THE ENTERPRISE MIDDLEWARE SOLUTION

BEA TUXEDO

for the AS/400 User's Guide

BEA TUXEDO Release 6.5
Document Edition 6.5
February 1999

Copyright

Restricted Rights Legend

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 TUXEDO for the AS/400 User's Guide

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

Contents

Preface

1. Installing the BEA TUXEDO System on the AS/400

2. Using the BEA TUXEDO System on UNIX and AS/400 Platforms

Preface

Purpose of This Manual

This book describes BEA TUXEDO™ Release 6.5 for the AS/400® platform.

Who Should Read This Manual

This book is primarily intended for users who are already familiar with BEA TUXEDO (perhaps on a UNIX system) but who are new to the AS/400 system.

How This Manual Is Organized

This book is organized as follows:

- Chapter 1, "Installing the BEA TUXEDO System on the AS/400," provides installation instructions for the BEA TUXEDO Core System and the Link-Level Encryption add-on packages.
- Chapter 2, "Using the BEA TUXEDO System on UNIX and AS/400 Platforms," describes the differences between running a BEA TUXEDO application on a UNIX platform and an AS/400 platform.
- Chapter 3, "Configuration for BEA TUXEDO System Administration," provides instructions for creating a profile for a BEA TUXEDO administrative group and requesting `*JOBCTL` authorization in an administrator's user profile.

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys sequentially.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR

Convention	Item
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f `*`file-list`*`]...`
	`[-l `*`file-list`*`]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	♦ That an argument can be repeated several times in a command line
	♦ That the statement omits additional optional arguments
	♦ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f `*`file-list`*`]...`
	`[-l `*`file-list`*`]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Related Documentation

This section lists books that may be helpful to an administrator of a BEA TUXEDO application running on an AS/400 platform.

BEA TUXEDO Manuals

- *Administering the BEA TUXEDO System*
- *BEA TUXEDO Application Development Guide*
- *BEA TUXEDO for the AS/400 User's Guide*
- *BEA TUXEDO COBOL Guide*
- *BEA TUXEDO Domains Guide*
- *BEA TUXEDO FML Programmer's Guide*
- *BEA TUXEDO Glossary*
- *BEA TUXEDO Product Overview*
- *BEA TUXEDO Programmer's Guide*
- *BEA TUXEDO /Q Guide*
- *BEA TUXEDO Reference Manual*
- *BEA TUXEDO System Message Manual*
- *BEA TUXEDO TxRPC Guide*
- *BEA TUXEDO for Windows NT User's Guide*
- *BEA TUXEDO Workstation Guide*

IBM Manuals

- *CL Reference Manual*
- *ILE Concepts*
- *ILE/C 400 Programmer's Guide*
- *ILE/C 400 Programmer's Reference*
- *ILE/COBOL 400 Programmer's Guide*
- *ILE/COBOL Programmer's Reference*
- *Introduction to IFS*
- *UNIX C Applications Porting to AS/400* (IBM Redbook)

BEA TUXEDO for the AS/400 Documentation

The BEA TUXEDO for the AS/400 information set consists of the following documents:

- *BEA TUXEDO for the AS/400 User's Guide*
- Platform Data Sheets information in the *BEA TUXEDO Installation Guide (available online only)*

Note: The BEA TUXEDO Online Documentation CD includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

Other Publications

For more information about BEA TUXEDO, refer to the following books:

- *Building Client/Server Applications Using TUXEDO* (Hall)
- *Instant CORBA* (Edwards, Harkey, Orfali)

- *The TUXEDO System* (Andrade, Carges, Dwyer, Felts)
- *TUXEDO: An Open Approach to OLTP* (Primatesta)

Contact Information

The following sections provide information about how to obtain support for the documentation and software.

Documentation Support

If you have questions or comments on the documentation, you can contact the BEA Information Engineering Group by e-mail at **docsupport@beasys.com**. (For information about how to contact Customer Support, refer to the following section.)

Customer Support

If you have any questions about this version of BEA TUXEDO for the AS/400, or if you have problems installing and running BEA TUXEDO for the AS/400, contact BEA Customer Support through BEA WebSupport at `www.beasys.com`. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

CHAPTER

1 Installing the BEA TUXEDO System on the AS/400

This chapter provides prerequisites and instructions for installing the BEA TUXEDO system on an AS/400 platform.

Note: For information on available BEA TUXEDO packages, hardware, software, and disk space requirements for the AS/400 platform, refer to Appendix A ("Platform Data Sheets") in the *BEA TUXEDO Installation Guide*. The guide is available on the documentation CD-ROM delivered with Release 6.5. It is also available (as part of the Release 6.5 online documentation set) through the Customer Support Page at the BEA Systems Web site.

Interoperability

BEA TUXEDO Release 6.5 running on an AS/400 platform can interoperate, as either master or non-master, with Release 6.5 running on any other supported platform.

Backward compatibility (that is, interoperability between Release 6.5 and prior releases) is supported only if the master node is running Release 6.5.

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

The Installation Process

This section provides instructions for preparing your environment for the BEA TUXEDO system, installing the BEA TUXEDO system from an NT, WIN95, or UNIX system platform, and installing the Link-Level Encryption add-on security package.

Before You Begin

Before you start installing the BEA TUXEDO system, sign on to your AS/400 system and make sure you have a *tuxedo* user profile. If you do not, create one now. The user defined in this profile will be the owner of all the program objects created during the installation process.

Installing from an NT or WIN95 Platform

1. On the network to which your AS/400 machine is connected, find a PC with a CD-ROM drive, and insert the CD.

2. Create a 3-line file containing the following `ftp` information:

   ```
   open as400-machine-name
   account-name
   password
   ```

   For example, suppose that (a) you are working on an AS/400 machine called `lc400b.beasys.com`, (b) your login is `rtl`, and (c) your password is `rtl123`. Your profile will contain the following three lines:

   ```
   open lc400b.beasys.com
   rtl
   rtl123
   ```

   This file will be used to automatically execute the pre-installation procedures necessary for installing the BEA TUXEDO system 6.5 through `ftp`. Once the pre-installation steps are completed, the file should be removed.

3. Invoke the pre-installation batch file, `preinst.bat`, with the appropriate command-line arguments.

   ```
   preinst.bat CD-drive tmplib tuxdir tuxlib filename version
   ```

   where:

   - *`CD-drive`* is a single-letter abbreviation (such as D:) for the CD-ROM device on the machine on which the BEA TUXEDO system 6.5 is loaded.
   - *`tmplib`* is the library on the AS/400 machine in which copies of the BEA TUXEDO save files (on the CD-ROM) will be stored.
   - *`tuxdir`* is the IFS file system on the AS/400 machine on which the BEA TUXEDO system will be installed.
   - *`tuxlib`* is the library on the AS/400 machine on which the save files will be restored.
   - *`filename`* is the full path to a file that contains `ftp` account and password information for the AS/400 machine on which the BEA TUXEDO system will be installed.
   - *`version`* is the version number of the OS/400 operating system for the AS/400 machine on which the BEA TUXEDO system will be installed.

