



BEA SNMP Agent MIB Reference

For BEA Tuxedo and
BEA WebLogic Enterprise

BEA SNMP Agent 2.1
Document Edition 2.1
October 2000

Copyright

Copyright © 2000 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, BEA Builder, BEA Jolt, BEA Manager, BEA MessageQ, ObjectBroker, TOP END, Tuxedo, and webLogic are registered trademarks of BEA Systems, Inc. BEA Connect, BEA WebLogic Collaborate, BEA WebLogic Process Integrator, eLink, eSolutions, M3, WebLogic Commerce Server, WebLogic Enterprise, and WebLogic Personalization Server are trademarks of BEA Systems, Inc.

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

BEA SNMP Agent MIB Reference for BEA Tuxedo and BEA WebLogic Enterprise

Document Edition	Date	Software Version
2.1	October 2000	BEA SNMP Agent 2.1

Contents

About This Document

What You Need to Know	vi
e-docs Web Site	vi
How to Print the Document.....	vi
Related Information.....	vii
Contact Us!	vii
Documentation Conventions	viii

1. The Tuxedo MIB for SNMP

MIB Object Identifiers	1-1
ASN.1 File.....	1-4
Relative and Absolute Object Identifiers	1-4
Specifying Object Identifiers	1-4
Updating MIB Objects	1-5
Supported MIB Objects.....	1-5
Managing Tuxedo and WLE Applications Using the Tuxedo MIB for SNMP	1-7
Querying Non-Existent MIB Objects.....	1-7
Structure of the MIB Definitions.....	1-8
MIB Event Trap Definitions Format.....	1-8
Differences Between the Tuxedo MIB and the Tuxedo MIB for SNMP	1-9

2. Tuxedo Core MIB

tuxTserverCtxTbl	2-159
------------------------	-------

3. Domains MIB	
4. BEA Domain List MIB	
5. WLE MIBs	
6. Access Control List MIB	
7. Workstation MIB	
tuxTwshTbl.....	7-2
8. Application Queue MIB	
9. Event Broker MIB	
10. Tuxedo Traps MIB	
Specific Trap Number	10-1
Variable Bindings	10-2
Trap Definitions.....	10-5
DOMAIN Traps.....	10-6
BRIDGE Traps	10-12
SERVER Event Traps	10-14
CLIENT Traps.....	10-18
TRANSACTION Traps.....	10-20
EVENT Traps.....	10-21

Index

About This Document

The *BEA SNMP Agent MIB Reference for BEA Tuxedo and BEA WebLogic Enterprise* provides reference information about the MIBs shipped in the BEA SNMP Agent software. This guide is organized as follows:

- Chapter 1, “The Tuxedo MIB for SNMP,” describes MIB Object Identifiers, structure of the MIB definitions, the difference between the Tuxedo MIB and the SNMP Tuxedo MIB, how to manage Tuxedo and WLE applications using the SNMP MIB, and how to manage Tuxedo and WLE resources from a management framework.
- Chapter 2, “Tuxedo Core MIB,” describes the basic groups, objects, and attributes of the objects that form a Tuxedo application, and that are defined in the Core MIB.
- Chapter 3, “Domains MIB,” describes the interaction among domains using improved groups and attributes terminology.
- Chapter 4, “BEA Domain List MIB,” describes the BEA domain list.
- Chapter 5, “WLE MIBs,” describes the MIB groups and member objects specific to WLE applications, and lists the WLE-specific objects that are included as part of the Tuxedo Core MIB.
- Chapter 6, “Access Control List MIB,” describes the access control list (ACL) MIB groups.
- Chapter 7, “Workstation MIB,” describes the workstation MIB groups.
- Chapter 8, “Application Queue MIB,” describes the application queue MIB groups.
- Chapter 9, “Event Broker MIB,” describes the event broker MIB.
- Chapter 10, “Tuxedo Traps MIB,” describes the Tuxedo event traps MIB.

What You Need to Know

This document is intended for network or system administrators who are responsible for administering SNMP master agents and SMUX subagents.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at <http://e-docs.beasys.com>.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the **File**—>**Print** option on your Web browser.

A PDF version of this document is available on the BEA SNMP Agent documentation Home page on the e-docs Web site. You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the BEA SNMP Agent documentation Home page, click the PDF files link and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at <http://www.adobe.com/> or by contacting your local Adobe sales office.

Related Information

The following BEA SNMP Agent documents contain additional information that is relevant to using the *BEA SNMP Agent MIB Reference*:

- *BEA SNMP Agent Installation Guide for BEA Tuxedo and BEA WebLogic Enterprise*
- *BEA SNMP Agent Administrator's Guide for BEA Tuxedo and BEA WebLogic Enterprise*
- *BEA SNMP Agent Release Notes for BEA Tuxedo and BEA WebLogic Enterprise*

Contact Us!

Your feedback on the BEA BEA SNMP Agent documentation is important to us. Send us e-mail at docsupport@beasys.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA SNMP Agent documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA SNMP Agent 2.1 release.

If you have any questions about this version of BEA SNMP Agent, or if you have problems installing and running BEA SNMP Agent, 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

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 simultaneously.
<i>italics</i>	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. <i>Examples:</i> #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. <i>Example:</i> void commit ()
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>

Convention	Item
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR
{ }	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. <i>Example:</i> 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: <ul style="list-style-type: none">■ 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. <i>Example:</i> 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.



1 The Tuxedo MIB for SNMP

This chapter describes the Tuxedo MIB for SNMP. It includes the following sections:

- MIB Object Identifiers
- Managing Tuxedo and WLE Applications Using the Tuxedo MIB for SNMP
- Structure of the MIB Definitions
- Differences Between the Tuxedo MIB and the Tuxedo MIB for SNMP

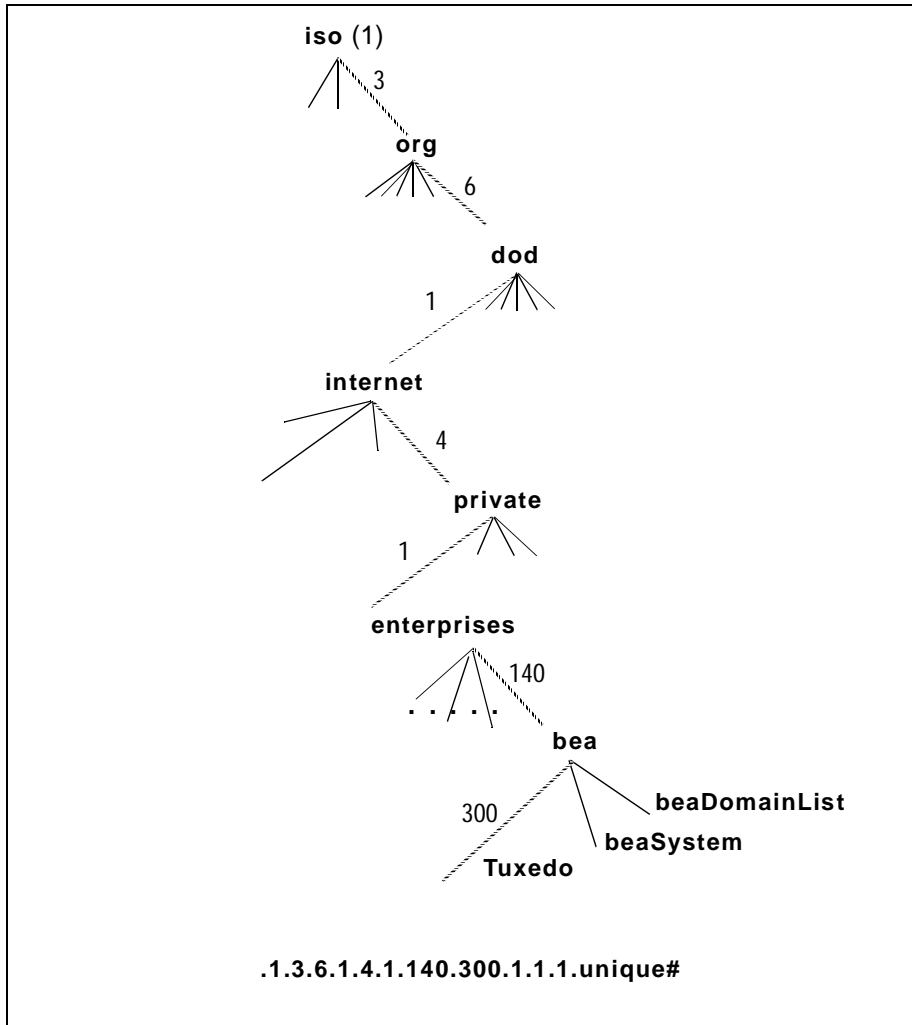
MIB Object Identifiers

An SNMP management framework manages a collection of objects in a hierarchy of information known as a Management Information Base (MIB). Each object in the MIB has an *object identifier* (OID), which the manager uses to request the object's value from the agent.

To use the BEA SNMP Agent Integrator polling feature described in the *BEA SNMP Agent Administrator's Guide*, OIDs are used to identify the managed objects whose values are retrieved by the BEA SNMP Agent Integrator when it checks for the occurrence of events in the managed resource.

An OID is a sequence of integers that uniquely identifies a managed object by defining a path to that object through a tree-like structure called the *OID tree* or registration tree. When an SNMP agent needs to access a specific managed object, it traverses the OID tree to find the object. The MIB object identifier hierarchy and format is shown in Figure 1-1.

Figure 1-1 MIB Object Identifier Hierarchy and Format



In this hierarchy, each BEA private MIB object that the BEA SNMP Agent software manages has a unique object identifier. The BEA Tuxedo object uses a prefix of `.1.3.6.1.4.1.140` to identify it as an object in the BEA private MIB.

For a complete listing of objects in the BEA private MIB in ASN.1 notation, read the file `bea.asn1` in the installed BEA SNMP Agent software.

ASN.1 File

An ASN.1 file is a standard SNMP file that defines the objects that make up an SNMP-compliant MIB. Each object in the file is defined in compliance with the SNMP standard. The BEA SNMP Agent provides the ASN.1 file `bea.asn1` for defining the Tuxedo MIB (with WebLogic Enterprise extensions) for SNMP.

Note: The Tuxedo and WLE MIB definitions are written in concise MIB format in accordance with RFC 1212, as required by the SNMP standard.

Relative and Absolute Object Identifiers

Absolute OIDs specify a path to an attribute from the root of the OID tree. Absolute OID names always begin with a dot and must specify every node of the OID tree from the top-most node to the specific managed object. For example:

```
.1.3.6.1.2.1.1.1
```

Relative OIDs specify a path to the attribute relative to some node in the OID tree. For example, `2.1.1.1.7` specifies the `sysContact` object in the `system` group, relative to the Internet node in the OID tree.

Specifying Object Identifiers

In addition to using the “dot-dot” notation, a series of integers separated by dots to describe OIDs, you can also express OIDs by using textual symbols instead of numbers to represent nodes in the path to the object, or by using a combination of both integers and textual symbols. A *symbolic* OID uses mnemonic keywords to specify the managed object. For example:

```
mgmt.mib-2.system.sysDescr
```

The following numeric OID uses integers to specify the same managed object:

```
2.1.1.1.1
```

Note that this example is a relative OID.

An OID can combine both symbolic and numeric representations of individual nodes of the OID tree; for example:

```
mgmt.mib-2.1.sysDescr
```

Note: When using OIDs to specify objects whose values are checked using BEA SNMP Agent Integrator polling rules, only the numeric form of the OID can be used. For details, see Chapter 8 of the *BEA SNMP Agent Administrator's Guide*.

Updating MIB Objects

Some objects in the BEA SNMP MIB for Tuxedo systems can be set (updated) only under certain states of the Tuxedo system. If you get an error while trying to set read-write objects in this MIB, please refer to the `ULOG` file.

Supported MIB Objects

To access MIB objects that are managed by agents or subagents, the scope of the OID tree for which each agent or subagent is responsible must be defined to the BEA SNMP Agent Integrator. For monolithic SNMP agents, and SMUX or DPI master agents, this is done by specifying an OID in one or more `NON_SMUX_PEER` entries in the `beamgr.conf` configuration file, as described in Chapter 5 of the *BEA SNMP Agent Administrator's Guide*. The BEA SNMP Agent Integrator then knows to access the managed objects in that branch of the OID tree through the specified agent.

The BEA SNMP Agent Integrator *directly* accesses MIB objects in the SMUX MIB, the MIB II system and snmp groups, and the `beaIntAgtTable` MIB object in the BEA SNMP Agent MIB. The `beaIntAgtTable` MIB objects define the polling capability of the BEA SNMP Agent Integrator.

The `bea.asn1` file contains definitions for Tuxedo and WLE objects that are SNMP compliant. The BEA SNMP Agent supports the following MIBs:

- **Tuxedo Core MIB**—Supplies the definitions for controlling the operation and configuration of the Tuxedo system. This MIB contains the main information groups for Tuxedo applications, including domains, machines, queues, servers, routing, clients, and services. See Chapter 2, “Tuxedo Core MIB.”

- **Domains MIB**—Uses improved group and attribute terminology to describe the interaction between domains. This improved terminology has also been applied to `DMCONFIG` file syntax. See Chapter 3, “Domains MIB.”
- **BEA Domain List MIB**—Contains information about the WLE or Tuxedo domains that the agent is monitoring. See Chapter 4, “BEA Domain List MIB.”
- **WLE MIBs**—An extension of the core MIB that incorporates definitions for management of WLE CORBA and JAVA features. See Chapter 5, “WLE MIBs.”
- **Access Control List MIB**—Enables you to define and control your application security options. See Chapter 6, “Access Control List MIB.”
- **Workstation MIB**—Specifies information about Tuxedo client workstations including workstation listeners and handlers. See Chapter 7, “Workstation MIB.”
- **Application Queue MIB**—Provides the administrative control required for managing access to application queues. The objects in this MIB include items for managing queue spaces, queues, messages, and transactions. See Chapter 8, “Application Queue MIB.”
- **Event Broker MIB**—Represents event subscriptions registered with the Event Broker for receiving event notifications. See Chapter 9, “Event Broker MIB.”
- **Tuxedo Traps MIB**—Defines all the trap notifications that are generated by the BEA SNMP Agent and the objects passed in the variable bindings for these traps. See Chapter 10, “Tuxedo Traps MIB.”

Tuxedo managed object names within the MIBs for SNMP are usually prefixed with the letters `tux`. WLE managed object names are usually prefixed with `wle`. For example, the Tuxedo Core MIB contains a group termed Machine. Within this group are managed objects such as the following:

`tuxTmachinePmid`

Represents a physical machine identifier

`tuxTmachineLmid`

Represents the logical machine identifier

Most Tuxedo objects also apply to WLE applications.

Managing Tuxedo and WLE Applications Using the Tuxedo MIB for SNMP

The Tuxedo and WLE systems identify application items in a hierarchy of information known as Tuxedo Management Information Bases . These databases contain definitions that describe the components found in the Tuxedo or WLE application. Included with the BEA SNMP Agent is an SNMP version of the Tuxedo MIBs. The Tuxedo MIB for SNMP also includes MIB objects that represent attributes of Tuxedo and WLE resources.

To monitor or modify values of managed objects through your systems management platform, you need to know which MIB objects represent the features of Tuxedo or WLE resources that are relevant to your management goals. You also need to know the data types, default values, and access permissions for these MIB objects.

Querying Non-Existent MIB Objects

If you attempt to retrieve the value of a MIB object, and that object does not exist, either no value is returned, or one of the following values is returned:

- -1 if the object is numeric
- A dash (-) if the object data type is `DisplayString`

Using the Tuxedo MIB for SNMP, the BEA SNMP Agent Integrator can be configured to perform local polling and generate SNMP trap events, or execute a system command when certain conditions are met. The same effect can be achieved by defining a `RULE_ACTION` entry in the `beamgr.conf` file. This configuration file is described in Chapter 8, “Configuration Files,” in the *BEA SNMP Agent Administrator’s Guide*.

Structure of the MIB Definitions

The following keywords are used to define MIB managed objects:

Syntax

Defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the SMI purposely restricts the ASN.1 constructs that can be used. These restrictions are made expressly for simplicity.

Access

Defines whether the object value can only be retrieved but not modified (read-only) or whether it can also be modified (read-write).

Note: For tabular objects, in some cases a read-write object can only be set during creation of a new row. Where this is true, it is noted in the Description section for that MIB object.

Description

Contains a textual definition of that object type that provides all semantic definitions necessary for interpretation. This clause typically contains information of the sort that would be communicated in any ASN.1 commentary annotations associated with the object.

Note: Each row in a table is an instance of the Entry object under that table. The Description section for the Entry object under a table (such as `tuxTmachineTable`) contains information on the columnar values that are minimally necessary for creation of a row—how a new row is created, whether the values pertain only to the local machine, and other pertinent information about the table objects.

MIB Event Trap Definitions Format

In addition to defining MIBs, the `bea.asn1` file also defines traps. These traps are defined in accordance with RFC 1215, Trap definitions. Chapter 10, “Tuxedo Traps MIB,” specifies the list of traps generated by the BEA SNMP Agent. The following keywords are used to define a trap:

Enterprise

An object identifier that specifies the management enterprise under whose registration authority this trap is defined. All the traps generated by the BEA SNMP Agent have an enterprise field set to the Tuxedo object identifier. The Tuxedo object identifier is .1.3.6.1.4.1.140.300. This value is passed in the `enterprise` field of the trap packet (Protocol Data Unit — PDU).

Variables

Defines the ordered sequence of MIB objects that are contained in each instance of the trap type. Each variable is placed, in order, inside the `variable-bindings` field of the SNMP trap packet (PDU).

Description

Contains a textual definition of the trap type.

Trap ID

Specifies the enterprise-specific trap ID for the trap definition. This is passed in the `specific trap ID` field of the trap packet (PDU).

Note: The value of the generic trap ID field in traps is always set to 6, indicating an enterprise-specific trap.

Differences Between the Tuxedo MIB and the Tuxedo MIB for SNMP

If you are familiar with the Tuxedo MIB, the primary differences to note when using the Tuxedo MIB for SNMP are a difference in terms and a few additional MIB items in the SNMP-based MIB.

The Tuxedo MIB identifies an abstract structure for Tuxedo resources. In a Tuxedo framework, a MIB is the classification of information in a Tuxedo application. However, instead of referring to *groups* and *managed objects*, as is common in SNMP terminology, the Tuxedo MIB defines application resources as *classes* and *attributes*.

Classes are the administrative class definitions that make up the Tuxedo MIB. Each class has a set of attributes that identify individual items in the class. Examples of Tuxedo classes are:

T_MACHINE

The class definition for a machine

T_SERVICE

The class definition for Tuxedo services

Attributes for these classes are identified by the prefix TA_ followed by the attribute name. A few examples for the T_MACHINE class are:

TA_P MID

Represents a physical machine name

TA_L MID

Represents the logical machine name

For more information about the standard Tuxedo MIB, refer to the *BEA Tuxedo Reference Manual*.

In contrast, the features of manageable resources in SNMP are called *objects* rather than attributes, and objects fall under MIB *groups* rather than classes.

2 Tuxedo Core MIB

The Tuxedo system Core MIB defines the set of groups through which the fundamental aspects of an application can be configured and managed. These include management of machines, servers, networking, and load balancing.

The Tuxedo Core MIB defines the basic objects that form a Tuxedo or WLE application. The Core MIB is the main information repository for controlling the operation and configuration of the application. When an application is active, the Core MIB contains groups related to the runtime activity of your application. You can use this information to monitor the behavior of your application. The Core MIB consists of the following groups.

Group Name	Description
tuxTBridgeTbl	Network connection
tuxTclientTbl	Client
tuxTconnTable	Conversation
tuxTdevice	Device
tuxTdomain	Domain information
tuxTgroupTable	Server group
tuxTmachineTable	Machine configuration attributes
tuxTmachineActive	Runtime machine characteristics
tuxTmsgTable	Message queue
tuxTqueueTable	Server queue
tuxTroutingTable	Routing criteria

Group Name	Description
tuxTsrvrTbl	Server configuration attributes
tuxTsrvrTblExt	Server runtime characteristics
tuxTsvcTbl	Service
tuxTsvcGrp	Service-group configuration attributes
tuxTlistenTbl	/T listeners
tuxTranTbl	Transaction
tuxTulogTable	Userlog
tuxTulogCtrl	Control filter MIB for tuxTulogTable
tuxTnetMapTbl	Maps logical machine IDs to network groups
tuxTnetGrpTbl	Application attributes of network groups
tuxTserverCtxtTbl	Configuration and runtime attributes of individual server dispatch contexts.
beaEventFilters	You can use these filters to define a subset of Tuxedo event notifications

tuxTBridgeTbl

This group represents runtime attributes pertaining to connectivity between logical machines that make up an application. These attribute values represent connection status and statistics. Objects in this table are accessible either through a Tuxedo SNMP agent installed on the local machin or using the `-c` option on the master machine. The index into the table consists of `tuxTBridgeLmid` and `tuxTBridgeNetworkGrpNo`. In Tuxedo 6.4, SET requests are allowed only for the `DEFAULTNET` network group, so all SET requests should use 0 for `tuxTBridgeNetworkGrpNo` in the SNMP index.

Variable Name	Object ID
tuxTBridgeLmid	.1.3.6.1.4.1.140.300.16.1.1.1
tuxTBridgeState	.1.3.6.1.4.1.140.300.16.1.1.2
tuxTBridgeCurTime	.1.3.6.1.4.1.140.300.16.1.1.3
tuxTBridgeConTime	.1.3.6.1.4.1.140.300.16.1.1.4
tuxTBridgeSuspTime	.1.3.6.1.4.1.140.300.16.1.1.5
tuxTBridgeRcvdByte	.1.3.6.1.4.1.140.300.16.1.1.6
tuxTBridgeSentByte	.1.3.6.1.4.1.140.300.16.1.1.7
tuxTBridgeRcvdNum	.1.3.6.1.4.1.140.300.16.1.1.8
tuxTBridgeSentNum	.1.3.6.1.4.1.140.300.16.1.1.9
tuxTBridgeFlowCnt	.1.3.6.1.4.1.140.300.16.1.1.10
tuxTBridgeCurEncryptBits	.1.3.6.1.4.1.140.300.16.1.1.11
tuxTBridgeNetworkGrpNo	.1.3.6.1.4.1.140.300.16.1.1.12
tuxTBridgeNetworkGrpName	.1.3.6.1.4.1.140.300.16.1.1.13

tuxTBridgeLmid

Syntax	<i>DisplayString</i> (SIZE(1..61))
Access	read-only
Description	<i>DisplayString</i> is of the format: <i>LMID1</i> [, <i>LMID2</i>] <i>LMID1</i> Is the logical machine identifier for network connection and is in the range from one to sixty-one characters. <i>LMID2</i> Is the destination logical machine identifier for network connection and is in the range from one to sixty-one characters.

tuxTBridgeState

Syntax	INTEGER { active(1), inactive(2), suspended(3), pending(4) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: { active(1) inactive(2) suspended(3) pending(4) }</p> <p>A GET operation retrieves runtime information for the selected tuxTBridgeTbl instance(s). A tuxTBridgeLmid attribute value with only one logical machine identifier matches all active connections from <i>LMID1</i> to other machines in the application. In this case, each retrieved record contains an expanded tuxTBridgeLmid attribute value with the destination LMID filled in. The following states indicate the meaning of a tuxTBridgeState returned in response to a GET request. States not listed are not returned.</p> <p>active(1)</p> <p>The connection is established and active.</p> <p>inactive(2)</p> <p>The connection is inactive. This state is only returned when status is requested on a particular connection, that is, both LMIDs are specified in the tuxTBridgeLmid attribute and the source logical machine is reachable.</p> <p>suspended(3)</p>

An established connection was terminated due to an error condition, and reconnection has been suspended for at least the amount of time indicated in the `tuxTBridgeSuspTime` attribute value.

`pending(4)`

An asynchronous connection has been requested but has not yet completed. The final outcome of the connection request has not been determined. This state is only supported on Tuxedo 6.4 or later.

`SET: {active(1)|inactive(2)|suspended(3)|pending(4)}`

A `SET` operation updates runtime information for the selected `tuxTBridgeTbl` object. The following states indicate the meaning of a `tuxTBridgeState` set in a `SET` request. States not listed cannot be set.

`active(1)`

Tuxedo 6.3 and earlier: Activate the `tuxTBridgeTbl` object by establishing a connection between the indicated logical machines. This operation fails if only one logical machine is specified, if either of the two machines is not active, or if the source logical machine is not reachable. State change allowed in the `inactive(2)` and `suspended(3)` states. Successful return leaves the object in the `active(1)` state. **Tuxedo 6.4 and later:** Activate the `tuxTBridgeTbl` instance by establishing an asynchronous connection between the indicated logical machines. This operation fails if only one machine is specified, if either of the machines is not active, or if the source machine is not reachable. When in the `pending(4)` state, the success or failure of the connection has not yet been determined. The `BRIDGE` can continue to process other events and data while the connection is outstanding. This state change is allowed in the `inactive(2)` and `suspended(3)` states. Successful return leaves the instance in the `active(1)` or `pending(4)` state.

`inactive(2)`

Deactivate the `tuxTBridgeTbl` object by closing the connection between the indicated logical machines. This operation fails if only one logical machine is specified or if the two machines are not connected. State change allowed only when in the `active(1)` state. Successful return leaves the object in the `inactive(2)` state.

`suspended(3)`

Suspend the `tuxTBridgeTbl` object by closing the connection between the indicated logical machines and by setting the `tuxTBridgeSuspTime`

parameter as indicated. State change allowed only when in the `active(1)` state. Successful return leaves the object in the `suspended(3)` state.

Note: Since the statistics reported are from the source logical machine, resetting those statistics causes them to be out of sync with the statistics reported by the destination logical machine for the same connection.

`pending(4)`

Activate the `tuxTBridgeTbl` instance by establishing an asynchronous connection between the indicated logical machines. This operation fails if only one logical machine is specified, if either of the two machines is inactive, or if the source logical machine is not reachable. When in the `pending(4)` state, the success or failure of the connection request has not yet been determined. However, the BRIDGE can continue to process other events and data while the connection request is outstanding. State change allowed in `inactive(2)` and `suspended(3)` states. Successful return leaves the instance in the `pending(4)` state. This state is supported only on 6.4 and later versions of Tuxedo systems.

tuxTBridgeCurTime

Syntax	INTEGER
Access	read-only
Description	Current time, in seconds, since 00:00:00 UTC, January 1, 1970, as returned by the <code>time(2)</code> system call on <code>tuxTBridgeLmid</code> . This attribute can be used to compute elapsed time from the following attribute values.

tuxTBridgeConTime

Syntax	INTEGER
Access	read-only
Description	Time, in seconds, that this connection has been active.

tuxTBridgeSuspTime

Syntax	INTEGER
Access	read-write

Description Time, in seconds, remaining in the suspension of this connection. After this amount of time, the connection automatically changes to a `tuxTBridgeState` of `inactive(2)` and can be activated by normal application traffic.

tuxTBridgeRcvdByte

Syntax `INTEGER`

Access `read-only`

Description Number of bytes sent from the destination logical machine to the source logical machine.

tuxTBridgeSentByte

Syntax `INTEGER`

Access `read-only`

Description Number of bytes sent from the source logical machine to the destination logical machine.

tuxTBridgeRcvdNum

Syntax `INTEGER`

Access `read-only`

Description Number of messages sent from the destination logical machine to the source logical machine.

tuxTBridgeSentNum

Syntax `INTEGER`

Access `read-only`

Description Number of messages sent from the source logical machine to the destination logical machine.

tuxTBridgeFlowCnt

Syntax	INTEGER
Access	read-only
Description	Number of times flow control has been encountered over this connection.

tuxTBridgeCurEncryptBits

Syntax	Integer {none(1), 40-bit(2), 128-bit(3), not-available(4)}
Access	read-only
Description	The current level of encryption for this link. This is negotiated between the machines when the link is established. The number specifies the encryption key length (in bits). This object is only supported on Tuxedo 6.4 and later.

tuxTBridgeNetworkGrpNo

Syntax	Integer
Access	read-only
Description	Logical network group number. When both the source and destination <code>tuxTBridgeLmid</code> machine identifiers are in the same network group, <code>tuxTBridgeTbl</code> presents all instances of related fields per network group. This object is supported only on Tuxedo 6.4 and later.

tuxTBridgeNetworkGrpName

Syntax	DisplayString
Access	read-only
Description	Logical network group name. This object is only supported on Tuxedo 6.4 or later.

tuxTclientTbl

This group represents runtime attributes of active clients within an application. These attribute values identify and track the activity of clients within a running application. Objects in this table are only accessible through a Tuxedo SNMP agent installed on the local machine.

Variable Name	Object ID
tuxTclientState	.1.3.6.1.4.1.140.300.17.1.1.1
tuxTclientBirthTime	.1.3.6.1.4.1.140.300.17.1.1.2
tuxTclientMachineId	.1.3.6.1.4.1.140.300.17.1.1.3
tuxTclientReg	.1.3.6.1.4.1.140.300.17.1.1.4
tuxTclientClntName	.1.3.6.1.4.1.140.300.17.1.1.5
tuxTclientIdleTime	.1.3.6.1.4.1.140.300.17.1.1.6
tuxTclientPid	.1.3.6.1.4.1.140.300.17.1.1.7
tuxTclientSrvGrp	.1.3.6.1.4.1.140.300.17.1.1.8
tuxTclientUsrName	.1.3.6.1.4.1.140.300.17.1.1.9
tuxTclientWsc	.1.3.6.1.4.1.140.300.17.1.1.10
tuxTclientWsh	.1.3.6.1.4.1.140.300.17.1.1.11
tuxTclientWshClientId	.1.3.6.1.4.1.140.300.17.1.1.12
tuxTclientRelease	.1.3.6.1.4.1.140.300.17.1.1.13
tuxTclientWsProto	.1.3.6.1.4.1.140.300.17.1.1.14
tuxTclientNumConv	.1.3.6.1.4.1.140.300.17.1.1.15
tuxTclientNumDeque	.1.3.6.1.4.1.140.300.17.1.1.16
tuxTclientNumEnque	.1.3.6.1.4.1.140.300.17.1.1.17

Variable Name	Object ID
tuxTclientNumPost	.1.3.6.1.4.1.140.300.17.1.1.18
tuxTclientNumReq	.1.3.6.1.4.1.140.300.17.1.1.19
tuxTclientNumSubscribe	.1.3.6.1.4.1.140.300.17.1.1.20
tuxTclientNumTran	.1.3.6.1.4.1.140.300.17.1.1.21
tuxTclientNumTranAbt	.1.3.6.1.4.1.140.300.17.1.1.22
tuxTclientNumTranCmt	.1.3.6.1.4.1.140.300.17.1.1.23
tuxTclientCmtRet	.1.3.6.1.4.1.140.300.17.1.1.24
tuxTclientCurConv	.1.3.6.1.4.1.140.300.17.1.1.26
tuxTclientCurReq	.1.3.6.1.4.1.140.300.17.1.1.27
tuxTclientCurTime	.1.3.6.1.4.1.140.300.17.1.1.28
tuxTclientLastGrp	.1.3.6.1.4.1.140.300.17.1.1.29
tuxTclientNaddr	.1.3.6.1.4.1.140.300.17.1.1.30
tuxTclientNotify	.1.3.6.1.4.1.140.300.17.1.1.31
tuxTclientNumUnSol	.1.3.6.1.4.1.140.300.17.1.1.32
tuxTclientRpid	.1.3.6.1.4.1.140.300.17.1.1.33
tuxTclientTimeLeft	.1.3.6.1.4.1.140.300.17.1.1.34
tuxTclientTimeStart	.1.3.6.1.4.1.140.300.17.1.1.36
tuxTclientTranLev	.1.3.6.1.4.1.140.300.17.1.1.37
tuxTclientId	.1.3.6.1.4.1.140.300.17.1.1.38
tuxTclientContextID	.1.3.6.1.4.1.140.300.17.1.1.50

tuxTclientState

Syntax `INTEGER { active(1), suspended(2), dead(3) }`

Access `read-write`

Description The values for GET and SET operations are as follows:

GET: `{active(1) | suspended(2) | dead(3)}`

A GET operation retrieves runtime information for the selected `tuxTclientTbl` instance(s). Note that client information is kept in local bulletin board tables only. Therefore, for maximum performance, inquiries on client status should be restricted, using key fields as much as possible. The following states indicate the meaning of a `tuxTclientState` returned in response to a GET request. States not listed are not returned.

`active(1)`

`tuxTclientTbl` instance active. This is not an indication of whether the client is idle or busy. A non-0 value retrieved for either the `tuxTclientCurConv` attribute or the `tuxTclientCurReq` attribute indicates a busy client.

`suspended(2)`

`tuxTclientTbl` instance active and suspended from making further service requests (`tpcall(3)` or `tpacall(3)`) and from initiating further conversations (`tpconnect(3)`). See SET `suspended(2)` below for details.

`dead(3)`

`tuxTclientTbl` instance identified as active in the bulletin board but currently not running due to an abnormal death. This state exists only until the BBL local to the client notices the death and takes action to clean up the client's bulletin board resources.

SET: `{active(1) | suspended(2) | dead(3)}`

A SET operation updates runtime information for the selected `tuxTclientTbl` object. The following states indicate the meaning of a `tuxTclientState` set in a SET request. States not listed cannot be set.

`active(1)`

Activate a `suspended(2)` `tuxTclientTbl` instance. State change allowed only when in the `suspended(2)` state. Successful return leaves the object in the `active(1)` state.

`suspended(2)`

Suspend the `tuxTclientTbl` instance from making service requests (`tpcall(3)` or `tpacall(3)`), initiating conversations (`tpconnect(3)`), beginning transactions (`tpbegin(3)`), and enqueueing new requests (`tpenqueue(3)`). Clients within a transaction are permitted to make these calls until they abort or commit the current transaction, at which time the clients become suspended. Invocations of these routines result in a `TPESYSTEM` error return and a system log message being generated that indicates the situation. State change is allowed only when the object is in the `active(1)` state. Successful return leaves the object in the `suspended(2)` state.

`dead(3)`

Abortively deactivate the `tuxTclientTbl` instance. State change is allowed only when the object is in the `active(1)` or `suspended(2)` state. The recommended method for deactivating clients is to first suspend them, and then to abortively deactivate them by setting the state to `dead(3)`. Successful return leaves the object in the `dead(3)` state

Note: Workstation handlers (`tuxTclientWsh == yes(1)`) cannot be set to a state of `dead(3)`. The system might not be able to kill the client, due to platform or signaling restrictions. In this case, a native client is abortively terminated at its next access to ATMI, and a workstation client's connection to a WSH is preemptively torn down.

tuxTclientBirthTime

Syntax	INTEGER
Access	read-only
Description	Client identifier. The data in this field should not be interpreted directly by the end user except for equality comparison.

tuxTclientMachineld

Syntax INTEGER

Access	read-only
Description	Client identifier. The data in this field should not be interpreted directly by the end user except for equality comparison.

tuxTclientReg

Syntax	INTEGER
Access	read-only
Description	Client identifier. The data in this field should not be interpreted directly by the end user except for equality comparison.

tuxTclientClnName

Syntax	<i>DisplayString</i> (SIZE(0..30))
Access	read-only
Description	Client name associated with client at <code>tpinit(3)</code> time through the <code>clname</code> element of the TPINIT structure.

tuxTclientIdleTime

Syntax	INTEGER
Access	read-only
Description	Approximate amount of time, in seconds, since this client last interacted with the system through an ATMI call. This value is accurate to within <code>tuxTdomainScanUnit</code> (see the <code>tuxTdomain</code> group) seconds. When specified as a key field, a positive value indicates that all clients with idle times of at least the indicated value match, a negative value indicates that all clients with no more than the indicated value match, and a 0 value matches all clients.

tuxTclientPid

Syntax	INTEGER
Access	read-only

Description	Process identifier of client. Note that for workstation clients, this identifier indicates the workstation handler through which the workstation client is connected. A negative number can be specified on a GET operation for the purpose of retrieving client information for the calling process. If the calling process is not a client, then an error is returned.
-------------	--

tuxTclientSrvGrp

Syntax	<i>DisplayString</i> (SIZE(0..30))
Access	read-only
Description	Server group with which the client is associated. This information is set through the grpname element of the TPINIT structure at tpinit(3) time.

tuxTclientUsrName

Syntax	<i>DisplayString</i> (SIZE(0..30))
Access	read-only
Description	User name associated with client at tpinit(3) time through the username element of the TPINIT structure.

tuxTclientWsc

Syntax	INTEGER { yes(1), no(2) }
Access	read-only
Description	If this attribute is set to yes(1), the indicated client is logged in to the application from a remote workstation.

tuxTclientWsh

Syntax	INTEGER { yes(1), no(2) }
Access	read-only
Description	Workstation handler. If this attribute is set to yes(1), the indicated client is a workstation handler process.

tuxTclientWshClientId

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Client identifier for the associated workstation handler (WSH) if this client is a workstation client (<code>tuxTclientWsc == yes(1)</code>); otherwise, this attribute is returned as a 0-length string.

tuxTclientRelease

Syntax	INTEGER
Access	read-only
Description	The Tuxedo System/T major protocol release number for the machine where the client is running. This can be different from the <code>tuxTmachineSWrelease</code> for the same machine. Note that for /WS clients (<code>tuxTclientWsc == yes(1)</code>), this value can be different from the major release associated with the application administered machine through which the /WS client accesses the application.

tuxTclientWsProto

Syntax	INTEGER
Access	read-only
Description	The Tuxedo System/T /WS protocol version number for a workstation client. This value is changed with each update to the /WS protocol. A value of 0 is returned for this attribute when it is associated with non-/WS clients (<code>tuxTclientWsc == no(2)</code>).

tuxTclientNumConv

Syntax	INTEGER
Access	read-only
Description	Number of conversations initiated by this client through <code>tpconnect(3)</code> .

tuxTclientNumDeque

Syntax	INTEGER
Access	read-only
Description	Number of dequeue operations initiated by this client through <code>tpdequeue(3)</code> .

tuxTclientNumEnque

Syntax	INTEGER
Access	read-only
Description	Number of enqueue operations initiated by this client through <code>tpenqueue(3)</code> .

tuxTclientNumPost

Syntax	INTEGER
Access	read-only
Description	Number of postings initiated by this client through <code>tppost(3)</code> .

tuxTclientNumReq

Syntax	INTEGER
Access	read-only
Description	Number of requests made by this client through <code>tpcall(3)</code> or <code>tpacall(3)</code> .

tuxTclientNumSubscribe

Syntax	INTEGER
Access	read-only
Description	Number of subscriptions made by this client through <code>tpsubscribe(3)</code> .

tuxTclientNumTran

Syntax	INTEGER
Access	read-only
Description	Number of transactions begun by this client.

tuxTclientNumTranAbt

Syntax	INTEGER
Access	read-only
Description	Number of transactions aborted by this client.

tuxTclientNumTranCmt

Syntax	INTEGER
Access	read-only
Description	Number of transactions committed by this client.

tuxTclientCmtRet

Syntax	INTEGER { complete(1) logged(2) }
Access	read-only
Description	Setting of the TP_COMMIT_CONTROL characteristic for this client. See the description of the System/T ATMI function <code>tpscmt(3)</code> for details on this characteristic.

tuxTclientCurConv

Syntax	INTEGER
Access	read-only
Description	Number of conversations initiated by this client through <code>tpconnect(3)</code> that are still active.

tuxTclientCurReq

Syntax	INTEGER
Access	read-only
Description	Number of requests initiated by this client through <code>tpcall(3)</code> or <code>tpacall(3)</code> that are still active.

tuxTclientCurTime

Syntax	INTEGER
Access	read-only
Description	Current time, in seconds, since 00:00:00 UTC, January 1, 1970, as returned by the <code>time(2)</code> system call on the local host. This attribute can be used to compute elapsed time from the <code>tuxTclientTimeStart</code> attribute value.

tuxTclientLastGrp

Syntax	INTEGER
Access	read-only
Description	Server group number of the last service request made or conversation initiated from this client.

tuxTclientNaddr

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	For workstation clients, this attribute indicates the network address of the client. Network addresses with unprintable characters are converted to the “0x...” network address format as described in the <code>tuxTmachineNaddr</code> attribute. Non-workstation clients have a 0-length string associated with them for this attribute value.

Note: The ability of the system to provide this information is determined by the transport provider in use. In some cases, workstation clients cannot have addresses associated with them if the provider does not make this information available.

tuxTclientNotify

Syntax	INTEGER { dipin(1) signal(2) ignore(3) }
Access	read-only
Description	Setting of the notification characteristic for this client. See the <code>tuxTdomain</code> group description of this attribute for more details.

tuxTclientNumUnSol

Syntax	INTEGER
Access	read-only
Description	Number of unsolicited messages queued for this client that are awaiting processing.

tuxTclientRpid

Syntax	INTEGER
Access	read-only
Description	UNIX system message queue identifier for the client's reply queue.

Note: This is a UNIX system specific attribute that might not be returned if the platform on which the application is being run is not UNIX-based.

tuxTclientTimeLeft

Syntax	INTEGER
Access	read-only
Description	Time left, in seconds, for this client to receive the reply for which it is currently waiting before it times out. This timeout can be a transactional timeout or a blocking timeout.

tuxTclientTimeStart

Syntax	INTEGER
Access	read-only
Description	Time, in seconds, since 00:00:00 UTC, January 1, 1970, as returned by the <code>time(2)</code> system call on local host, since the client joined the application.

tuxTclientTranLev

Syntax	INTEGER
Access	read-only
Description	Current transaction level for this client. 0 indicates that the client is not currently involved in a transaction.

tuxTclientId

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Client Identifier.

tuxTclientContextID

Syntax	INTEGER (-2..29999)
Access	read-only
Status	mandatory
Description	Identifier for this particular application association.

tuxTconnTable

This group represents runtime attributes of active conversations within an application. Objects in this table are only accessible through a Tuxedo SNMP agent installed on the local machine. All objects in this MIB group are local attributes, that is, values for these objects correspond to the local host only where the Tuxedo agent is running. Thus, the user needs to run an instance of the Tuxedo agent on every node for which these values are of interest. The index into this table is `tuxTconnSerNo`.

Variable Name	Object ID
tuxTconnSerNo	.1.3.6.1.4.1.140.300.18.1.1.1
tuxTconnState	.1.3.6.1.4.1.140.300.18.1.1.2
tuxTconnSvcName	.1.3.6.1.4.1.140.300.18.1.1.3
tuxTconnClientId	.1.3.6.1.4.1.140.300.18.1.1.4
tuxTconnOgrpNo	.1.3.6.1.4.1.140.300.18.1.1.5
tuxTconnOlmid	.1.3.6.1.4.1.140.300.18.1.1.6
tuxTconnOpid	.1.3.6.1.4.1.140.300.18.1.1.7
tuxTconnOsndcnt	.1.3.6.1.4.1.140.300.18.1.1.8
tuxTconnOsrvId	.1.3.6.1.4.1.140.300.18.1.1.9
tuxTconnSgrpNo	.1.3.6.1.4.1.140.300.18.1.1.10
tuxTconnSlmid	.1.3.6.1.4.1.140.300.18.1.1.11
tuxTconnSpid	.1.3.6.1.4.1.140.300.18.1.1.12
tuxTconnSsndcnt	.1.3.6.1.4.1.140.300.18.1.1.13
tuxTconnSsrvId	.1.3.6.1.4.1.140.300.18.1.1.14

tuxTconnSerNo

Syntax	INTEGER
Access	read-only
Description	A running number as an index for tuxTconnTable.

tuxTconnState

Syntax	INTEGER { active(1) }
Access	read-only
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET:</p> <p>A GET operation retrieves runtime information for the selected tuxTconnTable instance(s). The following state indicates the meaning of a tuxTconnState returned in response to a GET request. States not listed are not returned.</p> <p>active(1)</p> <p>The object returned reflects one or both sides of an active conversation within the application.</p> <p>SET:</p> <p>SET operations are not permitted for this class.</p>

tuxTconnSvcName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-only
Description	Service name of the conversational service invoked by the originator and processed by the subordinate.

tuxTconnClientId

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Client identifier. The data in this field should not be interpreted directly by the end user except for equality comparison.

tuxTconnOgrpNo

Syntax	INTEGER (1..30001)
Access	read-only
Description	Server group number for the originator of the conversation. If the originator is a client, then 30,000 is returned as the value for this attribute.

tuxTconnOlmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Logical machine identifier that indicates where the originator is running, or (in the case of /WS clients) is accessing the application .

tuxTconnOpid

Syntax	INTEGER
Access	read-only
Description	Process identifier for the originator of the conversation.

tuxTconnOsndcnt

Syntax	INTEGER
Access	read-only
Description	Number of <code>tpsend(3)</code> calls made by the originator.

tuxTconnOsrvid

Syntax	INTEGER (1..30001)
Access	read-only
Description	Server identifier for the originator of the conversation.

tuxTconnSgrpNo

Syntax	INTEGER (1..30001)
Access	read-only
Description	Server group number for the subordinate of the conversation. If the originator is a client, then 30,000 is returned as the value for this attribute.

tuxTconnSImid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Logical machine identifier that indicates where the subordinate is running or, (in the case of /WS clients) is accessing the application.

tuxTconnSpid

Syntax	INTEGER
Access	read-only
Description	Process identifier for the subordinate in the conversation.

tuxTconnSsndcnt

Syntax	INTEGER
Access	read-only
Description	Number of <code>tpsend(3)</code> calls made by the subordinate.

tuxTconnSsrvid

Syntax	INTEGER (1..30001)
Access	read-only
Description	Server identifier for the subordinate in the conversation.

tuxTdevice

Variable Name	Object ID
tuxTdeviceTbl	.1.3.6.1.4.1.140.300.19.1
tuxTwhichCfgDev	.1.3.6.1.4.1.140.300.19.2

tuxTdeviceTbl

This group represents configuration and runtime attributes of raw disk slices or UNIX system files being used to store Tuxedo System/T device lists. This class allows for the creation and deletion of device list entries within a raw disk slice or UNIX system file. Objects in this table are only accessible through a Tuxedo SNMP agent installed on the local machine. To create a new row in this table, the user needs to send a SET request, with at least a value for tuxTdevSize. The index into this table is tuxTdevCfgDev and tuxTdevIndex.

Variable Name	Object ID
tuxTdevLmid	.1.3.6.1.4.1.140.300.19.1.1.1
tuxTdevCfgDev	.1.3.6.1.4.1.140.300.19.1.1.2
tuxTdeviceName	.1.3.6.1.4.1.140.300.19.1.1.3
tuxTdevOffset	.1.3.6.1.4.1.140.300.19.1.1.4
tuxTdevSize	.1.3.6.1.4.1.140.300.19.1.1.5
tuxTdevIndex	.1.3.6.1.4.1.140.300.19.1.1.6
tuxTdevState	.1.3.6.1.4.1.140.300.19.1.1.7

tuxTdevLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Logical machine identifier where the device is located. Note that this attribute can be used as a key field in both unbooted and booted applications as long as they are already configured (that is, at least one tuxTmachineTable instance exists). It is required as a key field on SET operations when they are accessing a booted application. If specified when accessing the tuxTdeviceTbl table in an unconfigured application, this attribute is ignored.

Note: This object can be set only during row creation.

tuxTdevCfgDev

Syntax	<i>DisplayString</i> (SIZE(2..64))
Access	read-write
Description	Absolute pathname of the file or device where the Tuxedo System/T filesystem is stored or is to be stored.

Note: This object can be set only during row creation.

tuxTdeviceName

Syntax	<i>DisplayString</i> (SIZE(2..64))
Access	read-write
Description	Absolute pathname of the device list entry.

Note: This object can be set only during row creation.

tuxTdevOffset

Syntax	INTEGER
Access	read-write
Description	The offset, in blocks, at which space on this <code>tuxTdevice</code> begins for use within the Tuxedo System/T VTOC specified by <code>tuxTdevCfgDev</code> .

Note: This object can be set only during row creation.

tuxTdevSize

Syntax	INTEGER
Access	read-write
Description	The size in pages of the disk area to be used for the device list entry.

Note: This attribute can be set only in conjunction with row creation.

Note: This object can be set only during row creation.

tuxTdevIndex

Syntax	INTEGER
Access	read-only
Description	Device index for <code>tuxTdevice</code> within the device list addressed by <code>tuxTdevCfgDev</code> . This attribute value is used for identification purposes only in getting and setting attribute values that relate to particular devices within a Tuxedo System/T filesystem.

tuxTdevState

Syntax	INTEGER { valid(1) invalid(2) re-init(3) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: {valid(1)}</p> <p>A GET operation retrieves runtime information for the selected <code>tuxTdeviceTbl</code> instance(s). The following state indicates the meaning of a <code>tuxTdevState</code> returned in response to a GET request. States not listed are not returned.</p> <p>valid(1)</p> <p>The Tuxedo System/T filesystem indicated by <code>tuxTdevCfgDev</code> exists and contains a valid device list. <code>tuxTdevice</code> is a valid device within that filesystem with the device index <code>tuxTdevIndex</code>.</p> <p>SET: {invalid(2) re-init(3)}</p> <p>A SET operation updates information for the selected <code>tuxTdeviceTbl</code> instance or adds the indicated object. The following states indicate the meaning of a <code>tuxTdevState</code> set in a SET request. States not listed cannot be set.</p> <p>invalid(2)</p> <p>Delete <code>tuxTdeviceTbl</code> instance for application. State change is allowed only when the object is in the <code>valid(1)</code> state. Successful return leaves the object in the <code>invalid(2)</code> state. Note that <code>tuxTdevIndex 0</code> is special and must be deleted last.</p> <p>re-init(3)</p>

Re-initializes a valid device.

tuxTwhichCfgDev

Syntax *DisplayString* (SIZE(2..64))

Access read-write

Description The value of this object determines the device for which `tuxTdeviceTbl` returns configuration and runtime information.

The default value of this object is the TUXCONFIG file for the current domain.

tuxTdomain

The following objects of `tuxTdomain` represent global application attributes for the domain to which the Tuxedo SNMP Agent is currently connected. These object values serve to identify, customize, size, secure, and tune a Tuxedo System/T application. Many of the object values represented here serve as application defaults for other groups represented in this MIB.

There is exactly one instance of the `tuxTdomain` group for each application.

Variable Name	Object ID
<code>tuxTdomainKey</code>	<code>.1.3.6.1.4.1.140.300.3.1</code>
<code>tuxTdomainMaster</code>	<code>.1.3.6.1.4.1.140.300.3.2</code>
<code>tuxTdomainModel</code>	<code>.1.3.6.1.4.1.140.300.3.3</code>
<code>tuxTdomainState</code>	<code>.1.3.6.1.4.1.140.300.3.4</code>
<code>tuxTdomainID</code>	<code>.1.3.6.1.4.1.140.300.3.5</code>
<code>tuxTdomainUID</code>	<code>.1.3.6.1.4.1.140.300.3.7</code>
<code>tuxTdomainGID</code>	<code>.1.3.6.1.4.1.140.300.3.8</code>
<code>tuxTdomainPerm</code>	<code>.1.3.6.1.4.1.140.300.3.9</code>
<code>tuxTdomainMask</code>	<code>.1.3.6.1.4.1.140.300.3.10</code>
<code>tuxTdomainMaxAccessers</code>	<code>.1.3.6.1.4.1.140.300.3.11</code>
<code>tuxTdomainMaxConv</code>	<code>.1.3.6.1.4.1.140.300.3.12</code>
<code>tuxTdomainMaxGTT</code>	<code>.1.3.6.1.4.1.140.300.3.13</code>
<code>tuxTdomainMaxBufsType</code>	<code>.1.3.6.1.4.1.140.300.3.14</code>
<code>tuxTdomainMaxBufType</code>	<code>.1.3.6.1.4.1.140.300.3.15</code>
<code>tuxTdomainMaxDRT</code>	<code>.1.3.6.1.4.1.140.300.3.16</code>
<code>tuxTdomainMaxGroups</code>	<code>.1.3.6.1.4.1.140.300.3.17</code>

Variable Name	Object ID
tuxTdomainMaxMachines	.1.3.6.1.4.1.140.300.3.18
tuxTdomainMaxQueues	.1.3.6.1.4.1.140.300.3.19
tuxTdomainMaxRFT	.1.3.6.1.4.1.140.300.3.20
tuxTdomainMaxRTData	.1.3.6.1.4.1.140.300.3.21
tuxTdomainMaxServers	.1.3.6.1.4.1.140.300.3.22
tuxTdomainMaxServices	.1.3.6.1.4.1.140.300.3.23
tuxTdomainMaxACLgroups	.1.3.6.1.4.1.140.300.3.24
tuxTdomainCMTRET	.1.3.6.1.4.1.140.300.3.25
tuxTdomainLoadBalance	.1.3.6.1.4.1.140.300.3.26
tuxTdomainNotify	.1.3.6.1.4.1.140.300.3.27
tuxTdomainSystemAccess	.1.3.6.1.4.1.140.300.3.28
tuxTdomainOptions	.1.3.6.1.4.1.140.300.3.29
tuxTdomainSignal	.1.3.6.1.4.1.140.300.3.30
tuxTdomainSecurity	.1.3.6.1.4.1.140.300.3.31
tuxTdomainAuthsvc	.1.3.6.1.4.1.140.300.3.33
tuxTdomainScanUnit	.1.3.6.1.4.1.140.300.3.34
tuxTdomainBBLQuery	.1.3.6.1.4.1.140.300.3.35
tuxTdomainBlockTime	.1.3.6.1.4.1.140.300.3.36
tuxTdomainDBBLWait	.1.3.6.1.4.1.140.300.3.37
tuxTdomainSanityScan	.1.3.6.1.4.1.140.300.3.38
tuxTdomainCurDRT	.1.3.6.1.4.1.140.300.3.39
tuxTdomainCurGroups	.1.3.6.1.4.1.140.300.3.40
tuxTdomainCurMachines	.1.3.6.1.4.1.140.300.3.41

Variable Name	Object ID
tuxTdomainCurQueues	.1.3.6.1.4.1.140.300.3.42
tuxTdomainCurRFT	.1.3.6.1.4.1.140.300.3.43
tuxTdomainCurRTdata	.1.3.6.1.4.1.140.300.3.44
tuxTdomainCurServers	.1.3.6.1.4.1.140.300.3.45
tuxTdomainCurServices	.1.3.6.1.4.1.140.300.3.46
tuxTdomainCursType	.1.3.6.1.4.1.140.300.3.47
tuxTdomainCurType	.1.3.6.1.4.1.140.300.3.48
tuxTdomainHwDRT	.1.3.6.1.4.1.140.300.3.49
tuxTdomainHwGroups	.1.3.6.1.4.1.140.300.3.50
tuxTdomainHwMachines	.1.3.6.1.4.1.140.300.3.51
tuxTdomainHwQueues	.1.3.6.1.4.1.140.300.3.52
tuxTdomainHwRFT	.1.3.6.1.4.1.140.300.3.53
tuxTdomainHwRTdata	.1.3.6.1.4.1.140.300.3.54
tuxTdomainHwServers	.1.3.6.1.4.1.140.300.3.55
tuxTdomainHwServices	.1.3.6.1.4.1.140.300.3.56
tuxTdomainMaxNetGroups	.1.3.6.1.4.1.140.300.3.58
wleMaxObjects	.1.3.6.1.4.1.140.300.3.63
wleMaxInterfaces	.1.3.6.1.4.1.140.300.3.68
tuxTdomainSignatureAhead	.1.3.6.1.4.1.140.300.3.70
wleCurInterfaces	.1.3.6.1.4.1.140.300.3.73
wleHwInterfaces	.1.3.6.1.4.1.140.300.3.78
tuxTdomainSignatureBehind	.1.3.6.1.4.1.140.300.3.80
tuxTdomainEncryptionRequired	.1.3.6.1.4.1.140.300.3.90

Variable Name	Object ID
tuxTdomainSignatureRequired	.1.3.6.1.4.1.140.300.3.100

tuxTdomainKey

Syntax	INTEGER (32769 .. 262143)
Access	read-write
Description	Numeric key for the well-known address in a Tuxedo System/T bulletin board. In a single processor environment, this key “names” the bulletin board. In a multiple processor or LAN environment, this key names the message queue of the DBBL. In addition, this key is used as a basis for deriving the names of resources other than the well-known address, such as the names for bulletin boards throughout the application.

tuxTdomainMaster

Syntax	<i>DisplayString</i> (SIZE (1..30))
Access	read-write
Description	<i>DisplayString</i> is in format: <i>LMID1</i> [, <i>LMID2</i>] <i>LMID1</i> Is the master logical machine identifier and is in the range from one to thirty characters. <i>LMID2</i> Is the backup logical machine identifier and is in the range from one to thirty characters. The master identifier (<i>LMID1</i>) must correspond to the local machine for inactive applications. <i>single-machine</i> (1) mode applications (see <i>tuxTdomainModel</i> below) can set only the master logical machine identifier. Modifications to this attribute value in an active <i>multi-machine</i> (2) application (see <i>tuxTdomainModel</i> below) have the following semantics.

Assuming current active master LMID A, current backup master LMID B, and secondary LMIDs C, D, ..., the following scenarios define the semantics of permitted changes to the `tuxTdomainMaster` attribute in a running `multi-machine(2)` mode application.

A,B -> B,A — Master migration from A to B. A,B -> A,C — Change backup master LMID designation to C.

Note that master migration can be either orderly or partitioned. Orderly migration takes place when the master machine is active and reachable. Otherwise, partitioned migration takes place. All newly established or re-established network connections verify that the two sites connecting share a common view of where the master machine is located. Otherwise, the connection is refused and an appropriate log message is generated.

The master and backup machines in an active application must always have a Tuxedo System/T release number greater than or equal to all other machines active in the application. The master and backup machines must be of the same release. Modifications to the `tuxTdomainMaster` attribute must preserve this relationship.

tuxTdomainModel

Syntax	<code>INTEGER { single-machine(1) multi-machine(2) }</code>
Access	read-write
Description	The configuration type. <code>single-machine(1)</code> Specifies a single machine configuration; only one <code>tuxTmachineTable</code> object can be specified. <code>multi-machine(2)</code> Specifies a multi-machine or network configuration; this must be specified if a networked application is being defined.

tuxTdomainState

Syntax	<code>INTEGER { active(1) inactive(2) forcible-inactive(3) }</code>
Access	read-write

Description The values for GET and SET operations are as follows:

GET: {active(1)|inactive(2)}

A GET operation retrieves configuration and runtime information for the `tuxTdomain` group. The following states indicate the meaning of a `tuxTdomainState` returned in response to a GET request. States not listed are not returned.

active(1)

`tuxTdomain` group is defined and the master machine is active.

inactive(2)

`tuxTdomain` group is defined and application is inactive.

