

# Oracle® Tuxedo

Using Oracle Tuxedo ATMI on Windows

12c Release 2 (12.1.3)

April 2014

ORACLE®

Copyright © 1996, 2014, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

## 1. Using Oracle Tuxedo ATMI on Windows Server

Windows Considerations .....	1-1
Configuration Issues .....	1-2
Specifying Machine Type and User ID Numbers .....	1-2
Using Network Drives .....	1-3
Allocating and Releasing Memory Buffers .....	1-3
Using the bankapp Driver for a Template .....	1-3
Starting Oracle Tuxedo ATMI Applications Automatically .....	1-3
See Also .....	1-4

## 2. Configuring Oracle Tuxedo ATMI for Windows Server

Introducing the Oracle Administration Program .....	2-1
Invoking the Oracle Administration Program .....	2-2
Accessing Other Machines on a Network .....	2-4
Setting and Modifying Environment Variables .....	2-5
Directing Oracle Tuxedo Messages to the Windows Server Event Log .....	2-7
Log Output: Event Log .....	2-8
Setting Up the Event Log .....	2-8
Viewing Windows Server Event Log Entries .....	2-8
Log Output: User Log (ULOG) Disk File .....	2-10
Setting Up a ULOG .....	2-10
Viewing ULOG Entries .....	2-10

Configuring tlisten Processes to Start Automatically. . . . .	2-10
Configuring and Monitoring IPC Resources to Maximize System Performance . . . .	2-12
Reviewing the Windows Server Registry Content. . . . .	2-15
Developer Key . . . . .	2-16
Environment Key . . . . .	2-16
Security Key . . . . .	2-16

### 3. Using the Visual C++.Net IDE To Develop Oracle Tuxedo ATMI Applications

Before You Start. . . . .	3-1
Using Development Tool . . . . .	3-2
Using the buildserver and buildclient Commands . . . . .	3-2
Adding BuildTuxedo to the MSDEV Tools Menu . . . . .	3-4
Creating Oracle Tuxedo ATMI Project Files. . . . .	3-6
Setting Up Your Environment . . . . .	3-6
Specifying the Build Type, Header File, and Filename . . . . .	3-7
How BuildTuxedo Uses the Header File. . . . .	3-8
Specifying Function and Service Names . . . . .	3-9
Specifying a Resource Manager . . . . .	3-10
Debugging an Oracle Tuxedo ATMI Server Application . . . . .	3-11
Developing an ATMI Application Using the Command Line Instead of the Visual C++.Net IDE GUI . . . . .	3-12
See Also . . . . .	3-13

# Using Oracle Tuxedo ATMI on Windows Server

The following sections describe some basic differences *between* using Oracle Tuxedo ATMI on a UNIX server system *and* using Oracle Tuxedo ATMI on a Windows Server system:

- [Windows Considerations](#)
- [Configuration Issues](#)
- [Specifying Machine Type and User ID Numbers](#)
- [Using Network Drives](#)
- [Using the bankapp Configuration File as Template](#)
- [Starting Oracle Tuxedo ATMI Applications Automatically](#)

## Windows Considerations

Keep in mind the following general considerations when using Oracle Tuxedo ATMI on a Windows system:

- When you specify pathnames, use a back slash (\), not a forward slash (/) to delimit file and directory names. Use a drive letter (such as C:) for all fully qualified paths.
- You do not need to specify the .exe suffix for executable files. The suffix is always implied for Oracle Tuxedo ATMI for Windows.

- Filenames follow Windows naming conventions. For instance, names ending in `.dll` identify dynamically-linked libraries; names ending in `.lib` identify statically linked and imported libraries; names ending in `.cmd` identify command scripts.
- All Oracle Tuxedo system executables, command scripts, and dynamically-linked libraries are located in `%TUXDIR%\bin`. Statically-linked libraries are located in `%TUXDIR%\lib`.

## Configuration Issues

Keep in mind the following configuration issues when setting up the Oracle Tuxedo ATMI software on a Windows Server system:

- Server names are case sensitive.
- The names of Windows Server machines specified in Oracle Tuxedo ATMI application files should always be spelled in uppercase.
- The `OPENINFO` string in the Oracle Tuxedo configuration file (`UBBCONFIG`) must be presented in the following format:

