `bankapp-nlsaddr` という名前のネットワーク・サービス・データベースに追加されているとします。`-l` オプションで指定するアドレスは、次に示す方法で表現できます。

- `//155.2.193.18:bankapp-nlsaddr`
- `//155.2.193.18:2334`
- `//backus.company.com:bankapp-nlsaddr`
- `//backus.company.com:2334`
- `0x0002091E9B02C112`

最後に示す行は、アドレスを 16 進数形式で表したものです。0002 は TCP/IP アドレスの先頭部分、091E はポート番号 `2334` を 16 進数に変換したもの、および 9B02CU2 は IP アドレス `155.2.193.18` を要素ごとに 16 進数に変換したものです。最後の変換部分については、155 は 9B、2 は 02 のように変換されています。

## セクション 1 - コマンド

---

STARLAN ネットワークの場合は、通常、推奨アドレス `uname.wlisten` が一意の名前になります。

関連項目 [tuxadm\(1\)](#)、[tuxwsvr\(1\)](#)