SET: active(1)|inactive(2)|forcible-inactive(3)

A SET operation updates configuration and runtime information for the `tuxTdomain` group. The following states indicate the meaning of a `tuxTdomainState` set in a SET request. States not listed cannot be set.

active(1)

Activate administrative processes (DBBL, BBL, and so on) on the master machine. A state change is allowed only when the object is in the `inactive(2)` state. Successful return leaves the object in the `active(1)` state.

inactive(2)

Deactivate administrative processes (DBBL, BBL, and so on) on the master machine. A state change is allowed only when the object is in the `active(1)` state. Successful return leaves the object in the `inactive(2)` state. To do a complete shutdown of the application, you must first make all groups inactive. (See `tuxTgroupState`.) This state transition fails if any application servers or clients are still attached to the domain. To ignore any running clients or application servers, set to `forcible-inactive(3)` as explained below.

forcible-inactive(3)

Forcibly deactivate administrative processes (DBBL, BBL, and so on) on the master machine. Attached clients are ignored for the purpose of determining if shutdown should be allowed. State change is allowed only when the object is in the `active(1)` state. Successful return leaves the object in the

`inactive(2)` state. You need to restart any clients before they can be used to process services after this state transition.

tuxTdomainID

Syntax	<i>DisplayString</i> (SIZE (0..30))
Access	read-write
Description	Domain identification string.

tuxTdomainUID

Syntax	INTEGER
Access	read-write
Description	Default attribute setting for newly configured objects in the <code>tuxTmachineTable</code> group.

Note: Changes to this attribute do not affect active or already configured `tuxTmachineTable` instances.

tuxTdomainGID

Syntax	INTEGER
Access	read-write
Description	Default attribute setting for newly configured objects in the <code>tuxTmachineTable</code> group.

Note: Changes to this attribute do not affect active or already configured `tuxTmachineTable` instances.

tuxTdomainPerm

Syntax	<i>DisplayString</i> (SIZE(1..9))
Access	read-write

Description Default attribute setting for newly configured objects in the `tuxTmachineTable` group.

Note: Changes to this attribute do not affect active or already configured `tuxTmachineTable` instances.

tuxTdomainMask

Syntax *DisplayString* (SIZE(1..9))

Access read-write

Description Attribute access mask. User type/access mode combinations specified by this attribute value are no longer allowed for all class/attribute combinations defined in `TM_MIB(5)`. For example, a setting of 0003 disallows all updates to users other than the administrator or the operator. The value of this object should be provided as an octal number — 0 through 0777.

tuxTdomainMaxAccessers

Syntax INTEGER (1..32767)

Access read-write

Description Default attribute setting for newly configured objects in the `tuxTmachineTable` group.

Note: Changes to this attribute do not affect active or already configured `tuxTmachineTable` instances.

tuxTdomainMaxConv

Syntax INTEGER (0..32767)

Access read-write

Description Default attribute setting for newly configured objects in the `tuxTmachineTable` group.

Note: Changes to this attribute do not affect active or already configured `tuxTmachineTable` instances.

tuxTdomainMaxGTT

Syntax	INTEGER (0..32767)
Access	read-write
Description	Default attribute setting for newly configured objects in the tuxTmachineTable group.
Note:	Changes to this attribute do not affect active or already configured tuxTmachineTable instances.

tuxTdomainMaxBufsType

Syntax	INTEGER (1..32767)
Access	read-write
Description	Maximum number of buffer subtypes that can be accommodated in the bulletin board buffer subtype table.

tuxTdomainMaxBufType

Syntax	INTEGER (1..32767)
Access	read-write
Description	Maximum number of buffer types that can be accommodated in the bulletin board buffer type table.

tuxTdomainMaxDRT

Syntax	INTEGER (0..32767)
Access	read-write
Description	Maximum number of routing table entries that can be accommodated in the bulletin board routing table. One entry per tuxTroutingTable group object is required. Additional entries should be allocated to allow for runtime growth.

tuxTdomainMaxGroups

Syntax	INTEGER (100..32767)
Access	read-write
Description	Maximum number of server groups that can be accommodated in the bulletin board server group table.
Note: Tuxedo System/T Release 4.2.2 and earlier sites have a fixed setting of 100 for this attribute. Interoperability with these sites requires that no more than 100 server group entries be in use at any time. Release 4.2.2 and earlier sites are not allowed to join an application that has more than 100 defined server groups. Additionally, applications that already include Release 4.2.2 or earlier sites are not allowed to add server groups beyond 100.	

tuxTdomainMaxMachines

Syntax	INTEGER (256..8190)
Access	read-write
Description	Maximum number of machines that can be accommodated in the bulletin board machine table.
Note: Tuxedo System/T Release 4.2.2 has a fixed setting of 256 for this attribute. Releases prior to Release 4.2.2 have a fixed setting of 50 for this attribute. Interoperability with Release 4.2.2 and earlier sites requires that no more than the lowest fixed setting number of machine table entries be in use at any time. Release 4.2.2 sites are not allowed to join an application that has more than 256 defined machines. Pre-Release 4.2.2 sites are not allowed to join an application that has more than 50 defined machines. Additionally, applications that already include active Release 4.2.2 or earlier sites are not allowed to add machines beyond the lowest applicable limit.	

tuxTdomainMaxQueues

Syntax	INTEGER (1..8191)
Access	read-write
Description	Maximum number of queues to be accommodated in the bulletin board queue table. Note: Release 4.2.2 and earlier sites can join an active application only if the setting for tuxTdomainMaxQueues is equal to the setting for tuxTdomainMaxServers.

tuxTdomainMaxRFT

Syntax	INTEGER (0..32767)
Access	read-write
Description	Maximum number of routing criteria range table entries to be accommodated in the bulletin board range criteria table. One entry per individual range within a tuxTroutingRanges specification is required plus one additional entry per tuxTroutingTable class object. Additional entries should be allocated to allow for runtime growth.

tuxTdomainMaxRTData

Syntax	INTEGER (0..32760)
Access	read-write
Description	Maximum string pool space to be accommodated in the bulletin board string pool table. Strings and carrays specified within tuxTroutingRanges values are stored in the string pool. Additional space should be allocated to allow for runtime growth.

tuxTdomainMaxServers

Syntax	INTEGER (1..8191)
Access	read-write
Description	Maximum number of servers to be accommodated in the bulletin board server table. Allowances should be made in setting this attribute for system supplied administrative servers. Administration of each System/T site adds approximately one server. Additionally, if TMSs are specified for any server groups (see <code>tuxTgroupTMSname</code>), they are booted along with their server group and should be accounted for in setting <code>tuxTdomainMaxServers</code> .

tuxTdomainMaxServices

Syntax	INTEGER (1..32767)
Access	read-write
Description	Maximum number of services to be accommodated in the bulletin board service table. Allowances should be made in setting this attribute for system supplied servers that offer services for administrative purposes. Administration of each System/T site adds approximately five services. Other administrative components such as /WS, /Q, and /DM can also add administrative services that should be accounted for.

tuxTdomainMaxACLgroups

Syntax	INTEGER (1..16384)
Access	read-write
Description	Maximum number of group identifiers that can be used for checking ACL permissions. The maximum group identifier that can be defined is <code>tuxTdomainMaxACLgroups - 1</code> .

tuxTdomainCMTRET

Syntax	INTEGER { complete(1) logged(2) }
Access	read-write
Description	Initial setting of the TP_COMMIT_CONTROL characteristic for all client and server processes in a System/T application. <code>logged(2)</code> initializes the TP_COMMIT_CONTROL characteristic to TP_CMT_LOGGED; otherwise, it is initialized to TP_CMT_COMPLETE. See the description of the System/T ATMI function <code>tpscmt(3)</code> for details on the setting of this characteristic.

Note: Runtime modifications to this attribute do not affect active clients and servers.

tuxTdomainLoadBalance

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	<code>yes(1)</code>

Load balancing is on.

`no(2)`.

Load balancing is off.

Note: Runtime modifications to this attribute do not affect active clients and servers.

tuxTdomainNotify

Syntax	INTEGER { dipin(1) signal(2) ignore(3) }
Access	read-write
Description	Default notification detection method used by the system for unsolicited messages sent to client processes. This default value can be overridden on a per-client basis using the appropriate <code>tpinit(3)</code> flag value. Note that once unsolicited messages are detected, they are made available to the application through the application defined unsolicited message handling routine identified through the <code>tpsetunsol(3)</code> function.

`dipin(1)`

The value `dipin(1)` specifies that dip-in-based notification detection should be used. This means that the system only detects notification messages on behalf of a client process while within ATMI calls. The point of detection within any particular ATMI call is not defined by the system, and dip-in detection does not interrupt blocking system calls. `dipin(1)` is the default notification detection method.

`signal(2)`

The value `signal(2)` specifies that signal-based notification detection should be used. This means that the system sends a signal to the target client process after the notification message has been made available. The system installs a signal catching routine on behalf of clients that select this method of notification.

`ignore(3)`

The value `ignore(3)` specifies that by default, notification messages are to be ignored by application clients. This would be appropriate in applications where only clients that request notification at `tpinit(3)` time should receive unsolicited messages.

Note: Runtime modifications to this attribute do not affect active clients. All signaling of client processes is done by administrative system processes and not by application processes. Therefore, only clients running with the same UNIX system user identifier can be notified by use of the `signal(2)` method.

tuxTdomainSystemAccess

Syntax	INTEGER { <code>fastpath(1)</code> <code>protected(2)</code> <code>fastpath-no-override(3)</code> <code>protected-no-override(4)</code> }
Access	read-write
Description	Default mode used by System/T libraries within application processes to gain access to System/T's internal tables.
	<code>fastpath(1)</code>
	Specifies that System/T's internal tables are accessible by System/T libraries via unprotected shared memory for fast access.
	<code>protected(2)</code>

Specifies that System/T's internal tables are accessible by System/T libraries through protected shared memory for safety against corruption by application code.

`fastpath-no-override(3)` or `protected-no-override(4)`

These can be specified to indicate that the mode selected cannot be overridden by an application process that uses flags available for use with `tpinit(3)`.

Note: Updates to this attribute value in a running application affect only newly started clients and newly configured `tuxTsrvrTbl` objects.

tuxTdomainOptions

Syntax `INTEGER { lan(1) | migrate(2) | accstats(3) | lan-migrate(4) | lan-accstats(5) | migrate-accstats(6) | lan-migrate-accstats(7) | none(8) }`

Access read-write

Description Comma separated list of application options in effect. Valid options are defined as follows:

`lan(1)`

Networked application.

`migrate(2)`

Allow server group migration.

`accstats(3)`

Exact statistics (`single-machine(1)` mode only).

Note: Only the `accstats(3)` can be set or reset in an active application.

tuxTdomainSignal

Syntax `INTEGER { sigusr1(1) | sigusr2(2) }`

Access read-write

Description Signal to be used for signal-based notification (see `tuxTdomainNotify` above).

tuxTdomainSecurity

Syntax	INTEGER DisplayString
Access	read-write
Description	Type of application security. The format is:

security_mode[/app_password]

where *security_mode* can have the following values:

NONE
 APP_PW
 USER_AUTH
 ACL
 MANDATORY_ACL

NONE

A string value of NONE for this attribute indicates that security is/will be turned off.

APP_PW

The value APP_PW/*app_password* indicates that application password security is enforced. Clients must provide the application password during initialization.

USER_AUTH

The value USER_AUTH is similar to APP_PW, but indicates also that per-user authentication is done during client initialization.

ACL

The value ACL is similar to USER_AUTH, but also indicates that access control checks are done on service names, queue names, and event names. If an associated ACL is not found for a name, it is assumed that permission is granted

MANDATORY_ACL

The value MANDATORY_ACL is similar to ACL, but permission is denied if an associated ACL is not found for the name.

app_password

This value is needed whenever *security_mode* is being set to anything but *NONE*. To change the value of *app_password*, SET this object to:

current_security_mode/new_password

On a GET operation, this object only returns the security mode; the password is not returned.

tuxTdomainAuthsvc

Syntax *DisplayString* (SIZE (1..15))

Access read-write

Description Application authentication service invoked by the system for each client that joins the system. This attribute is ignored if the *tuxTdomainSecurity* attribute is set to *NONE* or to *APP-PW*.

tuxTdomainScanUnit

Syntax INTEGER (0..60)

Access read-write

Description Interval of time (in seconds) between periodic scans by the system. Periodic scans are used to detect old transactions and timed-out blocking calls within service requests. The *tuxTdomainBBLQuery*, *tuxTdomainBlockTime*, *tuxTdomainDBBLWait*, and *tuxTdomainSanityScan* objects are multipliers of this value. Passing a value of 0 for this attribute on a SET operation causes the attribute to be reset to its default value.

tuxTdomainBBLQuery

Syntax INTEGER (0..32767)

Access read-write

Description Multiplier of the *tuxTdomainScanUnit* object that indicates time between DBBL status checks on registered BBLs. The DBBL checks to ensure that all BBLs have reported within the *tuxTdomainBBLQuery* cycle. If a BBL has not been heard from, the DBBL sends a message to that BBL asking for status. If no reply is received, the BBL is partitioned. Passing a value of 0 for this attribute on a SET operation causes the attribute to be reset to its default value. This attribute value should be set to at least twice the value set for *tuxTdomainSanityScan*.

tuxTdomainBlockTime

Syntax	INTEGER (0..32767)
Access	read-write
Description	Multiplier of the <code>tuxTdomainScanUnit</code> object that indicates the minimum amount of time a blocking ATMI call blocks before timing out. Passing a value of 0 for this attribute on a SET operation causes the attribute to be reset to its default value.

tuxTdomainDBBLWait

Syntax	INTEGER (0..32767)
Access	read-write
Description	Multiplier of the <code>tuxTdomainScanUnit</code> object that indicates the maximum amount of time a DBBL should wait for replies from its BBLs before timing out. Passing a value of 0 for this attribute on a SET operation causes the attribute to be reset to its default value.

tuxTdomainSanityScan

Syntax	INTEGER (0..32767)
Access	read-write
Description	Multiplier of the <code>tuxTdomainScanUnit</code> object that indicates the time interval between basic sanity checks of the system. Sanity checking includes client/server viability checks done by each BBL for clients/servers running on the local machine as well as BBL status check-ins (<code>multi-machine(2)</code> mode only). Passing a value of 0 for this attribute on a SET operation causes the attribute to be reset to its default value.

tuxTdomainCurDRT

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board routing table entries in use.

tuxTdomainCurGroups

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board server group table entries that are in use.

tuxTdomainCurMachines

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of configured machines.

tuxTdomainCurQueues

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board queue table entries that are in use.

tuxTdomainCurRFT

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board routing criteria range table entries that are in use.

tuxTdomainCurRTdata

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current size of routing table string pool.

tuxTdomainCurServers

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board server table entries that are in use.

tuxTdomainCurServices

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board service table entries that are in use.

tuxTdomainCursType

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board subtype table entries that are in use.

tuxTdomainCurType

Syntax	INTEGER (0..32767)
Access	read-only
Description	Current number of bulletin board type table entries that are in use.

tuxTdomainHwDRT

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of bulletin board routing table entries that are in use.

tuxTdomainHwGroups

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of bulletin board server group table entries that are in use.

tuxTdomainHwMachines

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of configured machines.

tuxTdomainHwQueues

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of bulletin board queue table entries that are in use.

tuxTdomainHwRFT

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of bulletin board routing criteria range table entries that are in use.

tuxTdomainHwRTdata

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water size of routing table string pool.

tuxTdomainHwServers

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of bulletin board server table entries that are in use.

tuxTdomainHwServices

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of bulletin board service table entries that are in use.

tuxTdomainMaxNetGroups

Syntax	INTEGER (1..8191)
Access	read-write
Description	The maximum number of groups that can be configured. This object is only supported on Tuxedo 6.4 or later.

wleMaxObjects

Syntax	INTEGER
Access	read-write
Description	The default maximum number of active objects that can be accommodated in the Active Object Map tables in the WLE bulletin board.

Note: This object is supported for WLE applications only.

wleMaxInterfaces

Syntax	INTEGER (1..32765)
Access	read-write
Description	<p>Specifies the maximum number of interfaces that can be accommodated in the interface table of the bulletin board. If not specified, the default is 100.</p> <p>All instances of an interface occupy and re-use the same slot in the interface table in the bulletin board. For example, if server SVR1 advertises interfaces IF1 and IF2, SVR2 advertises IF2 and IF3, and SVR3 advertises IF3 and IF4, the interface count is 4, not 6, when calculating wleMaxInterfaces.</p> <p>Note: This object is supported for WLE applications only.</p>

tuxTdomainSignatureAhead

Syntax	INTEGER (1..2147483647)
Access	read-write
Description	Number of seconds a valid signature's timestamp can be ahead of the local machine's clock.

wleCurInterfaces

Syntax	INTEGER
Access	read-only
Description	<p>The current number of interface entries used in the bulletin board interface tables.</p> <p>Note: This object is supported for WLE applications only.</p>

wleHwInterfaces

Syntax	INTEGER
Access	read-only
Description	The high water mark for the number of interface entries used in the bulletin board interface tables.

Note: This object is supported for WLE applications only.

tuxTdomainSignatureBehind

Syntax	INTEGER (1..2147483647)
Access	read-write
Description	Number of seconds a valid signature's timestamp can be behind the local machine's clock.

tuxTdomainEncryptionRequired

Syntax	INTEGER {yes (1) no (2)}
Access	read-write
Description	If set to “yes,” every application service in this domain requires an encrypted input message buffer.

tuxTdomainSignatureRequired

Syntax	INTEGER {yes (1) no (2)}
Access	read-write
Description	If set to “yes,” every application service in this domain requires a valid digital signature on its input message buffer.

tuxTgroupTable

The tuxTgroupTable group represents application attributes pertaining to a particular server group. These attribute values represent group identification, location, and DTP information. The index for this table is tuxTgroupNo. To create a new row, it is necessary to issue a SET request for a non-existing instance that at least specifies values for tuxTgroupName and tuxTgroupLMID.

Variable Name	Object ID
tuxTgroupName	.1.3.6.1.4.1.140.300.4.1.1.1
tuxTgroupNo	.1.3.6.1.4.1.140.300.4.1.1.2
tuxTgroupLMID	.1.3.6.1.4.1.140.300.4.1.1.3
tuxTgroupState	.1.3.6.1.4.1.140.300.4.1.1.4
tuxTgroupCurLMID	.1.3.6.1.4.1.140.300.4.1.1.5
tuxTgroupCloseInfo	.1.3.6.1.4.1.140.300.4.1.1.6
tuxTgroupOpenInfo	.1.3.6.1.4.1.140.300.4.1.1.7
tuxTgroupTMScount	.1.3.6.1.4.1.140.300.4.1.1.8
tuxTgroupTMSname	.1.3.6.1.4.1.140.300.4.1.1.9
tuxTgroupEncryptionRequired	.1.3.6.1.4.1.140.300.4.1.1.20
tuxTgroupSignatureRequired	.1.3.6.1.4.1.140.300.4.1.1.30

tuxTgroupName

Syntax *DisplayString* (SIZE (1..30))
Access read-write

Description Logical name of the server group. The group name must be unique within all group names in the `tuxTgroupTable` class and `tuxTgroupLMID` values in the `tuxTmachineTable` class. Server group names cannot contain an asterisk (*), comma, or colon.

Note: This object can be set only during row creation.

tuxTgroupNo

Syntax INTEGER (1..29999)

Access read-write

Description Group number associated with this server group.

Note: This object can be set only during row creation.

tuxTgroupLMID

Syntax *DisplayString* (SIZE (1..61))

Access read-write

Description *DisplayString* is in the format: *LMID1*[, *LMID2*]

LMID1

Is the primary machine logical machine identifier for this server group and is in the range from one to sixty-one characters.

LMID2

Is the optional secondary logical machine identifier and is in the range from one to sixty-one characters.

The secondary LMID indicates the machine to which the server group can be migrated (if the MIGRATE option is specified in the `tuxTdomainOptions` attribute). A single LMID specified on a GET operation matches either the primary or secondary LMID. Note that the location of an active group is available in the `tuxTgroupCurLMID` object. Logical machine identifiers specified with the `tuxTgroupLMID` object must already be configured.

Note: Modifications to this attribute for an active object can only change the backup LMID designation for the group.

tuxTgroupState

Syntax	INTEGER { active(1) inactive(2) migrating(3) invalid(4) re-active(5) suspend-services(6) resume-services(7) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: {active(1) inactive(2) migrating(3)}</p> <p>A GET operation retrieves configuration and runtime information for the selected tuxTgroupTable object(s). The following states indicate the meaning of a tuxTgroupState returned in response to a GET request. States not listed are not returned.</p> <p>active(1)</p> <p>tuxTgroupTable object defined and active (TMS and/or application servers). Server groups with non-0 length values for the tuxTgroupTMSname attribute are considered active if the TMSs associated with the group are active. Otherwise, a group is considered active if any server in the group is active.</p> <p>inactive(2)</p> <p>tuxTgroupTable object defined and inactive.</p> <p>migrating(3)</p> <p>tuxTgroupTable object defined and currently in a state of migration to the secondary logical machine. The secondary logical machine is the one listed in tuxTgroupLMID that does not match tuxTgroupCurLMID.</p> <p>SET: {active(1) inactive(2) migrating(3) invalid(4) re-active(5) suspend-services(6) resume-services(7)}</p> <p>A SET operation updates configuration and runtime information for the selected tuxTgroupTable object. The following states indicate the meaning of a tuxTgroupState set in a SET request. States not listed cannot be set.</p> <p>active(1)</p> <p>Activate the tuxTgroupTable object. State change is allowed only when the group is in the inactive(2) or migrating(3) state. If the group is currently in the inactive(2) state and the primary logical machine is active, then TMS and application servers are started on the primary logical machine; otherwise, if the secondary logical machine is active, the TMS and</p>

application servers are started on the secondary logical machine. If neither machine is active, then the request fails. If the group is currently in the `migrating(3)` state, then the active secondary logical machine (identified as the alternate to `tuxTgroupCurLMID` in the `tuxTgroupLMID` list), if it is active, is used to start TMS and application servers. Otherwise, the request fails. Successful return leaves the object in the `active(1)` state.

`inactive(2)`

Deactivate the `tuxTgroupTable` instance. TMS and application servers are deactivated. State change is allowed only when the group is in the `active(1)` or `migrating(3)` state. Successful return leaves the object in the `inactive(2)` state.

`migrating(3)`

Deactivate the `tuxTgroupTable` object on its active primary logical machine (`tuxTgroupCurLMID`) and prepare the group to be migrated to the secondary logical machine. State change is allowed only when the group is in the `active(1)` state. Successful return leaves the object in the `migrating(3)` state.

`invalid(4)`

Delete `tuxTgroupTable` object for application. State change is allowed only when the group is in the `inactive(2)` state. Successful return leaves the object in the `invalid(4)` state.

`re-active(5)`

Identical to a transition to the `active(1)` state except that this state change is also allowed in the `active(1)` state as well as the `inactive(2)` and `migrating(3)` states.

`suspend-services(6)`

Suspend the application services in the group. A `SET` operation to this state is allowed only when the group is in the `active(1)` state. The operation leaves the group in `active(1)` state but with all its application services in a suspended state. This state is only available in Tuxedo 6.4 and later.

`resume-services(7)`

Unsuspend and resume all application services that are marked suspended in the group. This operation is allowed only when the group is in the `active(1)` state. The operation leaves the group in the `active(1)` state.

Note that this operation fails in an application environment that includes any machine where Tuxedo 6.3 or earlier applications are active.

tuxTgroupCurLMID

Syntax	<i>DisplayString</i> (SIZE (1..30))
Access	read-only
Description	Current logical machine on which the server group is running. This attribute is not returned for server groups that are not active.

tuxTgroupCloseInfo

Syntax	<i>DisplayString</i> (SIZE (0..256))
Access	read-write
Description	If a non-0 length value other than TMS is specified for the tuxTgroupTMSname object, then this object value indicates the resource manager-dependent information needed to terminate access to the resource manager. Otherwise, this attribute value is ignored.

The format for this object value is dependent on the requirements of the vendor providing the underlying resource manager. The information required by the vendor must be prefixed with `rm_name:`, which is the published name of the vendor's transaction (XA) interface followed immediately by a colon (:).

A 0-length string value for this attribute means that the resource manager for this group (if specified) does not require any application-specific information to close access to the resource.

Note: Runtime modifications to this attribute do not affect active servers in the group.

tuxTgroupOpenInfo

Syntax	<i>DisplayString</i> (SIZE (0..256))
Access	read-write
Description	If a non-0 length value other than TMS is specified for the tuxTgroupTMSname object, this object value indicates the resource manager-dependent information needed to initiate access to the resource manager. Otherwise, this object value is ignored.

The format for this object value is dependent on the requirements of the vendor that provides the underlying resource manager. The information required by the vendor must be prefixed with `rm_name:`, which is the published name of the vendor's transaction (XA) interface followed immediately by a colon (:).

A 0-length string value for this attribute means that the resource manager for this group (if specified) does not require any application-specific information to open access to the resource.

Note: Runtime modifications to this attribute do not affect active servers in the group.

tuxTgroupTMScount

Syntax	INTEGER (0..11)
Access	read-write
Description	If a non-0 length value is specified for the <code>tuxTgroupTMSname</code> object, this object value indicates the number of transaction manager servers to start for the associated group. Otherwise, this object value is ignored.

tuxTgroupTMSname

Syntax	<i>DisplayString</i> (SIZE (0..78))
Access	read-write
Description	<p>Transaction manager server <code>a.out</code> associated with this group. This parameter must be specified for any group entry whose servers participate in distributed transactions (transactions across multiple resource managers and possibly machines that are started with <code>tpbegin(3)</code> and ended with <code>tpcommit(3)</code> or <code>tpabort(3)</code>).</p> <p>The value <code>TMS</code> is reserved to indicate use of the null XA interface. If a non-empty value other than <code>TMS</code> is specified, a <code>tuxTmachineTlogDevice</code> must be specified for the machine(s) associated with the primary and secondary logical machines for this object</p> <p>A unique server identifier is selected automatically for each TM server, and the servers are restartable an unlimited number of times.</p>

tuxTgroupEncryptionRequired

Syntax	Integer { yes (1) no (2) }
Access	read-write
Description	If set to “yes,” every application service in this group requires an encrypted message buffer.

tuxTgroupSignatureRequired

Syntax	Integer { yes (1) no (2) }
Access	read-write
Description	If set to “yes,” every application service in this group requires a valid digital signature on its input message buffer.

tuxTmachineTable

The `tuxTmachineTable` group represents application attributes pertaining to a particular machine. These attribute values represent machine characteristics, per-machine sizing, statistics, customization options, and UNIX system filenames. This group is available for configured-inactive as well as configured-active machines in the application. The index into this table is `tuxTmachinePmid`. To create a new row, issue a SET request for a non-existing row that specifies at least the values for `tuxTmachineLmid`, `tuxTmachineTuxDir`, `tuxTmachineTuxConfig`, and `tuxTmachineAppDir`. For a multi-machine Tuxedo application, `tuxTmachineNaddr`, `tuxTmachineNlsAddr`, and `tuxTmachineBridge` must also be specified.

Variable Name	Object ID
<code>tuxTmachinePmid</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.1</code>
<code>tuxTmachineLmid</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.2</code>
<code>tuxTmachineTuxConfig</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.3</code>
<code>tuxTmachineTuxDir</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.4</code>
<code>tuxTmachineAppDir</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.5</code>
<code>tuxTmachineState</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.6</code>
<code>tuxTmachineUid</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.7</code>
<code>tuxTmachineGid</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.8</code>
<code>tuxTmachineEnvFile</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.9</code>
<code>tuxTmachinePerm</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.10</code>
<code>tuxTmachineUlogPfx</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.11</code>
<code>tuxTmachineType</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.12</code>
<code>tuxTmachineMaxAccessers</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.13</code>
<code>tuxTmachineMaxConv</code>	<code>.1.3.6.1.4.1.140.300.5.1.1.14</code>

Variable Name	Object ID
tuxTmachineMaxGtt	.1.3.6.1.4.1.140.300.5.1.1.15
tuxTmachineMaxWsClients	.1.3.6.1.4.1.140.300.5.1.1.16
tuxTmachineMaxAclCache	.1.3.6.1.4.1.140.300.5.1.1.17
tuxTmachineTlogDevice	.1.3.6.1.4.1.140.300.5.1.1.18
tuxTmachineTlogName	.1.3.6.1.4.1.140.300.5.1.1.19
tuxTmachineTlogSize	.1.3.6.1.4.1.140.300.5.1.1.20
tuxTmachineBridge	.1.3.6.1.4.1.140.300.5.1.1.21
tuxTmachineNaddr	.1.3.6.1.4.1.140.300.5.1.1.22
tuxTmachineNlsaddr	.1.3.6.1.4.1.140.300.5.1.1.23
tuxTmachineCmpLimit	.1.3.6.1.4.1.140.300.5.1.1.24
tuxTmachineTmNetLoad	.1.3.6.1.4.1.140.300.5.1.1.25
tuxTmachineSpinCount	.1.3.6.1.4.1.140.300.5.1.1.26
tuxTmachineRole	.1.3.6.1.4.1.140.300.5.1.1.27
tuxTmachineMinor	.1.3.6.1.4.1.140.300.5.1.1.28
tuxTmachineRelease	.1.3.6.1.4.1.140.300.5.1.1.29
tuxTmachineMaxPendingBytes	.1.3.6.1.4.1.140.300.5.1.1.30
wleMaxMachineObjects	.1.3.6.1.4.1.140.300.5.1.1.35
tuxTmachineEncryptionRequired	.1.3.6.1.4.1.140.300.5.1.1.50
tuxTmachineSignatureRequired	.1.3.6.1.4.1.140.300.5.1.1.60

tuxTmachinePmid

Syntax	<i>DisplayString</i> (SIZE (1..30))
Access	read-write
Description	Physical machine identifier. This identifier should match the UNIX system nodename returned by the <code>uname -n</code> command when run on the identified system. For a Windows NT system, this identifier should match the computer name and the name configured with the name server.

Note: This object can be set only during row creation.

tuxTmachineLmid

Syntax	<i>DisplayString</i> (SIZE (1..30))
Access	read-write
Description	Logical machine identifier.

Note: This object can be set only during row creation.

tuxTmachineTuxConfig

Syntax	<i>DisplayString</i> (SIZE (2..78))
Access	read-write
Description	Absolute pathname of the file or device where the binary Tuxedo System/T configuration file is found on this machine. The administrator need only maintain one such file, namely the one identified by the <code>tuxTmachineTuxConfig</code> attribute value on the master machine. The information contained in this file is automatically propagated to all other <code>tuxTmachineTable</code> objects as they are activated. See <code>tuxTmachineEnvFile</code> for a discussion of how this attribute value is used in the environment.

tuxTmachineTuxDir

Syntax	<i>DisplayString</i> (SIZE (2..78))
Access	read-write

Description Absolute pathname of the directory where the Tuxedo System/T software is found on this machine. See `tuxTmachineEnvFile` below for a discussion of how this attribute value is used in the environment.

tuxTmachineAppDir

Syntax `DisplayString (SIZE (2..78))`

Access read-write

Description Colon-separated list of application directory absolute pathnames. The first directory serves as the current directory for all application and administrative servers booted on this machine. All directories in the list are searched when application servers are started. See `tuxTmachineEnvFile` for a discussion of how this attribute value is used in the environment.

tuxTmachineState

Syntax `INTEGER { active(1) | inactive(2) | partitioned(3) | invalid(4) | re-activate(5) | cleaning(7) }`

Access read-write

Description The values for GET and SET operations are as follows:

GET: {active(1)|inactive(2)|partitioned(3)}

A GET operation retrieves configuration and runtime information for the selected `tuxTmachineTable` instance(s). The following states indicate the meaning of a `tuxTmachineState` returned in response to a GET request. States not listed are not returned

`active(1)`

`tuxTmachineTable` instance defined and active (administrative servers, that is, DBBL, BBL, and BRIDGE).

`inactive(2)`

`tuxTmachineTable` instance defined and inactive.

`partitioned(3)`

`tuxTmachineTable` instance defined, listed in accessible bulletin boards as active, but currently unreachable.

SET: {active(1)|inactive(2)|invalid(4)|re-activate(5)|cleaning(7)}

A SET operation updates configuration and runtime information for the selected `tuxTmachineTable` instance. The following states indicate the meaning of a `tuxTmachineState` set in a SET request. States not listed can not be set.

`active(1)`

Activate the `tuxTmachineTable` instance. Necessary administrative servers such as the DBBL, BBL, and BRIDGE are started on the indicated site as well as application servers configured to run on that site. State change is allowed only when the machine is in the `inactive(2)` state. Successful return leaves the object in the `active(1)` state.

`inactive(2)`

Deactivate the `tuxTmachineTable` instance. Necessary administrative servers such as the BBL and BRIDGE are stopped on the indicated site as well as application servers running on that site. State change allowed only when the machine is in the `active(1)` state and when no other application resources are active on the indicated machine. Successful return leaves the object in the `inactive(2)` state.

`invalid(4)`

Delete `tuxTmachineTable` instance for application. State change is allowed only when the machine is in the `inactive(2)` state. Successful return leaves the object in the `invalid(4)` state.

`re-activate(5)`

Activate the `tuxTmachineTable` instance. Necessary administrative servers such as the DBBL, BBL, and BRIDGE are started on the indicated site. State change is allowed only when the machine is in either the `active(1)` or `inactive(2)` state. Successful return leaves the object in the `active(1)` state.

`cleaning(7)`

Initiate cleanup/scanning activities on and relating to the indicated machine. If there are dead clients or servers on the machine, they are detected at this time. If the machine has been partitioned from the application master site, then global bulletin board entries for that machine are removed. This combination is allowed when the application is in the `active(1)` state and the `tuxTmachineTable` instance is in either the `active(1)` or

`partitioned(3)` state. Successful return for a non-partitioned machine leaves the state unchanged. Successful return for a partitioned machine leaves the object in the `inactive(2)` state.

Note: State change to `inactive(2)` is allowed only for non-master machines. The master site administrative processes are deactivated through the `tuxTdomain` class.

tuxTmachineUid

Syntax `INTEGER`

Access `read-write`

Description UNIX system user-identifier for the Tuxedo System/T application administrator on this machine. Administrative commands such as `tmboot(1)`, `tmshutdown(1)`, and `tmadmin(1)` must run as the indicated user on this machine. Application and administrative servers on this machine are started as this user.

Note: This is a UNIX system-specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTmachineGid

Syntax `INTEGER`

Access `read-write`

Description UNIX system group identifier for the Tuxedo System/T application administrator on this machine. Administrative commands such as `tmboot(1)`, `tmshutdown(1)`, and `tmadmin(1)` must run as part of the indicated group on this machine. Application and administrative servers on this machine are started as part of this group.

Note: This is a UNIX system-specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTmachineEnvFile

Syntax `DisplayString (SIZE (2..78))`

Access `read-write`

Description Environment file for clients and servers running on this machine.

tuxTmachinePerm

Syntax	<i>DisplayString</i> (SIZE(1..9))
Access	read-write
Description	UNIX system permissions associated with the shared memory bulletin board created on this machine. Default UNIX system permissions for system and application message queues. Note: Modifications to this attribute for an active object do not affect running servers or clients. Note: This is a UNIX system-specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTmachineUlogPfx

Syntax	<i>DisplayString</i> (SIZE (0..78))
Access	read-write
Description	Absolute pathname prefix of the path for the <code>userlog(3)</code> file on this machine. The <code>userlog(3)</code> file name is formed by appending the string <code>.mmddyy</code> to the <code>tuxTmachineUlogPfx</code> attribute value. <code>.mmddyy</code> represents the month, day, and year that the messages were generated. All application and system <code>userlog(3)</code> messages generated by clients and servers running on this machine are directed to this file. Note: Modifications to this attribute for an active object do not affect running servers or clients.

tuxTmachineType

Syntax	<i>DisplayString</i> (SIZE (1..15))
Access	read-write
Description	Machine type. This attribute is used to group machines into classes of like data representations. Data encoding is not performed when communicating between machines of identical types. This attribute can be given any string value; values are used only for comparison. Distinct <code>tuxTmachineType</code> attributes should be set when the application spans a heterogeneous network of machines or when compilers

generate dissimilar structure representations. The default value for this attribute, a 0-length string, matches any other machine with a 0-length string as its tuxTmachineType attribute value.

tuxTmachineMaxAccessers

Syntax	INTEGER (1..32767)
Access	read-write
Description	Maximum number of clients and servers that can have access to the bulletin board on this machine at one time. System administration processes such as the BBL and tmadmin need not be accounted for in this figure, but all application servers and clients and TMS servers should be counted. If the application is booting workstation listeners on this site, then both the listeners and the potential number of workstation handlers that can be booted should be counted.

tuxTmachineMaxConv

Syntax	INTEGER (0..32767)
Access	read-write
Description	Maximum number of simultaneous conversations in which clients and servers on this machine can be involved.

tuxTmachineMaxGtt

Syntax	INTEGER (0..32767)
Access	read-write
Description	Maximum number of simultaneous global transactions in which this machine can be involved.

tuxTmachineMaxWsClients

Syntax	INTEGER (0..32767)
Access	read-write
Description	Number of entries for accessers on this machine to be reserved for workstation clients. The number specified here takes a portion of the total slots for accessers specified with the <code>tuxTmachineMaxAccessers</code> attribute. The appropriate setting of this parameter helps to conserve IPC resources because workstation client access to the system is multiplexed through a System/T supplied surrogate, the workstation handler. It is an error to set this number greater than <code>tuxTmachineMaxAccessers</code> .

tuxTmachineMaxAclCache

Syntax	INTEGER (10..32000)
Access	read-write
Description	Number of entries in the cache used for ACL entries when <code>tuxTdomainSecurity</code> is set to <code>acl(4)</code> or <code>mandatory-acl(5)</code> . The appropriate setting of this parameter helps to conserve shared memory resources and yet reduce the number of disk access to do ACL checking.

tuxTmachineTlogDevice

Syntax	<i>DisplayString</i> (SIZE (0..64))
Access	read-write
Description	The device (raw slice) or UNIX system file containing the Tuxedo System/T filesystem that holds the DTP transaction log for this machine. The DTP transaction log is stored as a Tuxedo System/T VTOC table on the device. This device or file can be the same as that specified for the <code>tuxTmachineTuxConfig</code> attribute for this machine.

tuxTmachineTlogName

Syntax	<i>DisplayString</i> (SIZE (0..30))
Access	read-write
Description	The name of the DTP transaction log for this machine. If more than one DTP transaction log exists on the same <code>tuxTmachineTlogDevice</code> , they must have unique names. <code>tuxTmachineTlogName</code> must be different from the name of any other table on the <code>tuxTmachineTlogDevice</code> where the DTP transaction log table is created.

tuxTmachineTlogSize

Syntax	INTEGER (1..2048)
Access	read-write
Description	The numeric size, in pages, of the DTP transaction log for this machine. The <code>tuxTmachineTlogSize</code> attribute value is subject to limits based on available space in the Tuxedo System/T filesystem identified by the <code>tuxTmachineTlogDevice</code> attribute.

tuxTmachineBridge

Syntax	<i>DisplayString</i> (SIZE (0..78))
Access	read-write
Description	Device name to be used by the BRIDGE process placed on this logical machine to access the network. This is a required value for participation in a networked application through a TLI-based Tuxedo System/T binary. This attribute is not needed for sockets-based Tuxedo System/T binaries.

tuxTmachineNaddr

Syntax	<i>DisplayString</i> (SIZE (0..78))
Access	read-write
Description	Specifies the complete network address to be used by the BRIDGE process placed on the logical machine as its listening address. The listening address for a BRIDGE is the means by which it is contacted by other BRIDGE processes participating in the

application. This attribute must be set if the logical machine is to participate in a networked application, that is, if the LAN option is set in the `tuxTdomainOptions` attribute value.

If *DisplayString* has the form `0xhex-digits` or `\\xhex-digits`, it must contain an even number of valid hexadecimal digits. These forms are translated internally into a character array that contains the hexadecimal representations of the specified string.

tuxTmachineNlsaddr

Syntax	<i>DisplayString</i> (SIZE (0..78))
Access	read-write
Description	Network address used by the <code>tlisten(1)</code> process servicing the network on the node identified by this logical machine. This network address has the same format as that specified for the <code>tuxTmachineNaddr</code> attribute.

This attribute must be set if the logical machine is to participate in a networked application, that is, if the LAN option is set in the `tuxTdomainOptions` attribute value.

tuxTmachineCmplLimit

Syntax	<i>DisplayString</i>
Access	read-write
Description	Threshold message size at which compression occurs for remote traffic and, optionally, local traffic. Remote and local can be either non-negative numeric values or the string <code>MAXLONG</code> that is dynamically translated to the maximum long setting for the machine. Setting only the remote value defaults local to <code>MAXLONG</code> .

Note: This attribute value is not part of the `tuxTmachineTable` object for active sites running Tuxedo System/T Release 4.2.2 or earlier. However, site release identification is not determined until runtime, so this attribute can be set and accessed for any inactive object. When a Tuxedo System/T Release 4.2.2 or earlier site is activated, the configured value is not used.

tuxTmachineTmNetLoad

Syntax	INTEGER (0..32767)
Access	read-write
Description	Service load added to any remote service evaluated during load balancing on this machine.

Note: This attribute value is not part of the tuxTmachineTable object for active sites running Tuxedo System/T Release 4.2.2 or earlier. However, site release identification is not determined until runtime, so this attribute can be set and accessed for any inactive object. When a Tuxedo System/T Release 4.2.2 or earlier site is activated, the configured value is not used.

tuxTmachineSpinCount

Syntax	INTEGER
Access	read-write
Description	Spincount used on this machine for pre-ticket user-level semaphore access. Default values are built into the Tuxedo System/T binaries on each machine. For tuning purposes, these defaults can be overridden at runtime using this attribute. The spincount can be reset to the default built-in value for the site by resetting this attribute value to 0.

Note: This attribute value is not part of the tuxTmachineTable object for active sites running Tuxedo System/T Release 4.2.2 or earlier. However, site release identification is not determined until runtime, so this attribute can be set and accessed for any inactive object. When a Tuxedo System/T Release 4.2.2 or earlier site is activated, the configured value is not used.

tuxTmachineRole

Syntax	INTEGER { master(1) backup(2) other(3) }
Access	read-only
Description	The role of this machine in the application. master(1)

Indicates that this machine is the master machine,
`backup(2)`
Indicates that it is the backup master machine, and
`other(3)`
Indicates that the machine is neither the master nor the backup master machine.

tuxTmachineMinor

Syntax	INTEGER
Access	read-only
Description	The Tuxedo System/T minor protocol release number for this machine.

tuxTmachineRelease

Syntax	INTEGER
Access	read-only
Description	The Tuxedo System/T major protocol release number for this machine. This can be different from the <code>tuxTmachineSWrelease</code> for the same machine.

tuxTmachineMaxPendingBytes

Syntax	INTEGER
Access	read-write
Description	Specifies a limit for the amount of space that can be allocated for messages waiting to be transmitted by the BRIDGE process. The minimum value is 100000. This object is supported on Tuxedo 6.4 and later only.

wleMachineMaxObjects

Syntax	INTEGER
Access	read-write

Description The maximum number of objects that can be accommodated in the Active Object Map tables in the bulletin board.

Note: This object is supported for WLE applications only.

tuxTmachineEncryptionRequired

Syntax INTEGER { yes(1) | no (2) }

Access read-write

Description If set to “yes,” every application service on this machine requires an encrypted input message buffer.

tuxTmachineSignatureRequired

Syntax INTEGER { yes(1) | no (2) }

Access read-write

Description If set to “yes,” every application service on this machine requires a valid digital signature on its input message buffer.

tuxTmachineActive

This group represents runtime statistics on the local machine if it is active (that is, some component of the application is active on the machine). Objects in this group are only accessible through a Tuxedo SNMP agent installed on the local machine.

Variable Name	Object ID
tuxTmachineCurAccessers	.1.3.6.1.4.1.140.300.5.2.1
tuxTmachineCurClients	.1.3.6.1.4.1.140.300.5.2.2
tuxTmachineCurConv	.1.3.6.1.4.1.140.300.5.2.3
tuxTmachineCurGTT	.1.3.6.1.4.1.140.300.5.2.4
tuxTmachineCurLoad	.1.3.6.1.4.1.140.300.5.2.5
tuxTmachineCurWsClients	.1.3.6.1.4.1.140.300.5.2.6
tuxTmachineHwAccessers	.1.3.6.1.4.1.140.300.5.2.7
tuxTmachineHwClients	.1.3.6.1.4.1.140.300.5.2.8
tuxTmachineHwConv	.1.3.6.1.4.1.140.300.5.2.9
tuxTmachineHwGTT	.1.3.6.1.4.1.140.300.5.2.10
tuxTmachineHwWsClients	.1.3.6.1.4.1.140.300.5.2.11
tuxTmachineNumConv	.1.3.6.1.4.1.140.300.5.2.12
tuxTmachineNumDequeue	.1.3.6.1.4.1.140.300.5.2.13
tuxTmachineNumEnqueue	.1.3.6.1.4.1.140.300.5.2.14
tuxTmachineNumPost	.1.3.6.1.4.1.140.300.5.2.15
tuxTmachineNumReq	.1.3.6.1.4.1.140.300.5.2.16
tuxTmachineNumSubscribe	.1.3.6.1.4.1.140.300.5.2.17
tuxTmachineNumTran	.1.3.6.1.4.1.140.300.5.2.18

Variable Name	Object ID
tuxTmachineNumTranAbt	.1.3.6.1.4.1.140.300.5.2.19
tuxTmachineNumTranCmt	.1.3.6.1.4.1.140.300.5.2.20
tuxTmachineLicExpires	.1.3.6.1.4.1.140.300.5.2.21
tuxTmachineLicMaxUsers	.1.3.6.1.4.1.140.300.5.2.22
tuxTmachineLicSerial	.1.3.6.1.4.1.140.300.5.2.23
tuxTmachinePageSize	.1.3.6.1.4.1.140.300.5.2.24
tuxTmachineSWrelease	.1.3.6.1.4.1.140.300.5.2.25
tuxTmachineHwAclCache	.1.3.6.1.4.1.140.300.5.2.26
tuxTmachineAclCacheHits	.1.3.6.1.4.1.140.300.5.2.27
tuxTmachineAclCacheAccess	.1.3.6.1.4.1.140.300.5.2.28
tuxTmachineAclFail	.1.3.6.1.4.1.140.300.5.2.29
tuxTmachineWkCompleted	.1.3.6.1.4.1.140.300.5.2.30
tuxTmachineWkInitiated	.1.3.6.1.4.1.140.300.5.2.31
wleMachineCurObjects	.1.3.6.1.4.1.140.300.5.2.36
wleMachineHwObjects	.1.3.6.1.4.1.140.300.5.2.41

tuxTmachineCurAccessers

Syntax	INTEGER (0..32767)
Access	read-only
Description	Number of clients and servers that currently access the application either directly on this machine or through a workstation handler on this machine.

tuxTmachineCurClients

Syntax	INTEGER (0..32767)
Access	read-only
Description	Number of clients, both native and workstation, currently logged in to this machine.

tuxTmachineCurConv

Syntax	INTEGER (0..32767)
Access	read-only
Description	Number of active conversations with participants on this machine.

tuxTmachineCurGTT

Syntax	INTEGER (0..32767)
Access	read-only
Description	Number of in use transaction table entries on this machine.

tuxTmachineCurLoad

Syntax	INTEGER
Access	read-only
Description	Current service load enqueued on this machine.

Note: If the `tuxTdomainLoadBalance` attribute is `no(2)` or the `tuxTdomainModel` attribute is `multi-machine(2)`, then an FML32 NULL value (0) is returned.

tuxTmachineCurWsClients

Syntax	INTEGER (0..32767)
Access	read-only
Description	Number of workstation clients currently logged in to this machine.

tuxTmachineHwAccessers

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of clients and servers accessing the application either directly on this machine or through a workstation handler on this machine.

tuxTmachineHwClients

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of clients, both native and workstation, logged in to this machine.

tuxTmachineHwConv

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of active conversations with participants on this machine.

tuxTmachineHwGTT

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of in use transaction table entries on this machine.

tuxTmachineHwWsClients

Syntax	INTEGER (0..32767)
Access	read-only
Description	High water number of workstation clients currently logged in to this machine.

tuxTmachineNumConv

Syntax	INTEGER
Access	read-only
Description	Number of <code>tpconnect(3)</code> operations performed from this machine.

tuxTmachineNumDequeue

Syntax	INTEGER
Access	read-only
Description	Number of <code>tpdequeue(3)</code> operations performed from this machine.

tuxTmachineNumEnqueue

Syntax	INTEGER
Access	read-only
Description	Number of <code>tpenqueue(3)</code> operations performed from this machine.

tuxTmachineNumPost

Syntax	INTEGER
Access	read-only
Description	Number of <code>tppost(3)</code> operations performed from this machine.

tuxTmachineNumReq

Syntax	INTEGER
Access	read-only
Description	Number of <code>tpacall(3)</code> or <code>tpcall(3)</code> operations performed from this machine.

tuxTmachineNumSubscribe

Syntax	INTEGER
Access	read-only
Description	Number of <code>tpsubscribe(3)</code> operations performed from this machine.

tuxTmachineNumTran

Syntax	INTEGER
Access	read-only
Description	Number of transactions initiated (<code>tpbegin(3)</code>) from this machine.

tuxTmachineNumTranAbt

Syntax	INTEGER
Access	read-only
Description	Number of transactions aborted (<code>tpabort(3)</code>) from this machine.

tuxTmachineNumTranCmt

Syntax	INTEGER
Access	read-only
Description	Number of transactions committed (<code>tpcommit(3)</code>) from this machine.

tuxTmachineLicExpires

Syntax	<i>DisplayString</i> (SIZE(0..78))
Access	read-only
Description	Expiration date for the binary on the machine or a 0-length string if binary is not a Tuxedo System/T master binary.

tuxTmachineLicMaxUsers

Syntax	INTEGER (0..32767)
Access	read-only
Description	Maximum number of licensed users on that machine, or -1 if binary is not a Tuxedo System/T master binary.

tuxTmachineLicSerial

Syntax	<i>DisplayString</i> (SIZE(0..78))
Access	read-only
Description	Serial number for binary on the machine or a 0-length string if binary is not a Tuxedo System/T master binary.

tuxTmachinePageSize

Syntax	INTEGER
Access	read-only
Description	Disk pagesize used on this machine.

tuxTmachineSWrelease

Syntax	<i>DisplayString</i> (SIZE(0..78))
Access	read-only
Description	Software release for binary on that machine or a 0-length string if binary is not a Tuxedo System/T master binary.

tuxTmachineHwAclCache

Syntax	INTEGER
Access	read-only
Description	High water number of entries used in the ACL cache.

tuxTmachineAclCacheHits

Syntax	INTEGER
Access	read-only
Description	Number of accesses to the ACL cache that resulted in a “hit” (that is, the entry was already in the cache).

tuxTmachineAclCacheAccess

Syntax	INTEGER
Access	read-only
Description	Number of accesses to the ACL cache.

tuxTmachineAclFail

Syntax	INTEGER
Access	read-only
Description	Number of accesses to the ACL cache that resulted in a access control violation.

tuxTmachineWkCompleted

Syntax	INTEGER
Access	read-only
Description	Total service load dequeued and processed successfully by servers running on this machine. Note that for long running applications this attribute can wraparound, that is, exceed the maximum value for a long, and start back at 0 again.

tuxTmachineWkInitiated

Syntax	INTEGER
Access	read-only
Description	Total service load enqueued by clients/servers running on this machine. Note that for long running applications this attribute can wraparound, that is, exceed the maximum value for a long, and start back at 0 again.

wleMachineCurObjects

Syntax	INTEGER
Access	read-only
Description	The number of entries in use in the bulletin board object table for this machine.

Note: This object is supported for WLE applications only.

wleMachineHwObjects

Syntax	INTEGER
Access	read-only
Description	The high water mark of entries used in the bulletin board object table for this machine.

Note: This object is supported for WLE applications only.

tuxTmsgTable

The `tuxTmsgTable` class represents runtime attributes of the Tuxedo System/T managed UNIX system message queues. Objects in this table are only accessible through a Tuxedo SNMP agent installed on the local machine. `tuxTmsgId` is the index into this table.

Variable Name	Object ID
tuxTmsgId	.1.3.6.1.4.1.140.300.6.1.1.1
tuxTmsgState	.1.3.6.1.4.1.140.300.6.1.1.2
tuxTmsgCurTime	.1.3.6.1.4.1.140.300.6.1.1.3
tuxTmsgCbytes	.1.3.6.1.4.1.140.300.6.1.1.4
tuxTmsgCtime	.1.3.6.1.4.1.140.300.6.1.1.5
tuxTmsgLrPid	.1.3.6.1.4.1.140.300.6.1.1.6
tuxTmsgLsPid	.1.3.6.1.4.1.140.300.6.1.1.7
tuxTmsgQbytes	.1.3.6.1.4.1.140.300.6.1.1.8
tuxTmsgQnum	.1.3.6.1.4.1.140.300.6.1.1.9
tuxTmsgRtime	.1.3.6.1.4.1.140.300.6.1.1.10
tuxTmsgStime	.1.3.6.1.4.1.140.300.6.1.1.11

tuxTmsgId

Syntax	INTEGER
Access	read-only
Description	UNIX system message queue identifier.

Note: This is a UNIX system-specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTmsgState

Syntax	INTEGER { active(1) }
Access	read-only
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: active(1)</p> <p>A GET operation retrieves runtime information for the selected <code>tuxTmsgTable</code> object(s). The following state indicates the meaning of a <code>tuxTmsgState</code> returned in response to a GET request. States not listed are not returned.</p> <p>active(1)</p> <p><code>tuxTmsgTable</code> object active. This corresponds exactly to the related <code>tuxTmachineTable</code> object being active.</p> <p>SET:</p> <p>SET operations are not permitted on this class.</p>

tuxTmsgCurTime

Syntax	INTEGER
Access	read-only
Description	<p>Current time, in seconds, since 00:00:00 UTC, January 1, 1970, as returned by the <code>time(2)</code> system call on the local host.</p>

tuxTmsgCbytes

Syntax	INTEGER
Access	read-only
Description	<p>Current number of bytes on the queue.</p>

tuxTmsgCtime

Syntax	INTEGER
Access	read-only
Description	Time of the last <code>msgctl(2)</code> operation that changed a member of the <code>msgqid_ds</code> structure associated with the queue.

tuxTmsgLrPid

Syntax	INTEGER
Access	read-only
Description	Process identifier of the last process that read from the queue.

tuxTmsgLsPid

Syntax	INTEGER
Access	read-only
Description	Process identifier of the last process that wrote to the queue.

tuxTmsgQbytes

Syntax	INTEGER
Access	read-only
Description	Maximum number of bytes allowed on the queue.

tuxTmsgQnum

Syntax	INTEGER
Access	read-only
Description	Number of messages currently on the queue.

tuxTmsgRtime

Syntax	INTEGER
Access	read-only
Description	Time since the last read from the queue.

tuxTmsgStime

Syntax	INTEGER
Access	read-only
Description	Time since the last write to the queue.

tuxTqueueTable

The tuxTqueueTable group represents runtime attributes of queues in an application. These attribute values identify and characterize allocated Tuxedo System/T request queues associated with servers in a running application. They also track statistics related to application workloads associated with each queue object. The index into this table is tuxTqueueRqAddr. Objects in this table are only accessible through a Tuxedo SNMP agent installed on the local machine

Variable Name	Object ID
tuxTqueueRqAddr	.1.3.6.1.4.1.140.300.7.1.1.1
tuxTqueueState	.1.3.6.1.4.1.140.300.7.1.1.2
tuxTqueueRqId	.1.3.6.1.4.1.140.300.7.1.1.3
tuxTqueueSrvrCnt	.1.3.6.1.4.1.140.300.7.1.1.4
tuxTqueueTotNqueued	.1.3.6.1.4.1.140.300.7.1.1.5
tuxTqueueTotWkQueued	.1.3.6.1.4.1.140.300.7.1.1.6
tuxTqueueSource	.1.3.6.1.4.1.140.300.7.1.1.7
tuxTqueueNqueued	.1.3.6.1.4.1.140.300.7.1.1.8
tuxTqueueWkQueued	.1.3.6.1.4.1.140.300.7.1.1.9

tuxTqueueRqAddr

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Symbolic address of the request queue. Servers with the same tuxTsrvrRqAddr attribute value are grouped into a Multiple Server Single Queue (MSSQ) set. Attribute values returned with a tuxTqueueTable object apply to all active servers associated with this symbolic queue address.

tuxTqueueState

Syntax INTEGER { active(1) | migrating(2) | suspended(3) | partitioned(4) }

Access read-only

Description The values for GET and SET operations are as follows:

GET: {active(1)|migrating(2)|suspended(3)|partitioned(4)}

A GET operation retrieves runtime information for the selected `tuxTqueueTable` instance(s). The `tuxTqueueTable` group does not address configuration information directly. Configuration-related attributes discussed here must be set as part of the related `tuxTsrvrTbl` instances. The following states indicate the meaning of a `tuxTqueueState` returned in response to a GET request. States not listed are not returned.

`active(1)`

At least one server associated with this `tuxTqueueTable` instance is `active(1)`.

`migrating(2)`

The server(s) associated with this `tuxTqueueTable` instance is currently in the `migrating(2)` state. See the `tuxTsrvrTbl` group for more details on this state.

`suspended(3)`

The server(s) associated with this `tuxTqueueTable` instance is currently in the `suspended(3)` state. See the `tuxTsrvrTbl` group for more details on this state.

`partitioned(4)`

The server(s) associated with this `tuxTqueueTable` instance is currently in the `partitioned(4)` state. See the `tuxTsrvrTbl` group for more details on this state.

SET:

A SET operation updates runtime information for the selected `tuxTqueueTable` object. State changes are not allowed when updating `tuxTqueueTable` object information. Modification of an existing `tuxTqueueTable` object is allowed only when the object is in the `active(1)` state.

tuxTqueueRqld

Syntax	INTEGER
Access	read-only
Description	UNIX system message queue identifier.

Note: This is a UNIX system specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTqueueSrvrCnt

Syntax	INTEGER
Access	read-only
Description	Number of active servers associated with this queue.

tuxTqueueTotNqueued

Syntax	INTEGER
Access	read-only
Description	The sum of the queue lengths of this queue while it has been active. This sum includes requests enqueued to and processed by servers that are no longer active on the queue. Each time a new request is assigned to the queue, the sum is incremented by the length of the queue immediately before the new request is enqueued.

Note: If the `tuxTdomainLoadBalance` attribute is `no(2)` or the `tuxTdomainModel` attribute is `multi-machine(2)`, then `tuxTqueueTotNqueued` is not returned. In the same configuration, updates to this attribute are ignored. Consequently, when this attribute is returned `tuxTqueueSource` has the same value as the local host.

tuxTqueueTotWkQueued

Syntax	INTEGER
Access	read-only
Description	The sum of the workloads enqueued to this queue while it has been active. This sum includes requests enqueued to and processed by servers that are no longer active on the queue. Each time a new request is assigned to the queue, the sum is incremented by the workload on the queue immediately before the new request is enqueued.

