



BEATuxedo®

Windows NT での BEA Tuxedo システム の使用

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 Portal, BEA WebLogic Server and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Windows での BEA Tuxedo の使用

Part Number	Date	Software Version
N/A	January 2003	BEA Tuxedo リリース 8.1

目次

このマニュアルについて

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

1. BEA Tuxedo ATMI の Windows 2000 での使用

Windows の留意事項.....	1-2
コンフィギュレーションに関する注意事項.....	1-2
マシンのタイプとユーザ ID 番号を指定する.....	1-3
ネットワーク・ドライブを使用する.....	1-4
メモリ・バッファの割り当ておよび解放.....	1-4
TxRPC を更新する.....	1-5
bankapp ドライバを使用する.....	1-5
BEA Tuxedo ATMI のアプリケーションを自動的に起動する.....	1-6
関連項目.....	1-6

2. BEA Tuxedo ATMI for Windows 2000 のコンフィギュレーション

BEA Administration プログラムの概要.....	2-2
BEA Administration プログラムの起動.....	2-2
ネットワーク上のほかのマシンにアクセスする.....	2-5
環境変数を設定および修正する.....	2-6
BEA Tuxedo メッセージを Windows 2000 のイベント・ログに転送する...2-7	
ULOG を設定する.....	2-8
Windows 2000 のイベント・ログのエントリを表示する.....	2-9
ULOG エントリを表示する.....	2-11
tlisten プロセスを自動的に起動する.....	2-11
システムの性能が最大限になるように IPC 資源を設定する.....	2-13

Windows 2000 レジストリの内容を確認する	2-18
開発者キー	2-19
環境キー	2-19
セキュリティ・キー	2-19

3. Visual C++ IDE による BEA Tuxedo ATMI アプリケーションの開発

はじめに	3-2
開発ツールを使用する	3-2
buildserver コマンドおよび buildclient コマンドを使用する	3-3
MSDEV の [Tools] メニューに BuildTuxedo を追加する	3-5
BEA Tuxedo の ATMI プロジェクト・ファイルを作成する	3-7
環境を設定する	3-8
ビルド・タイプ、ヘッダ・ファイル、およびファイル名を指定する ..3-8	
BuildTuxedo でヘッダ・ファイルを使用する	3-10
関数名とサービス名を指定する	3-11
リソース・マネージャを指定する	3-13
BEA Tuxedo ATMI のサーバ・アプリケーションをデバッグする	3-15
Visual C++ IDE GUI の代わりにコマンド行オプションを使用して ATMI アプリケーションを開発する	3-16
Tuxdev アプリケーションを使用する	3-17
BEA Tuxedo ATMI のエディタを使用する	3-18
FML テーブル・エディタを使用する	3-18
VIEW テーブル・エディタを使用する	3-21
複数のドキュメントを同時に処理する	3-25
エディタを使用してエントリの妥当性を検証する	3-25
関連項目	3-26

このマニュアルについて

このマニュアルでは、Windows 98、Windows XP、または Windows 2000 システムで BEA Tuxedo ATMI を使用方法について説明します。Windows 98 または XP システムでは BEA Tuxedo クライアント・コンポーネントがサポートされていますが、BEA Tuxedo サーバ・コンポーネントはサポートされていません。Windows 2000 システムでは、BEA Administration プログラムと BEA Tuxedo Administration Console を含む、BEA Tuxedo のクライアント・コンポーネントとサーバ・コンポーネントがサポートされています。

このマニュアルでは、以下の内容について説明します。

- [第 1 章「BEA Tuxedo ATMI の Windows 2000 での使用」](#)では、UNIX サーバ・システム上の BEA Tuxedo ATMI と Windows 2000 サーバ・システム上の BEA Tuxedo ATMI の基本的な使い方の違いを説明します。
- [第 2 章「BEA Tuxedo ATMI for Windows 2000 のコンフィギュレーション」](#)では、BEA Tuxedo ATMI を Windows 2000 サーバ・システムで設定する方法を説明します。
- [第 3 章「Visual C++ IDE による BEA Tuxedo ATMI アプリケーションの開発」](#)では、Windows 98、Windows XP、または Windows 2000 システム上で Microsoft Visual C++ 統合開発環境 (msdev) を使用して ATMI アプリケーション・クライアントおよびサーバを開発する方法について説明します。

対象読者

このマニュアルは、Windows 98、XP、または 2000 プラットフォームで BEA Tuxedo ATMI クライアント・アプリケーションを開発するアプリケーション開発者および Windows 2000 プラットフォームで BEA Tuxedo ATMI アプリケーション・サーバを開発するアプリケーション開発者を対象とします。

e-docs Web サイト

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

マニュアルの印刷方法

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

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

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

関連情報

次の BEA Tuxedo のマニュアルには、Windows システムで BEA Tuxedo ATMI を使用することに関連する情報が含まれています。

- 『BEA Tuxedo システム入門』
- 『BEA Tuxedo Workstation コンポーネント』

BEA Tuxedo ATMI およびトランザクション処理の詳細については、『Bibliography』を参照してください。

サポート情報

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

電子メール メッセージには、BEA Tuxedo 8.1 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。
等幅テキスト	コード・サンプル、コマンドとオプション、データ構造とメンバ、データ型、ディレクトリ、およびファイル名と拡張子を示します。また、キーボードから入力する文字も示します。 例： <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>

規則	項目
等幅太字	コード内の重要な単語を示します。 例： <code>void commit ()</code>
等幅イタリック体	コード内の変数を示します。 例： <code>String expr</code>
大文字	デバイス名、環境変数、および論理演算子を示します。 例： LPT1 SIGNON OR
{ }	構文の行で選択肢を示します。かっこは入力しません。
[]	構文の行で省略可能な項目を示します。かっこは入力しません。 例： <code>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</code>
	構文の行で、相互に排他的な選択肢を分離します。記号は入力しません。
...	コマンド行で次のいずれかを意味します。 <ul style="list-style-type: none"> ■ コマンド行で同じ引数を繰り返し指定できること ■ 省略可能な引数が文で省略されていること ■ 追加のパラメータ、値、その他の情報を入力できること 省略符号は入力しません。 例： <code>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</code>
.	コード例または構文の行で、項目が省略されていることを示します。省略符号は入力しません。

1 BEA Tuxedo ATMI の Windows 2000 で の使用

以下の節では、UNIX サーバ・システム上の BEA Tuxedo ATMI と Windows 2000 サーバ・システム上の BEA Tuxedo ATMI の基本的な使い方の違いを説明します。

- [Windows の留意事項](#)
- [コンフィギュレーションに関する注意事項](#)
- [マシンのタイプとユーザ ID 番号を指定する](#)
- [ネットワーク・ドライブを使用する](#)
- [TxRPC を更新する](#)
- [bankapp ドライバを使用する](#)
- [BEA Tuxedo ATMI のアプリケーションを自動的に起動する](#)

Windows の留意事項

BEA Tuxedo ATMI を Windows システムで使用する場合、以下の点に注意してください。

- パス名を指定する場合は、ファイル名やディレクトリ名の区切り文字として、スラッシュ (/) ではなく円記号 (\) を使用します。完全修飾されたパス名には、ドライブ文字 (c: など) を使用します。
- 実行可能ファイルに拡張子 .exe を指定する必要はありません。この拡張子は、常に BEA Tuxedo ATMI for Windows 用であると見なされます。
- ファイル名は、Windows の命名規則に従って付けられます。たとえば、拡張子 .dll が付いたファイルは、ダイナミック・リンク・ライブラリを示します。また、.lib は、スタティック・リンク・ライブラリまたはインポートされたライブラリを示し、.cmd は、コマンド・スクリプトを示します。
- BEA Tuxedo システムの実行可能ファイル、コマンド・スクリプト、およびダイナミック・リンク・ライブラリは、すべて %TUXDIR%\bin に格納されています。スタティック・リンク・ライブラリは、%TUXDIR%\lib に格納されています。

コンフィギュレーションに関する注意事項

BEA Tuxedo ATMI ソフトウェアを Windows 2000 サーバ・システムにセットアップする場合は、以下の点に注意してください。

- サーバ名の大文字と小文字は区別されます。
- BEA Tuxedo ATMI アプリケーション・ファイルで指定する Windows 2000 のマシン名は、常に大文字で指定する必要があります。

- BEA Tuxedo のコンフィギュレーション・ファイル (UBBCONFIG) 内の OPENINFO 文字列は、次の形式で指定する必要があります。

```
OPENINFO="resource managers:resource(s)"
```

次に例を示します。

```
OPENINFO="TUXEDO\SQL:APPDIR1\bankd13;bankdb;readwrite"
```

注記 上記の例では、最初の区切り文字としてコロン (:) を使用し、以降の区切り文字としてセミコロン (;) を使用しています。

マシンのタイプとユーザ ID 番号を指定する

コンフィギュレーション・ファイルの MACHINES セクションで、次の 3 つのエントリを指定します。

- TYPE="WinNT"
- UID=0
- GID=0

注記 UNIX システムの場合は、異なる値を設定します。

UNIX システムと Windows 2000 システムが混在する環境の場合は、コンフィギュレーション内のすべての Windows 2000 ノードに対して、MACHINES セクションで上記の 3 つのエントリを指定してください。

ネットワーク・ドライブを使用する

信頼性を確保するため、ネットワーク・ドライブは使用しないことをお勧めします。ただし、TUXCONFIG ファイルをネットワーク・ドライブに持つ Windows 2000 マシン上で BEA Tuxedo ATMI アプリケーションを起動する場合は、パーミッションに関する次の規則に従う必要があります。

- 管理者 (Administrator) としてネットワーク・ドライブに接続します。
- 管理者は、ローカル・システムとリモート・システムで同じパスワードを使用します。
- tuxipc サービスのスタートアップ・オプションで、ENTRY オプションの Log On As This Account を管理者 (Administrator) に設定します。パスワードは、管理者用のパスワードと同じでなければなりません。これで、tuxipc サービスには、管理者としての完全なアクセス権が付与されます。

メモリ・バッファの割り当ておよび解放

Windows 2000 システムでメモリ・バッファの割り当てや解放を行う場合は、必ず、メモリ・バッファを割り当てたヒープを解放してください。これ以外のヒープを解放しようとする、セグメント・エラーが発生します。

