



BEATuxedo®

BEA Tuxedo Workstation コンポー ネント

Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

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

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

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

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic 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.

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. Workstation の概要

Workstation コンポーネントとは.....	1-1
ワークステーション・クライアントの制限事項.....	1-3
Workstation の管理.....	1-3

2. Workstation コンポーネントの使用

クライアント・プログラムを作成する.....	2-1
ワークステーション・クライアントの相互運用性に関する制約.....	2-2
クライアント・プログラムを作成する.....	2-2
BEA Tuxedo システムに組み込まれているクライアントを使用する.....	2-4
セキュリティ付きアプリケーションで wud を使用する.....	2-4
ワークステーションで BEA Tuxedo システムのクライアントを実行する..	2-5
ワークステーション・クライアントのディレクトリ構造.....	2-5
環境変数を設定する.....	2-6
環境ファイルを作成する.....	2-10
tuxreadenv を使用する.....	2-12
マルチスレッド化またはマルチコンテキスト化されたワークステーション・クライアントをアプリケーションに参加させる.....	2-14

3. Windows システムでの Workstation の使用

Windows システムで Workstation を使用する利点.....	3-1
ソフトウェア要件.....	3-2
クライアント・プログラムを作成する.....	3-3
ワークステーション・クライアントの相互運用性に関する制約.....	3-3

クライアント・プログラムを作成する	3-4
実行時	3-5
制限事項	3-5
マルチスレッド化またはマルチコンテキスト化されたワークステーション・ クライアントをアプリケーションに参加させる	3-6

4. ワークステーションでの bankapp の実行

ワークステーション・アプリケーションの特徴	4-2
ワークステーション・クライアントで bankapp を実行する	4-2
ネイティブ・サイトでの変更	4-3
コンフィギュレーション・ファイルを編集する	4-4
コンフィギュレーション・ファイルをロードおよび起動する	4-4
bankapp 変数を設定する	4-5
bankapp クライアントを作成する	4-6
bankapp クライアントを実行する	4-6

このマニュアルについて

このマニュアルでは、BEA Tuxedo ATMI Workstation コンポーネントの概要、Workstation コンポーネントを Windows および UNIX システムで使用する
方法、およびサンプル・アプリケーションの `bankapp` をワークステーション
で実行する方法について説明します。

このマニュアルでは、以下の内容について説明します。

- **第 1 章「Workstation の概要」**では、BEA Tuxedo ATMI Workstation コンポーネントの概要を説明します。
- **第 2 章「Workstation コンポーネントの使用」**では、BEA Tuxedo システムに組み込まれているクライアントを使用して、Windows および UNIX システム用のワークステーション・クライアント・プログラムを作成する方法と、BEA Tuxedo システムのクライアントをワークステーションで実行する方法を説明します。
- **第 3 章「Windows システムでの Workstation の使用」**では、Windows システムで BEA Tuxedo ATMI Workstation コンポーネントを使用する際の追加情報を示します。
- **第 4 章「ワークステーションでの `bankapp` の実行」**では、Windows または UNIX 上のワークステーションで BEA Tuxedo システムのサンプル・アプリケーションである `bankapp` を実行する手順を説明します。

対象読者

このマニュアルは、主に、アプリケーションで Workstation コンポーネントを使用するアプリケーション開発者を対象にしています。Workstation コンポーネントの概要、Workstation コンポーネントを使用してクライアント・プログラムを記述し作成する方法、およびワークステーションで bankapp アプリケーションを実行する方法を取り上げています。

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

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

- 『BEA Tuxedo システム入門』
- 『Windows での BEA Tuxedo の使用』

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 Workstation の概要

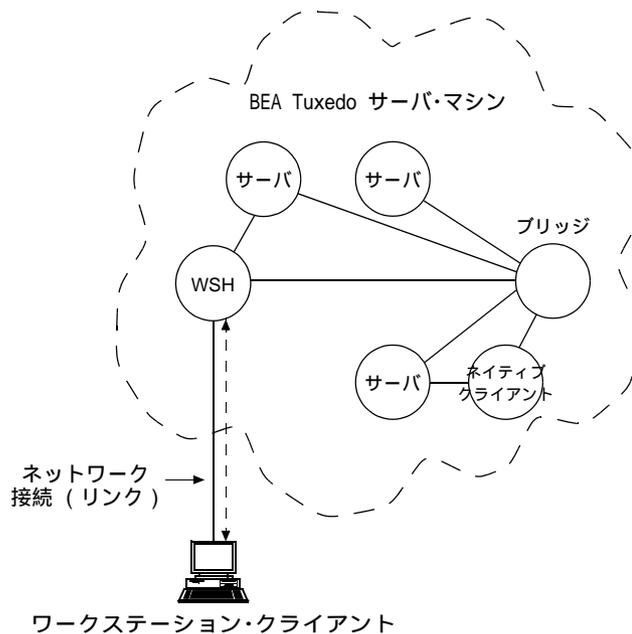
以下の節では、BEA Tuxedo ATMI Workstation コンポーネントの概要を説明します。