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

For example:

```
OPENINFO="TUXEDO\SQL:APPDIR1\bankdl3;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` section of the configuration file, include the following three entries:

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

**Note:** These entries require different settings on a UNIX system.

Whenever you create a configuration file for an environment with both UNIX and Windows systems, include these entries in the `MACHINES` section for every Windows Server node in your configuration.

## Using Network Drives

For reliability purposes, Oracle recommends that you do not use network drives. If, however, you try to start an Oracle Tuxedo ATMI application on a Windows Server machine in which the TUXCONFIG file resides on a network drive, you must set the following permissions:

- 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, the `ENTRY` option 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.

## Allocating and Releasing Memory Buffers

When allocating and releasing memory buffers on a Windows Server system, make sure each memory buffer is released from the same heap in which it was allocated. If it is not, a segmentation fault occurs.

For example, a memory buffer that is allocated using `Falloc()` must be released using `Ffree()`. If a memory buffer allocated with `malloc()` is freed using `Ffree()`, a segmentation fault occurs. The `free()` routine must be used, in the latter case, to free the memory buffer.

For more information about `Falloc`, `Falloc32(3fml)` and `Ffree`, `Ffree32(3fml)`, see *Oracle Tuxedo ATMI FML Function Reference*. For more information about `malloc()` and `free()`, see the documentation delivered with your operating system.

## Using the bankapp Configuration File as Template

The `bankapp` program is a small example application bundled with Oracle Tuxedo ATMI for Windows Server. Besides demonstrating the operation of Oracle Tuxedo ATMI and providing an example of Oracle Tuxedo ATMI application code, the file of

`%TUXDIR%\samples\atmi\bankapp\nt\ubbshm` (in Oracle Tuxedo SHM mode) or

`%TUXDIR%\samples\atmi\bankapp\nt\ubbmp` (in Oracle Tuxedo MP mode) can act as a template for configurations for any new applications.

## Starting Oracle Tuxedo ATMI Applications Automatically

When Oracle Tuxedo ATMI is installed on a Windows Server system, it may be useful to configure the machine to start an Oracle Tuxedo ATMI application automatically when booting

up your system using the `srvany.exe` utility program provided in Microsoft Windows Resource Kits. For configuration instructions, see `srvany.wri` and `rkttools.hlp`.

To ensure proper operation of Oracle Tuxedo ATMI programs that start automatically when booting up, you must set the Oracle Tuxedo system environment variables `%TUXDIR%` and `%NLSPATH%`. Set these variables using the conventional Windows method or by using the Oracle Administration program, as explained in [“Setting and Modifying Environment Variables” on page 2-5](#).

## See Also

For information about installing Oracle Tuxedo ATMI on a Windows Server system, see *Installing the Oracle Tuxedo System*. For more information about the Oracle Tuxedo system, see the following documents:

- [Setting Up an Oracle Tuxedo Application](#)
- [Administering an Oracle Tuxedo Application at Run Time](#)
- [Using the Oracle Tuxedo ATMI Workstation Component](#)
- [Oracle Tuxedo Command Reference](#)
- [Oracle Tuxedo ATMI C Function Reference](#)
- [Oracle Tuxedo ATMI COBOL Function Reference](#)
- [Oracle Tuxedo ATMI FML Function Reference](#)
- [Oracle Tuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference](#)

# Configuring Oracle Tuxedo ATMI for Windows Server

The following sections describe how to configure Oracle Tuxedo ATMI on a Windows Server system:

- [Introducing the Oracle Administration Program](#)
- [Invoking the Oracle Administration Program](#)
- [Accessing Other Machines on a Network](#)
- [Setting and Modifying Environment Variables](#)
- [Directing Oracle Tuxedo Messages to the Windows Server Event Log](#)
- [Configuring tlisten Processes to Start Automatically](#)
- [Configuring and Monitoring IPC Resources to Maximize System Performance](#)
- [Reviewing the Windows Server Registry Content](#)