たとえば、Falloc() を使用して割り当てたメモリ・バッファを解放するには、Ffree() を使用します。malloc() を使用して割り当てたメモリ・バッファを Ffree() で解放しようとする、セグメント・エラーが発生します。この場合は、free() を使用して解放します。

Falloc、Falloc32(3fml) および Ffree、Ffree32(3fml) の詳細については、『BEA Tuxedo FML リファレンス』を参照してください。malloc() および free() の詳細については、オペレーティング・システムに付属のマニュアルを参照してください。

TxRPC を更新する

BEA Tuxedo ATMI が OSF/DCE と相互運用する場合は、アプリケーションに対する適切な `PATH` 変数を指定して環境ファイルを更新します。これで、BEA Tuxedo ATMI プログラムでは、必要に応じて OSF/DCE の DLL を検索できます。`bldc_dce(1)` コマンドおよび `blds_dce(1)` コマンドを Windows 2000 システムで使用方法については、『BEA Tuxedo コマンド・リファレンス』にある、これらのコマンドのリファレンス・ページを参照してください。

- `bldc_dce` コマンドは、BEA Tuxedo ATMI クライアント (OSF/DCE RPC により呼び出される) を生成します。
- `blds_dce` コマンドは、BEA Tuxedo ATMI サーバ (OSF/DCE を呼び出す) を生成します。

bankapp ドライバを使用する

`bankapp` プログラムは、BEA Tuxedo ATMI for Windows 2000 に同梱される小規模なサンプル・アプリケーションです。このサンプル・アプリケーションでは、BEA Tuxedo ATMI の操作を試したり、BEA Tuxedo ATMI のアプリケーション・コードの例を参照することができるほか、`bankapp` ドライバを使用して生成した `%APPDIR%\UBB` ファイルを、新しいアプリケーションのテンプレートとして適用できます。`driver.exe` は、`%TUXDIR%\APPS\nt\nt\driver` に格納されています。

BEA Tuxedo ATMI のアプリケーションを自動的に起動する

BEA Tuxedo ATMI が Windows 2000 サーバ・システムにインストールされている場合は、Microsoft Resource Kit for Windows 2000 に格納されている `srvany.exe` ユーティリティ・プログラムを使用して、システムの起動時に BEA Tuxedo ATMI アプリケーションが自動的に起動されるようマシンを設定しておく便利です。この手順については、`srvany.wri` と `rkttools.hlp` を参照してください。

BEA Tuxedo ATMI プログラムの起動時にプログラムを正しく自動起動させるには、BEA Tuxedo システムの環境変数である `%TUXDIR%` および `%NLSPATH%` を設定する必要があります。これらの環境変数は、通常の Windows 2000 の方法、または BEA Administration プログラムを使用して設定します (2-6 ページの「環境変数を設定および修正する」を参照)。

関連項目

Windows 2000 システムに BEA Tuxedo ATMI をインストールする方法については、『BEA Tuxedo システムのインストール』を参照してください。BEA Tuxedo システムの詳細については、以下のマニュアルを参照してください。

- 『BEA Tuxedo アプリケーションの設定』
- 『BEA Tuxedo アプリケーション実行時の管理』
- 『BEA Tuxedo Workstation コンポーネント』
- 『BEA Tuxedo コマンド・リファレンス』
- 『BEA Tuxedo C リファレンス』
- 『BEA Tuxedo COBOL リファレンス』

- 『BEA Tuxedo FML リファレンス』
- 『BEA Tuxedo のファイル形式とデータ記述方法』

2 BEA Tuxedo ATMI for Windows 2000 のコンフィギュレー ション

以下の節では、Windows 2000 サーバ・システムで BEA Tuxedo ATMI を設定する方法について説明します。

- [BEA Administration プログラムの概要](#)
- [BEA Administration プログラムの起動](#)
- [ネットワーク上のほかのマシンにアクセスする](#)
- [環境変数を設定および修正する](#)
- [BEA Tuxedo メッセージを Windows 2000 のイベント・ログに転送する](#)
- [tlisten プロセスを自動的に起動する](#)
- [システムの性能が最大限になるように IPC 資源を設定する](#)
- [Windows 2000 レジストリの内容を確認する](#)

BEA Administration プログラムの概要

Windows 用の BEA Tuxedo 7.1 以降では、BEA Tuxedo Administration Console のほかに、Windows 2000 システム上で BEA Tuxedo システムを設定するための BEA Administration プログラムと 2 つの Windows サービス (Tlisten と BEA procMGR) が提供されます。インストールに BEA Tuxedo サーバ・コンポーネントが含まれている場合のみ、これらの管理ツールは Windows 2000 システムにインストールされます。

注記 BEA Tuxedo Administration Console には、充実したオンライン・ヘルプが用意されています。BEA Administration Console には、このコンソールで使用できる管理タスクの実行方法やコンフィギュレーション・ツール・フォルダに関する情報などを含む、充実したオンライン・ヘルプが用意されています。BEA Tuxedo Administration Console の使い方については、[BEA Tuxedo Administration Console オンライン・ヘルプ](#)を参照してください。

BEA Administration プログラムの起動

BEA Tuxedo リリース 7.1 以降が Windows 2000 システムにインストールされている場合、次の手順で BEA Administration プログラムにアクセスします。

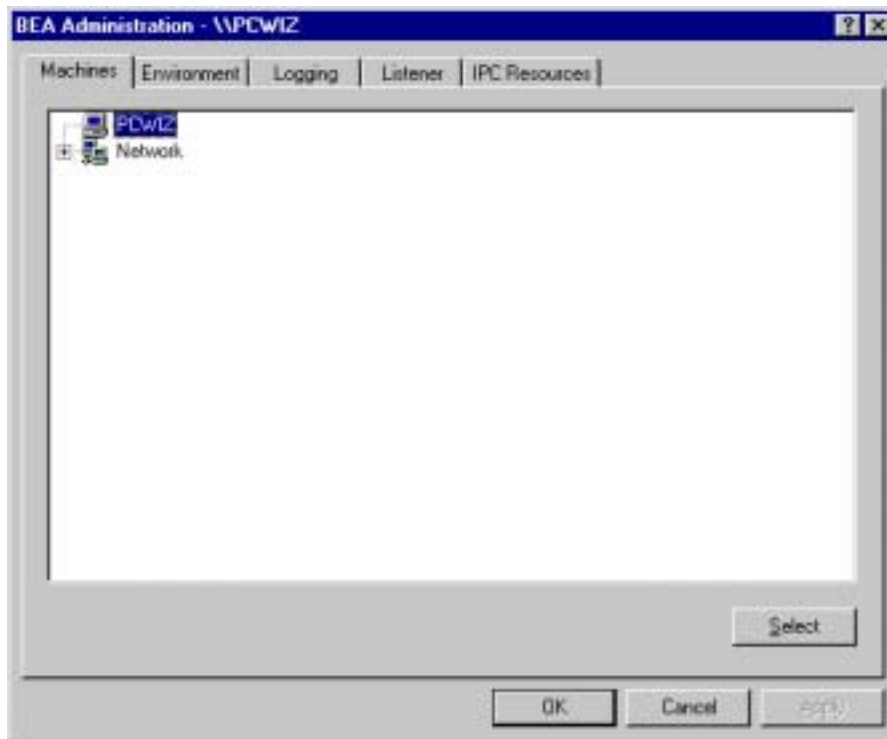
1. [スタート]メニューの [設定] をポイントし、[コントロール パネル] をクリックしてコントロール・パネルを開きます。→→

図 2-1 Microsoft Windows のコントロール・パネル



2. コントロール・パネルで、[BEA Administration] アイコンをダブルクリックして [BEA Administration] ウィンドウを開きます。

図 2-2[Machines] ページの表示された [BEA Administration] ウィンドウ



[BEA Administration] ウィンドウでは、次のことができます。

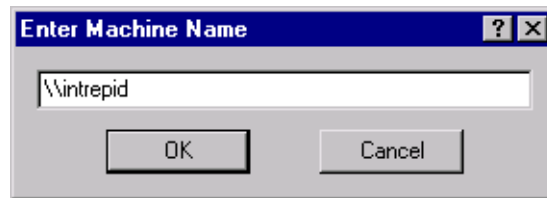
- BEA Tuxedo システムがインストールされているネットワーク上のほかのマシンにアクセスする。
- BEA Tuxedo システムの環境変数を設定および修正する。
- BEA Tuxedo のシステム・メッセージを Windows 2000 イベント・ログに転送する。
- 自動的に起動するように BEA Tuxedo tlisten(1) プロセスを設定する。
- プロセス間通信 (IPC) 資源を調整して、BEA Tuxedo システムの性能を最大限に引き出す。

ネットワーク上のほかのマシンにアクセスする

[BEA Administration] ウィンドウの [Machines] ページを使用すると、BEA Tuxedo のシステム管理者は、Microsoft Windows 2000 が動作する Microsoft Windows ネットワーク上のどのマシンにでもログイン権限があればアクセスできます。アクセスした後は、リモートでの環境変数の設定、BEA Tuxedo イベント・ログを保存する場所の指定、`tlisten` サービスの追加、削除、および開始、そして IPC 資源の調整ができます。

リモート・マシンにアクセスするには、ネットワーク・ツリーでそのマシンのアイコンを見つけてクリックします。リモート・マシンの名前はわかるが、所属するワーク・グループがわからない場合は、次の手順に従ってリモート・マシンを選択します。

