



BEA Tuxedo

Windows での
BEA Tuxedo の使用

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

Copyright

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

Restricted Rights Legend

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

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

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

Trademarks or Service Marks

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

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

Windows での BEA Tuxedo の使用

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

目次

このマニュアルについて	
対象読者	v
e-docs Web サイト	v
マニュアルの印刷方法	vi
サポート情報	vi
表記上の規則	vii
1. UNIX 版と Windows 2000 版の BEA Tuxedo ATMI の違い	
Windows 2000 プラットフォームに関する留意事項	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	1-3
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	1-4
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	1-5
2. BEA Tuxedo ATMI for Windows 2000 のコンフィギュレーション	
コントロール・パネルを使用して BEA Tuxedo ATMI を コンフィギュレーションする	2-2
ネットワーク上のマシンにアクセスする	2-4
環境変数を変更する	2-6
BEA Tuxedo のシステム・メッセージを Windows 2000 のイベント・ ログに転送する	2-7
ログを設定する	2-8
ログを表示する	2-8
tlisten プロセスを自動的に起動する	2-10
システムのパフォーマンスを最大化する	2-12
3. Microsoft Visual C++ 環境での BEA Tuxedo ATMI アプリケーションの開発	
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	3-2
MSDEV の [Tools] メニューに BuildTuxedo を追加する	3-4
BEA Tuxedo のプロジェクト・ファイルを作成する	3-5
環境を設定する	3-6
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	3-7

BuildTuxedo でヘッダ・ファイルを使用する	3-8
関数名とサービス名を指定する	3-9
リソース・マネージャを指定する	3-11
BEA Tuxedo ATMI のサーバ・アプリケーションをデバッグする	3-12
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	3-13
Tuxdev アプリケーションを使用する	3-14
BEA Tuxedo ATMI のエディタを使用する	3-15
FML テーブル・エディタを使用する	3-15
VIEW テーブル・エディタを使用する	3-17
複数のドキュメントを同時に処理する	3-20
エディタを使用してエントリの妥当性を検証する	3-20

このマニュアルについて

このマニュアルでは、BEA Tuxedo® ATMI を Microsoft Windows 2000 で使用方法について説明します。

このマニュアルでは、以下の内容について説明します。

- 「第 1 章 UNIX 版と Windows 2000 版の BEA Tuxedo ATMI の違い」では、BEA Tuxedo ATMI を UNIX と Windows 2000 で使用する場合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 に設定する方法を説明します。
- 「第 3 章 Microsoft Visual C++ 環境での BEA Tuxedo ATMI アプリケーションの開発」では、Microsoft Visual C++ 統合開発環境 (msdev) で ATMI アプリケーション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 2000 プラットフォームで BEA Tuxedo ATMI アプリケーションを開発するアプリケーション開発者を対象にしています。

e-docs Web サイト

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

マニュアルの印刷方法

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

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

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

サポート情報

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

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

BEA Tuxedo に関するご質問、または BEA Tuxedo のインストールや使用に際して問題が発生した場合は、www.bea.com の BEA WebSUPPORT を通して 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。
イタリック体	強調またはマニュアルのタイトルを示します。
等幅テキスト	コード・サンプル、コマンドとオプション、データ構造とメンバ、データ型、ディレクトリ、およびファイル名と拡張子を示します。また、キーボードから入力する文字も示します。 例： <pre>#include <iostream.h> void main () the pointer psz chmod u+w * ¥tux¥data¥ap .doc tux.doc BITMAP float</pre>
等幅太字	コード内の重要な単語を示します。 例： <pre>void commit ()</pre>
等幅イタリック体	コード内の変数を示します。 例： <pre>String <i>expr</i></pre>
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： <pre>LPT1 SIGNON OR</pre>
{ }	構文の行で選択肢を示します。かっこは入力しません。

規則	項目
[]	<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>
...	<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> <p>.</p> <p>.</p>

1 UNIX 版と Windows 2000 版の BEA Tuxedo ATMI の違い

ここでは、次の内容について説明します。

- Windows 2000 プラットフォームに関する留意事項
- 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 2000 プラットフォームに関する留意事項

BEA Tuxedo ATMI を Windows 2000 プラットフォームで使用する場合は、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 2000 用であると見なされます。
- ファイル名は、Windows 2000 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 プラットフォー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 のアプリケーション・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)"
```

たとえば、次のように指定します。

```
OPENINFO="TUXEDO¥SQL:APPDIR1¥bankdl3;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 プラットフォームの場合は、異なる値を設定します。

UNIX マシンと Windows 2000 マシン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 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()` で開放すると、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 プラットフォー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 のクライアント (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 のサーバ (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 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%\bankapp\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 がサーバとして Windows 2000 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 および rktools.hlp を参照してください。

BEA Tuxedo ATMI 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 Tuxedo のコントロール・パネルを使用して設定します。詳細については、第 2 章の 6 ページ「環境変数を変更する」を参照してください。

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 言語リファレンス』
- 『BEA Tuxedo COBOL リファレンス』
- 『BEA Tuxedo FML リファレンス』
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』

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

ここでは、次の内容について説明します。

- コントロール・パネルを使用して BEA Tuxedo ATMI をコンフィギュレーションする
- ネットワーク上のマシンにアクセスする
- 環境変数を変更する
- tlisten プロセスを自動的に起動する
- システムのパフォーマンスを最大化する