Note: If the `tuxTdomainLoadBalance` attribute is `no(2)` or the `tuxTdomainModel` attribute is `multi-machine(2)`, then `tuxTqueueTotWkQueued` is not returned. In the same configuration, updates to this attribute are ignored. Consequently, when this attribute is returned `tuxTqueueSource` has the same value as the local host.

tuxTqueueSource

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Logical machine from which local attribute values are retrieved.

tuxTqueueNqueued

Syntax	INTEGER
Access	read-only
Description	Number of requests currently enqueued to this queue from the <code>tuxTqueueSource</code> logical machine. This value is incremented at enqueue time and decremented when the server dequeues the request.

Note: If the `tuxTdomainLoadBalance` attribute is `no(2)` or the `tuxTdomainModel` attribute is `multi-machine(2)`, then `tuxTqueueNqueued` is not returned. Consequently, when this attribute is returned `tuxTqueueSource` has the same value as the local host.

tuxTqueueWkQueued

Syntax	INTEGER
Access	read-only
Description	Workload currently enqueued to this queue from the tuxTqueueSource logical machine. If the tuxTdomainModel attribute is set to single-machine(1) and the tuxTdomainLoadBalance attribute is set to yes(1), then this attribute reflects the application-wide workload enqueued to this queue. However, if tuxTdomainModel is set to multi-machine(2) and tuxTdomainLoadBalance is set to yes(1), this attribute reflects the workload enqueued to this queue from the tuxTqueueSource logical machine during a recent timespan. This attribute is used for load balancing purposes. In order not discriminate against newly started servers, this attribute value is zeroed out on each machine periodically by the BBL.

tuxTroutingTable

The `tuxTroutingTable` group represents configuration objects of routing specifications for an application. These object values identify and characterize application data dependent routing criteria with respect to field names, buffer types, and routing definitions. This table also represents configuration objects for factory-based routing for WLE applications. `wleRoutingFieldType` is valid only for factory-based routing. `tuxTroutingBufType` is valid only for service-based routing.

The index into this table consists of the following attributes: `tuxTroutingName`, `tuxRoutingType`, and `tuxInternalIdx`.

`wleRoutingFieldType` is valid only for factory-based routing. This is supported only for WLE applications.

`tuxTroutingBufType` is valid only for service-based routing (either Tuxedo or WLE applications).

When specifying the index in SET requests, `tuxInternalIdx` is used as an index.

For factory-based routing, `tuxInternalIdx` must always have a value of `-`.

For service-based routing, `tuxInternalIdx` should equal the first 30 characters in `tuxTroutingBufType`.

To create a new row in the table, it is necessary to issue a SET request for a non-existing row specifying the values of all objects applicable to the `tuxRoutingType`.

Variable Name	Object ID
<code>tuxTroutingName</code>	.1.3.6.1.4.1.140.300.8.1.1.1
<code>tuxTroutingBufType</code>	.1.3.6.1.4.1.140.300.8.1.1.2
<code>tuxTroutingField</code>	.1.3.6.1.4.1.140.300.8.1.1.3
<code>tuxTroutingRanges</code>	.1.3.6.1.4.1.140.300.8.1.1.4
<code>tuxTroutingState</code>	.1.3.6.1.4.1.140.300.8.1.1.5
<code>tuxRoutingType</code>	.1.3.6.1.4.1.140.300.8.1.1.6
<code>wleRoutingFieldType</code>	.1.3.6.1.4.1.140.300.8.1.1.7
<code>tuxInternalIdx</code>	.1.3.6.1.4.1.140.300.8.1.1.8

tuxTroutingName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-write
Description	Routing criterion name.

Note: This object can be set only during row creation.

tuxTroutingBufType

Syntax	<i>DisplayString</i> (SIZE(1..256))
Access	read-write
Description	List of types and subtypes of data buffers for which this routing entry is valid. A maximum of 32 type/subtype combinations are allowed. The types are restricted to one of FML, VIEW, X_C_TYPE, or X_COMMON. No subtype can be specified for type FML, and subtypes are required for types VIEW, X_C_TYPE, and X_COMMON (* is not allowed). Note that subtype names should not contain semicolon, colon, comma, or asterisk characters. Duplicate type/subtype pairs cannot be specified for the same routing criterion name. More than one routing entry can have the same criterion name as long as the type/subtype pairs are unique. If multiple buffer types are specified for a single routing entry, the data types of the routing field for each buffer type must be the same.

Note: This object is applicable only for service-based routing.

Note: This object can be set only during row creation.

tuxTroutingField

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Routing field name.

For Service-based Routing: This field is assumed to be an FML buffer or view field name that is identified in an FML field table (using the FLDTBLDIR and FIELDTBLS environment) or an FML view table (using the VIEWDIR and VIEWFILES environment), respectively. This information is used to get the associated field value for data dependent routing during the sending of a message.

For factory-based routing: This is assumed to be a field that is specified in an NVList parameter to:

```
PortableServer::POA::create_reference_with_criteria
```

for an interface that has this factory routing criteria associated with it. See the WLE documentation for more details.

tuxTroutingRanges

Syntax	<i>DisplayString</i> (SIZE(1..2048))
Access	read-write
Description	The ranges and associated server groups for a routing criterion are as follows:

```
criterion: range: group
range: value | lower - upper | *
lower: value
upper: value
value: MIN | MAX | numeric | string
group: string | *
numeric: [+ | -]digits[.digits][e | E[ | + | - ] digit
digit: 0-9
digits: digit[digit]
```

\ can be used to escape the single-quote character in strings.

lower must be less than *upper*. A group specified as a string must specify a valid tuxTgroupName.

Note: Attribute values greater than 256 bytes in length disable interoperability with Tuxedo System/T Release 4.2.2 and earlier.

tuxTroutingState

Syntax	INTEGER { valid(1) unknown(2) invalid(3) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: valid(1)</p> <p>A GET operation retrieves configuration information for the selected <code>tuxTroutingTable</code> instance(s). The following state indicates the meaning of a <code>tuxTroutingState</code> returned in response to a GET request. States not listed are not returned.</p> <p>valid(1)</p> <p><code>tuxTroutingTable</code> instance is defined. Note that this is the only valid state for this class. Routing criteria are never active; rather, they are associated through the configuration with service names and are acted upon at runtime to provide data dependent routing.</p> <p>SET: invalid(3)</p> <p>A SET operation updates configuration information for the selected <code>tuxTroutingTable</code> instance. The following state indicates the meaning of a <code>tuxTroutingState</code> set in a SET request. States not listed cannot be set.</p> <p>invalid(3)</p> <p>Delete <code>tuxTroutingTable</code> instance for application. State change allowed only when in the <code>valid(1)</code> state. Successful return leaves the object in the <code>invalid(2)</code> state.</p>

tuxRoutingType

Syntax	INTEGER { service(1) factory(2) }
Access	read-write
Description	<p><code>service(1)</code></p> <p>Specifies that routing criteria apply to data-dependent routing for a BEA Tuxedo service.</p> <p><code>factory(2)</code></p>

Specifies that the routing criterion applies to factory-based routing for a CORBA interface.

Note: The routing type affects the validity and possible values for other attributes defined for this table.

Note: This object can be set during row creation only.

wleRoutingFieldType

Syntax	INTEGER { short(1) long(2) float(3) double(4) char(5) string(6) }
Access	read-write
Description	This object specifies the type of <code>tuxTroutingField</code> on which this routing criterion is defined. This is valid only for factory-based routing.

Note: This object is supported only for WLE.

Note: This object can be set only during row creation.

tuxInternalIdx

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	This object is used as an index of this table instead of <code>tuxTroutingBufType</code> (for service-based routing) or <code>tuxTroutingField</code> (for factory-based routing) to reduce the size of the index. Its value, for service-based routing (<code>tuxRoutingType = service(1)</code>) is equal to the first 30 characters in <code>tuxTroutingBufType</code> . In case of entries for factory-based routing (<code>tuxRoutingType = factory(2)</code>), the value is always <code>tuxTroutingField</code> .

Note: This object can be set only during row creation.

tuxTsrvrTbl

This group represents configuration and runtime attributes of servers within an application. These attribute values identify and characterize configured servers as well as provide runtime tracking of statistics and resources associated with each server object. The index into this table is provided by the attributes `tuxTsrvrGrpNo` and `tuxTsrvrId`. To create a new row in the table, it is necessary to issue a SET request specifying the values of at least `tuxTsrvrGrp` and `tuxTsrvrName`.

Variable Name	Object ID
<code>tuxTsrvrGrp</code>	.1.3.6.1.4.1.140.300.20.1.1.1
<code>tuxTsrvrId</code>	.1.3.6.1.4.1.140.300.20.1.1.2
<code>tuxTsrvrName</code>	.1.3.6.1.4.1.140.300.20.1.1.3
<code>tuxTsrvrGrpNo</code>	.1.3.6.1.4.1.140.300.20.1.1.4
<code>tuxTsrvrState</code>	.1.3.6.1.4.1.140.300.20.1.1.5
<code>tuxTsrvrBaseSrvId</code>	.1.3.6.1.4.1.140.300.20.1.1.6
<code>tuxTsrvrClOpt</code>	.1.3.6.1.4.1.140.300.20.1.1.7
<code>tuxTsrvrEnvFile</code>	.1.3.6.1.4.1.140.300.20.1.1.8
<code>tuxTsrvrGrace</code>	.1.3.6.1.4.1.140.300.20.1.1.9
<code>tuxTsrvrMaxgen</code>	.1.3.6.1.4.1.140.300.20.1.1.10
<code>tuxTsrvrMax</code>	.1.3.6.1.4.1.140.300.20.1.1.11
<code>tuxTsrvrMin</code>	.1.3.6.1.4.1.140.300.20.1.1.12
<code>tuxTsrvrRcmd</code>	.1.3.6.1.4.1.140.300.20.1.1.13
<code>tuxTsrvrRestart</code>	.1.3.6.1.4.1.140.300.20.1.1.14
<code>tuxTsrvrSequence</code>	.1.3.6.1.4.1.140.300.20.1.1.15
<code>tuxTsrvrSystemAccess</code>	.1.3.6.1.4.1.140.300.20.1.1.16

Variable Name	Object ID
tuxTsrvrConv	.1.3.6.1.4.1.140.300.20.1.1.17
tuxTsrvrReplyQ	.1.3.6.1.4.1.140.300.20.1.1.18
tuxTsrvrRpPerm	.1.3.6.1.4.1.140.300.20.1.1.19
tuxTsrvrRqAddr	.1.3.6.1.4.1.140.300.20.1.1.20
tuxTsrvrRqPerm	.1.3.6.1.4.1.140.300.20.1.1.21
tuxTsrvrGeneration	.1.3.6.1.4.1.140.300.20.1.1.22
tuxTsrvrPid	.1.3.6.1.4.1.140.300.20.1.1.23
tuxTsrvrRpid	.1.3.6.1.4.1.140.300.20.1.1.24
tuxTsrvrRqId	.1.3.6.1.4.1.140.300.20.1.1.25
tuxTsrvrTimeRestart	.1.3.6.1.4.1.140.300.20.1.1.26
tuxTsrvrTimeStart	.1.3.6.1.4.1.140.300.20.1.1.27
tuxTsrvrMinDispatchThreads	.1.3.6.1.4.1.140.300.20.1.1.40
tuxTsrvrMaxDispatchThreads	.1.3.6.1.4.1.140.300.20.1.1.50
tuxTsrvrThreadStackSize	.1.3.6.1.4.1.140.300.20.1.1.60
wleSrvrSrvType	.1.3.6.1.4.1.140.300.20.1.1.70

tuxTsrvrGrp

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Logical name of the server group. Server group names cannot contain an asterisk (*), comma, or colon.
Note: This object can be set only during row creation.	

tuxTsrvrId

Syntax	INTEGER (1..30001)
Access	read-write
Description	Unique (within the server group) server identification number.
	Note: This object can be set only during row creation.

tuxTsrvrName

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-write
Description	Name of the server executable file. The server identified by <code>tuxTsrvrName</code> runs on the machine(s) identified by the <code>tuxTgroupLMID</code> object for this server's server group. If a relative pathname is given, the search for the executable file is done first in <code>tuxTmachineAppDir</code> , then in <code>tuxTmachineTuxDir/bin</code> , then in <code>/bin</code> and <code>/usr/bin</code> , and then in <code><path></code> , where <code><path></code> is the value of the first <code>PATH=</code> line that appears in the machine environment file, if one exists. Note that the attribute value returned for an active server is always a full pathname.

tuxTsrvrGrpNo

Syntax	INTEGER (1..30000)
Access	read-only
Description	Group number associated with this server's group.

tuxTsrvrState

Syntax	INTEGER { active(1) inactive(2) migrating(3) cleaning(4) restarting(5) suspended(6) partitioned(7) dead(8) invalid(10) }
Access	read-write
Description	The values for GET and SET operations are as follows:

GET: active(1) | inactive(2) | migrating(3) | cleaning(4) | restarting(5) | suspended(6) | partitioned(7) | dead(8)

A GET operation retrieves configuration and runtime information for the selected `tuxTsrvrTbl` instance(s). The following states indicate the meaning of a `tuxTsrvrState` returned in response to a GET request. States not listed are not returned.

active(1)

`tuxTsrvrTbl` instance is defined and active. This is not an indication of whether the server is idle or busy. An active server with a non-0 length `tuxTsrvrCurService` attribute should be interpreted as a busy server, that is, one that is processing a service request.

inactive(2)

`tuxTsrvrTbl` instance is defined and inactive.

migrating(3)

`tuxTsrvrTbl` instance is defined and currently in a state of migration to the server group's secondary logical machine. The secondary logical machine is the one listed in `tuxTgroupLMID` attribute that does not match the `tuxTgroupCurLMID` object.

cleaning(4)

`tuxTsrvrTbl` instance is defined and currently being cleaned up after by the system due to an abnormal death. Note that restartable servers can enter this state if they exceed `tuxTsrvrMaxgen` starts/restarts within their `tuxTsrvrGrace` period.

restarting(5)

`tuxTsrvrTbl` instance is defined and currently being restarted by the system due to an abnormal death.

suspended(6)

`tuxTsrvrTbl` instance is defined and currently suspended, pending shutdown.

partitioned(7)

`tuxTsrvrTbl` instance is defined and active; however, the machine where the server is running is currently partitioned from the `tuxTdomainMaster` site.

dead(8)

tuxTsrvrTbl instance is defined, identified as active in the bulletin board, but currently not running due to an abnormal death. This state exists only until the BBL local to the server notices the death and takes action (restarting(5) | cleaning(4)).

SET: {active(1) | inactive(2) | dead(8) | invalid(10)}

A SET operation updates configuration and runtime information for the selected tuxTsrvrTbl instance. The following states indicate the meaning of a tuxTsrvrState set in a SET request. States not listed cannot be set.

active(1)

Activate the tuxTsrvrTbl instance. State change is allowed only when the server is in the inactive(2) state. (Servers in the migrating(3) state must be restarted by setting the tuxTgroupState to active(1).) Successful return leaves the object in the active(1) state.

inactive(2)

Deactivate the tuxTsrvrTbl instance. State change is allowed only when the server is in the active(1) state. Successful return leaves the object in the inactive(2) state.

dead(8)

Deactivate the tuxTsrvrTbl instance by sending the server a SIGTERM signal followed by a SIGKILL signal if the server is still running after 20 seconds. Note that by default, a SIGTERM signal causes the server to initiate orderly shutdown and the server becomes inactive even if it is restartable. If a server is processing a long running service or has chosen to disable the SIGTERM signal, then SIGKILL can be used and is treated by the system as an abnormal termination. State change is allowed only when the server is in the active(1) or suspended(6) state. Successful return leaves the object in the inactive(2), cleaning(4), or restarting(5) state.

invalid(10)

Delete tuxTsrvrTbl instance for application. State change is allowed only when the server is in the inactive(2) state. Successful return leaves the object in the invalid(10) state.

tuxTsrvrBaseSrvId

Syntax	INTEGER (1..30001)
Access	read-only
Description	Base server identifier. For servers with a <code>tuxTsrvrMax</code> attribute value of 1, this attribute is always the same as <code>tuxTsrvrId</code> . However, for servers with a <code>tuxTsrvrMax</code> value of greater than 1, this attribute indicates the base server identifier for the set of servers configured identically.

tuxTsrvrCLOpt

Syntax	<i>DisplayString</i> (SIZE(0..256))
Access	read-write
Description	Command line options to be passed to server when it is activated. See the <code>servopts(5)</code> manual page for details.

Note: Runtime modifications to this attribute do not affect a running server.

tuxTsrvrEnvFile

Syntax	<i>DisplayString</i> (SIZE(0..78))
Access	read-write
Description	Server specific environment file. See <code>tuxTmachineEnvFile</code> for a complete discussion of how this file is used to modify the environment.

Note: Runtime modifications to this attribute do not affect a running server.

tuxTsrvrGrace

Syntax	INTEGER
Access	read-write
Description	The period of time, in seconds, over which the <code>tuxTsrvrMaxgen</code> limit applies. This attribute is meaningful only for restartable servers, that is, if the <code>tuxTsrvrRestart</code> attribute is set to <code>yes(1)</code> . When a restarting server would exceed the <code>tuxTsrvrMaxgen</code> limit but the <code>tuxTsrvrGrace</code> period has expired, the system resets

the current generation (`tuxTsrvrGeneration`) to 1 and resets the initial boot time (`tuxTsrvrTimeStart`) to the current time. A value of 0 for this attribute indicates that a server should always be restarted.

Note that servers sharing a request queue (that is, equal values for `tuxTsrvrRqAddr`) should have equal values for this attribute. If they do not, then the first server activated establishes the runtime value associated with all servers on the queue.

Note: Runtime modifications to this attribute affect a running server and all other active servers with which it is sharing a request queue. However, only the selected server's configuration parameter is modified. Thus, the behavior of the application depends on the order of boot in subsequent activations unless the administrator ensures that all servers sharing a queue have the same value for this attribute.

tuxTsrvrMaxgen

Syntax `INTEGER (0..256)`

Access read-write

Description Number of generations allowed for a restartable server (`tuxTsrvrRestart == yes(1)`) over the specified grace period (`tuxTsrvrGrace`). The initial activation of the server counts as one generation and each restart also counts as one. Processing after the maximum generations is exceeded is discussed above with respect to `tuxTsrvrGrace`.

Note that servers sharing a request queue (that is, equal values for `tuxTsrvrRqAddr`) should have equal values for this attribute. If they do not, then the first server activated establishes the runtime value associated with all servers on the queue.

Note: Runtime modifications to this attribute affect a running server and all other active servers with which it is sharing a request queue. However, only the selected server's configuration parameter is modified. Thus, the behavior of the application depends on the order of boot in subsequent activations unless the administrator ensures that all servers sharing a queue have the same value for this attribute.

tuxTsrvrMax

Syntax	INTEGER (1..1001)
Access	read-write
Description	Maximum number of occurrences of the server to be booted. Initially, <code>tmboot(1)</code> boots <code>tuxTsrvrMin</code> objects of the server, and additional objects can be started individually (by starting a particular server id) or through automatic spawning (conversational servers only). Runtime modifications to this attribute affect all running servers in the set of identically configured servers (see <code>tuxTsrvrBaseSrvId</code> above) as well as the configuration definition of the server.

tuxTsrvrMin

Syntax	INTEGER (1..1001)
Access	read-write
Description	Minimum number of occurrences of the server to be booted by <code>tmboot(1)</code> . If a <code>tuxTsrvrRqAddr</code> is specified and <code>tuxTsrvrMin</code> is greater than 1, then the servers form an MSSQ set. The server identifiers for the servers are <code>tuxTsrvrId</code> up to <code>tuxTsrvrId + tuxTsrvrMax - 1</code> . All occurrences of the server have the same sequence number, as well as any other server parameters.

Note: Runtime modifications to this attribute do not affect a running server.

tuxTsrvrRcmd

Syntax	<i>DisplayString</i> (SIZE(0..78))
Access	read-write
Description	Application-specified command to be executed in parallel with the system restart of an application server. This command must be an executable file.

Note that servers sharing a request queue (that is, equal values for `tuxTsrvrRqAddr`) should have equal values for this attribute. If they do not, then the first server activated establishes the runtime value associated with all servers on the queue.

Note: Runtime modifications to this attribute affect a running server and all other active servers with which it shares a request queue. However, only the selected server's configuration parameter is modified. Thus, the behavior of the

application depends on the order of boot in subsequent activations unless the administrator ensures that all servers sharing a queue have the same value for this attribute.

tuxTsrvrRestart

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	<p>Restartable yes(1) or non-restartable no(2) server. If server migration is specified for this server group (tuxTdomainOptions = migrate(2) and tuxTgroupLMID with alternate site), this attribute must be set to yes(1).</p> <p>Note that servers sharing a request queue (that is, equal values for tuxTsrvrRqAddr) should have equal values for this attribute. If they do not, the first server activated establishes the runtime value associated with all servers on the queue.</p> <p>Note: Runtime modifications to this attribute affect a running server and all other active servers with which it shares a request queue. However, only the selected server's configuration parameter is modified. Thus, the behavior of the application depends on the order of boot in subsequent activations unless the administrator ensures that all servers sharing a queue have the same value for this attribute.</p>

tuxTsrvrSequence

Syntax	INTEGER (1..10000)
Access	read-write
Description	<p>Specifies when this server should be booted (tmboot(1)) or shutdown (tmshutdown(1)) relative to other servers. If two servers are given the same sequence number, it is possible for tmboot(1) to boot them in parallel and for tmshutdown(1) to shut them down in parallel. tuxTsrvrTbl instances added without a tuxTsrvrSequence attribute specified or with an invalid value have one generated for them that is 10,000 or more and is higher than any other automatically selected default value. Servers are booted by tmboot(1) in increasing order of sequence number and shutdown by tmshutdown(1) in decreasing order. Runtime modifications to this attribute affect only tmboot(1) and tmshutdown(1) and affect the order in which running servers can be shutdown by a subsequent invocation of tmshutdown(1).</p>

tuxTsrvrSystemAccess

Syntax	INTEGER { fastpath(1) protected(2) }
Access	read-write
Description	Mode used by System/T libraries within this server process to gain access to System/T's internal tables. See <code>tuxTdomainSystemAccess</code> for a complete discussion of this attribute.

Note: Runtime modifications to this attribute do not affect a running server.

tuxTsrvrConv

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	Conversational server <code>yes(1)</code> or request/response server <code>no(2)</code> .

tuxTsrvrReplyQ

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	Specifies whether to allocate a separate reply queue for the server (<code>tuxTsrvrReplyQ == yes(1)</code>). MSSQ servers that expect to receive replies should set this attribute to <code>yes(1)</code> .

tuxTsrvrRpPerm

Syntax	<i>DisplayString</i> (SIZE(4))
Access	read-write
Description	UNIX system permissions for the server's reply queue. If a separate reply queue is not allocated (<code>tuxTsrvrReplyQ == no(2)</code>), this attribute is ignored. This is a string representation of octal numbers starting with a leading 0 0001 through 0777.

Note: This is a UNIX system-specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTsrvrRqAddr

Syntax	<i>DisplayString</i> (SIZE(0..30))
Access	read-write
Description	Symbolic address of the request queue for the server. Specifying the same <code>tuxTsrvrRqAddr</code> attribute value for more than one server is the way multiple server, single queue (MSSQ) sets are defined. Servers with the same <code>tuxTsrvrRqAddr</code> attribute value must be in the same server group.

tuxTsrvrRqPerm

Syntax	<i>DisplayString</i> (SIZE(4))
Access	read-write
Description	UNIX system permissions for the server's request queue. This is a string representation of octal numbers starting with a leading 0 0001 through 0777. Note: This is a UNIX system specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTsrvrGeneration

Syntax	INTEGER (1..32768)
Access	read-only
Description	Generation of the server. When a server is initially booted via <code>tmboot(1)</code> or activated through the SNMP agent, its generation is set to 1. Each time the server dies abnormally and is restarted, its generation is incremented. Note that when <code>tuxTsrvrMaxgen</code> is exceeded and <code>tuxTsrvrGrace</code> has expired, the server is restarted with the generation reset to 1.

tuxTsrvrPid

Syntax	INTEGER
Access	read-only

Description UNIX system process identifier for the server. Note that this cannot be a unique attribute since servers can be located on different machines allowing for duplication of process identifiers.

Note: This is a UNIX system specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTsrvrRpid

Syntax INTEGER

Access read-only

Description UNIX system message queue identifier for the server's reply queue. If a separate reply queue is not allocated (`tuxTsrvrReplyQ == no(2)`), this attribute value is the same as `tuxTsrvrRqId`.

Note: This is a UNIX system specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTsrvrRqId

Syntax INTEGER

Access read-only

Description UNIX system message queue identifier for the server's request queue. If a separate reply queue is not allocated (`tuxTsrvrReplyQ == no(2)`), this attribute value is the same as `tuxTsrvrRpid`.

Note: This is a UNIX system specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTsrvrTimeRestart

Syntax INTEGER

Access read-only

Description Time, in seconds, since 00:00:00 UTC, January 1, 1970, as returned by the `time(2)` system call on local host, when the server was last started or restarted.

tuxTsrvrTimeStart

Syntax	INTEGER
Access	read-only
Description	Time, in seconds, since 00:00:00 UTC, January 1, 1970, as returned by the <code>time(2)</code> system call on local host, when the server was first started. Restarts of the server do not reset this value; however, if <code>tuxTsrvrMaxgen</code> is exceeded and <code>tuxTsrvrGrace</code> is expired, this attribute is reset to the time of the restart.

tuxTsrvrMinDispatchThreads

Syntax	INTEGER (1..999)
Access	read-only
Description	Specifies the number of server dispatch threads started on the initial server boot. This parameter is effective only if the server has been built with the <code>buildserver -t</code> command.

The separate dispatcher thread that is used when the value of `tuxTsrvrMaxDispatchThreads` is greater than one is not counted as part of the `tuxTsrvrMinDispatchThreads` value. The value of `tuxTsrvrMinDispatchThreads` must be less than the value of `tuxTsrvrMaxDispatchThreads`. If this parameter is not specified, the default is 0.

tuxTsrvrMaxDispatchThreads

Syntax	INTEGER (0..999)
Access	read-only
Description	Specifies the maximum number of concurrently dispatched threads that each server process can spawn. This parameter is effective only if the server has been built with the <code>buildserver -t</code> command.

If `tuxTsrvrMaxDispatchThreads` is greater than one, a separate dispatcher thread is used and does not count against this limit. The value of `tuxTsrvrMinDispatchThreads` must be less than the value of `tuxTsrvrMaxDispatchThreads`. If this parameter is not specified, the default is 1.

tuxTsrvrThreadStackSize

Syntax	INTEGER (0..2147483647)
Access	read-write
Description	If this parameter is not specified or if the value specified is 0, the operating system default is used. This option affects the server only when a value greater than 1 is specified for <code>tuxTsrvrMaxDispatchThreads</code> .

wleSrvrSrvType

Syntax	INTEGER { java (1) }
Access	read-only
Description	If <code>SRVTYPE</code> is set to <code>java</code> , it indicates that the server is a Java server. If at least one <code>MODULE</code> attribute is specified for this server or the name of the server is <code>JavaServer</code> (not case-sensitive), <code>java</code> becomes the default value.

tuxTsrvrTblExt

An extension of the tuxTsrvrTbl. Objects in this table are only accessible through a Tuxedo SNMP agent installed on the local machine

Variable Name	Object ID
tuxTsrvrIdExt	.1.3.6.1.4.1.140.300.20.2.1.1
tuxTsrvrGrpNoExt	.1.3.6.1.4.1.140.300.20.2.1.2
tuxTsrvrNumConv	.1.3.6.1.4.1.140.300.20.2.1.3
tuxTsrvrNumDeque	.1.3.6.1.4.1.140.300.20.2.1.4
tuxTsrvrNumEnque	.1.3.6.1.4.1.140.300.20.2.1.5
tuxTsrvrNumPost	.1.3.6.1.4.1.140.300.20.2.1.6
tuxTsrvrNumReq	.1.3.6.1.4.1.140.300.20.2.1.7
tuxTsrvrNumSubscribe	.1.3.6.1.4.1.140.300.20.2.1.8
tuxTsrvrNumTran	.1.3.6.1.4.1.140.300.20.2.1.9
tuxTsrvrTranAbt	.1.3.6.1.4.1.140.300.20.2.1.10
tuxTsrvrTranCmt	.1.3.6.1.4.1.140.300.20.2.1.11
tuxTsrvrTotReqC	.1.3.6.1.4.1.140.300.20.2.1.12
tuxTsrvrTotWorkL	.1.3.6.1.4.1.140.300.20.2.1.13
tuxTsrvrClgLmid	.1.3.6.1.4.1.140.300.20.2.1.14
tuxTsrvrClgLpid	.1.3.6.1.4.1.140.300.20.2.1.15
tuxTsrvrClgLreply	.1.3.6.1.4.1.140.300.20.2.1.16
tuxTsrvrCmtRet	.1.3.6.1.4.1.140.300.20.2.1.17
tuxTsrvrCurConv	.1.3.6.1.4.1.140.300.20.2.1.18
tuxTsrvrCurReq	.1.3.6.1.4.1.140.300.20.2.1.19

Variable Name	Object ID
tuxTsrvrCurService	.1.3.6.1.4.1.140.300.20.2.1.20
tuxTsrvrCurTime	.1.3.6.1.4.1.140.300.20.2.1.21
tuxTsrvrLastGrp	.1.3.6.1.4.1.140.300.20.2.1.22
tuxTsrvrSvcTimeOut	.1.3.6.1.4.1.140.300.20.2.1.23
tuxTsrvrTimeLeft	.1.3.6.1.4.1.140.300.20.2.1.24
tuxTsrvrTranLev	.1.3.6.1.4.1.140.300.20.2.1.25
tuxTsrvrStateExt	.1.3.6.1.4.1.140.300.20.2.1.26
tuxTsrvrGrpExt	.1.3.6.1.4.1.140.300.20.2.1.27
wleSrvrCurObjsExt	.1.3.6.1.4.1.140.300.20.2.1.32
wleSrvrCurInterfaceExt	.1.3.6.1.4.1.140.300.20.2.1.37
wleSrvrClassPath	.1.3.6.1.4.1.140.300.20.2.1.50
wleSrvrjavaHeap	.1.3.6.1.4.1.140.300.20.2.1.60
wleSrvrjavaHeapuse	.1.3.6.1.4.1.140.300.20.2.1.70
wleSrvrjavaVendor	.1.3.6.1.4.1.140.300.20.2.1.80
wleSrvrjavaVersion	.1.3.6.1.4.1.140.300.20.2.1.90
tuxTsrvrCurDispatchThreads	.1.3.6.1.4.1.140.300.20.2.1.100
tuxTsrvrHwDispatchThreads	.1.3.6.1.4.1.140.300.20.2.1.110
tuxTsrvrNumDispatchThreads	.1.3.6.1.4.1.140.300.20.2.1.120

tuxTsrvrIdExt

Syntax	INTEGER (1..30001)
Access	read-only
Description	Unique (within the server group) server identification number.

tuxTsrvrGrpNoExt

Syntax	INTEGER (1..30000)
Access	read-only
Description	Group number associated with this server's group.

tuxTsrvrNumConv

Syntax	INTEGER
Access	read-only
Description	Number of conversations initiated by this server through <code>tpconnect(3)</code> .

tuxTsrvrNumDeque

Syntax	INTEGER
Access	read-only
Description	Number of dequeue operations initiated by this server through <code>tpdequeue(3)</code> .

tuxTsrvrNumEnque

Syntax	INTEGER
Access	read-only
Description	Number of enqueue operations initiated by this server through <code>tpenqueue(3)</code> .

tuxTsrvrNumPost

Syntax	INTEGER
Access	read-only
Description	Number of postings initiated by this server through <code>tppost(3)</code> .

tuxTsrvrNumReq

Syntax	INTEGER
Access	read-only
Description	Number of requests made by this server through <code>tpcall(3)</code> or <code>tpacall(3)</code> .

tuxTsrvrNumSubscribe

Syntax	INTEGER
Access	read-only
Description	Number of subscriptions made by this server through <code>tpsubscribe(3)</code> .

tuxTsrvrNumTran

Syntax	INTEGER
Access	read-only
Description	Number of transactions begun by this server since its last (re)start.

tuxTsrvrTranAbt

Syntax	INTEGER
Access	read-only
Description	Number of transactions aborted by this server since its last (re)start.

tuxTsrvrTranCmt

Syntax	INTEGER
Access	read-only
Description	Number of transactions committed by this server since its last (re)start.

tuxTsrvrTotReqC

Syntax	INTEGER
Access	read-only
Description	Total number of requests completed by this server. For conversational servers (<code>tuxTsrvrConv == yes(1)</code>), this attribute value indicates the number of completed incoming conversations. This is a runtime attribute that is kept across server restart but is lost at server shutdown.

tuxTsrvrTotWorkL

Syntax	INTEGER
Access	read-only
Description	Total workload completed by this server. For conversational servers (<code>tuxTsrvrConv == yes(1)</code>), this attribute value indicates the workload of completed incoming conversations. This is a runtime attribute that is kept across server restart but is lost at server shutdown.

tuxTsrvrClitLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Logical machine for the initiating client or server. The initiating client or server is the process that made the service request on which the server is currently working.

tuxTsrvrClitPid

Syntax	INTEGER
Access	read-only
Description	UNIX system process identifier for the initiating client or server. Note: This is a UNIX system specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTsrvrClItReply

Syntax	INTEGER { yes(1) no(2) null(3) }
Access	read-only
Description	yes(1) The initiating client or server expects a reply . no(2) . The initiating client or server does not expect a reply.

tuxTsrvrCmtRet

Syntax	INTEGER { complete(1) logged(2) }
Access	read-only
Description	The setting of the TP_COMMIT_CONTROL characteristic for this server. See the description of the System/T ATMI function <code>tpscmt(3)</code> for details on this characteristic.

tuxTsrvrCurConv

Syntax	INTEGER
Access	read-only
Description	Number of conversations initiated by this server through <code>tpconnect(3)</code> that are still active.

tuxTsrvrCurReq

Syntax	INTEGER
Access	read-only
Description	Number of requests initiated by this server through <code>tpcall(3)</code> or <code>tpacall(3)</code> that are still active.

tuxTsrvrCurService

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-only
Description	Service name, if any, on which the server is currently working.

tuxTsrvrCurTime

Syntax	INTEGER
Access	read-only
Description	Current time, in seconds, since 00:00:00 UTC, January 1, 1970, as on the local host. This attribute can be used to compute elapsed time from the <code>tuxTsrvrTimeStart</code> and <code>tuxTsrvrTimeRestart</code> object values.

tuxTsrvrLastGrp

Syntax	INTEGER (1..30000)
Access	read-only
Description	Server group number (<code>tuxTgroupNo</code>) of the last service request made or conversation initiated from this server outward.

tuxTsrvrSvcTimeOut

Syntax	INTEGER
Access	read-only
Description	Time left in seconds, if any, for this server to process the current service request. A value of 0 for an active service indicates that no timeout processing is being done. See <code>tuxTsvcTimeOut</code> for more information.

tuxTsrvrTimeLeft

Syntax	INTEGER
Access	read-only

Description Time left, in seconds, for this server to receive the reply for which it is currently waiting before it times out. This timeout can be a transactional timeout or a blocking timeout.

tuxTsrvrTranLev

Syntax INTEGER

Access read-only

Description Current transaction level for this server. 0 indicates that the server is not currently involved in a transaction.

tuxTsrvrStateExt

Syntax INTEGER { active(1) | inactive(2) | migrating(3) | cleaning(4) |
restarting(5) | suspended(6) | partitioned(7) | dead(8) }

Access read-only

Description Refer to description of tuxTsrvrState for details.

tuxTsrvrGrpExt

Syntax DisplayString

Access read-only

Description Name of group to which this server belongs. This object is included for readability purposes only.

wleSrvrCurObjsExt

Syntax INTEGER

Access read-only

Description The number of entries in use in the bulletin board object table for this server.

wleSrvrCurInterfaceExt

Syntax DisplayString (SIZE(1..128))

Access read-only

Description The interface name of the interface currently active in this server.

wleSrvrClassPath

Syntax DisplayString (SIZE(0..2047))

Access read-only

Description The current CLASSPATH for the runtime.

wleSrvrjavaHeap

Syntax INTEGER

Access read-only

Description The heap size, as specified in the runtime options.

wleSrvrjavaHeapuse

Syntax INTEGER(1..100)

Access read-only

Description The percentage of heap space used.

wleSrvrjavaVendor

Syntax DisplayString (SIZE(0..30))

Access read-only

Description

wleSrvrjavaVersion

Syntax DisplayString (SIZE(0..30))

Access read-only

tuxTsrvrCurDispatchThreads

Syntax	INTEGER
Access	read-only
Description	Current number of active service dispatch threads for this server.

tuxTsrvrHwDispatchThreads

Syntax	INTEGER
Access	read-only
Description	Highest number of active service dispatch threads created for this server since its last restart. This number can differ from the number of service calls, because an administrator can specify parameters that control the caching of idle service threads.

tuxTsrvrNumDispatchThreads

Syntax	INTEGER
Access	read-only
Description	Total number of active service dispatch threads created for this server since its last restart.

tuxTsvcTbl

This represents configuration attributes of services within an application. These attribute values identify and characterize configured services. A `tuxTsvcTbl` object provides activation time configuration attributes for services not specifically configured as part of the `tuxTsvcGrp` group. The index into this table is `tuxTsvcName`. Objects in this group are only accessible through a Tuxedo SNMP agent installed on the local machine. To create a new row in the table, it is necessary to issue a SET request for a non-existing row in the table.

Variable Name	Object ID
tuxTsvcName	.1.3.6.1.4.1.140.300.10.1.1.1
tuxTsvcType	.1.3.6.1.4.1.140.300.10.1.1.2
tuxTsvcState	.1.3.6.1.4.1.140.300.10.1.1.3
tuxTsvcAutoTran	.1.3.6.1.4.1.140.300.10.1.1.4
tuxTsvcLoad	.1.3.6.1.4.1.140.300.10.1.1.5
tuxTsvcPrio	.1.3.6.1.4.1.140.300.10.1.1.6
tuxTsvcTimeOut	.1.3.6.1.4.1.140.300.10.1.1.7
tuxTsvcTranTime	.1.3.6.1.4.1.140.300.10.1.1.8
tuxTsvcBufType	.1.3.6.1.4.1.140.300.10.1.1.9
tuxTsvcRoutingName	.1.3.6.1.4.1.140.300.10.1.1.10
tuxTsvcEncryptionRequired	.1.3.6.1.4.1.140.300.10.1.1.20
tuxTsvcSignatureRequired	.1.3.6.1.4.1.140.300.10.1.1.30

tuxTsvcName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-write
Description	Service name.

Note: This object can be set only during row creation.

tuxTsvcType

Syntax	INTEGER { app(1) callable(2) system(3) unknown(4) }
Access	read-only
Description	Type of service.

app(1)

Indicates an application-defined service name.

callable(2)

Indicates a system-provided callable service.

system(3)

Indicates a system-provided and system-callable service. *system(3)* services are not available to application clients and servers for direct access.

tuxTsvcState

Syntax	INTEGER { active(1) inactive(2) invalid(3) }
Access	read-write
Description	The values for GET and SET operations are as follows:

GET: {active(1)|inactive(2)}

A GET operation retrieves configuration information for the selected *tuxTsvcTbl* instance(s). The following states indicate the meaning of a *tuxTsvcState* returned in response to a GET request. States not listed are not returned.

active(1)

tuxTsvcTbl instance is defined and at least one tuxTsvcGrp object with a matching tuxTsvcName value is active.

inactive(2)

tuxTsvcTbl instance is defined and no tuxTsvcGrp object with a matching tuxTsvcName value is active.

SET: invalid(3)

A SET operation updates configuration information for the selected tuxTsvcTbl instance. The following state indicates the meaning of a tuxTsvcState set in a SET request. States not listed cannot be set.

invalid(3)

Delete tuxTsvcTbl instance for application. State change is allowed only when the service is in the inactive(2) state. Successful return leaves the object in the invalid(3) state.

tuxTsvcAutoTran

Syntax INTEGER { yes(1) | no(2) }

Access read-write

Description Automatically begin a transaction.

(yes(1))

If the request is not already in transaction mode when a service request message is received for this service, automatically begin a transaction.

(no(2))

Do not automatically begin a transaction

Note: Runtime updates to this attribute are not reflected in active tuxTsvcGrp objects.

tuxTsvcLoad

Syntax INTEGER (1..32768)

Access read-write

Description This `tuxTsvcTbl` object imposes the indicated load on the system. Service loads are used for load balancing purposes, that is, queues with higher enqueued workloads are less likely to be chosen for a new request. Service loads have meaning only if the `tuxTdomainLoadBalance` is set to `yes(1)`.

Note: Runtime updates to this attribute are not reflected in active `tuxTsvcGrp` objects.

tuxTsvcPrio

Syntax `INTEGER (1..100)`

Access `read-write`

Description This `tuxTsvcTbl` object has the indicated dequeuing priority. If multiple service requests are waiting on a queue for servicing, the higher priority requests are serviced first.

Note: Runtime updates to this attribute are not reflected in active `tuxTsvcGrp` objects.

tuxTsvcTimeOut

Syntax `INTEGER`

Access `read-write`

Description Time limit (in seconds) for processing requests for this service name. Servers processing service requests for this service are abortively terminated (`kill -9`) if they exceed the specified time limit in processing the request. A value of 0 for this attribute indicates that the service should not be abortively terminated.

Note: Runtime updates to this attribute are not reflected in active `tuxTsvcGrp` objects.

Note: This attribute value is not enforced on Tuxedo System/T Release 4.2.2 sites or earlier.

tuxTsvcTranTime

Syntax `INTEGER`

Access	read-write
Description	Transaction timeout value (in seconds) for transactions automatically started for this <code>tuxTsvcTbl</code> object. Transactions are started automatically when a request not in transaction mode is received and the <code>tuxTsvcAutoTran</code> attribute value for the service is <code>yes(1)</code> .
Note:	Runtime updates to this attribute are not reflected in active <code>tuxTsvcGrp</code> objects.

tuxTsvcBufType

Syntax	<i>DisplayString</i> (SIZE(1..256))
Access	read-write
Description	<p><code>type1[:subtype1[,subtype2 . . .]][:type2[:subtype3[, . . .]]] . . .</code></p> <p>List of types and subtypes of data buffers accepted by this service. A maximum of 32 type/subtype combinations are allowed. Types of data buffers provided with Tuxedo System/T are FML (for FML buffers), VIEW, X_C_TYPE, or X_COMMON (for FMLviews), STRING (for NULL terminated character arrays), and CARRAY or X_OCTET (for a character array that is neither encoded nor decoded during transmission). Of these types, only VIEW, X_C_TYPE, and X_COMMON have subtypes. A VIEW subtype gives the name of the particular VIEW expected by the service. Application types and subtypes can also be added (see <code>tuxtypes(5)</code>). For a buffer type that has subtypes, “*” can be specified for the subtype to indicate that the service accepts all subtypes for the associated buffer type.</p> <p>A single service can only interpret a fixed number of buffer types, namely those found in its buffer type switch (see <code>tuxtypes(5)</code>). If the <code>tuxTsvcBufType</code> value is set to ALL, that service accepts all buffer types found in its buffer type switch.</p> <p>A type name can be 8 characters or less in length and a subtype name can be 16 characters or less in length. Note that type and subtype names should not contain semicolon, colon, comma, or asterisk characters.</p> <p>Note: This attribute value represents the buffer types that must be supported by each and every instance of an application service with this service name. Since this attribute value is processed at service activation time, updates to this attribute are allowed only when there are no active <code>tuxTsvcGrp</code> objects with matching service names.</p>

tuxTsvcRoutingName

Syntax	<i>DisplayString</i> (SIZE(0..15))
Access	read-write
Description	This <code>tuxTsvcTbl</code> object has the indicated routing criteria name. Active updates to this attribute are reflected in all associated <code>tuxTsvcGrp</code> objects.

tuxTsvcEncryptionRequired

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	If set to <code>yes</code> , every application service in this group requires an encrypted input message buffer.

tuxTsvcSignatureRequired

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	If set to <code>yes</code> , every application service in this group requires a valid digital signature on its input message buffer.

tuxTsvcGrp

The `tuxTsvcGrp` group represents configuration and runtime attributes of services/groups within an application. These attribute values identify and characterize configured services/groups as well as provide runtime tracking of statistics and resources associated with each object.

Both `tuxTsvcTbl` and `tuxTsvcGrp` define activation time attribute settings for service names within the application. When a new service is activated (advertised), either due to initial activation of a server or due to a call to `tpadvertise(3)`, the following hierarchy exists for determining the attribute values to be used at service startup time.

1. If a matching configured `tuxTsvcGrp` entry exists (matching service name and server group), the attributes defined in that object are used to initially configure the advertised service.
2. Otherwise, if a matching configured `tuxTsvcTbl` entry exists (matching service name), the attributes defined in that object are used to initially configure the advertised service.
3. Otherwise, if any configured `tuxTsvcGrp` entries are found with matching service name value, the first one found is used to initially configure the advertised service.
4. If none of the preceding cases is used, the system defaults for service attributes are used to initially configure the advertised service.

Objects in this group are only accessible through a Tuxedo SNMP agent installed on the local machine.

To create a new row in the table, it is necessary to issue a `SET` request that specifies at least `tuxTsvcGrpName`. The combination of values specified for `tuxTsvcGrpName` and `tuxTsvcGrpSvcName` in the `SET` request should not correspond to an existing row. If the value of `tuxTsvcSrvrId` is zero in the `SET` request, the service entry is configured but not activated (advertised). If `tuxTsvcSrvrId` is not set to zero, the service is activated using the value of `tuxTsvcSrvrId` to identify the server instance.

Variable Name	Object ID
tuxTsvcGrpSvcName	.1.3.6.1.4.1.140.300.10.2.1.1
tuxTsvcGrpName	.1.3.6.1.4.1.140.300.10.2.1.2
tuxTsvcGrpNo	.1.3.6.1.4.1.140.300.10.2.1.3
tuxTsvcGrpState	.1.3.6.1.4.1.140.300.10.2.1.4
tuxTsvcGrpAutoTran	.1.3.6.1.4.1.140.300.10.2.1.5
tuxTsvcGrpLoad	.1.3.6.1.4.1.140.300.10.2.1.6
tuxTsvcGrpPrio	.1.3.6.1.4.1.140.300.10.2.1.7
tuxTsvcGrpSvcTimeOut	.1.3.6.1.4.1.140.300.10.2.1.8
tuxTsvcGrpTranTime	.1.3.6.1.4.1.140.300.10.2.1.9
tuxTsvcSrvrLmid	.1.3.6.1.4.1.140.300.10.2.1.10
tuxTsvcSrvrRqAddr	.1.3.6.1.4.1.140.300.10.2.1.11
tuxTsvcSrvrId	.1.3.6.1.4.1.140.300.10.2.1.12
tuxTsvcrName	.1.3.6.1.4.1.140.300.10.2.1.13
tuxTsvcSrvrNcompleted	.1.3.6.1.4.1.140.300.10.2.1.14
tuxTsvcSrvrNqueued	.1.3.6.1.4.1.140.300.10.2.1.15

tuxTsvcGrpSvcName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-only
Description	Service name.

tuxTsvcGrpName

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Server group name. Server group names cannot contain an asterisk.

tuxTsvcGrpNo

Syntax	INTEGER (1..29999)
Access	read-write
Description	Server group number.

tuxTsvcGrpState

Syntax	INTEGER { active(1) inactive(2) invalid(3) }
Access	read-write
Description	The values for GET and SET operations are as follows:

GET: active(1) | inactive(2)

A GET operation retrieves configuration information for the selected tuxTsvcGrpState instance(s). The following states indicate the meaning of a tuxTsvcGrpState returned in response to a GET request. States not listed are not returned.

active(1)

At least one instance is active, suspended, or partitioned.

inactive(2)

tuxTsvcGrp instance defined and inactive.

SET: invalid(3)

A SET operation removes the corresponding tuxTsvcGrp instance. When a tuxTsvcGrp instance is deleted it also removes the associated tuxTsvcSrvr instances that correspond to server instances that are a part of the group advertising this service. This transition is permissible only in inactive(2) state.

tuxTsvcGrpAutoTran

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	Automatically begin a transaction (yes(1)) when a service request message is received for this service if the request is not already in transaction mode.

tuxTsvcGrpLoad

Syntax	INTEGER (1..32767)
Access	read-write
Description	This <code>tuxTsvcGrp</code> instance imposes the indicated load on the system. Service loads are used for load balancing purposes, that is, queues with higher enqueued workloads are less likely to be chosen for a new request.

tuxTsvcGrpPrio

Syntax	INTEGER (1..100)
Access	read-write
Description	This <code>tuxTsvcGrp</code> object has the indicated dequeuing priority. If multiple service requests are waiting on a queue for servicing, the higher priority requests are serviced first.

tuxTsvcGrpSvcTimeOut

Syntax	INTEGER
Access	read-write
Description	Time limit (in seconds) for processing requests for this service name. Servers processing service requests for this service are abortively terminated (<code>kill -9</code>) if they exceed the specified time limit in processing the request. A value of 0 for this attribute indicates that the service should not be abortively terminated.

tuxTsvcGrpTranTime

Syntax	INTEGER
Access	read-write
Description	Transaction timeout value (in seconds) for transactions automatically started for this <code>tuxTsvcGrp</code> instance. Transactions are started automatically when a request not in transaction mode is received and the <code>tuxTsvcGrpAutoTran</code> attribute value for the service is <code>yes(1)</code> .

tuxTsvcSrvrLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Current logical machine on which an active server that offers this service is running.

tuxTsvcSrvrRqAddr

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Symbolic address of the request queue for an active server that offers this service. See <code>tuxTsrvrRqAddr</code> for more information on this attribute.

tuxTsvcSrvrId

Syntax	INTEGER (1..30000)
Access	read-write
Description	Server ID of which the service is a part. The user can also set the value of this object to activate (advertise) one or more <code>tuxTsvcGrp</code> instances. The value provided to set this object is used to activate another instance of <code>tuxTsvcGrp</code> .

tuxTsvcrName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-write
Description	Function name within the associated server assigned to process requests for this service. When this object is specified, the <code>tuxTsvcGrp</code> instance is activated (advertised). The user needs to specify the server ID of the corresponding server instance (<code>tuxTsvcSvrId</code>) in the SNMP index. This object can be updated only during row creation.

tuxTsvcSvrNcompleted

Syntax	INTEGER
Access	read-only
Description	Number of service requests completed with respect to the retrieved active or suspended instance since it was activated (advertised).
Note:	This attribute is returned only when <code>tuxTdomainLoadBalance</code> is equal to <code>yes(1)</code> .

tuxTsvcSvrNqueued

Syntax	INTEGER (0..32767)
Access	read-only
Description	Number of requests currently enqueued to this service. This attribute is incremented at enqueue time and decremented when the server dequeues the request.
Note:	This attribute is returned only when the <code>tuxTdomainModel</code> is set to <code>single-machine(1)</code> and the <code>tuxTdomainLoadBalance</code> attribute is set to <code>yes(1)</code> .

tuxTlistenTbl

This group represents runtime attributes of /T listener processes for a distributed application.

Variable Name	Object ID
tuxTlistenLmid	.1.3.6.1.4.1.140.300.21.1.1.1
tuxTlistenState	.1.3.6.1.4.1.140.300.21.1.1.2

tuxTlistenLmid

- Syntax

DisplayString (SIZE(1..30))
- Access

read-only
- Description

Logical machine identifier.

tuxTlistenState

- Syntax

INTEGER { inactive(2) | active(1) }
- Access

read-only
- Description

The values for GET and SET operations are as follows:

GET: {active(1)|inactive(2)}

A GET operation retrieves runtime information for the selected tuxTlistenTbl instance(s). The following states indicate the meaning of a tuxTlistenState returned in response to a GET request. States not listed are not returned.

active(1)

tuxTlistenTbl instance active.

inactive(2)

tuxTlistenTbl instance not active.

tuxTranTbl

This table represents runtime attributes of active transactions within the application. The following objects comprise the index for rows in this table: tuxTranIndx1, tuxTranIndx2, tuxTranIndx3, tuxTranIndx4, tuxTranIndx5. Objects in this table are accessible only through a Tuxedo SNMP agent running on the local machine.

Variable Name	Object ID
tuxTranCoordLmid	.1.3.6.1.4.1.140.300.23.1.1.1
tuxTpTranId	.1.3.6.1.4.1.140.300.23.1.1.2
tuxTranXid	.1.3.6.1.4.1.140.300.23.1.1.3
tuxTranIndx1	.1.3.6.1.4.1.140.300.23.1.1.4
tuxTranIndx2	.1.3.6.1.4.1.140.300.23.1.1.5
tuxTranIndx3	.1.3.6.1.4.1.140.300.23.1.1.6
tuxTranIndx4	.1.3.6.1.4.1.140.300.23.1.1.7
tuxTranIndx5	.1.3.6.1.4.1.140.300.23.1.1.8
tuxTranState	.1.3.6.1.4.1.140.300.23.1.1.9
tuxTranTimeOut	.1.3.6.1.4.1.140.300.23.1.1.10
tuxTranGrpCnt	.1.3.6.1.4.1.140.300.23.1.1.11
tuxTranGrpIndex	.1.3.6.1.4.1.140.300.23.1.1.12
tuxTranGrpNo	.1.3.6.1.4.1.140.300.23.1.1.13
tuxTranGstate	.1.3.6.1.4.1.140.300.23.1.1.14

tuxTranCoordLmid

Syntax *DisplayString* (SIZE(1..30))
Access read-only

Description Logical machine identifier of the server group responsible for coordinating the transaction.

tuxTpTranId

Syntax *DisplayString* (SIZE(2..78))

Access read-only

Description Transaction identifier as returned from `tpsuspend(3)` mapped to a string representation. The data in this field should not be interpreted directly by the user except for equality comparison.

tuxTranXid

Syntax *DisplayString* (SIZE(2..78))

Access read-only

Description Transaction identifier as returned from `tx_info(3)` mapped to a string representation. The data in this field should not be interpreted directly by the user except for equality comparison.

tuxTranIndx1

Syntax INTEGER

Access read-only

Description This number is purely for unique indexing of this table.

tuxTranIndx2

Syntax INTEGER

Access read-only

Description This number is purely for unique indexing of this table.

tuxTranIndx3

Syntax INTEGER

Access	read-only
Description	This number is purely for unique indexing of this table.

tuxTranIndx4

Syntax	INTEGER
Access	read-only
Description	This number is purely for unique indexing of this table.

tuxTranIndx5

Syntax	INTEGER
Access	read-only
Description	This number is purely for unique indexing of this table.

tuxTranState

Syntax	INTEGER { active(1) abort-only(2) aborted(3) com-called(4) ready(5) decided(6) suspended(7) }
Access	read-write
Description	The values for GET and SET operations are as follows:

GET: active(1) | abort-only(2) | aborted(3) | com-called(4) | ready(5) | decided(6) | suspended(7)

A GET operation retrieves runtime information for the selected `tuxTranTbl` instance(s). The following states indicate the meaning of a `tuxTranState` object. States not listed are not returned. Note that distinct objects pertaining to the same global transaction (equivalent transaction identifiers) can indicate differing states. In general, the state indicated on the coordinator's site (`tuxTranCoordLmid`) indicates the true state of the transaction. The exception is when a noncoordinator site notices a condition that transitions the transaction state to `abort-only(2)`. This transition is eventually propagated to the coordinator site and results in the rollback of the transaction, but this change cannot be immediately reflected on the coordinator site.

`active(1)`

The transaction is active.

`abort-only(2)`

The transaction has been identified for rollback on the retrieval site.

`aborted(3)`

The transaction has been identified for rollback and rollback has been initiated on the retrieval site.

`com-called(4)`

The initiator of the transaction has called `tpcommit(3)` and the first phase of two-phase commit has begun on the retrieval site.

`ready(5)`

All of the participating groups on the retrieval site have successfully completed the first phase of two-phase commit and are ready to be committed.

`decided(6)`

The second phase of the two-phase commit has begun on the retrieval site.

`suspended(7)`

The initiator of the transaction has suspended processing on the transaction. Note that this state is returned from the initiator's site only.

`SET: aborted(3)`

A `SET` operation updates runtime information for the selected `tuxTranTbl` instance. The following state indicates the meaning of a `tuxTranState` set in a `SET` request. States not listed cannot be set.

`aborted(3)`

Abort the `tuxTranTbl` instance for the application. State change is allowed only when the transaction is in the `active(1)`, `abort-only(2)`, or `com-called(4)` state. Cannot be accompanied by a change to `tuxTranGstate`. Successful return leaves the object in the `aborted(3)` state.

tuxTranTimeOut

Syntax	INTEGER
Access	read-only
Description	Time left (in seconds) before the transaction times out on the retrieval site. Note that this attribute value is returned only when the transaction state is <code>active(1)</code> .

tuxTranGrpCnt

Syntax	INTEGER
Access	read-only
Description	Number of groups identified as participants in the transaction by the information returned from the retrieval site.

tuxTranGrpIndex

Syntax	INTEGER
Access	read-only
Description	Index of the first group-specific attribute values (<code>tuxTranGrpNo</code> and <code>tuxTranGstate</code>) corresponding to this object.

tuxTranGrpNo

Syntax	INTEGER
Access	read-only
Description	Group number of the participating group.

tuxTranGstate

Syntax	INTEGER { <code>active(1)</code> <code>aborted(2)</code> <code>rd-only(3)</code> <code>ready(4)</code> <code>hcommit(5)</code> <code>habort(6)</code> <code>done(7)</code> <code>pre-prepare(8)</code> <code>post-abort(9)</code> <code>post-commit(10)</code> <code>unknown(11)</code> }
Access	read-write

Description The values for GET and SET operations are as follows:

GET: active(1) | aborted(2) | rd-only(3) | ready(4) | hcommit(5) | habort(6) | done(7)

A GET operation retrieves runtime information for the selected tuxTranTbl instance(s) pertaining to the indicated group. The following states indicate the meaning of a tuxTranGstate returned in response to a GET request. States not listed are not returned. Note that distinct objects pertaining to the same global transaction (equivalent transaction identifiers) can indicate differing states for individual groups. In general, the state indicated on the group's site indicates the true state of the group's participation in the transaction. The exception is when the coordinator site determines that the transaction should abort and sets each participant group state to aborted(2). This transition is propagated to the group's site and results in the rollback of the group's work in the transaction, but cannot be reflected immediately

active(1)

The transaction is active in the indicated group.

aborted(2)

The transaction has been identified for rollback and rollback has been initiated for the indicated group.

rd-only(3)

The group has successfully completed the first phase of two-phase commit and has performed only read operations on the resource manager, thus making it unnecessary to perform the second phase of commit for this group.

ready(4)

The group has successfully completed the first phase of two-phase commit and is ready to be committed.

hcommit(5)

The group has been heuristically committed. This might or might not agree with the final resolution of the transaction.

habort(6)

The group has been heuristically rolled back. This might or might not agree with the final resolution of the transaction.

done(7)

This group has completed the second phase of the two-phase commit.