1. [Machines] ページの右下隅で [Select] をクリックして、[Enter Machine Name] ダイアログ・ボックスを表示します。



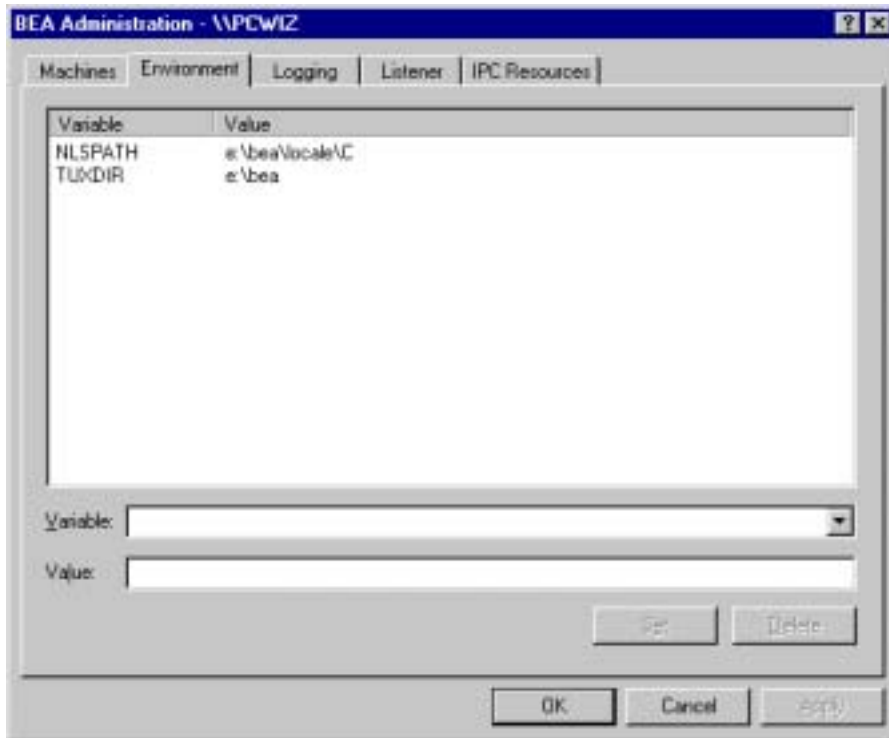
2. [Enter Machine Name] ダイアログ・ボックスで、リモート・マシンの名前 (\\intrepid など) を入力して [OK] をクリックします。

これ以降、[BEA Administration] ウィンドウのほかのページ ([Environment]、[Logging]、[Listener]、および [IPC Resources]) で行われるアクションはすべて選択されたマシン (この例では `intrepid`) で行われます。

環境変数を設定および修正する

[Environment] ページでは、Windows 2000 システム上の BEA Tuxedo 環境変数を表示、設定、および修正できます。

図 2-3 [Environment] ページの表示された [BEA Administration] ウィンドウ



[Environment] ページを使用して環境変数を追加、修正、または削除するには、次の手順に従います。

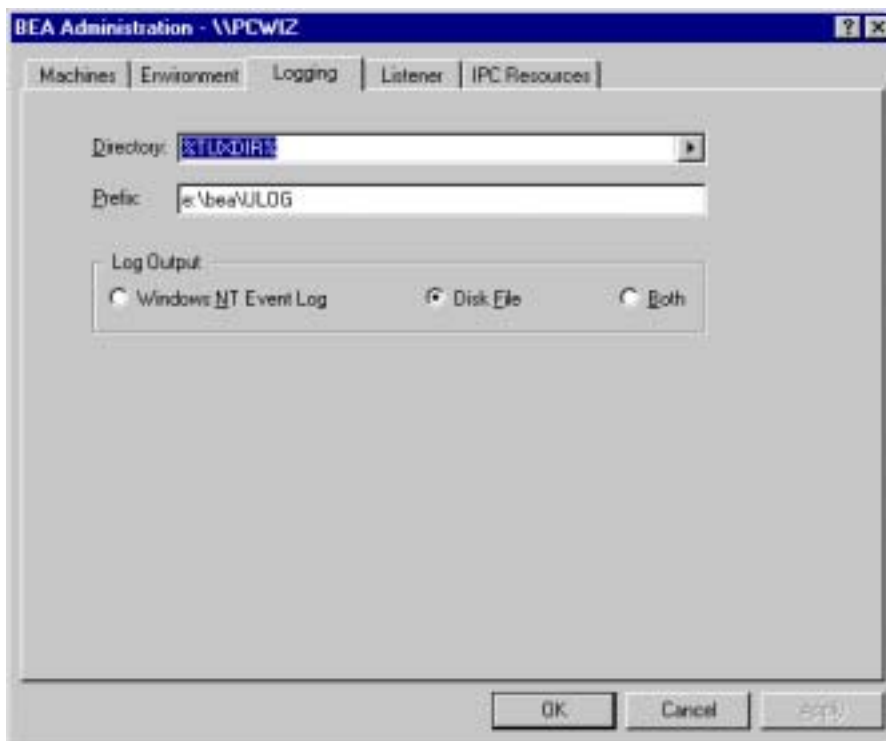
1. 変数を追加するには、[Variable] フィールドに名前、[Value] フィールドに値を入力して、[Set] をクリックします。

2. 変数を修正するには、変数を選択してから [Value] フィールドに新しい値を入力して、[Set] をクリックします。
3. 変数を削除するには、削除する変数を選択して [Delete] をクリックします。
4. [OK] または [Apply] をクリックして、変更を Windows 2000 レジストリに書き込みます。

BEA Tuxedo メッセージを Windows 2000 のイベント・ログに転送する

[Logging] ページでは、BEA Tuxedo のシステム・メッセージを Windows 2000 システム上のイベント・ログまたは従来のユーザ・ログ (ULOG) ディスク・ファイル、あるいはその両方に転送できます。

図 2-4[Logging] ページの表示された [BEA Administration] ウィンドウ



イベント・ログ、従来のユーザ・ログ (ULOG) ディスク・ファイル、または両方を選択できます。イベント・ログでセットアップは必要ありません。

ULOG を設定する

従来のユーザ・ログ (ULOG) メッセージをセットアップする場合は、次の手順で保存ディレクトリを選択します。

1. [Logging] ページで、[Directory] フィールドの右側の矢印をクリックしてディレクトリのドロップ・ダウン・リストを表示します。
2. ULOG メッセージを保存するディレクトリの名前を選択します。

3. [Prefix] フィールドに、ログ・ファイル名の接頭辞を入力します。デフォルトの接頭辞は ULOG、デフォルトのファイル名は ULOG.mmdyy です。mmdyy は、ログ・ファイルの作成された月、日、年です。
4. 選択した内容を Windows 2000 レジストリに保存するには、[OK] または [Apply] をクリックします。

Windows 2000 のイベント・ログのエントリを表示する

Windows 2000 のイベント・ログ・エントリを表示するには、次の手順に従います。

1. Windows 2000 のデスクトップで、[スタート]メニューの[設定]をポイントして[コントロールパネル]をクリックします。次に、[管理ツール]の[イベントビューア]をクリックして、アプリケーションが起動してから発生したすべてのイベントのリストを表示します。→→→

図 2-5[イベント ビューア] ウィンドウ



The screenshot shows the 'Event Viewer - System Log on WPCWIZ' window. The window title bar includes 'Log View Options Help'. The main area contains a table of event entries with the following columns: Date, Time, Source, Category, Event, User, and Computer. The table lists several events from 3/20/01, including EventLog entries and a stapi event.

Date	Time	Source	Category	Event	User	Computer
3/20/01	9:48:57 AM	EventLog	None	6005	N/A	PCWZ
3/20/01	9:48:57 AM	EventLog	None	6009	N/A	PCWZ
3/20/01	9:47:19 AM	EventLog	None	6006	N/A	PCWZ
3/20/01	9:39:32 AM	stapi	None	9	N/A	PCWZ
3/20/01	9:22:19 AM	EventLog	None	6005	N/A	PCWZ
3/20/01	9:22:19 AM	EventLog	None	6009	N/A	PCWZ

2. [イベント ビューア] ウィンドウの上端にあるメニュー・バーで[ログ]をクリックして、ドロップ・ダウン・メニューを表示します。
3. ドロップ・ダウン・メニューから[アプリケーション ログ]を選択します。これにより、[イベント ビューア] ウィンドウにアプリケーション固有のイベントだけが一覧表示されます。

4. 見たいイベントのエントリをダブルクリックして詳細を表示します。
Windows 2000 の [イベントの詳細] ウィンドウが開き、指定されたイベントに関する情報が表示されます。

図 2-6 [イベントの詳細] ウィンドウ



このウィンドウには、次の情報が表示されます。

- [日付] - イベントが発生した日付。
- [時刻] - イベントが発生した時刻。
- [ユーザー] - イベントが発生したときにソフトウェアを使用していたユーザ。

- [コンピュータ] - イベントが発生したマシン。
- [イベント ID] - 『システム・メッセージ』にメッセージを表示する際の番号。
- [ソース] - イベントの説明と推奨される操作を示す 『システム・メッセージ』のセクション。
- [種類] - メッセージの目的。特に、情報提供、警告、エラーの通知のいずれかであることを示します。
- [説明] - イベントの簡単な説明。

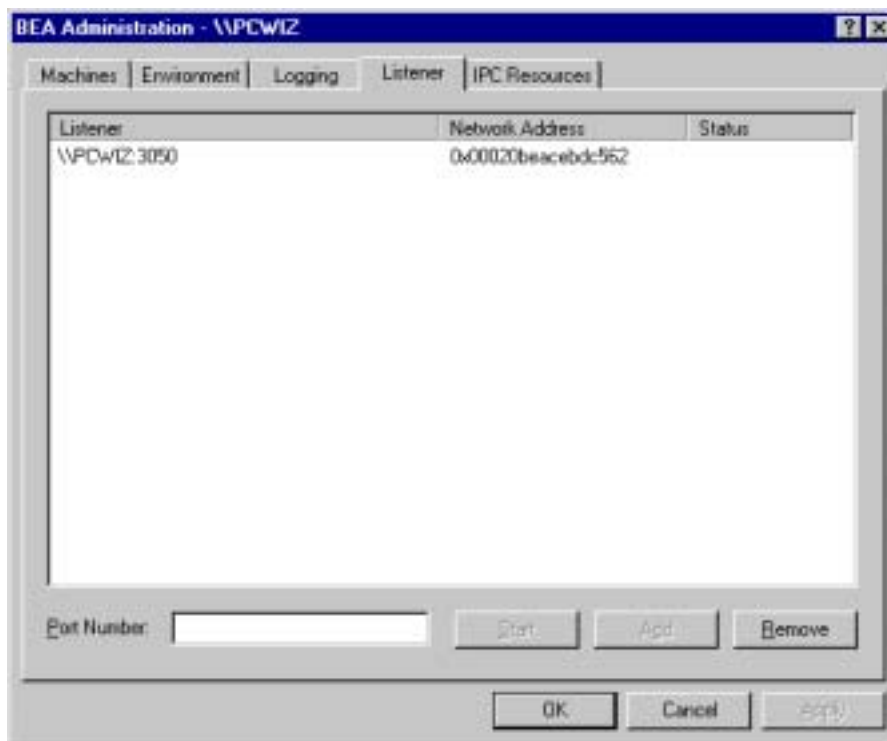
ULOG エントリを表示する

ULOG はテキスト・ファイルなので、テキスト・エディタを使用して ULOG を開き、内容を表示することができます。

tlisten プロセスを自動的に起動する

[Listener] ページでは、Windows 2000 システム上の Tlisten サービスを表示、作成、または修正できます。

図 2-7[Listener] ページの表示された [BEA Administration] ウィンドウ



Tlisten サービスは、BEA Tuxedo をインストールする過程でインストーラ・プログラムによって Windows 2000 システムにインストールされました。システムが起動するたびに、Tlisten サービスによってマシンのポート 3050 で tlisten プロセスが開始されます。tlisten プロセスに関連付けられたパスワードは、インストール時に入力したパスワードです。

tlisten プロセスがネットワーク接続された BEA Tuxedo アプリケーションの各マシンで最初に開始されないと、BEA Tuxedo システムとアプリケーション・サーバは起動できません。tlisten プロセスを使用すると、複数のマシンにまたがる管理アクションを実行できます。tlisten プロセスの詳細については、『BEA Tuxedo コマンド・リファレンス』の [tlisten\(1\)](#) を参照してください。