- [Workstation コンポーネントとは](#)
- [Workstation の管理](#)

Workstation コンポーネントとは

BEA Tuxedo システムの Workstation コンポーネントを使用すると、サーバ側の機能がフル・インストールされていないマシン、つまり管理サーバもアプリケーション・サーバもサポートしないマシンに、アプリケーション・クライアントを収容できます。次の図のように、ワークステーション・クライアント (Workstation コンポーネント上で動作するアプリケーション・クライアント) とサーバ・アプリケーションの間の通信はすべてネットワーク経由で行われます。

図 1-1 BEA Tuxedo ATMI Workstation コンポーネントの処理



注記 ワークステーション・クライアントは、ワークステーション・ハンドラ (WSH) のプロセスを介してサーバ・アプリケーションと通信します。

ワークステーション・クライアントは、Windows または UNIX システムのどちらで動作していても、ほとんどのクライアント ATMI にアクセスできますが、ネイティブ・クライアント (サーバ・プログラムを実行しているマシンと同じマシン上で動作するクライアント) が利用できるすべてのアクセス権を持っているわけではありません。ただし、どちらのクライアントも、次の処理は実行できます。

- メッセージを送受信する
- トランザクションを開始、終了、またはコミットする
- 任意通知型メッセージを送受信する
- アプリケーションのセキュリティ条件を満たす

- `tmadmin()` コマンドを使用してリモート・クライアントに関する情報を送受信する。詳細については、『BEA Tuxedo コマンド・リファレンス』の `tmadmin(1)` を参照してください。

ワークステーション・クライアントの制限事項

ワークステーション・クライアントは、ネイティブ・クライアントが利用できるすべての機能にアクセスできるわけではありません。たとえば、ワークステーション・クライアントは、ネイティブ・クライアントとは異なり、`tpsyzadm` としてアプリケーションに参加できません。つまり、サービス呼び出しを発行するイベントをサブスクライブできません。

関連項目

- 『BEA Tuxedo システム入門』の 3-10 ページの「[BEA Tuxedo ワークステーション・サーバ](#)」
- 『BEA Tuxedo アプリケーションの設定』の「[ワークステーション・クライアントについて](#)」

Workstation の管理

ワークステーション・クライアントを BEA Tuxedo アプリケーションに統合するには、アプリケーションのコンフィギュレーション・ファイルで、そのクライアントに必要なパラメータを定義する必要があります。詳細については、『BEA Tuxedo アプリケーションの設定』の「[ワークステーション・クライアントの設定](#)」を参照してください。

関連項目

- 『BEA Tuxedo コマンド・リファレンス』の [buildwsh\(1\)](#)
- 『BEA Tuxedo のファイル形式とデータ記述方法』の [WSL\(5\)](#)

2 Workstation コンポーネントの使用

以下の節では、Windows および UNIX システムにおける BEA Tuxedo ATMI Workstation コンポーネントの使用について説明します。

- クライアント・プログラムを作成する
- BEA Tuxedo システムに組み込まれているクライアントを使用する
- ワークステーションで BEA Tuxedo システムのクライアントを実行する
- マルチスレッド化またはマルチコンテキスト化されたワークステーション・クライアントをアプリケーションに参加させる

クライアント・プログラムを作成する

ワークステーション用のクライアント・プログラムは、BEA Tuxedo システムの管理ドメイン内のクライアント・プログラム（ネイティブ・クライアント）と同じ方法で開発できます。若干の例外を除いて、すべての ATMI および FML 関数をネイティブ・クライアントおよびワークステーション・クライアントで利用できます。

注記 ネイティブ・クライアントでは利用できるもののワークステーション・クライアントでは利用できない ATMI 関数の例として `tpadmcall()` があります。

ワークステーション・クライアントの相互運用性に関する制約

BEA Tuxedo リリース 7.1 のワークステーション・クライアントは、次のいずれかの場合に、リリース 7.1 より前の BEA Tuxedo システム・アプリケーションと相互運用します。

- マルチスレッド化またはマルチコンテキスト化されたクライアントではない。
- マルチコンテキスト化されたクライアントである。
- クライアントがマルチスレッド化されており、各スレッドが異なるコンテキストにある。

BEA Tuxedo リリース 7.1 以降のワークステーション・クライアントで、1 つのコンテキストに複数のスレッドがある場合は、リリース 7.1 より前の BEA Tuxedo システムとは相互運用しません。

