



THE ENTERPRISE MIDDLEWARE SOLUTION

BEA TUXEDO

for Windows NT User's Guide

BEA TUXEDO Release 6.5
Document Edition 6.5
February 1999

Copyright

Copyright © 1999 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, ObjectBroker, TOP END, TUXEDO, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Connect, BEA Manager, BEA MessageQ, Jolt and M3 are trademarks of BEA Systems, Inc.

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

BEA TUXEDO for Windows NT User's Guide

Document Edition	Date	Software Release
6.5	February 1999	BEA TUXEDO Release 6.5

Contents

1. Installing and Upgrading the BEA TUXEDO System on Windows NT

Interoperability	1-1
Installing BEA TUXEDO for Windows NT	1-2
Minimum Hardware Requirements	1-2
Optional Software Recommendations	1-2
The Installation Process	1-3
BEA TUXEDO System Libraries	1-4
Logging in as Administrator	1-4
Types of Installation	1-5
Registry Contents	1-5
Using The Security Key to Restrict Access	1-6
How to Install the BEA TUXEDO System for Windows NT.....	1-6
How to Upgrade an Existing BEA TUXEDO System for Windows NT... 1-7	
Prerequisite.....	1-7

2. Differences Between Using the BEA TUXEDO System on UNIX and on Windows NT

Reference Materials.....	2-2
Windows NT Platform Considerations	2-2
Configuration Issues	2-3
Specifying Machine Type and User ID Numbers	2-3
Using Network Drives	2-4
Updating TxRPC	2-4
Using the Bankapp Driver.....	2-5
Starting BEA TUXEDO Applications Automatically.....	2-5

3. Configuring the BEA TUXEDO System for Windows NT

Reference Materials	3-1
Accessing the Control Panel Applet	3-2
Accessing Machines on a Network.....	3-3
Modifying Environment Variables	3-4
Directing BEA TUXEDO Messages to the Windows NT Event Log	3-6
Configuring tlisten Processes to Start Automatically	3-8
Maximizing System Performance	3-9

4. Developing a BEA TUXEDO Application on Windows NT

Introduction	4-1
For More Information	4-2
Development Tools.....	4-2
Using the buildserver and buildclient Commands	4-2
Adding BuildTuxedo to the MSDEV Tools Menu.....	4-4
Creating BEA TUXEDO Project Files	4-6
Setting Up Your Environment	4-6
How to Specify the Build Type, Header File, and Filename.....	4-7
How BuildTuxedo Uses the Header File	4-8
Specifying Services	4-8
How to Add or Edit Names on the Function Names List.....	4-10
How to Specify a Resource Manager	4-10
How to Debug a BEA TUXEDO Server Application	4-12
Developing a BEA TUXEDO Application Without Visual C++	4-13
The New Tuxdev Application	4-13
Installing Tuxdev	4-13
BEA TUXEDO Editors	4-14
Using the FML Table Editor	4-15
Accessing the FML Table Editor	4-15
Description of the FML Table Editor	4-16
Using the VIEW Table Editor	4-17
Accessing the VIEW Table Editor	4-17
Description of the VIEW Table Editor.....	4-18
Working in Multiple Documents Simultaneously	4-20
How the Editors Validate Entries	4-20

The FML Table Editor	4-20
The VIEW Table Editor	4-21



1 Installing and Upgrading the BEA TUXEDO System on Windows NT

This chapter contains installation prerequisites and instructions for installing the BEA TUXEDO system on Windows NT. It also contains information on upgrading an existing BEA TUXEDO system.

Interoperability

BEA TUXEDO Release 6.5 running on a Windows NT platform can interoperate, as either master or non-master, with Release 6.5 running on any other supported platform.

All workstation platforms that are currently supported work with this release.

Installing BEA TUXEDO for Windows NT

Hardware Requirements

- ◆ x86 (INTEL) or an ALPHA AXP processor
- ◆ 32 MB of memory
- ◆ 50 MB of hard disk space

Software Requirements

- ◆ Windows NT Workstation or Server Release 4.0, service pack 3
- ◆ Microsoft© Visual C++ version 5, service pack 3
- ◆ MicroFocus COBOL Version 4.0 (BEA TUXEDO Development Environment only)
- ◆ For the BEA TUXEDO Web GUI: a Java-capable Web browser

The Installation Process

During installation, the following activities occur:

- ◆ Welcoming window and BEA Licensing Agreement
- ◆ User and company registration
- ◆ Checking for earlier versions of the BEA TUXEDO system or reinstallation
- ◆ Checking if the BEA TUXEDO IPC Helper Service is running, which must be stopped before installation can proceed (only if installing the BEA TUXEDO server package)
- ◆ Prompting for a destination location
- ◆ Prompting for an installation package and checking for required disk space
- ◆ Installation of files on the hard drive
- ◆ Updating of the Registry Table
- ◆ Prompting for destination location of the BEA TUXEDO Web GUI software (only if installing the BEA TUXEDO server package)
- ◆ Creation of a program folder for BEA TUXEDO System v6.5
- ◆ Updating of the `autoexec.bat` file, if installing Workstation on Windows95
- ◆ Prompting for the license file from a diskette (distributed with the software), which enables the software
- ◆ Prompting for a `tlisten` password and creation of a password file under `%TUXDIR%\udataobj\tlisten.pw`
- ◆ Checking version of `MSVC/MFC DLLs` and replacing them if necessary
- ◆ Additional Registry Table updates

BEA TUXEDO System Libraries

During the installation of BEA TUXEDO Release 6.5 on Windows NT, the system environment is modified. One of these modifications is the `PATH` environment variable. When an application is loaded, the link loader checks the local directory, followed by the system `PATH`, user `PATH`, and finally the path set in `autoexec.bat` (if applicable). Because different versions of a DLL can exist in a system, it is critical to verify that the correct DLL version is loaded.

You do this by ensuring that the first BEA TUXEDO system-related pathname in your system path is the pathname for the root directory of the BEA TUXEDO software or `%TUXDIR%`. To do this, perform the following steps.

1. Select Start→Control Panel→System→Environment.
2. Scroll through system variables to ensure that the first instance of `TUXDIR` points to BEA TUXEDO Release 6.5.
3. Set Variable and Value as necessary and click OK.
4. Restart the computer.

A further complication is that a DLL may be loaded in memory. As applications are loaded, they first attach to the library in memory, which may cause applications to attempt to load the incorrect DLL version. To prevent this, shut down all BEA TUXEDO processes before installing Release 6.5. (*Microsoft's*© *Resource Kit for Windows NT* provides additional information.)

Logging in as Administrator

To install the BEA TUXEDO system on Windows NT correctly, the person installing the application needs to log in with the user ID of *Administrator*.

If someone logs in from a non-administrator account, the `TUXIPC` service will not have an entry in the registry. The BEA TUXEDO system is still functional, however, because `TUXIPC` runs as a process, but all applications terminate when the user logs off.

Types of Installation

You can install all or part of the BEA TUXEDO software CD-ROM. Specifically, you can install any of the following:

- ◆ BEA TUXEDO Core System
- ◆ BEA TUXEDO System Workstation only
- ◆ BEA TUXEDO System Web GUI

The documentation is installed from a separate documentation CD-ROM.

Registry Contents

As part of the installation process, the BEA TUXEDO IPC Helper Service is installed as an NT Service; general installation information as well as the IPC parameters are installed in the registry. The layout of the registry is as follows:

```
HKEY_LOCAL_MACHINE\Software\BEA Systems\TUXEDO\6.5\
```

- ◆ Developer
- ◆ Environment
- ◆ Security

Product information, such as the major and minor version numbers of the release, user and company names, is stored under the Release 6.5 key.

Values for the BEA TUXEDO IPC Helper Service, as well as the locations of `TUXDIR`, `NLSPATH`, `ULOGDIR`, `ULOGPFX`, and `ULOGOUT`, are stored under the Environment key. You can tune the IPC values based on your application needs. (Refer to the control panel applet described in Chapter 3, “Configuring the BEA TUXEDO System for Windows NT.”)