[Listener] ページを使用して Tlisten サービスを追加、削除、または開始するには、次の手順に従います。

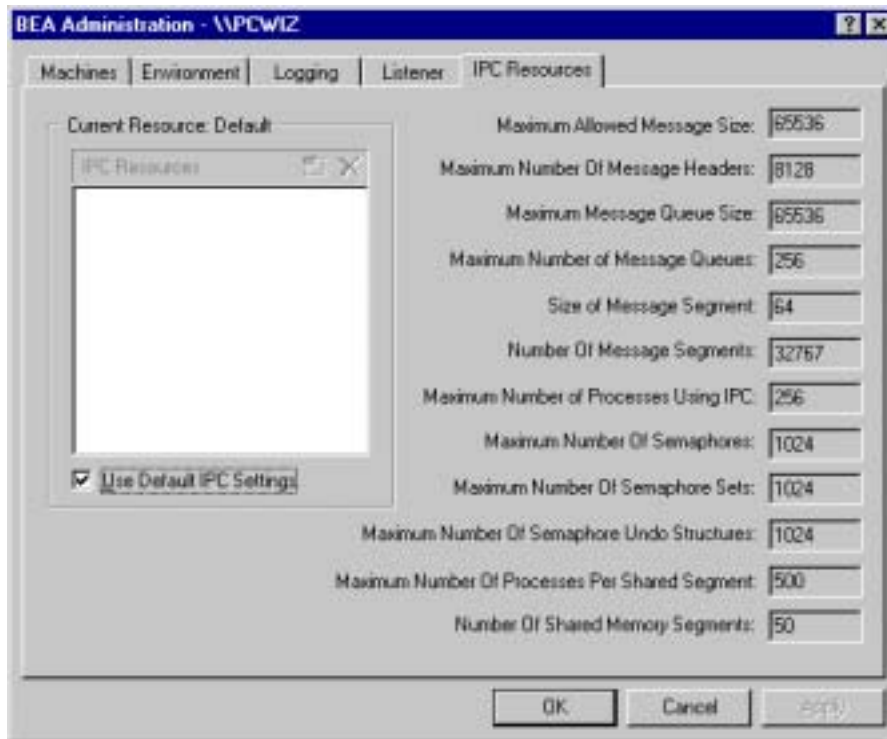
1. Tlisten サービスを追加するには、[Port Number] フィールドにポート番号を入力して [Add] をクリックします。通常は、マシンで動作する BEA Tuxedo アプリケーションごとに 1 つの `tlisten` プロセスが必要です。
2. Tlisten サービスを削除するには、削除する [Tlisten] サービスを選択して [Remove] をクリックします。
3. 開始されている Tlisten サービスを停止するには、停止する Tlisten サービスを選択して [Stop] をクリックします。
4. 停止されている Tlisten サービスを開始するには、開始する Tlisten サービスを選択して [Start] をクリックします。
5. [OK] または [Apply] をクリックして、変更を Windows 2000 レジストリに書き込みます。

[Listener] ページ以外にも、[スタート] メニューの [設定] をポイントして [コントロール パネル] をクリックし、[管理ツール] の [Services] をクリックして [Services] ウィンドウを開けば、そのウィンドウで Tlisten サービスを表示および管理できます。→ → →

システムの性能が最大限になるように IPC 資源を設定する

[IPC Resources] ページでは、BEA Tuxedo の性能を最大限に引き出すように Windows 2000 システム上のプロセス間通信 (IPC) 資源を設定できます。

図 2-8[IPC Resources] ページの表示された [BEA Administration] ウィンドウ



BEA ProcMGR サービスは、BEA Tuxedo をインストールする過程でインストーラ・プログラムによって Windows 2000 システムにインストールされました。システムを起動するたびに、BEA ProcMGR サービスによって、マシン上の IPC 資源が [IPC Resources] ページで設定された値に設定されます。前の例で表示されている値は、インストーラ・プログラムによって設定されたデフォルトの IPC 値です。

ほとんどのマシンで BEA ProcMGR はインストール時の設定で実行されますが、[IPC Resources] ページを使用すると IPC 資源を調整して性能を最大限に引き出すことができます。BEA Tuxedo アプリケーションの IPC 資源の最小値を確認するには、次の表と『BEA Tuxedo システムのインストール』の

「[IPC 要件のチェック](#)」を参照してください。次の表では、Windows 2000 システム上の IPC 資源の名前と UNIX システムでの従来の名前が対応付けられています。

表 2-1 Windows システムと UNIX システムの IPC 資源名の対応

Windows 2000 での名前	UNIX での従来の名前
Maximum Allowed Message Size	MSGMAX
Maximum Number of Message Headers	対応する名前なし
Maximum Message Queue Size	MSGMNB
Maximum Number of Message Queues	MSGMNI
Size of Message Segment	MSGSSZ
Number of Message Segments	MSGSEG
Maximum Number of Processes Using IPC	NPROC
Maximum Number of Semaphores	SEMMNS
Maximum Number of Semaphore Sets	SEMMNI
Maximum Number of Semaphore Undo Structures	SEMMNU
Maximum Number of Processes Per Shared Segment	対応する名前なし
Number of Shared Memory Segments	SHMMNI

[IPC Resources] ページを使用して IPC 資源の値を修正するには、次の手順に従います。

1. [Current Resource: Default] ボックスで、[Use Default IPC Settings] チェック・ボックスをクリックしてオフにします。[Current Resource: Default] ボックスに挿入ボックスが表示されます。
2. 挿入ボックスをクリックして Windows 2000 マシンの名前を入力し、[Enter] を押します。
3. 変更する IPC 資源の隣にあるフィールドをクリックし、必要な値を入力します。
4. [OK] または [Apply] をクリックして、変更を Windows 2000 レジストリに書き込みます。

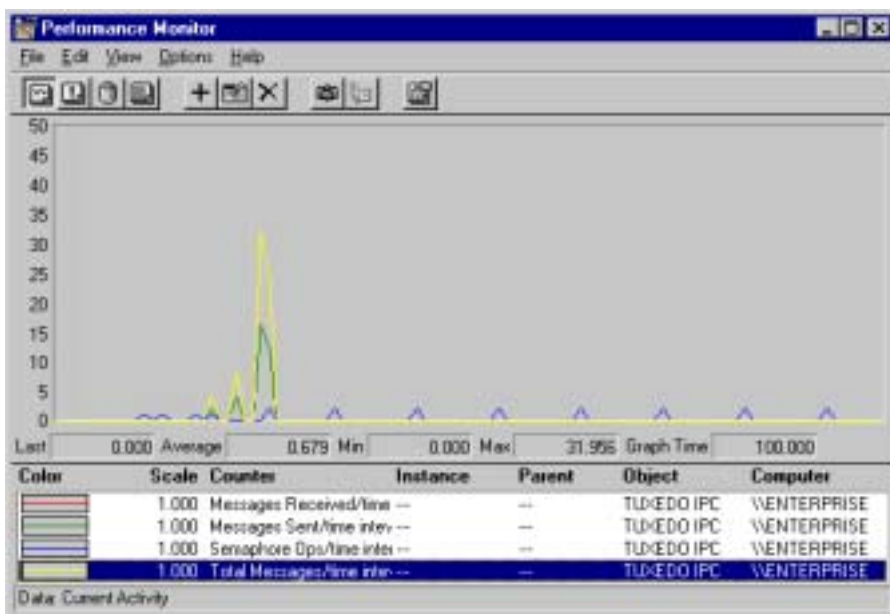
5. BEA ProcMGR サービスを再起動して、変更を有効にします。そのためには、[スタート]メニューの[設定]をポイントして[コントロールパネル]をクリックし、[管理ツール]の[Services]をクリックして[Services]ウィンドウを開き、そのウィンドウを通じて BEA ProcMGR を再起動します。→ → →

[IPC Resources] ページの [Maximum Number of Processes Using IPC] パラメータを指定する場合は、次の点に注意してください。

- マルチコンテキスト化された BEA Tuxedo ATMI クライアントは複数回カウントします。プロセスの合計は、同時に未処理状態にできるアプリケーションの関連付け (コンテキスト) の数になります。
- マルチコンテキスト化された BEA Tuxedo ATMI サーバは複数回カウントします。プロセスの合計は、コンテキストの数、つまりターゲット・アプリケーションのコンフィギュレーション (UBBCONFIG) ファイルの MAXDISPATCHTHREADS に 1 (主要なディスパッチャ・スレッド) を加えた数になります。

実行中の BEA Tuxedo アプリケーションの性能は、Windows 2000 のパフォーマンス・モニタで確認できます。パフォーマンス・モニタを起動するには、[スタート]メニューの[設定]をポイントして、[コントロールパネル]をクリックします。次に、[管理ツール]から [パフォーマンス] をクリックします。→ → →

図 2-9 パフォーマンス・モニタ



Windows 2000 レジストリの内容を確認する

Windows 2000 レジストリは、Windows 2000 システムのすべてのハードウェア、ソフトウェア、およびアプリケーションのコンフィギュレーション設定を格納する場所です。BEA Tuxedo をインストールする過程で、IPC 資源の値だけでなく全般的なインストール情報がインストーラ・プログラムによってレジストリに書き込まれます。BEA Tuxedo に関連するレジストリの構造は、次のようになっています。

```
HKEY_LOCAL_MACHINE\Software\BEA Systems\Tuxedo\8.1\...
```