4. Run the install program on the AS/400.

   ```
   addlible tuxlib
   call coreinst (tmplib tuxlib '/tuxdir' 'all|client')
   ```

   where you specify one of the following:

   - `client` (to install only the BEA TUXEDO workstation package)
   - `all` (to install both the server and workstation packages)

   Note: The words `client` and `all` must be entered in lowercase.

   During the installation process (which may take several minutes), a number of `restoring X to Y` messages are displayed, such as the following.

   ```
   restoring service programs to tuxlib
   ```

5. Remove, from *`tmplib`*, all the save files that were created in Step 3. Remove *`tmplib`*, too, if it is different from *`tuxlib`* and it is no longer needed.

6. Initialize the `tlisten` passwords. The `tlisten` password is used for authentication during the boot process of a multiple-node application. To initialize the password file, enter the following command.

```
call tlistpwd 'tuxdir'
```

where the value of *`tuxdir`* is the directory specified as *`tuxdir`* in Step 4. This command will prompt for passwords until a null password is entered. For further details about the password file, see `tlisten` in the *BEA TUXEDO Reference Manual*.

7. Your installation of the BEA TUXEDO system is now complete; you can start using it as soon as a license file is installed in `$TUXDIR/udataobj/lic.txt`. For information on obtaining a license, contact your BEA TUXEDO sales representative.

Installing from a UNIX System Platform

1. On the network to which your AS/400 machine is connected, find a UNIX system with a CD-ROM drive and insert the CD-ROM.

2. Set up a `.netrc` file to allow `ftp` to sign on to the AS/400 machine. For example, suppose that (a) you are working on an AS/400 machine called `lc400b.beasys.com`, (b) your login is `rtl`, and (c) your password is `rtl123`. Your `.netrc` file will contain the following three lines.

```
machine lc400b.beasys.com
login rtl
password rtl123
```

3. Invoke the preinstallation shell, `preinst.sh`, with the appropriate command-line arguments:

```
preinst.sh mount-point tmplib tuxdir tuxlib machine version
```

where:

- *`mount-point`* (such as `/cdrom`) is the file system on which the BEA TUXEDO 6.5 CD-ROM is mounted.
- *`tmplib`* is the library on the AS/400 to which the BEA TUXEDO save files on the CD-ROM are copied.
- *`tuxdir`* is the IFS file system (on the AS/400) in which the BEA TUXEDO system is installed.
- *`tuxlib`* is the library on the AS/400 in which the save files are restored.

- ♦ *machine* is the name of the AS/400 machine on which the BEA TUXEDO system is installed.
- ♦ *version* is the version number of the OS/400 operating system for the AS/400 machine on which the BEA TUXEDO system will be installed.

4. Run the install program on the AS/400.

```
addlible tuxlib
call coreinst (tmplib tuxlib '/tuxdir' 'all|client')
```

   where you specify one of the following:

   - ♦ `client` (to install only the BEA TUXEDO workstation package) or
   - ♦ `all` (to install both the server and workstation packages)

   Note: The words `client` and `all` must be entered in lowercase.

   During the installation process (which may take several minutes), a number of `restoring X to Y` messages will be displayed, such as.

```
restoring service programs to tuxlib
```

5. Remove, from *tmplib*, all the save files that were created in Step 3. Remove *tmplib*, too, if it is different from *tuxlib* and it is no longer needed.

6. Initialize the `tlisten` passwords. The `tlisten` passwords are used for authentication during the boot process of a multiple-node application. To initialize the password file, enter the following command.

```
call tlistpwd 'tuxdir'
```

   where the value of *tuxdir* is the directory specified as *tuxdir* in Step 4. This command will prompt for passwords until a null password is entered. For further details about the password file, see `tlisten` in the *BEA TUXEDO Reference Manual*.

7. Your installation of the BEA TUXEDO system is now complete; you can start using it as soon as a license file is installed in `$TUXDIR/udataobj/lic.txt`. For information on obtaining a license, contact your BEA TUXEDO sales representative.

Installing the Security Add-on Package

Release 6.5 offers two types of the optional Security Add-on Package:

- Security Add-on Package (International) provides 56-bit link-level encryption (LLE) for BEA TUXEDO applications outside the U.S. and Canada.
- Security Add-on Package (US/Canada) provides 128-bit link-level encryption (LLE) for BEA TUXEDO applications inside the U.S. and Canada.

Both packages are delivered on separate CD-ROMs. Neither can be installed unless the Core package has already been installed.

To install a Security Add-on Package, complete the following procedure.

1. On the AS/400, find an HFS library in which you can store the contents of the CD. We'll refer to this library as *tmplib*. If you cannot find such a library, create one by entering the following command.

   ```
   crtlib tmplib
   ```

2. Create a save file in *tmplib* as follows.

   ```
   crtsavf tmplib/ENCbit
   ```

   where the value of *bit* is one of the following:

   - 56 (for 56-bit encryption)
   - 128 (for 128-bit encryption)

3. Use `ftp` to move the corresponding file from the `as400/v4r1` subdirectory on the CD to *tmplib*. Remember to use binary mode for the transfer.

   For example, if you are working in a DOS Prompt window on a WIN95 machine, your session will look something like the following.

   ```
   C:\> e:
   E:\> cd \as400\v4r1
   E:\AS400\V4R1> ftp as400a
   Connected to as400a.
   220-QTCP at as400a.
   220 Connection will close if idle more than 5 minutes.
   User (as400a:(None)): rtl
   331 Enter password.
   Password:*****
   230 RTL logged on.
   ftp> binary
   200 Representation type is binary IMAGE.
   ftp> put ENC40 tmplib/ENC40
   200 PORT subcommand request successful.
   150 Sending file to member ENC40 in file ENC40 in library tmplib.
   250 File transfer completed successfully.
   ```

```
28854144 bytes sent in 76.62 seconds (376.59 Kbytes/sec)
ftp> quit
```

Note: Repeat this procedure for all save files created in Step 2.

4. Run the install program.

   Note: Remember that because the installation program for add-on products is installed along with the BEA TUXEDO Core package, the Core package must be installed before you can install any add-ons.

   Enter the following commands.