クライアント・プログラムを作成する

ワークステーション・クライアントのプログラムをコンパイルし、リンクするには、`buildclient(1)` コマンドを使用します。ネイティブ・ノード (BEA Tuxedo システムがフル・インストールされているノード) でワークステーション・クライアントを作成する場合は、`-w` オプションを指定し、クライアントの作成にはワークステーション・ライブラリが必要であることを示します。指定しない場合、デフォルトでは、ネイティブ・ノードにあるネイティブ・ライブラリとワークステーション・ライブラリのうち、ネイティブ・ライブラリが使用されます。つまり、`-w` オプションを指定すると、ワークステーション・クライアントでは正しいライブラリが使用されることが保証されます。ワークステーションには、ワークステーション・ライブラリしかないため、`-w` オプションを指定する必要はありません。

以下のリストは、ネイティブ・ノードでの `buildclient(1)` コマンド行の例です。

コード リスト 2-1 buildclient コマンド行

```
TUXDIR=/var/opt/tuxedo CC=ncc; export TUXDIR CC
buildclient -w -o wsclt -f wsclt.c -f "userlib1.a userlib2.a"
```

-o オプションには、出力ファイルの名前を指定します。入力ファイルを指定するには、`-f firstfiles` オプションを使用し、システム・ライブラリより先にリンクされることを示します。上の例に示すように、`TUXDIR` 環境変数を定義し、`buildclient` コマンドの実行時にシステム・ライブラリが検索されるようにしなければなりません。`CC` は、デフォルトで `cc` になりますが、上の例のように、別のコンパイラを設定することもできます。

関連項目

- 『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 4-1 ページの「クライアントのコーディング」および『COBOL 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』の 12-1 ページの「Workstation コンポーネントに対する COBOL 言語のバインディング」
- 『BEA Tuxedo CORBA アプリケーションのセキュリティ機能』の 3-5 ページの「ATMI アプリケーションにクライアント・プログラムを参加させるためのセキュリティ・コードの記述方法」
- 『BEA Tuxedo コマンド・リファレンス』の `buildclient(1)`

BEA Tuxedo システムに組み込まれているクライアントを使用する

wud および wud32 は、BEA Tuxedo システムに組み込まれているワークステーション用のドライバ・プログラムです。これらのドライバ・プログラムは、ワークステーション・ライブラリを使用して作成する、標準的な BEA Tuxedo のクライアント・プログラムである ud および ud32 に基づいています。

BEA Tuxedo システム・サーバに FML バッファを送信するには、wud(1) を使用します。FBFR32 型の FML32 フィールド化バッファの場合は、wud32 を使用します。

セキュリティ付きアプリケーションで wud を使用する

wud をセキュリティ付きアプリケーションで実行する場合は、アプリケーション・パスワードを指定しないとアプリケーションにアクセスできません。端末から標準入力を行う場合に wud を実行すると、アプリケーション・パスワードの入力を求められます。wud では通常、スクリプトからクライアント・プログラムを実行しますが、この場合、パスワードは APP_PW 環境変数から取得されます。アプリケーション・パスワードが必要な場合に、この環境変数が指定されていないと、wud は異常終了します。

APP_PW 環境変数と、コンフィギュレーション・ファイルの SECURITY パラメータ (APP_PW を指定するとセキュリティ機能が有効になる) を混同しないでください。

関連項目

- 『BEA Tuxedo コマンド・リファレンス』の [ud](#)、[wud\(1\)](#)

ワークステーションで BEA Tuxedo システムのクライアントを実行する

開発およびテストが終了したクライアント・プログラムは、ワークステーションに移動でき、ユーザ側で利用可能になります。

ワークステーション・クライアントのディレクトリ構造

次の表は、BEA Tuxedo システムの Workstation コンポーネントをインストールした場合のワークステーション・クライアントのディレクトリ構造を示しています。

Windows ディレクトリ	UNIX ディレクトリ	説明
%APPDIR%	\$APPDIR	クライアントの実行可能ファイルが格納されています。これらの実行可能ファイルは通常、アプリケーション ディレクトリに格納されます。
%TUXDIR%\bin	\$TUXDIR/bin	BEA Tuxedo システムのコマンド群およびシステム・クライアント (wud など) が格納されています
%TUXDIR%\cobinclude	\$TUXDIR/cobinclude	COBOL プログラムで使用する copylib エントリが格納されています。
%TUXDIR%\include	\$TUXDIR/include	BEA Tuxedo システムのヘッダ・ファイル (atmi.h など) が格納されています。
%TUXDIR%\lib	\$TUXDIR/lib	ランタイム・ライブラリが格納されています。
%TUXDIR%\locale\C	\$TUXDIR/locale/C	デフォルト・ロケールのメッセージ・カタログ (米国の英語) が格納されています

2 Workstation コンポーネントの使用

Windows ディレクトリ	UNIX ディレクトリ	説明
%TUXDIR%\samples	\$TUXDIR/samples	サンプル・アプリケーションのサブディレクトリがいくつか格納されています。

環境変数を設定する

ワークステーション・クライアントでは、いくつかの環境変数を使用します。次の表は、ワークステーション・クライアントがアプリケーションに参加しようとするときに、`tpinit(3c)` または `TPINITIALIZE(3cbl)` によってチェックされる環境変数を示しています。これらの環境変数の設定方法については、『BEA Tuxedo アプリケーションの設定』の [11-1 ページ](#) の「[ワークステーション・クライアントの定義](#)」を参照してください。