- Developer

- Environment
- Security

この構造を表示するには、[スタート]メニューの[ファイル名を指定して実行]を選択して[ファイル名を指定して実行]ダイアログ・ボックスを表示します。→次に、「regedt32」を入力して[OK]をクリックし、[Repository Editor] ウィンドウを表示します。

開発者キー

開発者キーには、製品情報（リリースのメジャーとマイナーのバージョン番号など）とユーザおよび会社の名前が格納されます。

環境キー

環境キーには、Windows 2000 システムで設定された BEA Tuxedo 環境変数によって示される場所が格納されます。また、IPC 資源の設定などのほかの値も格納されます。

セキュリティ・キー

セキュリティ・キーには、BEA Tuxedo のプロセスとサービスのアクセス権が格納されます。次のパーミッションは必須です。

- `tlisten(1)` を実行するユーザには、読み込みアクセス権が必要です。
- BEA ProcMGR が動作するアカウントには、読み込みアクセス権が必要です。

管理者 (Administrator) は、完全な制御権を持つことをお勧めします。

3 Visual C++ IDE による BEA Tuxedo ATMI アプリケーションの開発

以下の節では、Windows 98、Windows XP、または Windows 2000 システム上で Microsoft Visual C++ 統合開発環境 (msdev) を使用して ATMI アプリケーション・クライアントおよびサーバを開発する方法について説明します。

- はじめに
- 開発ツールを使用する
- buildserver コマンドおよび buildclient コマンドを使用する
- MSDEV の [Tools] メニューに BuildTuxedo を追加する
- BEA Tuxedo の ATMI プロジェクト・ファイルを作成する
- 環境を設定する
- BEA Tuxedo ATMI のサーバ・アプリケーションをデバッグする
- Visual C++ IDE GUI の代わりにコマンド行オプションを使用して ATMI アプリケーションを開発する

- Tuxdev アプリケーションを使用する
- BEA Tuxedo ATMI のエディタを使用する

はじめに

Windows 98 または Windows XP には BEA Tuxedo ATMI クライアント (サーバ・コンポーネントは別) をインストールできるので、それらのプラットフォームでは、Microsoft Visual C++ IDE で ATMI アプリケーション・クライアントのみ開発できます。一方、Windows 2000 には BEA Tuxedo ATMI クライアントとサーバの両コンポーネントをインストールできるので、Windows 2000 プラットフォームでは、Microsoft Visual C++ IDE で ATMI アプリケーション・クライアントとアプリケーション・サーバの両方を開発できます。

開発ツールを使用する

BEA Tuxedo ATMI は、Microsoft Visual C++ IDE (msdev) に統合されますが、統合できない場合は、msdev の機能がエミュレートされます。このような統合により、16 ビットまたは 32 ビットの Windows オペレーティング・システムで動作する BEA Tuxedo ATMI アプリケーションを簡単に開発できます。

BEA Tuxedo ATMI のサービス要求の作成や送信、会話型接続の確立、および応答の受信を実現するためのコーディングは、基本的に UNIX 環境と Windows 2000 環境で同じです。開発支援ツールとして、BuildTuxedo と TUXDEV が用意されています。

- BuildTuxedo は単一のツールであり、msdev 環境に密接に統合されているため、buildserver コマンド、buildclient コマンド、および buildclt コマンドの代わりに使用できます (もちろん、これらのコマンドも使用できます)。BuildTuxedo のシステムは、現在サポートされているすべての Windows 環境で同じように使用できます。

- TUXDEV を使用すると、複数の 16/32 ビットの FML テーブルおよび VIEW ファイルを作成、編集、およびコンパイルできます。TUXDEV には、MDI (Multiple Document Interface) のアーキテクチャが組み込まれているため、ファイル型が異なるビューを表示することもできます。

buildserver コマンドおよび buildclient コマンドを使用する

buildclient() コマンドと buildserver() コマンドは両方とも Windows 2000 で利用できますが、Windows 98 と XP では両方は利用できません。Windows 98 と XP では、buildclient() コマンドのみ利用できます。つまり、Windows 98 システムと Windows XP システムで作成できるのは Tuxedo アプリケーション・クライアントのみです。build コマンドの詳細については、『BEA Tuxedo コマンド・リファレンス』の [buildclient\(1\)](#) と [buildserver\(1\)](#) を参照してください。

buildclient() コマンドと buildserver() コマンドのオプションは、次の表で示されているように統合開発環境とそれ以外の環境では異なる働きをします。

表 3-1 buildserver コマンドおよび buildclient コマンド を使用する

目的 ..	統合開発環境以外の 場合に使用するオプション ..	統合開発環境の場合 ..
冗長モードのアク ティブ化	-v	デフォルトでは、すべてのオプションがタブ上に 表示されます。-v は不要なため、サポートされて いません。

3 Visual C++ IDE による BEA Tuxedo ATMI アプリケーションの開発

表 3-1 buildserver コマンドおよび buildclient コマンド (続き) を使用する

目的 ..	統合開発環境以外の 場合に使用するオプション ..	統合開発環境の場合 ..
出力ファイルの指定	-o (<i>output_filename</i>)	<ol style="list-style-type: none">1. msdev の [Build] メニューから [Settings] を選択します。2. [Project Settings] ダイアログ・ボックスの [Link] タブをクリックします。3. 編集機能を使用して、出力ファイルの名前を指定します。
最初にリンクするファイルの指定	-f	<ol style="list-style-type: none">1. msdev の [Build] メニューから [Settings] を選択します。2. [Project Settings] ダイアログ・ボックスの [Link] タブをクリックします。3. 編集機能を使って最初にリンクするファイルの名前を指定します。
最後にリンクするファイルの指定	-l	<ol style="list-style-type: none">1. msdev メニューから [Project] を選択します。2. [Project Settings] ダイアログ・ボックスの [Link] タブをクリックします。3. 編集機能を使って最後にリンクするファイルの名前を指定します。
リソース・マネージャの指定	-r	<ol style="list-style-type: none">1. BuildTuxedo の GUI にアクセスします。2. BuildTuxedo の [Test] ウィンドウを開き、[Resources] タブを選択します。3. [Tuxedo Resource Manager] フィールドに、リソース・マネージャの名前を入力します。
サーバ上で使用できるサービスの指定	-s	<ol style="list-style-type: none">1. BuildTuxedo の GUI にアクセスします。2. BuildTuxedo の [Test] ウィンドウを開き、[Services] ページを選択します。3. [Service Names] フィールドに各サービスの名前を 1 行ごとに入力します。
COBOL コンパイラの使用	-c	COBOL は使用できません。

3-4 『Windows NT での BEA Tuxedo システムの使用』

統合開発環境でビルド環境を変更するには、次の手順に従います。

1. msdev の [Project] メニューから [Settings] を選択します。
2. [C/C++] または [Link] タブを選択します。

注記 CC および CFLAGS は、不要になりました。

ライブラリを指定し、統合開発環境のパスを指定するには、次の手順に従います。

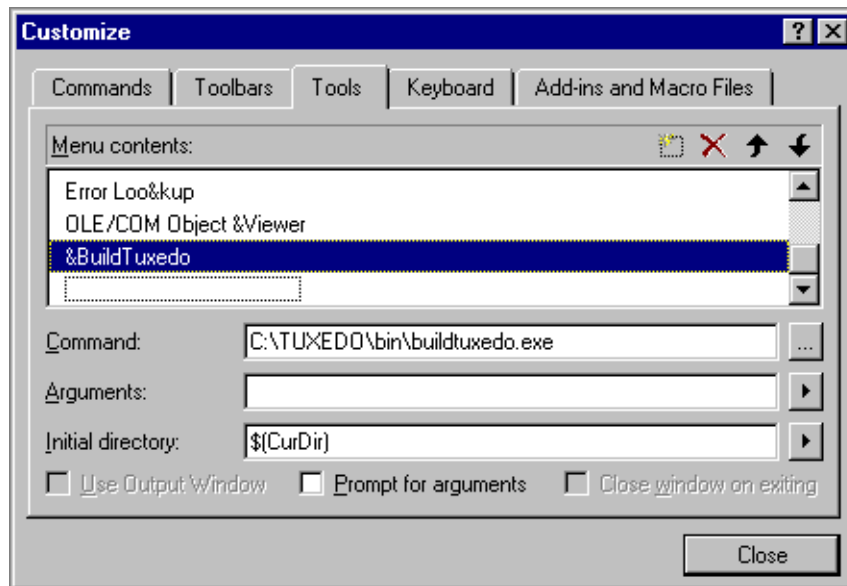
1. MSDEV の [Tools] メニューの [Options] を選択して、[Options] ダイアログ・ボックスを表示します。
2. [Options] ダイアログ・ボックスで、[Directories] タブを選択します。


MSDEV の [Tools] メニューに BuildTuxedo を追加する

MSDEV の [Tools] メニューに BuildTuxedo を追加するには、次の手順に従います。


1. Microsoft Visual C++ IDE (msdev) の GUI 画面で [Tools] を選択します。次に [Customize] を選択し、[Tools] を選択します。[Customize] ウィンドウが表示されます。
2. [Customize] ウィンドウで [Tools] タブをクリックして、[Tools] ページを表示します。

図 3-1[Customize] ウィンドウの [Tools] ページ



- [Tools] ページの [Menu contents] ペインで [New] アイコン  をクリックして [Menu Contents] ボックスを表示し、[Menu Contents] ボックスの最後の空白のボックスに「&BuildTuxedo」を入力して、Enter キーを押します。

注記 文字キーをホット・キーとして使用するには、その文字の前にアンパサンド (&) を追加します。

- [Tools] ページの [Command] フィールドで、次のいずれかの方法でパス名を設定します。
 - BuildTuxedo のフル・パス名 (%TUXDIR%\bin\BuildTuxedo) を入力します。
 - [Command] フィールドの右側にある [...] ボタン  をクリックして、適切なディレクトリで buildtuxedo.exe を選択し、[Open] をクリックします。選択したパス名が [Customize] ウィンドウに表示されます。

5. [Tools] ページの [Initial directory] フィールドで右矢印ボタンをクリックして、ポップ・アップ・メニューを表示します。ポップ・アップ・メニューから [Current Directory] ($\$(CurDir)$) を選択します。