   ```
   addlible tuxlib
   call addoninst (tmplib tuxlib 'ENC56|ENC128')
   ```

   where you choose one of the following:

   - `ENC56` (to install the 56-bit LLE package)
   - `ENC128` (to install the 128-bit LLE package)

   During installation (which may take several minutes), status messages are displayed.

5. Remove, from *`tmplib`*, all save files created in Step 3.

6. Remove *`tmplib`* if it is different from *`tuxlib`* and it is no longer needed.

7. You have now finished installing your BEA TUXEDO add-on package.

CHAPTER

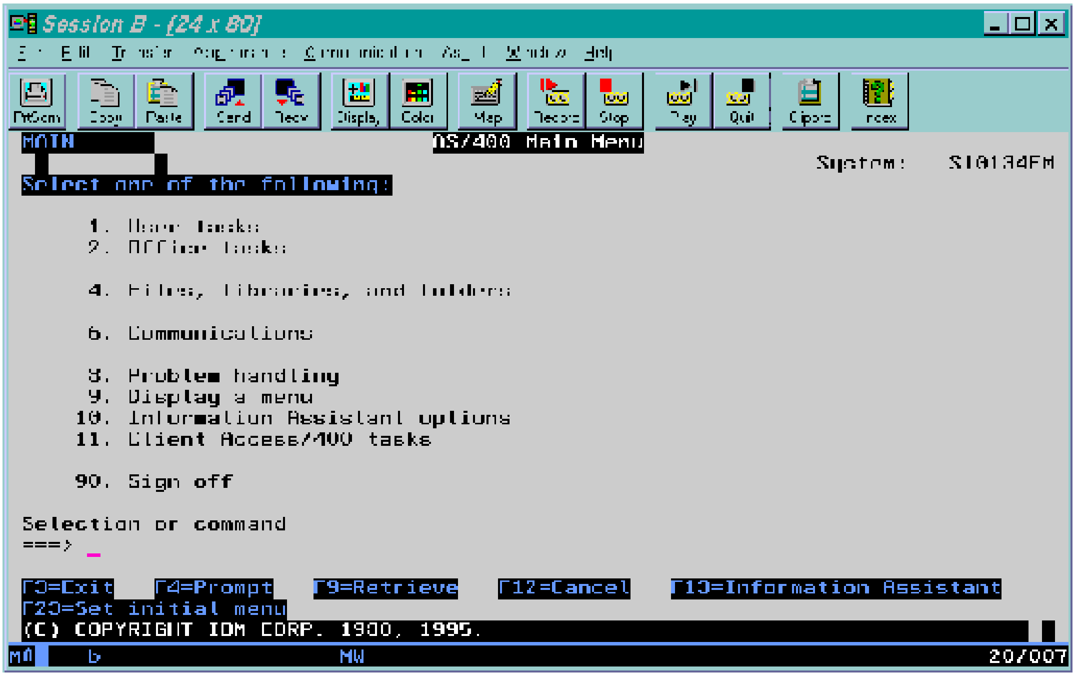
2 Using the BEA TUXEDO System on UNIX and AS/400 Platforms

User Interface and AS/400 Concepts

The user interface on an AS/400 system is completely menu-driven; there is no shell. The interface makes extensive use of function keys F1-F24.

Note: Although some screens in this chapter contain references to Release 6.3, all other information is them is accurate for Release 6.5.

Figure 2-1 Main Menu of Tasks for an AS/400 Platform



```
MAIN                          AS/400 Main Menu
                                                             System:   S10134FM
Select one of the following:

     1. User tasks
     2. Office tasks

     4. Files, libraries, and folders

     6. Communications

     8. Problem handling
     9. Display a menu
    10. Information Assistant options
    11. Client Access/400 tasks

    90. Sign off

Selection or command
===> _

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F23=Set initial menu
(C) COPYRIGHT IBM CORP. 1980, 1995.
MA   b                 MW                                              20/007
```

Invoking Programs from the Command Line

Virtually every action performed via the menu has a command-line equivalent. Application executables, such as BEA TUXEDO utilities, can be invoked from the command line using the `call` command. The `call` command has the following syntax:

```
call pgm parm(arg1 arg2 .  .  .  )
```

The notion of the `PATH` and `LD_LIBRARY_PATH` environment variables in the UNIX shell is replaced by the *`library list`*. For example, the following command:

```
call hello parm('world')
```

is a request to search in the library list for the *`hello`* program. The first library in the library list that contains this program becomes the *current library* and the *`hello`* program is *activated*.

During this activation, if a service program (the AS/400 equivalent of a DLL or shared object) is needed, then the library list is searched again. The library list can be displayed with the `dsplible` command, and manipulated with the `addlible` and `rmvlible` commands.

Interactive and Batch Jobs

Jobs define execution context (roughly equivalent to a process). There are two kinds of jobs: interactive and batch. When a user signs on to a system, an interactive job is assigned to the new session. All `call` commands are executed inside this job. The `wrkactjob` command lists all active jobs on the system. (It is equivalent to the `ps` command on a UNIX system.) A batch job can be submitted via the `sbmjob` command. The `wrksbmjob` command lists all submitted batch jobs.

HFS and IFS File Systems

This section describes the two file systems that are important on an AS/400 platform: HFS and IFS.

HFS

HFS is the traditional flat file system. At the highest level it contains libraries. Each library may contain files, e.g., module files, program files, and physical files. A physical file, in turn, may contain members. All text files (including C program source code files) are stored as members of a physical file.

Executables may reside only in HFS, but symbolic links can be created from IFS to HFS via the `addlnk` command.

Filenames

HFS filenames are limited to ten characters and are not case-sensitive.

IFS

IFS is a directory-oriented file system designed to support PC file-serving. IFS (especially `/QOpenSys`) closely resembles a UNIX file system.

Executables may reside only in HFS, but symbolic links can be created from IFS to HFS via the `addlnk` command.

Filenames

With one exception (`/QOpenSys`) filenames in IFS are case sensitive.

Editors

Because there are no native editors for IFS files, we recommend using Client Access to view and edit these files on a PC. In addition to providing a 5250 emulator, Client Access allows IFS files to be mapped as network drives on the PC. All PC Client Access users must be enrolled on the AS/400 first.

Commands for HFS and IFS Files

Separate commands must be used for HFS and IFS files:

- The `wrklib`, `wrkf`, and `wrkmbrpdm` commands work only with HFS objects.
- The `cd`, `mkdir`, `dspcurdir`, and `wrklnk` commands may be used with both IFS files and HFS files, but different syntax is required, depending on whether you are accessing HFS or IFS objects.
- The `cpytostmf` and `cpyfrmstmf` commands are available for copying a text file from one type of file system to another. For example, the following command copies an IFS file called `/QOpenSys/tuxedo/tmp/xyz` to an HFS physical file called `tmp` in a member called `xyz` (in a library called `apps`).

  ```
  cpyfrmstmf fromstmf('/QOpenSys/tuxedo/tmp/xyz') +
      tombr('/qsys.lib/tmp.file/xyz.mbr') +
       mbropt(*replace)
  ```

Character Set Conversion

Because the character sets on the AS/400 and the PC do not match, we also recommend turning on automatic EBCDIC/ASCII conversion for all files. To do so on a platform such as WIN95 or NT, complete the following procedure.

