**Detecting Errors Using Logs**

The BEA Tuxedo log files can help you detect failures in both your application and your system by:

* [Analyzing tlisten Messages in the ULOG](https://docs.oracle.com/cd/E13203_01/tuxedo/tux71/html/admon13.htm" \l "1092118)
* [Analyzing the Transaction Log (TLOG)](https://docs.oracle.com/cd/E13203_01/tuxedo/tux71/html/admon13.htm" \l "1092134)
* [Analyzing the User Log (ULOG)](https://docs.oracle.com/cd/E13203_01/tuxedo/tux71/html/admon13.htm" \l "1092968)

**Analyzing tlisten Messages in the ULOG**

Part of the ULOG records error messages to the tlisten process. You can view tlisten messages using any text editor. Each machine, including the MASTER machine contains a separate tlisten process. Though separate tlisten logs are maintained in the ULOG on each machine, they can be shared across remote file systems.

The ULOG records tlisten process failures. tlisten is used, during the boot process, by tmboot and, while an application is running, by tmadmin. tlisten messages are created as soon as the tlisten process is booted. Whenever a tlisten process failure occurs, a message is recorded in the ULOG.

**Note:** Application administrators are responsible for analyzing the tlisten messages in the ULOG, but programmers may also find it useful to check these messages.

The *BEA Tuxedo System Messages CMDTUX Catalog* contains the following information about tlisten messages:

* Descriptions of all messages
* Recommended actions that you (or a programmer) can take to resolve the error conditions reported in these messages

**tlisten Message Example**

Consider the following example of a tlisten message in the ULOG.

121449.gumby!simpserv.27190.1.0: LIBTUX\_CAT:262: std main starting

A ULOG message consists of a tag and text. The tag consists of the following:

* A 6-digit string (hhmmss) representing the time of day (in terms of hour, minute, and second)
* The name of the machine (as returned, on UNIX systems, by the uname -n command)
* The name and process identifier of the process that is logging the message. (This process ID can optionally include a transaction ID.) Also included is a thread ID (1) and a context ID (0).

**Note:** Placeholders are printed in the thread\_ID and context\_ID field of entries for single-threaded applications. (Whether an application is multithreaded is not apparent until more than one thread is used.)

The text consists of the following:

* The name of the message catalog
* The message number
* The BEA Tuxedo system message

**Note:** You can find this message in the *BEA Tuxedo System Messages LIBTUX Catalog*.

**Analyzing the Transaction Log (TLOG)**

The TLOG is a binary file that contains only messages about global transactions that are in the process of being committed. To view the TLOG, you must first convert it to text format so that it is readable. The BEA Tuxedo system provides two tmadmin operations to do this:

* dumptlog (dl) downloads (or dumps) the TLOG (a binary file) to a text file.
* loadtlog uploads (or loads) an text version of the TLOG into an existing TLOG (a binary file).

The dumptlog and loadtlog commands are also useful when you need to move the TLOG between machines as part of a server group migration or machine migration.

**Detecting Transaction Errors**

You can detect TLOG errors using the MIB to obtain the status of a transaction. You can also run the tmadmin command display transaction to detect any errors in transactions.

**Analyzing the User Log (ULOG)**

On each active machine in an application, the BEA Tuxedo system maintains a log file that contains BEA Tuxedo system error messages, warning messages, debugging messages, or other helpful information. This file is called the user log or ULOG. The ULOG simplifies the job of finding errors returned by the BEA Tuxedo ATMI, and provides a central repository in which the BEA Tuxedo system and applications can store error information.

You can use the information in the ULOG to identify the cause of system or application failures. Multiple messages about a given problem can be placed in the user log. Generally, earlier messages provide more useful diagnostic information than later messages.

**ULOG Message Example**

In the following example, message 358 from the LIBTUX\_CAT catalog identifies the cause of the trouble reported in subsequent messages, namely, that there are not enough UNIX system semaphores to boot the application.

**Sample ULOG Messages**

151550.gumby!BBL.28041.1.0: LIBTUX\_CAT:262: std main starting   
**151550.gumby!BBL.28041**.1.0: **LIBTUX\_CAT:358: reached UNIX limit on semaphore ids**151550.gumby!BBL.28041.1.0: LIBTUX\_CAT:248: fatal: system init function ...  
151550.gumby!BBL.28040.1.0: CMDTUX\_CAT:825: Process BBL at SITE1 failed ...  
151550.gumby!BBL.28040.1.0: WARNING: No BBL available on site SITE1.  
 Will not attempt to boot server processes on that site.