注記 項目を変更する場合は、対象の項目を強調表示してから上書きします。項目を移動する場合は、対象の項目を強調表示してから [Menu contents] ボックスの上部にある上矢印ボタンまたは下矢印ボタンをクリックします。

6. [Close] をクリックして、[Customize] ウィンドウを閉じます。

これで、BuildTuxedo が MSDEV の [Tools] メニューに表示されます。

BEA Tuxedo の ATMI プロジェクト・ファイルを作成する

BuildTuxedo は、BEA Tuxedo ATMI アプリケーションごとに作成される個別のプロジェクト・ファイルを、カレント・ディレクトリ内に保持します。BuildTuxedo が起動すると、カレント・ディレクトリ内にある有効なプロジェクト・ファイルが検索されます。有効なプロジェクト・ファイルが見つかると、ダイアログ・ボックス内の値がこのファイル内の値に設定され、ダイアログ・ボックスが表示されます。タイトル・バーには次のように表示されます。

BuildTuxedo *project_name*

BuildTuxedo プロジェクトは、カレント・ディレクトリ内の `msdev` プロジェクトと密接に関連するため、BuildTuxedo は、次の 2 つのファイルも検索します。

- `msdev` の有効な make ファイル (*filename.mak*)
- `msdev` のプロジェクト・ファイル (*filename.mdp*、*filename.dsw*、または *filename.dsp*)

BuildTuxedo が上記のどちらかのファイルを検索できない場合は、警告メッセージが表示され、起動は失敗します。ディレクトリに複数の BuildTuxedo プロジェクト・ファイル、msdev プロジェクト・ファイル、または make ファイルがある場合は、適切なファイル名があるメニュー項目が [System] メニューに追加されます。

作業中のプロジェクト・ファイルを保存するには、[OK] または [Apply] を選択します。プロジェクト・ファイルまたは BEA Tuxedo システムが保持するその他のファイルに対する変更内容をキャンセルするには、[Cancel] をクリックするか、または Esc キーを押します。

環境を設定する

統合開発環境で BEA Tuxedo ATMI アプリケーションをビルドするには、使用する環境内で次の基本的なパラメータを設定しておく必要があります。

- ビルド・タイプ
- ヘッダ・ファイル
- BuildTuxedo によって作成され、管理される C ファイルまたは C++ ファイルの名前
- 関数名
- サービス名

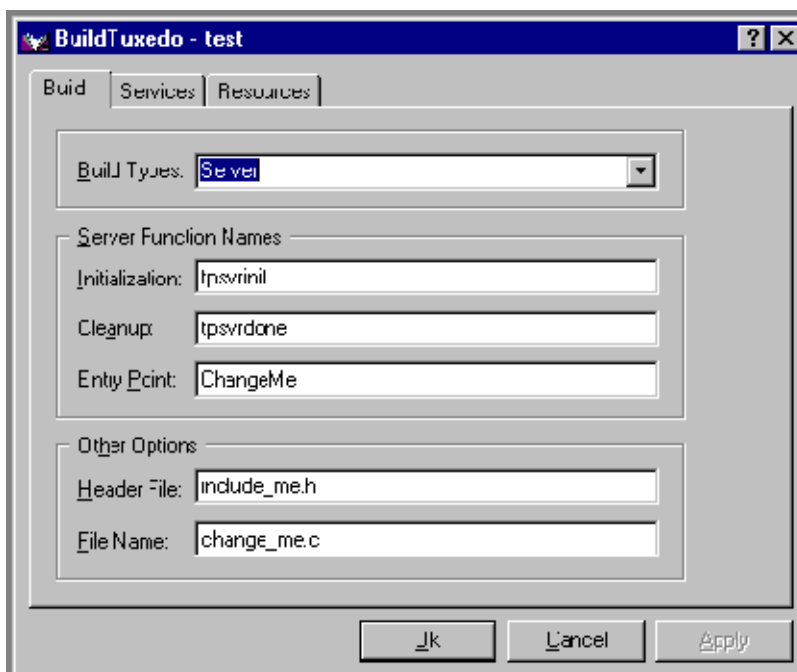
これらの情報を設定するには、msdev の GUI 画面で、[BuildTuxedo project_name] ダイアログ・ボックスを表示します。

ビルド・タイプ、ヘッダ・ファイル、およびファイル名を指定する

ビルド・タイプ、ヘッダ・ファイル、およびファイル名を指定するには、次の手順に従います。

1. [project_name] ダイアログ・ボックスで [Build] タブをクリックして、[Build] ページを表示します。

図 3-2[Build] ページ



2. [Build] ページの [Build Types] フィールドの下矢印をクリックし、次のどちらかを選択します。

- サーバ
- Native client or Workstation client (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)

注記 Windows 98 または XP のユーザが選択できるのは、ワークステーション・クライアントのみです。

ビルド・タイプ..	必要な操作..
サーバ	[Initialization]、[Cleanup]、および [Entry Point] フィールドに情報を入力します。手順 3 に進みます。
Native Client or Workstation Client (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)	手順 3 に進みます。

[Initialization] フィールドおよび [Cleanup] フィールドに指定された有効な関数名は、デフォルトの init/exit 関数をオーバーライドします。[Entry Point] フィールドには、BuildTuxedo が生成する関数の名前を指定できます。関数名を指定しておくと、後でアプリケーションのどこからでもこの関数を呼び出せます。

- [Build] ページの [Header File] フィールドに、「stdafx.h」と入力します。
- [Build] ページの [File Name] フィールドに、BuildTuxedo によって生成および管理される C または C++ ファイルの名前を入力します。

BuildTuxedo でヘッダ・ファイルを使用する

BuildTuxedo は、現在の BEA Tuxedo ATMI プロジェクトを正しく作成するために必要な pragma 文をヘッダ・ファイルに追加します。BuildTuxedo は、[Build] ページの [Header File] フィールドに指定されたファイルがあるときはこれを開きます。指定されたファイルがないときは、このファイルを作成します。次の行で始まり、

```
//Begin Tuxedo Section*****DO NOT EDIT*****
```

次の行で終わるセクションは、

```
//End Tuxedo Section
```

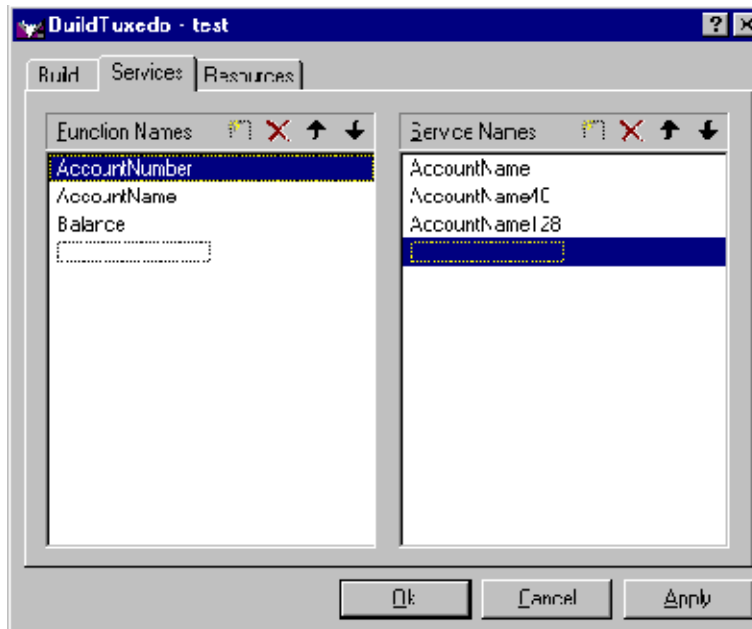

ヘッダ・ファイルで `pragma` 文が書き込まれる部分です。この部分は、`BuildTuxedo` によって管理されます。このセクションが見つからない場合、`BuildTuxedo` はヘッダ・ファイルの最後にこのセクションを付加します。これ以外のファイル内のテキストは変更されないため、ヘッダ・ファイルには「`stdafx.h`」という名前を指定できます。

注記 新しいプロジェクトを作成して [OK] または [Apply] を選択した場合は、MSDEV の [Insert] メニューの [Files into Project] を選択する必要があります。次に、`BuildTuxedo` が生成したファイルを現在のプロジェクトに追加します。この操作は、新しいプロジェクトの場合、または C/C++ 出力ファイルの名前を変更するときのみ実行します。

関数名とサービス名を指定する

[Services] タブを選択すると、[Services] ページが表示されます。[Services] ページでは、関数名とサービス名を指定します。

図 3-3[Service] ページ



[Services] ページでは、サービス・ディスパッチ・テーブルの管理に使用される 2 つの一覧が表示されます。

- [Function Name] は、サービスに関連付ける関数の一覧です。この一覧はユーザが管理します。
- [Service Names] は、関連するサービスの一覧です。この一覧はユーザが管理します。

注記 表を上または下にスクロール表示したいときは、矢印キーを使用します。

通常、サービスと、そのサービスを実行する関数には同じ名前を付けます。たとえば、関数 x はサービス x を実行します。ただし、実行するサービスとは異なる名前が付いた関数もあります。たとえば、関数 abc が、 x 、 y 、および z の 3 つのサービスを実行する場合があります。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。

サーバを構築するときは、サービスに関連する関数を指定する必要があります。サービスに関連する関数には、サービス、プロトタイプ、Cリンク、無効な戻り値、および TPSVCINFO ポインタ・パラメータを1つ指定する必要があります。サービス名をマッピングする関数を指定するには、この関数を [Function Name] 一覧に追加しておく必要があります。この情報は、サービス・ディスパッチ・テーブルに必要です。

注記 `buildserver(1)` コマンドを使用する場合は、`-s` オプションを指定すると、この情報が提供されます。`-s` オプションの詳細については、『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』または『COBOL 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。

[Function Names] に名前を追加または編集するには、次の手順に従います。

1. [Services] フォルダ内で、次のいずれかの操作を選択します。

目的	操作
項目の追加	<ol style="list-style-type: none"> 1. [New] アイコン (左上が点滅している破線ボックスのアイコン) をクリックします。 2. [Insert] をクリックするか、または一覧内にある点線で囲まれたボックスを選択します。 3. 新しい関数の名前を入力します。
既存の名前の変更	名前を強調表示し、上書きします。
名前の削除	一覧内にある、削除対象の名前を強調表示し、[X] アイコン または [Delete] をクリックします。