1. Open the Control Panel window.
2. Select Client Access.
3. Select Network Drive.
4. Add “*” to File extension.

Redirection of Standard IO and Standard Error

When you are working with ILE/C programs in an HFS file system, you may redirect standard input, standard output, and standard error (the `stdin`, `stdout`, and `stderr` files, respectively) by using the `ovrdbf` command. See the *CL Reference Manual* for details.

There is no simple way to redirect standard IO and standard error messages to and from IFS files.

Workarounds

- To redirect *to* an IFS file you must first redirect to an HFS file and then copy the HFS file to an IFS file.
- To redirect *from* an IFS file you must first copy from an IFS file to an HFS file, and then redirect from the HFS file.

Examples

- For example, to redirect `stdout` of an ILE/C program called `prog` to an IFS file called `/home/dougd/out`, enter the following.

```
ovrdbf file(stdout) tofile (tmplib/tmpfile) +
   mbr (stdout)
   call prog
```

```
cpytostmf frommbr +
     ('/qsys.lib/tmplib.lib/tmpfile.file/stdout.mbr') +
      tostmf ('/home/dougd/out') stmfopt (*replace)
```

Development Environment

The make, lex, and yacc utilities are not available on the AS/400. The native character set is EBCDIC, not ASCII.

Two BEA TUXEDO commands, buildclient and buildserver, have been renamed buildclt and buildsrvr, respectively, in order to conform to the 10-character maximum.

The buildclt and buildsrvr commands accept input only for module files (*MODULE) and service program files (*SRVPGM), and produce output program files (*PGM). The ILE/C modules must be produced by crtcmod. Likewise, ILE/COBOL modules must be produced by crtcblmod.

The following example command lines show how to compile a C source file in member cx of a file called tuxsrc in a library called apps. The object module is created in a file called cx in a library called apps.

```
crtcmod module(apps/cx) srcfile(apps/tuxsrc) srcmbr(cx) +
sysifcopt(*ifsio) dbgview(*all) define('DEF=1')
```

Note that the special system interface option sysifcopt(*ifsio) is specified here. This option allows the C runtime library to access IFS files. Because the BEA TUXEDO system is compiled with this option, we *strongly* recommend using it for compatibility.

Note also that debugging has been turned on by dbgview(*all) and the preprocessor defined for DEF has been set to 1. This is equivalent to specifying -DDEF=1 -g on a cc command line (on a UNIX platform).

Currently you cannot specify a C source file on the buildclt or buildsrvr command line.

The -f and -l options to the buildclt and buildsrvr commands also mean something slightly different from the same options to the equivalent commands (buildclient and buildserver) on a UNIX system.

On a UNIX platform, the `-f` (*first*) option allows you to specify a list of items to be passed to the `cc` command such that the objects produced by running `cc` on these items are linked *before* the BEA TUXEDO libraries are linked.

The `-l` (*last*) option allows you to specify a list of items to be passed to the `cc` command such that the objects produced by running `cc` on these items are linked *after* the BEA TUXEDO libraries are linked.

On an AS/400 platform the `-f` option accepts object modules (that is, files of type `*MODULE`) only, and the `-l` option accepts service programs (that is, files of type `*SVRPGM`) only.

buildclt Example

Consider the following example command line.

```
buildclt ('-fapps/cx' '-fapps/cy' '-oapps/client')
```

This command accomplishes two tasks:

- It links two objects (`cx` and `cy`) in an HFS library called `apps`.
- It produces a client executable program called `client` in the `apps` library.

buildsrvr Example

Consider the following example command line:

```
buildsrvr ('-fapps/sx' '-fapps/sy' '-oapps/server' '-sSVC')
```

This command accomplishes two tasks:

- It links objects sx and sy in an HFS library (apps).
- It produces a server executable program (server) in a library (apps) which offers a service (SVC).

If the -k option of buildsrvr is used, then the server main code is generated in member BS*pid* of the tuxsrc file in the qtemp library, where the value of *pid* is the process ID of the buildsrvr process. Because the qtemp library is removed at the end of each job, you must make a copy of it in a permanent directory before signing off, if you want to keep it.

Both mkfldhdr and viewc generate header files in IFS, usually in the current directory. You must copy these .h files into an HFS h file so the C compiler will be able to find them.

viewc also generates the binary output file with a .VV (instead of a .V) suffix.

Runtime Environment

The BEA TUXEDO system is distributed with a set of tools for manipulating the environment: `getenv`, `putenv`, `setenv`, and `env`.

Table 2-1 Command Syntax for Runtime Tools

Syntax	Function
`getenv` *`var`*	Displays the value of the specified environment variable (*`var`*)
`putenv` *`'var=value'`*	Sets the specified environment variable (*`var`*) to the specified value (*`value`*)
`setenv` *`file`*	Calls `putenv` on every line of the specified file (*`file`*)
`env`	Displays the values of all variables in your environment

V4R1 provides a single command, `wrkenvvar`, that offers functionality similar to that available through `getenv`, `putenv`, `setenv`, and `env`. The `wrkenvvar` command differs from the BEA TUXEDO commands, however, in several aspects. First, unlike the distributed tools delivered with the BEA TUXEDO system, `wrkenvvar` cannot handle the expansion of variable names marked with a `$` prefix (similar to shell variable expansion).

In addition, the `hostname` and `id` utilities are distributed only with the BEA TUXEDO system:

- The `hostname` command prints the name of the network node for the AS/400. The machine name used in the `UBBCONFIG` configuration file must match the name returned by `hostname`.
- The `id` command prints the `UID` and `GID` of the current user profile. The `UID` and `GID` values specified in the `UBBCONFIG` configuration file must match the values returned by the `id` command. (See Chapter 3, “Configuration for BEA TUXEDO System Administration.”)

All BEA TUXEDO executables and header files are packaged in an HFS library (see `tuxlib` during installation). Other data files (such as `udataobj/* and locale/C/*`) are placed in `$TUXDIR` in IFS. `$TUXDIR/bin` has symbolic links to HFS executables. `$APPDIR` should also reside in IFS. You must add the HFS library to the library list before invoking any BEA TUXEDO commands.