`pre-prepare(8)`

Indicates that the transaction group contains WLE servers that have called `xa_end` (TMSUSPEND) during the course of transactional work and that commit processing is beginning. This state exists until either (1) All servers that called `xa_end` (TMSUSPEND) have caused a call to `xa_end` (TMSUCCESS), at which point the group state becomes ready, or (2) One of the target servers does a rollback of the transaction at which point the group state becomes either `post-abort(9)` or `aborted(2)`.

Note: This state is supported for WLE applications only.

`post-abort(9)`

Indicates that an WLE server called `xa_end` (TPFAIL) and that the TMS has not yet called `xa_rollback()`. In this case, other WLE servers that called `xa_end` (TMSUSPEND) are being notified by the TMS in order to clean up their associated CORBA objects.

Note: This state is supported for WLE applications only.

`post-commit(10)`

This state is not implemented yet.

Note: This state is supported for WLE applications only.

`SET: hcommit(5) | habort(6)`

A SET operation updates runtime information for the first group in the originating request within the selected `tuxTranTbl` instance. The following states indicate the meaning of a `tuxTranGstate` set in a SET request. States not listed cannot be set. State transitions are allowed only when performed within the object representing the group's site.

`hcommit(5)`

Heuristically commit the group's work as part of the indicated transaction. State change is allowed only when `tuxTranGstate` is ready, `tuxTranState` is ready, and the indicated group is not on the coordinator's site. Successful return leaves the object in the `hcommit(5)` state.

`habort(6)`

Heuristically rollback the group's work as part of the indicated transaction. State change is allowed only when `tuxTranGstate` is active(1) or

`ready(4)`, `tuxTranState` is `ready(4)`, and the indicated group is not on the coordinator's site. Successful return leaves the object in the `habort(6)` state.

tuxTulogTable

The tuxTulogTable group represents runtime attributes of userlog files within an application. The index into this table is tuxTulogSerNo. The values returned for objects in this table are controlled by the MIB control group tuxTulogCtrl.

Variable Name	Object ID
tuxTulogSerNo	.1.3.6.1.4.1.140.300.9.1.1.1
tuxTulogLmid	.1.3.6.1.4.1.140.300.9.1.1.2
tuxTulogPmid	.1.3.6.1.4.1.140.300.9.1.1.3
tuxTulogMmDdYy	.1.3.6.1.4.1.140.300.9.1.1.4
tuxTulogTime	.1.3.6.1.4.1.140.300.9.1.1.5
tuxTulogLine	.1.3.6.1.4.1.140.300.9.1.1.6
tuxTulogMsg	.1.3.6.1.4.1.140.300.9.1.1.7
tuxTulogTpTranId	.1.3.6.1.4.1.140.300.9.1.1.8
tuxTulogXid	.1.3.6.1.4.1.140.300.9.1.1.9
tuxTulogPid	.1.3.6.1.4.1.140.300.9.1.1.10
tuxTulogSeverity	.1.3.6.1.4.1.140.300.9.1.1.11
tuxTulogCat	.1.3.6.1.4.1.140.300.9.1.1.12
tuxTulogMsgNum	.1.3.6.1.4.1.140.300.9.1.1.13
tuxTulogProcName	.1.3.6.1.4.1.140.300.9.1.1.14
tuxTulogThreadID	.1.3.6.1.4.1.140.300.9.1.1.20
tuxTulogContextID	.1.3.6.1.4.1.140.300.9.1.1.30

tuxTulogSerNo

Syntax	INTEGER
Access	read-only
Description	A running serial number for the rows in tuxTulogTable.

tuxTulogLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Retrieval machine logical machine identifier.

tuxTulogPmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Physical machine identifier.

tuxTulogMmDdYy

Syntax	INTEGER
Access	read-only
Description	Month, day, and year of the log file.

tuxTulogTime

Syntax	INTEGER
Access	read-only
Description	Time at which the message was generated.

tuxTulogLine

Syntax	INTEGER
Access	read-only
Description	Line number of the message in the log file.

tuxTulogMsg

Syntax	<i>DisplayString</i> (SIZE(1..256))
Access	read-only
Description	The entire text of the userlog message as it appears in the userlog file.

tuxTulogTpTranId

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Transaction identifier as returned from <code>tpsuspend(3)</code> . The data in this field should not be interpreted directly by the user except for equality comparison. Messages not associated with transactions retrieve a 0-length string as the value for this attribute.

tuxTulogXid

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Transaction identifier as returned from <code>tx_info(3)</code> . The data in this field should not be interpreted directly by the user except for equality comparison. Messages not associated with transactions retrieve a 0-length string as the value for this attribute.

tuxTulogPid

Syntax	INTEGER
Access	read-only
Description	Process identifier of the client or server that generated the userlog message.

tuxTulogSeverity

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Severity of message, if any.

tuxTulogCat

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Catalog name from which the message was derived, if any.

tuxTulogMsgNum

Syntax	INTEGER
Access	read-only
Description	Catalog message number, if the message was derived from a catalog.

tuxTulogProcName

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Process name of the client or server that generated the userlog message.

tuxTulogThreadID

Syntax	INTEGER
Access	read-only
Description	Identifier for the thread that wrote this userlog message.

tuxTulogContextID

Syntax	INTEGER
Access	read-only
Description	Identifier for this particular application association.

tuxTulogCtrl

The values of objects in this group control the ulog messages returned by the tuxTulogTable.

Variable Name	Object ID
tuxTulogLmidCtrl	.1.3.6.1.4.1.140.300.9.2.1
tuxTulogPmidCtrl	.1.3.6.1.4.1.140.300.9.2.2
tuxTulogMmdyyCtrl	.1.3.6.1.4.1.140.300.9.2.3
tuxTulogTimeCtrl	.1.3.6.1.4.1.140.300.9.2.4
tuxTulogEndTimeCtrl	.1.3.6.1.4.1.140.300.9.2.5
tuxTulogLineCtrl	.1.3.6.1.4.1.140.300.9.2.6
tuxTulogMsgCtrl	.1.3.6.1.4.1.140.300.9.2.7
tuxTulogTptranIdCtrl	.1.3.6.1.4.1.140.300.9.2.8
tuxTulogXidCtrl	.1.3.6.1.4.1.140.300.9.2.9
tuxTulogPidCtrl	.1.3.6.1.4.1.140.300.9.2.10
tuxTulogSeverityCtrl	.1.3.6.1.4.1.140.300.9.2.11
tuxTulogCatCtrl	.1.3.6.1.4.1.140.300.9.2.12
tuxTulogMsgNumCtrl	.1.3.6.1.4.1.140.300.9.2.13
tuxTulogProcNameCtrl	.1.3.6.1.4.1.140.300.9.2.14

tuxTulogLmidCtrl

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Logical machine ID to qualify machine from where the userlog file is read for <code>tuxTulogTable</code> . By default, the ULOG files from the local host are returned, per the ULOGPFX. To revert to the default setting, set this object to <code>null</code> .

tuxTulogPmidCtrl

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Physical machine name to qualify the source machine for userlog messages to be listed in <code>tuxTulogTable</code> . By default, messages from all hosts within ULOG files qualified by <code>tuxTulogLmidCtrl</code> are returned. To revert to the default setting, set this object to <code>null</code> .

tuxTulogMmddyCtrl

Syntax	INTEGER
Access	read-write
Description	Date value to qualify userlog messages listed in <code>tuxTulogTable</code> . Default value is current date. To reset the value of the qualifier to its default, set this object to 0.

tuxTulogTimeCtrl

Syntax	INTEGER
Access	read-write
Description	Starting time of the time range for which the userlog messages are listed in <code>tuxTulogTable</code> . This number is calculated as under - "hrs*10000 + mins*100 + secs". The default value is 0.

tuxTulogEndTimeCtrl

Syntax	INTEGER
Access	read-write
Description	Ending time of the time range for which the userlog messages are listed in <code>tuxTulogTable</code> . This number is calculated as under - “hrs*10000 + mins*100 + secs”. By default, the maximum value is considered. To revert to the default setting, set this object to 0.

tuxTulogLineCtrl

Syntax	INTEGER
Access	read-write
Description	Beginning line number from which the userlog messages are listed in <code>tuxTulogTable</code> . By default, all messages are returned. To revert to the default setting, set this object to 0.

tuxTulogMsgCtrl

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Regular expression to qualify userlog messages listed in <code>tuxTulogTable</code> on the basis of the message body. By default, all messages are listed. To revert to the default setting, set this object to <code>null</code> .

tuxTulogTptranIdCtrl

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-write
Description	Value of <code>tuxTpTranId</code> to qualify messages to be displayed in the in <code>tuxTulogTable</code> . By default, all messages are returned. To revert to the default setting, set it to <code>null</code> .

tuxTulogXidCtrl

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Value of <code>tuxTranXid</code> to qualify messages to be displayed in the <code>tuxTulogTable</code> . By default, all messages are returned. To revert to the default setting, set it to <code>null</code> .

tuxTulogPidCtrl

Syntax	INTEGER
Access	read-write
Description	Value of process Id of the source to qualify messages to be displayed in the <code>tuxTulogTable</code> . By default, messages with any pid are listed. To revert to the default setting, set this object to 0.

tuxTulogSeverityCtrl

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Regular expression to qualify userlog messages to be listed in <code>tuxTulogTable</code> on the basis of message severity, if any. By default, messages with any severity are listed. To revert to the default setting, set this object to <code>null</code> .

tuxTulogCatCtrl

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Regular expression to qualify userlog messages to be listed in <code>tuxTulogTable</code> on the basis of the catalog name, if any. By default, messages from all catalogs are listed. To revert to the default setting, set this object to <code>null</code> .

tuxTulogMsgNumCtrl

Syntax INTEGER

Access read-write

Description Message number in catalog to qualify userlog messages to be listed in `tuxTulogTable`. By default, all message numbers are returned. To revert to the default setting, set this object to 0.

tuxTulogProcNameCtrl

Syntax *DisplayString* (SIZE(1..30))

Access read-write

Description Regular expression to qualify userlog messages to be listed in `tuxTulogTable` on the basis of the process name that generated the message, if known. By default, all messages are returned. To revert to the default setting, set this object to `null`.

tuxTnetGrpTbl

This table represents application attributes of network groups. Network groups are groups of logical machine IDs that can communicate over the network address defined in the `tuxTnetMapNaddr` object in the `tuxTnetMapTbl` table entry. For row creation, a SET request with `tuxTnetGrpName`, `tuxTnetGrpNo` and `tuxTnetGrpPrio` is required. `tuxTnetGrpNo` provides the index into this table.

Note: This table is supported only on Tuxedo 6.4 or later.

Variable Name	Object ID
tuxTnetGrpName	.1.3.6.1.4.1.140.300.28.1.1
tuxTnetGrpNo	.1.3.6.1.4.1.140.300.28.1.2
tuxTnetGrpState	.1.3.6.1.4.1.140.300.28.1.3
tuxTnetGrpPrio	.1.3.6.1.4.1.140.300.28.1.4

tuxTnetGrpName

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Logical name of the network group. A group name is a string of printable characters and cannot contain a pound sign (#), comma (,), colon (:), or newline character. This object can be updated only during row creation.

tuxTnetGrpNo

Syntax	INTEGER (1..8191)
Access	read-write
Description	Group identifier of the network group. This object can be updated only during row creation.

tuxTnetGrpState

Syntax	INTEGER { valid(1) invalid(2) }
Access	read-write
Description	<p>A GET request retrieves configuration information for the selected <code>tuxTnetGrpTbl</code> instance (or instances). The following states indicate the meaning of the value that is returned:</p> <p>GET: <code>valid(1)</code></p> <p>The instance is defined. This is the only valid state for this object.</p> <p>SET: <code>invalid(2)</code></p> <p>Delete the selected <code>tuxTnetGrpTbl</code> instance from the application.</p> <p>States not listed are not returned.</p>

tuxTnetGrpPrio

Syntax	INTEGER (1..8191)
Access	read-write
Description	<p>The priority band for this network group. All network groups that have an equivalent band priority are used in parallel.</p>

tuxTnetMapTbl

The instances in the tuxTnetMapTbl associate tuxTmachineLmids to an instance in the tuxTnetGrpTbl. The rows in this table identify which logical machines belong to which network groups. For row creation, a SET request with at least tuxTnetMapNaddr is needed. The index into this table is provided by tuxTnetMapGrpNo and tuxTnetMapLmid.

Note: This table is supported only on Tuxedo 6.4 or later.

Variable Name	Object ID
tuxTnetMapGrpName	.1.3.6.1.4.1.140.300.33.1.1
tuxTnetMapGrpNo	.1.3.6.1.4.1.140.300.33.1.2
tuxTnetMapLmid	.1.3.6.1.4.1.140.300.33.1.3
tuxTnetMapState	.1.3.6.1.4.1.140.300.33.1.4
tuxTnetMapNaddr	.1.3.6.1.4.1.140.300.33.1.5
tuxTnetMapMinEncryptBit	.1.3.6.1.4.1.140.300.33.1.6
tuxTnetMapMaxEncryptBit	.1.3.6.1.4.1.140.300.33.1.7

tuxTnetMapGrpName

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	The logical name of the network group. A group name is a string of printable characters and cannot contain a pound sign (#), comma (,), colon (:), or a newline character.

tuxTnetMapGrpNo

Syntax	INTEGER (1..8191)
Access	read-write
Description	Identifier for this logical network group. This object can be updated only during row creation.

tuxTnetMapLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	Logical machine name for this network mapping. This object can be updated only during row creation.

tuxTnetMapState

Syntax	Integer { valid(1) invalid(2) }
Access	read-write
Description	A GET request retrieves configuration information for the selected tuxTnetMapTbl instance (or instances). The following states indicate the meaning of the value of tuxTnetMapState that is returned:

- GET: valid(1)
- The instance is defined. This is the only valid state for this object.
- SET: invalid(2)
- Delete the selected tuxTnetMapTbl instance from the application. If any network links are active as a result of the mapping, they are disconnected. This disconnection can cause a state change in tuxTnetBridgeTbl instances associated with the network links.

States not listed are not returned.

tuxTnetMapNaddr

Syntax	<i>DisplayString</i> (SIZE (1..78))
Access	read-write
Description	Specifies the complete network address to be used by the BRIDGE process placed on the logical machine as its listening address. The listening address for a BRIDGE is the means by which it is contacted by other BRIDGE processes participating in a networked application, that is, if the value of <code>tuxTdomainOptions</code> is <code>lan(1)</code> . If the string is of the form <i>0xhex-digits</i> or <i>\\xhex-digits</i> , it must contain an even number of valid hexadecimal digits. These forms are translated internally into a character array containing the hexadecimal representation of the string specified. For TCP/IP addresses, either the <i>//hostname:port</i> or <i>#.#.#.#:port</i> format is used.

tuxTnetMapMinEncryptBit

Syntax	INTEGER { none(1) 40-bit(2) 128-bit(3) unknown(4) }
Access	read-write
Description	Specifies the required level of encryption when establishing a network link to this machine. none(1) No encryption. 40-bit(2) and 128-bit(3) These values specify the encryption key length (in bits). If this minimum level of encryption cannot be met, the attempt to establish the link fails. The default value is none(1). Modifications to this object do not affect network links that have already been established.

tuxTnetMapMaxEncryptBit

Syntax	Integer {none(1) 40-bit(2) 128-bit(3) unknown(4) }
Access	read-write
Description	Encryption can be negotiated up to the specified level when establishing a network link.

none(1)

No encryption .

40-bit(2) and 128-bit(3)

These values specify the encryption key length (in bits).

The default value is 128-bit(3). Modifications to this object do not affect network links that are already established.

tuxTserverCtxtTbl

This table represents configuration and runtime attributes of individual server dispatch contexts within an application. This class is defined for both single-context and multi-context servers. For single-context servers, the values in this class are repeated as part of the `tuxTsrvrTbl` class. The objects in this group are read-only.

These attribute values provide runtime tracking of statistics and resources associated with each server dispatch context.

Variable Name	Object ID
tuxTserverCtxtGrp	.1.3.6.1.4.1.140.300.34.1.1.10
tuxTserverCtxtServerID	.1.3.6.1.4.1.140.300.34.1.1.20
tuxTserverCtxtContextID	.1.3.6.1.4.1.140.300.34.1.1.30
tuxTserverCtxtCltLmId	.1.3.6.1.4.1.140.300.34.1.1.40
tuxTserverCtxtCltPid	.1.3.6.1.4.1.140.300.34.1.1.50
tuxTserverCtxtCltReply	.1.3.6.1.4.1.140.300.34.1.1.60
tuxTserverCtxtCmtRet	.1.3.6.1.4.1.140.300.34.1.1.70
tuxTserverCtxtCurConv	.1.3.6.1.4.1.140.300.34.1.1.80
tuxTserverCtxtCurReq	.1.3.6.1.4.1.140.300.34.1.1.90
tuxTserverCtxtCurService	.1.3.6.1.4.1.140.300.34.1.1.100

Variable Name	Object ID
tuxTserverCtxtLastGrp	.1.3.6.1.4.1.140.300.34.1.1.110
tuxTserverCtxtSvcTimeOut	.1.3.6.1.4.1.140.300.34.1.1.120
tuxTserverCtxtTimeLeft	.1.3.6.1.4.1.140.300.34.1.1.130
tuxTserverCtxtTranLev	.1.3.6.1.4.1.140.300.34.1.1.140

tuxTserverCtxtGrp

- Syntax *DisplayString (SIZE (1..30))*
- Access read-only
- Description Logical name of the server group. Server group names cannot contain an asterisk (*), comma (,), or colon (:).

tuxTserverCtxtServerID

- Syntax *INTEGER (SIZE (1..30000))*
- Access read-only
- Description Unique (within the server group) server identification number.

tuxTserverCtxtContextID

- Syntax *INTEGER (SIZE (-2..29999))*
- Access read-only
- Description Identifier of this particular server context.

tuxTserverCtxtCltLmId

- Syntax *INTEGER (SIZE (1..30))*
- Access read-only

Description Logical machine for the initiating client or server. The initiating client or server is the process that made the service request on which the server is currently working.

tuxTserverCtxtCltPid

Syntax *INTEGER*

Access read-only

Description UNIX system process identifier for the initiating client or server.

Limitation: This is a UNIX system-specific attribute that cannot be returned if the platform on which the application is being run is not UNIX-based.

tuxTserverCtxtCltReply

Syntax *INTEGER { yes (1) | no (2) }*

Access read-only

Description The initiating client or server is expecting a reply (*yes (1)*) or is not expecting a reply (*no (2)*).

tuxTserverCtxtCmtRet

Syntax *INTEGER { complete (1) | logged (2) }*

Access read-only

Description This is the setting of the TP_COMMIT_CONTROL characteristic for this server.

See the description of the BEA Tuxedo ATMI function `tpscmt (3c)` for details on this characteristic.

tuxTserverCtxtCurConv

Syntax *INTEGER*

Access read-only

Description Number of conversations initiated by this server through `tpconnect()` that are still active.

tuxTserverCtxtCurReq

Syntax	INTEGER
Access	read-only
Description	Number of requests initiated by this server through <code>tpcall()</code> or <code>tpacall()</code> that are still active.

tuxTserverCtxtCurService

Syntax	<i>DisplayString (SIZE (1..15))</i>
Access	read-only
Description	Service name, if any, on which the server is currently working.

tuxTserverCtxtLastGrp

Syntax	<i>INTEGER (1..29999)</i>
Access	read-only
Description	Time left (in seconds), if any, for this server to process the current service request. A value of 0 for an active service indicates that no time out processing is being done. See <code>tuxTsvcTbl: tuxTsvcTimeOut</code> for more information.

tuxTserverCtxtSvcTimeOut

Syntax	<i>INTEGER</i>
Access	read-only
Description	Server group number (<code>tuxTgroupTable: tuxTgroupNo</code>) of the last service request made or conversation initiated from this server outward.

tuxTserverCtxtTimeLeft

Syntax	<i>INTEGER</i>
Access	read-only

Description	Time left (in seconds) for this server to receive the reply for which it is currently waiting before it will time out. This timeout can be a transactional timeout or a blocking timeout.
-------------	---

tuxTserverCtxtTranLev

Syntax	INTEGER
--------	---------

Access	read-only
--------	-----------

Description	Current transaction level for this server. 0 indicates that the server is not currently involved in a transaction.
-------------	--

beaEventFilters

You can use the Tuxedo event filters to define a subset of Tuxedo event notifications to be generated for each Tuxedo or WLE domain being monitored. The columnar objects in the `beaEvtFilterTable` correspond to fields in `TMEVENT_FILTER` entries in the BEA Manager configuration file (`beamgr.conf`). Refer to the “Configuration Files” chapter in the *Agent Integrator Reference Manual*.

Variable Name	Object ID
<code>beaEvtFilterTblStatus</code>	<code>.1.3.6.1.4.1.140.300.14.1</code>
<code>beaEvtFilterTable</code>	<code>.1.3.6.1.4.1.140.300.14.2</code>

beaEvtFilterTblStatus

Syntax	<code>INTEGER { sync(1) dirty(2) }</code>
Access	read-write
Description	When the agent starts, this value is always <code>sync(1)</code> . If any change is done to <code>beaEvtFilterTable</code> through <code>SET</code> requests, the value of this object becomes <code>dirty(2)</code> and the changes made to <code>beaEvtFilterTable</code> do not take effect. The changes made to the <code>beaEvtFilterTable</code> take effect only when you set the value of this object to <code>sync(1)</code> . When you set the value to <code>sync(1)</code> , all changes since the last synchronization are applied to the event-processing module.

beaEvtFilterTable

This MIB group represents all the event filters defined for the SNMP Agent. These are used to determine the collection of events to be forwarded as SNMP trap notifications.

Note: Changes to this table are applied only once beaEvtFilterStatus is set to sync(1).

Variable Name	Object ID
beaEvtFilterId	.1.3.6.1.4.1.140.300.14.1.1.1
beaEvtAgentName	.1.3.6.1.4.1.140.300.14.1.1.2
beaEvtExpr	.1.3.6.1.4.1.140.300.14.1.1.3
beaEvtFilter	.1.3.6.1.4.1.140.300.14.1.1.4
beaEvtFilterState	.1.3.6.1.4.1.140.300.14.1.1.5

beaEvtFilterId

- Syntax

DisplayString (SIZE (1..16))
- Access

read-write
- Description

A unique identifier for the event filter within the filter table.
- Note:**

This object can be SET only during row creation.

beaEvtAgentName

- Syntax

DisplayString (SIZE (1..32))
- Access

read-only
- Description

This logical agent name of the agent supporting this filter. This object is provided only for user convenience since the MIB only returns the event filters for the agent that was queried.

beaEvtExpr

Syntax	<code>DisplayString (SIZE (1..255))</code>
Access	read-write
Description	<p>An event name expression. Consult the <i>BEA Tuxedo Reference Manual</i> entry for <code>recomp(3)</code> for the format of this expression. For a Tuxedo system event to be forwarded as an SNMP trap, its name should match this expression. Consult the <i>BEA Tuxedo Reference Manual</i> (EVENTS(5)) for a list of Tuxedo event names. The default for this object is all system events.</p>
Examples	<p><code>\.Sys.*</code></p> <p>matches all system events. (This is the default.)</p> <p><code>\.SysServer.*</code></p> <p>matches all system events related to servers.</p> <p>A value of <code>NONE</code> blocks all events from being forwarded by the selected agent and overrides any other filter table entries for the same logical agent name.</p>

beaEvtFilter

Syntax	<code>DisplayString (SIZE (1..255))</code>
Access	read-write
Description	<p>An event filter expression. Each Tuxedo event is accompanied by an FML buffer that contains pertinent information about the event. The buffer's contents are evaluated with respect to this filter, if it is present. The filter must evaluate to <code>TRUE</code> or the event is not forwarded.</p> <p>The SNMP Agent uses this attribute as an argument to <code>tpsubscribe()</code>. Please refer to the <i>BEA Tuxedo Reference Manual</i> for further information.</p>
Example	<pre>TA_EVENT_SEVERITY== 'ERROR' TA_EVENT_SEVERITY== 'WARN' TA_EVENT_SEVERITY!= 'INFO' TA_EVENT_LMID== 'SITE1'</pre> <p>This filter selects events with a severity of either <code>ERROR</code> or <code>WARNING</code>.</p>

beaEvtFilterState

Syntax	INTEGER { active(1) inactive(2) invalid(3) }
Access	read-write
Description	<p>This object denotes the current state of the event filter instance.</p> <p>GET {active(1) inactive(2)}</p> <p>A GET operation retrieves configuration and runtime information for the selected beaEvtFilterTbl instance(s). The following states indicate the meaning of a beaEvtFilterState returned in response to a GET request. States not listed are not returned.</p> <p>active(1)</p> <p>This filter is being used.</p> <p>inactive(2)</p> <p>This filter is not being used.</p> <p>SET {active(1) inactive(2) invalid(3)}</p> <p>A SET operation updates configuration and runtime information for the selected beaEvtFilterTbl instance. The following states indicate the meaning of a beaEvtFilterState set in a SET request. States not listed cannot be set.</p> <p>active(1)</p> <p>Activate the event filter. This can be done only when the filter is in the inactive(2) state.</p> <p>inactive(2)</p> <p>Inactivate the event filter. This can be done only when the filter is in the active(1) state.</p> <p>invalid(3)</p> <p>Inactivate (if active) and remove this event filter.</p>

3 Domains MIB

The Domains MIB uses improved group and attribute terminology to describe the interaction between domains. This improved terminology has also been applied to DMCONFIG file syntax.

These terminology improvements eliminate multiple uses of the term “domain” and introduce terms that more clearly describe the actions that occur. For example, the term access point defines an object through which you gain access to another object. Therefore, you access a remote domain through a remote domain access point, and remote domains gain access to a local domain through a local domain access point. The Domains MIB consists of the following groups.

Group Name	Description
tuxDmAclTable	Domains Access Control List
tuxDmConnectionTable	Domain Access Points Connection Status
tuxDmExportTable	Resources for Exporting to Remote Domains
tuxDmImportTable	Resources imported through access points
tuxDmLocalTable	Defines a local domain access point
tuxDmOsitpTable	Defines the local or remote OSI TP protocol
tuxDmPasswordTable	Inter-domain authentication
tuxDmPrincipalMapTable	For mapping principal names
tuxDmRemoteTable	Remote domain configuration information
tuxDmResourcesTable	Domains-specific configuration information
tuxDmRoutingTable	Routing criteria information

Group Name	Description
tuxDmrPrincipalTable	Configuration info for remote principal names
tuxDmSnaCRMTable	SNA-CRM-specific configuration info
tuxDmSnaLinkTable	snax-specific configuration info
tuxDmSnaStackTable	Defines SNA stack used by a specific SNA CRM
tuxDmTdomainTable	Defines the TDomain specific configuration
tuxDmTopenTable	BEA TOP END-Specific Configuration info
tuxDmTransactionTable	Info about transactions that span domains

tuxDmAclTable

This Tuxedo MIB group represents access control information for domains.

Variable Name	Object ID
tuxDmAclName	.1.3.6.1.4.1.140.300.80.1.1.10
tuxDmrAccessPointList	.1.3.6.1.4.1.140.300.80.1.1.20
tuxDmAclState	.1.3.6.1.4.1.140.300.80.1.1.30

tuxDmAclName

- Syntax *DisplayString (SIZE(1..15))*
- Access read-only
- Description The access control list name.

tuxDmrAccessPointList

- Syntax *DisplayString (SIZE(1..1000))*
- Access read-write
- Description The list of remote domain access points associated with this access control list. *tuxDmrAccessPointlist* is a comma-separated list of remote access point names (that is, the value of the *tuxDMRemoteDmAccessPoint* attribute of a valid *tuxDmRemote* object). The list can contain up to 50 remote access point identifier elements. Setting this attribute to “*” means that all the remote domains in the configuration are associated with this entry. *blank string* means no remote access points are associated with this entry. The default is - (equivalent of NULL string).

tuxDmAclState

- Syntax INTEGER { valid(1) | invalid (2) }
- Access read-write
- Description This object denotes the current state of the *tuxDmAcl* instance.

GET requests:

valid (1): tuxDmAc1 object is defined and inactive. This is the only valid state for this class. ACL groups are never active.

SET requests:

invalid (2): Delete.

tuxDmConnectionTable

This class represents the status of connections between domain access points.

Variable Name	Object ID
tuxDmConDmlAccessPoint	.1.3.6.1.4.1.140.300.90.1.1.10
tuxDmConDmrAccessPoint	.1.3.6.1.4.1.140.300.90.1.1.20
tuxDmConDmType	.1.3.6.1.4.1.140.300.90.1.1.30
tuxDmConState	.1.3.6.1.4.1.140.300.90.1.1.40
tuxDmConDmCurEncryptBits	.1.3.6.1.4.1.140.300.90.1.1.50

tuxDmConDmlAccessPoint

Syntax	<i>DisplayString (SIZE(1..24))</i>
Access	read-only
Description	The name of the local domain access point that identifies the connection between the domains.

tuxDmConDmrAccessPoint

Syntax	<i>DisplayString (SIZE(1..24))</i>
Access	read-only
Description	The name of the remote domain access point that identifies the connection between the domains

tuxDmConDmType

Syntax	INTEGER { tdomain (1) topend (2) }
Access	read-only
Description	The type of domain—either tdomain or topend.

tuxDmConState

Syntax	INTEGER { active (1) suspended(2) initializing (3) inactive (4) unknown (5)}
Access	read-only
Description	This object denotes the current state of the <code>tuxDmConnection</code> instance. GET requests: active (1): The connection is active. suspended (2): The connection is awaiting retry. initializing (3): The connection is initializing. inactive (4): The specified domain access points are disconnected. (Returned in case of Tuxedo 7.1 and later only.) unknown (5): The state cannot be determined. SET requests: active (1): Connect the specified domain access points. If the current state is suspended or inactive, SET:active places the connection into the state initializing, otherwise, there is no change. inactive (4): Disconnect the specified domain access points and destroy the <code>tuxDmConnection</code> object

tuxDmConDmCurEncryptBits

Syntax	INTEGER { enc-0-bit (1) enc-40-bits (2) enc-56-bits (3) enc-128-bits (4)}
Access	read-only
Description	This attribute is available when <code>tuxDmConDmType=tdomain</code> . The level of encryption in use on this connection: enc-0-bit (1) means no encryption enc-40-bits (2) specifies the encryption length in bits enc-56-bits (3) specifies the encryption length in bits enc-128-bits (4) specifies the encryption length in bits.

tuxDmExportTable

This class represent local resources that are exported to one or more remote domains through a local access point.

Variable Name	Object ID
tuxDmExpDmResourceName	.1.3.6.1.4.1.140.300.100.1.1.10
tuxDmExpDmIAccessPoint	.1.3.6.1.4.1.140.300.100.1.1.20
tuxDmExpDmState	.1.3.6.1.4.1.140.300.100.1.1.30
tuxDmExpDmAclName	.1.3.6.1.4.1.140.300.100.1.1.40
tuxDmExpDmConv	.1.3.6.1.4.1.140.300.100.1.1.50
tuxDmExpDmResourceType	.1.3.6.1.4.1.140.300.100.1.1.60
tuxDmExpDmRemoteName	.1.3.6.1.4.1.140.300.100.1.1.70
tuxDmExpDmInBufType	.1.3.6.1.4.1.140.300.100.1.1.80
tuxDmExpDmOutBufType	.1.3.6.1.4.1.140.300.100.1.1.90
tuxDmExpDmTopendProduct	.1.3.6.1.4.1.140.300.100.1.1.100
tuxDmExpDmTopendFunction	.1.3.6.1.4.1.140.300.100.1.1.110
tuxDmExpDmTopendTarget	.1.3.6.1.4.1.140.300.100.1.1.120
tuxDmExpDmTopendQualifier	.1.3.6.1.4.1.140.300.100.1.1.130
tuxDmExpDmTopendRtqGroup	.1.3.6.1.4.1.140.300.100.1.1.140
tuxDmExpDmTopendRtqName	.1.3.6.1.4.1.140.300.100.1.1.150

tuxDmExpDmResourceName

Syntax *DisplayString (SIZE(1..15))*
Access read-only

Description The local resource name for entries of resource type `service` (the service name), `qspace` (the queue space name), and `qname` (the queue name). For a service entry, the value of this attribute corresponds to the value of an active `tuxTSrvGrp:tuxTsvcName` object. This resource is exported to other domains with the same name or with the alias defined in the `tuxDmExpDmRemoteName` or `tuxDmExpDmTopend*` variables.

tuxDmExpDmIAccessPoint

Syntax *DisplayString (SIZE(1..30))*

Access read-only

Description The local access point name. Setting this attribute to "*" means the resource is available at all local access points.

tuxDmExpState

Syntax `INTEGER { valid (1) | invalid (2) }`

Access read-write

Description This object denotes the current state of the `tuxDmExport` instance.

GET requests:

valid (1): The object exists.

SET requests:

invalid (2): Delete object.

tuxDmExpDmAclName

Syntax *DisplayString (SIZE(1..15))*

Access read-write

Description The name of a `tuxDmAcl` object to use for security on this local service. When access is permitted from topend remote access points, this optional attribute can be specified if `tuxDmExpDmResourceType=service` or `qspace`. This attribute is not permitted if `tuxDmExpDmResourceType=qname`.

tuxDmExpDmConv

Syntax	INTEGER { yes (1) no (2) }
Access	read-only
Description	Specifies whether this local service is conversational or not. When access is permitted from topend remote access points, this attribute must be set to no (2) for entries of tuxDmExpDmResourceType=qspace or qname.

tuxDmExpDmResourceType

Syntax	INTEGER { service (1) qspace (2) qname (3) }
Access	read-only
Description	Specifies whether this entry is for a service, qspace or qname. The default is service.

tuxDmExpDmRemoteName

Syntax	DisplayString (SIZE(1..30))
Access	read-write
Description	For entries of type service or qspace, this attribute specifies the name exported through non-topend remote access points.

tuxDmExpDmInBufType

Syntax	DisplayString (SIZE(1..513))
Access	read-write
Description	<p>Attributes available from remote access points of tuxDmExpDmResourceType=snax ositp topend:</p> <p>type[:subtype] -Input buffer type, optionally followed by subtype.</p> <p>If this attribute is present, it defines the buffer type (and subtype) accepted. This attribute should be defined for entries of tuxDmExpDmResourceType=service when access is permitted from remote access points using ositp with the UDT application context, or when using snax. When access is permitted from topend remote access</p>

points, this optional attribute can be specified if `tuxDmExpDmResourceType=service` and `qname`. This attribute is not permitted if `tuxDmExpDmResourceType=qspace`.

For BEA TopEnd service and queue name entries, the valid values for *type* are: FML32, CARRAY, and X_OCTET.

tuxDmExpDmOutBufType

Syntax `DisplayString (SIZE(1..513))`

Access read-write

Description Attributes available from remote access points of `tuxDmExpDmResourceType=snax|ositp|topend`:

`type[:subtype]` -Output buffer type, optionally followed by subtype.

If this attribute is present, it defines the buffer type (and subtype) output by the service. This attribute should be defined for entries of `tuxDmExpDmResourceType=service` when access is permitted from remote access points using `ositp` with the UDT application context, or when using `snax`. When access is permitted from `topend` remote access points, this optional attribute can be specified if `tuxDmExpDmResourceType=service`. This attribute is not permitted if `tuxDmExpDmResourceType=qspace` and `qname`.

For BEA TopEnd service and queue name entries, the valid values for *type* are FML32, CARRAY, and X_OCTET.

tuxDmExpDmTopendProduct

Syntax `DisplayString (SIZE(1..32))`

Access read-write

Description Attributes available from remote access points of `tuxDMRemoteDmType=topend`:

The BEA TOP END product name. When access is permitted from `topend` remote access points, this attribute must be specified if `tuxDmExpDmResourceType=service`. This attribute is not permitted if `tuxDmExpDmResourceType=qspace` or `qname`.

tuxDmExpDmTopendFunction

Syntax `DisplayString (SIZE(1..8))`

Access read-write

Description Attributes available from remote access points of `tuxDmRemoteDmType=topend`:

The BEA TOP END function name. When access is permitted from topend remote access points, this attribute must be specified if `tuxDmExpDmResourceType=service` or `qname`. This attribute is not permitted if `tuxDmExpDmResourceType=qspace`.

tuxDmExpDmTopendTarget

Syntax `DisplayString (SIZE(1..8))`

Access read-write

Description Attributes available from remote access points of `tuxDmRemoteDmType=topend`:

The BEA TOP END Message Sensitive Routing (MSR) target. This attribute is optional for entries of `tuxDmExpDmResourceType=service`, `qspace`, and `qname` when access is permitted from topend remote access points.

tuxDmExpDmTopendQualifier

Syntax `INTEGER`

Access read-write

Description Attributes available from remote access points of `tuxDmRemoteDmType=topend`:

This attribute is optional for entries of `tuxDmExpDmResourceType=service` or `qname` when access is permitted from topend remote access points. This attribute is not permitted if `tuxDmExpDmResourceType=qspace`.

tuxDmExpDmTopendRtqGroup

Syntax `DisplayString (SIZE(1..32))`

Access read-write

Description Attributes available from remote access points of `tuxDmRemoteDmType=topend`:

The BEA TOP END Recoverable Transaction Queuing (RTQ) queue group name. This attribute must be specified for `tuxDmRemoteDmType=qspace` when access is permitted from `topend` remote access points. This attribute is not permitted if `tuxDmRemoteDmType=service` or `qname`.

tuxDmExpDmTopendRtqName

Syntax `DisplayString (SIZE(1..8))`
Access `read-write`
Description

Attributes available from remote access points of `tuxDmRemoteDmType=topend`:

The BEA TOP End RTQ queue name. This attribute must be specified for `tuxDmExpDmResourceType=qspace` and access is permitted from `topend` remote access points. This attribute is not permitted if `tuxDmExpDmResourceType=service` or `qname`.

tuxDmImportTable

This group represents remote resources that are imported through one or more remote domain access points and made available to the local domain through one or more local domain access points.

Variable Name	Object ID
tuxDmImpDmResourceName	.1.3.6.1.4.1.140.300.110.1.1.10
tuxDmImpDmAccessPointList	.1.3.6.1.4.1.140.300.110.1.1.20
tuxDmImpDmAccessPoint	.1.3.6.1.4.1.140.300.110.1.1.30
tuxDmImpState	.1.3.6.1.4.1.140.300.110.1.1.40
tuxDmImpDmAutoTran	.1.3.6.1.4.1.140.300.110.1.1.50
tuxDmImpDmConv	.1.3.6.1.4.1.140.300.110.1.1.60
tuxDmImpDmLoad	.1.3.6.1.4.1.140.300.110.1.1.70
tuxDmImpDmPrio	.1.3.6.1.4.1.140.300.110.1.1.80
tuxDmImpDmResourceType	.1.3.6.1.4.1.140.300.110.1.1.90
tuxDmImpDmRemoteName	.1.3.6.1.4.1.140.300.110.1.1.100
tuxDmImpDmRoutingName	.1.3.6.1.4.1.140.300.110.1.1.110
tuxDmImpDmTranTime	.1.3.6.1.4.1.140.300.110.1.1.120
tuxDmImpDmInBufType	.1.3.6.1.4.1.140.300.110.1.1.130
tuxDmImpDmOutBufType	.1.3.6.1.4.1.140.300.110.1.1.140
tuxDmImpDmteProduct	.1.3.6.1.4.1.140.300.110.1.1.150
tuxDmImpDmteFunction	.1.3.6.1.4.1.140.300.110.1.1.160
tuxDmImpDmteTarget	.1.3.6.1.4.1.140.300.110.1.1.170
tuxDmImpDmteQualifier	.1.3.6.1.4.1.140.300.110.1.1.180
tuxDmImpDmteRtgGroup	.1.3.6.1.4.1.140.300.110.1.1.190

Variable Name	Object ID
tuxDmImpDmteRtqName	.1.3.6.1.4.1.140.300.110.1.1.200

tuxDmImpDmResourceName

Syntax	DisplayString (SIZE(1..15))
Access	read-only
Description	The remote resource name used for entries of resource type <code>service</code> (the service name, <code>qspace</code> (the queue space name), and <code>qname</code> (the queue name). This resource is imported from remote domains with the same name or with the alias defined in the <code>tuxDmImpDmRemoteName</code> or <code>tuxDmImpDmte*</code> variables.

tuxDmImpDmrAccessPointList

Syntax	DisplayString (SIZE(1..15))
Access	read-only
Description	Identifies the remote domain access point through which this resource should be imported. This is a comma-separated failover domain list; it can contain up to three remote domain access points. If this attribute is set to “*”, the resource can be imported from all remote access points.

tuxDmImpDmlAccessPoint

Syntax	DisplayString (SIZE(1..15))
Access	read-only
Description	The name of the local domain access point through which this imported resource should be made available. If set to the null string, the resource is made available through all local domain access points.

tuxDmImpState

Syntax	INTEGER { valid (1) invalid (2) }
Access	read-write

Description	This object denotes the current state of the tuxDmImport instance. GET requests: valid: The object exists. SET requests: invalid: The object is deleted. A state change is allowed in the active or suspended state and results in the invalid state.
-------------	---

tuxDmImpDmAutoTran

Syntax	INTEGER { yes (1) no (2) }
Access	read-write
Description	When a request is received for a resource that is not already within a transaction, this attribute automatically starts a transaction for the resource. The default is no (2).

tuxDmImpDmConv

Syntax	INTEGER { yes (1) no (2) }
Access	read-write
Description	A boolean value (yes or no) specifies whether the service is conversational. When access is permitted from topend remote access points, this attribute must be set to no (2) for entries of tuxDmImpDmResourceType=qspace and qname.

tuxDmImpDmLoad

Syntax	INTEGER (1..32767)
Access	read-write
Description	The service load.

tuxDmImpDmPrio

Syntax	INTEGER (1..100)
Access	read-write

Description The dequeuing priority. Service requests with a higher priority are serviced first.

tuxDmImpDmResourceType

Syntax `INTEGER { service (1) | qspace (2) | qname (3) }`

Access read-write

Description Specifies whether this entry is for a service, qspace, or qname. The default is `service`.

tuxDmImpDmRemoteName

Syntax `DisplayString (SIZE(1..30))`

Access read-write

Description For entries of type `service` or `qspace`, this attribute specifies the name imported through non-topend remote access points.

tuxDmImpDmRoutingName

Syntax `DisplayString (SIZE(1..15))`

Access read-write

Description The name of a `tuxDmRoutingTable` object to use for routing criteria for this `service` or `qspace`.

tuxDmImpDmTranTime

Syntax `INTEGER (1..32767)`

Access read-write

Description Transaction time value (in seconds) of transactions automatically started for this `service` or `qspace`. Transactions are started automatically when a request not in transaction mode is received and the `tuxDmImpDmAutoTran` attribute is set to `yes`.

Limitation: Runtime updates to this attribute are not reflected in active requests.

tuxDmImpDmInBufType

Syntax `DisplayString (SIZE(0..256))`

Access read-write

Description Attributes available from remote access points of
 `tuxDmRemoteDmType=snax|ositp|topend:`

`type[:subtype]` - Input buffer type, optionally followed by subtype. If this attribute is present, it defines the buffer type (and subtype) accepted. This attribute should be defined for entries of `DMRESOURCETYPE=service` when access is permitted to remote access points that use `ositp` with the UDT application context, or that use `snax`. When access is permitted from `topend` remote access points, this optional attribute can be specified if `tuxDmImpDmResourceType=service` and `qname`. This attribute is not permitted if `tuxDmImpDmResourceType=qspace`. For BEA TOP END service and queue name entries, the valid values for type are: `FML32`, `CARRAY`, AND `X_OCTET`.

tuxDmImpDmOutBufType

Syntax `DisplayString (SIZE(0..256))`

Access read-write

Description Attributes available from remote access points of
 `tuxDmRemoteDmType=snax|ositp|topend:`

tuxDmImpDmteProduct

Syntax `DisplayString (SIZE(1..32))`

Access read-write

Description Attributes available from remote access points of `tuxDmRemoteDmType=topend:`

The BEA TOP END product name. This attribute must be specified if `tuxDmImpDmResourceType=service` or `qname`. It is not permitted if `tuxDmImpDmResourceType=qspace`.

tuxDmImpDmteFunction

Syntax `DisplayString (SIZE(1..8))`

Access	read-write
Description	Attributes available from remote access points of <code>tuxDmRemoteDmType=topend</code> : The BEA TOP END product name. This attribute must be specified if <code>tuxDmImpDmResourceType=service</code> or <code>qname</code> . It is not permitted if <code>tuxDmImpDmResourceType=qspace</code> .

tuxDmImpDmteTarget

Syntax	<code>DisplayString (SIZE(1..8))</code>
Access	read-write
Description	Attributes available from remote access points of <code>tuxDmRemoteDmType=topend</code> : The BEA TOP END Message Sensitive Routing (MSR) target. This attribute is optional for entries of <code>tuxDmImpDmResourceType=service</code> , <code>qspace</code> , and <code>qname</code> .

tuxDmImpDmteQualifier

Syntax	<code>INTEGER</code>
Access	read-write
Description	Attributes available from remote access points of <code>tuxDmRemoteDmType=topend</code> : The BEA TOP END function qualifier. This attribute is optional for entries of <code>tuxDmImpDmResourceType=service</code> or <code>qname</code> . It is not permitted for entries of <code>tuxDmImpDmResourceType=qspace</code> .

tuxDmImpDmteRtqGroup

Syntax	<code>DisplayString (SIZE(1..32))</code>
Access	read-write
Description	Attributes available from remote access points of <code>tuxDmRemoteDmType=topend</code> : The BEA TOP END Recoverable Transaction Queuing (RTQ) queue group name. This attribute must be specified if <code>tuxDmImpDmResourceType=qspace</code> . It is not permitted if <code>tuxDmImpDmResourceType=service</code> or <code>qname</code> .

tuxDmImpDmteRtqName

Syntax DisplayString (SIZE(1..8))

Access read-write

Description Attributes available from remote access points of tuxDmRemoteDmType=topend:

The BEA TOP END RTQ queue name. This attribute mst be specified if
tuxDmImpDmResourceType=qspace. It is not permitted if
tuxDmImpDmResourceType=service or qname.

tuxDmLocalTable

This group defines a local domain access point. A local domain access point is used to control access to local services exported to remote domains and to control access to remote services imported from remote domains.

Variable Name	Object ID
tuxDmLclDmAccessPoint	.1.3.6.1.4.1.140.300.120.1.1.10
tuxDmLclDmAccessPointID	.1.3.6.1.4.1.140.300.120.1.1.20
tuxDmLclDmSrvGroup	.1.3.6.1.4.1.140.300.120.1.1.30
tuxDmLclDmType	.1.3.6.1.4.1.140.300.120.1.1.40
tuxDmLclDmState	.1.3.6.1.4.1.140.300.120.1.1.50
tuxDmLclDmAuditLog	.1.3.6.1.4.1.140.300.120.1.1.60
tuxDmLclDmBlockTime	.1.3.6.1.4.1.140.300.120.1.1.70
tuxDmLclDmMaxRapTran	.1.3.6.1.4.1.140.300.120.1.1.80
tuxDmLclDmMaxTran	.1.3.6.1.4.1.140.300.120.1.1.90
tuxDmLclDmSecurity	.1.3.6.1.4.1.140.300.120.1.1.100
tuxDmLclDmTlogDev	.1.3.6.1.4.1.140.300.120.1.1.110
tuxDmLclDmTlogName	.1.3.6.1.4.1.140.300.120.1.1.120
tuxDmLclDmTlogSize	.1.3.6.1.4.1.140.300.120.1.1.130
tuxDmLclDmConnectionPolicy	.1.3.6.1.4.1.140.300.120.1.1.140
tuxDmLclDmRetryInterval	.1.3.6.1.4.1.140.300.120.1.1.150
tuxDmLclDmMaxRetry	.1.3.6.1.4.1.140.300.120.1.1.160
tuxDmLclDmConnPrincipalName	.1.3.6.1.4.1.140.300.120.1.1.170
tuxDmLclDmMachineType	.1.3.6.1.4.1.140.300.120.1.1.180
tuxDmLclDmBlobShmSize	.1.3.6.1.4.1.140.300.120.1.1.190

tuxDmLclDmAccessPoint

Syntax	<code>DisplayString (SIZE(1..30))</code>
Access	read-only
Description	This is an identifier unique within the scope of <code>tuxDmLocal</code> and <code>tuxDmRemote</code> entry names in the domain configuration.

tuxDmLclDmAccessPoint

Syntax	<code>DisplayString (SIZE(1..30))</code>
Access	read-write
Description	The domain access point identifier. This identifier is unique across all local and remote domain access points.

tuxDmLclDmSrvGroup

Syntax	<code>DisplayString (SIZE(1..30))</code>
Access	read-write
Description	The group in which the administrative servers and gateway processes of the local domain reside.

tuxDmLclDmType

Syntax	<code>INTEGER { tdomain(1) ositp(2) snax(3) topend(4) }</code>
Access	read-write
Description	The type of domain: <code>tdomain</code> for a BEA Tuxedo system domain, <code>ositp</code> for an OSI domain, <code>snax</code> for an SNA domain, or <code>topend</code> for a BEA TOP END domain. The presence or absence of other attributes depends on the value of this attribute.

tuxDmLclState

Syntax	<code>INTEGER { valid(1) invalid(2) }</code>
Access	read-write

Description This object denotes the current state of the tuxDmLocal instance.

GET requests:

valid: The object exists.

SET requests:

invalid: The object is deleted.

tuxDmLclDmAuditLog

Syntax DisplayString (SIZE(1..78))

Access read-write

Description The name of the audit log file for this local domain.

tuxDmLclDmBlockTime

Syntax INTEGER (0..32767)

Access read-write

Description Specifies the maximum wait time allowed for a blocking call. The value sets a multiplier of the SCANUNIT parameters specified in the tuxTdomain group. The value `SCANUNIT * tuxDmLclDmBlockTime` must be greater than or equal to SCANUNIT and less than 32,768 seconds. If this attribute is not specified, the default is set to the value of the `tuxDmLclDmBlockTime` attribute specified in the tuxTdomain object. A timeout always implies a failure of the affected request. Notice that the timeout specified for transactions in the tuxTdomain is always used when the request is issued within a transaction.

tuxDmLclDmMaxRapTran

Syntax INTEGER (0..32767)

Access read-write

Description The maximum number of remote domain access points that can be involved in a single transaction.

tuxDmLclDmMaxTran

Syntax	INTEGER (0..32767)
Access	read-write
Description	The maximum number of simultaneous transactions allowed on this local domain access point. This number must be greater than or equal to the <code>tuxTdomainMaxGTT</code> object in the <code>tuxTdomain</code> group.

tuxDmLclDmSecurity

Syntax	INTEGER { none(1) app-pw(2) dm-pw(3) dm-user-pw(4) te-clear(5) te-safe(6) te-private(7) }
Access	read-write
Description	<p>The type of security enabled on this domain. This attribute must be set to one of the following:</p> <p>none:</p> <p>No security is enabled.</p> <p>app-pw:</p> <p>Valid only when <code>tuxDmRemoteDmType=tdomain</code>. Application password-based security is enabled.</p> <p>dm-pw:</p> <p>Valid only when <code>tuxDmRemoteDmType=tdomain</code>. Domain password-based security is enabled.</p> <p>dm-user-pw:</p> <p>Valid only when <code>tuxDmRemoteDmType=snax</code>. Translation of principal names is enabled.</p> <p>te-clear:</p> <p>Valid only when <code>tuxDmRemoteDmType=topend</code>. BEA TOP END security is enabled. between the local domain and the BEA TOP END system. Network messages are sent in plain text.</p> <p>te-safe:</p>

Valid only when `tuxDmRemoteDmType=topend`. BEA TOP End security is enabled between the local domain and the BEA TOP END system. Network messages are protected by a checksum.

`te-private:`

Valid only when `tuxDmRemoteDmType=topend`. BEA TOP END security is enabled between the local domain and the BEA TOP END system. Network messages are encrypted.

tuxDmLclDmTlogDev

Syntax	<code>DisplayString (SIZE(1..78))</code>
Access	read-write
Description	The device (raw slice) or file that contains the domain TLOG for this local domain access point. The TLOG is stored as a BEA Tuxedo System VTOC table on the device. For reliability, the use of a device (raw slice) is recommended.

tuxDmLclDmTlogName

Syntax	<code>DisplayString (SIZE(1..30))</code>
Access	read-write
Description	The domain TLOG name for this local domain access point. If more than one TLOG exists on the same device, each TLOG must have a unique name.

tuxDmLclDmTlogSize

Syntax	<code>INTEGER</code>
Access	read-write
Description	The size in pages of the TLOG for this local domain access point. This size is constrained by the amount of space available on the device identified in <code>tuxDmLclTlogDev</code> .

tuxDmLclDmConnectionPolicy

Syntax	<code>INTEGER { on-demand(1) on-startup(2) incoming-only(3) }</code>
--------	--

Access	read-write
Description	<p>Attributes available when <code>tuxDmRemoteDmType=tdomain</code> <code>topend</code>.</p> <p>Specifies the conditions under which a local domain gateway tries to establish a connection to a remote domain. Supported values are:</p> <p><code>on-demand</code></p> <p>Means that a connection is attempted only when requested by either a client request to a remote service or an administrative “connect” command. The default setting for this variable is <code>on-demand</code>. The <code>on-demand</code> policy provides the equivalent behavior to previous releases, in which this variable was not explicitly available.</p> <p><code>on-startup</code></p> <p>Means that a domain gateway attempts to establish a connection with its remote domain access points at gateway server initialization time. Remote services, (that is, services advertised by the domain gateway for this local access point) are advertised only if a connection is successfully established to that remote domain access point. Therefore, if there is no active connection to a remote domain access point, the remote services are suspended. By default, this connection policy retries failed connections every 60 seconds; however, you can specify a different value for this interval using the <code>tuxDmLclMaxRetry</code> and <code>tuxDmLclDmRetryInterval</code> attributes.</p>

tuxDmLclDmRetryInterval

Syntax	INTEGER
Access	read-write
Description	<p>The number of seconds between automatic attempts to establish a connection to remote domain access points. The minimum value is 0 and the maximum value is 2147483647. The default setting is 60. If <code>tuxDmLclDmMaxRetry</code> is set to 0, setting <code>tuxDmLclDmRetryInterval</code> is not allowed.</p> <p>This attribute is valid only when the <code>tuxDmLclDmConnectionPolicy</code> attribute is set to <code>on-startup</code>. For other connection policies, automatic retries are disabled.</p>

tuxDmLclDmMaxRetry

Syntax	INTEGER
--------	---------

Access	read-write
Description	<p>The number of times that a domain gateway tries to establish connections to remote domain access points. The minimum value is 0 and the maximum is <code>MAXLONG</code>. <code>MAXLONG</code> indicates that retry processing is repeated indefinitely, or until a connection is established. For a connection policy of <code>on-startup</code>, the default setting for <code>tuxDmLclMaxRetry</code> is <code>MAXLONG</code>. Setting this attribute to 0 turns off the auto retry mechanism. For other connection policies, auto retries are disabled.</p> <p>The <code>tuxDmLclMaxRetry</code> attribute is valid only when the connection policy is <code>on-startup</code>.</p>

tuxDmLclDmConnPrincipalName

Syntax	<code>DisplayString (SIZE(0..511))</code>
Access	read-write
Description	<p>The connection principal name identifier, which is the principal name used for verifying the identity of this local domain access point when establishing a connection to a remote domain access point. This attribute only applies to domains of type <code>TDOMAIN</code> that are running BEA Tuxedo 7.1 or later software.</p> <p>This variable can contain a maximum of 511 characters (excluding the terminating null character). If this attribute is not specified, the connection principal name defaults to the <code>tuxDmLclDmAccessPointId</code> string for this local domain access point.</p> <p>For default authentication plug-ins, if a value is assigned to this variable for this local domain access point, it must be the same as the value assigned to the <code>tuxDmLclAccessPointId</code> attribute for this local domain access point. If these values do not match, the local domain gateway process does not boot and the system generates the following <code>userlog(3c)</code> message: “ERROR: Unable to acquire credentials”.</p>

tuxDmLclDmMachineType

Syntax	<code>DisplayString (SIZE(0..15))</code>
Access	read-write
Description	<p>Used for grouping domains so that encoding/decoding of messages between domains can be bypassed. If it is not specified, the default is to turn encoding/decoding on. If the value set for this field is the same in both the <code>DM_LOCAL</code> and the <code>DM_REMOTE</code> section</p>

of a domain configuration file, data encoding/decoding is bypassed. The value set for this variable can be any string value up to 15 characters in length. It is used only for comparison.

This attribute is valid only when `tuxDmRemoteDmType=tdomain`.

tuxDmLclDmBlobShmSize

Syntax	INTEGER
Access	read-write
Description	This attribute is relevant only to local domain access point entries. It specifies the amount of shared memory allocated to storing binary large object log information specific to <code>ositp</code> or <code>topen</code> .

tuxDmOsiTpTable

This group defines the OSI TP protocol-related configuration information for a specific local or remote domain access point.

Variable Name	Object ID
tuxDmOsiDmAccessPoint	.1.3.6.1.4.1.140.300.130.1.1.10
tuxDmOsiDmState	.1.3.6.1.4.1.140.300.130.1.1.20
tuxDmOsiDmApt	.1.3.6.1.4.1.140.300.130.1.1.30
tuxDmOsiDmAeq	.1.3.6.1.4.1.140.300.130.1.1.40
tuxDmOsiDmNwDevice	.1.3.6.1.4.1.140.300.130.1.1.50
tuxDmOsiDmAcn	.1.3.6.1.4.1.140.300.130.1.1.60
tuxDmOsiDmApid	.1.3.6.1.4.1.140.300.130.1.1.70
tuxDmOsiDmAeid	.1.3.6.1.4.1.140.300.130.1.1.80
tuxDmOsiDmUrch	.1.3.6.1.4.1.140.300.130.1.1.90
tuxDmOsiDmMaxListeningEp	.1.3.6.1.4.1.140.300.130.1.1.100
tuxDmOsiDmXatmiEncoding	.1.3.6.1.4.1.140.300.130.1.1.110

tuxDmOsiDmAccessPoint

Syntax	DisplayString (SIZE(0..30))
Access	read-only
Description	The domain access point name for which this entry provides the protocol-specific configuration information. This field matches the domain access point name given in the tuxDmLocal or tuxDmRemote entry that defines the protocol-independent configuration of the domain access point.

tuxDmOsiDmState

Syntax	INTEGER { valid(1) invalid(2) }
--------	-----------------------------------

Access	read-write
Description	This object denotes the current state of the tuxDmOsitp instance.
	GET requests:
	valid: The object exists.
	SET requests:
	invalid: The object is deleted.

tuxDmOsiDmApt

Syntax	DisplayString (SIZE(1..78))
Access	read-write
Description	The application process title of the domain access point in object identifier form.

tuxDmOsiDmAeq

Syntax	DisplayString (SIZE(1..78))
Access	read-write
Description	The application entity qualifier of the domain access point in integer form.

tuxDmOsiDmNwDevice

Syntax	DisplayString (SIZE(1..78))
Access	read-write
Description	This attribute, which specifies the network device to be used, is relevant when it defines a local domain access point and ignored for a remote domain access point.

tuxDmOsiDmAcn

Syntax	INTEGER { atmi(1) udt(2) }
Access	read-write

Description The application context name to use with this domain access point. When you establish a dialogue to a remote domain access point, use the application context name from the remote domain access point, if it is present. If the application context name from the remote domain access point is absent, use the application context name from the local domain access point. The value `xatmi` selects the use of the X/Open-defined `xatmi` Application Service Element (ASE) and encoding. The value `udt` selects the use of the ISO/IEC 10026-5 User Data Transfer encoding.

tuxDmOsiDmApid

Syntax `INTEGER`

Access `read-write`

Description This optional attribute defines the application process-invocation identifier to be used on this domain access point.

tuxDmOsiDmAeid

Syntax `INTEGER`

Access `read-write`

Description This optional attribute defines the application entity-invocation identifier to be used on this domain access point.

tuxDmOsiDmUrch

Syntax `DisplayString (SIZE(0..30))`

Access `read-write`

Description This attribute specifies the user portion of the OSI TP recovery context handle. It can be required by an OSI TP provider in order to perform recovery of distributed transactions after a communication line or system failure.