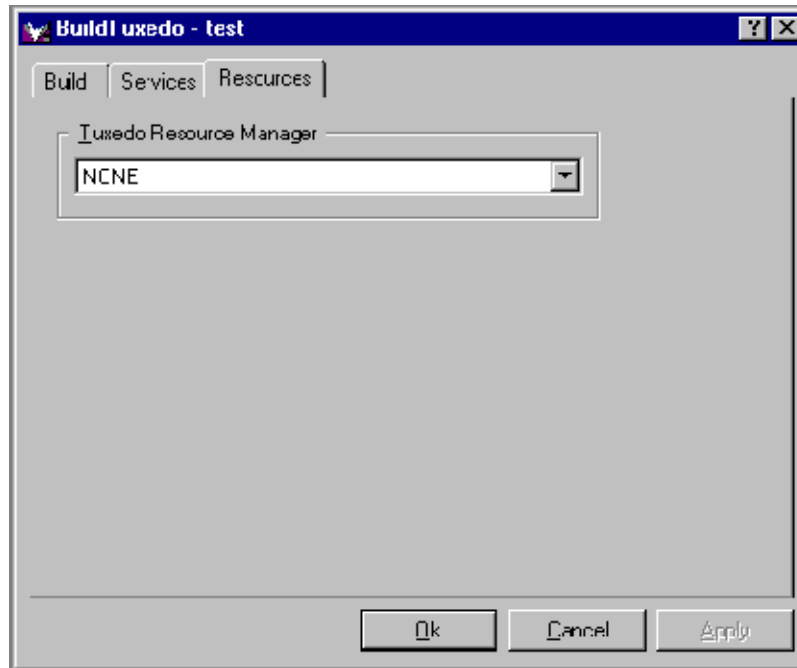
2. 操作が完了したら、[OK] をクリックするか Enter キーを押して変更を保存し、終了します。変更内容を保存しないで終了する場合は、Esc キーまたは Tab キーをクリックします。

リソース・マネージャを指定する

リソース・マネージャを指定するには、次の手順に従います。

1. [BuildTuxedo] ウィンドウの [Resources] タブを選択して、[Resources] ページを表示します。

図 3-4 [Resources] ページ



2. [Resources] ページの [Tuxedo Resource Manager] フィールドに、アプリケーション用のリソース・マネージャのパス名全体を入力します。

注記 [Tuxedo Resource Manager] フィールドには、システムで利用可能な BEA Tuxedo のリソース・マネージャの一覧が用意されています。これらのリソース・マネージャは、%TUXDIR%\udataobj\RM ファイルで定義されています。定義済みのファイルがない場合は、デフォルトの「NONE」が表示されます。

BEA Tuxedo ATMI のサーバ・アプリケーションをデバッグする

次の手順は、プロジェクトでデバッグ・コンフィギュレーションを使用してサーバ・アプリケーションをビルドした場合にのみ適用できます。

まだ起動していないサーバをデバッグするには、次の手順に従います。

1. Windows 2000 のコマンド・プロンプトで「`tmboot -n -d 1 -s servername`」と入力し、`tmboot(1)` が `servername` で指定したサーバを起動するために使用するコマンド行オプションを表示します。
2. `tmboot -M` コマンドを実行し、BBL を起動します。必要に応じて、ほかのアプリケーション・サーバやマシンも起動します。
3. `msdev` で、[Project] の [Settings] を選択します。→
4. [Program Arguments] フィールドの [Debug] ページで、手順 1 で使用したコマンド行オプションを入力します。
5. デバッガを起動し、通常の方法でサーバ・アプリケーションをデバッグします。

注記 BEA Tuxedo のライブラリにはデバッグ情報が組み込まれていない
うえ、ソース・コードも提示されていないため、直接 BEA
Tuxedo のコードにアクセスすることはできません。

6. デバッグ・セッションを終了するには、Windows 2000 のプロンプトで「`tmshutdown`」と入力します。

警告: BEA Tuxedo システムが再起動する可能性がありますので、
[Debug] を選択してから [Stop] を選択する、という手順でサーバを終了しないでください。

注記 実行中のサーバをデバッグする場合は、Windows 2000 のコマンド・
プロンプトに「`msdev -p nnn`」と入力してください。「`nnn`」には、
サーバのプロセス ID (10 進数) を指定します。

Visual C++ IDE GUI の代わりにコマンド行オプションを使用して ATMI アプリケーションを開発する

Microsoft Visual C++ IDE GUI の代わりにコマンド行を使用して BEA Tuxedo ATMI アプリケーションを開発する必要がある場合、`buildserver(1)` コマンドと `buildclient(1)` コマンドを使用します。まず、BEA Tuxedo ATMI アプリケーションのビルドに必要なコンパイラとリンク・オプションを指定します。これらのツールの使用方法については、以下のマニュアルを参照してください。



- 『BEA Tuxedo コマンド・リファレンス』
- 『サンプルを使用した BEA Tuxedo アプリケーションの開発方法』
- また、使用する言語に応じて、以下のマニュアルを参照してください。
 - 『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』
 - 『COBOL を使用した BEA Tuxedo アプリケーションのプログラミング』

`buildserver` または `buildclient` を使用して ATMI アプリケーションのデバッグ・バージョンをビルドするには、`/zi` および `/od` オプションを指定して、すべてのソース・ファイルをコンパイルする必要があります。`/zi` は、デバッグを有効にするオプションです。`/od` は、最適化を無効にするオプションです。さらに、`_DEBUG` プリプロセッサ・ディレクティブの定義が必要な場合もあります。このプロセスを完了するには、次のリンク・オプションを指定します。

```
-l"/link/debug:full /debugtype:both"
```

Tuxdev アプリケーションを使用する

Tuxdev アプリケーションをインストールするには、次の手順に従います。

1. Microsoft Visual C++ 環境 (msdev) で、
[Tools] を選択してから [Customize] を選択します。次に、[Tools] を選択し、[Add] を選択します。
[Customize] ウィンドウの [Tools] ページが表示されます。
2. [Tools] ページの [Menu contents] ペインで [New] アイコン  をクリックして [Menu Contents] ボックスを表示し、[Menu Contents] ボックスの最後の空白のボックスに「&Tuxdev」を入力して、Enter キーを押します。
注記 文字キーをホット・キーとして使用するには、その文字の前にアンパサンド (&) を追加します。
3. [Tools] ページの [Command] フィールドで、tuxdev アプリケーションの実行可能ファイルの絶対パス名を指定します。次のいずれかの方法でパス名を入力できます。
 - [Command] フィールドに、%TUXDIR%\bin\tuxdev.exe を拡張したフル・パス名を入力します。
 - [Command] フィールドの右側にある [...] ボタン  をクリックして、適切なディレクトリで tuxdev.exe を選択し、[Open] をクリックします。選択したパス名が [Customize] ウィンドウに表示されます。
4. [Tools] メニューに表示された項目を変更するには、[Menu Contents] フィールドのエントリを変更します。
注記 ツールのホット・キーを作成するには、ツール名の先頭にアンパサンド (&) を追加します。これで、その文字を入力するだけでツールを呼び出すことができます。

5. [Tools] ページの [Initial directory] フィールドで右矢印ボタンをクリックして、ポップ・アップ・メニューを表示します。ポップ・アップ・メニューから [Current Directory] ($\$(CurDir)$) を選択します。

BEA Tuxedo ATMI のエディタを使用する

この環境では、FML テーブル・エディタと VIEW テーブル・エディタを使用できます。どちらのインターフェイスでも、複数のドキュメントやビューを同時に表示できるため、ワークブックと同じ感覚で操作できます。また、さまざまな種類のファイルを同時に編集することもできます。これらのエディタは、Microsoft Excel のスプレッドシートに似ています。

- FML テーブル・エディタを使用すると、FML フィールド・テーブルの作成および編集を簡単に行うことができるほか、ユーザ側で選択できる 16 ビットまたは 32 ビットのフィールド・ヘッダ・ファイルを生成できます。
- VIEW テーブル・エディタを使用すると、VIEW フィールド・テーブルの作成および編集を簡単に行うことができるほか、ユーザ側で選択できる 16 ビットまたは 32 ビットの VIEWS およびヘッダ・ファイルを生成できます。

FML テーブル・エディタを使用する

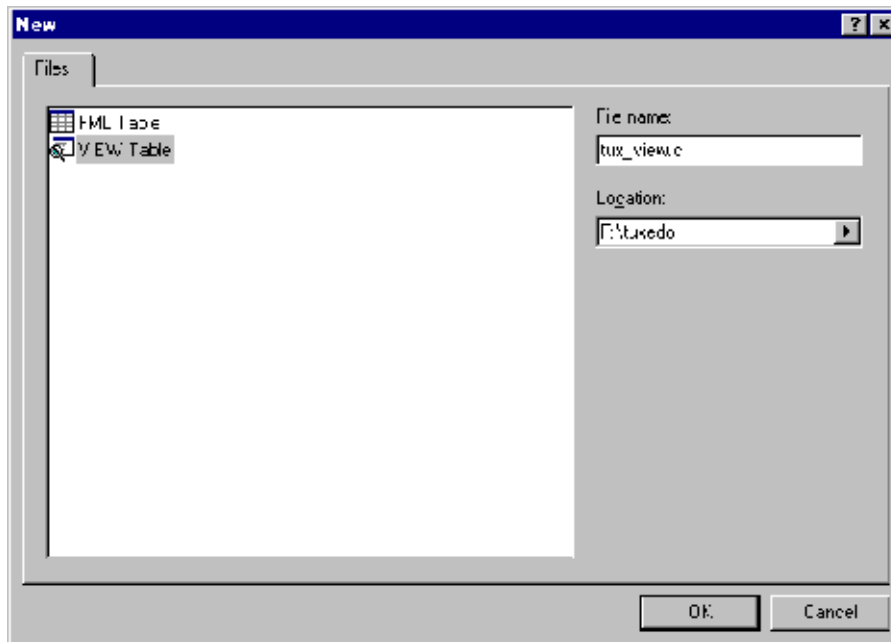
FML テーブル・エディタを起動するには、次の手順に従います。

1. Windows 2000 のコマンド行で、`tuxdev` のパス名全体 ($\%TUXDIR\bin$) を入力し、Enter キーを押します。
2. 「`tuxdev`」を入力して Enter キーを押し、[BEA Tuxedo Developer] ウィンドウを表示します。