The Security key has the following requirements:

- ◆ Any user who runs `tlisten` must have read access permissions.
- ◆ The account under which the BEA TUXEDO IPC Helper is running must have read access permissions.

- ◆ The Administrator is recommended to have full control permissions.

Using The Security Key to Restrict Access

Native clients are advised to call `tpinit()` with the `cltname` set to `tpsysadm` in order to have access to BEA TUXEDO functionality. However, because `tpsysadm` is a trusted user this results in a bypass of the `APP_PW/USER_AUTH` authentication.

To prevent this lapse in security the administrator should do the following:

1. In the Run dialog box, enter
`regedt32`
Click OK.
2. Select `HKEY_LOCAL_MACHINE`→`Software`→`BEA Systems`→`TUXEDO`→`6.5`
3. Select `SECURITY`→`Permissions`
4. Revoke control for Everyone and allow access only to users with administrative privileges.

How to Install the BEA TUXEDO System for Windows NT

1. Log in to Windows NT with the user name of Administrator.
2. Insert the BEA TUXEDO CD-ROM in the CD-ROM drive and select `Start`→`Programs`→`Windows NT Explorer`.
3. Select the CD-ROM icon and choose one of the following actions.

To Install	Select
The server version of the BEA TUXEDO system	<code>WINNT</code> → <code>Server</code> → <code>setup.exe</code>
Workstation client software	<code>WINNT</code> → <code>WS</code> → <code>Disk 1</code> → <code>setup.exe</code>

4. Follow the window prompts.

5. After you get a message that the installation was successful, reboot the machine.

Note: You can also select Start→Run or begin installing from a command prompt. Both installations prompt you for the license diskette before installing.

How to Upgrade an Existing BEA TUXEDO System for Windows NT

Prerequisite

If you are upgrading from an existing BEA TUXEDO system, and are using the BEA TUXEDO IPC Helper Service, you must stop the service and then proceed with the installation procedure.

1. Disable the BEA TUXEDO IPC Helper Service by selecting Start→Settings→Control Panel→Services.
2. From the Services dialog box, select TUXEDO IPC Helper and then select Stop.
3. Insert the BEA TUXEDO CD-ROM in the CD-ROM drive and select Start→Programs→Windows NT Explorer.
4. Select the CD-ROM icon and choose one of the following actions.

To Install	Select
The server version of the BEA TUXEDO system	WINNT→Server→setup.exe
Workstation client software	WINNT→WS→Disk1→setup.exe

5. Follow the window prompts.
6. After you get a message that the installation was successful, reboot the machine.

Note: To ensure proper operation, regenerate any applicable TUXCONFIG files.

1 *Installing and Upgrading the BEA TUXEDO System on Windows NT*

2 Differences Between Using the BEA TUXEDO System on UNIX and on Windows NT

This chapter highlights information on the following differences between using the BEA TUXEDO system on UNIX and on Windows NT:

- ◆ Platform considerations
- ◆ Configuration issues
- ◆ Specifying machine type and user ID numbers
- ◆ Using network drives
- ◆ Updating TxRPC
- ◆ Using the Bankapp driver
- ◆ Starting BEA TUXEDO applications automatically when booting up your system

Reference Materials

Refer to the following guides for information on the BEA TUXEDO system:

- ◆ *BEA TUXEDO Product Overview*
- ◆ *BEA TUXEDO Workstation Guide* for changes needed in the `UBBCONFIG` file
- ◆ *BEA TUXEDO Reference Manual* (Section 5)
- ◆ *Administering the BEA TUXEDO System*
- ◆ *BEA TUXEDO Application Development Guide* for an overview of the `bankapp` program
- ◆ *BEA TUXEDO TxRPC Guide* for information on configuration and operation of TxRPC

Windows NT Platform Considerations

- ◆ When you specify pathnames, use a back slash (“\”), not a forward slash (“/”) to delimit file and directory names on an NT node. We recommend a drive letter (“C:”) for all fully qualified paths.
- ◆ The `.exe` suffix for executable files is always implied for the BEA TUXEDO system for Windows NT.
- ◆ Filenames follow Windows NT naming conventions. For instance, files ending in `.dll` identify dynamically linked libraries; files ending in `.lib` identify statically linked and import libraries; files ending in `.cmd` identify command scripts. (All BEA TUXEDO system executables, command scripts, and dynamically linked libraries are located in `%TUXDIR%\bin`. Statically linked libraries are located in `%TUXDIR%\lib`.)

Configuration Issues

- ◆ Two or more versions of BEA TUXEDO software cannot be installed on the same machine.
- ◆ Server names are case sensitive.
- ◆ Windows NT machine names specified in BEA TUXEDO application files should always be upper case.
- ◆ The OPENINFO string in the BEA TUXEDO configuration file must be in the following format:

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

For example,

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

Note: The first separator in the preceding string is a colon; subsequent separators are semicolons.

Specifying Machine Type and User ID Numbers

In the `MACHINES` page of the configuration file, include the following three entries that are different from the BEA TUXEDO system on a UNIX system platform:

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

Whenever you create a configuration file for a mixed NT-UNIX configuration, include these entries in the `MACHINES` page for every Windows NT node in your configuration.

Using Network Drives

For reliability purposes, we recommend that you do not use network drives. If, however, you attempt to start up the BEA TUXEDO system on an NT machine that has the TUXCONFIG file on a network drive, you must set the following permissions correctly:

- ◆ The network drive must be connected as *administrator*.
- ◆ The administrator must use the same password on the local and remote systems.
- ◆ In the `tuxipc` service startup options, `ENTRY` must have *administrator* set for `Log On As This Account`; the password must be the same as the administrator's so that the `tuxipc` service has full administrator access rights.

Updating TxRPC

When the BEA TUXEDO system interoperates with OSF/DCE, you must update the environment file to include the following `PATH` variable for the application, allowing BEA TUXEDO programs to find the OSF/DCE DLLs whenever needed. Review the `bldc_dce` and `blds_dce` reference pages for information pertaining to using these commands on the Windows NT platform:

- ◆ The `bldc_dce` command is used to build a BEA TUXEDO system client that can be called via OSF/DCE RPC.
- ◆ The `blds_dce` command is used to build a BEA TUXEDO system server that calls OSF/DCE.

Using the Bankapp Driver

The `bankapp` program is a small example application bundled with the BEA TUXEDO system for Windows NT. Besides demonstrating the operation of the BEA TUXEDO system and providing an example of BEA TUXEDO application code, the `%APPDIR%\UBB` file generated by the `bankapp` driver (`driver.exe` is located in `%TUXDIR%\APPS\bankapp\NT\driver`) and can act as a template for configurations for any new applications.

Starting BEA TUXEDO Applications Automatically

When the BEA TUXEDO system is installed on Windows NT as a server, it may be useful to configure the machine to start a BEA TUXEDO application automatically when booting up your system. *Microsoft's© Resource Kit for Windows NT* contains a utility program named `srvany.exe` to do this. Refer to `srvany.wri` and `rkttools.hlp` for instructions on this procedure.

To ensure proper operation of BEA TUXEDO programs that start automatically when booting up, you must set the BEA TUXEDO environment variables `%TUXDIR%` and `%NLSPATH%`. Set these variables using the conventional Windows NT method or by using the BEA TUXEDO control panel. (See “Modifying Environment Variables” in Chapter 3, “Configuring the BEA TUXEDO System for Windows NT.”)

2 *Differences Between Using the BEA TUXEDO System on UNIX and on Windows NT*

3 Configuring the BEA TUXEDO System for Windows NT

In addition to the Web GUI, the BEA TUXEDO system for Windows NT provides a control panel applet that you can use to configure the BEA TUXEDO system for Windows NT. This chapter describes how to use the applet to configure the following items:

- ◆ Access machines on a network by setting the `MACHINES` page
- ◆ Modify environment variables on the `ENVIRONMENT` page
- ◆ Direct BEA TUXEDO system messages to the Windows NT Event Log by setting the `LOGGING` page
- ◆ Configure one or more `tlisten` processes to start automatically by setting the `LISTENER` page
- ◆ Maximize system performance by tuning the `IPC Resources` page setting

Reference Materials

The Web GUI offers extensive online help. Through the GUI help, you can find details about performing all administrative tasks that the GUI helps you perform, plus reference information for all configuration tool folders.

