Oracle Tuxedo for IBM i 6.1

Users Guide

11*g* Release 1 (11.1.1.2.0)

August 2010

Oracle Tuxedo IBM i 6.1 Users Guide, 11*g* Release 1 (11.1.1.2.0)

TxRPC 14
/COBOL 14

Installing the Oracle Tuxedo System on IBM i 6.1

This chapter contains the following topics:

- Interoperability
- The Installation Process

Interoperability

Oracle Tuxedo 11g Release 1 (11.1.1.2.0) running on an IBM i 6.1 platform can interoperate, as either master or non-master, with 11g Release 1 (11.1.1.2.0) running on any other supported platform. Backward compatibility (that is, interoperability between 11g Release 1 (11.1.1.2.0) and prior releases) is supported only if the master node is running 11g Release 1 (11.1.1.2.0) .

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

Note: If remote domain is Oracle Tuxedo 6.5 on other OS, local domain on IBM i 6.1 must set environment variable whose name is `<remote domainid>`ISOS400 to "`n`".

The Installation Process

This section provides instructions for preparing your environment for the Oracle Tuxedo system, installing the Oracle Tuxedo system from a Microsoft Windows or UNIX system platform.

Before You Begin

Before you start installing the Oracle Tuxedo system, sign on to your IBM i 6.1 system and make sure you have a `tuxedo` user profile for Oracle Tuxedo installation. 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 a Microsoft Windows Platform

1. Oracle Tuxedo 11g Release 1 (11.1.1.2.0) for IBM i 6.1 is distributed in a .ZIP file. Before installation, extract this file to any directory and then start the next installation.

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

   ```
   open IBM i 6.1-machine-name
   account-name
   password
   ```

   For example, suppose that (a) you are working on an IBM i 6.1 machine called lc400b.oracle.com, (b) your login is `rtl`, and (c) your password is `rtl123`. Your profile contains the following three lines:

   ```
   open lc400b.oracle.com
   rtl
   rtl123
   ```

   This file is used to automatically execute the pre-installation procedures necessary for installing Oracle Tuxedo 11g Release 1 (11.1.1.2.0) through ftp. Once the pre-installation steps are completed, the file should be removed.

   Note: A new line is required after the password line.

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

   ```
   preinst.bat pkgdir tmplib tuxdir tuxlib filename
   ```

   where:

   - `pkgdir` is the extracted ZIP file directory.
   - `tmplib` is the library on the IBM i 6.1 machine in which copies of the Oracle Tuxedo save files will be stored.
   - `tuxdir` is the IFS file system on the IBM i 6.1 machine on which the Oracle Tuxedo system will be installed.

- `tuxlib` is the library on the IBM i 6.1 machine where the saved files are restored.
- `filename` is the full path to a file that contains ftp account and password information for the IBM i 6.1 machine on which the Oracle Tuxedo system is installed.

4. Run the install program on the IBM i 6.1 machine.

   ```
   addlible tuxlib

   call coreinst (tmplib tuxlib '/tuxdir'

   'full|server|client|joltclt|atmiclt|jrly' 'y|n')
   ```

   where you specify one of the following Oracle Tuxedo installation component packages:

   - full (to install Oracle Tuxedo installation component packages)
   - server (to install all Oracle Tuxedo server package)
   - client (to install only the Oracle Tuxedo all client packages)
   - joltclt (to install only the Oracle Tuxedo Jolt client)
   - atmiclt (to install only the Oracle Tuxedo ATMI client)
   - jrly (to install only the Oracle Tuxedo Jolt JRLY)

   Specify one of the following for sample's installation:

   - 'y' (to install samples when package type is full or server.
   - 'n' (do not install samples)

   Note: The words full, server, client atmiclt, joltclt and jrly, y and n 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 all the saved files that were created in Step 3 from `tmplib`.

   Note: 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 non-null password is entered. For further details about the password file, see tlisten in the Oracle Tuxedo Reference Manual.

7. Before using SSL, enter the following command to configure SSL:
   `addlible tuxlib`

   Note: `tuxlib`: The library on the IBM i 6.1 where the save files are restored.

   ```
   call sslconf ('tuxdir' 'hostname' 'portid' 'baseobj' 'filename')
   ```

   `tuxdir`: The IFS file system (on the IBM i 6.1) where Oracle Tuxedo is installed.

   `hostname`: The URL of the LDAP server system.

   `portid`: A port number for the URL of the LDAP server system.

   `baseobj`: A base object for search in LDAP server.

   `filename`: The name of LDAP filter file, if the input is '' (null), the default value of "`$tuxdir/udataobj/security/bea_ldap_filter.dat`" will be used.