## Introducing the Oracle Administration Program

In addition to the Oracle Tuxedo Administration Console, the Oracle Tuxedo 7.1 or later software for Windows provides an Oracle Administration program and two Windows services (`tlisten` and Oracle `procMGR`) for configuring the Oracle Tuxedo system on a Windows Server system. Only if the installation includes Oracle Tuxedo server components will these additional administration tools be installed on the Windows Server system.

You can use the Oracle Administration Program to perform the following tasks:

- [Accessing Other Machines on a Network](#)
- [Setting and Modifying Environment Variables](#)
- [Directing Oracle Tuxedo Messages to the Windows Server Event Log](#)
- [Configuring tlisten Processes to Start Automatically](#)
- [Configuring and Monitoring IPC Resources to Maximize System Performance](#)

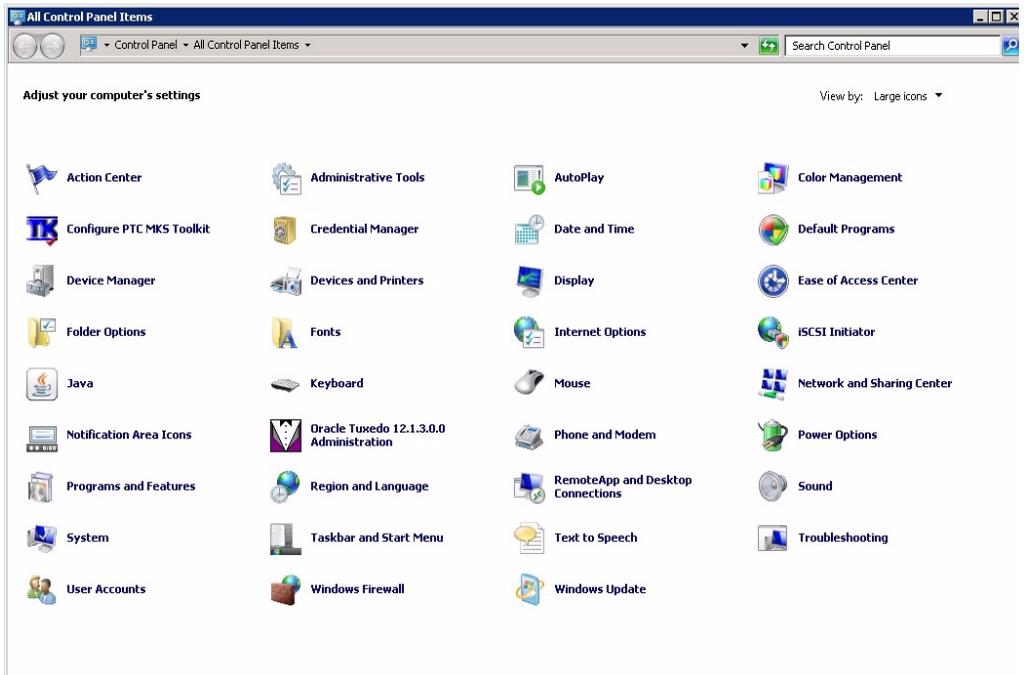
**Note:** The Oracle Tuxedo Administration Console offers extensive online help: instructions for all the administrative tasks that can be performed through the console, plus reference information for all configuration tool folders. For information about how to use the Oracle Tuxedo Administration Console, see [Oracle Tuxedo Administration Console Online Help](#).

## Invoking the Oracle Administration Program

With Oracle Tuxedo release 7.1 or later software installed on your Windows Server system, perform the following steps to access the Oracle Administration program:

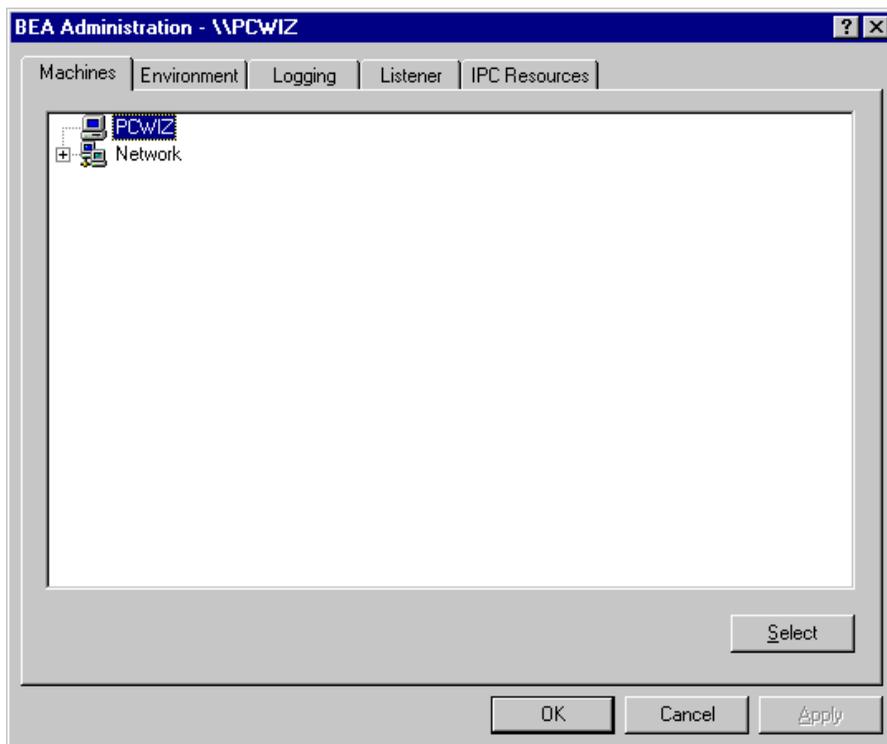
1. Choose Start → Settings → Control Panel to launch the Control Panel.

Figure 2-1 Microsoft Windows Control Panel



2. In the Control Panel, double-click the Oracle Administration icon to launch the Oracle Administration window.

**Figure 2-2 Oracle Administration Window with Machines Page Displayed**



## Accessing Other Machines on a Network

The Machines page of the Oracle Administration window enables you, as the Oracle Tuxedo system administrator, to access any machine (where you have login privileges) on the Microsoft Windows Network running Microsoft Windows Server. You can then set environment variables remotely; determine the location of Oracle Tuxedo event logging; add, remove, stop, or start `tlisten` services; and tune IPC resources.

To access a remote machine, locate and click the machine's icon on the network tree. If you know the name of a remote machine, but do not know its work group, perform the following steps to select it:

1. In the lower right-hand corner of the Machines page, click `Select` to display the Enter Machine Name dialog box.



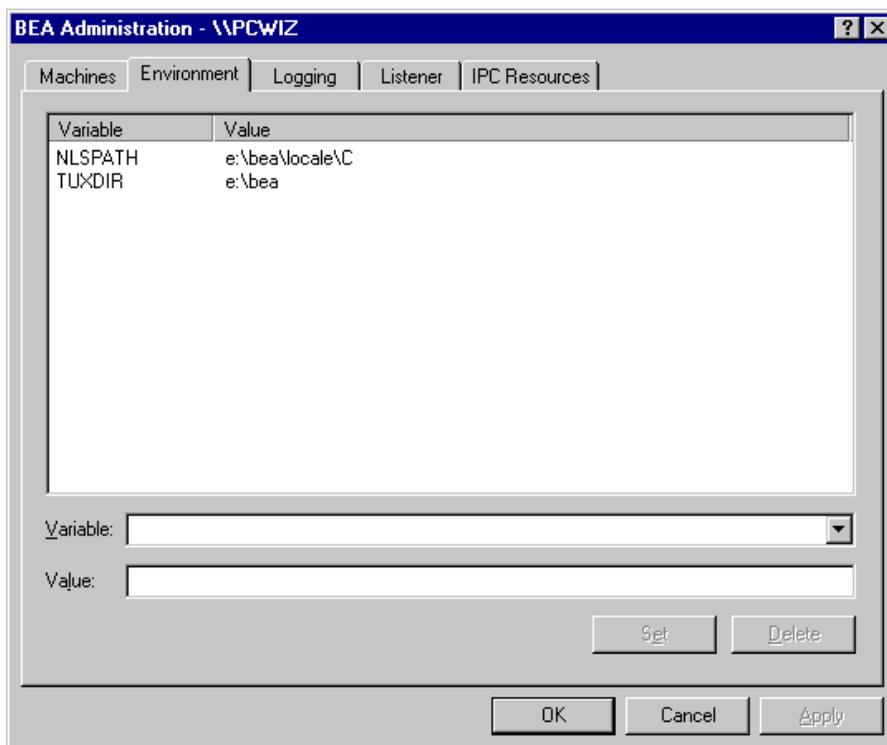
2. In the Enter Machine Name dialog box, enter the name of the remote machine (for example, \\intrepid) and click OK.

At this point, all subsequent actions performed on other pages (Environment, Logging, Listener, and IPC Resources) in the Oracle Administration window will take place on the selected machine (intrepid in this example).

## Setting and Modifying Environment Variables

The Environment page enables you to view, set, or modify Oracle Tuxedo environment variables on your Windows Server system.

**Figure 2-3 Oracle Administration Window with Environment Page Displayed**



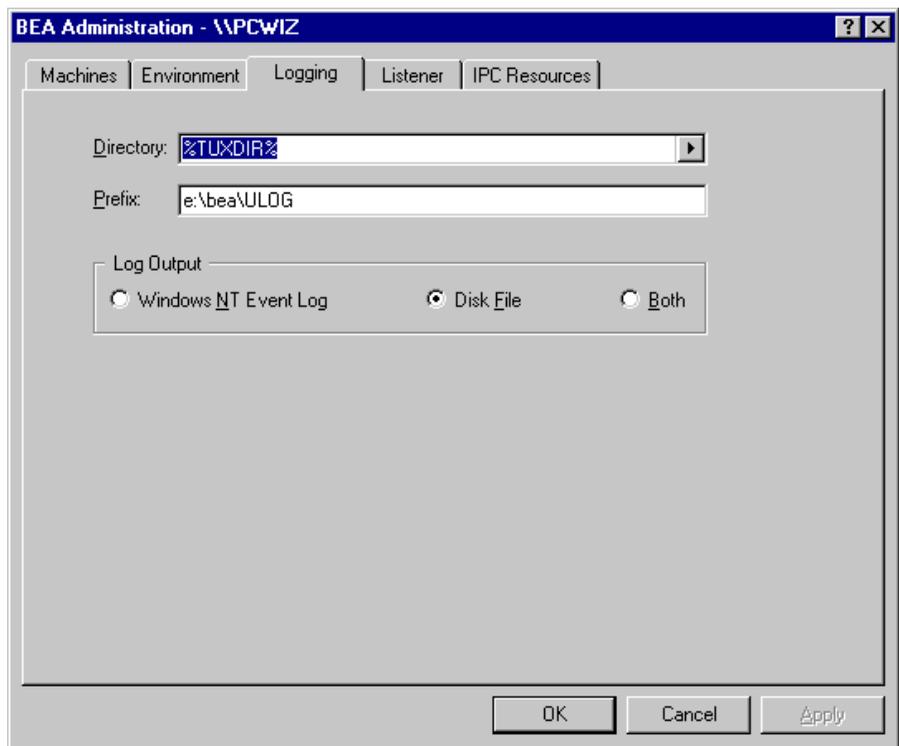
To add, modify, or delete environment variables using the Environment page, follow these steps.

1. To add a variable, enter its name in the Variable field and its value in the Value field, and then click Set.
2. To modify a variable, select the variable, enter its new value in the Value field, and then click Set.
3. To delete a variable, select the variable you want to delete and then click Delete.
4. Click OK or Apply to write your changes to the Windows Server Registry.

## Directing Oracle Tuxedo Messages to the Windows Server Event Log

The Logging page enables you to direct Oracle Tuxedo system messages to the Event Log on your Windows Server system, to the traditional user log (ULOG) disk file, or both.

Figure 2-4 Oracle Administration Window with Logging Page Displayed



You may select the Event Log option, or the traditional user log (ULOG) Disk File option, or both.

- [Log Output: Windows Event Log](#)
- [Log Output: User Log \(ULOG\) Disk File](#)

# Log Output: Windows Event Log

## Setting Up to Log to Windows Event Log

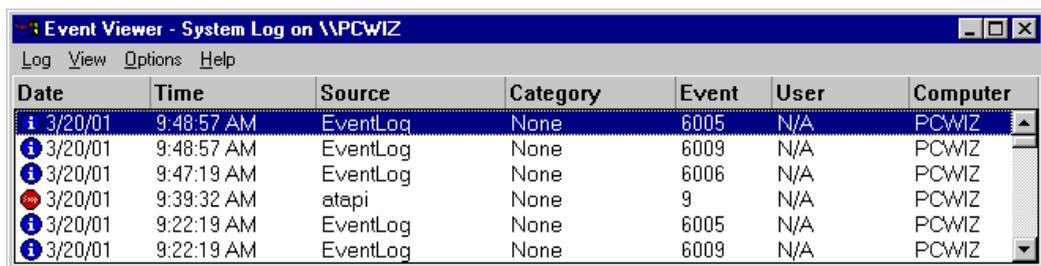
No setup is required for the Event Log.