Accessing the Control Panel Applet

To open the control panel applet, select Start→Control Panel→BEA TUXEDO.

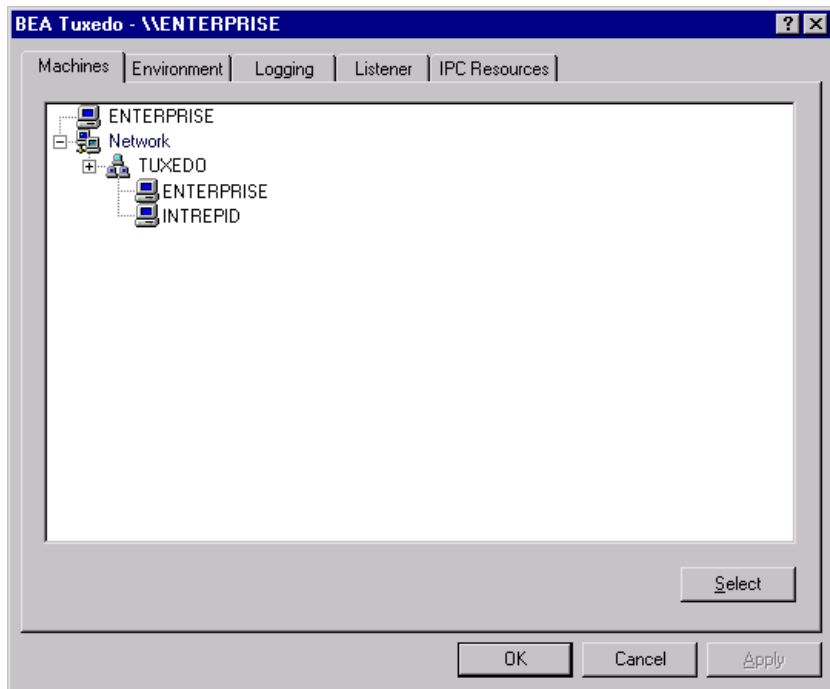
Figure 3-1 BEA TUXEDO System for Windows NT Control Panel



Accessing Machines on a Network

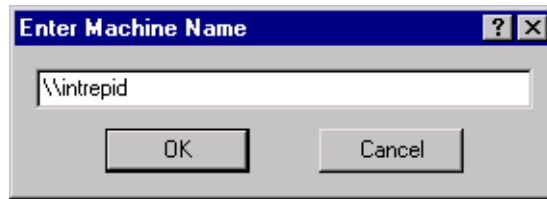
The MACHINES page enables the BEA TUXEDO administrator to access any machine on the Microsoft Windows Network running Windows NT, where the administrator has logon privileges. The administrator can then set environment variables remotely, determine the location of BEA TUXEDO event logging, add/start/remove `tlisten` services, and tune IPC resources. The administrator can access a remote machine by locating the machine on a network tree.

Figure 3-2 BEA TUXEDO System for Windows NT Machines Page



If you know a machine's name, but not its work group, click Select. Then type the name of the remote machine on the Enter Machine Name window and click OK.

Figure 3-3 BEA TUXEDO System for Windows NT Enter Machine Name Box



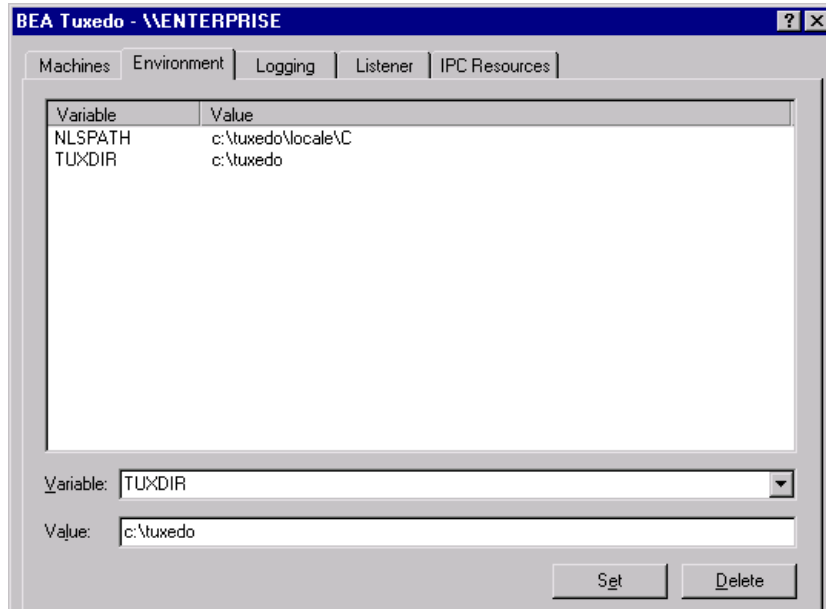
All subsequent actions on other folders in the control panel applet take place on the selected machine.

Modifying Environment Variables

Modifying BEA TUXEDO environment variables is almost identical to modifying Windows NT environment variables. The Variable field contains a list of the most commonly used BEA TUXEDO environment variables.

To add or edit a variable, select the variable, enter its value in its Value field, and click Set. To delete a variable, click the variable you want to delete and click Delete. Clicking OK or Apply saves any changes made to the BEA TUXEDO environment.

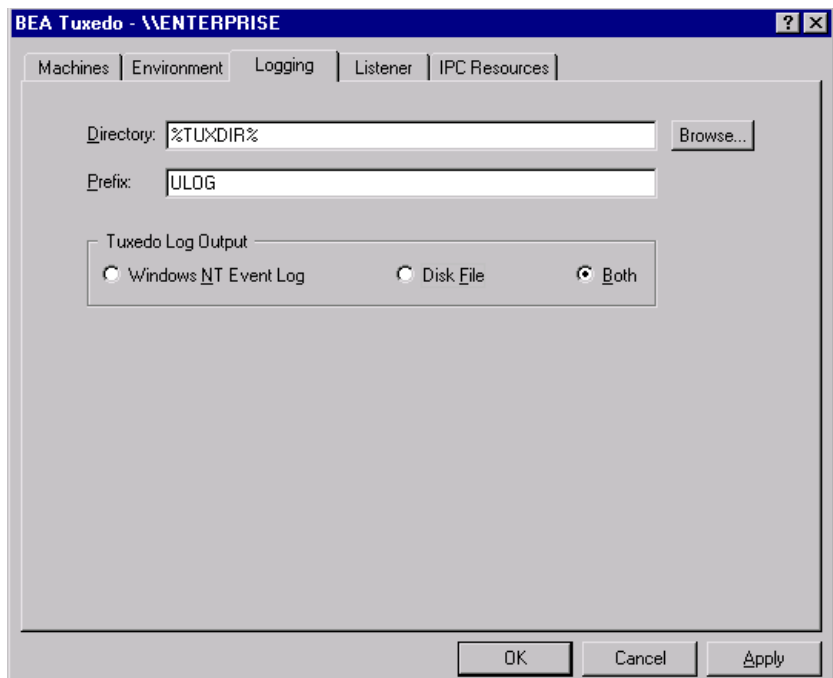
Figure 3-4 BEA TUXEDO System for Windows NT Environment Page