8. Your installation of the Oracle Tuxedo system is now complete.

Installing from a UNIX System Platform

1. Oracle Tuxedo 11g Release 1 (11.1.1.2.0) for IBM i 6.1 is distributed in a .ZIP file. Before installation, extract this file to any directory and then start the next installation.

2. Set up a `.netrc` file to allow ftp to sign on to the IBM i 6.1 machine. For example, suppose that (a) you are working on an IBM i 6.1 machine called lc400b.oracle.com, (b) your login is `rtl`, and (c) your password is `rtl123`. Your `.netrc` file will contain the following three lines.

   ```
   machine lc400b.oracle.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`, where:

   - `mount-point` is the extracted ZIP file directory.
   - `tmplib` is the library on the IBM i 6.1 where Oracle Tuxedo save files are located.
   - `tuxdir` is the IFS file system (on the IBM i 6.1) where Oracle Tuxedo is installed.
   - `tuxlib` is the library on the IBM i 6.1 where the save files are restored.
   - `machine` is the name of the IBM i 6.1 machine where Oracle Tuxedo is installed.

4. Run the install program on the IBM i 6.1 machine.

   ```
   addlible tuxlib
   ```

   ```
   call coreinst (tmplib tuxlib '/tuxdir'
   'full|server|client|joltclt|atmiclt|jrly' 'y|n')
   ```

   where you specify one of the following Oracle Tuxedo installation component packages:

   - full (to install Oracle Tuxedo installation component packages)
   - server (to install all Oracle Tuxedo server package)
   - client (to install only the Oracle Tuxedo all client packages)
   - joltclt (to install only the Oracle Tuxedo Jolt client)
   - atmiclt (to install only the Oracle Tuxedo ATMI client)
   - jrly (to install only the Oracle Tuxedo Jolt JRLY)

   Specify one of the following for sample's installation:

   - '`y`' (to install samples when package type is full or server.
   - '`n`' (do not install samples)

   Note: The words full, server, client atmiclt, joltclt and jrly, y and n 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 all the saved files that were created in Step 3 from `tmplib`.

   Note: 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 non-null password is entered. For further details about the password file, see tlisten in the Oracle Tuxedo Reference Manual.

7. Before using SSL, enter the following command to configure SSL:

   ```
   addlible tuxlib
   ```

`tuxlib`: The library on the IBM i 6.1 where the save files are restored.

```
call sslconf ('tuxdir' 'hostname' 'portid' 'baseobj' 'filename')
```

`tuxdir`: The IFS file system (on the IBM i 6.1) where Oracle Tuxedo is installed.

`hostname`: The URL of the LDAP server system.

`portid`: A port number for the URL of the LDAP server system.

`baseobj`: A base object for search in LDAP server.

`filename`: The name of LDAP filter file, if the input is '' (null), the default value of "`$tuxdir/udataobj/security/bea_ldap_filter.dat`" will be used.

8. Your installation of the Oracle Tuxedo system is now complete.

Configuration for Oracle Tuxedo System Administration

This chapter contains the following topics:

- Creating a Profile for a Oracle Tuxedo Administrative Group
- Granting Special Authority
- Setting the Correct Time Offset

Creating a Profile for a Oracle 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 Oracle Tuxedo system to bypass some security checks when it is trying to verify the state of the following:

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

Without this authorization, the Oracle 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 Oracle Tuxedo Administration Console 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.

CHAPTER 3

Using the Oracle Tuxedo System on UNIX and IBM i 6.1 Platforms

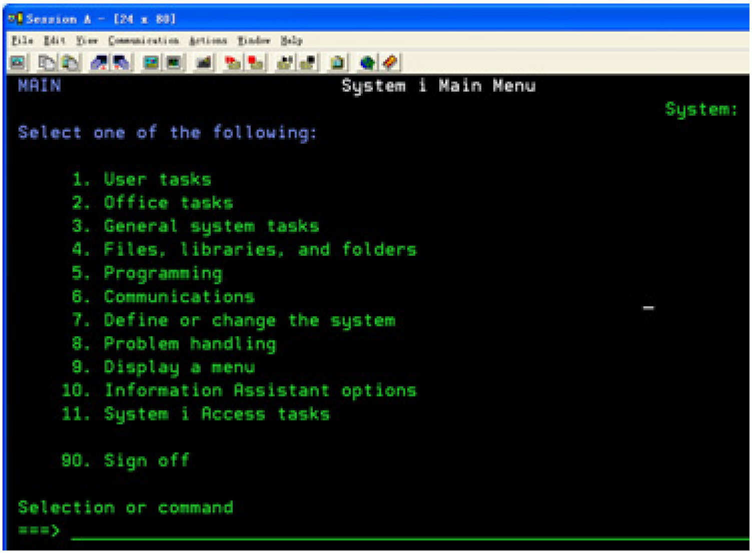
This chapter contains the following topics:

- User Interface and IBM i 6.1 Concepts
- Runtime Environment

User Interface and IBM i 6.1 Concepts

The user interface on an IBM i 6.1 system is completely menu-driven. The interface makes extensive use of function keys F1-F24.

Figure 3-1 Main Menu of Tasks for IBM i 6.1 Platform