Sample Application

Let's use `simpapp` as an example. Assume *`tuxdir`* is installed in `/QOpenSys/tuxedo`, *`tuxlib`* is installed as `tuxedo`, and `APPDIR` is set to `/home/apps/simpapp`.

Begin by building the `simpcl` and `simpserv` programs (in Steps 1-9).

1. Create an HFS library called `apps`.

   ```
   crtlib apps
   ```

2. Create a physical source file called `tuxsrc`.

   ```
   crtsrcpf apps/tuxsrc rcdlen (120)
   ```

3. Copy `simpcl.c` from `/qopensys/tuxedo/apps/simpapp` to member `simpcl` or `tuxsrc`.

```
cpyfrmstmf fromstmf('/qopensys/tuxedo/apps/simpapp/simpcl.mbr') +
       tombr('/qsys.lib/apps.lib/tuxsrc.file/simpcl.mbr') +
        mbropt(*replace)
```

4. Generate a client object module by running the ILE/C compiler.

   ```
   crtcmod module(apps/simpcl) srcfile(apps/tuxsrc) +
        srcmbr(simpcl) sysifcopt(*ifsio)
   ```

5. Generate a client program by running `buildclt`.

   ```
   call buildclt ('-fapps/simpcl' '-oapps/simpcl')
   ```

6. Copy `simpserv.c` from `/qopensys/tuxedo/apps/simpapp` to member `simpserv` of `tuxsrc`.

```
cpyfrmstmf fromstmf('/qopensys/tuxedo/apps/simpapp/simpserv.c') +
     tombr('/qsys.lib/apps.lib/tuxsrc.file/simpserv.mbr') +
      mbropt(*replace)
```

7. Generate a server object module by running the ILE/C compiler.

   ```
   crtcmod module(apps/simpserv) srcfile(apps/tuxsrc) +
        srcmbr(simpserv) sysifcopt(*ifsio)
   ```

8. Generate a server program by running the `buildsrvr` command.

   ```
   call buildsrvr ('-sTOUPPER' '-fapps/simpserv' '-oapps/simpserv')
   ```

9. Create a symbolic link from `/home/apps/simpapp/simpserv to apps/simpserv`.

   ```
   addlnk obj('/qsys.lib/apps.lib/simpserv.pgm') +
        newlink('/home/apps/simpapp/simpserv')
   ```

10. Add *`tuxlib`* to the library list.

    ```
    addlible tuxedo
    ```

11. Set the necessary environment variables as follows.

    ```
    call putenv 'TUXDIR=/QOpenSys/tuxedo'
    call putenv 'APPDIR=/home/apps/simpapp'
    ```

12. Create the `TUXCONFIG` file from the `UBBCONFIG` configuration file `UBBCONFIG`.

    ```
    call putenv 'TUXCONFIG=$APPDIR/tuxconfig'
    call tmloadcf ('-y' 'UBB')
    ```

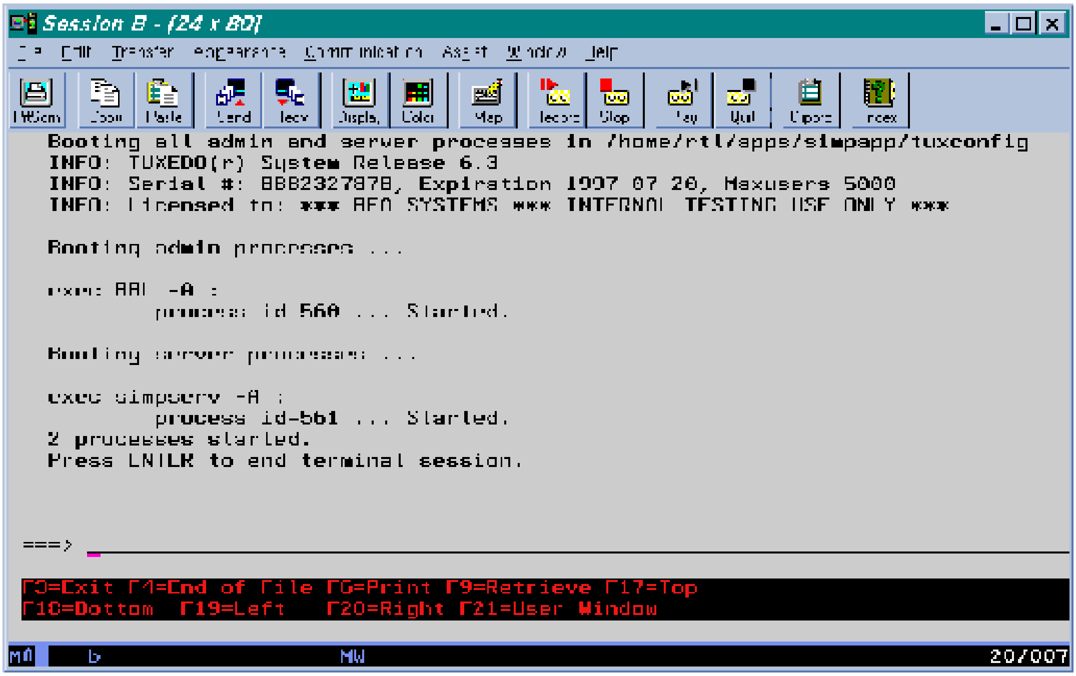
13. Run `wrklnk` to verify that the file `tuxconfig` has been created in `/home/apps/simpapp`.

14. Boot the application.

    ```
    call tmboot '-y'
    ```

    Messages similar to those shown in Figure 2-2 are displayed.