Directing BEA TUXEDO Messages to the Windows NT Event Log

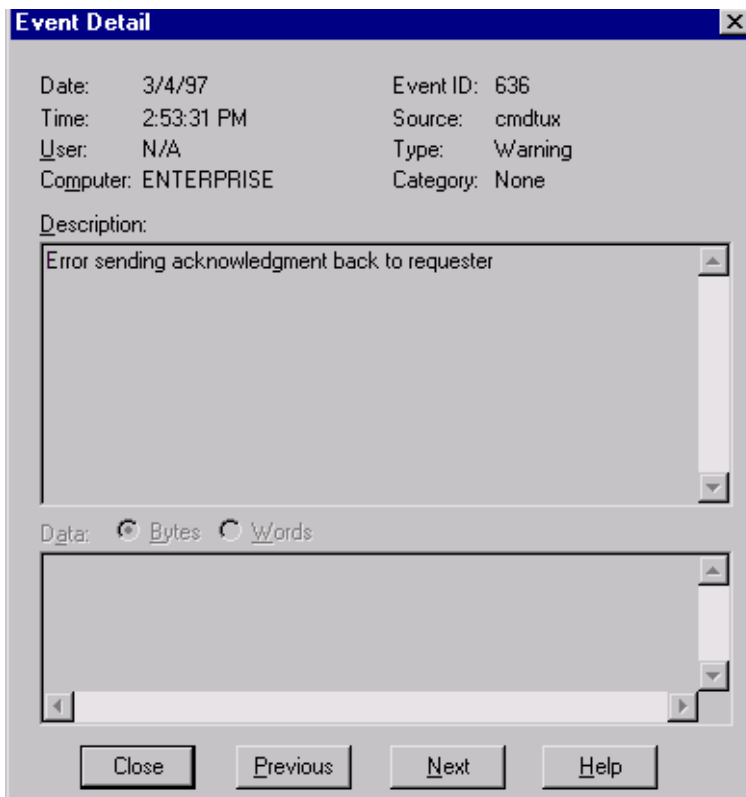
You can direct BEA TUXEDO system messages to the Windows NT Event Log by setting the LOGGING page. You can select the Logging option (Windows NT Event Log) or the traditional ULOG (Disk File), or both. If you want traditional ULOG messages, select the directory into which ULOG messages will be written, as well as the prefix for the filename. The default prefix is ULOG, and the default filename is ULOG.mmdyy.

Figure 3-5 BEA TUXEDO System for Windows NT Logging Page



To view Event Log entries, select Start→Programs→Administrative Tools →Event Viewer. From the Event Viewer Log window, verify that Application is checked.

Figure 3-6 BEA TUXEDO System for Windows NT Event Detail Window

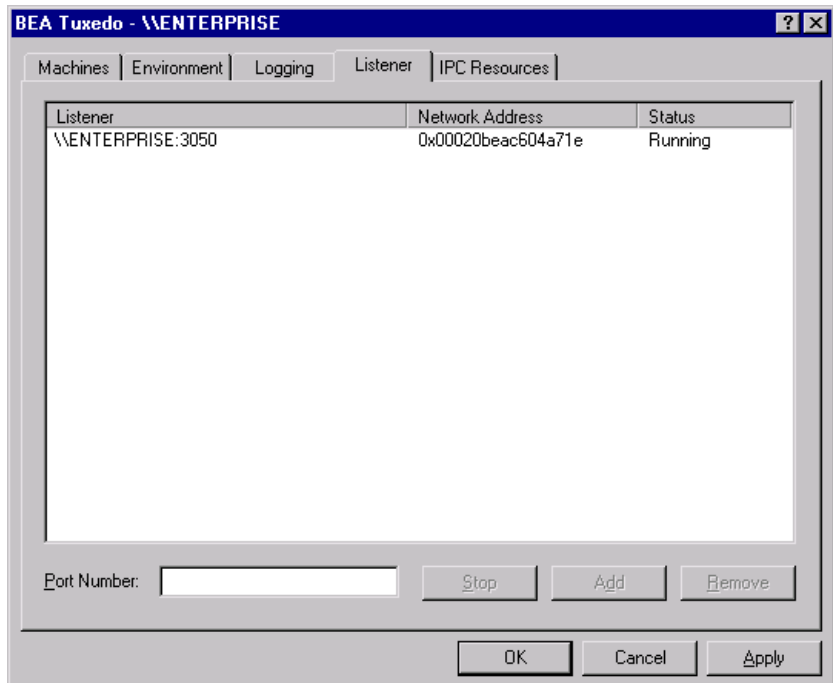


Each message identifies the message number and catalog that you can use to look up the description and recommended action in the *BEA TUXEDO System Message Manual*. Messages also have a type (that is, Information, Warning, or Error) and brief description.

Configuring tlisten Processes to Start Automatically

You can configure one or more `tlisten` processes to start automatically when you boot up your system. On the Listener page, enter a port number in the Port Number field and click Add to add the service to the list. After you click OK or Apply, and reopen the control panel, you can start or stop `tlisten` services right on the LISTENER page. You can also use the Windows NT Services control panel to start or stop a `tlisten` service or to configure the service to start automatically.

Figure 3-7 BEA TUXEDO System for Windows NT Listener Page



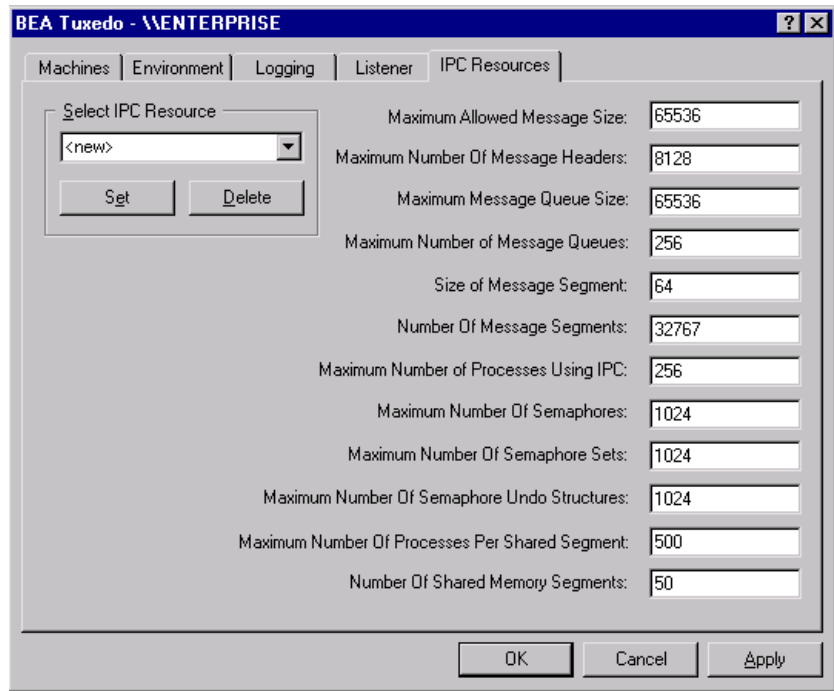
You can use the `tlisten` program to perform administrative actions in an application across multiple computers. You must start the `tlisten` program on each computer before running the application. Generally, you need one `tlisten` for each BEA TUXEDO application running on the computer.

Maximizing System Performance

The BEA TUXEDO system for Windows NT provides you with BEA TUXEDO IPC Helper (TUXIPC), an interprocess communication subsystem, installed with the product. On most systems, IPC Helper runs as installed; however, you can use the `RESOURCES` page of the control panel applet to tune the TUXIPC subsystem and maximize performance.