```
MAIN                          System i Main Menu
                                                                   System:
Select one of the following:

     1. User tasks
     2. Office tasks
     3. General system tasks
     4. Files, libraries, and folders
     5. Programming
     6. Communications
     7. Define or change the system
     8. Problem handling
     9. Display a menu
    10. Information Assistant options
    11. System i Access tasks

    90. Sign off

Selection or command
===> ____________________________________________________________
```

Invoking Programs from the Command Line

Virtually every action performed via the menu has a command-line equivalent. Application executables, such as Oracle 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 IBM i 6.1 equivalent of a DLL or shared object) is needed, then the library list is searched again. The library list can be displayed with the `dsplib` 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 IBM i 6.1 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.

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)
```

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 text shown in Listing 3-1.

Listing 3-1 Redirection Example

```
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 IBM i 6.1. The native character set is EBCDIC, not ASCII.

Two Oracle 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), service program files (*SRVPGM), binding directory (*BNDDIR), 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 Oracle 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 Oracle 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 Oracle Tuxedo libraries are linked.

On an IBM i 6.1 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.

The -B option allows you to specify a binding directory (*BNDDIR).

buildclt Example

Consider the following example command line.

```
call 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:

```
call 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 BSpid 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.

The mkfldhdr/ mkfldhdr32 -H option creates a header file in HFS.

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

tmconfig

On the IBM i 6.1, the valid EDITOR environment variable values for tmconfig command are "STRSEU" and "EDTF". If editing lines longer than 240 characters is required, please use EDTF.

epifreg

For the epifreg command -f option, the library name and service program name must be upper cased as shown in the following example:

```
call epifreg ('-r' '-p qa/fanatz1' '-i engine/security/authorization' '-o
SYSTEM' '-v 1.0' '-f AAA101/FANATZ1' '-e qa_atz_entry').
```

viewc, viewc32

There is a difference in IBM i 6.1 and UNIX. If the -C option is specified, one COBOL copy file is created for each VIEW defined in the viewfile. These copy files are created in root directory (/), not in current directory like UNIX, unless an alternate directory is specified through the -d option.

Xerces-C++ Parser

Because of the 10 characters limit, all include files for Xerces-C++ Parser are located only in IFS. So the source code that includes those header files should be copied to IFS, and set INCLUDE environment variable to $TUXDIR/include before compiles. For an introduce to programming with Xerces-C++ , see the Xerces-C++ Version 2.8.0 Programming Guide.

XML Encoding With EBCDIC Character

If the XML buffer is encoded as EBCDIC characters, the encoding attribute should be set to "EBCDIC-CP-US".

IBM i 6.1 Command Names

Because of the 10 characters limit some others Tuxedo commands are changed on IBM i 6.1.

Table 3-1 IBM i 6.1 Command Names

On Unix	On IBM i 6.1
tmloadrepos	tmldrep
tmunloadrepos	tmunldrep
TMS_rac_refresh	TMS_racrf
snmp_integrator	snmpintgrt

Table 3-1 IBM i 6.1 Command Names

`snmp_version`	`snmp_ver`
`snmpgetnext`	`snmpgetn`
`reinit_agent`	`rinitagent`

Runtime Environment

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

Table 3-2 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

IBM i 6.1 provides a single command, `wrkenvvar`, that offers functionality similar to that available through `getenv`, `putenv`, `setenv`, and `env`. The `wrkenvvar` command differs from the Oracle Tuxedo commands, however, in several aspects. First, unlike the distributed tools delivered with the Oracle 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 Oracle Tuxedo system:

- The `hostname` command prints the name of the network node for the IBM i 6.1. 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 1, "Configuration for Oracle Tuxedo System Administration,")

All Oracle 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 Oracle Tuxedo commands.

Sample Application

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/samples/atmi/simpapp` to `tuxsrc simpcl` member.

   ```
   cpyfrmstmf fromstmf('/qopensys/tuxedo/samples/atmi/simpapp/simpcl.c') +
   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/samples/atmi/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') +
newlnk('/home/apps/simpapp/simpserv')
```

10. Add `tuxlib` to the library list.