Figure 2-2 Messages Displayed at Boot Time


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Index
MA b MW 20/007

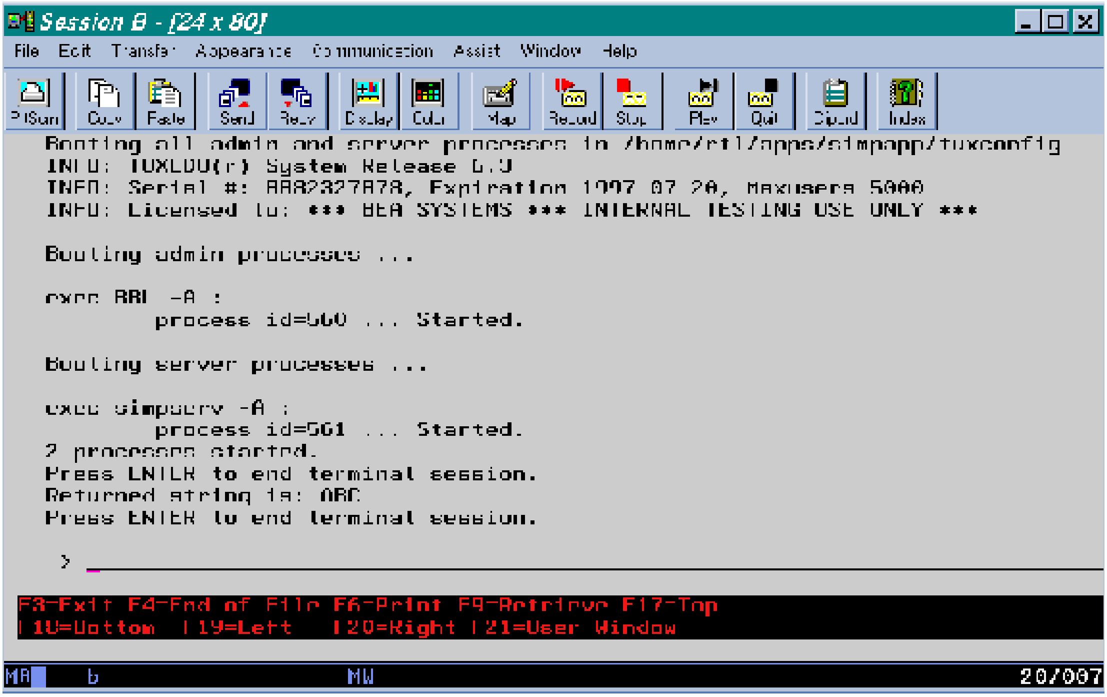
```
    Booting all admin and server processes in /home/rtl/apps/simpapp/tuxconfig
    INFO: TUXEDO(r) System Release 6.3
    INFO: Serial #: 8882327878, Expiration 1997 07 20, Maxusers 5000
    INFO: Licensed to: *** BEA SYSTEMS *** INTERNAL TESTING USE ONLY ***

    Booting admin processes ...

    exec BBL -A :
            process id 560 ... Started.

    Booting server processes ...

    exec simpserv -A :
            process id=561 ... Started.
    2 processes started.
    Press ENTER to end terminal session.



 ===> _

 F3=Exit F4=End of File F6=Print F9=Retrieve F17=Top
 F18=Bottom  F19=Left   F20=Right F21=User Window
```

15. Now run the client by entering a command such as the following.

    ```
    call apps/simpcl 'abc'
    ```

    Figure 2-3 shows the messages that are displayed.

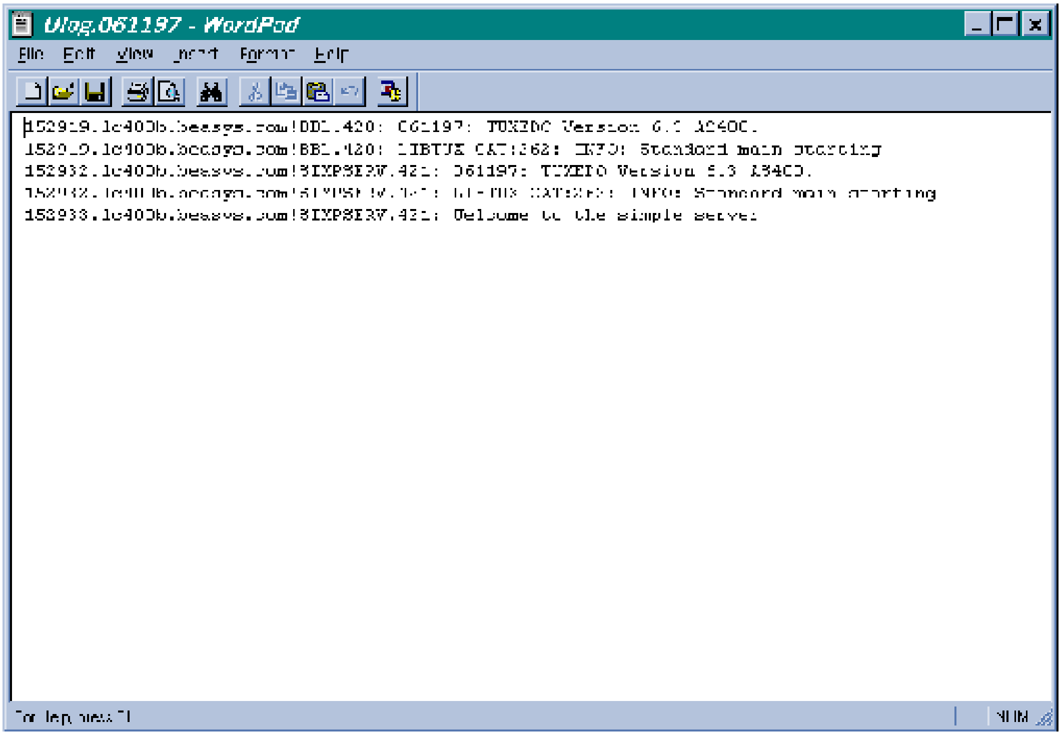
Figure 2-3 Messages Displayed after the Client Is Called


Session B - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Index

```
Booting all admin and server processes in /home/rtl/apps/simpapp/tuxconfig
INFO: TUXEDO(r) System Release 6.3
INFO: Serial #: AAA2327A78, Expiration 1997 07 20, Maxusers 5000
INFO: Licensed to: *** BEA SYSTEMS *** INTERNAL TESTING USE ONLY ***

Booting admin processes ...

exec BBL -A :
        process id=560 ... Started.

Booting server processes ...

exec simpserv -A :
        process id=561 ... Started.
2 processes started.
Press ENTER to end terminal session.
Returned string is: ABC
Press ENTER to end terminal session.

 > _

F3=Exit F4=End of File F6=Print F9=Retrieve F17=Top
F18=Bottom  F19=Left   F20=Right F21=User Window
```

16. To view the userlog file (which was created in `$APPDIR` in IFS) on a Win95 PC, use the Windows Explorer utility to map the `AS/400/home/apps` IFS directory to a network drive (such as F). Then use WordPad to open the userlog file in that drive, as shown in Figure 2-4.