## Viewing Windows Server Event Log Entries

To view Windows Server Event Log entries, follow these steps:

1. From the Windows Server desktop, select Start → Programs → Administrative Tools → Event Viewer to display a list of all the events that have occurred since the application booted.

Figure 2-5 Event Viewer Window



The screenshot shows the 'Event Viewer - System Log on \\PCWIZ' window. The menu bar includes 'Log', 'View', 'Options', and 'Help'. The main area is a table with the following columns: Date, Time, Source, Category, Event, User, and Computer. The table contains six entries, with the first one selected.

Date	Time	Source	Category	Event	User	Computer
3/20/01	9:48:57 AM	EventLog	None	6005	N/A	PCWIZ
3/20/01	9:48:57 AM	EventLog	None	6009	N/A	PCWIZ
3/20/01	9:47:19 AM	EventLog	None	6006	N/A	PCWIZ
3/20/01	9:39:32 AM	atapi	None	9	N/A	PCWIZ
3/20/01	9:22:19 AM	EventLog	None	6005	N/A	PCWIZ
3/20/01	9:22:19 AM	EventLog	None	6009	N/A	PCWIZ

2. On the menu bar (across the top of the Event Viewer window), click Log to display a drop-down Log menu.
3. From the drop-down Log menu, choose Application. As a result of this selection, only application-specific events are listed in the Event Viewer window.
4. Double-click the entry for the event for which you want more information. The Windows Server Event Detail window appears, with information about the event you have specified.

Figure 2-6 Event Detail Window



This window provides the following information:

- Date—date on which the event occurred.
- Time—time at which the event occurred.
- User—person using the software at the time the event occurred.
- Computer—machine on which the event occurred.
- Event ID—number under which the message is listed in *System Messages*.
- Source—section of *System Messages* in which the event is described and an appropriate action is recommended.

- Type—purpose of the message; specifically, whether the message is intended to provide information, a warning, or notification of an error.
- Description—brief summary of the event.

## Log Output: User Log (ULOG) Disk File

### Setting Up to Log to ULOG File

If you want to set up the traditional user log (ULOG) messages, perform the following steps to select a storage directory:

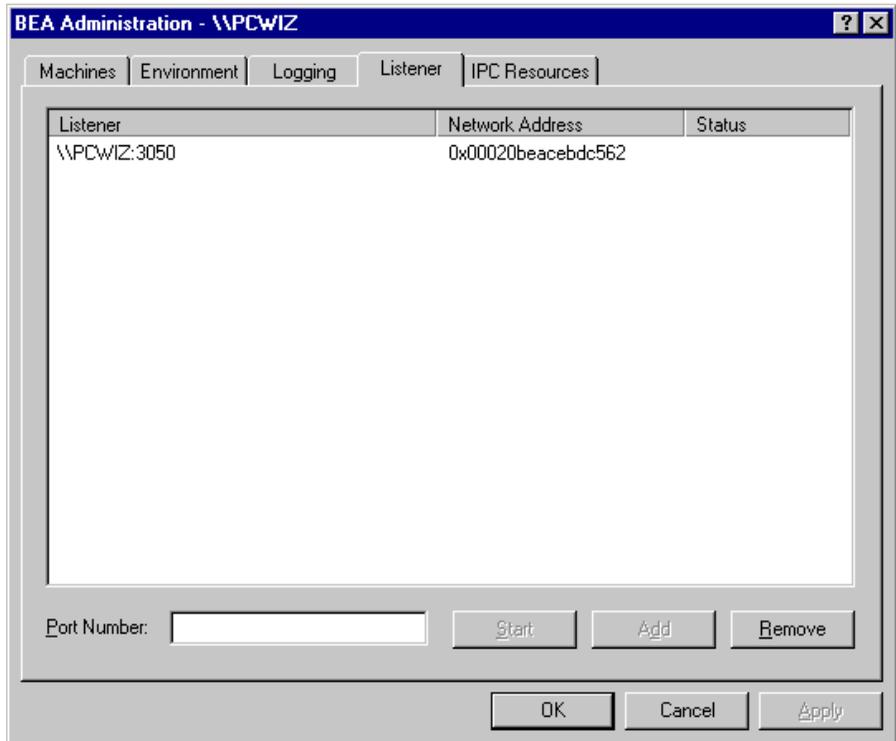
1. On the Logging page, click the arrow to the right of the Directory field to display a drop-down list of directories.
2. Select the name of the directory in which you want ULOG messages to be stored.
3. In the Prefix field, type a prefix for the name of the log file. The default prefix is ULOG, and the default file name is ULOG.*mmddyy*, where *mmddyy* is the month, day and year that the log file was created.
4. To save your selections to the Windows Server Registry, click OK or Apply.

### Viewing ULOG Entries

A ULOG is a text file. You can use any text editor to open a ULOG and view its contents.

## Configuring tlisten Processes to Start Automatically

The Listener page enables you to view, create, or modify `tlisten` services on your Windows Server system.

**Figure 2-7 Oracle Administration Window with Listener Page Displayed**

During the Oracle Tuxedo installation, the installer program installed a `tlisten` service on your Windows Server system. Each time you boot your system, the `tlisten` service starts a `tlisten` process on port 3050 of your machine. The password associated with the `tlisten` process is the one you entered during the installation.

A `tlisten` process must be started on each machine of a networked Oracle Tuxedo application before the Oracle Tuxedo system and application servers can boot. You use the `tlisten` process to perform administrative actions across multiple machines. To learn more about `tlisten` processes, see `tlisten(1)` in *Oracle Tuxedo Command Reference*.

To add, remove, stop, or start `tlisten` services using the Listener page, follow these steps.

1. To add a `tlisten` service, enter a port number in the Port Number field and then click Add. (Generally, you need one `tlisten` process for each Oracle Tuxedo application running on your machine.)

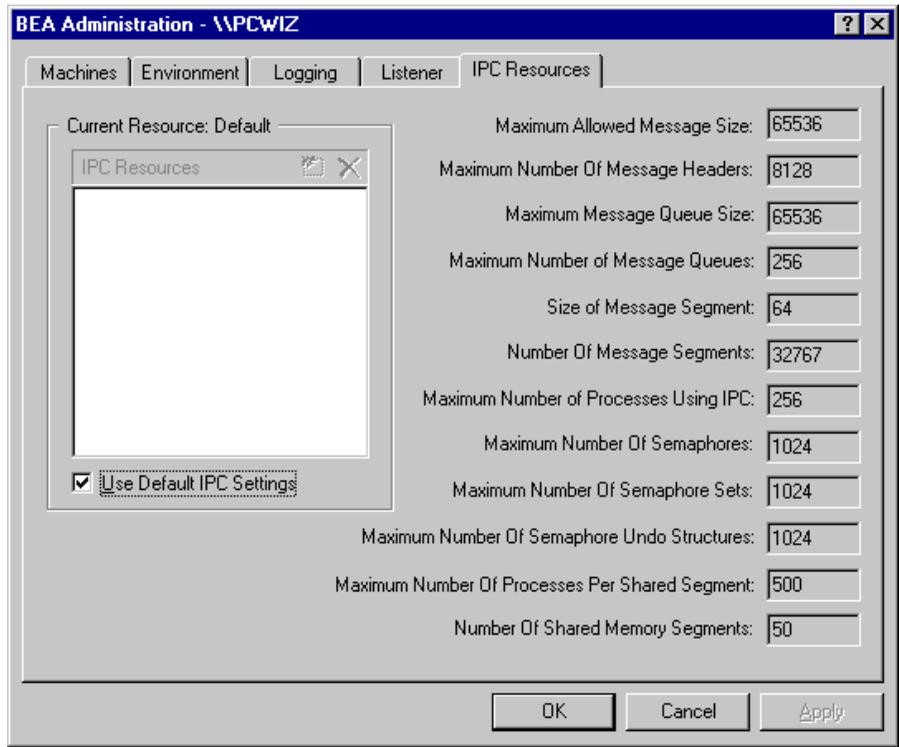
2. To remove a `tlisten` service, select the `tlisten` service you want to delete and then click Remove.
3. To stop a `tlisten` service that has been started, select the `tlisten` service you want to stop and then click Stop.