3. 次のいずれかの操作を選択します。

目的 ..	操作 ..
ファイルの作成	<ol style="list-style-type: none"> 1. [BEA Tuxedo Developer] ウィンドウの [File] から [New] を選択します。3-19 ページの「[New] ウィンドウ」に示すような [New] ウィンドウが表示されます。 2. 手順 4 に進みます。
既存ファイルの変更	<ol style="list-style-type: none"> 1. [BEA Tuxedo Developer] ウィンドウの [File] から [Open] を選択します。 2. 編集するファイルを選択します。FML ファイルには .fml という拡張子が付きます。 3. 残りの手順は省略します。

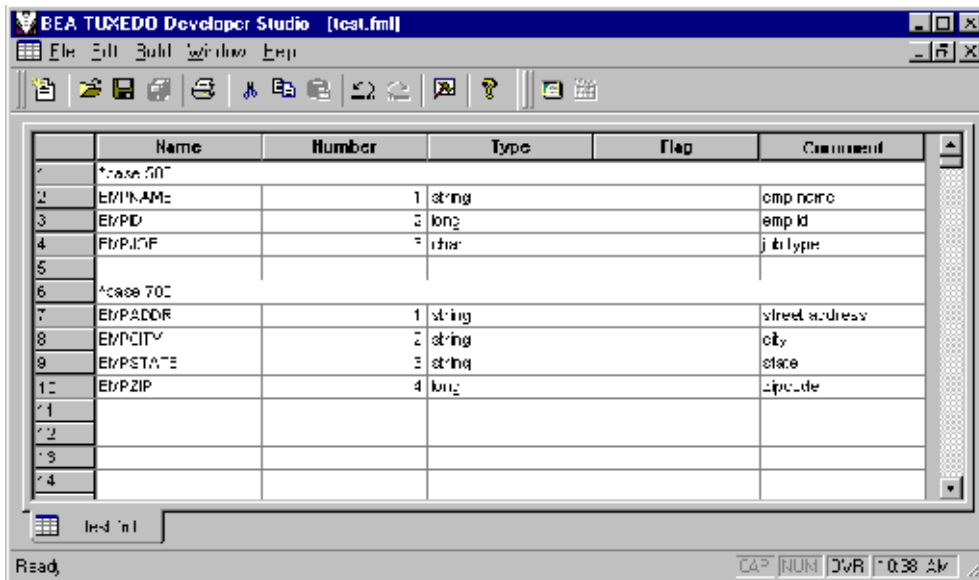
図 3-5[New] ウィンドウ



4. [New] ウィンドウで [FML Table] を選択し、[File Name] フィールドに新しいファイルの名前を入力します。

5. [OK] をクリックして、FML テーブル・エディタを起動します。

図 3-6FML テーブル・エディタ



前の図に示すように、FML テーブル・エディタには、[Name]、[Number]、[Type]、[Flag]、[Comment]、の 5 つの列があります。行数の制限はありません。

次の表は、FML テーブル・エディタの各列の説明です。

表 3-2FML テーブル・エディタの列

列 . .	目的 . .
Name	コメントまたはベース番号、あるいはその両方を入力します。コメントまたは番号の先頭にはシャープ記号 (#) またはアスタリスク (*) が必要です。これらの記号がない場合は、アクティブなテーブル・エントリと見なされます。空のセルに入力されたデータは、新しいエントリと見なされます。空白行があってもかまいません。
Number	フィールドの相対数値を指定します。

表 3-2FML テーブル・エディタの列

列 ..	目的 ..
タイプ	このフィールドに指定可能な値の一覧から型を選択します。
フラグ	フラグの設定を選択します (列がアクティブな場合のみ)。
Comment	[Name] に指定されたエントリの詳細を入力します。

新しい (名前の付いていない) FML テーブルを開くと、FML Table x というテーブル名で空白行が作成されます。x は、MDI によって割り当てられ、新しいテーブルが作成されるたびに 1 つずつ増えます。テーブルを保存するときは名前を指定することができます。また、既存のテキスト・ファイルを開いて、編集することもできます。特に指定されていない限り、ファイルは、タブ区切り形式で、ファイルを開いたディレクトリ内に保存されます。保存時には、必要に応じて、ファイル名の最後に拡張子 `.fml` が追加されます。このファイルをコンパイルして、16 ビットまたは 32 ビットの FML ヘッド・ファイルを作成することもできます。

VIEW テーブル・エディタを使用する

VIEW テーブル・エディタを起動するには、次の手順に従います。

1. Windows 2000 のコマンド行で、`tuxdev` のパス名全体 (`%TUXDIR%\bin`) を入力し、Enter キーを押します。
2. 「`tuxdev`」を入力して Enter キーを押し、[BEA Tuxedo Developer] ウィンドウを表示します。
3. 次のいずれかの操作を選択します。

表 3-3View テーブル・エディタの操作

目的 ..	操作 ..
ファイルの作成	<ol style="list-style-type: none">1. [BEA Tuxedo Developer] ウィンドウの [ファイル] から [New] を選択して、[New] ウィンドウを表示します (3-19 ページの「[New] ウィンドウ」参照)。2. 手順 4 に進みます。
既存ファイルの変更	<ol style="list-style-type: none">1. [BEA Tuxedo Developer] ウィンドウの [File] から [Open] を選択します。2. 編集するファイルを選択します。FML ファイルには .fml という拡張子が付きます。3. 残りの手順は省略します。

4. [New] ウィンドウで [VIEW Table] を選択し、[File Name] フィールドに新しいファイルの名前を入力します。
5. [OK] をクリックして、VIEW テーブル・エディタを起動します。

図 3-7VIEW テーブル・エディタ

図が示すように、VIEW テーブル・エディタには、[CName]、[FBName]、[Type]、[Count]、[Flag]、[Size]、[Null] の 7 つの列があります。行数の制限はありません。

[CName] 列にはコメントを入力できます。ただし、コメントの先頭にはシャープ (#) を付ける必要があります。空白行はあってもかまいません。[CName] 列のエントリの先頭に # がなく、ヌルでもない場合、この行はアクティブなテーブル・エントリと見なされます。

次の表は、VIEW テーブル・エディタの各列の説明です。

表 3-4VIEW テーブル・エディタの列

列 ..	目的 ..
CName	コメントまたはベース番号、あるいはその両方を入力します。コメントまたは番号の先頭にはシャープ記号 (#) またはアスタリスク (*) が必要です。これらの記号がない場合は、アクティブなテーブル・エントリと見なされます。空白行があってもかまいません。
FBName	フィールド化バッファ内のフィールド名を指定します。この名前は、フィールド・テーブル・ファイルに表示されます。
タイプ	このフィールドに指定可能な値の一覧から型を選択します。
Count	割り当てる要素の数、つまり、このメンバに保存するオカレンスの最大数を指定します。
フラグ	フラグの設定を選択します (列がアクティブな場合のみ)。
Size	Size には、Type (型) が string、carray、または dec_t である場合のメンバのサイズを指定します。その他の型に対しては「-」を指定します。この場合は、View コンパイラがサイズを計算します。
NULL	このフィールドに対し、NULL または「-」(デフォルト) を指定します。

注記 VIEW 情報の始めと終わりを示すため、次の情報を指定する必要があります。

```
VIEW table_name
.
.
.
END
```

この情報は、[CName] 列の行に表示されなければなりません。同じファイル内に複数の VIEW を入力することもできます。ただし、各テーブル・エントリは必ず「VIEW table_name」と「END」で囲んでください。

新しい(名前が付いていない)VIEW ファイルを開くと、View x という VIEW 名で空白行が作成されます。 x は、MDI によって割り当てられ、新しい VIEW が作成されるたびに 1 つずつ増えます。VIEW を保存するときは名前を指定することができます。また、既存の VIEW ファイル(テキスト形式またはバイナリ形式)を開いて、編集することもできます。特に指定されていない限り、ファイルは、タブ区切り形式で、ファイルを開いたディレクトリ内に保存されます。保存時には、必要に応じて、ファイル名の最後に拡張子 `.v` が追加されます。このファイルをコンパイルし、16 ビットまたは 32 ビットの VIEW ヘッダ・ファイルを作成することもできます。

複数のドキュメントを同時に処理する

基本的なフレームワークに組み込まれている MDI により、複数の異なる形式のドキュメントや VIEW を同時に開くことができます。たとえば、`msdev`、`Excel`、`Word` などがその良い例です。BEA Tuxedo では、 x 個の FML テーブルと、 y 個の VIEW ファイルを開き、それらの 1 つを使用することができます。各ドキュメントは、オープン状態のドキュメントごとにタブが付いたワークブックのようなものです。

エディタを使用してエントリの妥当性を検証する

次の表は、FML テーブル・エディタの各列で検証される情報の説明です。

表 3-5FML テーブル・エディタで検証される情報

列名	検証の対象...
..	
Name	コメント、ベース番号、および有効なテキスト文字列。
Number	数値のみ。範囲は、16/32 ビットのユーザ・モードで決まります。
タイプ	有効な FML 型。

表 3-5FML テーブル・エディタで検証される情報

列名	検証の対象..
..	
フラグ	有効な FML フラグ。相互に排他的なフラグについては、さらに検証が行われます。
Comment	このエントリは検証されません。

次の表は、VIEW テーブル・エディタの各列で検証される情報の説明です。

表 3-6VIEW テーブル・エディタで検証される情報

列名..	検証の対象..
CName	このエントリは検証されません。
FBName	このエントリは検証されません。
タイプ	有効な BEA Tuxedo の種類。
Count	数値のみ。範囲は、16/32 ビットのユーザ・モードで決まります。
フラグ	有効な FML フラグ。相互に排他的なフラグについては、さらに検証が行われます。
Size	数値のみ。範囲は、16/32 ビットのユーザ・モードで決まります。
NULL	このエントリは検証されません。

関連項目

BEA Tuxedo ATMI については、以下のマニュアルを参照してください。

- 『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』

- 『COBOL 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』
- 『TxRPC 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 コマンド・リファレンス』
- 『BEA Tuxedo C リファレンス』
- 『BEA Tuxedo COBOL リファレンス』
- 『BEA Tuxedo のファイル形式とデータ記述方法』