Figure 2-4 Viewing the Userlog File with WordPad



```
152919.lc400b.beasys.com!BBL.420: 061197: TUXEDO Version 6.3 AS400.
152919.lc400b.beasys.com!BBL.420: LIBTUX CAT:262: INFO: Standard main starting
152932.lc400b.beasys.com!SIMPSERV.421: 061197: TUXEDO Version 6.3 AS400.
152932.lc400b.beasys.com!SIMPSERV.421: LIBTUX CAT:262: INFO: Standard main starting
152933.lc400b.beasys.com!SIMPSERV.421: Welcome to the simple server
```

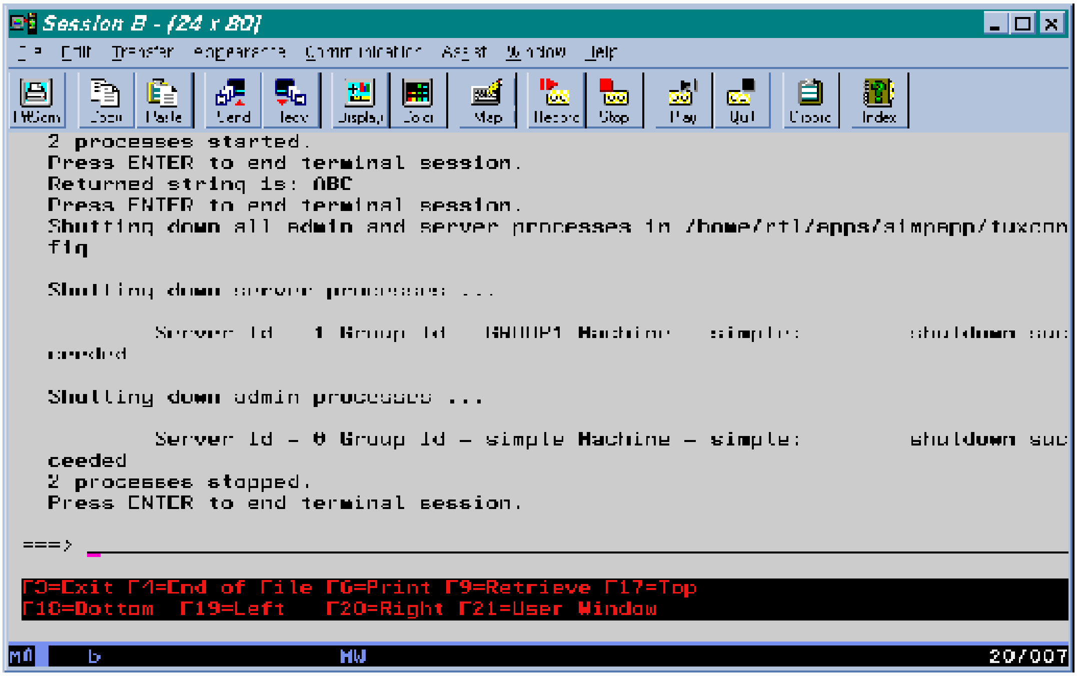
Shutting Down the Application

To shut down an application, enter the following command.

```
call tmshutdown '-y'
```

Figure 2-5 shows the messages displayed during the shutdown process.

Figure 2-5 Messages Displayed during Shutdown



```
    2 processes started.
    Press ENTER to end terminal session.
    Returned string is: ABC
    Press ENTER to end terminal session.
    Shutting down all admin and server processes in /home/rtl/apps/simpapp/tuxcon
    fig

    Shutting down server processes ...

            Server Id - 1 Group Id - GROUP1 Machine - simple:        shutdown suc
    ceeded

    Shutting down admin processes ...

            Server Id - 0 Group Id - simple Machine - simple:        shutdown suc
    ceeded
    2 processes stopped.
    Press ENTER to end terminal session.

 ===>

 F3=Exit F4=End of File F6=Print F9=Retrieve F17=Top
 F18=Bottom  F19=Left   F20=Right F21=User Window
```

IPC Resources

IBM ships a set of IPC tools as a PTF. These tools provide the functionality available on a UNIX system through the ipcs and ipcrm commands. There are some differences, however, the IBM tools have different names and produce output in a different manner.

To view the IPC resources consumed by all your applications, enter the following command.

```
call qp0zipcs
```

The output of qp0zipcs is generated in *QPRINT which means that it can be viewed as a spool file with the wrksplf command.

To remove a particular IPC resource, such as a message queue ID (*qid*), enter the following command.

```
call qp0zipcr ('-qqid')
```

When an error occurs while you are running qp0zipcr, a message is generated in the joblog. Use the dspjoblog command to view it.

Configuration Issues

- No more than one version of the BEA TUXEDO system can be installed on one machine.
- Server names are case-sensitive if $APPDIR is in /QOpenSys.
- Servers must reside in IFS under $APPDIR. These IFS files must be symbolic links to the HFS *PGM files. (Use the addlnk command to create a symbolic link.) The IFS name must be identical to the HFS name; thus each IFS name is restricted to 10 characters.
- The machine name must be a string returned by the hostname command.
- The tlisten command does not start itself in the background. To submit a batch job with tlisten, use the sbmjob command.
- Because a TCP/IP socket is used, no BRIDGE device needs to be specified.

- The RCMD parameter to be executed on a server when the latter is restarted must refer to a symbolic link in $PATH, typically $APPDIR. This symbolic link must point to an executable program in HFS. UNIX shell scripts will not work here.
- Due to the difference between the collating sequences in EBCDIC and ASCII, ranges specified for data-dependent routing may not behave the same way on AS/400 and UNIX platforms.
- On an AS/400 platform, a superuser (QSECOFR) cannot perform BEA TUXEDO administrative tasks. For example, you cannot boot an application while signed on as QSECOFR. Therefore BEA TUXEDO administrators must not sign on as QSECOFR.

/Q

The threshold command parameter specified on the qmadmin qcreate command line (such as the server RCMD), must refer to a symbolic link to an executable program in HFS. The link must be included in $PATH; typically it is found in $APPDIR.

UNIX shell scripts will not work here.

TxRPC

By default, the tidl compiler does not run the C preprocessor. This means that if you have #define statements or any other pre-processor constructs in the input file, then you will need to specify the -cpp_cmd option:

```
call tidl ('-cpp_cmd' 'crtcmod' . . . )
```

By default, the tidl command also produces server and client stub files in the current IFS directory. The stub files are named x_cstub.c and x_sstub.c. It is also possible to generate stub files in HFS by using the -out option to specify a library in the directory /qsys.lib, as in the following example:

```
call tidl ('-out' '/qsys.lib/tuxedo.lib' . . . )
```

In this case, the stub files will be generated in the physical file `idl` with members named `x_cs` and `x_ss` for client and server stubs, respectively. The `tidl` command also generates a header file in the current working directory of IFS. For example, if the `idl` compiler is run on input file `x.idl` as follows:

```
call tidl ('-cpp_cmd' 'crtcmod' '-keep' 'c_source' x.idl)
```

then `tidl` will generate the following three files in the current working directory of IFS:

- `x_cstub.c`
- `x_sstub.c`
- `x.h`

In addition, the `tidl` compiler will copy the header file to member `x` belonging to file `h` in the `qtemp` library.