This attribute is relevant for defining a local domain access point and ignored for a remote domain access point.

tuxDmOsiDmMaxListeningEp

Syntax `INTEGER (1..32767)`

Access	read-write
Description	This attribute specifies the number of endpoints awaiting incoming OSI TP dialogue. It is relevant for defining a local domain access point and ignored for a remote domain access point.

tuxDmOsiDmXatmiEncoding

Syntax	INTEGER { cae(1) preliminary(2) oltp-tm2200(3) }
Access	read-write
Description	<p>This attribute specifies the version of the XATMI protocol used to communicate with a remote system. Valid values are: cae, preliminary, and oltp-tm2200.</p> <p>This attribute is relevant for remote domain access points and ignored for local domain access points.</p>

tuxDmPasswordTable

This group represents configuration information for inter-domain authentication through access points of type `tdomain`.

Variable Name	Object ID
tuxDmPasswdDmlAccessPoint	.1.3.6.1.4.1.140.300.140.1.1.10
tuxDmPasswdDmrAccessPoint	.1.3.6.1.4.1.140.300.140.1.1.20
tuxDmPasswdDmlPWD	.1.3.6.1.4.1.140.300.140.1.1.30
tuxDmPasswdDmrPWD	.1.3.6.1.4.1.140.300.140.1.1.40
tuxDmPasswdState	.1.3.6.1.4.1.140.300.140.1.1.50

tuxDmPasswdDmlAccessPoint

- Syntax

DisplayString (SIZE(1..24))
- Access

read-only
- Description

The name of the local domain access point to which the password applies.

tuxDmPasswdDmrAccessPoint

- Syntax

DisplayString (SIZE(1..24))
- Access

read-only
- Description

The name of the remote domain access point to which the password applies.

tuxDmPasswdDmlPWD

- Syntax

DisplayString (SIZE(1..30))
- Access

write-only
- Description

The local password used to authenticate connections between the local domain access point identified by `tuxDmPasswdDmlAccessPoint` and the remote domain access point identified by `tuxDmPasswdDmrAccessPoint`.

tuxDmPasswdDmrPWD

Syntax	DisplayString (SIZE(1..30))
Access	write-only
Description	The remote password used to authenticate connections between the local domain access point identified by <code>tuxDmPasswdDmlAccessPoint</code> and the remote domain access point identified by <code>tuxDmPasswdDmrAccessPoint</code> .

tuxDmPasswdState

Syntax	INTEGER { valid(1) invalid(2) recrypt(3) }
Access	read-write
Description	<p>This object denotes the current state of the <code>tuxDmPassword</code> instance.</p> <p>GET requests:</p> <ul style="list-style-type: none"><code>valid</code>: The object exists. <p>SET requests:</p> <ul style="list-style-type: none"><code>invalid</code>: The object is deleted.<code>recrypt</code>: Re-encrypt all passwords using a new encryption key.

tuxDmPrincipalMapTable

This group represents configuration information for mapping principal names to and from external principal names across access point of type snax.

Variable Name	Object ID
tuxDmPrinMapDmlAccessPoint	.1.3.6.1.4.1.140.300.150.1.1.10
tuxDmPrinMapDmrAccessPoint	.1.3.6.1.4.1.140.300.150.1.1.20
tuxDmPrinMapDmlPrinName	.1.3.6.1.4.1.140.300.150.1.1.30
tuxDmPrinMapDmrPrinName	.1.3.6.1.4.1.140.300.150.1.1.40
tuxDmPrinMapDirection	.1.3.6.1.4.1.140.300.150.1.1.50
tuxDmPrinMapState	.1.3.6.1.4.1.140.300.150.1.1.60

tuxDmPrinMapDmlAccessPoint

- Syntax

DisplayString (SIZE(1..12))
- Access

read-only
- Description

The local domain access point to which the principal mapping applies.

tuxDmPrinMapDmrAccessPoint

- Syntax

DisplayString (SIZE(1..12))
- Access

read-only
- Description

The remote domain access point to which the principal mapping applies.

tuxDmPrinMapDmlPrinName

- Syntax

DisplayString (SIZE(1..12))
- Access

read-only
- Description

The local principal name in the principal mapping.

tuxDmPrinMapDmrPrinName

Syntax	<code>DisplayString (SIZE(1..12))</code>
Access	read-only
Description	The remote principal name in the principal mapping.

tuxDmPrinMapDirection

Syntax	<code>INTEGER { in(1) out(2) both(3) }</code>
Access	read-write
Description	<p>The direction to which the principal mapping applies.</p> <p><code>in:</code></p> <p>Is incoming to this BEA Tuxedo domain through the given remote domain access point and local domain access point.</p> <p><code>out:</code></p> <p>Is outgoing from this BEA Tuxedo domain through the given local domain access point and remote domain access point.</p> <p><code>both:</code></p> <p>Applies to both incoming to and outgoing from this BEA Tuxedo domain through the given local domain access point and remote domain access point.</p>

tuxDmPrinMapState

Syntax	<code>INTEGER { valid(1) invalid(2) }</code>
Access	read-write
Description	<p>This object denotes the current state of the tuxDmPrincipalMap instance.</p> <p>GET requests:</p> <p><code>valid:</code> The object exists.</p> <p>SET requests:</p> <p><code>invalid:</code> The object is deleted.</p>

tuxDmRemoteTable

The tuxDmRemote class represents remote domain access point configuration information. Local resources that can be exported through one or more local domain access points are made accessible to a remote domain through a remote domain access point. Similarly, remote resources are imported from a remote domain through a remote domain access point.

Variable Name	Object ID
tuxDmRemoteDmAccessPoint	.1.3.6.1.4.1.140.300.160.1.1.10
tuxDmRemoteDmAccessPointID	.1.3.6.1.4.1.140.300.160.1.1.20
tuxDmRemoteType	.1.3.6.1.4.1.140.300.160.1.1.30
tuxDmRemoteState	.1.3.6.1.4.1.140.300.160.1.1.40
tuxDmRemoteDmCodePage	.1.3.6.1.4.1.140.300.160.1.1.50
tuxDmRemoteDmMachineType	.1.3.6.1.4.1.140.300.160.1.1.90

tuxDmRemoteDmAccessPoint

Syntax	DisplayString (SIZE(1..30))
Access	read-only
Description	The name of this tuxDmRemote entry. This is an identifier unique within the scope of tuxDmLocal and tuxDmRemote entry names in the domain configuration.

tuxDmRemoteDmAccessPointID

Syntax	DisplayString (SIZE(1..30))
Access	read-write
Description	The access point identifier. This identifier is unique across all local and remote domain access points.

tuxDmRemoteType

Syntax	INTEGER { tdomain(1) ositp(2) snax(3) topend(4) }
Access	read-write
Description	<p>The type of domain:</p> <p>tdomain:</p> <p>A BEA Tuxedo system domain.</p> <p>ositp:</p> <p>An OSI domain.</p> <p>snax:</p> <p>An SNA domain.</p> <p>topen:</p> <p>A BEA TOP END domain.</p> <p>The presence or absence of other attributes depends on the value of this attribute.</p>

tuxDmRemoteState

Syntax	INTEGER { valid(1) invalid(2) }
Access	read-write
Description	<p>This object denotes the current state of the tuxDmRemote instance.</p> <p>GET requests:</p> <p>valid: The object exists.</p> <p>SET requests:</p> <p>invalid: The object is deleted.</p>

tuxDmRemoteDmCodePage

Syntax	DisplayString (SIZE(1..20))
Access	read-write

Description Attributes available when `tuxDmRemoteDmType=snax`. The name of the default translation tables to use when translating requests and replies that are sent through this access point.

tuxDmRemoteDmMachineType

Syntax `DisplayString (SIZE(0..15))`

Access read-write

Description Attributes available when `tuxDmRemoteDmType=tdomain`.

These attributes are used for grouping domains so that encoding/decoding of messages between domains can be bypassed. If it is not specified, the default is to turn encoding/decoding on. If the value set for this field is the same in both the `DM_LOCAL` and the `DM_REMOTE` sections of a domain configuration file, data encoding/decoding is bypassed. The value set for this variable can be any string value up to 15 characters in length. It is used only for comparison.

tuxDmResourcesTable

This group represents Domains-specific configuration information.

Variable Name	Object ID
tuxDmResourcesDmVersion	.1.3.6.1.4.1.140.300.170.1.1.10

tuxDmResourcesDmVersion

Syntax	DisplayString (SIZE(1..30))
Access	read-only
Description	A user-supplied identifier for the Domains configuration.

tuxDmRoutingTable

The tuxDmRoutingTable class represents routing criteria information for routing requests to a domain through a remote domain access point.

Variable Name	Object ID
tuxDmRoutingDmRoutingName	.1.3.6.1.4.1.140.300.180.1.1.10
tuxDmRoutingDmBufType	.1.3.6.1.4.1.140.300.180.1.1.20
tuxDmRoutingDmField	.1.3.6.1.4.1.140.300.180.1.1.30
tuxDmRoutingDmFieldType	.1.3.6.1.4.1.140.300.180.1.1.40
tuxDmRoutingDmRanges	.1.3.6.1.4.1.140.300.180.1.1.50
tuxDmRoutingState	.1.3.6.1.4.1.140.300.180.1.1.60

tuxDmRoutingDmRoutingName

- Syntax

DisplayString (SIZE(1..15))
- Access

read-only
- Description

The name of the routing criteria table entry.

tuxDmRoutingDmBufType

- Syntax

DisplayString (SIZE(1..30))
- Access

read-only
- Description

List of types and subtypes of data buffers for which this routing entry is valid.

type1[:subtype1[, subtype2...]][:type2[:subtype3[, subtype4...]]...]

A maximum of 32 type/subtype combinations is allowed. The types are restricted to the following: FML, XML, VIEW, X-C-TYPE, or X_COMMON. No subtype can be specified for FML or XML; subtypes are required for types VIEW, X_C_TYPE, and X_COMMON (“*” is not allowed).

Note that subtype names should not contain semicolon (;), colon (:), comma (,), or asterisk (*) characters. Duplicate type/subtype pairs cannot be specified for the same routing criterion name; more than one routing entry can have the same criterion name as long as the type/subtype pairs are unique. If multiple buffer types are specified for a single routing entry, the data types of the routing field for each buffer type must be the same.

tuxDmRoutingDmField

Syntax	DisplayString (SIZE(1..30))
Access	read-write
Description	This attribute is the routing field name. This field is assumed to be an FML buffer, XML buffer, or VIEW field name that is identified in an FML field table (using the FLDTBLDIR and FIELDTBLS environment variables), or an FML VIEW table (using the VIEWDIR and VIEWFILES environment variables), respectively. This information is used to get the associated field value for data-dependent routing to an access point of a remote domain.

For an XML buffer type, this field contains either a routing element type (or name) or a routing element attribute name.

The syntax of this variable for an XML buffer type is as follows:

```
root_element[/child_element][/child_element][...][/@attribute_\nname]
```

The element is assumed to be an XML document or datagram element type. Indexing is not supported. Therefore, the BEA Tuxedo system recognizes only the first occurrence of a given element type when it processes an XML buffer for data-dependent routing. This information is used to get the associated element content for data-dependent routing while sending a message. The content must be a string encoded in UTF-8.

The attribute is assumed to be an XML document or datagram attribute of the defined element. This information is used to get the associated attribute value for data-dependent routing while sending a message. The value must be a string encoded in UTF-8.

The combination of element name and attribute name can contain up to 30 characters.

tuxDmRoutingDmFieldType

Syntax	INTEGER { char(1) short(2) long(3) float(4) double(5) string (6) }
Access	read-write
Description	The type can be char, short, long, float, double, or string. Only one type is allowed. This attribute is used only for routing XML buffers.

tuxDmRoutingDmRanges

Syntax	DisplayString (SIZE(1..1000))
Access	read-write
Description	This attribute includes the ranges and associated server groups for the tuxDmRoutingFieldType routing field. The format of the string is a comma-separated, ordered list of range/group name pairs. A range/group pair has the following format:

lower[-upper]:raccesspoint

where `lower` and `upper` are assigned numeric values or character strings in single quotes. `lower` must be less than or equal to `upper`. To embed a single quote in a character string value, the quote must be preceded by two backslashes (for example, `'O\\'Brien'`). The value `MIN` can be used to indicate the minimum value for the data type of the associated field on the machine. The value `MAX` can be used to indicate the maximum value for the data type of the associated field on the machine. Thus, `"MIN-5"` is all numbers less than or equal to -5, and `"6-MAX"` is all numbers greater than or equal to 6.

The meta-character `"*"` (wild-card) in the position of a range indicates any values not covered by the other ranges previously seen in the entry. Only one wild-card range is allowed per entry and it should be the last range (ranges that follow it are ignored).

The routing field can be of any data type supported in FML. A numeric routing field must have numeric range values and a string routing field must have string range values.

String range values for `string`, `carray`, and `character` field types must be placed inside a pair of single quotes, and cannot be preceded by a sign. The `short` and `long` integer values are a string of digits, optionally preceded by a plus or minus sign.

Floating point numbers are of the form accepted by the C compiler or `atof(3)`: an optional sign, then a string of digits (that optionally contains a decimal point), then an optional `e` or `E` followed by an optional sign or space, followed by an integer.

The `raccesspoint` parameter indicates the remote domain access point to which the request is routed if the field matches the range. A `raccesspoint` of “*” indicates that the request can go to any remote domain access point that imports the desired service.

tuxDmRoutingState

Syntax	INTEGER { valid(1) invalid(2) }
Access	read-write
Description	This object denotes the current state of the tuxDmRouting instance. GET requests: valid: The object exists. SET requests: invalid: The object is deleted.

tuxDmrPrincipalTable

This group represents password configuration information for remote principal names.

Variable Name	Object ID
tuxDmrPrincipalDmrAccessPoint	.1.3.6.1.4.1.140.300.190.1.1.10
tuxDmrPrincipalDmrPrinName	.1.3.6.1.4.1.140.300.190.1.1.20
tuxDmrPrincipalDmrPrinPasswd	.1.3.6.1.4.1.140.300.190.1.1.30
tuxDmrPrincipalState	.1.3.6.1.4.1.140.300.190.1.1.40

tuxDmrPrincipalDmrAccessPoint

- Syntax

DisplayString (SIZE(1..24))
- Access

read-only
- Description

The remote domain access point to which the principal is applicable.

tuxDmrPrincipalDmrPrinName

- Syntax

DisplayString (SIZE(1..24))
- Access

read-only
- Description

The remote principal name.

tuxDmrPrincipalDmrPrinPasswd

- Syntax

DisplayString (SIZE(0..8))
- Access

write-only
- Description

The remote password used for the principal name when communicating through the remote domain access point identified in tuxDmrPrincipalDmrAccessPoint.

tuxDmrPrincipalState

Syntax `INTEGER { valid(1) | invalid(2) }`

Access `read-write`

Description This object denotes the current state of the tuxDmrPrincipal instance.

GET requests:

`valid`: The object exists.

SET requests:

`invalid`: The object is deleted.

tuxDmSnaCRMTable

This group defines the SNM-CRM-specific configuration information for the named local domain access point.

Variable Name	Object ID
tuxDmSnaCRMDmSNACRM	.1.3.6.1.4.1.140.300.200.1.1.10
tuxDmSnaCRMDmAccessPoint	.1.3.6.1.4.1.140.300.200.1.1.20
tuxDmSnaCRMState	.1.3.6.1.4.1.140.300.200.1.1.30
tuxDmSnaCRMDmNWAddr	.1.3.6.1.4.1.140.300.200.1.1.40
tuxDmSnaCRMDmNWDevice	.1.3.6.1.4.1.140.300.200.1.1.50

tuxDMSnaCRMDmSNACRM

Syntax	DisplayString (SIZE(1..30))
Access	read-only
Description	This is an identifier, unique within the scope of the SNA CRM entries in the domain configuration, used to identify this SNA CRM entry.

tuxDMSnaCRMDmAccessPoint

Syntax	DisplayString (SIZE(1..30))
Access	write-only
Description	The name of the local domain access point entry with which this SNA CRM is used.

tuxDMSnaCRMState

Syntax	INTEGER { valid(1) invalid(2)}
Access	read-write
Description	This object denotes the current state of the tuxDmSnaCRM instance.

GET requests:

valid: The object exists.

SET requests:

invalid: The object is deleted.

tuxDMSnaCRMDmNWAddr

Syntax `DisplayString (SIZE(1..78))`

Access read-write

Description Specifies the network address for communication between the domain gateway of the local domain access point and the SNA CRM.

tuxDMSnaCRMDmNWDevice

Syntax `DisplayString (SIZE(1..78))`

Access read-write

Description Specifies the network device to be used for communication between the domain gateway of the local domain access point and the SNA CRM.

tuxDmSnaLinkTable

This group represents snax-specific configuration information for a remote domain access point.

Variable Name	Object ID
tuxDmSnaLinkDmSNALink	.1.3.6.1.4.1.140.300.210.1.1.10
tuxDmSnaLinkDmSNAStack	.1.3.6.1.4.1.140.300.210.1.1.20
tuxDmSnaLinkDmrAccessPoint	.1.3.6.1.4.1.140.300.210.1.1.30
tuxDmSnaLinkDmlSysID	.1.3.6.1.4.1.140.300.210.1.1.40
tuxDmSnaLinkDmrSysID	.1.3.6.1.4.1.140.300.210.1.1.50
tuxDmSnaLinkDmlUname	.1.3.6.1.4.1.140.300.210.1.1.60
tuxDmSnaLinkDmMinWin	.1.3.6.1.4.1.140.300.210.1.1.70
tuxDmSnaLinkDmModeName	.1.3.6.1.4.1.140.300.210.1.1.80
tuxDmSnaLinkState	.1.3.6.1.4.1.140.300.210.1.1.90
tuxDmSnaLinkDmSecType	.1.3.6.1.4.1.140.300.210.1.1.100
tuxDmSnaLinkDmStartType	.1.3.6.1.4.1.140.300.210.1.1.110
tuxDmSnaLinkDmMaxSNAsess	.1.3.6.1.4.1.140.300.210.1.1.120
tuxDmSnaLinkDmMaxSyncLvl	.1.3.6.1.4.1.140.300.210.1.1.130

tuxDmSnaLinkDmSNALink

Syntax	DisplayString (SIZE(1..30))
Access	read-only
Description	This is an identifier, unique within the scope of the SNA LINK entries within the domain configuration, used to identify rows in this table.

tuxDmSnaLinkDmSNASStack

Syntax	DisplayString (SIZE(1..30))
Access	read-write
Description	The name of the <code>snax</code> stack entry to be used to reach this remote domain access point.

tuxDmSnaLinkDmrAccessPoint

Syntax	DisplayString (SIZE(1..30))
Access	read-write
Description	Identifies the remote domain access point name for which this entry provides the <code>snax</code> configuration data.

tuxDmSnaLinkDmISysID

Syntax	DisplayString (SIZE(1..4))
Access	read-write
Description	The local SYSID used when establishing an SNA link to the remote logical unit (LU).

tuxDmSnaLinkDmrSysID

Syntax	DisplayString (SIZE(1..4))
Access	read-write
Description	The remote SYSID used when establishing an SNA link to the remote logical unit (LU).

tuxDmSnaLinkDmIUname

Syntax	DisplayString (SIZE(1..8))
Access	read-write
Description	Specifies the logical unit (LU) name associated with the remote domain access point.

tuxDmSnaLinkDmMinWin

Syntax	INTEGER
Access	read-write
Description	The minimum number of winner sessions to the remote LU.

tuxDmSnaLinkDmModeName

Syntax	DisplayString (SIZE(1..8))
Access	read-write
Description	Specifies the name associated with the session characteristics for sessions to the remote LU.

tuxDmSnaLinkState

Syntax	INTEGER { valid(1) invalid(2) }
Access	read-write
Description	This object denotes the current state of the tuxDmSnaLink instance. GET requests: valid: The object exists. SET requests: invalid: The object is deleted.

tuxDmSnaLinkDmSecType

Syntax	INTEGER { local(1) identify(2) verify(3) persistent(4) mixidpe(5) }
Access	read-write
Description	Specifies the type of SNA security to be used on sessions to the remote logical unit. Valid values for this attribute are local, identify, verify, persistent, and mixidpe.

tuxDmSnaLinkDmStartType

Syntax	INTEGER { auto(1) cold(2) }
Access	read-write
Description	Specifies the type of session start-up for the destination logical unit (LU). cold Forces a COLDSTART with the LU. auto The SNACRM, in conjunction with the domain gateway, chooses whether to COLDSTART or WARMSTART the LU.

tuxDmSnaLinkDmSNAsess

Syntax	INTEGER (0 . . 32767)
Access	read-write
Description	Specifies maximum number of sessions to establish with the remote LU.

tuxDmSnaLinkDmMaxSyncLvl

Syntax	INTEGER (0 . . 2)
Access	read-only
Description	The maximum SYNC LEVEL that can be support to this remote LU.

tuxDmSnaStackTable

This group defines an SNA stack to be used by a specific SNA CRM.

Variable Name	Object ID
tuxDmSnaStackDmSnaStack	.1.3.6.1.4.1.140.300.220.1.1.10
tuxDmSnaStackDmSnaCRM	.1.3.6.1.4.1.140.300.220.1.1.20
tuxDmSnaStackDmStackType	.1.3.6.1.4.1.140.300.220.1.1.30
tuxDmSnaStackDmLUname	.1.3.6.1.4.1.140.300.220.1.1.40
tuxDmSnaStackDmTpName	.1.3.6.1.4.1.140.300.220.1.1.50
tuxDmSnaStackDmStackParams	.1.3.6.1.4.1.140.300.220.1.1.60
tuxDmSnaStackState	.1.3.6.1.4.1.140.300.220.1.1.70

tuxDmSnaStackDmSnaStack

Syntax	DisplayString (SIZE(1..30))
Access	read-only
Description	The name of this tuxDmSnaStack entry. This is an identifier, unique within the scope of the tuxDmSnaStack table in the domain configuration.

tuxDmSnaStackDmSnaCRM

Syntax	DisplayString (SIZE(1..30))
Access	read-write
Description	Identifies the tuxDmSnaCRM table entry of the SNA CRM in which this SNA protocol stack definition is used.

tuxDmSnaStackDmStackType

Syntax	DisplayString (SIZE(1..30))
--------	-----------------------------

Access	read-write
Description	Identifies the protocol stack to be used.

tuxDmSnaStackDmIUname

Syntax	DisplayString (SIZE(1..8))
Access	read-write
Description	Specifies the LU name to be used on sessions established using this stack definition.

tuxDmSnaStackDmTpName

Syntax	DisplayString (SIZE(1..8))
Access	read-write
Description	Specifies the TP name associated with the SNA stack. A value of "*" means accept any TP name.

tuxDmSnaStackDmStackParams

Syntax	DisplayString (SIZE(1..128))
Access	read-write
Description	Provides protocol stack specific parameters.

tuxDmSnaStackState

Syntax	INTEGER { valid(1) invalid(2) }
Access	read-write
Description	This object denotes the current state of the tuxDmSnaStack instance. GET requests: valid: The object exists. SET requests: invalid: The object is deleted.

tuxDmTdomainTable

This group defines the domain-specific configuration for a local or remote domain access point.

Variable Name	Object ID
tuxDmTdomainDmAccessPoint	.1.3.6.1.4.1.140.300.240.1.1.10
tuxDmTdomainDmNwAddr	.1.3.6.1.4.1.140.300.240.1.1.20
tuxDmTdomainState	.1.3.6.1.4.1.140.300.240.1.1.30
tuxDmTdomainDmNwDevice	.1.3.6.1.4.1.140.300.240.1.1.40
tuxDmTdomainDmCmpLimit	.1.3.6.1.4.1.140.300.240.1.1.50
tuxDmTdomainDmFailOverSeq	.1.3.6.1.4.1.140.300.240.1.1.60
tuxDmTdomainDmMinEncryptBits	.1.3.6.1.4.1.140.300.240.1.1.70
tuxDmTdomainDmMaxEncryptBits	.1.3.6.1.4.1.140.300.240.1.1.80

tuxDmTdomainDmAccessPoint

Syntax	<i>DisplayString (SIZE(1..24))</i>
Access	read-only
Description	The local or remote domain access point name for which this entry provides the TDomain-specific configuration data. When domain-level failover is in use, more than one tuxDmTdomain Table entry can be defined with the same tuxTDmTdomainDmAccessPoint.

tuxDmTdomainDmNwAddr

Syntax	<i>DisplayString (SIZE(1..24))</i>
Access	read-only
Description	Specifies the network address associated with the access point.

For a local domain access point, this attribute supplies the address used to listen for incoming connections.

For a remote domain access point, this attribute supplies the destination used when you connect to a remote domain access point.

The value of this field must be unique across all tuxDmTdomainTable entries.

tuxDmTdomainState

Syntax	INTEGER { valid(1) invalid(2) }
Access	read-write
Description	This object denotes the current state of the tuxDmTdomain instance. GET requests: valid: The object exists. SET requests: invalid: Delete the object.

tuxDmTdomainDmNwDevice

Syntax	DisplayString (SIZE(1..78))
Access	read-write
Description	Specifies the network device used. For a local domain access point, this attribute specifies the device used for listening. For a remote domain access point, this attribute specifies the device used to connect to the remote domain access point.

tuxDmTdomainDmCmplLimit

Syntax	INTEGER
Access	read-write
Description	This attribute is relevant to remote domain access points only. It specifies the threshold over which compression occurs for traffic on connections to this access point.

tuxDmTdomainDmFailOverSeq

Syntax	INTEGER (0..32767)
Access	read-write
Description	This attribute is relevant to remote domain access points only. It specifies the position of this set of addressing in the failover sequence for this remote domain access point. If no failover sequence number is supplied, the first entry for this remote domain access point is allocated the number 10 greater than the highest failover sequence number know for the remote domain access point. Thus, the first entry gets 10, the second, 20, and so on.

tuxDmTdomainDmMinEncriptBits

Syntax	INTEGER { enc-0-bit(1) enc-40-bits(2) enc-56-bits (3) enc-128-bits (4) }
Access	read-write
Description	<p>This attribute is relevant to remote domain access points only. When establishing a network link to this access point, this attribute specifies the minimum level of encryption required.</p> <p>enc-0-bit (1): No encryption</p> <p>enc-40-bits (2), enc-56-bits (3), and enc-128-bits (4): These specify the encryption length (in bits).</p> <p>If this minimum level of encryption is not met, link establishment fails. The default value is enc-0-bit.</p>

tuxDmTdomainDmMaxEncriptBits

Syntax	INTEGER { enc-0-bit(1) enc-40-bits(2) enc-56-bits (3) enc-128-bits (4) }
Access	read-write
Description	<p>This attribute is relevant to remote domain access points only. When establishing a network link to this access point, this attribute specifies the maximum level of encryption required.</p> <p>enc-0-bit (1): No encryption</p>

enc-40-bits (2), enc-56-bits (3), and enc-128-bits (4): These specify the encryption length (in bits).

The default value is enc-128-bits.

Note: Modifications to this attribute do not affect established connections.

tuxDmTopendTable

This group defines the configuration for a local or remote domain access point specific to a BEA TOP END system.

Variable Name	Object ID
tuxDmTopendDmAccessPoint	.1.3.6.1.4.1.140.300.250.1.1.10
tuxDmTopendDmNwAddr	.1.3.6.1.4.1.140.300.250.1.1.20
tuxDmTopendDmteTpSystem	.1.3.6.1.4.1.140.300.250.1.1.30
tuxDmTopendState	.1.3.6.1.4.1.140.300.250.1.1.40
tuxDmTopendDmNwDevice	.1.3.6.1.4.1.140.300.250.1.1.50
tuxDmTopendDmtePwd	.1.3.6.1.4.1.140.300.250.1.1.60
tuxDmTopendDmFailoverSeq	.1.3.6.1.4.1.140.300.250.1.1.70

tuxDmTopendDmAccessPoint

- Syntax

DisplayString (SIZE(1..24))
- Access

read-only
- Description

Specifies the local or remote domain access point name for which this entry provides the BEA TOP END system-specific configuration data.

tuxDmTopendDmNwAddr

- Syntax

DisplayString (SIZE(1..24))
- Access

read-only
- Description

Specifies the network address associated with the local or remote domain access point.

tuxDmTopendDmteTpSystem

- Syntax

DisplayString (SIZE(1..8))

Access	read-write
Description	Specifies the name of the BEA TOP END system.

Note: All remote domain access points accessible through a local domain access point must have the same BEA TOP END system name.

tuxDmTopendState

Syntax	INTEGER { valid(1) invalid(2) reencrypt(3) }
Access	read-write
Description	This object denotes the current state of the tuxDmTopend instance.

GET requests:

valid: The object exists.

SET requests:

invalid: Delete the object.

reencrypt: Re-encrypt all passwords that use a new encryption key.

tuxDmTopendDmNwDevice

Syntax	<i>DisplayString (SIZE(1..78))</i>
Access	read-write
Description	Specifies the network device associated with the local or remote domain access point.

tuxDmTopendDmtePwd

Syntax	<i>DisplayString (SIZE(1..12))</i>
Access	read-write
Description	Specifies the password to use when sending messages to the BEA TOP END system. This attribute is relevant only to local domain access point entries.

tuxDmTopendDmFailoverSeq

Syntax INTEGER (0..32767)

Access read-write

Description This attribute is relevant to remote domain access points only. It specifies the position of this set of addressing in the failover sequence for this remote domain access point. If no failover sequence number is supplied, the first entry for this remote domain access point is allocated the number 10 greater than the highest failover sequence number known for the remote domain access point. Thus, the first entry gets 10, the second, 20, and so on.

The domain gateway uses the tuxDmTopend addressing entries for a particular remote domain access point strictly in the order of its failover sequence numbers—lowest to highest.

tuxDmTransactionTable

This group represents information about transactions that span domains. This object can be used to find out what remote domain access points are involved in the transaction, the parent domain access point, the transaction state, and other information.

Variable Name	Object ID
tuxDmTransactionDmlAccessPoint	.1.3.6.1.4.1.140.300.260.1.1.10
tuxDmTransactionDmTpTranID	.1.3.6.1.4.1.140.300.260.1.1.20
tuxDmTransactionState	.1.3.6.1.4.1.140.300.260.1.1.30
tuxDmTransactionDmTxAccessPoint	.1.3.6.1.4.1.140.300.260.1.1.40
tuxDmTransactionDmTxNetTranID	.1.3.6.1.4.1.140.300.260.1.1.50
tuxDmTransactionDmBranchCount	.1.3.6.1.4.1.140.300.260.1.1.60
tuxDmTransactionDmBranchIndex	.1.3.6.1.4.1.140.300.260.1.1.70
tuxDmTransactionDmBranchNo	.1.3.6.1.4.1.140.300.260.1.1.80
tuxDmTransactionDmrAccessPoint	.1.3.6.1.4.1.140.300.260.1.1.90
tuxDmTransactionDmNetTranID	.1.3.6.1.4.1.140.300.260.1.1.100
tuxDmTransactionDmBranchState	.1.3.6.1.4.1.140.300.260.1.1.110

tuxDmTransactionDmlAccessPoint

Syntax	<i>DisplayString (SIZE(1..30))</i>
Access	read-only
Description	The name of the local domain access point with which the transaction is associated. This is a required field for GET operations. For SET operations, this variable must be specified.

tuxDmTransactionDmTPTranID

Syntax	<i>DisplayString (SIZE(1..24))</i>
Access	read-write
Description	The transaction identifier returned from <code>tpsuspend(3c)</code> mapped to a string representation. The data in this field should not be interpreted directly by the user, except for equality comparison. For SET operations, this variable must be specified.

tuxDmTransactionState

Syntax	<code>INTEGER { aborted(1) abortonly(2) active(3) comcalled(4) decided(5) done(6) habort(7) hcommit(8) heuristic(9) ready(10) unknown(11) invalid(12) }</code>
Access	read-write
Description	<p>This object denotes the current state of the <code>tuxDmTransaction</code> instance.</p> <p>GET requests:</p> <p><code>aborted(1)</code>: The transaction is being rolled back.</p> <p><code>abortonly(2)</code>: The transaction has been identified for rollback.</p> <p><code>active(3)</code>: The transaction is active.</p> <p><code>comcalled(4)</code>: The transaction has initiated the first phase of commitment.</p> <p><code>decided(5)</code>: The transaction has initiated the second phase of commitment.</p> <p><code>done(6)</code>: The transaction has completed the second phase of commitment.</p> <p><code>habort(7)</code>: The transaction has been heuristically rolled back.</p> <p><code>hcommit(8)</code>: The transaction has been heuristically committed.</p> <p><code>heuristic(9)</code>: The transaction commitment or rollback has completed heuristically.</p> <p><code>ready(10)</code>: The transaction has completed the first phase of a two-phase commit. All the participating groups and remote domains have completed the first phase of commitment and are ready to be committed.</p> <p><code>unknown(11)</code>: It was not possible to determine the state of the transaction.</p>

SET requests:

`invalid`: Forget the specified table entry. This state change is only valid in states `HCommit` and `HABort`.

tuxDmTransactionDmTxAccessPoint

Syntax *DisplayString (SIZE(1..30))*

Access read-only

Description If the transaction originated from a remote domain, this is the name of the remote domain access point through which it originated. If the transaction originated within this domain, this is the name of the local domain access point.

tuxDmTransactionDmTxNetTranID

Syntax *DisplayString (SIZE(1..78))*

Access read-only

Description If the transaction originated from a remote domain, this is the external transaction identifier received from the remote domain access point through which the transaction originated. If the transaction originated within this domain, it contains the same value as the

tuxDmTransactionDmBranchCount

Syntax INTEGER

Access read-only

Description The number of branches to remote domain access points involved in the transaction. For a domain gateway that does not make branch information available, this value is zero.

tuxDmTransactionDmBranchIndex

Syntax INTEGER

Access read-only

Description The index of the first branch-specific attribute values (tuxDmTransactionDmBranchNo, tuxDmTransactionDmrAccessPoint, tuxDmTransactionDmNetTranID, and tuxDmTransactionDmBranchState) corresponding to this object.

tuxDmTransactionDmBranchNo

Syntax INTEGER

Access read-only

Description The branch number of the participating branch (numbered from zero).

tuxDmTransactionDmrAccessPoint

Syntax *DisplayString (SIZE(1..30))*

Access read-only

Description The name of the remote domain access point for this branch.

tuxDmTransactionDmNetTranID

Syntax *DisplayString (SIZE(1..78))*

Access read-only

Description The external transaction identifier used with the remote domain access point for this branch. Some types of domain gateways do not return this information; in this scenario, this attribute is set to the empty string. For example, TDomains uses the local transaction identifier in tuxDmTransactionDmTpTranID for branches to remote domain access points and sets this value to the empty string.

tuxDmTransactionDmBranchState

Syntax INTEGER { aborted(1) | abortonly(2) | active(3) | comcalled(4) | decided(5) | done(6) | habort(7) | hcommit(8) | heuristic-hazard(9) | heuristic-mixed(10 | ready(11) | unknown(12) }

Access read-write

Description A GET operation retrieves runtime information for the transaction branch (when the information is available for a particular domain gateway type).

GET requests:

ABorted(1): The transaction branch is being rolled back.

ABortonly(2): The transaction branch has been identified for rollback.

ACTIVE(3): The transaction branch is active.

COMcalled(4): The transaction branch has initiated the first phase of commitment.

DECided(5): The transaction branch has initiated second phase of commitment.

DONE(6): The transaction branch has completed the second phase of commitment.

HABort(7): The transaction has been heuristically rolled back.

HCOMmit(8): The transaction has been heuristically committed.

Heuristic HAZard(9): Communications for the transaction branch failed and it has not been determined if rollback completed successfully.

Heuristic MIXed(10): The transaction commitment or rollback for the transaction branch has completed and the remote domain has reported that the state of some of the resources used for the commitment or rollback is not consistent with the outcome of the transaction.

READY(11): The transaction has completed the first phase of a two-phase commit. All the participating groups and remote domains have completed the first phase of commitment and are ready to be committed.

UNKNOWN(12): The state of the transaction could not be determined.

4 BEA Domain List MIB

The MIB group `beaDomainList` represents information about the Tuxedo or WLE domain the agent is monitoring, as specified at startup.

Note: Row creation is not allowed in this MIB group. A minimal `tuxconfig` file must exist before you can start the agent.

Variable Name	Object ID
<code>beaDomainKey</code>	<code>.1.3.6.1.4.1.140.305.1.1</code>
<code>beaLogicalAgentName</code>	<code>.1.3.6.1.4.1.140.305.1.2</code>
<code>beaDomainID</code>	<code>.1.3.6.1.4.1.140.305.1.3</code>
<code>beaDomainTuxdir</code>	<code>.1.3.6.1.4.1.140.305.1.4</code>
<code>beaDomainTuxconfig</code>	<code>.1.3.6.1.4.1.140.305.1.5</code>
<code>beaDomainStatus</code>	<code>.1.3.6.1.4.1.140.305.1.6</code>

`beaDomainKey`

Syntax `INTEGER (32769 .. 262143)`

Access read-only

Description Numeric key for the well-known address in a Tuxedo System/T bulletin board. In a single-processor environment, this key names the bulletin board. In a multi-processor environment, this key names the message queue of the DBBL. This key is used as the basis for deriving the names of resources other than the well-known address, such as the names for the bulletin boards throughout the application.

beaLogicalAgentName

Syntax	<i>DisplayString</i> (SIZE(1..32))
Access	read-only
Description	The logical agent name of the agent as specified in the -l option when the agent was started (UNIX systems). On Windows NT systems, the logical agent name is the name of the Windows NT service used to start the agent. This is the agent that monitors the domain. If there are multiple SNMP agents running on a managed node, this name needs to be appended to the community string with an @ sign when sending an SNMP request to the agent. For example, if there are two logical agents <i>simp_snmpd</i> and <i>bank_snmpd</i> , the default communities used to query values from these agents would be <i>public@simp_snmpd</i> and <i>public@bank_snmpd</i> , respectively. To run multiple agents on the same managed node, they must be run as subagents (without the -s option) with the agent integrator.

beaDomainId

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	This is the BEA domain identifier of the domain being managed by this agent. This object is optional.

beaDomainTuxdir

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	The <i>tuxdir</i> value for the domain being managed by this agent. <i>tuxdir</i> is the absolute path name to the directory where the Tuxedo software is found on the master machine.

beaDomainTuxconfig

Syntax	<i>DisplayString</i> (SIZE(1..64))
Access	read-only
Description	The absolute location, including file name, for the configuration file of the domain being managed by this agent.

beaDomainStatus

Syntax	INTEGER { active(1), inactive(2) }
Access	read-only
Description	This object represents the current state of the domain being managed. The values and their interpretation are the same as for <code>tuxTdomainState</code> .

5 WLE MIBs

This chapter describes five MIB tables that are specific to WLE. To access the objects in these MIB tables, the WLE version of the BEA SNMP Agent should be running on the machine where WLE application resources are accessible. Table 5-1 lists the WLE groups.

In addition to the objects in these WLE specific groups, the Tuxedo Core MIB contains the following WLE specific objects:

- `wleMaxObjects`
- `wleMaxInterfaces`
- `wleCurInterfaces`
- `wleHwInterfaces`
- `wleMachineMaxObjects`
- `wleMachineCurObjects`
- `wleMachineHwObjects`
- `wleSrvrCurObjsExt`
- `wleSrvrCurInterfaceExt`
- `wleSrvrSrvType`
- `wleSrvrClassPath`
- `wleSrvrjavaHeap`
- `wleSrvrjavaHeapuse`
- `wleSrvrjavaVendor`
- `wleSrvrjavaVersion`

The object `tuxTranGstate` also has WLE specific states. For more information on these objects, refer to Chapter 2, “Tuxedo Core MIB.”

Table 5-1 WLE Specific MIB Groups

Group Name	Description
	This table represents occurrences of factories registered with the FactoryFinder. The available factories for the WLE application are reflected in this MIB group.
wleInterfaceTable	This group represents the configuration and runtime attributes of CORBA interfaces at both the domain and server-group levels.
wleLclInterfaceTable	The instances in this table return local wleInterfaceTable attributes for the local host on which the BEA SNMP Agent is running.
wleIfQueueTable	The instances in this table represent the runtime attributes of interface as it pertains to a particular server queue (tuxTqueue).
wleLclIfQueueTable	The instances in this table represent the local attributes of wleIfQueueTable instances. These values are specific to the host on which the BEA SNMP Agent is running.
wleJdbcConPoolTable	This class represents the configuration and runtime attributes of JDBC connection pools on a Java server.
wleJdbcConPoolExtnTable	This class represents the extentions to the configuration and runtime attributes of JDBC connection pools on a Java server.
wleModuleTable	This group represents the modules installed on a Java server.

This table represents occurrences of factories registered with the FactoryFinder.

Variable Name	Object ID
wleFactorySerNo	.1.3.6.1.4.1.140.300.48.1.1

Variable Name	Object ID
wleFactoryId	.1.3.6.1.4.1.140.300.48.1.2
wleFactoryIfName	.1.3.6.1.4.1.140.300.48.1.3
wleFactoryState	.1.3.6.1.4.1.140.300.48.1.4

wleFactorySerNo

Syntax INTEGER

Access read-only

Description This object is the running number. This is used as the index to instances in this table.

wleFactoryId

Syntax *DisplayString* (SIZE(1..256))

Access read-only

Description The registered ID for the factory.

wleFactoryIfName

Syntax *DisplayString* (SIZE(1..128))

Access read-only

Description The fully qualified interface name for the factory. This is the interface repository ID for the factory. The format of this name is dependent on the options specified in the IDL that generates the interface implementation. Consult the CORBA 2.1 specification, section 7.6, for details.

wleFactoryState

Syntax INTEGER { active(1) }

Access read-only

Description A GET operation retrieves runtime information for the selected `wleFactoryTable` instance or instances. The returned value is 1 (`active`) if the instance is registered with the `FactoryFinder`.

wleInterfaceTable

The `wleInterfaceTable` represents configuration and runtime attributes of CORBA interfaces at both the domain and server-group levels.

There are certain semantic differences in the objects of the `wleInterfaceTable` between server-group and domain level instances.

A domain-level `wleInterfaceTable` instance is not associated with a Server group. In this case, its `wleIfSrvGrp` attribute has the invalid value `*`.

A server-group level instance has an associated Server group. In this case, its `wleIfSrvGrp` attribute has a valid server group name for the domain. This server-group level representation of an interface also provides a container for managing the interface state (the `wleIfState` object) and for collecting accumulated statistics.

Every CORBA interface that is activated in a server must have a a server-group level `wleInterfaceTable` instance. The activation of interfaces in a server is controlled by the state of a `wleIfQueue` instance for the interface. Activation of an `wleIfQueue` instance causes its attributes to be initialized with values specified for the associated server-group level `wleInterfaceTable` instance. If such an instance does not already exist, then one is dynamically created. This dynamically created server-group level `wleInterfaceTable` instance is initialized with the attributes of the domain-level `wleInterfaceTable` instance for the interface, if one exists. If an associated domain-level instance does not exist, system-specified default configuration values are used. After they are activated, interfaces are always associated with a server-group level `wleInterfaceTable` instance.

The specification of configuration attributes for interfaces at any level is optional. Interfaces offered by a server are identified through the ICF file used for generating skeletons. The interfaces are advertised automatically by the system when the server is activated.

The following table lists the objects within the `wleInterfaceTable`.

Variable Name	Object ID
wleIfSerNo	.1.3.6.1.4.1.140.300.53.1.1.1
wleIfName	.1.3.6.1.4.1.140.300.53.1.1.2
wleIfSrvGrp	.1.3.6.1.4.1.140.300.53.1.1.3
wleIfState	.1.3.6.1.4.1.140.300.53.1.1.4
wleIfAutoTran	.1.3.6.1.4.1.140.300.53.1.1.5
wleIfLoad	.1.3.6.1.4.1.140.300.53.1.1.6
wleIfPrio	.1.3.6.1.4.1.140.300.53.1.1.7
wleIfTimeout	.1.3.6.1.4.1.140.300.53.1.1.8
wleIfTranTime	.1.3.6.1.4.1.140.300.53.1.1.9
wleIfFbRoutingName	.1.3.6.1.4.1.140.300.53.1.1.10
wleIfLmid	.1.3.6.1.4.1.140.300.53.1.1.11
wleIfNumServers	.1.3.6.1.4.1.140.300.53.1.1.12
wleIfTpPolicy	.1.3.6.1.4.1.140.300.53.1.1.13
wleIfTxPolicy	.1.3.6.1.4.1.140.300.53.1.1.14

wleIfSerNo

Syntax	INTEGER
Access	read-only
Description	This is the running number. This object is used as an index to instances in this table.

wleIfName

Syntax	<i>DisplayString</i> (SIZE(1..128))
Access	read-only
Description	The fully qualified interface name. This is the interface ID. The format of this name is one of the options specified in the IDL that generates the interface implementation. Consult the CORBA 2.1 specification, Section 7.6, for details.

wleIfSrvGrp

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	The server group name. Server group names cannot contain an asterisk, comma, or colon. An asterisk (*) specified as a value for this object specifies a domain level instance.

Note: This object can be SET only during creation of a new row.

wleIfState

Syntax	INTEGER { active(1), inactive(2), suspended(3), partitioned(4), invalid(5), reactivate(6) }
Access	read-write
Description	The semantics for GET and SET requests differ between server-group and domain level instances as noted in the following list.

GET: {active(1) | inactive(2) | suspended(3) | partitioned(4) }

A GET request retrieves configuration information for the selected wleInterfaceTable instance or instances. The only states that can be returned are: active, inactive, suspended, partitioned.

active(1)

The wleInterfaceTable instance is defined and at least one corresponding wleIfQueueTable instance is in the active state. For a server-group level wleInterfaceTable instance, corresponding wleIfQueueTable instances are those with matching wleIfName and wleIfSrvGrp objects. For a

domain-level `wleInterfaceTable` instance, corresponding `wleIfQueueTable` instances are those with matching `wleIfName` value regardless of their `wleIfSrvGrp` value.

`inactive(2)`

The `wleInterfaceTable` instance is defined and there are no corresponding `wleIfQueueTable` instances in any active state.

`suspended(3)`

The `wleInterfaceTable` instance is defined and among all corresponding `wleIfQueueTable` instances, there are none in the active state and at least one in the suspended state.

`partitioned(4)`

The `wleInterfaceTable` instance is defined and among all the corresponding `wleIfQueueTable` instances, there are none in the active state, none in the suspended state, and at least one in the partitioned state.

SET: { `invalid(5)` | `active(1)` | `inactive(2)` | `reactivate(6)` | `suspended(3)` }

A SET request updates runtime and configuration information for the selected `wleInterfaceTable` instance. Modifications can affect more than one server group when domain-level changes are made, and runtime modifications can affect more than one server if multiple servers are currently offering an interface. Only the following values can be used in a SET request: `invalid`, `active`, `reactivate`, or `suspended`.

`invalid(5)`

Delete the `wleInterfaceTable` instance for the application. This state change is allowed only when the instance is in the inactive state.

`active(1)`

Activate the `wleInterfaceTable` instance for the application. Setting this state on a domain level instance has the effect of activating all corresponding `wleIfQueueTable` instances that are currently suspended throughout the domain. Setting this state on a server-group level instance affects only servers within the group offering the interface. This state change is allowed only when the instance is in the suspended state. A successful return leaves the object in the `active(1)` state.

`reactivate(6)`

Reactivates the `wleInterfaceTable` instance. Setting this state on a domain level instance has the effect of activating all corresponding `wleIfQueueTable` instances that are currently suspended throughout the domain. Setting this state on a server-group level instance affects only servers within the group offering the interface. This state change is allowed only when the instance is in the `active(1)` or `suspended(3)` state. Successful return leaves the instance in the `active(1)` state. Setting this state permits global activation of `wleIfQueueTable` instances suspended at the server-group level without having to individually activate each server-group level `wleInterfaceTable` instance.

`suspended(3)`

Suspend the `wleInterfaceTable` instance. Setting this state on the domain-level object has the effect of suspending all corresponding `wleIfQueueTable` instances that are currently active throughout the domain. Setting this state on a server-group level instance affects only servers within the group offering the interface. This state change is permitted only in the `active(1)` state. Successful return leaves the object in the `suspended(3)` state.

Note: Dynamic advertisement of interfaces (that is, state change from `inactive(2)` or `invalid(5)` to `active(1)`) is not supported, nor is removal of advertisement (that is, state change from `active(1)` to `inactive(2)`).

wleIfAutoTran

Syntax	INTEGER { yes(1), no(2) }
Access	read-write
Description	Signifies whether a transaction is automatically started for invocations made outside a transaction context.

This object has the following limitations:

- Runtime updates to this attribute are not reflected in active equivalent `wleInterfaceTable` instances.
- The `wleIfTxPolicy` object can override the value specified for this attribute in the `ubbcconfig` file. If `wleIfTxPolicy` is `always(1)`, a `wleIfAutoTran` value of `no(2)` has no effect at runtime. Behavior is as though the setting were `yes(1)`. If `wleIfTxPolicy` is `never(2)`, a `wleIfAutoTran` value of `yes(1)` has no effect. The interface is never involved in a transaction. If

wleIfTxPolicy is ignore(4), an wleIfAutoTran value of yes(1) has no effect. The interface is never involved in a transaction.

wleIfLoad

Syntax	INTEGER (1..32767)
Access	read-write
Description	This object imposes the indicated load on the system. Interface loads are used for load-balancing. That is, queues with higher enqueued workloads are less likely to be chosen for a new request.
Note:	Runtime updates to this attribute for domain level instances do not affect corresponding server-group level instances for the same interface.

wleIfPrio

Syntax	INTEGER (1..100)
Access	read-write
Description	Dequeueing priority. If multiple interface requests are waiting on a queue for servicing, the higher priority requests are handled first.
Note:	Runtime updates to this attribute for domain-level instances do not affect corresponding server-group level instances for the same interface.

wleIfTimeout

Syntax	INTEGER
Access	read-write
Description	The time limit (in seconds) for processing individual method invocations for this interface. Servers that process method invocations for this interface are terminated abortively if they exceed the specified time limit in processing the request. A value of 0 for this attribute indicates that the server should not be terminated abortively.
Note:	Runtime updates to this attribute for domain-level instances do not affect corresponding server-group level instances for the same interface.

wleIfTranTime

Syntax	INTEGER
Access	read-write
Description	Transaction timeout value in seconds for transactions automatically started for this wleInterfaceTable instance. Transactions are started automatically when a request not in transaction mode is received and the wleIfAutoTran object value for the interface is yes(1).
Note:	Runtime updates to this attribute for domain-level instances do not affect corresponding server-group level instances for the same interface.

wleIfFbRoutingName

Syntax	DisplayString (SIZE(1..15))
Access	read-write
Description	The factory-based routing criteria associated with this interface.
Note:	This attribute can be set only for a domain-level wleInterfaceTable instance, that is, only if wleIfSrvGrp is *.

wleIfLmid

Syntax	DisplayString (SIZE(1..30))
Access	read-only
Description	Current logical machine with which the active equivalent server-group level wleInterfaceTable instance is associated. This attribute is NULL for domain-level instances.

wleIfNumServers

Syntax	INTEGER
Access	read-only
Description	The number of corresponding servers that offer this interface.

wleIfTpPolicy

Syntax	INTEGER { method(1), transaction(2), process(3) }
Access	read-only
Description	The TP framework deactivation policy. This reflects the policy registered with the framework at server startup. The first server to register with the interface sets the value in <code>wleInterfaceTable</code> . This cannot be changed.

wleIfTxPolicy

Syntax	INTEGER { always(1), never(2), optional(3), ignore(4) }
Access	read-only
Description	The transaction policy for the interface. The setting in this attribute affects the <code>wleIfAutoTran</code> object. This policy is set by the application developer and is registered when the server starts.

wleLclInterfaceTable

The table returns values for the local host on which the Agent Connection is running. The following table lists the columnar objects that comprise each row (instance) in the table.

Variable Name	Object ID
wleLclIfSerNo	.1.3.6.1.4.1.140.300.53.2.1.1
wleLclIfName	.1.3.6.1.4.1.140.300.53.2.1.2
wleLclSrvGrp	.1.3.6.1.4.1.140.300.53.2.1.3
wleLclIfNcompleted	.1.3.6.1.4.1.140.300.53.2.1.4
wleLclIfNqueued	.1.3.6.1.4.1.140.300.53.2.1.5

wleLclIfSerNo

Syntax	INTEGER
Access	read-only
Description	This is the running number. This number is used as an index into the table.

wleLclIfName

Syntax	<i>DisplayString</i> (SIZE(1..128))
Access	read-only
Description	The fully qualified interface name. The interface repository ID for the interface. The format of this name is dependent on the options specified in the IDL that generates the interface implementation. See the CORBA 2.1 Specification Section 7.6 [CORBA] for details.

wleLclIfSrvGrp

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	The server group name. Server group names cannot contain an asterisk, comma, or colon. A value of * for this object indicates a domain-level interface.

wleLclIfNcompleted

Syntax	INTEGER
Access	read-only
Description	The number of method invocations completed for the corresponding <code>wleIfQueueTable</code> instances since they were initially offered. The values returned are for the indicated interface on the local host where the BEA SNMP Agent is running.
Note:	This attribute is returned only when <code>tuxTdomainLoadBalance</code> is <code>yes(1)</code> .

wleLclIfNqueued

Syntax	INTEGER
Access	read-only
Description	The number of requests currently enqueued for this interface. The values returned are for the indicated interface on the local host where the BEA SNMP Agent is running.
Note:	This attribute is returned only when <code>tuxTdomainLoadBalance</code> is <code>yes(1)</code> .

wleIfQueueTable

This table represents the runtime attributes of an interface for a particular server queue. This group provides access to the inherited configuration attributes of an interface as well as statistics relating to the interface on the queue. This class gives administrators finer granularity in suspending and activating interfaces. This group provides the link between the interface name and the server processes capable of processing method invocations on the interface. For example, `wleIfQRqAddr` can be used to identify the corresponding server in the `tuxTsrvrTbl` and `tuxTsrvrTblExt` groups.

Variable Name	Object ID
<code>wleIfQueueSerNo</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.1</code>
<code>wleIfQueueName</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.2</code>
<code>wleIfQueueSrvGrp</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.3</code>
<code>wleIfQueueRqAddr</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.4</code>
<code>wleIfQueueState</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.5</code>
<code>wleIfQueueAutoTran</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.6</code>
<code>wleIfQueueLoad</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.7</code>
<code>wleIfQueuePrio</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.8</code>
<code>wleIfQueueTimeout</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.9</code>
<code>wleIfQueueTranTime</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.10</code>
<code>wleIfQueueFbRoutingName</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.11</code>
<code>wleIfQueueLmid</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.12</code>
<code>wleIfQueueNumServers</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.13</code>
<code>wleIfQueueTpPolicy</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.14</code>
<code>wleIfQueueTxPolicy</code>	<code>.1.3.6.1.4.1.140.300.53.3.1.15</code>

wlelfQueueSerNo

Syntax	INTEGER
Access	read-only
Description	The running number used as an index into this table.

wlelfQueueName

Syntax	<i>DisplayString</i> (SIZE(1..128))
Access	read-only
Description	The fully qualified interface name. The interface repository ID for the interface. The format of this name is dependent on the options specified in the IDL that generates the interface implementation. See the CORBA 2.1 specification Section 7.6 for details.

wlelfQueueSrvGrp

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	The server group name. Server group names cannot contain an asterisk, comma, or colon.

wlelfQueueRqAddr

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	The symbolic address of the request queue for an active server offering this interface. See <code>tuxTsrvrRqAddr</code> for more information about this attribute.

wleIfQueueState

Syntax	INTEGER { active(1), suspended(2), partitioned(3), unknown(4) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: {active(1) suspended(2) partitioned(3) }</p> <p>A GET request retrieves configuration information for the selected wleIfQueueTable instances. The meaning of the possible return values are as follows:</p> <p>active(1)</p> <p>Represents an available interface in the running system.</p> <p>suspended(2)</p> <p>Represents a currently suspended interface in the running system.</p> <p>partitioned(3)</p> <p>Represents a currently partitioned interface in the running system.</p> <p>SET: {active(1) suspended(2) }</p> <p>The values for SET are:</p> <p>active(1)</p> <p>Activates the wleIfQueueTable instance. This state change is allowed only in the suspended(2) state. A successful return leaves instances in the active(1) state.</p> <p>suspended(2)</p> <p>Suspends the wleIfQueueTable instance. This state change is allowed only in the active(1) state. A successful return leaves the object in the suspended(2) state.</p> <p>Note: Dynamic advertisement of interfaces (that is., a state change from inactive or invalid to active) is not supported, nor is unadvertisement (that is, a state change from active to inactive).</p>

wleIfQueueAutoTran

Syntax	INTEGER { yes(1), no(2) }
Access	read-only
Description	Signifies whether a transaction is automatically started for invocations made outside a transaction context.