環境変数	説明
TPMBENC	<p>BEA Tuxedo 8.1 以降を実行するワークステーション・マシンが、割り当てられた MBSTRING 型付きバッファに格納するコードセットの符号化名を指定します。ワークステーション・クライアントが MBSTRING バッファを割り当てて送信すると、TPMBENC で定義されたコードセットの符号化名は属性として自動的にバッファに追加され、バッファ・データと共に送信先のサーバ・プロセスに送信されます。</p> <p>ワークステーション・マシンが MBSTRING バッファを受信して、別の環境変数 TPMBACONV が設定されていると見なすと、TPMBENC で定義されたコードセットの符号化名は、受信されたバッファ内のコードセット符号化名と自動的に比較されます。名前が同じではない場合、MBSTRING バッファのデータは、ワークステーション・クライアントに配信される前に、TPMBENC で定義された符号化に自動的に変換されます。</p> <p>TPMBENC にはデフォルト値はありません。MBSTRING 型付きバッファを使用するワークステーション・クライアントの場合、ワークステーション・マシン上で TPMBENC を定義する必要があります。</p> <p>注記 TPMBENC は FML32 型付きバッファの FLD_MBSTRING フィールドと同じように使用されます。</p>
TPMBACONV	<p>BEA Tuxedo 8.1 以降を実行するワークステーション・マシンが、受信した MBSTRING バッファのデータを、TPMBENC で定義された符号化に自動的に変換するかどうかを指定します。デフォルトでは自動変換は無効になっています。受信した MBSTRING バッファのデータは、符号化の変換はされないままでワークステーション・クライアントに配信されます。TPMBACONV を Y (yes) などの NULL 以外の値に設定すると、自動変換が有効になります。</p> <p>注記 TPMBACONV は FML32 型付きバッファの FLD_MBSTRING フィールドと同じように使用されます。</p>

環境変数	説明
URLENTITYCACHING	<p>BEA Tuxedo 8.1 以降を実行するワークステーション・マシンが文書型定義 (DTD)、XML スキーマ、およびエンティティ・ファイルをキャッシュするかどうかを指定します。特に、ワークステーション・クライアント上で動作する Apache Xerces-C++ パーサが、検証が必要な場合に DTD と XML スキーマをキャッシュするの、または DTD で指定された外部エンティティ・ファイルをキャッシュするのを指定します。デフォルトでは、キャッシングは有効になっています (Y)。</p> <p>URLENTITYCACHING を N (no) に設定すると、キャッシングは無効になります。</p>
URLENTITYCACHEDIR	<p>URLENTITYCACHING=Y (yes) の場合、または設定されていない場合にのみ適用されます。詳細については、この表の URLENTITYCACHING の説明を参照してください。</p> <p>BEA Tuxedo 8.1 以降を実行するワークステーション・マシンが DTD、スキーマ、およびエンティティ・ファイルをキャッシュするディレクトリを指定します。特に、ワークステーション・クライアント上で動作する Apache Xerces-C++ パーサが、DTD、XML スキーマ、およびエンティティ・ファイルをキャッシュする場所を指定します。URLENTITYCACHEDIR 変数には、キャッシュされるファイルの絶対パス名を指定します。URLENTITYCACHEDIR を指定しない場合、デフォルトのディレクトリは URLEntityCachedir になります。このディレクトリはワークステーション・クライアント・プロセスの現在の作業ディレクトリ内に作成され、適切な書き込み権が設定されています。</p>
WSALLOWPRE71	<p>BEA Tuxedo リリース 7.1 以降を実行しているワークステーション・マシンと、リリース 7.1 より前の BEA Tuxedo アプリケーションを相互運用できるかどうかを指定します。変数を Y に設定すると (WSALLOWPRE71=Y)、相互運用が可能になります。</p>
WSBUFFERS	<p>アプリケーションごとのパケット数。</p>

環境変数	説明
WSDEVICE	ネットワークのアクセス時に使用するデバイス。この変数は、BEA Tuxedo システムが TLI ネットワーク・インターフェイスを使用している場合のみ必要です。
WSENVFILE	クライアントの環境で使用される環境変数が定義されたファイルの名前。
WSFADDR	ワークステーション・クライアントがワークステーション・リスナまたはワークステーション・ハンドラに接続するのに使用するネットワーク・アドレス。この変数は、WSFRANGE 変数とともに、ワークステーション・クライアントがアウトバウンド接続を行う前にバインドしようとする TCP/IP ポートの範囲を決定します。このアドレスには、TCP/IP アドレスを指定する必要があります。
WSFRANGE	ワークステーション・クライアントのプロセスが、アウトバウンド接続を確立する前にバインドする TCP/IP ポートの範囲。範囲のベースとなるアドレスは、WSFADDR パラメータで指定します。デフォルト値は 1 です。
WSNADDR	ワークステーション・リスナ (WSL) ・プロセスのネットワーク・アドレス。クライアントは、これを使用してアプリケーションにアクセスします。ワークステーション・リスナを呼び出すには、アプリケーションのコンフィギュレーション・ファイルの値を使用してください。「0x」で始まる値は 16 進値を表す文字列と見なされ、それ以外の値は、ASCII 文字列と見なされません。