```
addlible tuxedo
```

11. Set the necessary environment variables as follows.

```
call putenv 'LANG=C’
call putenv 'TUXDIR=/QOpenSys/tuxedo'
call putenv 'APPDIR=/home/apps/simpapp'
call putenv  'PATH=/QOpenSys/tuxedo/bin:/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'
```

15. Run the client by entering a command such as the following.

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

16. View the userlog file (which was created in `$APPDIR` in IFS) on IBM i 6.1 with EDTF.

Shutting Down the Application

To shut down an application, enter the following command.

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

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

- 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, for example:

  ```
  Addlible tuxlib
  ```

  ```
  Addenvvar envvar('TUXDIR') value('/QOpenSys/tuxedo')
  ```

  ```
  SBMJOB CMD(CALL PGM(TLISTEN) PARM('-l' '//hostname:8899'))
  JOB(TLISTEN) CPYENVVAR(*YES).
  ```
- 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 IBM i 6.1 and UNIX platforms.
- DDR for decimal is not supported.
- On an IBM i 6.1 platform, a superuser (QSECOFR) cannot perform Oracle Tuxedo administrative tasks. For example, you cannot boot an application while signed on as QSECOFR. Therefore Oracle Tuxedo administrators must not sign on as QSECOFR.
- Effective blocktime values are rounded up to the nearest `SCANUNIT*2-1` value.
- The field type of ptr (a pointer to a buffer) is not supported in FML32.
- The `tpusr` file is required for `AUTHSVR`.
- The environment variable LANG should be set to C when the language environment is English (that is to say, `/QSYS.LIB/EN_US.LOCALE`). An easier way is to just set locale to C in the user profile.

/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.

Administration Console

With Oracle Tuxedo 11g Release 1 (11.1.1.2.0), you can administer an Oracle 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 Oracle Tuxedo Administration Console on an IBM i 6.1 platform.

Three files `$TUXDIR/udataobj/webgui/webguitop.html`, `$TUXDIR/udataobj/tuxwsvr.ini`,and `$TUXDIR/udataobj/webgui.ini` need to be modified. For example:

- Replace "`TUXDIRVALUE`" in file `$TUXDIR/udataobj/webgui/webguitop.htm`l with the value of `TUXDIR`.
- Replace "`http://HOST/cgi-bin/tuxadm`" in file `$TUXDIR/udataobj/webgui/webguitop.html with "/cgi-bin/tuxadm`".

- Replace "<TUXDIR>" in file $TUXDIR/udataobj/ tuxwsvr.ini with the value of TUXDIR.
- Replace "<TUXDIR>" in file $TUXDIR/udataobj/webgui/webgui.ini with the value of TUXDIR.
- Replace "<LOCAL_HOST>" in $TUXDIR/udataobj/webgui/webgui.ini with the host name.

Two programs, tuxwsvr and wlisten, need to be run in the background. This can be accomplished by using sbmjob like tlisten. For example:

```
addlible tuxlib
addenvvar envvar('TUXDIR') value('/QOpenSys/tuxedo')
SBMJOB CMD(CALL PGM(tuxwsvr) PARM('-l' '//hostname:8899' '-i'
'TUXDIR/udataobj/tuxwsvr.ini')) JOB(tuxwsvr) CPYENVVAR(*YES)
```

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 setting up a batch job for tuxwsvr or tlisten.

To start the Administration Console for Oracle Tuxedo administration on the IBM i 6.1, enter the following URL.

http://machine_name:port/webguitop.html (where machine_name:port is the IBM i 6.1 machine/port combination on which tuxwsvr is running).

For additional information on preparing to use the Oracle Tuxedo Administration Console, refer to Starting the Oracle Tuxedo Administration Console in the Oracle Tuxedo Installation Guide.

Jolt Relay

The combination of the Jolt Relay (JRLY) and its associated Jolt Relay Adapter (JRAD) is typically referred to as the Internet Relay. Jolt Relay routes messages from a Jolt client to a JSL or JSH. This eliminates the need for the JSH and Oracle Tuxedo to run on the same machine as the Web server (which is generally considered insecure).

The program JRLY need to be run background. This can be accomplished by using sbmjob or sbmdbjob like tlisten or tuxwsvr.

For additional information on Jolt Relay refer to Oracle Jolt.

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 Oracle 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 Oracle Tuxedo system (CSIMPAPP and STOCKAPP) also show how some tasks are performed on an IBM i 6.1 platform.

You can override the Oracle Tuxedo `TMDISPATCH9()` to `TMDISPATCH16()` for COBOL routines only when set environment variable `TM_COB_STATIC` to `YES` before create COBOL program.