注記 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。

コントロール・パネルを使用して BEA Tuxedo ATMI をコンフィギュレーションする

BEA Tuxedo ATMI for Windows 2000 には、BEA Administration Console のほか、BEA Tuxedo ATMI for Windows 2000 をコンフィギュレーションするためのコントロール・パネルが用意されています。コントロール・パネルで実行できるタスクは、以下のとおりです。

- ネットワーク上のマシンにアクセスできます ([Machines] ページで設定)。
- 環境変数を変更できます ([Environment] ページで設定)。
- BEA Tuxedo のシステム・メッセージを Windows 2000 イベント・ログに送ります ([Logging] ページで設定)。
- 1 つまたは複数の `tlisten` プロセスを設定して、これらのプロセスを自動的に起動できます ([Listener] ページで設定)。
- IPC 資源を調整して、システムのパフォーマンスを最大化できます ([Resources] で設定)。

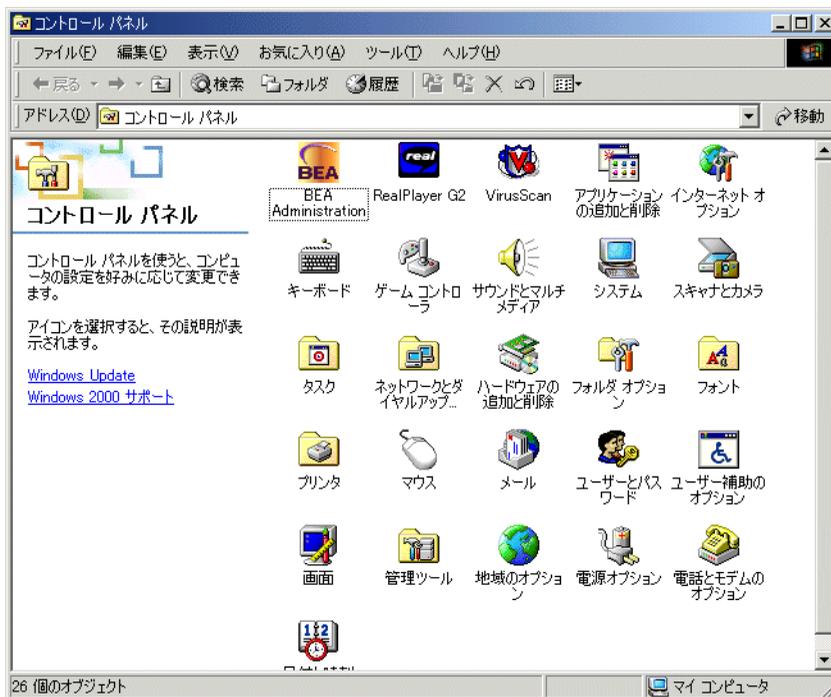
コントロール・パネルを開くには、次の手順に従います。

1. Windows 2000 のデスクトップで次の順にクリックします。

[スタート]メニューの [設定] [コントロールパネル] を選択します。

次の図に示すような Microsoft Windows のコントロール・パネルが表示されます。

図 2-1 Microsoft Windows のコントロール・パネル



2. [BEA Administration] アイコンをダブルクリックして、コントロール・パネル(図 2-2 参照)を開きます。

ネットワーク上のマシンにアクセスする

管理者は、コントロール・パネルの [BEA Administration] にある [Machines] ページで、Microsoft Windows ネットワーク上にある次の条件のマシンにアクセスできます。

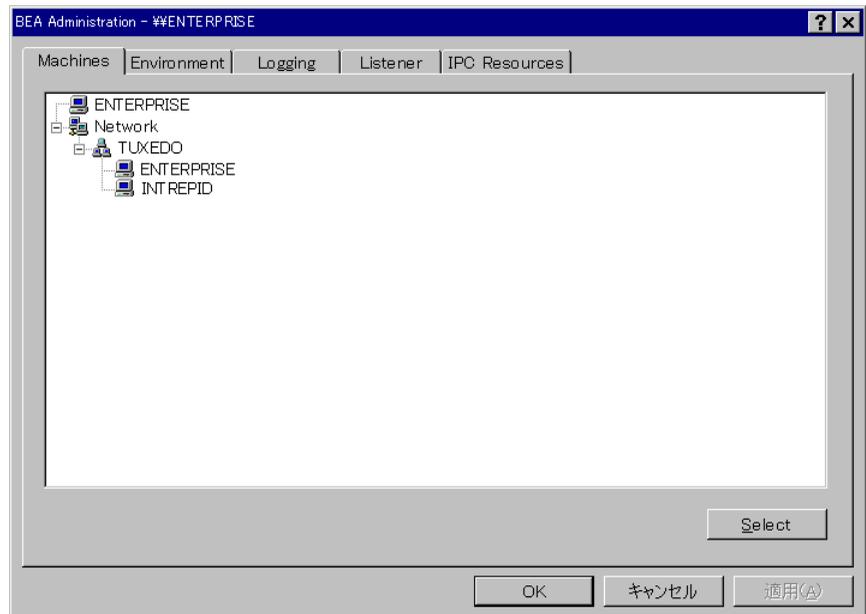
- Windows 2000 を実行するマシン
- 管理者側にログオン権限があるマシン