4. To start a `tlisten` service that has been stopped, select the `tlisten` service you want to start and then click Start.
5. Click OK or Apply to write your changes to the Windows Server Registry.

Besides viewing and controlling `tlisten` services using the Listener page, you can click Start → Programs → Administrative Tools → Services to launch the Services window and then view and control `tlisten` services via the Services window.

## Configuring and Monitoring IPC Resources to Maximize System Performance

The IPC Resources page enables you to configure the interprocess communication (IPC) resources on your Windows Server system to maximize Oracle Tuxedo performance.

Figure 2-8 Oracle Administration Window with IPC Resources Page Displayed



During the Oracle Tuxedo installation, the installer program installed an Oracle ProcMGR service on your Windows Server system. Each time you boot your system, the Oracle ProcMGR service configures the IPC resources on your machine to whatever values you set on the IPC Resources page. The values shown in the preceding display are the default IPC values set by the installer program.

On most machines, Oracle ProcMGR runs as installed; however, you can use the IPC Resources page to tune the IPC resources and maximize performance. To determine the minimum IPC Resource values required for an Oracle Tuxedo application, see [“Checking IPC Requirements”](#) in *Installing the Oracle Tuxedo System*. The following table maps the names of the IPC Resources on a Windows Server system to the traditional names on a UNIX system.

**Table 2-1 IPC Resource Name Mappings Between Windows and UNIX Systems**

<b>Windows Server Name</b>	<b>Traditional UNIX Name</b>
Maximum Allowed Message Size	MSGMAX
Maximum Number of Message Headers	No matching name
Maximum Message Queue Size	MSGMNB
Maximum Number of Message Queues	MSGMNI
Size of Message Segment	MSGSSZ
Number of Message Segments	MSGSEG
Maximum Number of Processes Using IPC	NPROC
Maximum Number of Semaphores	SEMMNS
Maximum Number of Semaphore Sets	SEMMNI
Maximum Number of Semaphore Undo Structures	SEMMNU
Maximum Number of Processes Per Shared Segment	No matching name
Number of Shared Memory Segments	SHMMNI

To modify IPC Resource values using the IPC Resources page, follow these steps:

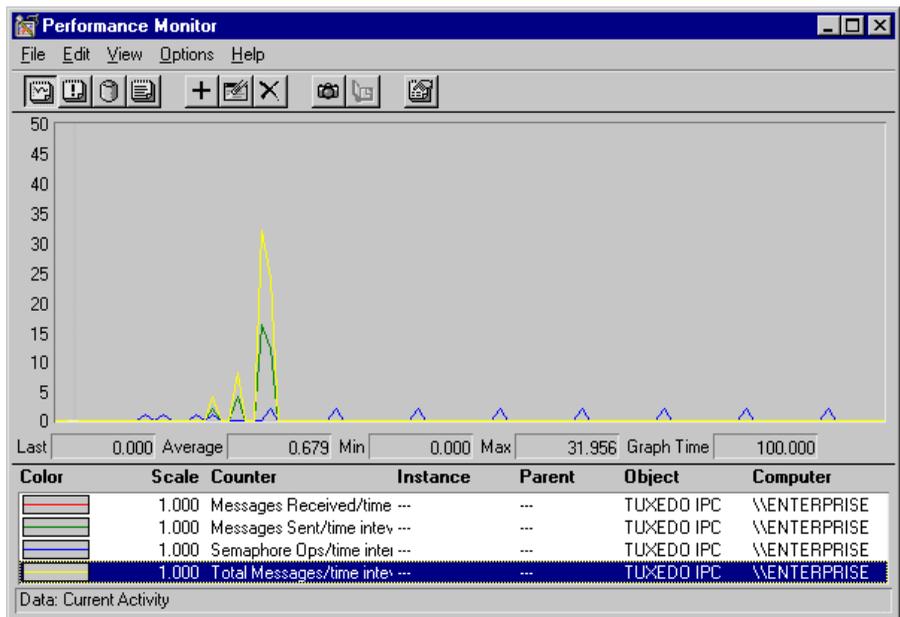
1. In the “Current Resource: Default box,” click the Use Default IPC Settings check box to clear it. An insert box appears in the Current Resource: Default box.
2. Click the insert box, enter the name of your Windows Server machine, and press Enter.
3. Click the fields next to the IPC resources you want to change and enter the desired values.
4. Click OK or Apply to write your changes to the Windows Server Registry.
5. Stop and restart the Oracle ProcMGR service to put your changes into effect: click Start → Programs → Administrative Tools → Services to launch the Services window and then stop and restart the Oracle ProcMGR via the Services window.

When interpreting the `Maximum Number of Processes Using IPC` parameter on the IPC Resources page, keep the following information in mind:

- You must count any multicontexted Oracle Tuxedo ATMI client multiple times. Your total should match the number of application associations (contexts) that can be outstanding concurrently.
- You must count any multicontexted Oracle Tuxedo ATMI server multiple times. Your total should match the number of contexts calculated by adding 1 to the value of MAXDISPATCHTHREADS in the target application's configuration (UBBCONFIG) file, where 1 represents the main dispatcher thread.

You can view the performance of a running Oracle Tuxedo application on the Windows Server Performance Monitor. Choose Start → Programs → Administration Tools → Performance Monitor to launch the Performance Monitor window.

**Figure 2-9 Performance Monitor**



## Reviewing the Windows Server Registry Content

The Windows Server Registry is the repository for all hardware, software, and application configuration settings for the Windows Server system. During the Oracle Tuxedo installation, the

installer program writes general installation information as well as IPC Resource values to the Registry. The structure of the Registry relevant to Oracle Tuxedo is as follows.

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\TUXEDO\<>version>\...
```