This object has the following limitations:

- Runtime updates to this attribute are not reflected in active equivalent `wleInterfaceTable` instances.
- The `wleIfTxPolicy` object can override the value specified for this attribute in the `ubbconfig` file. If `wleIfTxPolicy` is `always(1)`, an `wleIfQueueAutoTran` value of `no(2)` has no effect at runtime. Behavior is as though the setting were `yes(1)`. If `wleIfTxPolicy` is `never(2)`, an `wleIfQueueAutoTran` value of `yes(1)` has no effect. The interface is never involved in a transaction. If `wleIfTxPolicy` is `ignore(4)`, an `wleIfQueueAutoTran` value of `yes(1)` has no effect. The interface is never involved in a transaction.

wleIfQueueLoad

Syntax	INTEGER (1..32767)
Access	read-only
Description	Load imposed on the system by this instance. Interface loads are used for load-balancing. Queues with higher enqueued workloads are less likely to be chosen for a new request.

wleIfQueuePrio

Syntax	INTEGER (1..101)
Access	read-only
Description	Dequeueing priority. If multiple interface requests are waiting on a queue for servicing, the higher priority requests are handled first.

wleIfQueueTimeout

Syntax	INTEGER
Access	read-only
Description	The time limit (in seconds) for processing individual method invocations for this interface. Servers processing method invocations for this interface are abortively terminated if they exceed the specified time limit in processing the request. A value of 0 for this attribute indicates that the server should not be abortively terminated.

wleIfQueueTranTime

Syntax	INTEGER
Access	read-only
Description	The transaction timeout value in seconds for transactions automatically started for this instance. Transactions are started automatically when a request not in transaction mode is received and the wleIfAutoTran attribute value for the interface is yes(1).

wleIfQueueFbRoutingName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-only
Description	The factory-based routing criterion associated with this interface.

wleIfQueueLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	The current logical machine on which this queue is offering this interface.

wlIfQueueNumServers

Syntax	INTEGER
Access	read-only
Description	The number of corresponding servers that offer this interface on this queue.

wlIfQueueTpPolicy

Syntax	INTEGER { method(1), transaction(2), process(3) }
Access	read-only
Description	The TP framework deactivation policy. This reflects the policy registered with the framework at the server startup. The first server to register the interface sets the value. This value cannot be changed.

wlIfQueueTxPolicy

Syntax	INTEGER { always(1), never(2), optional(3), ignore(4) }
Access	read-only
Description	The transaction policy for the interface. The setting in this attribute affects the effect of the wlIfQueueAutoTran object. See wlIfQueueAutoTran for further explanation. This attribute is always read-only. It is set by the developer when the server is built and registered at server startup.

wleLclIfQueueTable

This table represents the local attributes of the wleIfQueueTable. These values are specific to the host on which the BEA SNMP Agent is running.

Variable Name	Object ID
wleLclIfQueueSerNo	.1.3.6.1.4.1.140.300.53.4.1.1
wleLclIfQueueName	.1.3.6.1.4.1.140.300.53.4.1.2
wleLclIfQueueSrvGrp	.1.3.6.1.4.1.140.300.53.4.1.3
wleLclIfQueueRqAddr	.1.3.6.1.4.1.140.300.53.4.1.4
wleLclIfQueueNcompleted	.1.3.6.1.4.1.140.300.53.4.1.5
wleLclIfQueueNqueued	.1.3.6.1.4.1.140.300.53.4.1.6
wleLclIfQueueCurObjs	.1.3.6.1.4.1.140.300.53.4.1.7
wleLclIfQueueCurTrans	.1.3.6.1.4.1.140.300.53.4.1.8

wleLclIfQueueSerNo

Syntax	INTEGER
Access	read-only
Description	The running number used as an index into this table.

wleLclIfQueueName

Syntax	<i>DisplayString</i> (SIZE(1..128))
Access	read-only
Description	The fully qualified interface name. The interface repository ID for this interface. The format of this name is dependent on the options specified in the IDL that generates the interface implementation. See the CORBA 2.1 specification Section 7.6 for details.

wleLclIfQueueSrvGrp

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	The server group name. Server group names cannot contain an asterisk, comma, or colon.

wleLclIfQueueRqAddr

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	The symbolic address of the request queue for an active server that offers this interface. See <code>tuxTsrvRqAddr</code> for more information about this attribute.

wleLclIfQueueNcompleted

Syntax	INTEGER
Access	read-only
Description	The number of interface method invocations completed since the interface was initially offered. Note: This attribute is returned only when <code>tuxTdomainLoadBalance</code> is equal to <code>yes(1)</code> .

wleLclIfQueueNqueued

Syntax	INTEGER
Access	read-only
Description	The number of requests currently enqueued for this interface. Note: This attribute is returned only when <code>tuxTdomainLoadBalance</code> is equal to <code>yes(1)</code> .

wleLclIfQueueCurObjs

Syntax	INTEGER
Access	read-only
Description	The number of active objects for this interface for the associated queue. This number represents the number of entries in the active object table for this queue on the associated machine. This number includes objects that are not in memory but were invoked within an active transaction.

wleLclIfQueueCurTrans

Syntax	INTEGER
Access	read-only
Description	The number of active global transactions associated with this interface for its associated queue.

wleJdbcConPoolTable

This class represents the configuration and runtime attributes of JDBC connection pools on a Java server. The attributes consist of statistics or values associated with each connection pool. Except for `tuxJdbcConPoolState`, attributes are persistent in TUXCONFIG. Local attributes are local to the memory allocated to a Java server.

Variable Name	Object ID
wleJdbcConPoolSrvId	.1.3.6.1.4.1.140.300.60.1.1.10
wleJdbcConPoolSrvGrp	.1.3.6.1.4.1.140.300.60.1.1.20
wleJdbcConPoolDsName	.1.3.6.1.4.1.140.300.60.1.1.30
wleJdbcConPoolDriver	.1.3.6.1.4.1.140.300.60.1.1.40
wleJdbcConPoolUrl	.1.3.6.1.4.1.140.300.60.1.1.50
wleJdbcConPoolState	.1.3.6.1.4.1.140.300.60.1.1.60
wleJdbcConPoolDbName	.1.3.6.1.4.1.140.300.60.1.1.70
wleJdbcConPoolDbUser	.1.3.6.1.4.1.140.300.60.1.1.80
wleJdbcConPoolDbPassword	.1.3.6.1.4.1.140.300.60.1.1.90
wleJdbcConPoolUserRole	.1.3.6.1.4.1.140.300.60.1.1.100
wleJdbcConPoolDbHost	.1.3.6.1.4.1.140.300.60.1.1.110
wleJdbcConPoolDbNetProtocol	.1.3.6.1.4.1.140.300.60.1.1.120
wleJdbcConPoolDbPort	.1.3.6.1.4.1.140.300.60.1.1.130
wleJdbcConPoolProps	.1.3.6.1.4.1.140.300.60.1.1.140
wleJdbcConPoolEnableXa	.1.3.6.1.4.1.140.300.60.1.1.150
wleJdbcConPoolCreateOnStartUp	.1.3.6.1.4.1.140.300.60.1.1.160
wleJdbcConPoolLoginDelay	.1.3.6.1.4.1.140.300.60.1.1.170
wleJdbcConPoolInitCapacity	.1.3.6.1.4.1.140.300.60.1.1.180
wleJdbcConPoolMaxCapacity	.1.3.6.1.4.1.140.300.60.1.1.190

Variable Name	Object ID
wleJdbcConPoolCapacityIncr	.1.3.6.1.4.1.140.300.60.1.1.200
wleJdbcConPoolAllowShrinking	.1.3.6.1.4.1.140.300.60.1.1.210
wleJdbcConPoolShrinkPeriod	.1.3.6.1.4.1.140.300.60.1.1.220
wleJdbcConPoolTestTable	.1.3.6.1.4.1.140.300.60.1.1.230
wleJdbcConPoolRefresh	.1.3.6.1.4.1.140.300.60.1.1.240
wleJdbcConPoolTestOnReserve	.1.3.6.1.4.1.140.300.60.1.1.250
wleJdbcConPoolTestOnRelease	.1.3.6.1.4.1.140.300.60.1.1.260
wleJdbcConPoolWaitForConn	.1.3.6.1.4.1.140.300.60.1.1.270
wleJdbcConPoolWaitTimeOut	.1.3.6.1.4.1.140.300.60.1.1.280

wleJdbcConPoolSrvId

Syntax	<i>INTEGER (1..30001)</i>
Access	read-only
Description	Together with the server group name, this value is used to identify a Java server, specified in the SERVERS section of the UBBCONFIG file, for which the connection pool is being described.

wleJdbcConPoolSrvGrp

Syntax	<i>DisplayString (SIZE (1..15))</i>
Access	read-only
Description	Name of a server group. This attribute is used to identify a Java server specified in the SERVERS section of the UBBCONFIG file, for which the connection pool is being described.

wleJdbcConPoolDsName

Syntax *DisplayString (SIZE (0..30))*

Access read-only

Description The data source name for the connection pool.

wleJdbcConPoolDriver

Syntax *DisplayString (SIZE (0..256))*

Access read-write

Description The class name for the Java driver.

wleJdbcConPoolUrl

Syntax *DisplayString (SIZE (2..256))*

Access read-write

Description URL for a JDBC driver that is not JDBC 2.0-compliant.

wleJdbcConPoolState

Syntax INTEGER { valid(1) | invalid(2) }

Access read-write

Description This object denotes the current state of the tuxDmConnection instance.

GET requests:

valid(1): The object exists.

SET requests:

invalid(1): Delete object.

wleJdbcConPoolDbName

Syntax	<i>DisplayString</i> (SIZE (0..30))
Access	read-write
Description	The database name.

wleJdbcConPoolDbUser

Syntax	<i>DisplayString</i> (SIZE (0..30))
Access	read-write
Description	The user's account name.

wleJdbcConPoolDbPassword

Syntax	<i>DisplayString</i> (SIZE (0..64))
Access	read-write
Description	The user's password. The password entered by the user should not exceed 24 bytes.

wleJdbcConPoolUserRole

Syntax	<i>DisplayString</i> (SIZE (0..30))
Access	read-write
Description	The user's SQL role.

wleJdbcConPoolDbHost

Syntax	<i>DisplayString</i> (SIZE (0..30))
Access	read-write
Description	The database server name.

wleJdbcConPoolDbNetProtocol

Syntax	<i>DisplayString (SIZE (0..30))</i>
Access	read-write
Description	The protocol used to communicate with the database.

wleJdbcConPoolDbPort

Syntax	<i>INTEGER (1..65624)</i>
Access	read-write
Description	The port used for database connections.

wleJdbcConPoolProps

Syntax	<i>DisplayString (SIZE (0..256))</i>
Access	read-write
Description	Vendor-specific information for the JDBC driver.

wleJdbcConPoolEnableXa

Syntax	<i>INTEGER { yes(1) no(2) }</i>
Access	read-write
Description	If set to <i>yes</i> , indicates that the pool supports XA mode.

wleJdbcConPoolCreateOnStartUp

Syntax	<i>INTEGER { yes(1) no(2) }</i>
Access	read-write
Description	If set to <i>yes</i> , indicates that the connection pool is created when the first request arrives.

wleJdbcConPoolLoginDelay

Syntax	INTEGER
Access	read-write
Description	The login delay in seconds.

wleJdbcConPoolInitCapacity

Syntax	INTEGER
Access	read-write
Description	The number of connections initially supported in the connection pool. This number should not exceed the value of <code>tuxJdbcConPoolMaxCapacity</code> .

wleJdbcConPoolMaxCapacity

Syntax	INTEGER
Access	read-write
Description	The maximum number of connections supported in the connection pool.

wleJdbcConPoolCapacityIncr

Syntax	INTEGER
Access	read-write
Description	The number of connections added to the pool when the current limit is exceeded but the maximum capacity is not yet reached.

wleJdbcConPoolAllowShrinking

Syntax	INTEGER { <code>yes(1)</code> <code>no(2)</code> }
Access	read-write
Description	If set to <code>yes(1)</code> , allows connection pool shrinking.

wleJdbcConPoolShrinkPeriod

Syntax	INTEGER
Access	read-write
Description	The interval (in minutes) after which shrinking occurs.

wleJdbcConPoolTestTable

Syntax	<i>DisplayString</i> (SIZE (0..256))
Access	read-write
Description	<p>The name of a table in the database that is used to test the viability of connections in the connection pool.</p> <p>The query <code>select count (*) from TESTTABLE</code> is used to test a connection. The table must exist and be accessible to the database user for the connection.</p>

wleJdbcConPoolRefresh

Syntax	INTEGER
Access	read-write
Description	<p>The refresh interval (in minutes).</p> <p><code>wleJdbcConPoolRefresh 0<num</code></p>

wleJdbcConPoolTestOnReserve

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	<p>If set to <code>yes(1)</code>, the Java server tests a connection after removing it from the pool and before giving it to the client. The test adds a small delay in serving the client's request for a connection from the pool, but ensures that the client receives a working connection. A value for <code>wleJdbcConPoolTestTable</code> must be set for this feature to work.</p>

wleJdbcConPoolTestOnRelease

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	If set to yes(1), the Java server tests a connection before returning it to the connection pool. If all the connections in the pool are already in use and a client is waiting for a connection, the client's wait is slightly longer due to the test of the connection. A value for tuxJdbcConPoolTestTable must be set for this feature to work.

wleJdbcConPoolWaitForConn

Syntax	INTEGER { yes(1) no(2) }
Access	read-write
Description	<p>If set to yes(1), this feature enables an application to wait for a connection indefinitely if none is currently available. If set to no(2), a request for a connection returns to the caller immediately if there is no connection available.</p> <p>The default is yes(1) unless wleJdbcConPoolWaitTimeOut is specified, in which case it becomes no(2).</p>

wleJdbcConPoolWaitTimeOut

Syntax	INTEGER
Access	read-only
Description	Time (in seconds) that an application waits for a connection to become available.

wleJdbcConPoolExtnTable

An extension of the wleJdbcConPoolTable table.

Variable Name	Object ID
wleJdbcConPoolExtnSrvId	.1.3.6.1.4.1.140.300.60.2.1.10
wleJdbcConPoolExtnSrvGrp	.1.3.6.1.4.1.140.300.60.2.1.20
wleJdbcConPoolExtnDsName	.1.3.6.1.4.1.140.300.60.2.1.30
wleJdbcConPoolExtnConnUsed	.1.3.6.1.4.1.140.300.60.2.1.40
wleJdbcConPoolExtnConnAvailable	.1.3.6.1.4.1.140.300.60.2.1.50
wleJdbcConPoolExtnHwmConnUsed	.1.3.6.1.4.1.140.300.60.2.1.60
wleJdbcConPoolExtnHwmConnCreated	.1.3.6.1.4.1.140.300.60.2.1.70
wleJdbcConPoolExtnAwaitingConn	.1.3.6.1.4.1.140.300.60.2.1.80
wleJdbcConPoolExtnHwmForWait	.1.3.6.1.4.1.140.300.60.2.1.90

wleJdbcConPoolExtnSrvId

Syntax	INTEGER (1..30001)
Access	read-only
Description	Together with the server group name, this value is used to identify a Java server, specified in the SERVERS section of the UBBCONFIG file, for which the connection pool is being described.

wleJdbcConPoolExtnSrvGrp

Syntax	DisplayString (SIZE (1..15))
Access	read-only
Description	Name of a server group. This attribute is used to identify a Java server specified in the SERVERS section of the UBBCONFIG file, for which the connection pool is being described.

wleJdbcConPoolExtnDsName

Syntax *DisplayString* (SIZE (0..30))

Access read-only

Description The data source name for the connection pool.

wleJdbcConPoolExtnConnUsed

Syntax INTEGER

Access read-only

Description T

wleJdbcConPoolExtnConnAvailable

Syntax INTEGER

Access read-only

Description T

wleJdbcConPoolExtnHwmConnUsed

Syntax INTEGER

Access read-only

Description T

wleJdbcConPoolExtnHwmConnCreated

Syntax INTEGER

Access read-only

Description T

wleJdbcConPoolExtnAwaitingConn

Syntax INTEGER

Access read-only

Description T

wleJdbcConPoolExtnHwmForWait

Syntax INTEGER

Access read-only

Description T

wleModuleTable

This group represents the modules installed on a Java server. The class attributes identify and characterize the module.

Variable Name	Object ID
wleModuleSrvId	.1.3.6.1.4.1.140.300.65.1.1.10
wleModuleSrvGrp	.1.3.6.1.4.1.140.300.65.1.1.20
wleModuleModule	.1.3.6.1.4.1.140.300.65.1.1.30
wleModuleClassPath	.1.3.6.1.4.1.140.300.65.1.1.40
wleModuleModuleArgs	.1.3.6.1.4.1.140.300.65.1.1.50
wleModuleState	.1.3.6.1.4.1.140.300.65.1.1.60
wleModuleModuleType	.1.3.6.1.4.1.140.300.65.1.1.70

wleModuleSrvId

Syntax	<i>INTEGER (1..30001)</i>
Access	read-only
Description	With the server group name, this value is used to identify a Java server, specified in the SERVERS section of the UBBCONFIG file for which the connection pool is being described.

wleModuleSrvGrp

Syntax	<i>DisplayString (SIZE (0..20))</i>
Access	read-only
Description	The name of a server group. This is used to identify a Java server, specified in the SERVERS section of the UBBCONFIG file, for which the connection pool is being described.

wleModuleModule

Syntax	<i>DisplayString</i> (SIZE (0..256))
Access	read-only
Description	The Java Archive (JAR) file that contains the module.

wleModuleModuleArgs

Syntax	<i>DisplayString</i> (SIZE (0..256))
Access	read-write
Description	The startup arguments for the module.

wleModuleState

Syntax	INTEGER { valid(1) invalid(2) }
Access	read-write
Description	This object denotes the current state of the wle module.
	GET: valid(1) The object exists.
	SET: invalid(2) Delete object.

wleModuleModuleType

Syntax	INTEGER
Access	read-only
Description	The type of the module. This is either CORBA or EJB.

6 Access Control List MIB

An *access control list* (ACL) specifies who and what is authorized to access Tuxedo or WLE system objects. The ACL MIB enables a system manager to administer Tuxedo or WLE security by authenticating users, setting permissions, and controlling access. The ACL MIB defines the objects controlled by the ACL facility. These MIB objects are grouped into three major categories. The following table lists groups that make up the ACL MIB.

Group Name	Description
<code>tuxTAclGrpTable</code>	ACL group
<code>tuxTAclPermTable</code>	ACL permissions
<code>tuxTAclPrinTbl</code>	ACL principal (users or domains)

For Tuxedo or WLE security, define application security options in the Domain group. This group lets you specify a user identity and security type used by your Tuxedo application. The users and remote domains in an application that need authentication and authorization are collectively known as *principals*. The managed objects for getting or setting the values of principals are defined in the `tuxTAclPrinTbl` group. The managed objects for getting or setting the values of ACL groups are defined in the `tuxTAclGrpTable`. The ACL MIB, as a whole, specifies the principals and access control lists for Tuxedo applications services, application queues, and events. You can define these *ACL permissions* for service, event, and application queue names. The managed objects that enable you to do this are defined in the `tuxTAclPermTable` group. All these ACL MIB groups and their objects are described in the following sections.

tuxTAclGrpTable

The tuxTAclGrpTable group represents groups of Tuxedo application users and domains. The following table lists the managed objects that are part of the tuxTAclGrpTable group. To create a new row in the table, it is necessary to issue a SET request for a non-existing row.

Variable Name	Object ID
tuxTAclGrpName	.1.3.6.1.4.1.140.300.11.1.1.1.1
tuxTAclGrpId	.1.3.6.1.4.1.140.300.11.1.1.1.2
tuxTAclGrpState	.1.3.6.1.4.1.140.300.11.1.1.1.3

tuxTAclGrpName

- Syntax

DisplayString (SIZE(1..30))
- Access

read-write
- Description

Logical name of the group. A group name is a string of printable characters and cannot contain a pound sign, comma, colon, or newline.

Note: This object can be set only during row creation.

tuxTAclGrpId

- Syntax

INTEGER (0..16384)
- Access

read-write
- Description

Group identifier associated with this user. A value of 0 indicates the default group other. If the group identifier is not specified at creation time, it defaults to the next available (unique) identifier greater than 0.

tuxTAclGrpState

Syntax INTEGER { valid(1), invalid(2) }

Access read-write

Description The values for GET and SET operations are as follows:

GET: valid(1)

A GET operation retrieves configuration information for the selected tuxTAclGrpTable instance(s). The following state indicates the meaning of a tuxTAclGrpState returned in response to a GET request. States not listed are not returned.

valid(1)

tuxTAclGrpTable instance is defined and inactive. Note that this is the only valid state for this class. ACL groups are never active.

SET: invalid(2)

A SET operation updates configuration information for the selected tuxTAclGrpTable instance. The following state indicates the meaning of a tuxTAclGrpState set in a SET request. States not listed might not be set.

invalid(2)

Delete tuxTAclGrpTable instance for application. Successful return removes the instance from the table.

tuxTAclPermTable

The `tuxTAclPermTable` group indicates what groups are allowed to access Tuxedo system entities. These entities are named by a string. The names currently represent service names, event names, and application queue names. To create a new row in this table, it is necessary to issue a `SET` request for a non-existing row that specifies at least the values for `tuxTAclPermName` and `tuxTAclPermType`.

Variable Name	Object ID
tuxTAclPermName	.1.3.6.1.4.1.140.300.11.2.1.1.1
tuxTAclPermType	.1.3.6.1.4.1.140.300.11.2.1.1.2
tuxTAclPermGrpIds	.1.3.6.1.4.1.140.300.11.2.1.1.3
tuxTAclPermState	.1.3.6.1.4.1.140.300.11.2.1.1.4

tuxTAclPermName

- Syntax

`DisplayString (SIZE(1..30))`
- Access

read-write
- Description

The name of the entity for which permissions are being granted. The name can represent a service name, an event name, and/or a queue name. An ACL name is a string of printable characters and cannot contain a colon, pound sign, or newline.

Note: This object can be set only during row creation.

tuxTAclPermType

- Syntax

`INTEGER { enq(1), deq(2), service(3), postevent(4) }`
- Access

read-write
- Description

The type of the entity for which permissions are being granted.

Note: This object can be set only during row creation.

tuxTAclPermGrpIds

Syntax	<i>DisplayString</i> (SIZE(0..800))
Access	read-write
Description	A comma-separated list of group identifiers (numbers) that are permitted access to the associated entity.

tuxTAclPermState

Syntax	INTEGER { valid(1), invalid(2) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: valid(1)</p> <p>A GET operation retrieves configuration information for all selected entities. The following state indicates the meaning of a tuxTAclPermState returned in response to a GET request. States not listed are not returned.</p> <p>valid(1)</p> <p>tuxTAclPermState instance is defined. Note that this is the only valid state for this class. ACL permissions are never active.</p> <p>SET: invalid(2)</p> <p>A SET operation updates configuration information for the selected tuxTAclPermState instance. The following state indicates the meaning of a tuxTAclPermState set in a SET request. States not listed might not be set.</p> <p>invalid(2)</p> <p>Delete tuxTAclPermState instance for application. State change allowed only when in the valid(1) state. Successful return leaves the object in the invalid(2) state.</p> <p>Note that the tuxTAclPermTable instance refers to all groupids related to a particular tuxTAclPermName in the table.</p>

tuxTAclPrinTbl

The `tuxTAclPrinTbl` group represents users or domains that can access a Tuxedo application and the group with which they are associated. To join the application as a specific user, it is necessary to present a user-specific password. To create a new row in this table, it is necessary to issue a `SET` request for a non-existing row (instance).

Variable Name	Object ID
tuxTAclPrinName	.1.3.6.1.4.1.140.300.11.3.1.1.1
tuxTAclCltnName	.1.3.6.1.4.1.140.300.11.3.1.1.2
tuxTAclPrinId	.1.3.6.1.4.1.140.300.11.3.1.1.3
tuxTAclPrinGrp	.1.3.6.1.4.1.140.300.11.3.1.1.4
tuxTAclPrinPasswd	.1.3.6.1.4.1.140.300.11.3.1.1.5
tuxTAclPrinState	.1.3.6.1.4.1.140.300.11.3.1.1.6

tuxTAclPrinName

- Syntax *DisplayString* (SIZE(1..30))
- Access read-write
- Description Logical name of the user or domain (a principal). A principal name is a string of printable characters and cannot contain a pound sign, colon, or newline.

Note: This object can be set only during row creation.

tuxTAclCltName

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-write
Description	The client name associated with the user. It generally describes the role of the associated user and provides a further qualifier on the user entry. If the client name is not specified at creation time, the default is the wildcard asterisk (*). A client name is a string of printable characters and cannot contain a colon or newline.

tuxTAclPrinId

Syntax	INTEGER (1..131072)
Access	read-write
Description	Unique user identification number. If not specified at creation time, it defaults to the next available (unique) identifier greater than 0.

Note: This object can be set only during row creation.

tuxTAclPrinGrp

Syntax	INTEGER (0..16384)
Access	read-write
Description	Group identifier associated with this user. A value of 0 indicates the default group <i>other</i> . If the group identifier is not specified at creation time, the default value 0 is assigned.

tuxTAclPrinPasswd

Syntax	<i>DisplayString</i>
Access	read-write
Description	The clear-text authentication password for the associated user. Note that the system automatically encrypts this information on behalf of the administrator.

tuxTAclPrinState

Syntax	INTEGER { valid(1), invalid(2) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: valid(1)</p> <p>A GET operation retrieves configuration information for the selected tuxTAclPrinTbl instance(s). The following state indicates the meaning of tuxTAclPrinState:</p> <p>valid(1)</p> <p>tuxTAclPrinTbl instance is defined and inactive. Note that this is the only valid state for this class. ACL principals are never active.</p> <p>SET: invalid(2)</p> <p>A SET operation updates configuration information for the selected tuxTAclPrinTbl instance. The following state indicates the meaning of a tuxTAclPrinState set in a SET request. States not listed might not be set.</p> <p>invalid(2)</p> <p>Delete tuxTAclPrinTbl instance for application. State change is allowed only when in the valid(1) state. Successful return leaves the object in the invalid(2) state.</p>

7 Workstation MIB

Tuxedo and WLE systems can require clients to run on a workstation for purposes of security, performance, and convenience. A network administrator can define the environment required to control workstation clients using the Workstation MIB. This MIB is an extension of the Tuxedo Core MIB and specifies the information required to control access to a Tuxedo or WLE application from multiple workstations.

The Tuxedo Workstation subsystem consists of a workstation clients (WSC) library, the workstation listener (WSL) executable, and the workstation handler (WSH) executable. The Workstation MIB specifies information about workstation listeners and workstation handlers. The following table lists the two WSL and WSH groups through which you can manage a workstation listener and its associated workstation handler processes.

The Workstation MIB consists of the following groups.

Group Name	Description
<code>tuxTwshTbl</code>	Workstation Handler
<code>tuxTws1Tbl</code>	Workstation Listener

You can define new workstation listeners in the `tuxTws1Tbl` group, and you can obtain information about active workstation handlers from the `tuxTwshTbl` group.

tuxTwshTbl

The tuxTwshTbl table represents runtime attributes of WSH client processes. These objects characterize workstation statistics specific to a particular WSH client process. Objects in this table are only accessible through a BEA SNMP agent installed on the local machine. Objects are only accessible when the corresponding WSH is active.

Variable Name	Object ID
tuxTwshTaClientId	.1.3.6.1.4.1.140.300.1.1.1.1
tuxTwshTaWshClientId	.1.3.6.1.4.1.140.300.1.1.1.2
tuxTwshTaSrvGrp	.1.3.6.1.4.1.140.300.1.1.1.3
tuxTwshTaSrvId	.1.3.6.1.4.1.140.300.1.1.1.4
tuxTwshTaGrpNo	.1.3.6.1.4.1.140.300.1.1.1.5
tuxTwshTaState	.1.3.6.1.4.1.140.300.1.1.1.6
tuxTwshTaLmid	.1.3.6.1.4.1.140.300.1.1.1.7
tuxTwshTaPid	.1.3.6.1.4.1.140.300.1.1.1.8
tuxTwshTaNaddr	.1.3.6.1.4.1.140.300.1.1.1.9
tuxTwshTaHwClients	.1.3.6.1.4.1.140.300.1.1.1.10
tuxTwshTaMultiplex	.1.3.6.1.4.1.140.300.1.1.1.11
tuxTwshTaCurClients	.1.3.6.1.4.1.140.300.1.1.1.12
tuxTwshTaTimeleft	.1.3.6.1.4.1.140.300.1.1.1.13
tuxTwshTaActive	.1.3.6.1.4.1.140.300.1.1.1.14
tuxTwshTaTotacttime	.1.3.6.1.4.1.140.300.1.1.1.15
tuxTwshTaTotidlttime	.1.3.6.1.4.1.140.300.1.1.1.16
tuxTwshTaCurwork	.1.3.6.1.4.1.140.300.1.1.1.17

Variable Name	Object ID
tuxTwshTaFlowcnt	.1.3.6.1.4.1.140.300.1.1.1.18
tuxTwshTaNumblockQ	.1.3.6.1.4.1.140.300.1.1.1.19
tuxTwshTaRcvdByt	.1.3.6.1.4.1.140.300.1.1.1.20
tuxTwshTaRcvdNum	.1.3.6.1.4.1.140.300.1.1.1.21
tuxTwshTaSentByt	.1.3.6.1.4.1.140.300.1.1.1.22
tuxTwshTaSentNum	.1.3.6.1.4.1.140.300.1.1.1.23

tuxTwshTaClientId

Syntax	<i>DisplayString</i> (SIZE (1..78))
Access	read-only
Description	Client identifier for this WSH. The data in this field should not be interpreted directly by the end user except for equality comparison.

tuxTwshTaWshClientId

Syntax	<i>DisplayString</i> (SIZE (1..78))
Access	read-only
Description	Client identifier for this WSH. The data in this field should not be interpreted directly by the end user except for equality comparison. Value is same as tuxTwshTaClientId.

tuxTwshTaSrvGrp

Syntax	<i>DisplayString</i> (SIZE (1..30))
Access	read-only
Description	Logical name of the server group for the associated WSL.

tuxTwshTaSrvId

Syntax	INTEGER (1..30001)
Access	read-only
Description	Unique (within the server group) server identification number for the associated WSL.

tuxTwshTaGrpNo

Syntax	INTEGER (1..30000)
Access	read-only
Description	Group number.

tuxTwshTaState

Syntax	INTEGER { active(1), suspended(2), dead(3) }
Access	read-write
Description	<p>State for the WSH client within the application. Any state defined for the <code>tuxTclientTbl</code> group can be returned or set. State changes to the <code>suspended(2)</code> state are transitive to all clients associated with this WSH as is the resetting of a <code>suspended(2)</code> WSH to <code>active(1)</code>. Additionally, <code>suspended(2)</code> WSH clients are not assigned any additional incoming clients by the WSL.</p> <p>Note that the state of a WSH client might not be set to <code>dead(3)</code> when accessing the <code>tuxTclientTbl</code> group. However, the state transition to <code>dead(3)</code> is allowed via the <code>tuxTwshTbl</code> group and results in all connections handled by the targeted WSH being dropped abortively.</p>

tuxTwshTaLmid

Syntax	<i>DisplayString</i> (SIZE (1..30))
Access	read-only
Description	Current logical machine on which the WSH is running.

tuxTwshTaPid

Syntax	INTEGER
Access	read-only
Description	Native operating system process identifier for the WSH client.

tuxTwshTaNaddr

Syntax	<i>DisplayString</i> (SIZE (1..78))
Access	read-only
Description	Network address of workstation handler. Hexadecimal addresses are converted to an ASCII format with a leading 0x.

tuxTwshTaHwClients

Syntax	INTEGER (1..32767)
Access	read-only
Description	High water number of clients accessing application through this WSH.

tuxTwshTaMultiplex

Syntax	INTEGER (1..32767)
Access	read-only
Description	Maximum number of clients that can access the application through this WSH.

tuxTwshTaCurClients

Syntax	INTEGER (1..32767)
Access	read-only
Description	Current number of clients accessing the application through this WSH.

tuxTwshTaTimeleft

Syntax	INTEGER
Access	read-only
Description	A non-0 value for this attribute indicates that the WSH has been assigned a newly connecting workstation client that has the indicated amount of time, in seconds, to complete the initialization process with the WSH.

tuxTwshTaActive

Syntax	INTEGER { yes(1), no(2), unknown(3) }
Access	read-only
Description	A value of yes(1) indicates that the WSH is currently performing work on behalf of one of its associated workstation clients. A value of no(2) indicates that the WSH is currently waiting for work to perform on behalf of one of its associated workstation clients.

tuxTwshTaTotacttime

Syntax	INTEGER
Access	read-only
Description	Time, in seconds, that the WSH has been active since it started processing.

tuxTwshTaTotidlttime

Syntax	INTEGER
Access	read-only
Description	Time, in seconds, that the WSH has been idle since it started processing.

tuxTwshTaCurwork

Syntax	INTEGER
Access	read-only
Description	Amount of work processed by this WSH since the last WSH assignment by the WSL. This value is used by the WSL to load balance new incoming connections among a set of WSH processes.

tuxTwshTaFlowcnt

Syntax	INTEGER
Access	read-only
Description	Number of times flow control has been encountered by this WSH. This attribute should be considered only in relation to recent past values because it might wrap around during the lifetime of the WSH.

tuxTwshTaNumblockQ

Syntax	INTEGER
Access	read-only
Description	Number of times this WSH has been unable to enqueue a message to a local UNIX system message queue due to queue blocking conditions. This attribute should be considered only in relation to recent past values because it might wrap around during the lifetime of the WSH.

tuxTwshTaRcvdByt

Syntax	INTEGER
Access	read-only
Description	Number of bytes received from the network by this WSH from all its present and past workstation clients. This attribute should be considered only in relation to recent past values because it might wrap around during the lifetime of the WSH.

tuxTwshTaRcvdNum

Syntax	INTEGER
Access	read-only
Description	Number of Tuxedo System/T messages received from the network by this WSH from all its present and past workstation clients. This attribute should be considered only in relation to recent past values because it might wrap around during the lifetime of the WSH.

tuxTwshTaSentByt

Syntax	INTEGER
Access	read-only
Description	Number of bytes sent to the network by this WSH to all its present and past workstation clients. This attribute should be considered only in relation to recent past values because it might wrap around during the lifetime of the WSH.

tuxTwshTaSentNum

Syntax	INTEGER
Access	read-only
Description	Number of Tuxedo System/T messages sent to the network by this WSH to all its present and past workstation clients. This attribute should be considered only in relation to recent past values because it might wrap around during the lifetime of the WSH.

tuxTwsITbl

The tuxTwsITbl table represents configuration and runtime attributes of WSL server processes configured to manage workstation groups. These attribute values identify and characterize workstation-specific configuration attributes for WSL tuxTsrvrTbl objects within the application. To create a new row in this table, use a SET request that specifies the values for at least tuxTwsITaSrvGrp, tuxTwsITaSrvId, and tuxTwsITaNaddr.

Variable Name	Object ID
tuxTwsITaSrvGrp	.1.3.6.1.4.1.140.300.1.2.1.1
tuxTwsITaSrvId	.1.3.6.1.4.1.140.300.1.2.1.2
tuxTwsITaGrpNo	.1.3.6.1.4.1.140.300.1.2.1.3
tuxTwsITaState	.1.3.6.1.4.1.140.300.1.2.1.4
tuxTwsITaLmid	.1.3.6.1.4.1.140.300.1.2.1.5
tuxTwsITaPid	.1.3.6.1.4.1.140.300.1.2.1.6
tuxTwsITaDevice	.1.3.6.1.4.1.140.300.1.2.1.7
tuxTwsITaNaddr	.1.3.6.1.4.1.140.300.1.2.1.8
tuxTwsITaWshName	.1.3.6.1.4.1.140.300.1.2.1.9
tuxTwsITaMinHandlers	.1.3.6.1.4.1.140.300.1.2.1.10
tuxTwsITaMaxHandlers	.1.3.6.1.4.1.140.300.1.2.1.11
tuxTwsITaMultiplex	.1.3.6.1.4.1.140.300.1.2.1.12
tuxTwsITaMaxIdleTime	.1.3.6.1.4.1.140.300.1.2.1.13
tuxTwsITaMaxInitTime	.1.3.6.1.4.1.140.300.1.2.1.14
tuxTwsITaClOpt	.1.3.6.1.4.1.140.300.1.2.1.15
tuxTwsITaEnvFile	.1.3.6.1.4.1.140.300.1.2.1.16
tuxTwsITaGrace	.1.3.6.1.4.1.140.300.1.2.1.17

Variable Name	Object ID
tuxTws1TaMaxGen	.1.3.6.1.4.1.140.300.1.2.1.18
tuxTws1TaRcmd	.1.3.6.1.4.1.140.300.1.2.1.19
tuxTws1TaRestart	.1.3.6.1.4.1.140.300.1.2.1.20
tuxTws1TaSequence	.1.3.6.1.4.1.140.300.1.2.1.21
tuxTws1TaCurHandlers	.1.3.6.1.4.1.140.300.1.2.1.22
tuxTws1TaHwHandlers	.1.3.6.1.4.1.140.300.1.2.1.23
tuxTws1TaWsProto	.1.3.6.1.4.1.140.300.1.2.1.24
tuxTws1TaSuspended	.1.3.6.1.4.1.140.300.1.2.1.25
tuxTws1TaViewRefresh	.1.3.6.1.4.1.140.300.1.2.1.26
tuxTws1TaKeepAlive	.1.3.6.1.4.1.140.300.1.2.1.28
tuxTws1TaNetTimeOut	.1.3.6.1.4.1.140.300.1.2.1.29

tuxTws1TaSrvGrp

- Syntax

DisplayString (*SIZE*(1..30))
- Access

read-write
- Description

Logical name of the server group. Server group names cannot contain an asterisk (*), comma, or colon.
- Note:

This object can be updated only during row creation.

tuxTws1TaSrvId

- Syntax

INTEGER (1..30001)
- Access

read-write
- Description

Unique (within the server group) server identification number.
- Note:

This object can be updated only during row creation.

tuxTwsITaGrpNo

Syntax	INTEGER (1..30001)
Access	read-only
Description	Group number associated with this servers group.

tuxTwsITaState

Syntax	INTEGER { active(1), inactive(2), migrating(3), cleaning(4), restarting(5), suspended(6), partitioned(7), dead(8), invalid(9) }
Access	read-write
Description	State for the WSL server within the application. Any state defined for the tuxTsrvrTbl group can be returned or set as indicated.

tuxTwsITaLmid

Syntax	DisplayString (SIZE (1..30))
Access	read-only
Description	Current logical machine on which the server is running.

tuxTwsITaPid

Syntax	INTEGER
Access	read-only
Description	Native operating system process identifier for the WSL server.

tuxTwsITaDevice

Syntax	DisplayString (SIZE (0..78))
Access	read-write
Description	Device name to be used by the WSL process to access the network. This is a required value for access to a network through a TLI-based Tuxedo System/T binary. This attribute is not needed for sockets-based Tuxedo System/T binaries.

tuxTwslTaNaddr

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-write
Description	<p>Specifies the complete network address to be used by the WSL process as its listening address. The listening address for a WSL is the means by which it is contacted by workstation client processes participating in the application.</p> <p>If string has the form 0xhex-digits or \\xhex-digits, it must contain an even number of valid hexadecimal digits. These forms are translated internally into a character array containing the hexadecimal representations of the string specified.</p>

tuxTwslTaWshName

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-write
Description	<p>The name of the executable that provides workstation handler services for this workstation listener. The default value for this is WSH, which corresponds to the system provided workstation handler. Workstation handlers can be customized using the command <code>buildwsh</code>.</p>

tuxTwslTaMinHandlers

Syntax	INTEGER (0..256)
Access	read-write
Description	<p>The minimum number of handlers that should be available in conjunction with this WSL at any given time. Upon being activated, the WSL starts this many WSHs immediately and does not deplete the supply of WSHs below this number until the administrator issues a shutdown to the WSL. Modifications to this attribute for a running WSL might cause additional handlers to be activated.</p>

tuxTwsITaMaxHandlers

Syntax	INTEGER (0..32767)
Access	read-write
Description	The maximum number of handlers that should be available in conjunction with this WSL at any given time. Handlers are started as necessary to meet the demand of workstation clients attempting to access the system. This attribute must be greater than or equal to the setting for the minimum number of handlers.

tuxTwsITaMultiplex

Syntax	INTEGER (0..32767)
Access	read-write
Description	Maximum number of clients that are supported by any one handler process concurrently.

tuxTwsITaMaxIdleTime

Syntax	INTEGER
Access	read-write
Description	Maximum amount of time, in minutes, that a workstation client is permitted to be idle before it is abortively disconnected from the application by the handler. A value of 0 allows clients to be idle as long as is necessary without being timed out.

tuxTwsITaMaxInitTime

Syntax	INTEGER
Access	read-write
Description	The minimum number of seconds that should be allowed for a workstation client to complete initialization processing through the WSH before being timed out by the WSL.

tuxTwsITaCLOpt

Syntax	<i>DisplayString</i> (SIZE (0..128))
Access	read-write
Description	Command-line options to be passed to the WSL server when it is activated. See the <i>servopts(5)</i> reference page for details. Note: Runtime modifications to this attribute do not affect a running WSL server. Server-specific options (that is, those after a double-dash "--") cannot be set and are not returned.

tuxTwsITaEnvFile

Syntax	<i>DisplayString</i> (SIZE (0..78))
Access	read-write
Description	WSL server-specific environment file. See <i>tuxTmachineEnvFile</i> for a complete discussion of how this file is used to modify the environment. Note: Runtime modifications to this attribute do not affect a running WSL server.

tuxTwsITaGrace

Syntax	INTEGER
Access	read-write
Description	The period of time, in seconds, during which the <i>tuxTwsITaMaxGen</i> limit applies. This attribute is meaningful only for restartable WSL servers, that is, if the <i>tuxTwsITaRestart</i> attribute is set to <i>yes(1)</i> . When a restarting server would exceed the <i>tuxTwsITaMaxGen</i> limit but the <i>tuxTwsITaGrace</i> period has expired, the system resets the current generation (<i>tuxTsrvrGeneration</i>) to 1 and resets the initial boot time (<i>tuxTsrvrTimeStart</i>) to the current time. A value of 0 for this attribute indicates that the WSL server should always be restarted.

tuxTwsITaMaxGen

Syntax	INTEGER (0..256)
Access	read-write
Description	Number of generations allowed for a restartable WSL server (<code>tuxTwsITaRestart == yes(1)</code>) over the specified grace period (<code>tuxTwsITaGrace</code>). The initial activation of the WSL server counts as one generation and each restart also counts as one. Processing after the maximum number of generations is exceeded is discussed above with respect to <code>tuxTwsITaGrace</code> .

tuxTwsITaRcmd

Syntax	<i>DisplayString</i> (SIZE(0..78))
Access	read-write
Description	Application specified command to be executed in parallel with the system restart of an application server. This command must be an executable file in the native operating system.

tuxTwsITaRestart

Syntax	INTEGER { yes(1), no(2) }
Access	read-write
Description	Restartable (<code>yes(1)</code>) or non-restartable (<code>no(2)</code>) WSL server. If server migration is specified for this server group (<code>tuxTdomainOptions = migrate(2)</code> and <code>tuxTgroupLMID</code> with alternate site), then this attribute must be set to <code>yes(1)</code> .

tuxTwsITaSequence

Syntax	INTEGER (1..10000)
Access	read-write
Description	Specifies when this server should be booted (<code>tmboot(1)</code>) or shutdown (<code>tmshutdown(1)</code>) relative to other servers. If two servers are given the same sequence number, it is possible for <code>tmboot(1)</code> to boot them in parallel and for <code>tmshutdown(1)</code> to shut them down in parallel. <code>tuxTwsITbl</code> instances added without a

`tuxTwsITaSequence` attribute specified or with an invalid value have a value generated for them that is 10,000 or more and is higher than any other automatically selected default value. Servers are booted by `tmboot(1)` in increasing order of sequence number and shutdown by `tmshutdown(1)` in decreasing order. Runtime modifications to this attribute affect only `tmboot(1)` and `tmshutdown(1)` and affect the order in which running servers can be shutdown by a subsequent invocation of `tmshutdown(1)`.

tuxTwsITaCurHandlers

Syntax	INTEGER
Access	read-only
Description	Number of currently active handlers associated with this WSL.

tuxTwsITaHwHandlers

Syntax	INTEGER
Access	read-only
Description	Maximum number of currently active handlers associated with this WSL at any one time.

tuxTwsITaWsProto

Syntax	INTEGER
Access	read-only
Description	The Tuxedo System/T/WS protocol version number for this /WS group. Note that /WS clients connecting to this group might themselves have a different protocol version number associated with them.

tuxTwsITaSuspended

Syntax	INTEGER { new(1), all(2), none(3) }
Access	read-write

Description A value of `new(1)` indicates that new incoming clients cannot connect through this `tuxTwsITbl` instance. A value of `all(2)` indicates that workstation clients already connected to the application through this WSL have been `suspended(2)` (see `tuxTclientState`) in addition to new incoming connections being disallowed. A value of `none(3)` indicates that no suspension characteristics are in effect.

tuxTwsITaViewRefresh

Syntax `INTEGER { yes(1), no-value-returned(2) }`

Access `read-write`

Description Setting a value of `yes(1)` causes all active WSHs in the /WS group to refresh their **VIEW** buffer type cache. A `GET` request on this object always returns `no-value-returned(2)` and does not mean anything. This object has meaning only for `SET` requests.

tuxTwsITaKeepAlive

Syntax `INTEGER { client(1), handler(2), both(3), none(4), not-available(5) }`

Access `read-write`

Description The network “keep alive” option is configured for the client, the handler, or both the client and the handler, or not on either side of the connection. Changing this value only affects future connections. This object is supported only on Tuxedo 6.4 or later.

tuxTwsITaNetTimeOut

Syntax `INTEGER (0..35204650)`

Access `read-write`

Description The minimum number of seconds that should be allowed for a workstation client to wait for a response from WSL/WSH. A value of 0 indicates no network time-out. Changing this value affects only future connections. This object is supported only on Tuxedo 6.4. -1 is returned if the object is not available.

8 Application Queue MIB

The Tuxedo and WLE systems incorporate the capability to use application queues for time-independent communication. The Tuxedo and WLE Application Queue MIB provides the administrative environment required for managing and controlling access to application queues. The Application Queue MIB defines the structure of the application queues.

In Tuxedo and WLE applications, messages are stored on a queue, and queues are defined within a particular queue space. Queueing and dequeuing is done within a transaction. The Application Queue MIB consists of five different groups for defining queue access, queues, messages, queues spaces, and queue transactions. The following table lists the groups for managing each of the queue components.

Group Name	Description
tuxTAppQctrl	Access control to application queues
tuxTAppQTbl	Application queues within a queue space
tuxTAppQmsgTbl	Messages within an application queue
tuxTQspaceTbl	Application queue spaces
tuxTQtransTbl	Transactions associated with application queues

tuxTAppQctrl

This is a control MIB to enable controlled access to all Application Queue related MIB groups.

Variable Name	Object ID
tuxTAppQctrlLmid	.1.3.6.1.4.1.140.300.12.5.1
tuxTAppQctrlQmConfig	.1.3.6.1.4.1.140.300.12.5.2
tuxTAppQctrlSpaceName	.1.3.6.1.4.1.140.300.12.5.3
tuxTAppQctrlQname	.1.3.6.1.4.1.140.300.12.5.4
tuxTAppQctrlMsgLoPrio	.1.3.6.1.4.1.140.300.12.5.5
tuxTAppQctrlMsgHiPrio	.1.3.6.1.4.1.140.300.12.5.6
tuxTAppQctrlMsgEndTime	.1.3.6.1.4.1.140.300.12.5.7
tuxTAppQctrlMsgStartTime	.1.3.6.1.4.1.140.300.12.5.8
tuxTAppQctrlMsgExpireEndTime	.1.3.6.1.4.1.140.300.12.5.20
tuxTAppQctrlMsgExpireStartTime	.1.3.6.1.4.1.140.300.12.5.30

tuxTAppQctrlLmid

Syntax	INTEGER { local(1), all(2) }
Access	read-write
Description	<p>This applies to all Application Queue related MIB groups. The value of this object controls the machines for which the values are returned.</p> <p>If the value is local(1), only the local host where the BEA SNMP Agent is running is considered; alternatively, all LMIDs known to the application are considered if the value is all(2).</p>

The default value of this object is `local(1)`.

tuxTAppQctrlQmConfig

Syntax *DisplayString* (SIZE(1..78))

Access read-write

Description This applies to all Application Queue related MIB groups. The value of this object controls the device for which the values are returned.

The default value of this object is “*”, in which case all known devices (which are a part of some group) are considered.

tuxTAppQctrlSpaceName

Syntax *DisplayString* (SIZE(1..15))

Access read-write

Description This applies to all Application Queue related MIB groups. The value of this object controls the queue space for which the values are returned.

The default value of this object is “*”, in which case all queue spaces for the devices (qualified by `tuxTAppQctrlQmConfig`) are considered.

tuxTAppQctrlQname

Syntax *DisplayString* (SIZE(1..15))

Access read-write

Description The value of this object controls the queue for which the values are returned. This applies to `tuxTAppQTbl` and `tuxTAppQmsgTbl`.

The default value of this object is “*”, in which case all queues for the devices (qualified by `tuxTAppQctrlQmConfig`) and queue spaces (qualified by `tuxTAppQctrlSpaceName`) are considered.

tuxTAppQctrlMsgLoPrio

Syntax	INTEGER
Access	read-write
Description	This object applies only to tuxTAppQmsgTbl. The lowest priority within which to search for occurrences of tuxTAppQmsgTbl instances. This is valid only for PRIO-based queues. By default, the minimum value of priority is considered. To revert to the default setting, set this object to 0.

tuxTAppQctrlMsgHiPrio

Syntax	INTEGER
Access	read-write
Description	This object applies only to tuxTAppQmsgTbl. The highest priority within which to search for occurrences of tuxTAppQmsgTbl instances. This is valid only for PRIO-based queues. By default, the maximum value of priority is considered. To revert to the default setting, set this object to 0.

tuxTAppQctrlMsgEndTime

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-write
Description	This object applies only to tuxTAppQmsgTbl. The end time within which to search for occurrences of tuxTAppQmsgTbl instances. The range is inclusive. This is valid only for TIME-based queues. The default value is the maximum number possible on that machine. To use the default setting, set this object to “*”.

YY[MM[DD[hh[mm[ss]]]]]

Specifies the year, month, date, hour, minute, and second respectively. Any value which is not specified defaults to its minimum value (e.g., 9506 is taken as 950601000000). The years 00 through 37 are treated as 2000 through 2037, 70 through 99 as 1970 through 1999, and 38 through 69 are invalid.

tuxTAppQctrlMsgStartTime

Syntax *DisplayString* (SIZE(1..15))

Access read-write

Description This object applies only to tuxTAppQmsgTbl. The start time within which to search for occurrences of tuxTAppQmsgTbl instances. The range is inclusive. This is valid only for TIME-based queues. By default, the minimum time value is considered to be 0. To use the default setting, set this object to “*”.

YY[MM[DD[hh[mm[ss]]]]]

Specifies the year, month, date, hour, minute, and second respectively. Any value which is not specified defaults to its minimum value (e.g., 9506 is taken as 950601000000). The years 00 through 37 are treated as 2000 through 2037, 70 through 99 as 1970 through 1999, and 38 through 69 are invalid.

tuxTAppQctrlMsgExpireEndTime

Syntax *DisplayString* (SIZE(1..12))

Access read-write

Description This object applies only to tuxTAppQmsgTbl. The expire end time within which to search for occurrences of tuxTAppQmsgTbl instances. The range is inclusive. This is valid only for TIME-based queues. The default value is the maximum number possible on that machine. To use the default setting, set this object to “*”.

YY[MM[DD[hh[mm[ss]]]]]

Specifies the year, month, date, hour, minute, and second respectively. Any value which is not specified defaults to its minimum value (e.g., 9506 is taken as 950601000000). The years 00 through 37 are treated as 2000 through 2037, 70 through 99 as 1970 through 1999, and 38 through 69 are invalid.

tuxTAppQctrlMsgExpireStartTime

Syntax *DisplayString* (SIZE(1..12))

Access read-write

Description This object applies only to `tuxTAppQmsgTbl`. The Expire start time within which to search for occurrences of `tuxTAppQmsgTbl` instances. The range is inclusive. This is valid only for TIME-based queues. By default, the minimum time value is considered to be 0. To use the default setting, set this object to “*”.

`YY[MM[DD[hh[mm[ss]]]]]`

Specifies the year, month, date, hour, minute, and second respectively. Any value which is not specified defaults to its minimum value (e.g., 9506 is taken as 950601000000). The years 00 through 37 are treated as 2000 through 2037, 70 through 99 as 1970 through 1999, and 38 through 69 are invalid.

tuxTAppQTbl

The tuxTAppQTbl group represents application queues. One or more application queues can exist in a single application queue space. Objects in this table are only accessible through a BEA SNMP Agent installed on the local machine.

Creation of a New Queue — For creating a new queue(row), in this group the SET request should have the value of tuxTAppQname, tuxTAppQspaceName, and tuxTAppQmConfig. Also the value of tuxTAppQgrpNo (which is a part of the index) should be the corresponding group number for that queue space or “40000” (if no such group exists).

Note: For this and all other Application Queue related MIB groups there is a control MIB which can be used to filter the data returned as a part of all Application Queue related MIB groups. Please refer to tuxTAppQctrl.

To create a new row in this table, issue a SET request that specifies at least the values for tuxTAppQname, tuxTAppQspaceName, and tuxTAppQmConfig.

Variable Name	Object ID
tuxTAppQname	.1.3.6.1.4.1.140.300.12.1.1.1
tuxTAppQspaceName	.1.3.6.1.4.1.140.300.12.1.1.2
tuxTAppQmConfig	.1.3.6.1.4.1.140.300.12.1.1.3
tuxTAppQlmid	.1.3.6.1.4.1.140.300.12.1.1.4
tuxTAppQgrpNo	.1.3.6.1.4.1.140.300.12.1.1.5
tuxTAppQstate	.1.3.6.1.4.1.140.300.12.1.1.6
tuxTAppQorder	.1.3.6.1.4.1.140.300.12.1.1.7
tuxTAppQcmd	.1.3.6.1.4.1.140.300.12.1.1.8
tuxTAppQcmdHw	.1.3.6.1.4.1.140.300.12.1.1.9
tuxTAppQcmdLw	.1.3.6.1.4.1.140.300.12.1.1.10
tuxTAppQmaxRetries	.1.3.6.1.4.1.140.300.12.1.1.11

Variable Name	Object ID
tuxTAppQoutOfOrder	.1.3.6.1.4.1.140.300.12.1.1.12
tuxTAppQretryDelay	.1.3.6.1.4.1.140.300.12.1.1.13
tuxTAppQcurBlocks	.1.3.6.1.4.1.140.300.12.1.1.14
tuxTAppQcurMsg	.1.3.6.1.4.1.140.300.12.1.1.15
tuxTAppQDefExpirationTime	.1.3.6.1.4.1.140.300.12.1.1.30
tuxTAppQDefDeliveryPolicy	.1.3.6.1.4.1.140.300.12.1.1.40
tuxTAppQCmdNonPersist	.1.3.6.1.4.1.140.300.12.1.1.50
tuxTAppQCmdNonPersistHw	.1.3.6.1.4.1.140.300.12.1.1.60
tuxTAppQCmdNonPersistLw	.1.3.6.1.4.1.140.300.12.1.1.70
tuxTAppQCurNonPersistBytes	.1.3.6.1.4.1.140.300.12.1.1.80
tuxTAppQCurNonPersistMsg	.1.3.6.1.4.1.140.300.12.1.1.90

tuxTAppQname

- Syntax *DisplayString* (SIZE(1..15))
- Access read-write
- Description Name of the application queue.

Note: This object can be updated only during row creation.

tuxTAppQspaceName

- Syntax *DisplayString* (SIZE(1..15))
- Access read-write
- Description Name of the application queue space containing the application queue.

Note: This object can be updated only during row creation.

tuxTAppQmConfig

Syntax *DisplayString* (SIZE(1..78))

Access read-write

Description Absolute pathname of the file or device where the application queue space is located.

Note: This object can be updated only during row creation.

tuxTAppQlmid

Syntax *DisplayString* (SIZE(1..30))

Access read-write

Description Identifier of the logical machine where the application queue space is located.

Note: This object can be updated only during row creation.

tuxTAppQgrpNo

Syntax INTEGER (1..29999)

Access read-write

Description Group number of any server group for which this queue is a resource manager, in other words that group's openinfo string tuxTgroupOpenInfo contains the device name and queue space name for this queue.

Note: This object can be updated only during row creation.

tuxTAppQstate

Syntax INTEGER { valid(1), invalid(2) }

Access read-write

Description The values for GET and SET operations are as follows:

GET: valid(1)

A GET operation retrieves information about the selected application queues. The following list describes the meaning of the `tuxTAppQstate` attribute returned in response to a GET request. States not listed are not returned.

`valid(1)`

The specified queue exists.

SET: `invalid(2)`

A SET operation changes characteristics of the selected application queue or creates a new queue. The following list describes the meaning of the `tuxTAppQstate` attribute returned by a SET request. States not listed cannot be set.

`invalid(2)`

Delete the specified queue. If the queue space has processes attached to it, the queue is not deleted. In addition, if the queue has messages in it, it is not deleted. Successful return leaves the object in the `invalid(2)` state.

tuxTAppQorder

Syntax *DisplayString* (SIZE(1..30))

Access read-write

Description The order in which messages in the queue are to be processed. Legal values are `PRIO` or `TIME`, followed by a comma, optionally followed by another occurrence of `PRIO` or `TIME`, followed by one of the values `LIFO` or `FIFO`. If neither `FIFO` nor `LIFO` is specified, `FIFO` is assumed. If nothing is specified when a queue is created, the default is `FIFO`. For example, these are some legal settings:

```
PRIO
PRIO,TIME,LIFO
TIME,PRIO,FIFO
TIME,FIFO
```

tuxTAppQcmd

Syntax *DisplayString* (SIZE(0..78))

Access read-write

Description The command to be automatically executed when the high water mark, `tuxTAppQcmdHw`, is reached. The command is re-executed when the high water mark is reached again after the low water mark, `tuxTAppQcmdLw`, has been reached.

tuxTAppQcmdHw

Syntax *DisplayString*

Access read-write

Description The high water mark. Refer to `tuxTAppQcmdLw` for further information.

tuxTAppQcmdLw

Syntax *DisplayString*

Access read-write

Description The low water marks that control the automatic execution of the command specified in the `tuxTAppQcmd` attribute. Each is an integer greater than or equal to zero optionally followed by one of the following keyletters. The keyletters must be consistent for `tuxTAppQcmdHw` and `tuxTAppQcmdLw`.

b

The high and low water marks pertain to the number of bytes used by messages in the queue.

B

The high and low water marks pertain to the number of blocks used by messages in the queue.

m

The high and low water marks pertain to the number of messages in the queue.

%

The high and low water marks are expressed in terms of a percentage of queue capacity.