環境変数	説明
WSRPLYMAX	ATMI 関数が、アプリケーションからの応答をディスクにダンプする前にバッファに格納するため使用するコア・メモリの最大容量。 <code>tpinit(3c)</code> および <code>TPINITIALIZE(3cbl)</code> で使用されます。 <code>tpgetrply(3c)</code> および <code>TPGETRPLY(3cbl)</code> で取得した応答および任意通知型メッセージは、この領域に格納されます。この領域がメッセージでいっぱいの場合、オーバーフローしたメッセージはディスク・ファイルに書き込まれます。システムで指定されているデフォルトの最大値は、256,000 バイトです。 <code>WSRPLYMAX</code> を使用して低い制限値を設定するかどうかは、マシンの空きメモリに応じて異なります。応答をディスクに書き込むと、パフォーマンスが大幅に低下します。
WSTYPE	ワークステーションの種類。ワークステーション・クライアントによって <code>tpinit(3c)</code> および <code>TPINITIALIZE(3cbl)</code> が呼び出され、ネイティブ・サイトとの間で符号化または復号化を行うかどうかを判断するときに使用されます。 <code>WSTYPE</code> を指定しない場合は、ネイティブ・サイトで指定していなくても符号化が行われます。ワークステーションのサイトとネイティブ・サイトには、明示的に同じ <code>WSTYPE</code> を指定し、符号化または復号化の処理が実行されないようにします。

UNIX ワークステーション上のワークステーション・クライアントでは、使用する BEA Tuxedo システムの機能に応じて、上記以外の環境変数が必要になる場合があります。状況に応じて必要となる変数については、『BEA Tuxedo のファイル形式とデータ記述方法』の `compilation(5)` リファレンス・ページを参照してください。

環境ファイルを作成する

作成した環境ファイルは、 `tpinit(3c)` または `TPINITIALIZE(3cbl)` の呼び出し時に読み込まれます。次のリストは、2つの異なるアプリケーションで使用されるサンプル・ファイルです。

コードリスト 2-2 環境ファイル

```
TUXDIR=/opt/tuxedo
[application1]
; これはコメントです
/* これはコメントです */
# これはコメントです
// これはコメントです
set FIELDTBLS=app1_flds
set FLDTBLDIR=/opt/app1/udataobj
[application2]
FIELDTBLS=app2_flds
FLDTBLDIR=/opt/app2/udataobj
```

このファイルの形式は次のとおりです。

- 各行の先頭のスペースまたはタブ文字は無視され、以下に述べる事項でも無視されます。
- 環境に適用する変数を含む行は、次の形式になります。

```
variable=value
または
set variable=value
```

ここで、*variable* は、先頭がアルファベット文字またはアンダーライン文字で、全体が英数字またはアンダーライン文字のみからなる文字列です。また、*value* には改行文字を除くすべての文字を使用できます。

- *value* では、`${env}` という形式の文字列は、既に環境内にある変数を使用して展開されます。前方参照はサポートされていません。値が設定されていない場合、変数は空の文字列に置換されます。バックスラッシュ (`\`) を使用して、ドル記号およびそれ自体をエスケープすることができます。その他すべてのシェルのクォートおよびエスケープのメカニズムは無視され、展開された *value* がそのまま環境に入れられます。
- スラッシュ (`/`)、シャープ記号 (`#`)、または感嘆符 (`!`) で始まる行は、コメント行と見なされ、無視されます。これらのコメント文字、左角かっこ、アルファベット文字またはアンダーライン文字以外の文字で始まる行は、将来の用途のため予約されています。使用法は未定義です。
- ファイルは、次の形式の行によってセクションごとに区切られます。

[*label*]

label はセクションの名前であり、上記の *variable* の規則が適用されず、31 文字を超える部分は切り捨てられます。

- ファイルの先頭から、最初のラベルの間にある変数行は、すべてのアプリケーションの環境に組み込まれます。つまり、この部分はグローバルなセクションです。[] というラベルもグローバルなセクションを示します。その他の変数は、ラベルがアプリケーション用に指定したラベルと一致する場合にのみ、環境に組み込まれます。

tuxreadenv を使用する

`tuxreadenv(3c)` 関数を呼び出すと、環境ファイルが読み込まれ、プラットフォームに関係なく、プロセス全体の環境に環境変数が追加されます。変数を利用するには `tuxgetenv(3c)` を使用し、変数をリセットするには、`tuxputenv(3c)` を使用します。

```
void tuxreadenv(char *file, char *label)
```

file が NULL の場合、デフォルトのファイル名が使用されます。各プラットフォームのデフォルトのファイル名は次のとおりです。

- %TUXDIR%\TUXEDO.ENV (Windows)
- \$TUXDIR/TUXEDO.env (UNIX)

label の値が NULL の場合は、グローバル・セクションの変数だけが環境に組み込まれます。*label* が他の値の場合、グローバル・セクションの変数と、*label* に一致するセクション内の変数が環境に入れられます。