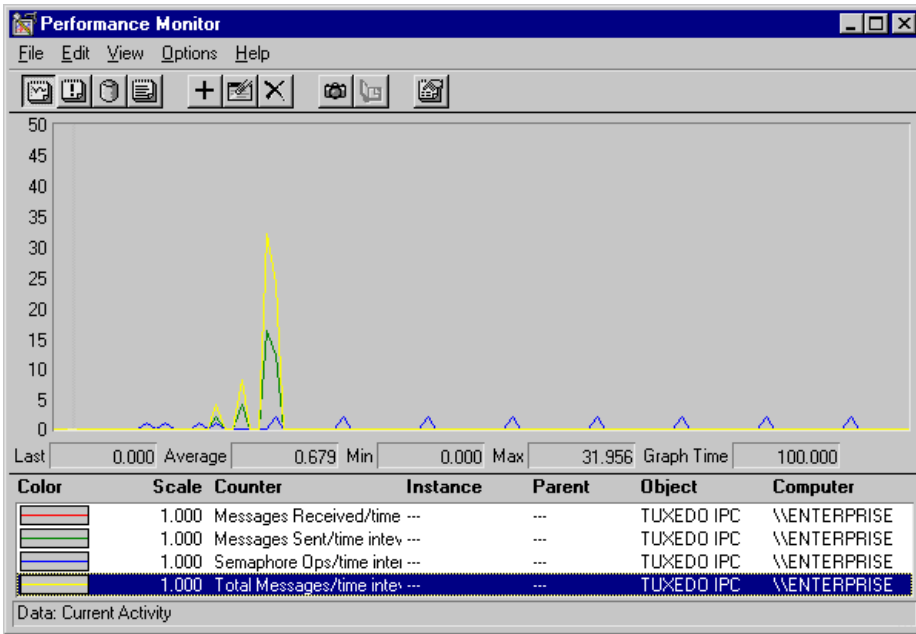
With the IPC `RESOURCES` page of the control panel, you can set a variety of IPC resources to the right of each box. In the Select IPC Resources field, you can name a set of resources. Selecting the default `Medium` disables any changes to the parameters. If you backspace over `Medium` (a single backspace removes the whole string), the parameters are made available for changes. To add a new configuration, type `New`, edit the BEA TUXEDO IPC parameters as needed, and click `Set` to associate the name with the revised set of resources. Clicking `Apply` saves the changes in the Registry Table. You must then stop and then restart the `tuxipc.exe` service for the changes to take effect.

Figure 3-8 BEA TUXEDO System for Windows NT IPC Resources Page



You can view the performance of a running BEA TUXEDO application on the NT Performance Monitor.

Figure 3-9 BEA TUXEDO System for Windows NT Performance Monitor



3 *Configuring the BEA TUXEDO System for Windows NT*

4 Developing a BEA TUXEDO Application on Windows NT

Introduction

This chapter provides instructions for the following:

- ◆ Using the BuildTuxedo tool, available with the BEA TUXEDO system on Windows NT, for developing a BEA TUXEDO application in the Microsoft Integrated Development Environment (IDE)
- ◆ Debugging a BEA TUXEDO server application in IDE
- ◆ Developing a BEA TUXEDO application without Visual C++

The BEA TUXEDO system integrates the software into the Microsoft Visual C++ (msdev) environment and emulates the functionality of msdev when integration is not possible. This new feature makes it easier for you to develop BEA TUXEDO system enabled applications for 16-bit and 32-bit Windows operating systems. The BuildTuxedo system operates seamlessly on Windows 95 and all currently supported Windows NT environments (Intel and Alpha).

The coding required to create BEA TUXEDO service requests, send requests, set up conversational connections, and get replies is fundamentally the same in both UNIX and NT environments.

For More Information

Refer to the following books for information on the BEA TUXEDO ATMI:

- ◆ *BEA TUXEDO Programmer's Guide*
- ◆ *BEA TUXEDO FML Programmer's Guide*
- ◆ *BEA TUXEDO COBOL Guide*
- ◆ *BEA TUXEDO Reference Manual* (Sections 3c, 3cbl, and 3fml document the BEA TUXEDO ATMI)

Development Tools

- ◆ `BuildTuxedo` is a single tool, tightly integrated with the `msdev` build environment, that developers can use instead of the `buildserver`, `buildclient`, and `buildcvt` commands. (Of course, the commands are still available for those who prefer to use them.)
- ◆ `TUXDEV` creates, edits, and compiles multiple 16/32-bit FML tables and multiple 16/32-bit VIEW files. It also uses Multiple Document Interface (MFC) architecture so that you can use multiple views of these filetypes simultaneously.

Using the `buildserver` and `buildclient` Commands

Although the `buildserver` and `buildclient` commands are available on the Windows NT platform, there are differences between how the options to these commands work on NT platforms and non-NT platforms. The following table lists these differences.

To	In a Non-Integrated Development Environment, Use this Option(s)	In an Integrated Development Environment
Turn on verbose mode	-v	All options are displayed on tabs by default. (The -v option is unnecessary and unsupported.)
Specify an output file	-o (<i>output filename</i>)	<ul style="list-style-type: none"> ◆ Select Settings from the msdev Build menu. ◆ Select the Link tab in the Project Settings dialog box. ◆ Specify the name of your output file by using the edit control.
Specify first file to be linked	-f	<ul style="list-style-type: none"> ◆ Select Settings from the msdev Build menu. ◆ Select the Link tab in the Project Settings dialog box. ◆ Specify the first file to be linked by using the edit control.
Specify last file to be linked	-l	<ul style="list-style-type: none"> ◆ Select Settings from the msdev Build menu. ◆ Select the Link tab in the Project Settings dialog box. ◆ Specify the last file to be linked by using the edit control.
Specify a resource manager	-r	<ul style="list-style-type: none"> ◆ Access the BuildTuxedo GUI. ◆ Access the BuildTuxedo Test window and select the Resources page. ◆ In the Tuxedo Resource Manager field, enter the name of the resource manager.
Specify services that will be available on a server	-s	<ul style="list-style-type: none"> ◆ Access the BuildTuxedo GUI. ◆ Access the BuildTuxedo Test window and select the Services page. ◆ In the Service Names field, enter the name of each service on a separate line.
Use the COBOL compiler	-c	COBOL is unavailable.

To modify the build environment in an integrated development environment: (a) select Settings from the msdev Build menu, and then (b) select either the C/C++ or Link tab.

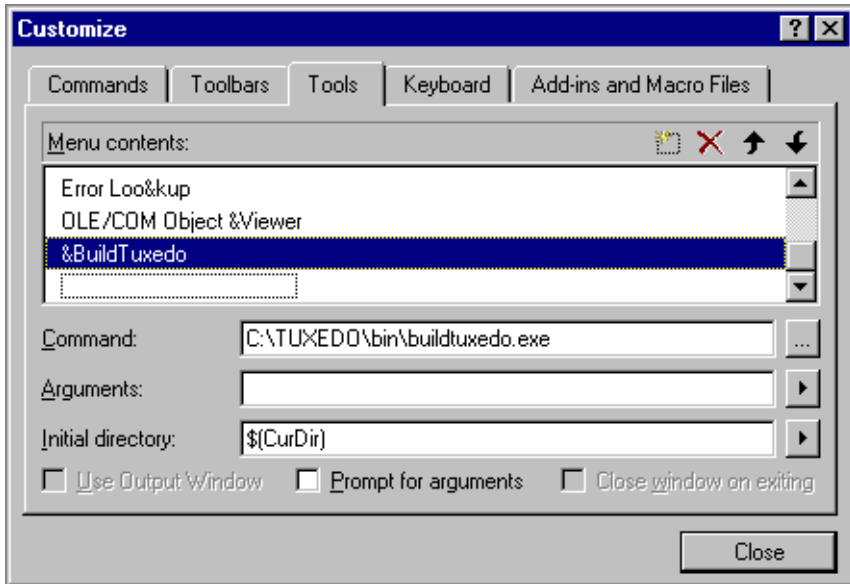
Note: CC and CFLAGS are no longer needed.

To specify the library and include paths in an integrated development environment, select Options from the msdev Tools menu. From the Options dialog box, select the Directories tab.

Adding BuildTuxedo to the MSDEV Tools Menu

1. From within Microsoft Visual C++ (msdev) graphical user interface (GUI), select Tools→Customize→Tools. The Customize window is displayed.
2. Click New (broken box icon with twinkle) and type &BuildTuxedo in the blank box at the end of the Menu Contents box. Placing an ampersand (&) before any letter enables it as the hot key.
3. In the Command field, type the entire path name of BuildTuxedo (%TUXDIR%\bin\BuildTuxedo) or select ... to search for the name. Click Open to display the name in the Command field.
4. In the Initial Directory field, click the right arrow and select Current Directory.
Note: To modify an item, highlight it and type over it. To move any item, highlight it and select the up or down arrow at the top of the Menu Contents box.
5. Click Close.

Figure 4-1 BEA TUXEDO System for Windows NT Tools Menu



Creating BEA TUXEDO Project Files

BuildTuxedo maintains a separate project file in the current directory for each BEA TUXEDO application using it. When BuildTuxedo begins, it searches for a valid project file in the current directory. If one is found, the various dialog controls are set to the values stored in the file and the dialog is displayed. The title bar displays the following.