For example, if `tuxTAppQcmdLw` is 50m and `tuxTAppQcmdHw` is 100m, then the command specified in `tuxTAppQcmd` is executed when 100 messages are on the queue, and it is not executed again until the queue is drained below 50 messages and is filled again to 100 messages.

tuxTAppQmaxRetries

Syntax	INTEGER
Access	read-write
Description	The maximum number of retries for a failed queue message. When the number of retries is exhausted, the message is placed on the error queue of the associated application queue space. If there is no error queue, the message is dropped. The default is zero.

tuxTAppQoutOfOrder

Syntax	INTEGER { none(1), top(2), msgid(3) }
Access	read-write
Description	The way in which out-of-order message processing is to be handled. The default is <code>none(1)</code> .

tuxTAppQretryDelay

Syntax	INTEGER
Access	read-write
Description	The delay, in seconds, between retries for a failed queue message. The default is zero.

tuxTAppQcurBlocks

Syntax	INTEGER
Access	read-only
Description	The number of disk pages currently consumed by the queue.

tuxTAppQcurMsg

Syntax	INTEGER
Access	read-only
Description	The number of messages currently in the queue.

tuxTAppQDefExpirationTime

Syntax	DisplayString
Access	read-write
Description	<p>This attribute specifies an expiration time for messages enqueued with no explicit expiration time. The expiration time can be either a relative expiration time or none. The relative expiration time is determined by associating a fixed amount of time with a message after the message arrives at the queue manager process. When a message's expiration time is reached and the message has not been dequeued or administratively deleted, all resources associated with the message are reclaimed by the system and statistics are updated. If a messages expires during a transaction, the expiration does not cause the transaction to fail. Messages that expire while being enqueued or dequeued within a transaction are removed from the queue when the transaction ends. There is no notification that the message has expired. If no default expiration time is specified for a queue, the message without an explicit expiration time does not expire. When the queue's expiration time is modified, the expiration times of messages that were in the queue before the modification are not changed.</p>

The format is `+seconds`, where `seconds` is the number of seconds allowed to lapse between the time that the queue manager successfully completes the operation and the time that the message is to expire. If `seconds` is set to zero (0), the message expires immediately.

The value of this attribute may also be set to the string "none." The `none` string indicates that messages enqueued to the queue with no explicit expiration time do not expire. You can change the expiration time for messages already in a queue with the `tuxTAppQmsgExpireTime` object of the `tuxTAppQmsgTbl` group.

tuxTAppQDefDeliveryPolicy

Syntax	INTEGER { persist (1), non-persist (2) }
Access	read-write
Description	This attribute specifies the default delivery policy for the queue when no delivery mode is specified for a message enqueued to the queue. When the value is “persist,” messages enqueued to the queue without an explicitly specified delivery mode are delivered using the persistent (disk-based) delivery method. When the value is non-persist, messages enqueued to the queue without an explicitly specified delivery method are delivered using the non-persistent (in memory) delivery method. When a queue’s default delivery policy is modified, the delivery quality of service of messages that are in the queue before the modification are not changed. If the queue being modified is the reply queue named for any messages currently in the queue space, the reply quality of service is not changed for those messages as a result of changing the default delivery policy of the queue.

For non-persistent delivery, if the memory area is exhausted or fragmented so that a message cannot be enqueued, the enqueueing operation fails, even if there is sufficient persistent storage for the message. Similarly, if the persistent storage area is exhausted or fragmented so that a message cannot be enqueued, the enqueueing operation fails, even if there is sufficient non-persistent storage for the message. If the `tuxTQspaceMemNonPersist` object of the `tuxTQspaceTbl` group is zero (0) for a queue space, no space is reserved for non -persistent messages. In such a case, any attempt to enqueue a non-persistent message fails. This type of failure results, for example, when no delivery quality of service has been specified for a message and the `tuxTAppQDefDeliveryPolicy` object for the target queue has been set to “non-persist.”

tuxTAppQCmdNonPersist

Syntax	DisplayString (Size(0..78))
Access	read-write
Description	This attribute specifies the command to be executed automatically when the high-water mark for non -persistent (memory-based delivery) messages, <code>tuxTAppQCmdNonPersistHw</code> , is reached. The command is re-executed when the high-water mark is reached again after the low-water mark for non-persistent (memory-based delivery) messages, <code>tuxTAppQCmdNonPersistLw</code> , has been reached.

tuxTAppQCmdNonPersistHw

Syntax	DisplayString
Access	read-write
Description	<p>These attributes specify the high- and low-water marks that control the automatic execution of the command specified in the <code>tuxTAppQCmdNonPersist</code> attribute. Each is an integer greater than or equal to zero, followed by one of the following keyletters. The keyletters must be consistent for <code>tuxTAppQCmdNonPersistHw</code> and <code>tuxTAppQCmdNonPersistLw</code>.</p> <p>b</p> <p>The high- and low-water marks are expressed as the number of bytes used by non-persistent (in-memory) messages in the queue.</p> <p>B</p> <p>The high- and low-water marks are expressed as the number of blocks used by non-persistent (in-memory) messages in the queue.</p> <p>%</p> <p>The high- and low-water marks are expressed as a percentage of the shared memory capacity reserved for non-persistent messages in the queue space used by the queue.</p> <p>The messages threshold type specified through the <code>tuxTAppQCmdHw</code> and <code>tuxTAppQCmdLw</code> attributes (when followed by an <code>m</code>) applies to all messages in a queue, including both persistent and non-persistent messages, and therefore is not available as a threshold type for <code>tuxTAppQCmdNonPersistHw</code> and <code>tuxTAppQCmdNonPersistLw</code>.</p>

tuxTAppQCmdNonPersistLw

Syntax	DisplayString
Access	read-write
Description	<p>These attributes specify the high- and low-water marks that control the automatic execution of the command specified in the <code>tuxTAppQCmdNonPersist</code> attribute. Each is an integer greater than or equal to zero, followed by one of the following keyletters. The keyletters must be consistent for <code>tuxTAppQCmdNonPersistHw</code> and <code>tuxTAppQCmdNonPersistLw</code>.</p>

b

The high- and low-water marks are expressed as the number of bytes used by non-persistent (in-memory) messages in the queue.

B

The high- and low-water marks are expressed as the number of blocks used by non-persistent (in-memory) messages in the queue.

%

The high- and low-water marks are expressed as a percentage of the shared memory capacity reserved for non-persistent messages in the queue space used by the queue.

The messages threshold type specified through the `tuxTAppQCmdHw` and `tuxTAppQCmdLw` attributes (when followed by an `m`) applies to all messages in a queue, including both persistent and non-persistent messages, and therefore is not available as a threshold type for `tuxTAppQCmdNonPersistHw` and `tuxTAppQCmdNonPersistLw`.

tuxTAppQCurNonPersistBytes

Syntax	Integer
Access	read-write
Description	This attribute specifies the number of shared memory bytes currently consumed by the non-persistent messages on the queue.

tuxTAppQCurNonPersistMsg

Syntax	Integer
Access	read-write
Description	This attribute specifies the number of non-persistent messages currently in the queue. To determine the total number of messages in the queue, add the value of <code>tuxTAppQCurMsg</code> to this value.

tuxTAppQmsgTbl

The `tuxTAppQmsgTbl` group represents messages stored in application queues. A message is not created by an administrator; instead, it comes into existence as a result of a call to `tpenqueue(3)`. A message can be destroyed either by a call to `tpdequeue(3)` or by an administrator. In addition, certain attributes of a message can be modified by an administrator. For example, an administrator can move a message from one queue to another queue within the same queue space or change its priority.

Objects in this table are only accessible through a BEA SNMP Agent installed on the local machine.

Variable Name	Object ID
<code>tuxTAppQmsgId</code>	.1.3.6.1.4.1.140.300.12.2.1.1
<code>tuxTAppQmsgSerNo</code>	.1.3.6.1.4.1.140.300.12.2.1.2
<code>tuxTAppQmsgGrpNo</code>	.1.3.6.1.4.1.140.300.12.2.1.3
<code>tuxTAppQmsgQname</code>	.1.3.6.1.4.1.140.300.12.2.1.4
<code>tuxTAppQmsgQmConfig</code>	.1.3.6.1.4.1.140.300.12.2.1.5
<code>tuxTAppQmsgQspaceName</code>	.1.3.6.1.4.1.140.300.12.2.1.6
<code>tuxTAppQmsgLmid</code>	.1.3.6.1.4.1.140.300.12.2.1.7
<code>tuxTAppQmsgState</code>	.1.3.6.1.4.1.140.300.12.2.1.8
<code>tuxTAppQmsgNewQname</code>	.1.3.6.1.4.1.140.300.12.2.1.9
<code>tuxTAppQmsgPrior</code>	.1.3.6.1.4.1.140.300.12.2.1.10
<code>tuxTAppQmsgTime</code>	.1.3.6.1.4.1.140.300.12.2.1.11
<code>tuxTAppQmsgCorId</code>	.1.3.6.1.4.1.140.300.12.2.1.12
<code>tuxTAppQmsgCurRetries</code>	.1.3.6.1.4.1.140.300.12.2.1.13
<code>tuxTAppQmsgSize</code>	.1.3.6.1.4.1.140.300.12.2.1.14
<code>tuxTAppQmsgExpireTime</code>	.1.3.6.1.4.1.140.300.12.2.1.20

Variable Name	Object ID
tuxTAppQmsgPersistent	.1.3.6.1.4.1.140.300.12.2.1.30
tuxTAppQmsgReplyPersistent	.1.3.6.1.4.1.140.300.12.2.1.40

tuxTAppQmsgId

Syntax	<i>DisplayString</i> (SIZE(1..32))
Access	read-only
Description	A unique identifier for the queue message, which can be used to select the message for GET or SET operations. No significance should be placed on this value beyond using it for equality comparisons.

tuxTAppQmsgSerNo

Syntax	INTEGER
Access	read-only
Description	A running number corresponding to tuxTAppQmsgId for the queue message, which is a part of the composite index of this table.

tuxTAppQmsgGrpNo

Syntax	INTEGER
Access	read-only
Description	Group number of any server group for which this queue is a resource manager, in other words that group's openinfo string tuxTgroupOpenInfo contains the device name and queue space name for this queue.

tuxTAppQmsgQname

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-only
Description	Name of the application queue in which the message is stored.

tuxTAppQmsgQmConfig

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Absolute pathname of the file or device where the application queue space for the queue containing this message is located.

tuxTAppQmsgQspaceName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-only
Description	Name of the application queue space containing the application queue in which this message is located.

tuxTAppQmsgLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Logical machine id for the machine on which the queue containing this message is located.

tuxTAppQmsgState

Syntax	INTEGER { valid(1), invalid(2) }
Access	read-write
Description	The values for GET and SET operations are as follows:

GET: valid(1)

A GET operation retrieves information about the selected messages. The following list describes the meaning of the `tuxTAppQmsgState` attribute returned in response to a GET request. States not listed are not returned.

valid(1)

The message exists.

SET: `invalid(2)`

A SET operation changes characteristics of the selected message. The following list describes the meaning of the `tuxTAppQmsgState` attribute returned by a SET request. States not listed cannot be set.

`invalid(2)`

The message is deleted from its queue space. The message must be in state `valid(1)` before attempting this operation. Successful return leaves the object in the `invalid(2)` state.

tuxTAppQmsgNewQname

Syntax	DisplayString (SIZE(1..15))
Access	read-write
Description	Name of the queue into which to move the selected message. This queue must be an existing queue in the same queue space. The message must be in state <code>valid(1)</code> for this operation to succeed. This attribute is not returned by a GET operation.

tuxTAppQmsgPrior

Syntax	INTEGER
Access	read-write
Description	The priority of the message. This attribute is valid only for PRIO-based queues. The value -1 is returned by a GET operation if the queue is not PRIO-based.

tuxTAppQmsgTime

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-write
Description	The time when the message is processed. This attribute is valid only for TIME-based queues. The empty string is returned by a GET operation if the queue is not TIME-based. The format is one of the following: <i>+seconds</i>

Specifies that the message is processed *seconds* in the future. The value zero specifies that the message should be processed immediately.

YY[MM[DD[hh[mm[ss]]]]]

Specifies the year, month, day, hour, minute, and second when the message should be processed. Omitted units default to their minimum possible values. For example, 9506 is equivalent to 950601000000. The years 00 through 37 are treated as 2000 through 2037, 70 through 99 are treated as 1970 through 1999, and 38 through 69 are invalid.

tuxTAppQmsgCorId

Syntax *DisplayString* (SIZE(0..32))

Access read-only

Description The correlation identifier for this message provided by the application in the `topenqueue(3)` request. The empty string indicates that a correlation identifier is not present.

tuxTAppQmsgCurRetries

Syntax INTEGER

Access read-only

Description The number of retries that have been attempted so far on this message.

tuxTAppQmsgSize

Syntax INTEGER

Access read-only

Description The size of the message, in bytes.

tuxTAppQmsgExpireTime

Syntax *DisplayString* (SIZE(1..15))

Access read-write

Description This attribute specifies the time at which a message expires (that is, the time at which the message should be removed from the queue if it has not already been dequeued or administratively deleted). When a message expires, all resources it uses are reclaimed by the system and statistics are updated. If a message expires during a transaction, the expiration does not cause the transaction to fail. Messages that expire while being enqueued or dequeued within a transaction are removed from the queue when the transaction ends. There is no notification that the message has expired. Expiration times cannot be added to messages enqueued by versions of the BEA Tuxedo system that do not support message expiration, even when the queue manager responsible for changing this value supports message expiration. Attempts to add an expiration time fail.

The empty string is returned by a GET operation if the expiration time is not set. The expiration time format is one of the following:

`+seconds`

Specifies that the message will be removed after the specified number of seconds. If the value of seconds is set to zero (0), the message is removed immediately from the queue. Relative expiration time is calculated on the basis of the time at which the MIB request arrives and has been processed by the corresponding queue manager.

`YY[MM[DD[hh]MM[SS]]]`

Specifies the year, month, day, hour, minute, and second when the message will be removed if it has not already been dequeued or administratively deleted. Omitted units default to their minimum possible values. For example, 9506 is equivalent to 950601000000. The years 00 through 37 are treated as 2000 through 2037, 70 through 99 are treated as 1970 through 1999, and 38 through 69 are invalid. An absolute expiration time is determined by the clock on the machine where the queue manager process resides.

`none`

Specifies that the message will never expire.

tuxTAppQmsgPersistent

Syntax INTEGER { yes (1), no (2) }

Access read-only

Description This read-only state is set to “no” for non-persistent messages and “yes” for persistent messages. It is the delivery quality of service for the message.

tuxTAppQmsgReplyPersistent

Syntax INTEGER { yes (1), no (2) }

Access read-only

Description This read-only state is set to “no” for non-persistent messages and “yes” for persistent messages. It is the delivery quality that replies to the message

tuxTQspaceTbl

This group represents application queue spaces. Objects in this table are only accessible through a BEA SNMP Agent installed on the local machine.

Note: The values returned by this MIB are controlled by `tuxTAppQctrl`. Refer to the description of the above group for details.

To create a new row in this table, a `SET` request should be issued with an index (`tuxTQspaceGrpNo`) of 40000. This is a reserved value for row creation in the table. The `SET` request also needs to specify values for at least `tuxTQspaceQmConfig`, `tuxTQspaceName`, `tuxTQspaceLmid`, `tuxTQspaceIpckey`, `tuxTQspaceMaxMsg`, `tuxTQspaceMaxPages`, `tuxTQspaceMaxProc`, `tuxTQspaceMaxQueues`, and `tuxTQspaceMaxTrans`. The newly created instance (row) is not visible until it is attached to some server group.

Variable Name	Object ID
tuxTQspaceName	.1.3.6.1.4.1.140.300.12.3.1.1
tuxTQspaceQmConfig	.1.3.6.1.4.1.140.300.12.3.1.2
tuxTQspaceLmid	.1.3.6.1.4.1.140.300.12.3.1.3
tuxTQspaceGrpNo	.1.3.6.1.4.1.140.300.12.3.1.4
tuxTQspaceState	.1.3.6.1.4.1.140.300.12.3.1.5
tuxTQspaceBlocking	.1.3.6.1.4.1.140.300.12.3.1.6
tuxTQspaceErrQname	.1.3.6.1.4.1.140.300.12.3.1.7
tuxTQspaceForceInit	.1.3.6.1.4.1.140.300.12.3.1.8
tuxTQspaceIpckey	.1.3.6.1.4.1.140.300.12.3.1.9
tuxTQspaceMaxMsg	.1.3.6.1.4.1.140.300.12.3.1.10
tuxTQspaceMaxPages	.1.3.6.1.4.1.140.300.12.3.1.11
tuxTQspaceMaxProc	.1.3.6.1.4.1.140.300.12.3.1.12

Variable Name	Object ID
tuxTQspaceMaxQueues	.1.3.6.1.4.1.140.300.12.3.1.13
tuxTQspaceMaxTrans	.1.3.6.1.4.1.140.300.12.3.1.14
tuxTQspaceCurExtent	.1.3.6.1.4.1.140.300.12.3.1.15
tuxTQspaceCurMsg	.1.3.6.1.4.1.140.300.12.3.1.16
tuxTQspaceCurProc	.1.3.6.1.4.1.140.300.12.3.1.17
tuxTQspaceCurQueues	.1.3.6.1.4.1.140.300.12.3.1.18
tuxTQspaceCurTrans	.1.3.6.1.4.1.140.300.12.3.1.19
tuxTQspaceHwMsg	.1.3.6.1.4.1.140.300.12.3.1.20
tuxTQspaceHwProc	.1.3.6.1.4.1.140.300.12.3.1.21
tuxTQspaceHwQueues	.1.3.6.1.4.1.140.300.12.3.1.22
tuxTQspaceHwTrans	.1.3.6.1.4.1.140.300.12.3.1.23
tuxTQspacePercentInit	.1.3.6.1.4.1.140.300.12.3.1.24
tuxTQspaceMaxActions	.1.3.6.1.4.1.140.300.12.3.1.40
tuxTQspaceMaxHandles	.1.3.6.1.4.1.140.300.12.3.1.50
tuxTQspaceMaxOwners	.1.3.6.1.4.1.140.300.12.3.1.60
tuxTQspaceMaxTmpQueues	.1.3.6.1.4.1.140.300.12.3.1.70
tuxTQspaceMaxCursors	.1.3.6.1.4.1.140.300.12.3.1.80
tuxTQspaceMemNonPersist	.1.3.6.1.4.1.140.300.12.3.1.90
tuxTQspaceMemFilters	.1.3.6.1.4.1.140.300.12.3.1.100
tuxTQspaceMemOverFlow	.1.3.6.1.4.1.140.300.12.3.1.110
tuxTQspaceMemSystemReserved	.1.3.6.1.4.1.140.300.12.3.1.120
tuxTQspaceMemTotalAllocated	.1.3.6.1.4.1.140.300.12.3.1.130
tuxTQspaceCurActions	.1.3.6.1.4.1.140.300.12.3.1.140

Variable Name	Object ID
tuxTQspaceCurHandles	.1.3.6.1.4.1.140.300.12.3.1.150
tuxTQspaceCurOwners	.1.3.6.1.4.1.140.300.12.3.1.160
tuxTQspaceCurTmpQueues	.1.3.6.1.4.1.140.300.12.3.1.170
tuxTQspaceCurCursors	.1.3.6.1.4.1.140.300.12.3.1.180
tuxTQspaceCurMemNonPersist	.1.3.6.1.4.1.140.300.12.3.1.190
tuxTQspaceCurMemFilters	.1.3.6.1.4.1.140.300.12.3.1.200
tuxTQspaceCurMemOverFlow	.1.3.6.1.4.1.140.300.12.3.1.210
tuxTQspaceHwActions	.1.3.6.1.4.1.140.300.12.3.1.220
tuxTQspaceHwHandles	.1.3.6.1.4.1.140.300.12.3.1.230
tuxTQspaceHwOwners	.1.3.6.1.4.1.140.300.12.3.1.240
tuxTQspaceHwTmpQueues	.1.3.6.1.4.1.140.300.12.3.1.250
tuxTQspaceHwCursors	.1.3.6.1.4.1.140.300.12.3.1.260
tuxTQspaceHwMemNonPersist	.1.3.6.1.4.1.140.300.12.3.1.270
tuxTQspaceHwMemFilters	.1.3.6.1.4.1.140.300.12.3.1.280
tuxTQspaceHwMemOverFlow	.1.3.6.1.4.1.140.300.12.3.1.290

tuxTQspaceName

Syntax *DisplayString* (SIZE(1..15))

Access read-write

Description Name of the application queue space.

Note: This object can be updated only during row creation.

tuxTQspaceQmConfig

Syntax *DisplayString* (SIZE(1..78))

Access read-write

Description Absolute pathname of the file or device where the application queue space is located.

Note: This object can be updated only during row creation.

tuxTQspaceLmid

Syntax *DisplayString* (SIZE(1..30))

Access read-write

Description Identifier of the logical machine where the application queue space is located.

Note: This object can be updated only during row creation.

tuxTQspaceGrpNo

Syntax INTEGER (1..29999)

Access read-write

Description Group number of any server group for which this queue space is a resource manager, in other words that group's openinfo string `tuxTgroupOpenInfo` contains the device name and queue space name for this queue space.

Note: This object can be updated only during row creation.

tuxTQspaceState

Syntax INTEGER { inactive(1), initializing(2), open(3), active(4), cleaning(5), invalid(6) }

Access read-write

Description The values for GET and SET operations are as follows:

GET: inactive(1) | initializing(2) | open(3) | active(4)

A GET operation retrieves information about the selected application queue space. The following list describes the meaning of the `tuxTQspaceState` attribute returned in response to a GET request. States not listed are not returned.

`inactive(1)`

The queue space exists; i.e., disk space for it has been reserved in a device and the space has been initialized (if requested or if necessary).

`initializing(2)`

Disk space for the queue space is currently being initialized.

`open(3)`

Shared memory and other IPC resources for the queue space have been allocated and initialized, but no processes are currently attached to the shared memory.

`active(4)`

Shared memory and other IPC resources for the queue space have been allocated and initialized, and at least one process is currently attached to the shared memory. These processes can be the queue servers (`TMS_QM`, `TMQUEUE`, and perhaps `TMQFORWARD`) associated with the queue space, or they can be administrative processes such as `qmadmin(1)`, or they can be processes associated with another application.

`SET: open(3) | cleaning(5) | invalid(6)`

A `SET` operation changes the selected application queue space or creates a new one. The following list describes the meaning of the `tuxTQspaceState` attribute returned by a `SET` request. States not listed cannot be set.

`open(3)`

Allocate and initialize shared memory and other IPC resources for the queue space. This is allowed only if the queue space is in the `inactive(1)` state.

`cleaning(5)`

Remove the shared memory and other IPC resources for the queue space. This is allowed only when the queue space is in the `active(4)` or `open(3)` state. Successful return leaves the object in the `inactive(1)` state.

`invalid(6)`

Delete the queue space. An error is reported if the state is `active(4)` or if messages exist on any queues in the queue space. Successful return leaves the object in the `invalid(6)` state.

tuxTQspaceBlocking

Syntax	INTEGER
Access	read-write
Description	The blocking factor used for disk space management of the queue space. The default when a new queue space is created is 16.

tuxTQspaceErrQname

Syntax	<i>DisplayString</i> (SIZE(0..15))
Access	read-write
Description	Name of the error queue associated with the queue space. If there is no error queue, an empty string is returned by a GET request.

tuxTQspaceForceInit

Syntax	INTEGER { yes(1), no(2) }
Access	read-write
Description	The value of this object determines whether or not to initialize disk pages on new extents for the queue space. The default is not to initialize. Depending on the device type (e.g., regular file or raw slice), initialization can occur even if not requested.

tuxTQspaceIpckey

Syntax	INTEGER (32769..262143)
Access	read-write
Description	The IPC key used to access queue space shared memory.

tuxTQspaceMaxMsg

Syntax	INTEGER
Access	read-write
Description	The maximum number of messages that the queue space can contain.

tuxTQspaceMaxPages

Syntax	INTEGER
Access	read-write
Description	The maximum number of disk pages for all queues in the queue space. Each time the <code>tuxTQspaceMaxPages</code> attribute is increased, a new extent is allocated (see <code>tuxTQspaceCurExtent</code>). It is not possible to decrease the number of pages by setting this attribute to a lower number; an error is reported in this case.

tuxTQspaceMaxProc

Syntax	INTEGER
Access	read-write
Description	The maximum number of processes that can attach to the queue space.

tuxTQspaceMaxQueues

Syntax	INTEGER
Access	read-write
Description	The maximum number of queues that the queue space can contain.

tuxTQspaceMaxTrans

Syntax	INTEGER
Access	read-write
Description	The maximum number of simultaneously active transactions allowed by the queue space.

tuxTQspaceCurExtent

Syntax	INTEGER
Access	read-only
Description	The current number of extents used by the queue space. The largest number allowed is 100. Each time the <code>tuxTQspaceMaxPages</code> attribute is increased, a new extent is allocated.

tuxTQspaceCurMsg

Syntax	INTEGER
Access	read-only
Description	The current number of messages in the queue space. This number can be determined only if the queue space is <code>open(3)</code> or <code>active(4)</code> , or if the queue space is newly created. If none of the conditions apply, the value -1 is returned.

tuxTQspaceCurProc

Syntax	INTEGER
Access	read-only
Description	The current number of processes accessing the queue space.

tuxTQspaceCurQueues

Syntax	INTEGER
Access	read-only
Description	The current number of queues existing in the queue space. This number can be determined only if the queue space is <code>open(3)</code> or <code>active(4)</code> , or if the queue space is newly created. If none of these conditions apply, the value -1 is returned.

tuxTQspaceCurTrans

Syntax	INTEGER
Access	read-only
Description	The current number of outstanding transactions involving the queue space.

tuxTQspaceHwMsg

Syntax	INTEGER
Access	read-only
Description	The highest number of messages in the queue space since the queue space was last opened. The number is reset to 0 when the queue space state is set to <code>cleaning(5)</code> .

tuxTQspaceHwProc

Syntax	INTEGER
Access	read-only
Description	The highest number of processes simultaneously attached to the queue space since the queue space was last opened. The number is reset to 0 when the queue space state is set to <code>cleaning(5)</code> .

tuxTQspaceHwQueues

Syntax	INTEGER
Access	read-only
Description	The highest number of queues existing in the queue space since the queue space was last opened. The number is reset to 0 when the queue space state is set to <code>cleaning(5)</code> .

tuxTQspaceHwTrans

Syntax	INTEGER
Access	read-only
Description	The highest number of outstanding transactions involving the queue space since the queue space was last opened. If the queue space is accessed by more than one application, this number reflects all applications — not just the application represented by the TUXCONFIG environment variable. The number is reset to 0 when the queue space state is set to <code>cleaning(5)</code> .

tuxTQspacePercentInit

Syntax	INTEGER (0..100)
Access	read-only
Description	The percentage (as an integer between 0 and 100 inclusive) of disk space that has been initialized for the queue space.

tuxTQspaceMaxActions

Syntax	INTEGER
Access	read-write
Description	This attribute specifies the number of additional actions that the Queuing Services component of the BEA Engine can handle concurrently. When a blocking operation is encountered and additional actions are available, the blocking operation is set aside until it can be satisfied. After setting aside the blocking operation, another operation request can be handled. When the blocking operation is completed, the action associated with the operation is made available for a subsequent operation. The system reserve actions are equivalent to the number of processes that can attach to a queue space, so that each queue manager process can have at least one blocking action. Beyond the system-reserved number of blocking actions, the administrator can configure the system to enable it to accommodate additional blocking actions beyond the reserve. An operation fails if a blocking operation is requested and cannot be immediately satisfied and there are no actions available.

tuxTQspaceMaxHandles

Syntax	INTEGER
Access	read-write
Description	<p>This attribute specifies the number of handles that users of the Queuing Services component of the BEA Engine can use concurrently. Objects manipulated by the queuing services API require handles to access the objects. When an object is opened by a call to the Queuing Services API, a new handle is created and returned to the user. When an object handle is closed, the handle is made available for subsequent open object operations. When the Queuing Services API is used by an application, the administrator must configure the system to accommodate the maximum number of handles that are opened concurrently. An operation fails if a user attempts to open a queuing services object and there are no handles available. Adjusting this value has no effect on BEA Tuxedo applications other than unnecessarily consuming shared memory resources.</p>

tuxTQspaceMaxOwners

Syntax	INTEGER
Access	read-write
Description	<p>This attribute specifies the number of additional BEA Engine authenticated users that can concurrently use Queuing Services resources. There is one owner record per user, regardless of the number of open handles for the user. When there are no open handles for a user, the owner record is made available to subsequent users. The system reserves a number of owners equivalent to the number of actions, so that each action can be initiated by a different owner. Beyond the system-reserved number of owners that can concurrently use queuing services resources, the administrator can configure the system to accommodate additional owners beyond the reserved number. An operation fails if a user attempts to open a handle when there currently are no open handles, and there are no owners available. Adjusting this value has no effect on BEA Tuxedo applications other than unnecessarily consuming shared memory resources. Adjusting this value has no effect on BEA Tuxedo applications other than unnecessarily consuming shared memory resources.</p>

tuxTQspaceMaxTmpQueues

Syntax	INTEGER
Access	read-write
Description	This attribute specifies the number of temporary queues that can be opened concurrently in the Queuing Services component of the BEA Engine. Temporary queues are used by dynamic, self-configuring applications and reduce the need for administrators to configure each queue used by an application. Messages enqueued to temporary queues are not persistent. When all handles to a temporary queue are closed, the temporary queue resources are made available for subsequent temporary queue creation. When the temporary queues are used by an application, the administrator must configure the system to accommodate the maximum number of temporary queues that are active concurrently. An open operation fails if a user attempts to open a temporary queue and there are no temporary queue resources available. This attribute specifies the number of additional BEA Engine authenticated users that can concurrently use Queuing Services

tuxTQspaceMaxCursors

Syntax	INTEGER
Access	read-write
Description	This attribute specifies the number of cursors that user of the Queuing Services component of the BEA Engine can use concurrently. CURsors are used to navigate a queue. When a cursor is destroyed, the cursor resources are made available for subsequent cursor creation operations. When the cursors are used by an application, the administrator must configure the system to accommodate the maximum number of cursors that are allocated concurrently. An operation fails if a user attempts to create a cursor and there are no cursor resources available. This attribute specifies the number of additional BEA Engine authenticated users that can concurrently use Queuing Services.

tuxTQspaceMemNonPersist

Syntax	DisplayString
Access	read-write

- Description** This attribute specifies the size of the area reserved in shared memory to hold non-persistent messages for all queues in the queue space. The memory size can be specified in bytes (b) or blocks (B). (The size of a block in this context is equivalent to the size of a disk block.)
- The [bB] suffix is optional and, if not specified, the default is blocks. Note that the number of bytes requested can be rounded up to the next internal data size. When read, the value is always the actual amount of memory allocated in bytes (b).
- All non-persistent messages in the specified queue space are permanently lost when this variable is successfully changed.
- If the variable for a queue space is zero (0), no queue space is reserved. for non-persistent messages. In this case, any attempt to enqueue a non-persistent message fails. This type of failure results, for example, when no delivery quality of service has been specified for a message and the `tuxTAppQDefDeliverPolicy` object of the `tuxTAppTbl` group for the target queue has been set to `NONPERSIST`. For non-persistent delivery, if the memory area is exhausted or fragmented so that a message cannot be enqueued, the enqueueing operation fails, even if there is sufficient persistent storage for the message. Similarly, if the persistent storage area is exhausted or fragmented so that a message cannot be enqueued, the enqueueing operation fails, even if there is sufficient non-persistent storage for the message.

tuxTQspaceMemFilters

- Syntax** INTEGER
- Access** read-write
- Description** This attribute specifies the size of the memory area to reserve in shared memory to hold the compiled representation of user-defined filters. The memory size is specified in bytes. Filters are used by the Queuing Services component of the BEA Engine for message selection in dequeuing and cursor operations. Filters can be specified using various grammars, but are compiled into an engine normal form and stored in shared memory. Filters are referenced by a handle that is returned when they are compiled. When a filter is destroyed, the memory used by the filter is made available for subsequent compiled filters. When the filters are defined by an application, the administrator must configure the system to accommodate the maximum number of filters that will be concurrently compiled. An operation fails if a user attempts to create a new filter and there is not enough memory allocated for the compiled version of the filter. Adjusting this value has no effect on BEA Tuxedo applications other than unnecessarily consuming shared memory resources.

tuxTQspaceMemOverFlow

Syntax	INTEGER Access
	read-write
Description	This attribute specifies the size of the memory area to reserve in shared memory to accommodate peak load situations where some or all of the allocated shared memory resources are exhausted. The memory size is specified in bytes. Additional objects are allocated from this additional memory on a first-come, first-served basis. When an object created in the additional memory is closed or destroyed, the memory is released for subsequent overflow situations. This additional memory space can yield more objects than the configured number, but there is no guarantee that additional memory is available for any particular object at any given point in time. Currently, only actions, handles, cursors, owners, temporary queues, timers, and filters use the overflow.

tuxTQspaceMemSystemReserved

Syntax	INTEGER Access
	read-only
Description	This attribute specifies the total amount of memory (in bytes) reserved from shared memory for queuing services system use.

tuxTQspaceMemTotalAllocated

Syntax	INTEGER
Access	read-only
Description	This attribute specifies the total amount of memory (in bytes) allocated from shared memory for all queuing services objects.

tuxTQspaceCurActions

Syntax	INTEGER
Access	read-only

Description This attribute specifies the current number of actions in use in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value -1 is returned.

tuxTQspaceCurHandles

Syntax INTEGER

Access read-only

Description This attribute specifies the current number of cursors in use in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value -1 is returned.

tuxTQspaceCurOwners

Syntax INTEGER

Access read-only

Description This attribute specifies the current number of owners in use in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value -1 is returned.

tuxTQspaceCurTmpQueues

Syntax INTEGER

Access read-only

Description This attribute specifies the current number of temporary queues in use in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value -1 is returned.

tuxTQspaceCurCursors

Syntax INTEGER

Access read-only

Description This attribute specifies the current number of cursors in use in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value **-1** is returned.

tuxTQspaceCurMemNonPersist

Syntax INTEGER

Access read-only

Description This attribute specifies the current amount of memory, in bytes, consumed by non-persistent messages in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value **-1** is returned.

tuxTQspaceCurMemFilters

Syntax INTEGER

Access read-only

Description This attribute specifies the current number of bytes in use for filters in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value **-1** is returned.

tuxTQspaceCurMemOverFlow

Syntax INTEGER

Access read-only

Description This attribute specifies the current number of bytes of overflow memory in use in the queue space. This number can be determined if the queue space is **OPEn** or **ACTive**, or if the queue space is newly created. If none of the conditions apply, the value **-1** is returned.

tuxTQspaceHwActions

Syntax INTEGER (0..100)

Access read-only

Description This attribute specifies the highest number of concurrent actions reached in the queue space since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQspaceHwHandles

Syntax INTEGER (0..100)

Access read-only

Description This attribute specifies the highest number of concurrent handles opened in the queue space since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQspaceHwOwners

Syntax INTEGER

Access read-only

Description This attribute specifies the highest number of concurrent owners reached in the queue space since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQspaceHwTmpQueues

Syntax INTEGER

Access read-only

Description This attribute specifies the highest number of concurrent temporary queues opened in the queue space since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQspaceHwCursors

Syntax INTEGER

Access read-only

Description This attribute specifies the highest number of concurrent cursors created in the queue space since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQspaceHwMemNonPersist

Syntax INTEGER

Access read-only

Description This attribute specifies the largest amount of memory in bytes consumed by non-persistent messages since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQspaceHwMemFilters

Syntax INTEGER

Access read-only

Description This attribute specifies the highest number of bytes used for filters in the queue space since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQspaceHwMemOverflow

Syntax INTEGER

Access read-only

Description This attribute specifies the highest number of bytes used in the overflow memory in the queue space since the queue space was last opened. The number is reset to 0 when the queue space is set to CLEaning.

tuxTQtransTbl

This group represents runtime attributes of transactions associated with application queue spaces. Objects in this table are only accessible through a BEA SNMP Agent installed on the local machine.

Variable Name	Object ID
tuxTQtransXid	.1.3.6.1.4.1.140.300.12.4.1.1
tuxTQtransIndx1	.1.3.6.1.4.1.140.300.12.4.1.2
tuxTQtransIndx2	.1.3.6.1.4.1.140.300.12.4.1.3
tuxTQtransIndx3	.1.3.6.1.4.1.140.300.12.4.1.4
tuxTQtransIndx4	.1.3.6.1.4.1.140.300.12.4.1.5
tuxTQtransIndx5	.1.3.6.1.4.1.140.300.12.4.1.6
tuxTQtransGrpNo	.1.3.6.1.4.1.140.300.12.4.1.7
tuxTQtranSpaceName	.1.3.6.1.4.1.140.300.12.4.1.8
tuxTQtransQmConfig	.1.3.6.1.4.1.140.300.12.4.1.9
tuxTQtransLmid	.1.3.6.1.4.1.140.300.12.4.1.10
tuxTQtransState	.1.3.6.1.4.1.140.300.12.4.1.11

tuxTQtransXid

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Transaction identifier as returned by <code>tx_info(3)</code> and mapped to a string representation. The data in this field should not be interpreted directly by the user except for equality comparison.

tuxTQtransIdx1

Syntax	INTEGER
Access	read-only
Description	An integer index for tuxTQtransTbl. This should not be interpreted by the user. It is used only for uniquely identifying a particular row in this table by the combination of Indx1 through Indx5.

tuxTQtransIdx2

Syntax	INTEGER
Access	read-only
Description	An integer index for tuxTQtransTbl. This should not be interpreted by the user. It is used only for uniquely identifying a particular row in this table by the combination of Indx1 through Indx5.

tuxTQtransIdx3

Syntax	INTEGER
Access	read-only
Description	An integer index for tuxTQtransTbl. This should not be interpreted by the user. It is used only for uniquely identifying a particular row in this table by the combination of Indx1 through Indx5.

tuxTQtransIdx4

Syntax	INTEGER
Access	read-only
Description	An integer index for tuxTQtransTbl. This should not be interpreted by the user. It is used only for uniquely identifying a particular row in this table by the combination of Indx1 through Indx5.

tuxTQtransIdx5

Syntax	INTEGER
Access	read-only
Description	An integer index for <code>tuxTQtransTbl</code> . This should not be interpreted by the user. It is used only for uniquely identifying a particular row in this table by the combination of <code>Idx1</code> through <code>Idx5</code> .

tuxTQtransGrpNo

Syntax	INTEGER
Access	read-only
Description	Group number of any server group for which the queue space concerning this transaction is a resource manager, in other words that group's <code>openinfo</code> string <code>tuxTgroupOpenInfo</code> contains the device name and queue space name for the queue space concerning this transaction.

tuxTQtranSpaceName

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-only
Description	Name of the application queue space associated with the transaction.

tuxTQtransQmConfig

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Absolute pathname of the file or device where the application queue space is located.

tuxTQtransLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	read-only
Description	Identifier of the logical machine where the application queue space is located.

tuxTQtransState

Syntax	INTEGER { active(1), abort-only(2), aborted(3), com-called(4), ready(5), decided(6), suspended(7), habort(8), hcommit(9) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: {actdive(1) abort-only(2) aborted(3) com-called(4) ready(5) decided(6) suspended(7) }</p> <p>A GET operation retrieves runtime information about the selected transactions. The following list describes the meaning of the tuxTQtransState attribute returned in response to a GET request. States not listed are not returned.</p> <p>active(1)</p> <p>The transaction is active.</p> <p>abort-only(2)</p> <p>The transaction has been identified for rollback.</p> <p>aborted(3)</p> <p>The transaction has been identified for rollback and rollback has been initiated.</p> <p>com-called(4)</p> <p>The initiator of the transaction has called tpccommit(3) and the first phase of two-phase commit has begun.</p> <p>ready(5)</p> <p>All of the participating groups on the retrieval site have successfully completed the first phase of the two-phase commit and are ready to be committed.</p>

`decided(6)`

The second phase of the two-phase commit has begun.

`suspended(7)`

The initiator of the transaction has suspended processing on the transaction.

`SET: {habort(8) | hcommit(9)}`

A SET operation updates the state of the selected transactions. The following list describes the meaning of the `tuxTQtransState` attribute returned by a SET request. States not listed cannot be set.

`habort(8)`

Heuristically abort the transaction. Successful return leaves the object in the `habort(8)` state.

`hcommit(9)`

Heuristically commit the transaction. Successful return leaves the object in the `hcommit(9)` state.

9 Event Broker MIB

There are two types of Tuxedo and WLE events: application events and system events. Application events are usually controlled or trapped by the application code. System events are generated by the Tuxedo or WLE run-time system when important changes in that system are detected. Application programs (clients or services) can subscribe to these system events.

The Event Broker MIB defines the characteristics of an event subscription. You can use the Event Broker MIB to obtain the characteristics of current event subscriptions, define new subscriptions, or invalidate subscriptions. To enable both system event and application event notification, you need to define the system event broker and the application event broker in the Tuxedo Core MIB.

Event subscriptions can be temporary or persistent. Persistent subscriptions survive across application activations and can be removed through the Event Broker MIB. The Tuxedo Event Broker MIB contains five groups of event subscriptions through which the Event Broker can be managed. The following table lists the event broker subscription groups.

Group Name	Description
<code>tuxEventClientTbl</code>	Subscriptions that trigger unsolicited notification
<code>tuxEventCmdTbl</code>	Subscriptions that trigger system commands
<code>tuxEventQueTbl</code>	Subscriptions for queue-based notification
<code>tuxEventSvcTbl</code>	Subscriptions for server-based notification
<code>tuxEventUlogTbl</code>	Subscriptions for writing userlog messages

Each object in these groups represents a single subscription request. Client Notifications (`tuxEventClientTbl` group) indicate which events trigger an unsolicited message to a client. Service Notifications (`tuxEventSvcTbl` group) indicate which events trigger a request to an application service. Application Queue Notifications (`tuxEventQueTbl` group) indicate which events send a message to an application queue. System Command Notifications (`tuxEventCmdTbl` group) indicate which events trigger an operating system command. Log File Notifications (`tuxEventUlogTbl` group) indicate which events generate a record in the central event log (ulog). The Event Broker automatically removes temporary subscriptions when it detects that the corresponding target is no longer active.

Event subscriptions and the ability to change the Tuxedo MIB enables system administrators and application designers to write event-adaptive applications. When a failure is detected through a system event notification, a management framework program can perform the corrective measures. For example, a management framework task can be triggered to activate servers on a backup machine when it receives an event notification about a failure on a primary machine.

tuxEventClientTbl

This represents a set of subscriptions registered with the Event Broker for client-based notification.

When an event is detected, it is compared to each `tuxEventClientTbl` instance. If the event name matches the value in the event expression and the optional filter rule is true, then the event buffer is sent to the specified client's unsolicited message handling routine. To create a new row in this table, it is necessary to issue a `SET` request that at least specifies the values for `tuxEventClientExpr` and `tuxEventClientId`.

Variable Name	Object ID
<code>tuxEventClientIndx</code>	.1.3.6.1.4.1.140.300.2.1.1.1.1
<code>tuxEventClientExpr</code>	.1.3.6.1.4.1.140.300.2.1.1.1.2
<code>tuxEventClientFilter</code>	.1.3.6.1.4.1.140.300.2.1.1.1.3
<code>tuxEventClientState</code>	.1.3.6.1.4.1.140.300.2.1.1.1.4
<code>tuxEventClientId</code>	.1.3.6.1.4.1.140.300.2.1.1.1.5

tuxEventClientIndx

Syntax	INTEGER
Access	read-only
Description	A running number as the unique identifier for a row in the table.

tuxEventClientExpr

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-only
Description	Event pattern expression. This expression, in <code>recomp(3)</code> format, controls which event names match this subscription.

Note: This object can be updated only during row creation.

tuxEventClientFilter

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-only
Description	Event filter expression. This expression, if present, is evaluated with respect to the posted buffer's contents. It must evaluate to TRUE or this subscription is not matched. If the value of this is "-", it means that the filter expression is in binary format.

Note: This object can be updated only during row creation.

tuxEventClientState

Syntax	INTEGER { active(1), invalid(2) }
Access	read-write
Description	The values for GET and SET operations are as follows: <div style="margin-left: 20px;"> GET: active(1) A GET operation retrieves configuration information for the matching tuxEventClientTbl row(s). SET: invalid(2) A SET operation updates configuration information for the row in tuxEventClientTbl. The following state indicates the meaning of a tuxEventClientState set in a SET request. States not listed cannot be set. invalid(2) Delete row. Successful return leaves the row in the invalid(2) state. </div>

tuxEventClientId

Syntax	<i>DisplayString</i> (SIZE(1..78))
Access	read-only
Description	Send an unsolicited notification message to this client when a matching event is detected.

Note: This object can be updated only during row creation.

tuxEventCmdTbl

This represents a set of subscriptions registered with the Event Broker that trigger execution of system commands.

When an event is detected, it is compared to each row in this table. If the event name matches the value in the event expression and the optional filter rule is true, then the event buffer is formatted and passed to the system’s command interpreter.

Create a new Row: To create a new instance of tuxEventCmdTbl the user must specify at least tuxEventCmdExpr and tuxEventCmd. All objects except tuxEventCmdState can be updated only during creation of a new instance.

Variable Name	Object ID
tuxEventCmdIndx	.1.3.6.1.4.1.140.300.2.2.1.1.1
tuxEventCmdExpr	.1.3.6.1.4.1.140.300.2.2.1.1.2
tuxEventCmdFilter	.1.3.6.1.4.1.140.300.2.2.1.1.3
tuxEventCmdState	.1.3.6.1.4.1.140.300.2.2.1.1.4
tuxEventCmd	.1.3.6.1.4.1.140.300.2.2.1.1.5

tuxEventCmdIndx

Syntax	INTEGER
Access	read-only
Description	A running number as the unique identifier for a row in the table.

tuxEventCmdExpr

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-write
Description	Event pattern expression. This expression, in <i>recomp</i> (3) format, controls which event names match this subscription.

Note: This object can be updated only during row creation.

tuxEventCmdFilter

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-write
Description	Event filter expression. This expression, if present, is evaluated with respect to the posted buffer's contents. It must evaluate to TRUE or this subscription is not matched. If the value of the filter is "-", it means that the filter is in a binary format.

Note: This object can be updated only during row creation.

tuxEventCmdState

Syntax	INTEGER { active(1), invalid(2) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: active(1)</p> <p>A GET operation retrieves configuration information for the tuxEventCmdTbl instance(s).</p> <p>SET: invalid(2)</p> <p>A SET operation updates configuration information for the tuxEventCmdTbl instance. The following state indicates the meaning of a tuxEventCmdState set in a SET request. States not listed cannot be set.</p> <p>invalid(2)</p>

Delete tuxEventCmdTbl instance. Successful return leaves the object in the invalid(2) state.

tuxEventCmd

Syntax *DisplayString* (SIZE(1..255))

Access read-write

Description Execute this system command when an event matching this object is detected. For UNIX system platforms, the command is executed in the background using `system(3)`.

Note: This object can be updated only during row creation.

tuxEventQueTbl

This represents a set of subscriptions registered with the Event Broker for queue-based notification.

When an event is detected, it is compared to each tuxEventQueTbl instance. If the event name matches the value in the event expression and the optional filter rule is true, then the event buffer is stored in the specified reliable queue. To create a new row in this table, it is necessary to issue a SET request that at least specifies tuxEventQueExpr, tuxEventQspace, and tuxEventQname.

Variable Name	Object ID
tuxEventQueIndx	.1.3.6.1.4.1.140.300.2.3.1.1.1
tuxEventQueExpr	.1.3.6.1.4.1.140.300.2.3.1.1.2
tuxEventQueFilter	.1.3.6.1.4.1.140.300.2.3.1.1.3
tuxEventQueState	.1.3.6.1.4.1.140.300.2.3.1.1.4
tuxEventQspace	.1.3.6.1.4.1.140.300.2.3.1.1.5
tuxEventQname	.1.3.6.1.4.1.140.300.2.3.1.1.6
tuxEventQctlQtop	.1.3.6.1.4.1.140.300.2.3.1.1.7
tuxEventQctlBeforeMsgid	.1.3.6.1.4.1.140.300.2.3.1.1.8
tuxEventQctlQtimeAbs	.1.3.6.1.4.1.140.300.2.3.1.1.9
tuxEventQctlQtimeRel	.1.3.6.1.4.1.140.300.2.3.1.1.10
tuxEventQctlDeqTime	.1.3.6.1.4.1.140.300.2.3.1.1.11
tuxEventQctlPrior	.1.3.6.1.4.1.140.300.2.3.1.1.12
tuxEventQctlMsgId	.1.3.6.1.4.1.140.300.2.3.1.1.13
tuxEventQctlCorrId	.1.3.6.1.4.1.140.300.2.3.1.1.14
tuxEventQctlReplyQ	.1.3.6.1.4.1.140.300.2.3.1.1.15

Variable Name	Object ID
tuxEventQctlFailQ	.1.3.6.1.4.1.140.300.2.3.1.1.16
tuxEventPersist	.1.3.6.1.4.1.140.300.2.3.1.1.17
tuxEventTran	.1.3.6.1.4.1.140.300.2.3.1.1.18

tuxEventQueIndx

Syntax	INTEGER
Access	read-only
Description	Running number which is the unique identifier for an event in this table.

tuxEventQueExpr

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-write
Description	Event pattern expression. This expression, in <code>recomp(3)</code> format, controls which event names match this subscription.

Note: This object can be updated only during row creation.

tuxEventQueFilter

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-write
Description	Event filter expression. This expression, if present, is evaluated with respect to the posted buffer’s contents. It must evaluate to TRUE or this subscription is not matched. If the value of this object is “-”, it means the filter is in binary format.

Note: This object can be updated only during row creation.

tuxEventQueState

Syntax	INTEGER { active(1), invalid(2) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: active(1)</p> <p>A GET operation retrieves configuration information for the matching tuxEventQueTbl row(s).</p> <p>SET: invalid(2)</p> <p>A SET operation updates configuration information for the tuxEventQueTbl instance. The following state indicates the meaning of a tuxEventQueState set in a SET request. States not listed cannot be set.</p> <p>invalid(2)</p> <p>Delete tuxEventQueTbl row. Successful return leaves the object in the invalid(2) state.</p>

tuxEventQspace

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-write
Description	<p>Enqueue a notification message to a reliable queue in this queue space when a matching event is detected.</p> <p>Note: This object can be updated only during row creation.</p>

tuxEventQname

Syntax	<i>DisplayString</i> (SIZE(1..15))
Access	read-write
Description	<p>Enqueue a notification message to this reliable queue when a matching event is detected.</p> <p>Note: This object can be updated only during row creation.</p>

tuxEventQctlQtop

Syntax	INTEGER
Access	read-write
Description	This value, if present, is passed in to <code>topenqueue(3)</code> 's TPQCTL control structure to request notification via the /Q subsystem with the message to be placed at the top of the queue.
Note: This object can be updated only during row creation.	

tuxEventQctlBeforeMsgid

Syntax	INTEGER
Access	read-write
Description	This value, if present, is passed in to <code>topenqueue(3)</code> 's TPQCTL control structure to request notification via the /Q subsystem with the message to be placed on the queue ahead of the specified message.
Note: This object can be updated only during row creation.	

tuxEventQctlQtimeAbs

Syntax	INTEGER
Access	read-write
Description	This value, if present, is passed in to <code>topenqueue(3)</code> 's TPQCTL control structure to request notification via the /Q subsystem with the message to be processed at the specified time.
Note: This object can be updated only during row creation.	

tuxEventOctlQtimeRel

Syntax	INTEGER
Access	read-write
Description	This value, if present, is passed in to <code>tpenqueue(3)</code> 's TPQCTL control structure to request notification via the /Q subsystem with the message to be processed relative to the dequeue time.

Note: This object can be updated only during row creation.

tuxEventOctlDeqTime

Syntax	INTEGER
Access	read-write
Description	This value, if present, is passed in to <code>tpenqueue(3)</code> 's TPQCTL control structure.

Note: This object can be updated only during row creation.

tuxEventOctlPrior

Syntax	INTEGER
Access	read-write
Description	This value, if present, is passed in to <code>tpenqueue(3)</code> 's TPQCTL control structure.

Note: This object can be updated only during row creation.

tuxEventOctlMsgId

Syntax	<i>DisplayString</i> (SIZE(1..31))
Access	read-write
Description	This value, if present, is passed in to <code>tpenqueue(3)</code> 's TPQCTL control structure.

Note: This object can be updated only during row creation.

tuxEventQctlCorrId

Syntax *DisplayString* (SIZE(1..31))

Access read-write

Description This value, if present, is passed in to `topenqueue(3)`'s TPQCTL control structure.

Note: This object can be updated only during row creation.

tuxEventQctlReplyQ

Syntax *DisplayString* (SIZE(1..15))

Access read-write

Description This value, if present, is passed in to `topenqueue(3)`'s TPQCTL control structure.

Note: This object can be updated only during row creation.

tuxEventQctlFailQ

Syntax *DisplayString* (SIZE(1..15))

Access read-write

Description This value, if present, is passed in to `topenqueue(3)`'s TPQCTL control structure.

Note: This object can be updated only during row creation.

tuxEventPersist

Syntax INTEGER

Access read-write

Description If non-zero, do not cancel this subscription if the designated queue is no longer available.

Note: This object can be updated only during row creation.

tuxEventTran

Syntax	INTEGER
Access	read-write
Description	If non-zero and the client's <code>tpost(3)</code> call is transactional, include the <code>tpenqueue(3)</code> call in the client's transaction.

Note: This object can be updated only during row creation.

tuxEventSvcTbl

This represents a set of subscriptions registered with the Event Broker for service-based notification.

When an event is detected, it is compared to each tuxEventSvcTbl instance. If the event name matches the value in the event expression and the optional filter rule is true, then the event buffer is sent to the specified Tuxedo service routine.

To create a new row in this table, a SET request must be issued that specifies values for at least tuxEventSvcExpr and tuxEventSvcName.

Variable Name	Object ID
tuxEventSvcIndx	.1.3.6.1.4.1.140.300.2.4.1.1.1
tuxEventSvcExpr	.1.3.6.1.4.1.140.300.2.4.1.1.2
tuxEventSvcFilter	.1.3.6.1.4.1.140.300.2.4.1.1.3
tuxEventSvcState	.1.3.6.1.4.1.140.300.2.4.1.1.4
tuxEventSvcName	.1.3.6.1.4.1.140.300.2.4.1.1.5
tuxEventSvcPersist	.1.3.6.1.4.1.140.300.2.4.1.1.6
tuxEventSvcTran	.1.3.6.1.4.1.140.300.2.4.1.1.7

tuxEventSvcIndx

- Syntax

INTEGER
- Access

read-only
- Description

A running number which is a unique key for a row in this table.

tuxEventSvcExpr

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-only
Description	Event pattern expression. This expression, in <i>recomp</i> (3) format, controls which event names match this subscription.

Note: This object can be updated only during row creation.

tuxEventSvcFilter

Syntax	<i>DisplayString</i> (SIZE(1..255))
Access	read-only
Description	Event filter expression. This expression, if present, is evaluated with respect to the posted buffer's contents. It must evaluate to TRUE or this subscription is not matched. If this is "-", it means the filter is in binary format.

Note: This object can be updated only during row creation.

tuxEventSvcState

Syntax	INTEGER { active(1), invalid(2) }
Access	read-write
Description	<p>The values for GET and SET operations are as follows:</p> <p>GET: active(1)</p> <p>A GET operation retrieves configuration information for the matching tuxEventSvcTbl instance(s).</p> <p>SET: invalid(2)</p> <p>A SET operation updates configuration information for the tuxEventSvcTbl instance. The following state indicates the meaning of a tuxEventSvcState set in a SET request. States not listed cannot be set.</p> <p>invalid(2)</p>

Delete tuxEventSvcTbl row. Successful return leaves the object in the invalid(2) state.

tuxEventSvcName

Syntax *DisplayString* (SIZE(1..15))

Access read-only

Description Call this Tuxedo service when a matching event is detected.

Note: This object can be updated only during row creation.

tuxEventSvcPersist

Syntax INTEGER

Access read-write

Description If non-zero, do not cancel this subscription if the tuxEventSvcName service is no longer available.

Note: This object can be updated only during row creation.

tuxEventSvcTran

Syntax INTEGER

Access read-write

Description If non-zero and the client's tppost(3) call is transactional, include the tuxEventSvcName service call in the client's transaction.

Note: This object can be updated only during row creation.

tuxEventUlogTbl

This represents a set of subscriptions registered with the Event Broker for writing system `userlog(3)` messages.

When an event is detected, it is compared to each `tuxEventUlogTbl` instance. If the event name matches the value in the event expression and the optional filter rule is true, then the event buffer is formatted and passed to the Tuxedo `userlog(3)` function.

Create a new Row: To create a new instance of `tuxEventUlogTbl` the user must at least specify values for `tuxEventUlogExpr` and `tuxEventUserlog`. All objects except `tuxEventUlogState` can be updated only during creation of a new instance.

Variable Name	Object ID
tuxEventUlogIndx	.1.3.6.1.4.1.140.300.2.5.1.1.1
tuxEventUlogExpr	.1.3.6.1.4.1.140.300.2.5.1.1.2
tuxEventUlogFilter	.1.3.6.1.4.1.140.300.2.5.1.1.3
tuxEventUlogState	.1.3.6.1.4.1.140.300.2.5.1.1.4
tuxEventUserlog	.1.3.6.1.4.1.140.300.2.5.1.1.5

tuxEventUlogIndx

Syntax	INTEGER
Access	read-only
Description	A running number which is a unique key in this table.

tuxEventUlogExpr

Syntax *DisplayString* (SIZE(1..255))

Access read-write

Description Event pattern expression. This expression, in `recomp(3)` format, controls which event names match this subscription.

Note: This object can be updated only during row creation.

tuxEventUlogFilter

Syntax *DisplayString* (SIZE(1..255))

Access read-write

Description Event filter expression. This expression, if present, is evaluated with respect to the posted buffer's contents. It must evaluate to TRUE or this subscription is not matched. If this is "-", it means the filter is in binary form.

Note: This object can be updated only during row creation.

tuxEventUlogState

Syntax INTEGER { active(1), invalid(2) }

Access read-write

Description The values for GET and SET operations are as follows:

GET: active(1)

A GET operation retrieves configuration information for the matching tuxEventUlogTbl instance(s).

SET: invalid(2)

A SET operation updates configuration information for the tuxEventUlogTbl instance. The following state indicates the meaning of a tuxEventUlogState set in a SET request. States not listed cannot be set.

invalid(2)

Delete `tuxEventUlogTbl` row. Successful return leaves the object in the `invalid(2)` state.

tuxEventUserlog

Syntax *DisplayString*(SIZE(1..255))

Access read-write

Description Write a `userlog(3)` message when a matching event is detected.

Note: This object can be updated only during row creation.

10 Tuxedo Traps MIB

The event monitor feature of the Tuxedo and WLE systems detects and reports certain predefined events—primarily failures of which a system operator should be aware. The BEA SNMP Agent on the master node subscribes to all system events and generates a corresponding SNMP trap notification whenever any of these events occur. The enterprise ID used for these traps is `.1.3.6.1.4.1.140.tuxedo`, where `tuxedo` is 300. For the BEA SNMP Agent to receive Tuxedo or WLE system events, the Tuxedo system Event Broker (`TMSYSEVT`) must be running because that is the entity that generates the system events.

The Event Traps MIB defines all the traps that are generated and the objects that are passed in the variable bindings for these traps. The cause and recommended action for each event is described in the following sections.