- Developer
- Environment
- Security

To view this structure, choose Start → Run to launch the Run dialog box, enter `Regedit`, and click OK to launch the Registry Editor window.

## Developer Key

The Developer key stores product information, including the major and minor version numbers of the release, and user and company names.

## Environment Key

The Environment key stores the locations referenced by the Oracle Tuxedo environment variables set on your Windows Server system. It also stores other values such as IPC resource settings.

## Security Key

The Security key holds the access permissions for Oracle Tuxedo processes and services. The following permissions are mandatory:

- Any user who runs `tlisten(1)` must have read access permissions.
- The account under which the Oracle ProcMGR service is running must have read access permissions.

Oracle recommends that the `Administrator` have full control permissions.

# Using the Visual C++.Net IDE To Develop Oracle Tuxedo ATMI Applications

The following sections describe how to develop ATMI application clients and/or servers in the Microsoft Visual C++ Integrated Development Environment (`msdev`) on a Windows desktop system, or Windows Server system:

- [Before You Start](#)
- [Using Development Tool](#)
- [Using the `buildserver` and `buildclient` Commands](#)
- [Adding BuildTuxedo to the MSDEV Tools Menu](#)
- [Creating Oracle Tuxedo ATMI Project Files](#)
- [Setting Up Your Environment](#)
- [Debugging an Oracle Tuxedo ATMI Server Application](#)
- [Developing an ATMI Application Using the Command Line Instead of the Visual C++.Net IDE GUI](#)

## Before You Start

Because Oracle Tuxedo ATMI client but not server components can be installed on Windows desktop systems, you can use the Microsoft Visual C++ IDE on these platforms to develop only ATMI application clients. In contrast, because both Oracle Tuxedo ATMI client and server components can be installed on Windows Server, you can use the Microsoft Visual C++.Net IDE on the Windows Server platform to develop both ATMI application clients and servers.

## Using Development Tool

Oracle Tuxedo ATMI integrates into the Microsoft Visual C++.Net IDE (`msdev`) and emulates the functionality of `msdev` when integration is not possible. This integration makes it easier for you to develop Oracle Tuxedo ATMI applications for 64-bit and 32-bit Windows operating systems.

The coding required to create Oracle Tuxedo ATMI service requests, send requests, set up conversational connections, and get replies is fundamentally the same in both UNIX and Windows Server environments.

`BuildTuxedo` is a single development tool, tightly integrated with the `msdev` build environment, that developers can use instead of the `buildserver`, `buildclient`, and `buildclt` commands to develop an Oracle Tuxedo application. (Of course, these commands are still available for those who prefer to use them.) The `BuildTuxedo` system operates seamlessly on all currently supported Windows environments.

## Using the `buildserver` and `buildclient` Commands

Both the `buildclient` and `buildserver` commands are available to Windows Server systems; however, only the `buildclient` command is available to Windows desktop systems, meaning that only Tuxedo application clients can be built on a Windows desktop system. For more information about the build commands, see [buildclient\(1\)](#) and [buildserver\(1\)](#) in *Oracle Tuxedo Command Reference*.

There are differences between how the options to the `buildclient` and `buildserver` commands work on non-integrated development environments and on integrated development environments, as indicated in the following table.

**Table 3-1 Using the buildserver and buildclient Commands**

To . . .	In a Non-Integrated Development Environment, Use This Option . . .	In an Integrated Development Environment . . .
Turn on verbose mode	<code>-v</code>	All options are displayed on tabs by default. (The <code>-v</code> option is unnecessary and unsupported.)
Specify an output file	<code>-o (output_filename)</code>	<ol style="list-style-type: none"> <li>1. Select Settings from the <code>msdev</code> Build menu.</li> <li>2. Select the Link tab in the Project Settings dialog box.</li> <li>3. Specify the name of your output file by using the edit control.</li> </ol>
Specify the first file to be linked	<code>-f</code>	<ol style="list-style-type: none"> <li>1. Select Settings from the <code>msdev</code> Build menu.</li> <li>2. Select the Link tab in the Project Settings dialog box.</li> <li>3. Specify the first file to be linked by using the edit control.</li> </ol>
Specify the last file to be linked	<code>-l</code>	<ol style="list-style-type: none"> <li>1. Select Project from the <code>msdev</code> menu.</li> <li>2. Select the Link tab in the Project Settings dialog box.</li> <li>3. Specify the last file to be linked by using the edit control.</li> </ol>
Specify a resource manager	<code>-r</code>	<ol style="list-style-type: none"> <li>1. Access the <code>BuildTuxedo</code> GUI.</li> <li>2. Access the <code>BuildTuxedo</code> Test window and select the Resources tab.</li> <li>3. In the Tuxedo Resource Manager field, enter the name of the resource manager.</li> </ol>
Specify services that will be available on a server	<code>-s</code>	<ol style="list-style-type: none"> <li>1. Access the <code>BuildTuxedo</code> GUI.</li> <li>2. Access the <code>BuildTuxedo</code> Test window and select the Services tab.</li> <li>3. In the Service Names field, enter the name of each service on a separate line.</li> </ol>
Use the COBOL compiler	<code>-c</code>	COBOL is unavailable.