If the `-keep` option is not specified, then the C compiler will be invoked on the stub files to produce object modules `x_cs` and `x_ss` in library `qtemp`. (`qtemp` is the default output library which can be overridden by the `-out` option, as described earlier.)

/COBOL

The BEA TUXEDO system supports clients and servers written in the ILE/COBOL language. The major differences between MicroFocus COBOL and ILE/COBOL are listed below.

- The ILE/COBOL compiler does not allow tabs in source code; use spaces whenever possible.
- The COMP-5 data type is COMP-4 in ILE/COBOL.
- The command-line interface for ILE/COBOL must use the linkage section rather than the `ACCEPT` statement.
- The method of redirecting output to a file is different. Consult the *COBOL Programmer's Guide* and COBOL reference manuals for details. Two applications shipped with the BEA TUXEDO system (`CSIMPAPP` and `STOCKAPP`) also show how some tasks are performed on an AS/400 platform.

Web GUI

With Release 6.5, you can administer a BEA TUXEDO application through any Web browser (such as Netscape or Internet Explorer) by using the Web-based GUI for administration. This section provides some tips for running the Web GUI on an AS/400 platform.

Two programs, `tuxwsvr` and `wlisten`, need to be run in the background. This can be accomplished by running a CL script using `sbmdbjob`. For example, the following CL script could be used to start `tuxwsvr`.

```
//BCHJOB JOB(WEBSERVER) JOBD(TUXADM/JOBD)
addlible tuxlib
cd '/home/tuxadm/web'
call setenv env
call tuxwsvr ('-l//lc400b.beasys.com:3184' '-iwinput.ini')
```

The first line should begin with //`BCHJOB` which allows a number of parameters to be set. (See the *CL Programming Manual*.) The `JOB` parameter, for example, sets the name of the job such that it appears on a `WRKACTJOB` screen. A customized Job Description is needed here also so that the job can be run in batch. The default Job Description for interactive sessions (`QDFTJOBD`) cannot be used here. To create a new one, use `WRKJOBD` to copy `QDFTJOBD` to `TUXADM/JOBD`, and then add a `USER` name to `TUXADM/JOB`.

Several environment variables, such as `TUXDIR`, also need to be set. To set an environment variable, list all necessary environment settings in an `env` file and then run the `setenv` command, specifying the `env` file on the command line. Set any other application-specific variables (such as `APPDIR`) as well.

One additional input file that `tuxwsvr` expects is an initialization or `ini` file. The `ini` file tells the server where to find HTML pages and Java images. For example:

```
CGI /cgi-bin tuxdir/bin
HTML /java/images tuxdir/web/gui/java/bea/tuxadm/images
JAVA /java tuxdir/web/gui/java
HTML /docs tuxdir/docs
HTML / tuxdir/udataobj/webgui
```

For example, suppose the BEA TUXEDO system has been installed in `/qopensys/tuxr65` and you are now creating an ini file called `winput.ini`. If you substitute the root directory of the BEA TUXEDO system for the word `tuxdir` in the lines shown above, your `ini` file will look like the following.

```
CGI /cgi-bin /qopensys/tuxr65/bin
HTML /java/images /qopensys/tuxr65/web/gui/java/bea/tuxadm/images
JAVA /java /qopensys/tuxr65/web/gui/java
HTML /docs /qopensys/tuxr65/docs
HTML / /qopensys/tuxr65/udataobj/webgui
```

For more detailed information on creating an `ini` file for `tuxwsvr`, see `tuxwsvr`(1) in the *BEA TUXEDO Reference Manual*.

Current implementation also requires that for all user CGI programs, the HFS library in which they reside be added to the library list.

The procedure for setting up a batch job with `wlisten` is similar to the one you follow to set up a batch job for `tuxwsvr`.

All `wlisten` options are taken from an ini file that is specified with the `-i` option. This ini file must be created; *IT IS NOT GENERATED WHEN YOU INSTALL BEA TUXEDO ON THE AS/400*. For detailed information on creating an `ini` file for `wlisten`, see the `wlisten`(1) in the *BEA TUXEDO Reference Manual*. The following is an example.

```
TUXDIR=/qopensys/tuxr65
NADDR=0x00020c71c604a68d
CODEBASE=/jav
DOCBASE=http://lc400b:3184/docs
```

To start the Web GUI for BEA TUXEDO administration on the AS/400, enter the following URL.

```
http://machine_name:port/cgi-bin/tuxadm?TUXDIR=tuxdir&INIFILE=ini_file
```

where:

- *`machine_name:port`* is the AS/400 machine/port combination on which `tuxwsvr` is running.
- *`tuxdir`* is the directory in which the BEA TUXEDO system was installed (as specified on the command line during the installation procedure).
- *`ini_file`* refers to the initialization file for `wlisten`.

For additional information on preparing to use the Web GUI, refer to the *BEA w Installation Guide*.

CHAPTER

3 Configuration for BEA TUXEDO System Administration

Creating a Profile for a BEA TUXEDO Administrative Group

To set the group ID correctly for members of an administrative group, you must set up a group profile. A group profile is the same as a user profile except that the group ID number is set to a numeric value. Each user's profile can then refer (in the *group profile* field) to the group profile name.

For example, consider the user and group profiles for an administrator named *Joe* with a login ID of `joe`:

- In the user profile for user `joe`:
    - The group profile field (`GRPPRF`) is set to `tuxadm`.
    - The user ID number (`UID`) field is set to `6494`.
- In the group profile for group `tuxadm`, the group ID number (`GID`) is set to `601`.

  If user `joe` runs the `id` utility as follows.

  ```
  call id
  ```

then the following output is displayed on the screen.

```
uid = 6494, euid = 6494, gid = 601, egid = 601
```

User profiles and group profiles can be set up with the CRTUSRPRF command.

Granting Special Authority

The user profile for an administrator must include *JOBCTL special authority. This special authorization allows the BEA TUXEDO system to bypass some security checks when it is trying to verify the state of the following:

- System processes within the BEA TUXEDO system, and
- Clients being run under a user profile other than that of the administrator.

Without this authorization, the BEA TUXEDO system fails to do the following:

- Restart servers and
- Clean up after clients should they fail to detach from an application when they exit or when they end unexpectedly.

You can add this authorization to your user profile by adding *JOBCTL to the Special Authority (SPCAUT) list. To do so, complete the following tasks:

- Sign on as QSECOFR.
- Run the chgusrprf command.

Setting the Correct Time Offset

In order for the Web GUI administration tool to work properly, the system variable QUTCOFFSET must be properly set. Use the wrksysval command to specify the offset (in hours and minutes) between the current system time and the Coordinated Universal Time (UTC), also known as Greenwich mean time. For example, in the Eastern Standard Time time zone, QUTCOFFSET should be set to +4 hours.