次の場合は、`userlog` にエラー・メッセージが記録されます。

- メモリ障害が発生した場合
- NULL 以外のファイル名が存在しない場合
- NULL 以外のラベルが存在しない場合

`tpinit(3c)` を明示的に呼び出すか、または別の ATMI 関数を呼び出して暗黙的に呼び出すたびに、ワークステーション・クライアントには `tuxreadenv(3c)` が自動的に呼び出されます。環境に設定されている `WSENVFILE` は、環境ファイルを示します。設定されていない場合は、`tuxreadenv()` にファイル名として `NULL` が渡され、デフォルトのファイルが使用されます。環境に設定されている `WSAPP` は、環境ファイルのセクションのラベルとして使用されます。設定されていない場合は、`tuxreadenv()` にラベル名として `NULL` が渡されます。アプリケーション・クライアントは、`tuxreadenv()` を明示的に呼び出すこともできます。

インプリメントされた環境は、プラットフォームごとに異なる方法で使用できます。環境に対する共通のインターフェイスは、既存の `tuxgetenv(3c)` 関数および `tuxputenv(3c)` 関数により提供されます。これらの関数により、次の変数にアクセスできます。

- 指定した `WSAPP` ラベル (または指定されていない場合はデフォルト値) が設定された、指定した `WSENVFILE` ファイルのすべての変数
- オペレーティング・システム環境の環境変数

関連項目

- 『BEA Tuxedo C リファレンス』の `tpinit(3c)`
- 『BEA Tuxedo C リファレンス』の `tuxreadenv(3c)`

マルチスレッド化またはマルチコンテキスト化されたワークステーション・クライアントをアプリケーションに参加させる

BEA Tuxedo アプリケーションに参加するには、マルチスレッドのワークステーション・クライアントは、たとえシングルコンテキスト・モードで動作している場合でも、必ず `TPMULTICONTEXTS` フラグを設定して `tpinit()` 関数を呼び出さなければなりません。

関連項目

- 『BEA Tuxedo C リファレンス』の [tpinit\(3c\)](#)

3 Windows システム での Workstation の 使用

以下の節では、Windows 98、Windows XP、または Windows 2000 システムにおける BEA Tuxedo ATMI Workstation コンポーネントの使用に関する追加情報を示します。

- Windows システムで Workstation を使用する利点
- ソフトウェア要件
- クライアント・プログラムを作成する
- マルチスレッド化またはマルチコンテキスト化されたワークステーション・クライアントをアプリケーションに参加させる

Windows システムで Workstation を使用する利点

Windows 用にインスタンス化されたワークステーション・クライアントを使用すると、アプリケーション開発者は、次のような利点を活用できます。

- 実行可能テキストをアプリケーション間で共有できるため、メモリを節約できます。
- アプリケーション・プログラムの実行可能ファイルに再リンクしたり、このファイルを変更せずに、BEA Tuxedo の Workstation をアップグレードできます。
- ダイナミック・リンクにより、インタープリタ形式でグラフィカルなアプリケーション生成ツール (Visual Basic、ObjectVision、および SQLWindows) を使用して、BEA Tuxedo システムのサービスを呼び出すことができます。

ソフトウェア要件

Windows システムで Workstation コンポーネントを実行するためのソフトウェア要件は、以下のとおりです。

- 使用できるオペレーティング・システムは、Windows 98、Windows XP、Windows 2000 です。
- ネイティブの TCP/IP スタックが使用されます。
- TCP/IP を使用する場合は、Windows ソケット準拠の任意の TCP/IP スタックを使用できます。
- サーバ・マシンには、BEA Tuxedo システムおよびネイティブ・サイドの BEA Tuxedo Workstation がインストールされていなければなりません。

クライアント・プログラムを作成する

Windows ワークステーション用のクライアント・プログラムは、BEA Tuxedo システムの管理ドメイン内のネイティブなクライアント・プログラムと同じ方法で開発できます。ATMI 関数はすべて使用できます。

ワークステーション・クライアントの相互運用性に関する制約

BEA Tuxedo リリース 7.1 以降のワークステーション・クライアントは、次のいずれかの場合に、リリース 7.1 より前の BEA Tuxedo システム・アプリケーションと相互運用します。

- マルチスレッド化またはマルチコンテキスト化されたクライアントではない。
- マルチコンテキスト化されたクライアントである。
- クライアントがマルチスレッド化されており、各スレッドが異なるコンテキストにある。

BEA Tuxedo リリース 7.1 以降のワークステーション・クライアントで、1つのコンテキストに複数のスレッドがある場合は、リリース 7.1 より前の BEA Tuxedo システムとは相互運用しません。

クライアント・プログラムを作成する

C を使用して記述されたクライアント・プログラムをコンパイルする場合は、Microsoft の C インポート・ライブラリを読み込めるコンパイラであれば、どんなコンパイラでも使用できます。ATMI 関数を呼び出す COBOL ソース・プログラムをコンパイルする場合は、COBOL コンパイラの LITLINK オプションを使用します。詳細については、『COBOL を使用した BEA Tuxedo アプリケーションのプログラミング』の 12-1 ページの「Workstation コンポーネントに対する COBOL 言語のバインディング」を参照してください。

作成したクライアント・プログラムにリンクするには、`buildclient(1)` で `-w` フラグを指定します。

`buildclient(1)` ユーティリティを使わずに BEA Tuxedo クライアントを作成することもできます。Microsoft Visual C++ プロジェクトを使用している場合は、次のように設定してください。

- プリプロセッサ・オプションを `-DWIN32` に設定します。
- リンカ・オプションの入力ライブラリに `WTUXWS32.LIB MSVCRT.LIB` を追加します。

また、適切な `INCLUDE`、`LIB`、および `PATH` 検索ディレクトリを設定してください。

実行時

クライアント・プログラムを実行する場合は、PATH に %TUXDIR%\bin が含まれていなければなりません。

制限事項

BEA Tuxedo リリース 7.1 より前の BEA Tuxedo ライブラリ (DLL) は、スレッド・セーフではありません。リリース 7.1 以前の DLL を使用して作成したアプリケーションでは、スレッドを使用しないでください。スレッドを使用すると、BEA Tuxedo のすべての呼び出し (ATMI、FML、userlog() など) で、スレッド・アクセスがシリアル化されます。

関連項目

- 『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』の 4-1 ページの「クライアントのコーディング」
- 『COBOL を使用した BEA Tuxedo アプリケーションのプログラミング』の 12-1 ページの「Workstation コンポーネントに対する COBOL 言語のバインディング」
- 『BEA Tuxedo CORBA アプリケーションのセキュリティ機能』の 3-5 ページの「ATMI アプリケーションにクライアント・プログラムを参加させるためのセキュリティ・コードの記述方法」
- 『BEA Tuxedo コマンド・リファレンス』の `buildclient(1)`

マルチスレッド化またはマルチコンテキスト化されたワークステーション・クライアントをアプリケーションに参加させる

BEA Tuxedo アプリケーションに参加するため、マルチスレッド化されたクライアントは、シングル・コンテキスト・モードで実行していても、TPMULTICONTEXTS フラグ・セットを指定した `tpinit(3c)` を常に呼び出す必要があります。

関連項目

- 『BEA Tuxedo C リファレンス』の `tpinit(3c)`

4 ワークステーション での bankapp の実 行

以下の節では、Windows または UNIX のワークステーションで BEA Tuxedo システムのサンプル・アプリケーションである bankapp を実行する手順を説明します。

- [ワークステーション・アプリケーションの特徴](#)
- [ワークステーション・クライアントで bankapp を実行する](#)
- [ネイティブ・サイトでの変更](#)
- [bankapp 変数を設定する](#)
- [bankapp クライアントを作成する](#)
- [bankapp クライアントを実行する](#)

ワークステーション・アプリケーション の特徴

ワークステーション・アプリケーションでは、クライアント・プロセスはネイティブ・サイトから切りはなされて実行されます。リスナ・プロセス (WSL) が既知のネットワーク・アドレスを使用して実行されると、必要に応じて代理のワークステーション・ハンドラ (WSH) が起動します。サーバは、BEA Tuxedo 管理ドメインにある 1 台または複数のマシン上で実行されます。

既存のサーバは、シングル・プロセッサ・モード (SHM) またはマルチプロセッサ・モード (MP) の BEA Tuxedo システムのノードで実行できます。

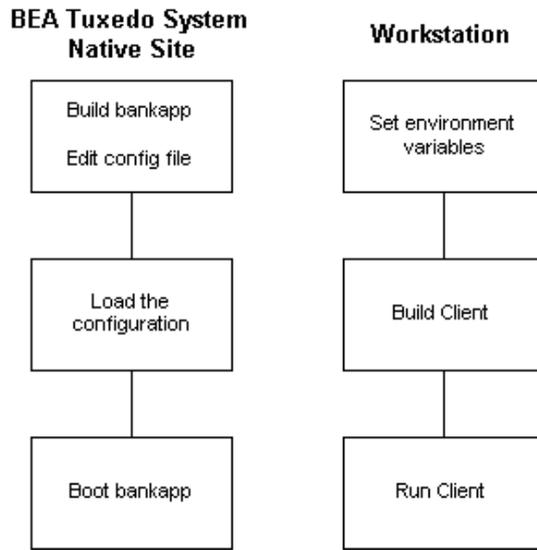
ワークステーションの次のディレクトリには、サンプル・アプリケーションが用意されています。