BuildTuxedo <*project name*>

Because your BuildTuxedo project is closely associated with the msdev project in the current directory, BuildTuxedo also searches for two other files:

- ◆ A valid msdev make file (*filename.mak*)
- ◆ An msdev project file (*filename.mdp*, *filename.dsw*, or *filename.dsp*).

If BuildTuxedo cannot find one or both of these files, it displays a warning and/or fails to activate. If the directory contains multiple BuildTuxedo project files, multiple msdev project files or make files, menu items that contain appropriate target names are added to the System menu. To save the current project file, select OK or Apply. To cancel any changes that you made to the project file or any file that BEA TUXEDO maintains, select Cancel or press Esc.

Setting Up Your Environment

Before you can build your BEA TUXEDO application in an IDE, you must set the following fundamental parameters in your environment:

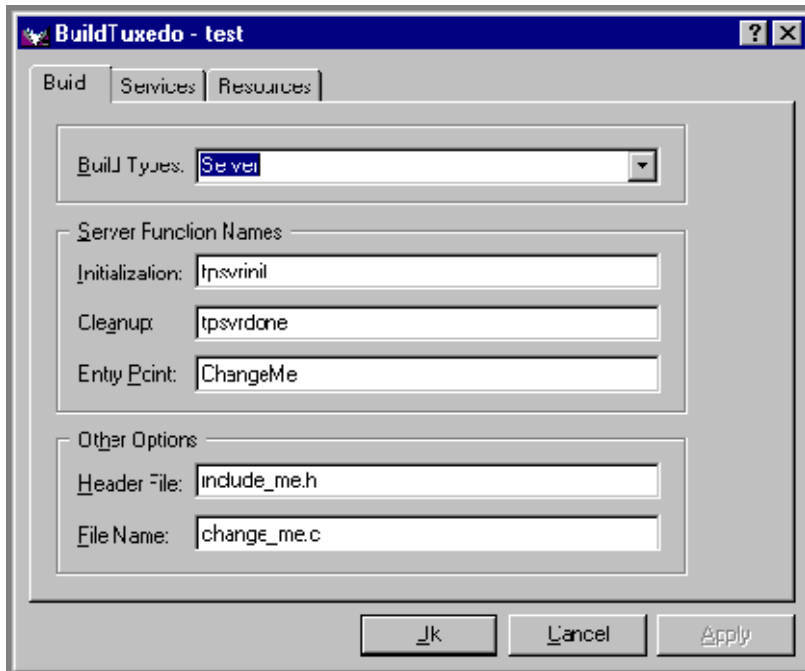
- ◆ A build type
- ◆ A header file
- ◆ The name of the C or C++ file to be created and maintained by BuildTuxedo
- ◆ Function names
- ◆ Service names

To provide this information, access the BuildTuxedo Project dialog box in the msdev GUI.

How to Specify the Build Type, Header File, and Filename

1. From the BuildTuxedo dialog box, select the Build page.

Figure 4-2 BEA TUXEDO System for Windows NT Build Page



2. In the Build Types field, click the down arrow and choose one of the following.

Select	Then
Server	Enter information in the Initialization, Cleanup, and Entry Point fields. Then go to Step 3.
Native Client or Workstation Client	Go to Step 3.

Note: Windows 95 users can select Workstation Client type only.

- Note:** The Initialization and Cleanup options allow you to override the default init/exit functions by specifying valid function names. The Entry Point option allows you to specify the name of the function that BuildTuxedo generates. You can then call this function from anywhere in the application.
3. In the Header File field, enter `stdafx.h` (the standard `msdev/` /MFC precompiled header file).
 4. In the File Name field, enter the name of the C or C++ file to be generated and maintained by BuildTuxedo.

How BuildTuxedo Uses the Header File

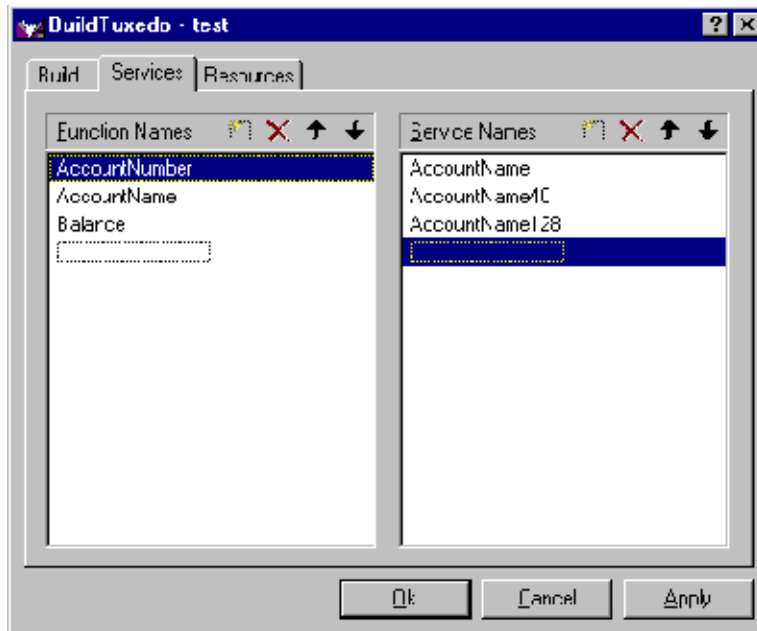
In the Header File, BuildTuxedo adds the necessary pragma statements to build the current BEA TUXEDO project correctly. BuildTuxedo opens the file specified by Header File if the file is present; otherwise, it creates one. A section titled: “//Begin TUXEDO Section*****DO NOT EDIT*****” and ending with “//End TUXEDO Section” designates the area in the header file that BuildTuxedo maintains and into which pragma statements are written. If BuildTuxedo searches for and does not find this section, it appends the section to the end of the header file. Because all other text within the file remains unchanged, you can specify `stdafx.h` as the header file.

Note: If this is a new project and you select OK or Apply, you must select Files into Project from the `msdev` Insert menu. You then add the file generated by BuildTuxedo to the current project. You need to do this only for a new project or when you change the name of the C/C++ output file.

Specifying Services

Two lists are used to maintain the service dispatch table. Function Names is a user-maintained list of functions that you can associate with a service. Service Names is a user-maintained list of the associated services. Generally, the service and the function that performs the service are represented by the same name: function `x` performs service `x`. In some cases, the function may have a different name from the service it performs. For example, in one case, function `abc` performs services, `x`, `y`, and `z`. In another case, the service name is not known until runtime.

Figure 4-3 BEA TUXEDO System for Windows NT Services Menu



You must specify any function associated with a service when you build the server. For any function associated with a service, you must specify the service, the appropriate prototype, a C linkage, a void return, and take a single `TPSVCINFO` pointer parameter. To specify a function to which a service name can be mapped, you must add the function to the Function Name list. This information is required for the service dispatch table. (If you are using the `buildserver` command, you can provide this information with the `-s` option. Refer to the *BEA TUXEDO Programmer's Guide* for details.)

How to Add or Edit Names on the Function Names List

1. In the Services folder, choose one of the following actions.

To	Action
Add an item	Click New (broken box with twinkle in the upper left corner), press Insert, or select a blank area on the list, (i.e., a dotted lined box). Type the name of the new function.
Modify an existing name	Highlight the name and type your changes over the entry.
Delete a name	Highlight the name on the list and click X (X icon) or press Delete.