Specific Trap Number

Each enterprise-specific trap notification generated by the BEA SNMP Agent has a value in the specific trap ID field of the SNMP trap packet that identifies the Tuxedo or WLE event. For each trap listed in this chapter, “Trap ID” is the specific trap number that is sent in the trap packet.

Variable Bindings

SNMP trap notifications generated by the BEA SNMP Agent contain 12 variables (attribute/value pairs) in the variable bindings of the trap packet:

`beaEventsDomainId`

This is the ID of the domain that generated the Tuxedo or WLE event notification.

`beaEventsIpckey`

This is the IPC key of the Tuxedo or WLE domain.

`beaLogicalAgentName`

The logical agent name of the BEA SNMP Agent generating the trap. The executable name is the default logical agent name.

The `tuxEventTrapVars` group contains all objects that are sent as a part of the variable bindings of the traps generated in relation to Tuxedo or WLE system events, as defined in EVENTS.

tuxEventsName

Syntax	<i>DisplayString</i>
Access	not-accessible
Description	A string that uniquely identifies this event. All system-generated events begin with <code>.Sys.</code>

tuxEventsSeverity

Syntax	INTEGER { Error(1), Warn(2), Infor(3) }
Access	not-accessible
Description	Indicates the severity of the system event.

tuxEventsLmid

Syntax	<i>DisplayString</i> (SIZE(1..30))
Access	not-accessible
Description	A string that identifies the machine where the event was detected.

tuxEventsTime

Syntax	INTEGER
Access	not-accessible
Description	A long integer containing the event detection time, in seconds, according to the clock on the machine where detection took place.

tuxEventsUsec

Syntax	INTEGER
Access	not-accessible
Description	A long integer containing the event detection time, in microseconds, according to the clock on the machine where detection took place. While the units of this value are always microseconds, the actual resolution depends on the underlying operating system and hardware.

tuxEventsDescription

Syntax	<i>DisplayString</i>
Access	not-accessible
Description	A one-line string summarizing the event.

tuxEventsClass

Syntax	<i>DisplayString</i>
Access	not-accessible
Description	The class of the object associated with the event. Depending on TA_CLASS, the event notification buffer can contain additional fields specific to an object of this class.

tuxEventsUlogCat

Syntax	<i>DisplayString</i>
Access	not-accessible
Description	Catalog name from which the message was derived, if any.

tuxEventsUlogMsgNum

Syntax	INTEGER
Access	not-accessible
Description	Catalog message number, if the message was derived from a catalog.

tuxTdomainID

Syntax	<i>DisplayString</i> (SIZE (0..30))
Access	not-accessible
Description	Domain identification string. Refer to Chapter 2, “Tuxedo Core MIB.”

tuxTdomainKey

Syntax	INTEGER (32769 .. 262143)
Access	not-accessible
Description	Numeric key for the well-known address in a Tuxedo System/T bulletin board. In a single processor environment, this key “names” the bulletin board. In a multiple processor or LAN environment, this key names the message queue of the DBBL. In addition, this key is used as a basis for deriving the names of resources other than the well-known address, such as the names for bulletin boards throughout the application. Refer to Chapter 2, “Tuxedo Core MIB.”

beaLogicalAgentName

Syntax	<i>DisplayString</i>
Access	not-accessible
Description	<p>The logical name of the agent as provided in the <code>-l</code> option (service name in case of Windows NT) when the agent was started. This is the agent that is monitoring this domain. If there are multiple SNMP agents running on a managed node, this name needs to be appended to the community with an <code>@</code> sign to get the MIB values from the appropriate agent. For example, if there are two logical agents, <code>simp_snmpd</code> and <code>bank_snmpd</code>, the communities used to query values from these agents would be <code>public@simp_snmpd</code> and <code>public@bank_snmpd</code> respectively. The component after the <code>@</code> sign is used to identify the agent to which the MIB query is to be sent.</p> <p>This object is passed in the variable binding of all SNMP traps generated on behalf of Tuxedo system events.</p>

Note: To run multiple SNMP agents on the same managed node, they must be started as sub-agents (without `-s` option) and run after starting the agent integrator.

Trap Definitions

This section defines all the traps generated by the BEA SNMP Agent when system events occur.

DOMAIN Traps

The Domain Traps group defines the Tuxedo domain specific event traps.

resourceConfigTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when .SysResourceConfig occurs. It denotes a system configuration change.
Action	This is an informational message.
Trap ID	1

MACHINE Traps

The Machine Traps group defines the Tuxedo machine specific event traps.

machineBroadcastTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysMachineBroadcast</code> occurs. It denotes broadcast delivery failure. This message indicates that <code>tpbroadcast()</code> failed for at least one accessor on the LMID of the application.
Action	Since the broadcast messages are sent in no-blocking mode, it is possible that the process doing the broadcasting encountered a blocking condition and dropped a message. Configure larger message queues or load-balance clients and servers such that excessive load is not put on some machines.
Trap ID	2

machineConfigTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysMachineConfig</code> occurs. It denotes a change in a particular machine configuration.
Action	This is an informational message.
Trap ID	3

machineFullMaxAccessersTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when .SysMachineFullMaxaccessers occurs. This message indicates that the given LMID reached the capacity limit on the number of accessers.
Action	Increase the MAXACCESSERS value for the particular machine. Or, if the hardware/software limits have been reached for the maximum number of users on the machine, move additional users to other machines.
Trap ID	4

machineFullMaxConvTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when .SysMachineFullMaxconv occurs. This message indicates that the given LMID reached the capacity limit on the number of concurrent conversations.
Action	Increase the value of MAXCONV for the particular machine to the point that this event is not generated.
Trap ID	5

machineFullMaxGttTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when .SysMachineFullMaxgtt is raised. This message indicates that the given machine reached the capacity limit on the number of concurrent transactions.
Action	Increase the value of MAXGTT for the particular machine to the point that this event is not generated.
Trap ID	6

machineFullMaxWsClientsTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when .SysMachineFullMaxwsclients is raised. This message indicates that the given machine reached the capacity limit on the number of workstation clients.
Action	Increase the value of MAXWSCLIENTS for the particular machine to the point that this event is not generated.
Trap ID	7

machineMsgQTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when <code>.SysMachineMsgq</code> occurs. This message indicates that the server posting a message encountered a blocking condition while putting a message on the message queue.
Action	Configure larger message queues and/or distribute the load equally on all the machines.
Trap ID	8

machinePartitionedTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when <code>.SysMachinePartitioned</code> occurs. This message indicates that DBBL partitioned the stated machine either because the BBL on the machine is slow or the network link between the master and the machine is broken.
Action	<p>This can occur due to several reasons:</p> <ul style="list-style-type: none">■ The entire network might be bogged down due to heavy traffic.■ The BBL or BRIDGE on the non-master is either dead or slow.■ The BRIDGE process on the non-master is extremely busy. <p>The software is capable of unpartitioning the machine if things stabilize.</p>
Trap ID	9

machineSlowTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysMachineSlow</code> occurs. This message indicates that BBL on the non-master machine is slow in generating IAMOK messages. These messages are sent periodically from BBLs to the DBBL that helps the DBBL maintain the pulse of the system.
Action	<p>This can occur due to several reasons:</p> <ul style="list-style-type: none">■ The entire network might be bogged down due to heavy traffic.■ The BBL on the non-master might be either dead or slow.■ The BRIDGE process on the non-master is extremely busy. <p>This problem can be intermittent.</p>
Trap ID	10

machineStateTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysMachineState</code> occurs. This denotes that a particular machine changed its state.
Action	This is an informational message.
Trap ID	11

BRIDGE Traps

The Bridge Traps group defines the Tuxedo bridge specific traps.

networkConfigTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysNetworkConfig</code> occurs. This message indicates the network link between the two machines specified changed to a new state.
Action	This is an informational message.
Trap ID	12

networkDroppedTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysNetworkDropped</code> occurs. This message indicates the network link between the two machines specified was dropped abnormally.
Action	This can happen either because the BRIDGE on either machine died or one of the machines crashed.
Trap ID	13

networkFailureTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysNetworkFailure</code> occurs. This indicates a network connection failure between BRIDGE processes.
Action	This can happen either because the BRIDGE on remote machine died or the remote machine itself crashed.
Trap ID	14

networkFlowTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysNetworkFlow</code> occurs. This message states that the virtual circuit between machines changed to a new state.
Action	This is an informational message.
Trap ID	15

networkStateTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when <code>.SysNetworkState</code> occurs. This message indicates that the server died abnormally and BBL cleaned up the slot allocated by the server.
Action	Debug the server and fix the problem before the server is restarted.
Trap ID	16

SERVER Event Traps

The Server Traps group defines the Tuxedo server specific traps.

serverCleaningTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when <code>.SysServerCleaning</code> occurs. This message indicates that the server died abnormally and BBL cleaned up the slot allocated by the server.
Action	Debug the server and fix the problem before the server is restarted.
Trap ID	17

serverConfigTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysServerConfig</code> occurs. This message indicates that the configuration parameters for the server have been updated.
Action	This is an informational message.
Trap ID	18

serverDiedTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysServerDied</code> occurs. This message indicates that the server died abnormally and the BBL detected this condition in its periodic scan of the BB.
Action	Debug the server and fix the problem before the server is restarted.
Trap ID	19

serverInitTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when .SysServerInit occurs. This message indicates that the server specified above failed in tpsvrinit() and therefore could not be booted.
Action	Fix the problem and then reboot the server. This problem might be due to a Tuxedo resource limit or an application-specific problem.
Trap ID	20

serverMaxgenTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when .SysServerMaxgen occurs. This message indicates that the server died abnormally. Since the server has been marked as restartable, it has been restarted MAXGEN-1 times in the specified GRACE period.
Action	Tuxedo application servers should not die abnormally. If this happens, it is most likely due to an application-specific problem. Debug the server and resolve the problem before restarting the server.
Trap ID	21

serverRestartingTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysServerRestarting</code> occurs.. This message indicates that the server died abnormally. Since this has been marked as a restartable server, it has been restarted.
Action	Tuxedo application servers should not die abnormally. If this happens, it is most likely due to an application-specific problem. Debug the server and resolve the problem before restarting the server.
Trap ID	22

serverStateTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysServerState</code> occurs. This message indicates that the server changed state.
Action	This is an informational message.
Trap ID	23

serverTpExitTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysServerTpexit</code> occurs. This message indicates that the server received a request and the service routine code did a <code>tpreturn(TPEXIT)</code> while the server was executing application-specific code.
Action	This is an informational message.
Trap ID	24

CLIENT Traps

The Client Traps group defines the Tuxedo client-specific traps.

clientConfigTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysClientConfig</code> is raised. This denotes that a particular user on a machine changed its configuration.
Action	This is an informational message.
Trap ID	25

clientDiedTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysClientDied</code> occurs. This message indicates that the client exited the application without doing a <code>tpterm()</code> . Normally, clients should do a <code>tpterm()</code> before exiting the application.
Action	This is an informational message.
Trap ID	26

clientSecurityTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysClientSecurity</code> occurs. This message indicates that the client failed security validation when trying to join the application.
Action	Check to make sure that this is not an unauthorized user trying to gain access to your application data.
Trap ID	27

clientStateTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when <code>.SysClientState</code> occurs. This message indicates that a particular client on a machine changed state.
Action	This is an informational message.
Trap ID	28

TRANSACTION Traps

The Transaction Traps group defines the Tuxedo transaction-specific traps.

transHeuristicAbortTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName }
Description	This trap is generated when <code>.SysTransactionHeuristicAbort</code> occurs. This message indicates that the database in a particular group performed an heuristic abort on a transaction.
Action	Check to make sure that the coordinator of the transaction is still running.
Trap ID	29

transHeuristicCommitTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysTransactionHeuristicCommit</code> occurs. This message indicates that the database in a particular group performed an heuristic commit on a transaction.
Action	Check to make sure that the coordinator of the transaction is still running.
Trap ID	30

EVENT Traps

The Event Traps group defines the Tuxedo event specific traps.

eventDeliveryTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when <code>.SysEventDelivery</code> occurs. This message indicates that the event server failed to perform at least one notification for a posted event.
Action	Check to make sure that the notifications specified in the subscriptions that match the posted event are doable.
Trap ID	31

eventFailureTrap

Enterprise	tuxedo
Variables	{ tuxEventsName, tuxEventsSeverity, tuxEventsLmid, tuxEventsTime, tuxEventsUsec, tuxEventsDescription, tuxEventsClass, tuxEventsUlogCat, tuxEventsUlogMsgNum, beaDomainId, beaDomainKey, beaLogicalAgentName}
Description	This trap is generated when .SysEventFailure occurs. The system event server periodically sends a message to itself to detect blocking conditions on the message queues. This event is generated if the server cannot put a message on the queue in no-block mode. It can also be generated if the received message does not match what was sent out earlier. The second possible case is very unlikely. This denotes a system event monitor subsystem failure on a particular host.
Action	Configure larger message queues or distribute the load in the application equally among all the machines.
Trap ID	32

Index

A

- absolute OIDs 1-4
- access control list
 - MIB groups for 6-1
 - what it is 6-1
- access control lists
 - retrieving and modifying values of 6-1
- access to Tuxedo System entities, of groups 6-4
- active applications
 - runtime statistics by machine 2-76
- active transactions, attributes of
 - as MIB objects 2-136
- administrative processes on master machine,
 - deactivating or activating 2-36
- Agent Integrator
 - supported MIB objects 1-5
- application attributes
 - by machine 2-62
- application events 9-1
- application queues, attributes of
 - as MIB objects 2-89
- application queues, features of
 - as MIB objects 8-1
- application security, type of
 - as MIB object 2-46
- application users and domains, attributes of
 - groups of 6-2
- applications, attributes of
 - for server group 2-55
- attributes of domain

- as MIB objects 2-31

B

BBL

- See Also bulletin board 2-47*

- beaDomainId 4-2
- beaDomainKey 4-1
- beaDomainList 4-1
- beaDomainStatus 4-3
- beaDomainTuxconfig 4-3
- beaDomainTuxdir 4-2
- beaEvtAgentName 2-165
- beaEvtExpr 2-166
- beaEvtFilter 2-166
- beaEvtFilterid 2-165
- beaEvtFilterState 2-167
- beaEvtFilterTable 2-165
- beaIntAgtTable
 - role of 1-5
- beaLogicalAgentName 4-2
- beaRoutingType 2-97
- bridge, Tuxedo
 - MIB table 2-3
- bulletin board
 - machine table 2-40
 - number of queue table entries 2-49
 - number of server table entries 2-50
 - number of service table entries 2-50
 - number of type table entries 2-50
 - queue table 2-41

- routing criteria range table 2-41
- routing table entries 2-48
- server group table entries 2-49
- server table 2-42
- service table 2-42
- starting 2-65
- string pool table 2-41

C

- client
 - network address of 2-18
- client, death of
 - trap 10-19
- client, elapsed time of
 - calculation of 2-18
- clientConfigTrap 10-18
- clientDiedTrap 10-19
- clients, attributes of
 - as MIB objects 2-9
- clientStateTrap 10-20
- conversations, attributes of
 - as MIB objects 2-21
- CORBA 5-4
- CORBA interfaces
 - configuration and runtime attributes as MIB objects 5-5
- customer support contact information vii

D

- device lists, Tuxedo System /T
 - files storing 2-27
- dip-in notification method
 - activating 2-44
- documentation, where to find it vi
- domain, global application attributes of
 - as MIB objects 2-31
- domains, Tuxedo or wle
 - list of as MIB table 4-1
- domain-specific traps 10-6

- DTP transaction log size, by machine
 - as MIB object 2-71

E

- encryption level 2-158
 - tuxTBridgeCurEncryptBits 2-8
- enterprise OID
 - as value of enterprise field in traps 1-9
- entry object in a table 1-8
- Event Broker subscriptions
 - for service-based notifications 9-15
 - for system commands 9-5
 - for writing system userlog messages 9-18
- event filters, Tuxedo
 - as MIB objects 2-165
- event subscription requests
 - as MIB objects 9-2
- eventDeliveryTrap 10-21
- eventFailureTrap 10-22

F

- factory-based routing specifications 2-94

G

- groups with access to Tuxedo System
 - as MIB objects 6-4

H

- handler, workstation
 - See WSH* 7-6

L

- listener processes, attributes of
 - as MIB objects 2-135
- listener, workstation
 - See WSL* 7-9

- load balancing
 - as MIB object 2-43
- local machine, statistics of
 - as MIB objects 2-76
- logical agent name 4-2
- logical machine identifier
 - where originator is accessing application 2-23
- logical machines
 - network group membership 2-156
- logical machines, connectivity of
 - as MIB objects 2-3

M

- machineBroadcastTrap 10-7
- machineConfigTrap 10-7
- machineFullMaxAccessesTrap 10-8
- machineFullMaxConvTrap 10-8
- machineFullMaxGttTrap 10-9
- machineFullMaxWsClientsTrap 10-9
- machineMsgQTrap 10-10
- machinePartitionedTrap 10-10
- machineSlowTrap 10-11
- machineStateTrap 10-11
- master machine
 - deactivating or activating administrative processes on 2-35
- message queues, Tuxedo System/T
 - as MIB objects 2-85
- message, moving
 - via SNMP SET request 8-20
- messages, in queues
 - as MIB objects 8-17
- MIB objects
 - values returned when nonexistent 1-7
- MIB, TUXEDO standard
 - See TMIB 1-9
- MIBs
 - how to use 1-7

N

- network address
 - of workstation clients 2-18
- network address of logical machine 2-72
- network groups, attributes of
 - as MIB objects 2-154
- networkConfigTrap 10-12
- networkDroppedTrap 10-12
- networkFailureTrap 10-13
- networkFlowTrap 10-13
- networkStateTrap 10-14

O

- object identifier
 - of TUXEDO 1-9
 - See OID 1-1
- OID
 - expressing with textual names 1-4
 - relative and absolute 1-4
 - use with Agent Integrator polling rules 1-5
 - what it is 1-2
- OID tree
 - what it is 1-2

P

- printing product documentation vi

Q

- queue space, attributes of
 - as MIB objects 8-24
- queue space, state of
 - as MIB object 8-27
- queue-based notifications
 - subscriptions for 9-8
- queues
 - maximum number of in bulletin board queue table 2-41

- queues, creating
 - via SNMP SET request 8-7
- queues, number of
 - as MIB object 8-31

R

- related information vii
- relative OIDs 1-4
- resourceConfigTrap 10-6
- RFC 1212 1-4
- RFC 1215 1-8
- routing specifications, of applications
 - as MIB objects 2-94

S

- security, application
 - as MIB object 2-46
- security, of application
 - as MIB object 2-46
- security, Tuxedo 6-1
- server group
 - application attributes 2-56
- server table, extension of 2-113
- server workload
 - as MIB object 2-117
- server, death of
 - trap 10-15
- serverCleaningTrap 10-14
- serverConfigTrap 10-15
- serverDiedTrap 10-15
- serverInitTrap 10-16
- serverMaxgenTrap 10-16
- serverRestartingTrap 10-17
- servers
 - maximum number of in bulletin board
 - server table 2-42
- servers in an application, attributes of
 - as MIB objects 2-99
- servers, administrative

- starting 2-65
- servers, Tuxedo
 - traps specific to 10-14
- serverStateTrap 10-17
- serverTpExitTrap 10-18
- services
 - maximum number of in bulletin board
 - service table 2-42
- services in applications, attributes of
 - as MIB objects 2-123
- services within an application, attributes of
 - as MIB objects 2-129
- severity, of system event 10-2
- slow, response to DBBL
 - trap 10-11
- specific trap ID 1-9
- support
 - technical vii

T

- tables
 - Entry object under 1-8
- timeout, of service requests
 - as MIB object 2-132
- TMIB 1-7
 - differences from SNMP MIBs 1-9
- transactions
 - traps specific to 10-20
- transactions, active
 - MIB table for 2-136
 - state of 2-138
- transactions, attributes of
 - as MIB objects 8-42
- transHeuristicAbortTrap 10-20
- transHeuristicCommitTrap 10-21
- traps
 - domain-specific 10-6
 - machine-specific 10-7
 - Tuxedo bridge-specific 10-12
 - Tuxedo client-specific 10-18

Tuxedo event-specific 10-21
Tuxedo server-specific 10-14
Tuxedo transaction-specific 10-20
traps, SNMP
 generated by Tuxedo agent 10-1
tuxDmAclName 3-3
tuxDmAclState 3-3
tuxDmAclTable 3-3
tuxDmConDmCurEncryptBits 3-6
tuxDmConDmlAccessPoint 3-5
tuxDmConDmrAccessPoint 3-5
tuxDmConDmType 3-5
tuxDmConnectionTable 3-5
tuxDmConState 3-6
tuxDmExpDmAclName 3-8
tuxDmExpDmConv 3-9
tuxDmExpDmInBufType 3-9
tuxDmExpDmlAccessPoint 3-8
tuxDmExpDmOutBufType 3-10
tuxDmExpDmRemoteName 3-9
tuxDmExpDmResourceName 3-7
tuxDmExpDmResourceType 3-9
tuxDmExpDmTopenFunction 3-11
tuxDmExpDmTopenProduct 3-10
tuxDmExpDmTopenQualifier 3-11
tuxDmExpDmTopenRtqGroup 3-11
tuxDmExpDmTopenRtqName 3-12
tuxDmExpDmTopenTarget 3-11
tuxDmExportTable 3-7
tuxDmExpState 3-8
tuxDmImpDmAutoTran 3-15
tuxDmImpDmConv 3-15
tuxDmImpDmInBufType 3-17
tuxDmImpDmlAccessPoint 3-14
tuxDmImpDmLoad 3-15
tuxDmImpDmOutBufType 3-17
tuxDmImpDmPrio 3-15
tuxDmImpDmrAccessPointList 3-14
tuxDmImpDmRemoteName 3-16
tuxDmImpDmResourceName 3-14
tuxDmImpDmResourceType 3-16

tuxDmImpDmRoutingName 3-16
tuxDmImpDmteFunction 3-17
tuxDmImpDmteProduct 3-17
tuxDmImpDmteQualifier 3-18
tuxDmImpDmteRtqGroup 3-18
tuxDmImpDmteRtqName 3-19
tuxDmImpDmteTarget 3-18
tuxDmImpDmTranTime 3-16
tuxDmImportTable 3-13
tuxDmImpState 3-14
tuxDmLclDmAccessPoint 3-21
tuxDmLclDmAuditLog 3-22
tuxDmLclDmBlobShmSize 3-27
tuxDmLclDmBlockTime 3-22
tuxDmLclDmConnectionPolicy 3-24
tuxDmLclDmConnPrincipalName 3-26
tuxDmLclDmMachineType 3-26
tuxDmLclDmMaxRapTran 3-22
tuxDmLclDmMaxRetry 3-25
tuxDmLclDmMaxTran 3-23
tuxDmLclDmRetryInterval 3-25
tuxDmLclDmSecurity 3-23
tuxDmLclDmSrvGroup 3-21
tuxDmLclDmTlogDev 3-24
tuxDmLclDmTlogName 3-24
tuxDmLclDmTlogSize 3-24
tuxDmLclDmType 3-21
tuxDmLclState 3-21
tuxDmLocalTable 3-20
tuxDmOsiDmAccessPoint 3-28
tuxDmOsiDmAcl 3-29
tuxDmOsiDmAid 3-30
tuxDmOsiDmAeq 3-29
tuxDmOsiDmApid 3-30
tuxDmOsiDmApt 3-29
tuxDmOsiDmMaxListeningEp 3-30
tuxDmOsiDmNwDevice 3-29
tuxDmOsiDmState 3-28
tuxDmOsiDmUrch 3-30
tuxDmOsiDmXatmiEncoding 3-31
tuxDmOsitpTable 3-28

tuxDmPasswdDmlAccessPoint 3-32	tuxDmSnaLinkDmlSysID 3-49
tuxDmPasswdDmlPWD 3-32	tuxDmSnaLinkDmlUname 3-49
tuxDmPasswdDmrAccessPoint 3-32	tuxDmSnaLinkDmMaxSyncLvl 3-51
tuxDmPasswdDmrPWD 3-33	tuxDmSnaLinkDmMinWin 3-50
tuxDmPasswdState 3-33	tuxDmSnaLinkDmModeName 3-50
tuxDmPasswordTable 3-32	tuxDmSnaLinkDmrAccessPoint 3-49
tuxDmPrincipalMapTable 3-34	tuxDmSnaLinkDmrSysID 3-49
tuxDmPrinMapDirection 3-35	tuxDmSnaLinkDmSecType 3-50
tuxDmPrinMapDmlAccessPoint 3-34	tuxDmSnaLinkDmSNALink 3-48
tuxDmPrinMapDmlPrinName 3-34	tuxDmSnaLinkDmSNAAssess 3-51
tuxDmPrinMapDmrAccessPoint 3-34	tuxDmSnaLinkDmSNAStack 3-49
tuxDmPrinMapDmrPrinName 3-35	tuxDmSnaLinkDmStartType 3-51
tuxDmPrinMapState 3-35	tuxDmSnaLinkState 3-50
tuxDmrAccessPointList 3-3	tuxDmSnaLinkTable 3-48
tuxDmRemoteDmAccessPoint 3-36	tuxDmSnaStackDmlUname 3-53
tuxDmRemoteDmAccessPointID 3-36	tuxDmSnaStackDmSnaCRM 3-52
tuxDmRemoteDmCodePage 3-37	tuxDmSnaStackDmSnaStack 3-52
tuxDmRemoteDmMachineType 3-38	tuxDmSnaStackDmStackParams 3-53
tuxDmRemoteState 3-37	tuxDmSnaStackDmStackType 3-52
tuxDmRemoteTable 3-36	tuxDmSnaStackDmTpName 3-53
tuxDmRemoteType 3-37	tuxDmSnaStackState 3-53
tuxDmResourcesDmVersion 3-39	tuxDmSnaStackTable 3-52
tuxDmResourcesTable 3-39	tuxDmTdomainDmAccessPoint 3-54
tuxDmRoutingDmBufType 3-40	tuxDmTdomainDmCmpLimit 3-55
tuxDmRoutingDmField 3-41	tuxDmTdomainDmFailOverSeq 3-56
tuxDmRoutingDmFieldType 3-42	tuxDmTdomainDmMaxEncryptBits 3-56
tuxDmRoutingDmRanges 3-42	tuxDmTdomainDmMinEncryptBits 3-56
tuxDmRoutingDmRoutingName 3-40	tuxDmTdomainDmNwAddr 3-54
tuxDmRoutingState 3-43	tuxDmTdomainDmNwDevice 3-55
tuxDmRoutingTable 3-40	tuxDmTdomainState 3-55
tuxDmrPrincipalDmrAccessPoint 3-44	tuxDmTdomainTable 3-54
tuxDmrPrincipalDmrPrinName 3-44	tuxDmTopendDmAccessPoint 3-58
tuxDmrPrincipalDmrPrinPasswd 3-44	tuxDmTopendDmDmtePwd 3-59
tuxDmrPrincipalState 3-45	tuxDmTopendDmFailoverSeq 3-60
tuxDmrPrincipalTable 3-44	tuxDmTopendDmNwAddr 3-58
tuxDMSnaCRMDmAccessPoint 3-46	tuxDmTopendDmNwDevice 3-59
tuxDMSnaCRMDmNWAddr 3-47	tuxDmTopendDmteTpSystem 3-58
tuxDMSnaCRMDmNWDevice 3-47	tuxDmTopendState 3-59
tuxDMSnaCRMDmSNACRM 3-46	tuxDmTopendTable 3-58
tuxDMSnaCRMState 3-46	tuxDmTransactionDm 3-61
tuxDmSnaCRMTable 3-46	tuxDmTransactionDmBranchCount 3-63

tuxDmTransactionDmBranchIndex 3-63
 tuxDmTransactionDmBranchNo 3-64
 tuxDmTransactionDmBranchState 3-64
 tuxDmTransactionDmNetTranID 3-64
 tuxDmTransactionDmrAccessPoint 3-64
 tuxDmTransactionDmTPTranID 3-62
 tuxDmTransactionDmTxAccessPoint 3-63
 tuxDmTransactionDmTxNetTranID 3-63
 tuxDmTransactionState 3-62
 tuxDmTransactionTable 3-61
 TUXEDO
 object identifier of 1-9
 Tuxedo application, access to 6-6
 Tuxedo configuration file, by machine
 as MIB object 2-64
 Tuxedo events
 types of 9-1
 TUXEDO MIB
 See TMIB 1-1
 Tuxedo software pathname, by machine
 as MIB object 2-65
 Tuxedo software, pathname to
 as MIB object 4-2
 TUXEDO standard MIB
 See TMIB 1-10
 Tuxedo System /T
 listener processes 2-135
 Tuxedo system events 9-1
 Tuxedo, states of 1-5
 tuxEventClientExpr 9-3
 tuxEventClientFilter 9-4
 tuxEventClientId 9-4
 tuxEventClientIndx 9-3
 tuxEventClientState 9-4
 tuxEventClientTbl 9-3
 tuxEventCmd 9-7
 tuxEventCmdExpr 9-6
 tuxEventCmdFilter 9-6
 tuxEventCmdIndx 9-5
 tuxEventCmdState 9-6
 tuxEventCmdTbl 9-5
 tuxEventPersist 9-13
 tuxEventQctlBeforeMsgid 9-11
 tuxEventQctlCorrId 9-13
 tuxEventQctlDeqTime 9-12
 tuxEventQctlFailQ 9-13
 tuxEventQctlMsgid 9-12
 tuxEventQctlPrior 9-12
 tuxEventQctlQtimeAbs 9-11
 tuxEventQctlQtimeRel 9-12
 tuxEventQctlQtop 9-11
 tuxEventQctlReplyQ 9-13
 tuxEventQname 9-10
 tuxEventQSpace 9-10
 tuxEventQueExpr 9-9
 tuxEventQueFilter 9-9
 tuxEventQueIndx 9-9
 tuxEventQueState 9-10
 tuxEventQueTbl 9-8
 tuxEventsClass 10-4
 tuxEventsDescription 10-3
 tuxEventsLmid 10-3
 tuxEventsName 10-2
 tuxEventsSeverity 10-2
 tuxEventsTime 10-3
 tuxEventsUlogCat 10-4
 tuxEventsUlogMsgNum 10-4
 tuxEventsUsec 10-3
 tuxEventSvcExpr 9-16
 tuxEventSvcFilter 9-16
 tuxEventSvcIndx 9-15
 tuxEventSvcName 9-17
 tuxEventSvcPersist 9-17
 tuxEventSvcState 9-16
 tuxEventSvcTbl 9-15
 tuxEventSvcTran 9-17
 tuxEventTran 9-14
 tuxEventUlogExpr 9-19
 tuxEventUlogFilter 9-19
 tuxEventUlogIndx 9-18
 tuxEventUlogState 9-19
 tuxEventUlogTbl 9-18

tuxEventUserLog	9-20	tuxTAppQmaxRetries	8-12
tuxInternalIdx	2-98	tuxTAppQmConfig	8-9
tuxTAclCltname	6-7	tuxTAppQmsgCorId	8-21
tuxTAclGrpId	6-2	tuxTAppQmsgCurRetries	8-21
tuxTAclGrpName	6-2	tuxTAppQmsgExpireTime	8-21
tuxTAclGrpState	6-3	tuxTAppQmsgGrpNo	8-18
tuxTAclGrpTable	6-2	tuxTAppQmsgId	8-18
tuxTAclPermGrpIds	6-5	tuxTAppQmsgLmid	8-19
tuxTAclPermName	6-4	tuxTAppQmsgNewQname	8-20
tuxTAclPermState	6-5	tuxTAppQmsgPersistent	8-22
tuxTAclPermTable	6-4	tuxTAppQmsgPrior	8-20
tuxTAclPermType	6-4	tuxTAppQmsgQmConfig	8-19
tuxTAclPrinGrp	6-7	tuxTAppQmsgQname	8-18
tuxTAclPrinId	6-7	tuxTAppQmsgQspaceName	8-19
tuxTAclPrinName	6-6	tuxTAppQmsgReplyPersistent	8-23
tuxTAclPrinPasswd	6-7	tuxTAppQmsgSerNo	8-18
tuxTAclPrinState	6-8	tuxTAppQmsgSize	8-21
tuxTAclPrinTbl	6-6	tuxTAppQmsgState	8-19
tuxTAppQcmd	8-10	tuxTAppQmsgTbl	8-17
tuxTAppQcmdHw	8-11	tuxTAppQmsgTime	8-20
tuxTAppQcmdLw	8-11	tuxTAppQname	8-8
tuxTAppQCmdNonPersist	8-14	tuxTAppQorder	8-10
tuxTAppQCmdNonPersistHw	8-15	tuxTAppQoutOfOrder	8-12
tuxTAppQCmdNonPersistLw	8-15	tuxTAppQretryDelay	8-12
tuxTAppQctrl	8-2	tuxTAppQspaceName	8-8
tuxTAppQctrlLmid	8-2	tuxTAppQstate	8-9
tuxTAppQctrlMsgEndTime	8-4, 8-5	tuxTAppQTbl	8-7
tuxTAppQctrlMsgHiPrio	8-4	tuxTBridgeConTime	2-6
tuxTAppQctrlMsgLoPrio	8-4	tuxTBridgeCurEncryptBits	2-8
tuxTAppQctrlMsgStartTime	8-5	tuxTBridgeCurTime	2-6
tuxTAppQctrlQmConfig	8-3	tuxTBridgeFlowCnt	2-8
tuxTAppQctrlQname	8-3	tuxTBridgeLmid	2-4
tuxTAppQctrlSpaceName	8-3	tuxTBridgeNetworkGroupName	2-8
tuxTAppQcurBlocks	8-12	tuxTBridgeNetworkGrpNo	2-8
tuxTAppQcurMsg	8-13	tuxTBridgeRcvdByte	2-7
tuxTAppQCurNonPersistBytes	8-16	tuxTBridgeRcvdNum	2-7
tuxTAppQCurNonPersistMsg	8-16	tuxTBridgeSentByte	2-7
tuxTAppQDefDeliveryPolicy	8-14	tuxTBridgeSentNum	2-7
tuxTAppQDefExpirationTime	8-13	tuxTBridgeState	2-4
tuxTAppQgrpNo	8-9	tuxTBridgeSuspTime	2-6
tuxTAppQLmid	8-9	tuxTBridgeTbl	2-3

tuxTclientBirthTime	2-12	tuxTconnOpid	2-23
tuxTclientCCIntName	2-13	tuxTconnOsndcnt	2-23
tuxTclientCmtRet	2-17	tuxTconnOsrvid	2-24
tuxTclientContextID	2-20	tuxTconnSerNo	2-22
tuxTclientCurConv	2-17	tuxTconnSgrpNo	2-24
tuxTclientCurReq	2-18	tuxTconnSlmid	2-24
tuxTclientCurTime	2-18	tuxTconnSpid	2-24
tuxTclientId	2-20	tuxTconnSsndcnt	2-24
tuxTclientIdleTime	2-13	tuxTconnSsrvId	2-25
tuxTclientLastGrp	2-18	tuxTconnState	2-22
tuxTclientMachineId	2-12	tuxTconnSveName	2-22
tuxTclientNaddr	2-18	tuxTconnTable	2-21
tuxTclientNotify	2-19	tuxTdevCfgDev	2-28
tuxTclientNumConv	2-15	tuxTdeviceName	2-28
tuxTclientNumDeque	2-16	tuxTdeviceTbl	2-27
tuxTclientNumEnque	2-16	tuxTdevIndex	2-29
tuxTclientNumPost	2-16	tuxTdevLmid	2-27
tuxTclientNumReq	2-16	tuxTdevOffset	2-28
tuxTclientNumSubscribe	2-16	tuxTdevSize	2-28
tuxTclientNumTran	2-17	tuxTdevState	2-29
tuxTclientNumTranAbt	2-17	tuxTdomain	2-31
tuxTclientNumTranCmt	2-17	tuxTdomainAuthsvc	2-47
tuxTclientNumUnSol	2-19	tuxTdomainBBLQuery	2-47
tuxTclientPid	2-13	tuxTdomainBlockTime	2-48
tuxTclientReg	2-13	tuxTdomainCMTRET	2-43
tuxTclientRelease	2-15	tuxTdomainCurDRT	2-48
tuxTclientRpid	2-19	tuxTdomainCurGroups	2-49
tuxTclientSrvGrp	2-14	tuxTdomainCurMachines	2-49
tuxTclientState	2-11	tuxTdomainCurQueues	2-49
tuxTclientTbl	2-9	tuxTdomainCurRFT	2-49
tuxTclientTimeLeft	2-19	tuxTdomainCurRTdata	2-49
tuxTclientTimeStart	2-20	tuxTdomainCurServers	2-50
tuxTclientTranLev	2-20	tuxTdomainCurServices	2-50
tuxTclientUsrName	2-14	tuxTdomainCursType	2-50
tuxTclientWsc	2-14	tuxTdomainCurType	2-50
tuxTclientWsh	2-14	tuxTdomainDBBLWait	2-48
tuxTclientWshClientId	2-15	tuxTdomainEncryptionRequired	2-54
tuxTclientWsProto	2-15	tuxTdomainGID	2-37
tuxTconnClientId	2-23	tuxTdomainHwDRT	2-50
tuxTconnOgrpNo	2-23	tuxTdomainHwGroups	2-51
tuxTconnOlmid	2-23	tuxTdomainHwMachines	2-51

tuxTdomainHwQueues 2-51	tuxTgroupEncryptionRequired 2-61
tuxTdomainHwRFT 2-51	tuxTgroupLMID 2-56
tuxTdomainHwRTdata 2-51	tuxTgroupName 2-55
tuxTdomainHwServers 2-52	tuxTgroupNo 2-56
tuxTdomainHwServices 2-52	tuxTgroupOpenInfo 2-59
tuxTdomainID 2-37	tuxTgroupSignatureRequired 2-61
tuxTdomainKey 2-34	tuxTgroupState 2-57
tuxTdomainLoadBalance 2-43	tuxTgroupTable 2-55
tuxTdomainMask 2-38	tuxTgroupTMScount 2-60
tuxTdomainMaster 2-34	tuxTgroupTMSname 2-60
tuxTdomainMaxAccessers 2-38	tuxTlistenLmid 2-135
tuxTdomainMaxACLgroups 2-42	tuxTlistenState 2-135
tuxTdomainMaxBufsType 2-39	tuxTlistenTbl 2-135
tuxTdomainMaxBufType 2-39	tuxTmachineAclCacheAccess 2-83
tuxTdomainmaxConv 2-38	tuxTmachineAclCacheHits 2-83
tuxTdomainMaxDRT 2-39	tuxTmachineAclFail 2-83
tuxTdomainMaxGroups 2-40	tuxTmachineActive 2-76
tuxTdomainMaxGTT 2-39	tuxTmachineAppDir 2-65
tuxTdomainMaxMachines 2-40	tuxTmachineBridge 2-71
tuxTdomainMaxNetGroups 2-52	tuxTmachineCmpLimit 2-72
tuxTdomainMaxQueues 2-41	tuxTmachineCurAccessers 2-77
tuxTdomainMaxRFT 2-41	tuxTmachineCurClients 2-78
tuxTdomainMaxRTData 2-41	tuxTmachineCurConv 2-78
tuxTdomainMaxServers 2-42	tuxTmachineCurGTT 2-78
tuxTdomainMaxServices 2-42	tuxTmachineCurLoad 2-78
tuxTdomainMode 2-35	tuxTmachineCurWsClients 2-78
tuxTdomainNotify 2-43	tuxTmachineEncryptionRequired 2-75
tuxTdomainOptions 2-45	tuxTmachineEnvFile 2-67
tuxTdomainPerm 2-37	tuxTmachineGid 2-67
tuxTdomainSanityScan 2-48	tuxTmachineHwAccessers 2-79
tuxTdomainScanUnit 2-47	tuxTmachineHwAclCache 2-82
tuxTdomainSecurity 2-46	tuxTmachineHwClients 2-79
tuxTdomainSignal 2-45	tuxTmachineHwConv 2-79
tuxTdomainSignatureAhead 2-53	tuxTmachineHwGTT 2-79
tuxTdomainSignatureBehind 2-54	tuxTmachineHwWsClients 2-79
tuxTdomainSignatureRequired 2-54	tuxTmachineLicExpires 2-81
tuxTdomainState 2-35	tuxTmachineLicMaxUsers 2-82
tuxTdomainSystemAccess 2-44	tuxTmachineLicSerial 2-82
tuxTdomainUID 2-37	tuxTmachineLmid 2-64
tuxTgroupCloseInfo 2-59	tuxTmachineLogDevice 2-70
tuxTgroupCurLMID 2-59	tuxTmachineMaxAccessers 2-69

tuxTmachineMaxActCache 2-70	tuxTmsgLsPid 2-87
tuxTmachineMaxConv 2-69	tuxTmsgQbytes 2-87
tuxTmachineMaxGtt 2-69	tuxTmsgQnum 2-87
tuxTmachineMaxWsClients 2-70	tuxTmsgRtime 2-88
tuxTmachineMinor 2-74	tuxTmsgState 2-86
tuxTmachineNaddr 2-71	tuxTmsgStime 2-88
tuxTmachineNlsaddr 2-72	tuxTmsgTable 2-85
tuxTmachineNumConv 2-80	tuxTnetGrpName 2-154
tuxTmachineNumDequeue 2-80	tuxTnetGrpNo 2-154
tuxTmachineNumEnqueue 2-80	tuxTnetGrpPrio 2-155
tuxTmachineNumPost 2-80	tuxTnetGrpState 2-155
tuxTmachineNumReq 2-80	tuxTnetGrpTbl 2-154
tuxTmachineNumSubscribe 2-81	tuxTnetMapGrpName 2-156
tuxTmachineNumTran 2-81	tuxTnetMapGrpNo 2-157
tuxTmachineNumTranAbt 2-81	tuxTnetMapLmid 2-157
tuxTmachineNumTranCmt 2-81	tuxTnetMapMaxEncryptBit 2-158
tuxTmachinePageSize 2-82	tuxTnetMapMinEncryptBit 2-158
tuxTmachinePerm 2-68	tuxTnetMapNaddr 2-158
tuxTmachinePmid 2-64	tuxTnetMapState 2-157
tuxTmachineRelease 2-74	tuxTnetMapTbl 2-156
tuxTmachineRole 2-73	tuxTpTranId 2-137
tuxTmachineSignatureRequired 2-75	tuxTQspaceBlocking 8-29
tuxTmachineSpinCount 2-73	tuxTQspaceCurExtent 8-31
tuxTmachineState 2-65	tuxTQspaceCurMsg 8-31
tuxTmachineSWrelease 2-82	tuxTQspaceCurProc 8-31
tuxTmachineTable 2-62	tuxTQspaceCurQueues 8-31
tuxTmachineTlogName 2-71	tuxTQspaceCurTrans 8-32
tuxTmachineTlogSize 2-71	tuxTQspaceErrQname 8-29
tuxTmachineTmNetLoad 2-73	tuxTQspaceForceInit 8-29
tuxTmachineTuxConfig 2-64	tuxTQspaceGrpNo 8-27
tuxTmachineTuxDir 2-64	tuxTQspaceHwMsg 8-32
tuxTmachineType 2-68	tuxTQspaceHwProc 8-32
tuxTmachineUid 2-67	tuxTQspaceHwQueues 8-32
tuxTmachineUlogPfx 2-68	tuxTQspaceHwTrans 8-33
tuxTmachineWkComplete 2-83	tuxTQspaceIpckey 8-29
tuxTmachineWkInitiated 2-84	tuxTQspaceLmid 8-27
tuxTmsgCbytes 2-86	tuxTQspaceMaxMsg 8-29
tuxTmsgCtime 2-87	tuxTQspaceMaxPages 8-30
tuxTmsgCurTime 2-86	tuxTQspaceMaxProc 8-30
tuxTmsgId 2-85	tuxTQspaceMaxQueues 8-30
tuxTmsgLrPid 2-87	tuxTQspaceMaxTrans 8-30

tuxTQspaceName 8-26	tuxTroutingBufType 2-95
tuxTQspacePercentInit 8-33	tuxTroutingField 2-95
tuxTQspaceQmConfig 8-26	tuxTroutingName 2-95
tuxTQspaceState 8-27	tuxTroutingRanges 2-96
tuxTQspaceTbl 8-24	tuxTroutingState 2-97
tuxTQtransGrpNo 8-44	tuxTroutingTable 2-94
tuxTQtransIndx1 8-43	tuxTserverCtxtCltLmId 2-160
tuxTQtransIndx2 8-43	tuxTserverCtxtCltPid 2-161
tuxTQtransIndx3 8-43	tuxTserverCtxtCltReply 2-161
tuxTQtransIndx4 8-43	tuxTserverCtxtCmtRet 2-161
tuxTQtransIndx5 8-44	tuxTserverCtxtContextID 2-160
tuxTQtransLmid 8-45	tuxTserverCtxtCurConv 2-161
tuxTQtransQmConfig 8-44	tuxTserverCtxtCurReq 2-162
tuxTQtransSpaceName 8-44	tuxTserverCtxtCurService 2-162
tuxTQtransState 8-45	tuxTserverCtxtGrp 2-160
tuxTQtransTbl 8-42	tuxTserverCtxtLastGrp 2-162
tuxTQtransXid 8-42	tuxTserverCtxtServerID 2-160
tuxTqueueNqueued 2-92	tuxTserverCtxtSvcTimeOut 2-162
tuxTqueueRqAddr 2-89	tuxTserverCtxtTimeLeft 2-162
tuxTqueueRqId 2-91	tuxTserverCtxtTranLev 2-163
tuxTqueueSource 2-92	tuxTsrvrBaseSrvId 2-104
tuxTqueueSrvrCnt 2-91	tuxTsrvrCLOpt 2-104
tuxTqueueState 2-90	tuxTsrvrCltLmid 2-117
tuxTqueueTable 2-89	tuxTsrvrCltPid 2-117
tuxTqueueTotNqueued 2-91	tuxTsrvrCltReply 2-118
tuxTqueueTotWkQueued 2-92	tuxTsrvrCmtRet 2-118
tuxTqueueWkQueued 2-93	tuxTsrvrConv 2-108
tuxTranCoordLmid 2-136	tuxTsrvrCurConv 2-118
tuxTranGrpCnt 2-140	tuxTsrvrCurDispatchThreads 2-122
tuxTranGrpIndex 2-140	tuxTsrvrCurReq 2-118
tuxTranGrpNo 2-140	tuxTsrvrCurService 2-119
tuxTranGstate 2-140	tuxTsrvrCurTime 2-119
tuxTranIndx1 2-137	tuxTsrvrEnvFile 2-104
tuxTranIndx2 2-137	tuxTsrvrGeneration 2-109
tuxTranIndx3 2-137	tuxTsrvrGrace 2-104
tuxTranIndx4 2-138	tuxTsrvrGrp 2-100
tuxTranIndx5 2-138	tuxTsrvrGrpExt 2-120
tuxTranState 2-138	tuxTsrvrGrpNo 2-101
tuxTranTbl 2-136	tuxTsrvrGrpNoExt 2-115
tuxTranTimeOut 2-140	tuxTsrvrHwDispatchThreads 2-122
tuxTranXid 2-137	tuxTsrvrId 2-101

tuxTsrvrIdExt 2-114
tuxTsrvrLastGrp 2-119
tuxTsrvrMax 2-106
tuxTsrvrMaxDispatchThreads 2-111
tuxTsrvrMaxgen 2-105
tuxTsrvrMin 2-106
tuxTsrvrMinDispatchThreads 2-111
tuxTsrvrName 2-101
tuxTsrvrNumConv 2-115
tuxTsrvrNumDeque 2-115
tuxTsrvrNumDispatchThreads 2-122
tuxTsrvrNumEnque 2-115
tuxTsrvrNumPost 2-115
tuxTsrvrNumReq 2-116
tuxTsrvrNumSubscribe 2-116
tuxTsrvrNumTran 2-116
tuxTsrvrPid 2-109
tuxTsrvrRcmd 2-106
tuxTsrvrReplyQ 2-108
tuxTsrvrRestart 2-107
tuxTsrvrRpid 2-110
tuxTsrvrRpPerm 2-108
tuxTsrvrRqAddr 2-109
tuxTsrvrRqId 2-110
tuxTsrvrRqPerm 2-109
tuxTsrvrSequence 2-107
tuxTsrvrState 2-101
tuxTsrvrStateExt 2-120
tuxTsrvrSvcTimeOut 2-119
tuxTsrvrSystemAccess 2-108
tuxTsrvrTbl 2-99
tuxTsrvrTblExt 2-113
tuxTsrvrThreadStackSize 2-112
tuxTsrvrTimeLeft 2-119
tuxTsrvrTimeRestart 2-110
tuxTsrvrTimeStart 2-111
tuxTsrvrTotReqC 2-117
tuxTsrvrTotWorkL 2-117
tuxTsrvrTranAbt 2-116
tuxTsrvrTranCmt 2-116
tuxTsrvrTranLev 2-120

tuxTsvcAutoTran 2-125
tuxTsvcBufType 2-127
tuxTsvcEncryptionRequired 2-128
tuxTsvcGrp 2-129
tuxTsvcGrpAutoTran 2-132
tuxTsvcGrpLoad 2-132
tuxTsvcGrpName 2-131
tuxTsvcGrpNo 2-131
tuxTsvcGrpPrio 2-132
tuxTsvcGrpState 2-131
tuxTsvcGrpSvcName 2-130
tuxTsvcGrpSvcTimeOut 2-132
tuxTsvcGrpTranTime 2-133
tuxTsvcLoad 2-125
tuxTsvcName 2-124
tuxTsvcPrio 2-126
tuxTsvcrName 2-134
tuxTsvcRoutingName 2-128
tuxTsvcSignatureRequired 2-128
tuxTsvcSrvrId 2-133
tuxTsvcSrvrLmid 2-133
tuxTsvcSrvrNcompleted 2-134
tuxTsvcSrvrNqueued 2-134
tuxTsvcSrvrRqAddr 2-133
tuxTsvcState 2-124
tuxTsvcTbl 2-123
tuxTsvcTimeOut 2-126
tuxTsvcTranTime 2-126
tuxTsvcType 2-124
tuxTulogCat 2-147
tuxTulogCatCtrl 2-152
tuxTulogContextID 2-148
tuxTulogCtrl 2-149
tuxTulogEndTimeCtrl 2-151
tuxTulogLine 2-146
tuxTulogLineCtrl 2-151
tuxTulogLmid 2-145
tuxTulogLmidCtrl 2-150
tuxTulogMmDdDYy 2-145
tuxTulogMmddyCtrl 2-150
tuxTulogMsg 2-146

tuxTulogMsgCtrl 2-151
tuxTulogMsgNum 2-147
tuxTulogMsgNumCtrl 2-153
tuxTulogPid 2-146
tuxTulogPidCtrl 2-152
tuxTulogPmid 2-145
tuxTulogPmidCtrl 2-150
tuxTulogPocNameCtrl 2-153
tuxTulogProcName 2-147
tuxTulogSerNo 2-145
tuxTulogSeverity 2-147
tuxTulogSeverityCtrl 2-152
tuxTulogTable 2-144
tuxTulogThreadID 2-147
tuxTulogTime 2-145
tuxTulogTimeCtrl 2-150
tuxTulogTpTranId 2-146
tuxTulogTptranIdCtrl 2-151
tuxTulogXid 2-146
tuxTulogXidCtrl 2-152
tuxTwshTaActive 7-6
tuxTwshTaClientId 7-3
tuxTwshTaCurClients 7-5
tuxTwshTaCurwork 7-7
tuxTwshTaFlowcnt 7-7
tuxTwshTaGrpNo 7-4
tuxTwshTaHwClients 7-5
tuxTwshTaLmid 7-4
tuxTwshTaMultiplex 7-5
tuxTwshTaNumblockQ 7-7
tuxTwshTaPid 7-5
tuxTwshTaRcvdByt 7-7
tuxTwshTaRcvdNum 7-8
tuxTwshTaSentByt 7-8
tuxTwshTaSentNum 7-8
tuxTwshTaSrvGrp 7-3
tuxTwshTaSrvId 7-4
tuxTwshTaState 7-4
tuxTwshTaTimeleft 7-6
tuxTwshTaTotacttime 7-6
tuxTwshTaTotidtime 7-6

tuxTwshTaWshClientId 7-3
tuxTwslTaClOpt 7-14
tuxTwslTaCurHandlers 7-16
tuxTwslTaDevice 7-11
tuxTwslTaEnvFile 7-14
tuxTwslTaGrace 7-14
tuxTwslTaGrpNo 7-11
tuxTwslTaHwHandlers 7-16
tuxTwslTaKeepAlive 7-17
tuxTwsltaLmid 7-11
tuxTwslTaMaxGen 7-15
tuxTwslTaMaxHandlers 7-13
tuxTwslTaMaxIdleTime 7-13
tuxTwslTaMaxInitTime 7-13
tuxTwslTaMinHandlers 7-12
tuxTwslTaMuxplex 7-13
tuxTwslTaNaddr 7-12
tuxTwslTaNetTimeOut 7-17
tuxTwslTaPid 7-11
tuxTwslTaRcmd 7-15
tuxTwslTaRestart 7-15
tuxTwslTaSequence 7-15
tuxTwslTaSrvGrp 7-10
tuxTwslTaSrvId 7-10
tuxTwslTaState 7-11
tuxTwslTaSuspended 7-16
tuxTwslTaViewRefresh 7-17
tuxTwslTaWshName 7-12
tuxTwslTaWsProto 7-16
tuxTwslTbl 7-9
tuxWshTaNaddr 7-5

U

userlog messages

- Event Broker subscriptions for 9-18
- userlog table messages, control of 2-149
- userlogs, attributes of
 - as MIB objects 2-144

V

- variable bindings
 - contents of 1-9, 10-2
- variable bindings, trap
 - components of 10-2

W

- wleCurInterfaces 2-53
- wleFactoryId 5-3
- wleFactoryIfName 5-4
- wleFactorySerNo 5-3
- wleFactoryState 5-4
- wleFactoryTable 5-3
- wleHwInterfaces 2-53
- wleIfAutoTran 5-9
 - effect of m3IfTxPolicy on 5-9
 - effect of wleIfTxPolicy on 5-18
- wleIfFbRoutingName 5-11
- wleIfLmid 5-11
- wleIfLoad 5-10
- wleIfName 5-7
- wleIfNumServers 5-11
- wleIfPrio 5-10
- wleIfQueueAutoTran 5-18
- wleIfQueueFbRoutingName 5-19
- wleIfQueueLmid 5-19
- wleIfQueueLoad 5-18
- wleIfQueueName 5-16
- wleIfQueueNumServers 5-20
- wleIfQueuePrio 5-18
- wleIfQueueRqAddr 5-16
- wleIfQueueSerNo 5-16
- wleIfQueueSrvGrp 5-16
- wleIfQueueState 5-17
- wleIfQueueTimeout 5-19
- wleIfQueueTpPolicy 5-20
- wleIfQueueTranTime 5-19
- wleIfQueueTxPolicy 5-20
- wleIfSerNo 5-6
- wleIfSrvGrp 5-7

- wleIfState 5-7
- wleIfTimeout 5-10
- wleIfTpPolicy 5-12
- wleIfTranTime 5-11
- wleIfTxPolicy 5-12
 - effect on m3IfAutoTran 5-9
 - effect on wleIfAutoTran 5-18
- wleInterfaceTable 5-5
- wleJdbcConPoolAllowShrinking 5-29
- wleJdbcConPoolCapacityIncr 5-29
- wleJdbcConPoolCreateOnStartUp 5-28
- wleJdbcConPoolDbHost 5-27
- wleJdbcConPoolDbName 5-27
- wleJdbcConPoolDbNetProtocol 5-28
- wleJdbcConPoolDbPassword 5-27
- wleJdbcConPoolDbPort 5-28
- wleJdbcConPoolDbUser 5-27
- wleJdbcConPoolDriver 5-26
- wleJdbcConPoolDsName 5-26
- wleJdbcConPoolEnableXa 5-28
- wleJdbcConPoolExtnConnAvailable 5-33
- wleJdbcConPoolExtnConnUsed 5-33
- wleJdbcConPoolExtnDsName 5-33
- wleJdbcConPoolExtnHwmForWait 5-34
- wleJdbcConPoolExtnSrvGrp 5-32
- wleJdbcConPoolExtnSrvId 5-32
- wleJdbcConPoolExtnTable 5-32
- wleJdbcConPoolHwmConnCreated 5-33
- wleJdbcConPoolHwmConnUsed 5-33
- wleJdbcConPoolInitCapacity 5-29
- wleJdbcConPoolLoginDelay 5-29
- wleJdbcConPoolMaxCapacity 5-29
- wleJdbcConPoolProps 5-28
- wleJdbcConPoolRefresh 5-30
- wleJdbcConPoolShrinkPeriod 5-30
- wleJdbcConPoolSrvGrp 5-25
- wleJdbcConPoolSrvId 5-25
- wleJdbcConPoolState 5-26
- wleJdbcConPoolTable 5-24
- wleJdbcConPoolTestOnRelease 5-31
- wleJdbcConPoolTestOnReserve 5-30

wleJdbcConPoolTestTable 5-30
wleJdbcConPoolUrl 5-26
wleJdbcConPoolUserRole 5-27
wleJdbcConPoolWaitForConn 5-31
wleJdbcConPoolWaitTimeOut 5-31
wleLclIfName 5-13
wleLclIfNcompleted 5-14
wleLclIfNqueued 5-14
wleLclIfQueueCurObjs 5-23
wleLclIfQueueCurTrans 5-23
wleLclIfQueueName 5-21
wleLclIfQueueNcompleted 5-22
wleLclIfQueueNqueued 5-22
wleLclIfQueueRqAddr 5-22
wleLclIfQueueSerNo 5-21
wleLclIfQueueSrvGrp 5-22
wleLclIfQueueTable 5-21
wleLclIfSerNo 5-13
wleLclInterfaceTable 5-13
wleLclSrvGrp 5-14
wleMachineCurObjects 2-84
wleMachineHwObjects 2-84
wleMachineMaxObjects 2-74
wleMaxInterfaces 2-53
wleMaxObjects 2-52
wleModuleModule 5-36
wleModuleModuleArgs 5-36
wleModuleModuleType 5-36
wleModuleSrvGrp 5-35
wleModuleSrvId 5-35
wleModuleState 5-36
wleModuleTable 5-35
wleRoutingFieldType 2-98
wleSrvrClassPath 2-121
wleSrvrCurInterfaceExt 2-120
wleSrvrCurObjsExt 2-120
wleSrvrjavaHeap 2-121
wleSrvrjavaHeapuse 2-121
wleSrvrjavaVendor 2-121
wleSrvrjavaVersion 2-121
wleSrvrSrvType 2-112
wlexJdbcConPoolExtnAwaitingConn 5-34
work station handler
 See WSH 7-2
workload, of server
 as MIB object 2-117
workstation client
 status of 2-11
WSH client processes, attributes of
 as MIB objects 7-2
WSL server processes, attributes of
 as MIB objects 7-9