- %TUXDIR%\samples\atmi\bankapp (Windows)
- \$TUXDIR/samples/atmi/bankapp (UNIX)

ワークステーション・クライアントで bankapp を実行する

次の図は、ワークステーション・クライアントで bankapp を実行する手順を示しています。

図 4-1 bankapp の実行手順



ネイティブ・サイトでの変更

bankapp ソフトウェアは、ネイティブ・サイトにインストールして作成します。操作方法は、『サンプルを使用した BEA Tuxedo アプリケーションの開発方法』の 3-1 ページの「bankapp (複雑な C 言語アプリケーション) のチュートリアル」、および BEA Tuxedo システム・ソフトウェアのインストール先マスタ・マシンにある以下の README ファイルに説明されています。

- %TUXDIR%\samples\atmi\bankapp\README.nt (Windows)
- \$TUXDIR/samples/atmi/bankapp/README (UNIX)

コンフィギュレーション・ファイルを編集する

使用するコンフィギュレーション・ファイル (ubbshh または ubbhp) の GROUPS セクションおよび SERVERS セクションでワークステーション・リスナ (WSL) をサーバとして指定し、MACHINES セクションで MAXWSCLIENTS を指定する必要があります。GROUPS セクションを編集する場合は、DEFAULT 行の先頭に WSGRP 用のエントリを作成するか、または、TMSNAME および TMSCOUNT の仕様を該当するサーバ・グループに移動します。仕様を WSGRP に割り当てないでください。新しい仕様は、次の形式で指定します。

```
*MACHINES
DEFAULT:  MAXWSCLIENTS=50

#
*GROUPS

WSGRP      GRPNO=<next available group #> LMID=SITE1
#
*SERVERS

WSL        SRVGRP=WSGRP          SRVID=1
           CLOPT="-A - -n //machine:port -m 1 -M 5 -x 10"
```

新しいワークステーション・クライアントを設定した場合は、必ず RESOURCES セクションまたは MACHINES セクションの MAXACCESSERS パラメータの値を上げてください。

コンフィギュレーション・ファイルをロードおよび起動する

ワークステーション・クライアントを使用するには、まず、`tmloadcf(1)` を実行してコンフィギュレーション・ファイルをバイナリ形式に変換し、`tmboot(1)` を実行してアプリケーションを起動する必要があります。これらのコマンドは、直ちに実行する必要はありません。bankapp クライアントをワークステーションにインストールし、作成する作業があります。ただし、ワークステーションからアプリケーションに参加する場合、アプリケーションは既に BEA Tuxedo システムのネイティブ・サイトで実行されていなければなりません。ネイティブ・サイトで bankapp をロードし、起動する方法

は、『サンプルを使用した BEA Tuxedo アプリケーションの開発方法』の 3-1 ページの「bankapp (複雑な C 言語アプリケーション) のチュートリアル」に説明されています。

関連項目

- 『BEA Tuxedo コマンド・リファレンス』の [tmloadcf\(1\)](#)
- 『BEA Tuxedo のファイル形式とデータ記述方法』の [WSL\(5\)](#)
- 『サンプルを使用した BEA Tuxedo アプリケーションの開発方法』の 3-1 ページの「bankapp (複雑な C 言語アプリケーション) のチュートリアル」
- 『BEA Tuxedo アプリケーションの設定』の 11-5 ページの「ワークステーション・リスナ (WSL) をサーバとして定義する」

bankapp 変数を設定する

bankapp を実行する環境を設定するには、ワークステーション・クライアントで次の手順を実行します。

1. 設定する環境変数

WSNADDR=<WSL で宣言されたアドレス>
WSTYPE=<ワークステーションの種類>

2. PATH に %TUXDIR%\bin (Windows) または \$TUXDIR/bin (UNIX) を指定します。
3. C コンパイラを実行できるように正しく環境が設定されているかどうかを確認します。

bankapp クライアントを作成する

クライアント・プログラムを作成するには、次のコマンドを入力します。

```
mkfldhdr bankflds  
buildclient -w -o bankclt -f bankclt.c
```

bankapp クライアントを実行する

ワークステーションで bankapp クライアントを実行するには、次の手順に従います。

1. ワークステーション・クライアントの WSNADDR の値が、ネイティブ・サイトのコンフィギュレーション・ファイルの SERVERS セクションにある WSL の CLOPT-n オプションの値と一致することを確認します。
2. bankapp がネイティブ・サイトで起動していない場合は、起動されていることを確認してからワークステーション・クライアントを実行します。
3. bankclt を実行してワークステーション・クライアントを実行します。