2. When complete, press Enter to save your changes and exit. (To exit without saving your changes, press Esc or Tab.)

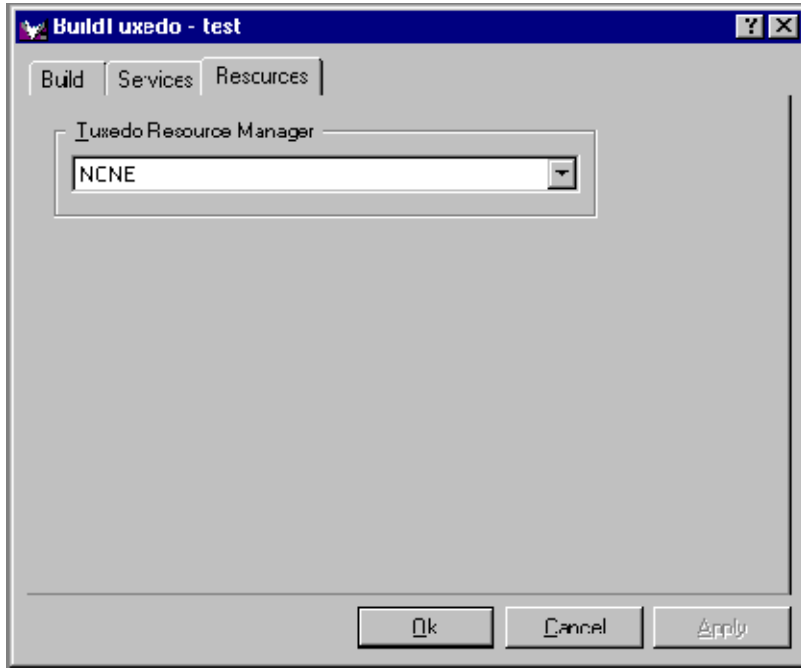
Note: To scroll up and down the list, use the arrow keys.

How to Specify a Resource Manager

1. Select the `Resources` page from the BuildTuxedo menu.
2. In the Tuxedo Resource Manager field, type the entire pathname of the resource manager for your application.

Note: The Tuxedo Resource Manager field contains a list of BEA TUXEDO resource managers available on the system as defined in the `%TUXDIR%\udataobj\RM` file. If the file is not present, a default (`NONE`) is displayed.

Figure 4-4 BEA TUXEDO System for Windows NT Resources Page



How to Debug a BEA TUXEDO Server Application

To debug a server that has not been booted, complete the following procedure. Use this procedure only if you have built the server application using the Debug configuration for your project.

1. At any Windows NT Command prompt, type `tmboot -n -d 1 -s <servername>` to display the command line options `tmboot` uses to start `<servername>`.
2. Run `tmboot -M` to boot the BBL. (If necessary, boot any additional application servers or machines as needed.)
3. Select Project→Settings in `msdev`.
4. In the Program Arguments field, select Debug and type the command line options used in Step 1.
5. Start the debugger and proceed as needed to debug the server application.

Note: Because BEA TUXEDO libraries are not built with debugging information and source code is not provided, you will not be able to access the BEA TUXEDO code directly.

6. To end the debugging session, type `tmshutdown` at a Windows NT prompt.

Warning: Do not select Debug→Stop to stop the server as the BEA TUXEDO system may attempt to restart it.

Note: To debug a server that is already running, type `msdev -p nnn` at a Windows NT Command prompt, replacing `nnn` with the server's process ID (represented by a decimal number).

Developing a BEA TUXEDO Application Without Visual C++

If you need to develop a BEA TUXEDO application without Visual C++, use the `buildserver` and `buildclient` commands. To do this, specify the compiler and link options necessary to build a BEA TUXEDO application. Refer to Section 5 of the *BEA TUXEDO Reference Manual*, *BEA TUXEDO Application Development Guide*, and *BEA TUXEDO Programmer's Guide* for information on using these tools.

To build a debug version of your application using `buildserver` or `buildclient`, you must compile all source files with the `/zi` and `/od` options. The `/zi` option enables debugging; the `/od` option disables optimization. You may also need to define the `_DEBUG` preprocessor directive. To complete the process, indicate the link option as follows:

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

The New Tuxdev Application

Installing Tuxdev

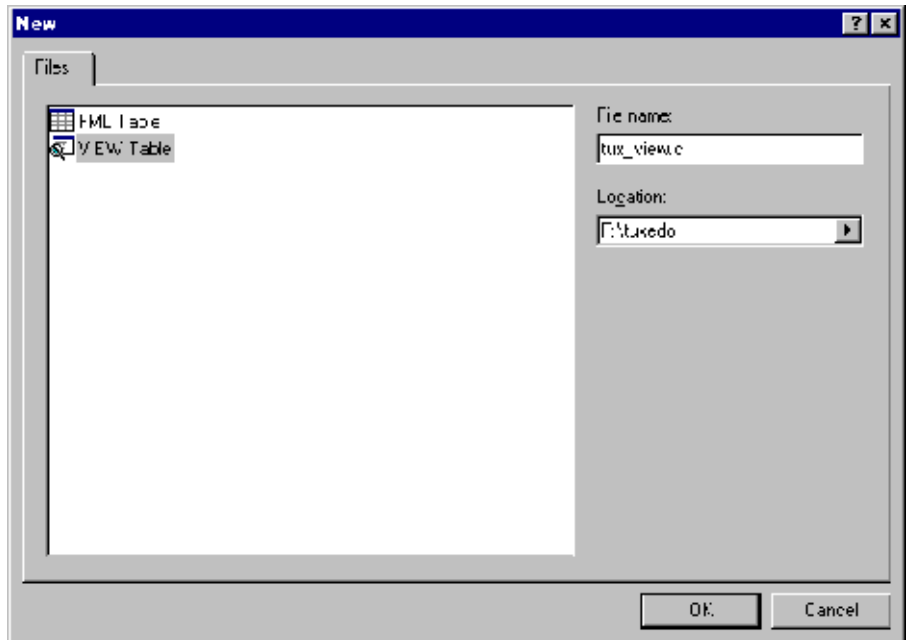
1. In the Microsoft Visual C++ (msdev) environment, select Tools→Customize→Tools→Add.
2. In the Add Tool dialog box, type the entire path name in the Command field (`TuxDev.exe` is located in `%TUXDIR%\bin`), or select Browse to search for the name. Click OK to display the pathname in the Customize window.
3. To change the item displayed on the Tools menu, change the entry in the Menu Text field.
4. To create a hot key for a tool, position an ampersand (&) before any letter in the tool name. (Now you can bring up the tool at any time by typing that letter.)
5. In the Initial Directory field, click on the right arrow and select Current Directory to display its name (`$CurDir`).

BEA TUXEDO Editors

Two editors are available in this environment: the FML Table editor and the VIEW Table editor. The user interface for both editors is similar to a workbook in which you can work on multiple documents/views simultaneously. You can also edit multiple files of various types at the same time. Both editors closely resemble a Microsoft Excel spreadsheet with all its associated functionality:

- ◆ The FML Table editor enables developers to create and/or edit an FML field table easily, and to generate user selectable 16- or 32-bit field header files.
- ◆ The VIEW Table editor enables developers to create and/or edit a VIEW Table file easily, and to generate user selectable 16- or 32-bit VIEWS and header files.

Figure 4-5 Accessing the Editors



Using the FML Table Editor

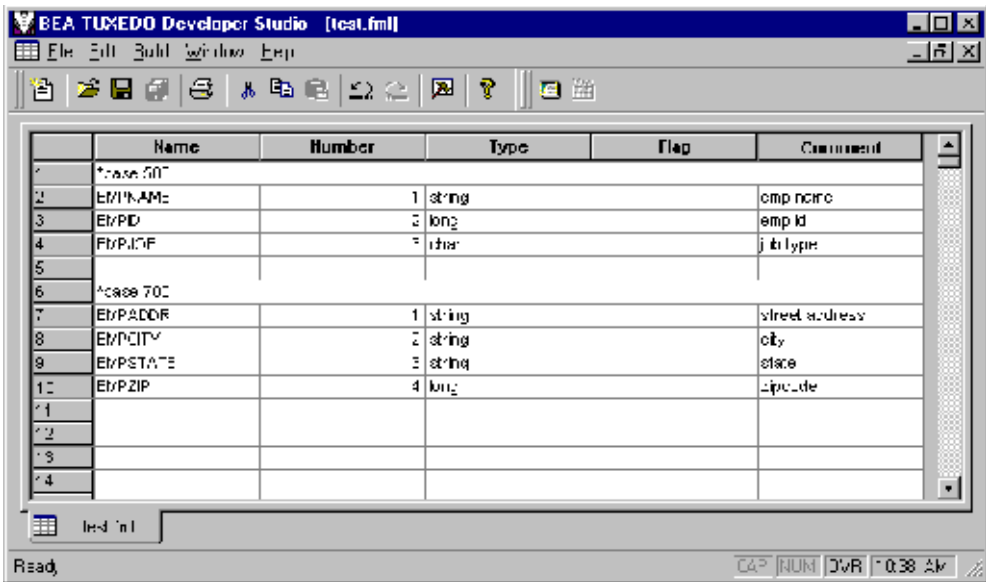
Accessing the FML Table Editor

1. On the Command line, type the entire pathname of the Tuxdev (%TUXDIR%\bin). Type tuxdev and press Enter.
2. Choose one of the following actions.