これらのマシンに対し、管理者は以下のタスクを実行できます。

- 環境変数をリモート設定する
- BEA Tuxedo のイベント・ログを保存する場所を決定する
- tlisten サービスの追加、開始、削除を行う
- IPC 資源を調整する

図 2-2 は、[Machines] ページを示しています。

図 2-2 [Machines] ページ



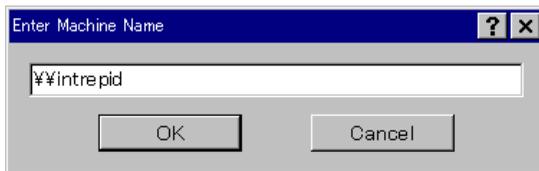
リモート・マシンをネットワーク・ツリー内に配置すると、管理者はそのリモート・マシンを選択するだけでアクセスできますし、図 2-2 では、ローカル・マシン ENTERPRISE が選択されています。

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. [Select] をクリックします。

図 2-3 に示す [Enter Machine Name] ダイアログ・ボックスが表示されます。

図 2-3 [Enter Machine Name] ダイアログ・ボックス



2. フィールドにリモート・マシンの名前を入力し、[OK] をクリックします。たとえば、「¥¥intrepid」と入力します。

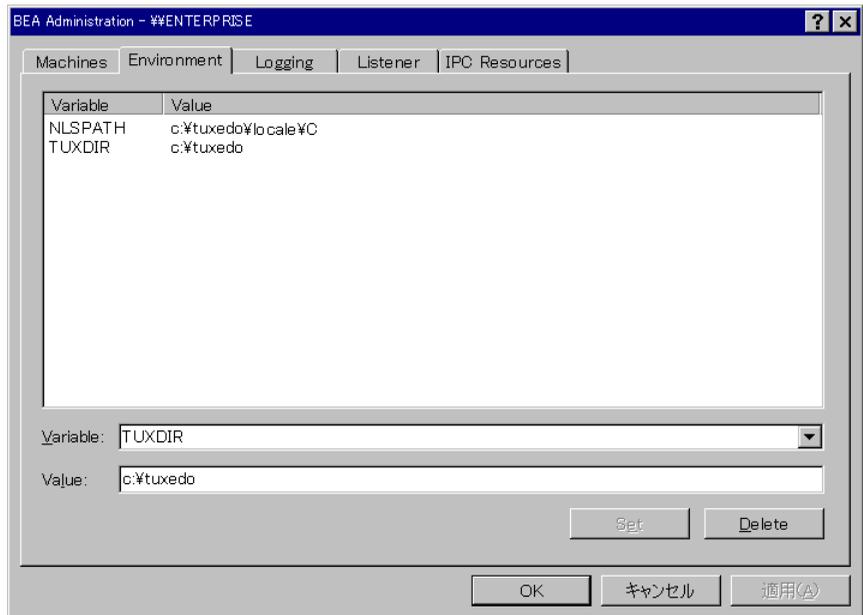
この手順を完了すると、以降、コントロール・パネルのこれ以外のページ ([Environment]、[Logging]、[Listener]、[IPC Resources]) で行うすべての操作は、選択したマシンに対して実行されます。

環境変数を変更する

管理者は、BEA アプリケーションのコントロール・パネルにある [Environment] ページを使用して、Windows 2000 の環境変数を変更する場合と同じように、BEA Tuxedo の環境変数を変更できます。

図 2-4 は、[Environment] ページを示しています。

図 2-4 [Environment] ページ



[Variable] フィールドには、最も頻繁に使用される BEA Tuxedo の環境変数が一覧表示されています。

環境変数を変更するには

1. 環境変数を追加するには、次の手順に従います。
 - a. 現在のリストから任意の環境変数を選択し、[Variable] フィールドで変数を選択し、新しい変数に名前を付けます。
 - b. [Value] フィールドで値を選択して実際の値を入力し、[Set] をクリックします。新しい変数がリストに追加されます。

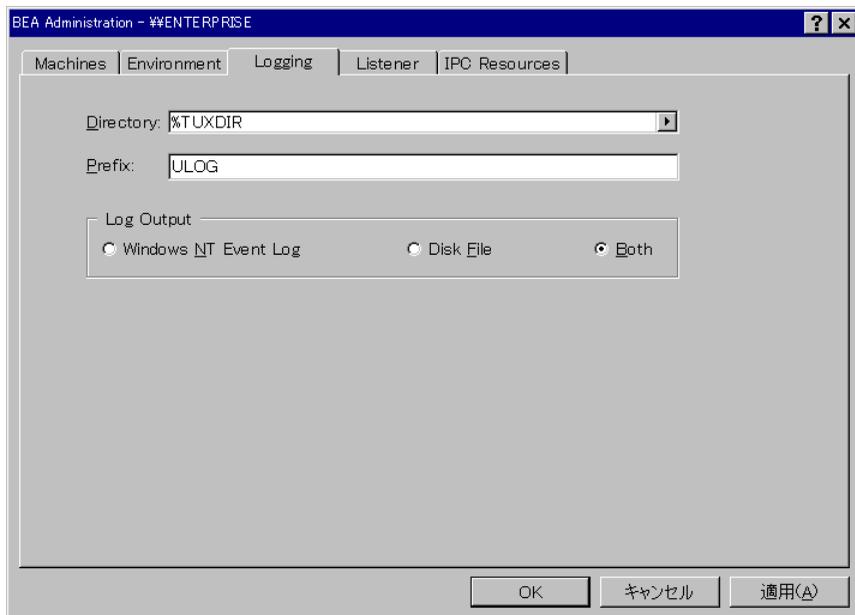
2. 変数を編集するには、現在のリストから変数を選択し、[Value] フィールドで変数を選択して新しい値を入力し、[Set] をクリックします。リスト内の変数は更新されます。
3. 変数を削除するには、現在のリストから変数を選択して [Delete] をクリックします。変数がリストから削除されます。
4. [Set] をクリックします。環境変数に対する変更が適用されます。