To modify the build environment in an integrated development environment, follow these steps:

1. Choose Settings from the `msdev` Project menu.
2. Select 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, follow these steps:

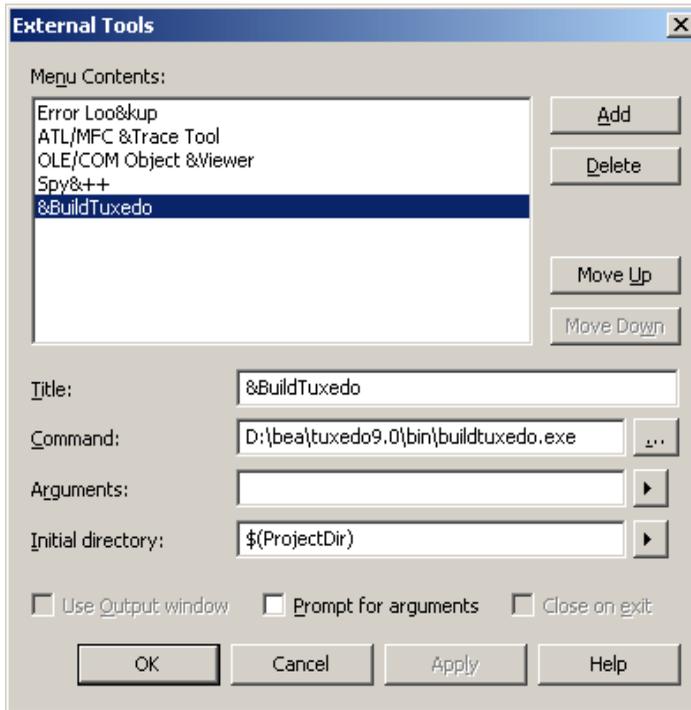
1. From the MSDEV Tools menu, choose Options to display the Options dialog box.
2. In the Options dialog box, select the Directories tab.

## Adding BuildTuxedo to the MSDEV Tools Menu

To add `BuildTuxedo` to the MSDEV Tools menu, follow these steps:

1. In the Microsoft Visual C++.Net IDE (`msdev`) graphical user interface (GUI), choose Tools → External Tools. The External Tools window displays.
2. In the External Tools window, click Add to clear the Title field.

Figure 3-1 External Tools Window



3. In the Title field of the External Tools page, type `&BuildTuxedo` in the empty field.
 

**Note:** To enable a letter key as a *hot key*, put an ampersand (&) before the appropriate letter.
4. In the Command field of the External Tools page, provide a pathname by doing one of the following:
  - Type the full pathname of `BuildTuxedo` (`%TUXDIR%\bin\BuildTuxedo`).
  - To the right of the Command field, click the ellipsis button , browse to the desired directory, select `buildtuxedo.exe`, and click Open. The selected pathname appears in the External Tools window.
5. In the Initial directory field of the External Tools page, click the right arrow button to display a pop-up menu. In the pop-up menu, choose Project Directory (`$(ProjectDir)`).
 

**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.

6. Click Apply to save the changes and click OK to close the External Tools window.

BuildTuxedo is now part of the MSDEV Tools menu.

## Creating Oracle Tuxedo ATMI Project Files

BuildTuxedo maintains a separate project file in the current directory for each Oracle Tuxedo ATMI 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 information:

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 makefile (*filename.mak*)
- An msdev project file (*filename.mdp*, *filename.dsw*, or *filename.dsp*)

If BuildTuxedo cannot find either of these files, it displays a warning and/or fails to activate. If the directory contains multiple BuildTuxedo project files, or 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 make to the project file or any file maintained by the Oracle Tuxedo system, click Cancel or Esc.

## Setting Up Your Environment

Before you can build your Oracle Tuxedo ATMI application in an IDE, you must set the following fundamental parameters in your environment:

- Build type
- Header file
- Filename of the C or C++ file to be created and maintained by BuildTuxedo
- Function names
- Service names

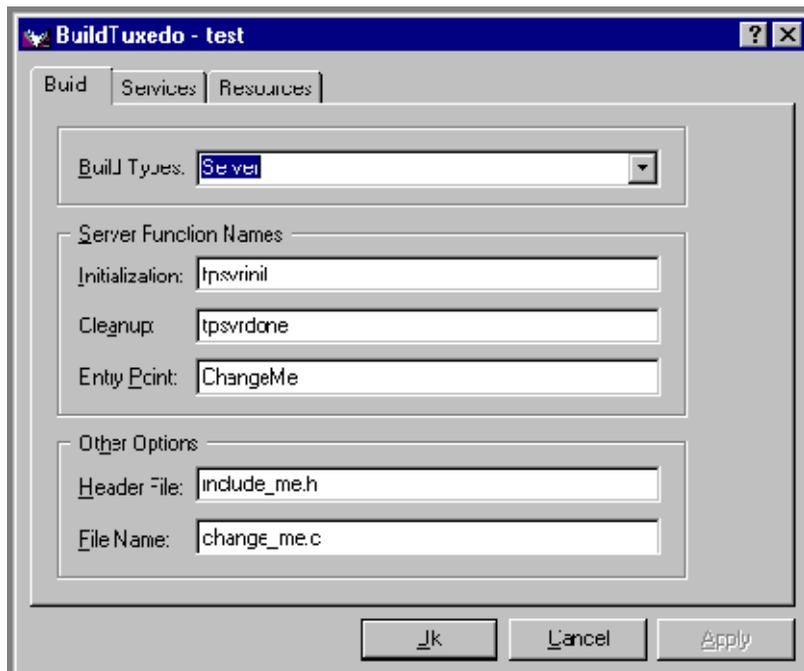
To provide this information, access the BuildTuxedo *project\_name* dialog box in the msdev GUI.

## Specifying the Build Type, Header File, and Filename

To specify the build type, header file, and filename, follow these steps:

1. In the project\_name dialog box, click the Build tab to display the Build page.

**Figure 3-2 Build Page**



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

- Server
- Native client or Workstation client

**Note:** Windows desktop users can select only Workstation Client type.

If You Select . . .	Then, After Making Your Selection . . .
Server	Enter information in the Initialization, Cleanup, and Entry Point fields. Proceed to step 3.
Native Client or Workstation Client	Proceed to step 3.

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 generated by `BuildTuxedo`. After specifying its name, you can call the function from anywhere in the application.

3. In the Header File field of the Build page, enter `stdafx.h`.
4. In the File Name field of the Build page, 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 Oracle Tuxedo ATMI project correctly. `BuildTuxedo` opens the file specified in the Header File field of the Build page, if the file is present; otherwise, it creates a new one. A section starting with:

```
//Begin Tuxedo Section*****DO NOT EDIT*****
```