To	Action
Create a file	From the BEA TUXEDO Developer's window, select File →New. Go to Step 3.
Modify an existing file	From the BEA TUXEDO Developer's window, select File →Open. Select the file (fml files have a .fml extension) to edit.

3. On the New window, highlight FML Table and type in the name of the new file in the File Name field. Select OK.

Figure 4-6 FML Table Editor



Description of the FML Table Editor

The FML Table Editor illustrated in Figure 4-6 contains five columns: Name, Number, Type, Flag and Comment, with an unlimited number of rows. The following chart outlines the purpose of each column in the editor.

This Column	Enables You to
Name	Enter comments and/or base numbers provided each begins with a pound sign (#) or an asterisk (*). If the Name is not preceded by # or *, the line is assumed to be an active table entry. Data entered in an empty cell is assumed to be a new entry. (Blank lines are allowed.)
Number	Specify the relative number of the field
Type	Select from a list of all allowable values for this field
Flag	Select the flags displayed next to check boxes (when column is active)
Comment	Expand or clarify any entry designated in the Name column

You can open new (unnamed) FML tables in which a blank grid is created for the table named “FML Table x ”, where x is a value tracked by Multiple Document Interface (MFC) and incremented by one each time a new table is created. You can specify a name if and when the table is saved. You can also open an existing text file for editing. Unless otherwise specified, files are saved in the directory in which the file was opened in tabbed delimited format, with .FML appended to the end of the filename when applicable. You can compile this file to produce either a 16-bit or 32-bit FML header file.

Using the VIEW Table Editor

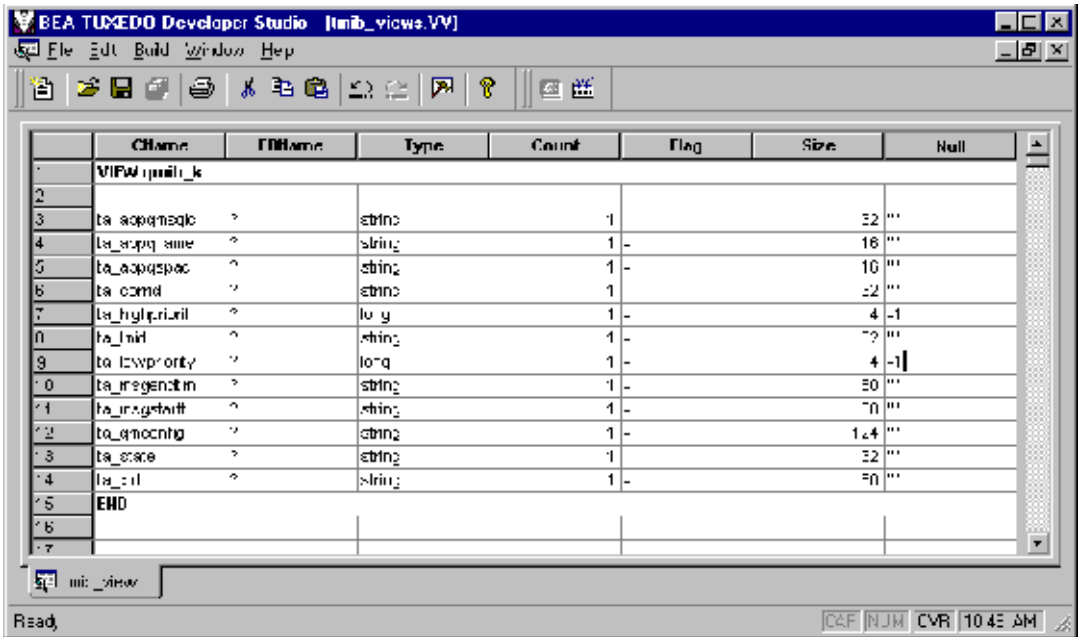
Accessing the VIEW Table Editor

1. On the Command line, type the entire pathname of `Tuxdev (%TUXDIR%\bin)`. Type `tuxdev` and press Enter.
2. Choose one of the following actions.

To	Action
Create a file	From the BEA TUXEDO Developer's window, select File →New. Go to Step 3.
Modify an existing file	From the BEA TUXEDO Developer's window, select File →Open. Select the file (view files have a .vv extension) to edit.

3. On the New window, highlight VIEW Table and type in the name of the new file in the File Name field. Select OK.

Figure 4-7 VIEW Table Editor



Description of the VIEW Table Editor

The VIEW Table Editor illustrated in Figure 4-7 contains seven columns: CName, FBName, Type, Count, Flag, Size, and Null, with an unlimited number of rows.

You can enter comments in the CName column provided each begins with the required pound sign (#). Blank lines are also allowed. When a CName entry is not preceded by a pound sign and is not NULL, the line is assumed to be an active table entry.

This Column	Enables You to
CName	Enter comments provided each begins with a pound sign (#). (If the CName is not preceded by #, the line is assumed to be an active table entry. (Blank lines are allowed.)
FBName	Name the field in the fielded buffer; this name displays in the field table file
Type	Select from a list of all allowable values for this field
Count	Specify the number of elements to allocate (that is the maximum number of occurrences to be stored for this member).
Flag	Select the flags displayed next to check boxes (when column is active)
Size	Indicate the size of the member if the type is string, carray, or dec_t; otherwise, specify '-', and the view compiler computes the size
Null	Specify a null value or '-' (default null value) for that field

Note: You must type `VIEW <table name>` in a row by itself in the CName column before the view information, and type `END` in a row by itself in the CName column to end the view information. You can enter multiple views within the same file provided that each table entry is preceded by `VIEW <table name>` and followed by `END`.

You can open new (unnamed) view files in which a blank grid is created for the view, *Viewx*, where *x* is a value tracked by MFC and incremented by one each time you create a new view. You can specify a name if and when you save the view file. You can also edit an existing view file (either text or binary). Unless otherwise specified, files are saved in the directory where the file was opened in tabbed delimited format, with *v* appended to the end of the filename when applicable. You can compile this file to produce either a 16- or 32-bit binary `VIEW` and header files.

Working in Multiple Documents Simultaneously

The Multiple Document Interface (MFC), as part of the basic framework, provides a multiple document architecture in which you can open documents and views, regardless of type, at the same time. Examples of this design are msdev, Excel, and Word. In BEA TUXEDO terms, you can open x number of FML tables and y number of VIEW Table files at any time, and then select to use any one of them. Documents look and feel like a workbook that contains tabs for each open document.

How the Editors Validate Entries

The following tables describe what information is validated in each column of the FML Table and VIEW Table editors.

The FML Table Editor

In This Column	This Information Is Validated
Name	Comments, base numbers, and valid text strings
Number	Numbers only. (The range is determined by a 16/32-bit user mode.)
Type	Valid FML types
Flag	Valid FML flags. Extra checking is done for mutually exclusive flags.
Comment	Entries are not validated.

The VIEW Table Editor

In This Column	This Information Is Validated
CName	Entries are not validated.
FBName	Entries are not validated.
Type	Valid BEA TUXEDO types
Count	Numbers only. (The range is determined by a 16/32-bit user mode.)
Flag	Valid FML flags. Extra checking is done for mutually exclusive flags.
Size	Numbers only. (The range is determined by a 16/32-bit user mode.)
Null	Entries are not validated.