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

[BEA Administration] コントロール・パネルにある [Logging] ページを使用すると、BEA Tuxedo のシステム・メッセージを Windows 2000 のイベント・ログに転送できます。

図 2-5 は、[Logging] ページを示しています。

図 2-5 [Logging] ページ



[Tuxedo Log Output] ボックスで、3つのオプション ([Windows 2000 Event Log] (ログ・オプション)、[Disk File] (従来の ULOG)、[Both]) の中から 1つ選択します。

ログを設定する

この節では、ログのタイプを選択した後でログを設定する方法を説明します。

Windows 2000 イベント・ログを設定する

セットアップは不要です。

ULOG を設定する

ULOG メッセージを記録する場合は、次の手順に従って保存ディレクトリを選択します。

1. [Logging] ページ上部の [Directory] フィールドの右にある [Browse] ボタンをクリックします。ディレクトリのドロップ・ダウン・リストが表示されます。
2. ULOG メッセージを保存するディレクトリの名前をクリックします。
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 です。デフォルトのファイル名の形式は ULOG.mmdyyy です。

ログを表示する

この節では、Windows 2000 のイベント・ログと ULOG を表示する手順を説明します。

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・ログの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]をクリックします。

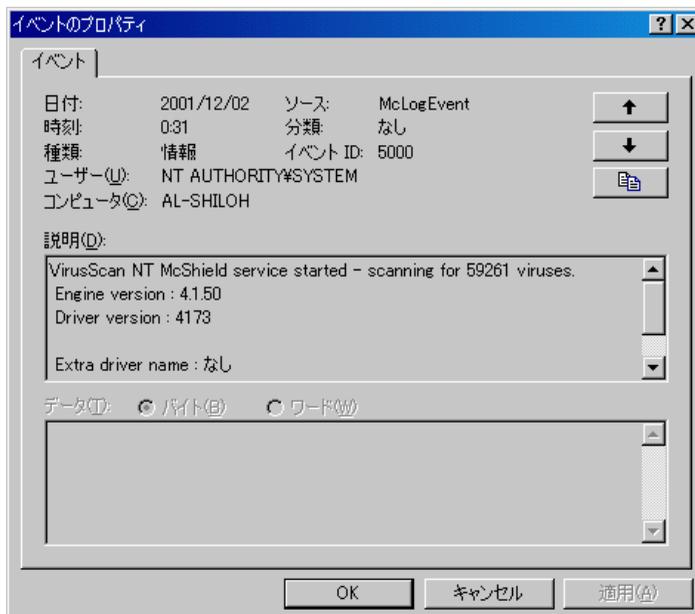
図 2-6 に示すように、アプリケーション起動後に発生したすべてのイベントの一覧が表示されます。

図 2-6 [イベント ビューア] ウィンドウ



2. 左ペインのコンソール ツリーから [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。
3. 見たいイベントのエントリをダブルクリックして詳細情報を表示します。Windows 2000 の [イベントの詳細] ウィンドウが開き、指定したイベントに関する情報が表示されます (図 2-7 参照)。

図 2-7 [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。情報のみ、警告、エラー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

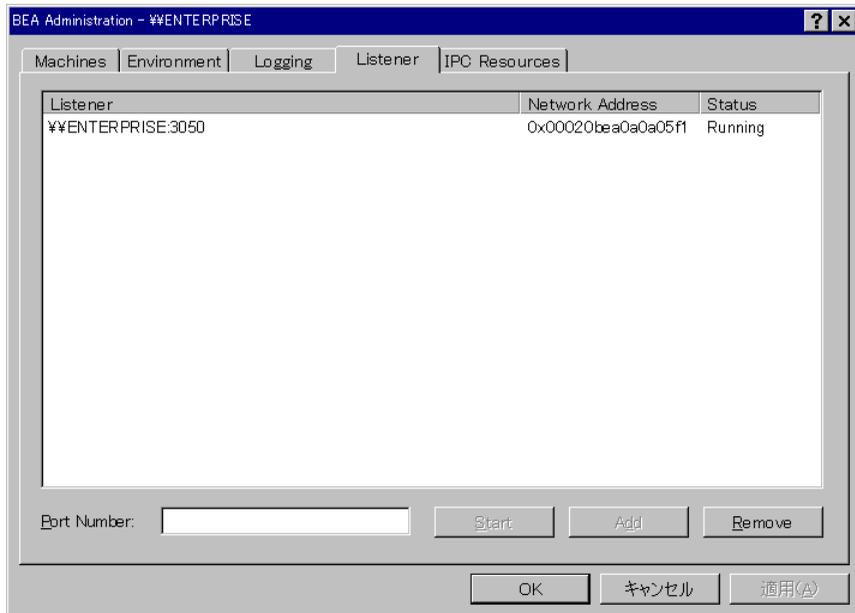
ULOG はテキスト・ファイルなので、テキスト・エディタで ULOG を開き、内容を表示することができます。

tlisten プロセスを自動的に起動する

[BEA Administration] コントロール・パネルにある [Listener] ページを使用すると、1 つまたは複数の `tlisten(1)` プロセスがシステムの起動時に自動的に起動するようにコンフィギュレーションできます。

図 2-8 は、[Listener] ページを示しています。

図 2-8 [Listener] ページ



リスナを追加するには、次の手順に従います。

1. [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。
2. [OK] をクリックします。
3. 新しく追加したリスナを起動するか停止するかに応じて、[Start] または [Stop] ボタンをクリックします。

注記 Windows 2000 サービスのコントロール・パネルを使って、tlisten サービスを起動または停止したり、tlisten サービスを自動的に起動するようにコンフィギュレーションすることもできます。

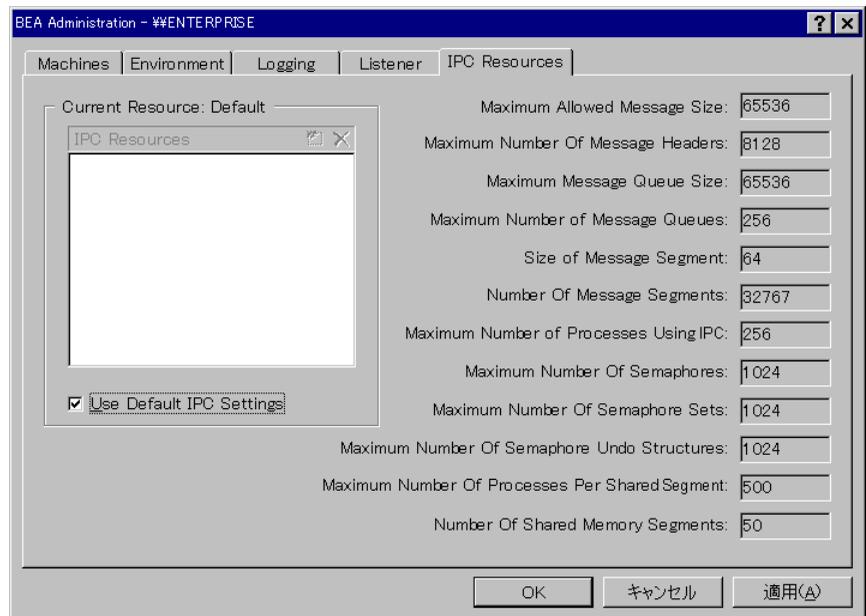
tlisten(1) プログラムを使用すると、複数のコンピュータに渡るアプリケーションを管理することができます。tlisten プログラムは、アプリケーションを実行する前に各コンピュータ上で起動しておく必要があります。通常、コンピュータ上の各 BEA Tuxedo ATMI アプリケーションに対して 1 つの tlisten が必要です。

システムのパフォーマンスを最大化する

BEA Tuxedo ATMI for Windows 2000 には、プロセス間通信のサブシステムである BEA Tuxedo IPC Helper (TUXIPC) が同梱されています。IPC Helper は、ほとんどのシステムではインストール時の設定で実行されますが [BEA Administration] のコントロール・パネルにある [IPC Resources] ページを使用して、TUXIPC サブシステムを調整し、パフォーマンスを最大化することもできます。

図 2-9 は、[IPC Resources] ページを示しています。

図 2-9 [IPC Resources] ページ



[IPC Resources] ページでは、さまざまな IPC 資源を設定することができます。BEA Tuxedo マシンの IPC 資源を設定するには、次の操作を行います。

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。
2. [IPC Resources] ペインの [New (Insert)] アイコン  をクリックします。
3. 使用するマシンの名前を入力して、Enter キーを押します。

4. 設定する IPC 資源の隣にあるフィールドをクリックし、必要な値を入力して、[Apply] をクリックします。[Apply] をクリックすると、変更内容がレジストリ・テーブルに保存されます。変更を有効にするには、Tuxedo IPC Helper を停止して再起動する必要があります。
5. [OK] をクリックして BEA Administration Console を閉じます。

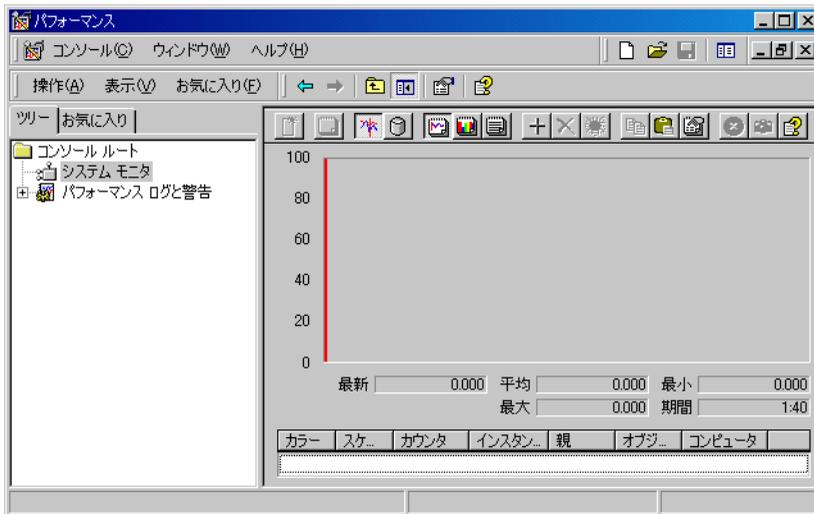
[IPC Resources] ページの [Maximum Number of Processes Using IPC] パラメータを指定する場合は、次の点に注意してください。

- 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 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 ATMI アプリケーションのパフォーマンスは、図 2-10 のようなパフォーマンス・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、Windows 2000 のタスクバーで [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] から [パフォーマンス] をクリックします。パフォーマンス・モニタの画面が表示されます (図 2-10 参照)。

図 2-10 パフォーマンス・モニタ



3 Microsoft Visual C++ 環境での BEA Tuxedo ATMI アプリケー ションの開発

ここでは、次の内容について説明します。

- 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

開発ツールを使用する

BEA Tuxedo ATMI は、Microsoft Visual C++ (msdev) 環境に統合されますが、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

buildserver(1) コマンドおよび buildclient(1) コマンドは、Windows 2000 プラットフォームでも使用できます。ただし、Windows 2000 プラットフォームの場合とそれ以外の場合では、コマンド・オプションの使い方が異なります。次の表は、これらの違いをまとめたものです。

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。
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] タブを選択します。 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 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。 BuildTuxedo の [Test] ウィンドウを開き、[Resources] ページを選択します。 2. [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。

統合開発環境 (IDE: Integrated Development Environment) 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 [Build] 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。

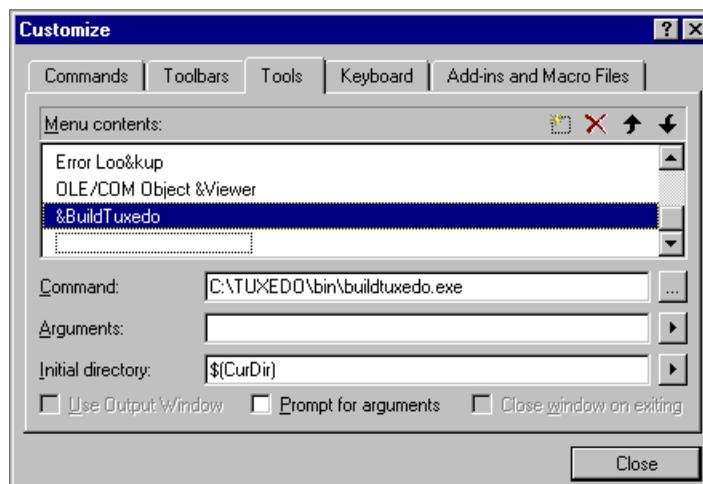
1. [MSDEV Tools] メニューの [Options] を選択すると、[Options] ダイアログ・ボックスが表示されます。
2. [Options] ダイアログ・ボックスで、[Directories] タブを選択します。

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++ (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. [Tool] タブをクリックすると、図 3-1 のように [Tool] タブが表示されます。

図 3-1 [Customize] ダイアログ・ボックスの [Tools] タブ



3. [Menu contents] ペインの [New] アイコン  をクリックし、[Menu contents] ボックスの一番下にある空白のボックス内に「&BuildTuxedo」と入力します。次に 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。

4. 次のいずれかの方法で [Command] フィールドにパス名を入力します。
 - 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. [Initial Directory] フィールドの右矢印ボタンをクリックします。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] をクリックします。

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

BEA Tuxedo のプロジェクト・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・ファイルが見つかったら、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

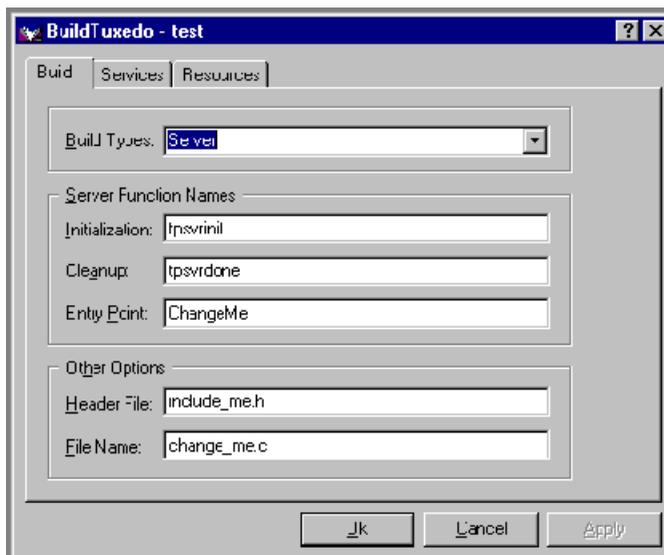
これらの情報を設定するには、msdev の GUI 画面で、[BuildTuxedo project_name] ダイアログ・ボックスを表示します。

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・ボックスの [Build] タブをクリックします。

図 3-2 [Build] ページ



2. [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。

- Server (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 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
Server	[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	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、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。

- [Header File] フィールドに、「stdafx.h」と入力します。
- [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
```

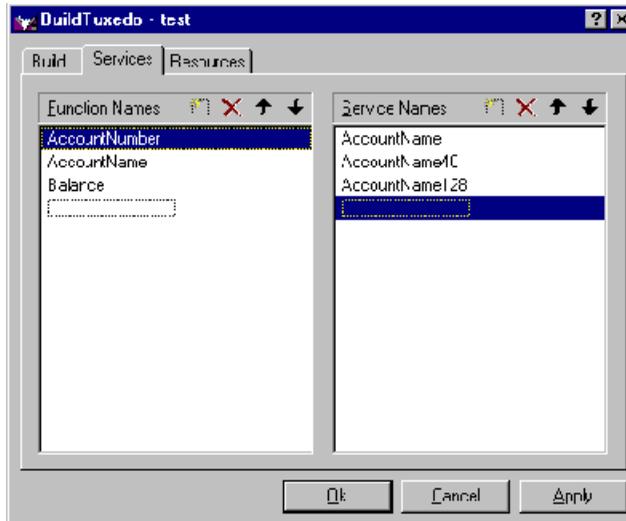
ヘッダ・ファイルで pragma 文が書き込まれる部分です。この部分は、BuildTuxedo によって管理されます。このセクションが見つからない場合、BuildTuxedo はヘッダ・ファイルの最後にこのセクションを付加します。これ以外のファイル内のテキストは変更されないため、ヘッダ・ファイルには「stdafx.h」という名前を指定できます。

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

関数名とサービス名を指定するには、[Services] タブを選択します。次の図は、[Services] ページを示しています。

図 3-3 [Service] ページ



ここでは、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ています。

- [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。また、実行時までサービス名が分からない場合もあります。

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。

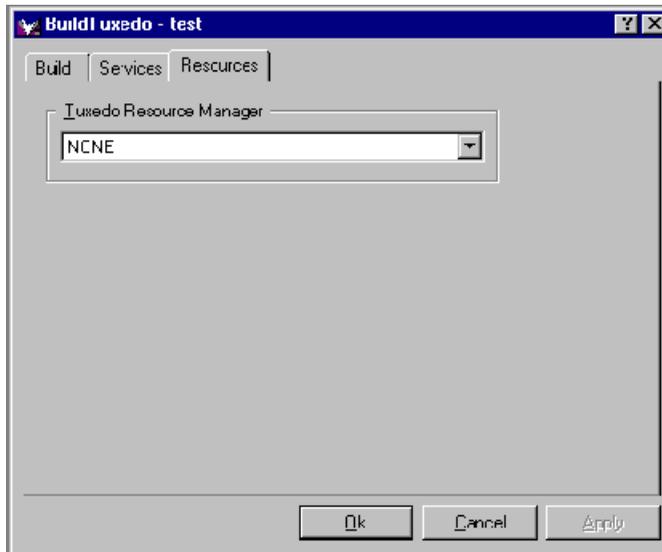
4. 操作が完了したら、[OK] をクリックするか [Enter] を押して変更を保存し、終了します。変更内容を保存しないで終了する場合は、Esc キーまたは Tab キーをクリックします。

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] メニューの [Resources] タブを選択して、次の図のような [Resources] ページを表示します。

図 3-4 [Resources] ページ



2. [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%\dataobj\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

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。

注記 この手順は、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。

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] フィールドで [Debug] を選択し、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. デバッグを起動し、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 のコマンド・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++ 統合開発環境 (IDE) のグラフィカル・ユーザ・インターフェイス (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。

- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』
- 『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、以下のマニュアルも参照してください。
 - 『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++ (msdev) 環境で、[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。

2. [Menu contents] ペインで、[New] アイコン  をクリックし、[Menu Contents] ボックスの一番下にある空白のボックス内に「&Tuxdev」と入力します。次に、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バサンド (&) を追加します。

3. [Customize] ダイアログ・ボックスの [Tools] タブで、tuxdev アプリケーションの実行可能ファイルのフル・パス名を [Command] フィールドに入力します。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バサンド (&) 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. [Initial Directory] フィールドの右矢印ボタンをクリックします。ポップ・アップ・メニューが表示されます。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。

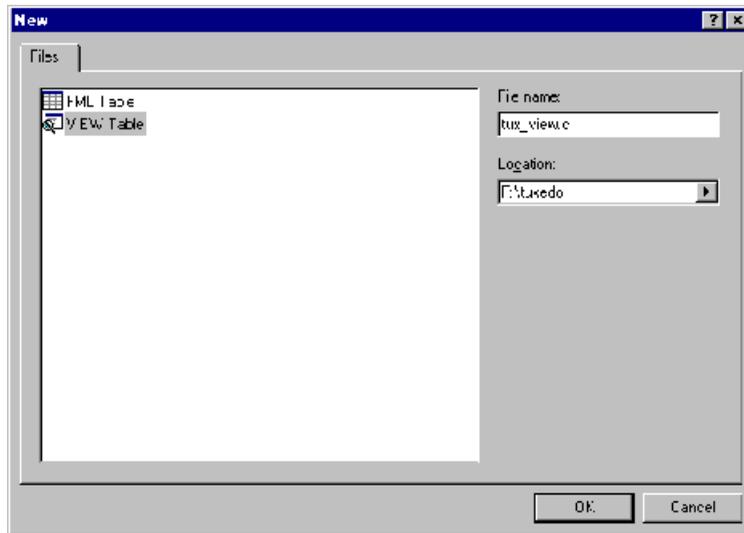
FML テーブル・エディタを使用する

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 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」と入力して Enter キーを押すと、[BEA Tuxedo Developer] ウィンドウが表示されます。
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。[New] ウィンドウが表示されます (図 3-5 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。

図 3-5 [New] ウィンドウ



4. [New] ウィンドウで [FML Table] を強調表示し、[File Name] フィールドに新しいファイル名を入力します。
5. [OK] を選択します。FML テーブル・エディタが起動し、図 3-6 に示すような画面が表示されます。

図 3-6 FML テーブル・エディタ

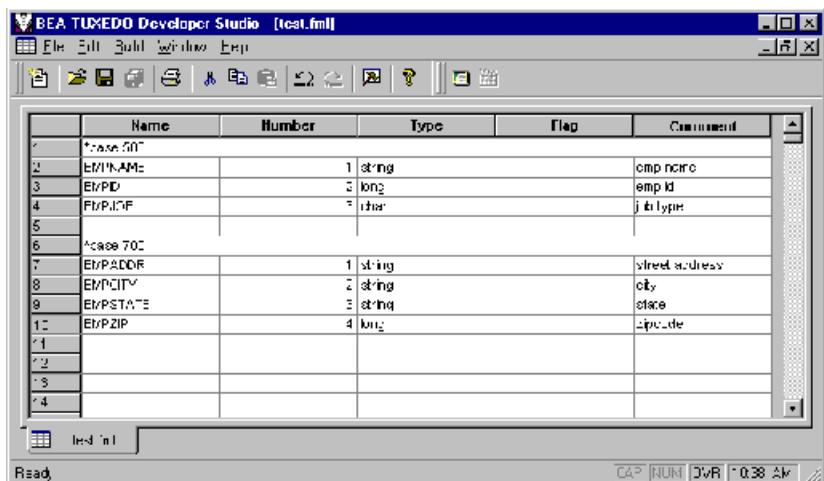


図 3-6 が示すように、FML テーブル・エディタには、[Name]、[Number]、[Type]、[Flag]、[Comment]、の 5 つの列があります。行数の制限はありません。

次の表は、FML テーブル・エディタの各列の説明です。

表 3-2 FML 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。
Type	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。
Flag	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 Tablex 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 ヘッダ・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

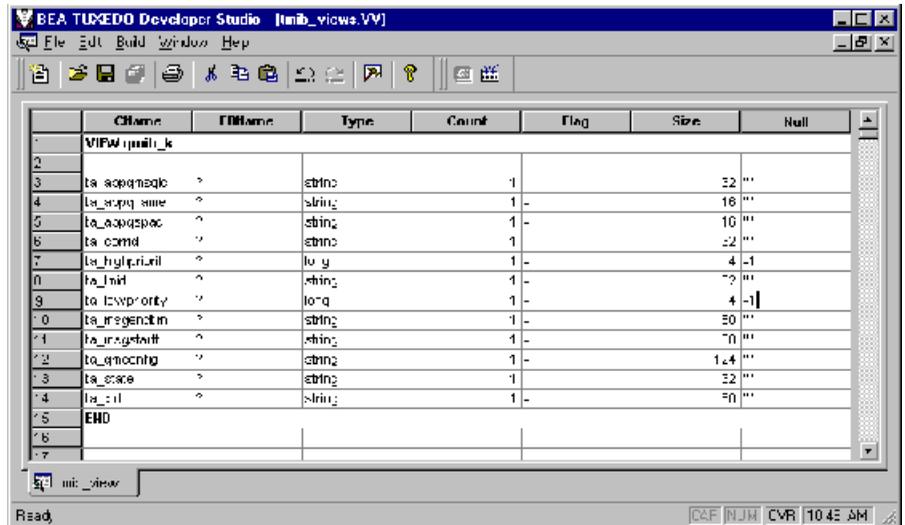
VIEW テーブル・エディタを起動するには、次の手順に従います。

1. Windows 2000 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」と入力して Enter キーを押します。[BEA Tuxedo Developer] ウィンドウが表示されます。
3. 次のいずれかの操作を選択します。

表 3-3

目的	操作
ファイルの作成	<ol style="list-style-type: none"> [BEA Tuxedo Developer] ウィンドウの [File] から [New] を選択します (図 3-5 参照)。 手順 4 に進みます。
既存ファイルの変更	<ol style="list-style-type: none"> [BEA Tuxedo Developer] ウィンドウの [File] から [Open] を選択します。 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。 残りの手順は省略します。
	<ol style="list-style-type: none"> [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名を入力します。 [OK] を選択します。VIEW テーブル・エディタが起動し、図 3-7 に示すような画面が表示されます。

図 3-7 VIEW 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記号 (#) 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。

表 3-4 は、VIEW テーブル・エディタの各列の説明です。

表 3-4 VIEW 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。
Type	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。
Flag	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、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(名前の付いていない)VIEW ファイルを開くと、viewx 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、オーバーラップ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

表 3-5 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 3-5 FML 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。
Type	有効な FML 型。
Flag	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 3-6 VIEW 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。
Type	有効な BEA Tuxedo の型。
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。
Flag	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 の 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 言語リファレンス』
- 『BEA Tuxedo COBOL リファレンス』
- 『ファイル形式、データ記述方法、MIB、およびシステム・プロセスのリファレンス』