and ending with:

```
//End Tuxedo Section
```

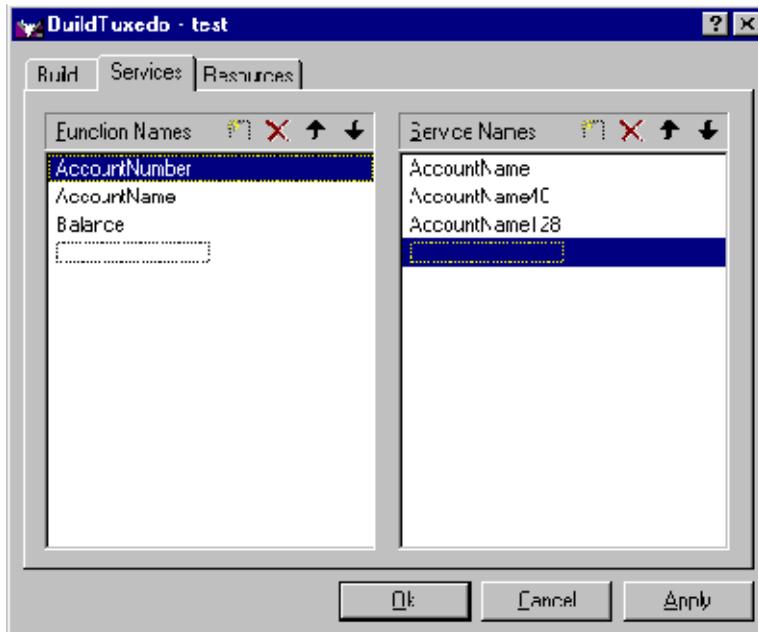
designates the area in the header file in which pragma statements are written. This area is maintained by `BuildTuxedo`. If `BuildTuxedo` does not locate this section in the file, it appends it to the end of the header file. Because all other text within the file remains unchanged, you can specify `stdafx.h` as the header file.

**Note:** If your project is new, 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 Function and Service Names

Select the Services tab to display the Services page, in which you specify function and service names.

**Figure 3-3 Services Page**



In the Services page, 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.

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

Generally, the service and the function that performs the service are represented by the same name. For example, 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 may not be known until run time.

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 a single `TPSVCIINFO` 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.

**Note:** If you are using the `buildserver(1)` command, you can provide this information with the `-s` option. For more information about the `-s` option, see *Programming an Oracle Tuxedo ATMI Application Using C* or *Programming an Oracle Tuxedo ATMI Application Using COBOL*.

To add or edit names on the Function Names list, follow these steps:

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

To	Action
Add an item	<ol style="list-style-type: none"><li>1. Click the New icon (broken box with twinkle in the upper left corner).</li><li>2. Click Insert, or select a blank area on the list (that is, a dotted lined box).</li><li>3. Type the name of the new function.</li></ol>
Modify an existing name	Highlight the name and type over it.
Delete a name	Highlight the name on the list and click the X icon or click Delete.

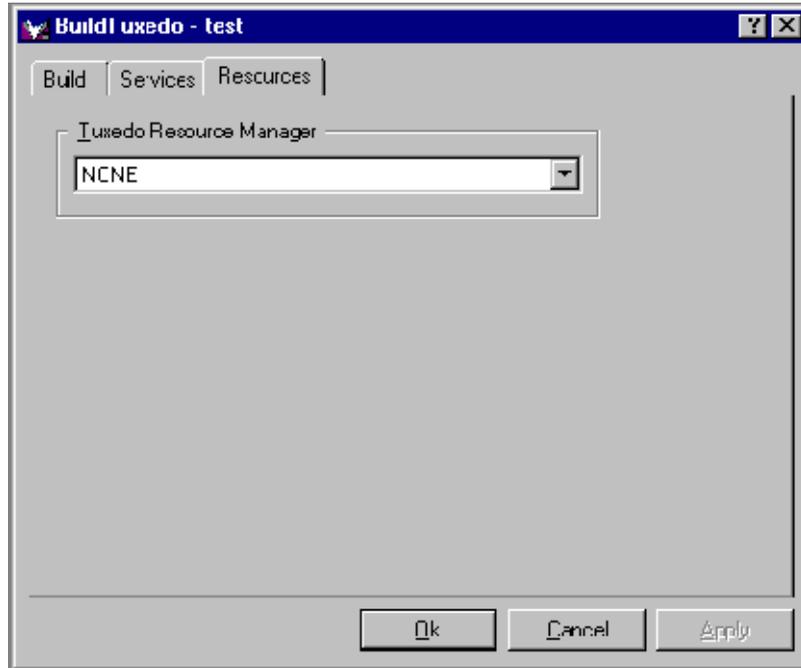
2. When complete, click OK or press Enter to save your changes and exit. (To exit without saving your changes, click Esc or Tab.)

## Specifying a Resource Manager

To specify a resource manager, follow these steps:

1. Select the Resources tab in the `BuildTuxedo` window to display the Resources page.

Figure 3-4 Resources Page



2. In the Tuxedo Resource Manager field of the Resources page, type the full pathname of the resource manager for your application.

**Note:** The Tuxedo Resource Manager field contains a list of Oracle 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.

## Debugging an Oracle Tuxedo ATMI Server Application

You should use the following procedure only if you have built the ATMI server application using the Debug configuration for your project.

To debug a server that has not been booted, follow these steps.

1. At any Windows command prompt, type `tmboot -n -d 1 -s servername` to display the command-line options used by `tmboot (1)` to start `servername`.

2. Execute the `tmboot -M` command to boot the BBL. (If necessary, boot additional application servers or machines.)
3. In `msdev`, select Project → Settings.
4. In the Program Arguments field of the Debug page, type the command-line options used in step 1.
5. Start the debugger and start debugging the server application.

**Note:** Because Oracle Tuxedo libraries are not built with debugging information and source code is not provided, you cannot access the Oracle Tuxedo code directly.

6. To end the debugging session, type `tmshutdown` at any Windows command prompt.

**WARNING:** Do not stop the server by selecting Debug → Stop; the Oracle Tuxedo system may subsequently attempt to restart the server.

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

## Developing an ATMI Application Using the Command Line Instead of the Visual C++.Net IDE GUI

If you must develop an Oracle Tuxedo ATMI application using the command line instead of the Microsoft Visual C++ IDE GUI, use the `buildserver (1)` and `buildclient (1)` commands. To do this, specify the compiler and link options necessary to build an Oracle Tuxedo ATMI application. For information about using these tools, see the following documents:

- *Oracle Tuxedo Command Reference*
- *Tutorials for Developing Oracle Tuxedo ATMI Applications*
- One of the following language-specific guides:
  - *Programming an Oracle Tuxedo ATMI Application Using C*
  - *Programming an Oracle Tuxedo ATMI Application Using COBOL*

To build a debug version of your ATMI 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. In addition, you may need to define the `_DEBUG` preprocessor directive. To complete the process, indicate the link option as follows:

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

## See Also

For information about the Oracle Tuxedo ATMI, see the following documents:

- *Programming an Oracle Tuxedo ATMI Application Using C*
- *Programming an Oracle Tuxedo ATMI Application Using Java*
- *Programming an Oracle Tuxedo ATMI Application Using COBOL*
- *Programming an Oracle Tuxedo ATMI Application Using FML*
- *Oracle Tuxedo Command Reference*
- *Oracle Tuxedo ATMI C Function Reference*
- *Oracle Tuxedo ATMI COBOL Function Reference*
- *Oracle Tuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference*