

Oracle® Tuxedo System and Application Monitor (TSAM)

Reference Guide

11g Release 1 (11.1.1.1.0)

March 2010

ORACLE®

Copyright © 2007, 2010, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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

Contents

Oracle TSAM Reference Guide

- LMS (Local Monitor Server) 1
- tpgetcallinfo 3

Oracle TSAM Error Messages

- Oracle TSAM Error Messages 1

Oracle TSAM Reference Guide

The Oracle TSAM Reference Guide describes system processes and commands delivered with the Oracle TSAM software.

[Table 1](#) lists the Oracle TSAM system processes and commands.

Table 1 Oracle TSAM System Processes and Commands

Name	Description
LMS (Local Monitor Server)	The Oracle TSAM Agent Local Monitor Server
tpgetcallinfo	Routine for retrieving call path message monitoring attributes

LMS (Local Monitor Server)

Name

LMS—The Oracle TSAM Agent Local Monitor Server

Synopsis

```
LMS SRVGRP="identifier" SRVID="number" [other_parms]
CLOPT= "-A -- -l tsam-manager-dataserver-url [-t heartbeat-interval]
[-m metrics-shm-size]"
```

Description

LMS is an Oracle TSAM Agent Tuxedo server. It provides the following functions:

- Acts as the local Tuxedo domain data collection proxy

The performance metrics collected by the Oracle TSAM framework are passed to the plug-in. Oracle TSAM default plug-in sends the data to the LMS using the Tuxedo service infrastructure.

- Plug-in metrics are stored in the LMS before being sent to the Oracle TSAM manager data server. The LMS and Oracle TSAM manager use the HTTP+XML protocol.
- Other management information exchanges between the LMS and Oracle TSAM manager.

The LMS must be configured in the UBBCONFIG file and set with the proper options. One Tuxedo machine must be configured with one LMS. Multiple LMS on one machine is not supported. LMS is recommended to be configured at the end of UBBCONFIG so that it can retrieve all server information when Tuxedo domain booted. LMS still can synchronize the configuration to TSAM manager periodically.

Options

-l

Mandatory parameter. It is followed by the Oracle TSAM manager data server. The host address and port number are set based on your Oracle TSAM manager installation. The format is:

`host:port/tsam`

- host is the host DNS name or IP address of the box where TSAM manager is installed.
- Port is the TCP port number.
- tsam is the Oracle TSAM manager reserved path name.

-m

Optional parameter. Specifies the size of shared memory used to store raw data metrics. The optional trailing letter k or m denotes KB or MB bytes, otherwise the unit is in bytes. The effective value is rounded up to 4K size since it is the page size for most platforms.

The default value is 10MB if this option is not present. The size of the shared memory will not grow at run time when there is no free space to store new data. In this case oldest data is replaced with new data.

-t

Optional parameter. It specifies the time interval in seconds that LMS should connect to the Oracle TSAM manager with configuration synchronization. The default value is 100 seconds.

Example(s)

[Listing 1-1](#) shows the LMS in UBBCONFIG.

Listing 1-1 LMS in UBBCONFIG

```
...
*SERVERS
LMS SRVGRP=LMSGRP SRVID=1
CLOPT="-A -- -l tsamweb.abc.com:8080/tsam/-m 20M -t 180"
...
```

tpgetcallinfo

Name

tpgetcallinfo()— Routine for retrieving call path message monitoring attributes.

Synopsis

```
int tpgetcallinfo(const char *msg, FBFR32 **obuf, long flags)
```

Description

tpgetcallinfo() is used for call path monitoring only. It supports the following parameters:

msg

The typed buffer use for measurement.

obuf

The FML32 buffer used to contain the fields.

flags

Reserved for future use.

For monitored calls, tpgetcallinfo() uses the following fields: following fields:

`tpgetcallinfo()`

Retrieves the message monitoring attributes when call path monitoring is enabled.

`tpgetcallinfo()`

Can be used in different scenarios to accomplish different functions. Typical usage is as follows:

- Application server calls `tpgetcallinfo()` to check the requested message monitoring attributes. It can provide the following information:
 - Correlation ID of the request. It is generated by the caller
 - Begins time stamping when the monitoring initiator starts the call
 - Last stop time stamp on the call path tree of the monitored request. Usually, it is used to measure the process requested message waiting time for a service
 - Workstation client address if the request is from a Tuxedo workstation client
- The monitoring initiator calls `tpgetcallinfo()` to get the end to the monitored call end time.

Note: `tpgetcallinfo()` can be called at any time for a reply buffer after a reply is received. This is especially useful with `tpacall/tpgetrply`.

Table 2 lists the FML monitor metrics field names.

Table 2 Monitor Initiator Field Names

Field Name	Type	Description	Service	Monitoring Initiator
TA_MONCORRID	string	The monitored call correlation ID. It is a critical call path monitoring metric.	Y	Y
TA_MONLASTTIMESEC	long	Timestamp for the last stop on the call path tree in seconds.	Y	Y
TA_MONLASTTIMEUSEC	long	Timestamp of last stop on the call path tree in microseconds.	Y	Y
TA_MONSTARTTIMESEC	long	Timestamp when the monitoring initiator starts the call in seconds.	Y	Y
TA_MONSTARTTIMEUSEC	long	Timestamp of the monitoring initiator starts the call in microseconds.	Y	Y

Table 2 Monitor Initiator Field Names

TA_MONCLTADDR	string	The workstation client address	Y	N
TA_MONTOTALTIME	long	The end-to-end time used for a monitored call in milliseconds.	N	Y

Return Values

Upon successfully getting a FML32 buffer containing the monitoring attributes, returns 0.

Upon failure, `tpgetcallinfo()` returns -1 and sets `tperrno` to indicate the error condition.

Errors

Upon failure, `tpgetcallinfo()` sets `tperrno` to one of the following values,

[TPEINVAL]

Invalid arguments were given (for example, `msg` is NULL or `obuf` is not a valid FML32 buffer)

[TPESYSTEM]

The message input does not contain monitoring attributes. Usually this is because the call path monitoring is not turned on for the message.

[TPEOS]

An operating system error has occurred.

Example(s)

[Listing 1-1](#) provides a service-side `tpgetcallinfo` example.

Listing 1-1 Service-Side `tpgetcallinfo` Example

```
#include <stdio.h>
#include <atmi.h>
#include <userlog.h>
#include <fml32.h>
#include <tpadm.h>
#if defined(__STDC__) || defined(__cplusplus)
tpsvrinit(int argc, char *argv[])
#else
tpsvrinit(argc, argv)
```

```

int argc;
char **argv;
#endif
{

    /* tpsvrinit logic */
#ifdef __cplusplus
extern "C"
#endif
void
#if defined(__STDC__) || defined(__cplusplus)
APPSVC(TPSVCINFO *rqst)
#else
TOUPPER(rqst)
TPSVCINFO *rqst;
#endif
{
    FBFR32 *metainfo;
    int len = 0;
    /* Allocate the metainfo space */
    metainfo = tmalloc("FML32", NULL, 1024);
    if (metainfo == NULL ) {
        userlog("Memory allocation failed");
        tpreturn(TPFAIE, 0, 0, 0);
    }
    /* Get the monitoring attributes*/
    if ( tpgetcallinfo(rqst->data, &metainfo, 0) == 0 )
    {
        char *corrid;
        long laststopsec, starttimesec;
        if ((corrid = Ffind32(metainfo, TA_MONCORRID, 0, &len) ) {
            userlog("Correlation ID = %s", corrid);
        }
        len = sizeof(starttimesec);
        if (( Fget32(metainfo, TA_MONSTARTTIMESEC, &starttimesec,
&len) == 0) {
            userlog("Message beginning time = %ld", starttimesec);

```

```
        }
    }
    len = sizeof(lasttimesec);
    if (( Fget32(metainfo, TA_MONLASTTIMESEC, &lasttimesec, &len) == 0)
    {
        userlog("Message entering my queue time = %ld", lasttimesec);
    }
    }
    tpfree(metainfo);
    /* rest of service processment */
    .....
}
```

Oracle TSAM Error Messages

Oracle TSAM Error Messages

Table 2-1 lists the Oracle TSAM Error Messages.

Table 2-1 Oracle TSAM Error Messages

Message Code	Description
TSAM-0001	Cannot connect to database
TSAM-0002	Error occurred when execute SQL statement: {0}
TSAM-0003	SQL exception
TSAM-0101	Performance metrics message parsing error
TSAM-0102	No PERFDATA element found
TSAM-0103	No LOCATION element found
TSAM-0104	Acesse database error: {0}
TSAM-0105	DataServer failed to initialize: {0}
TSAM-0106	Configuration message parsing error: {0}
TSAM-0107	No DOMAIN element found
TSAM-0108	No DOMAINID element found

Table 2-1 Oracle TSAM Error Messages

Message Code	Description
TSAM-0109	No MASTER element found
TSAM-0110	No LMID element found in MACHINCE section
TSAM-0111	No RELEASE element found in MACHINE section
TSAM-0112	No PMID element found in MACHINE section
TSAM-0113	No GRPNAME element found in GROUP section
TSAM-0114	No GRPNO element found in GROUP section
TSAM-0115	No SRVNAME element found in SERVER section
TSAM-0116	No SRVID element found in SERVER section
TSAM-0117	No SVCNAME element found in SERVICE section
TSAM-0118	Original message {0} dropped
TSAM-0119	Heartbeat message parsing error: {0}
TSAM-0120	RequestServer failed to initialize: {0}
TSAM-0121	Monitoring policy reply message parsing error: {0}
TSAM-0122	No SEQUENCE found for monitoring policy reply message
TSAM-0123	Invalid Tuxedo message: {0}
TSAM-0124	No HEADER element found
TSAM-0125	No TYPE element found
TSAM-0126	No supported message type: {0}
TSAM-0127	TSAM Data Server started
TSAM-0128	TSAM Request Server started
TSAM-0129	JDBC Driver: {0}
TSAM-0130	JDBC URL: {0}
TSAM-0131	JDBC Username: {0}

Table 2-1 Oracle TSAM Error Messages

Message Code	Description
TSAM-0132	Invalid topnsvc {0} web.xml value
TSAM-0133	Invalid minapptime {0} web.xml value
TSAM-0134	Invalid recsperpage {0} web.xml value
TSAM-0135	Invalid timetrans {0} web.xml value
TSAM-0136	Invalid windowsvc {0} web.xml value
TSAM-0137	Invalid windowsrv {0} web.xml value
TSAM-0138	Invalid minapppattern {0} web.xml value
TSAM-0139	Invalid windowevt {0} web.xml value
TSAM-0140	Invalid windowhs {0} web.xml value
TSAM-0141	Invalid maxappactive {0} web.xml value
TSAM-0142	Invalid maxappdone {0} web.xml value
TSAM-0143	Invalid maxappsize {0} web.xml value
TSAM-0144	Invalid timeoutwithtuxedo {0} web.xml value
TSAM-0145	Failed to update Tuxedo objects cache from database
TSAM-0146	Failed to update Alert objects cache from database
TSAM-0160	ERROR: Unknown plug-in event type {0}
TSAM-0161	ERROR: XML data parsing error
TSAM-0162	ERROR: Missing required XML data field {0}
TSAM-0163	Plug-in generated event
TSAM-0164	INFO: {0} events deleted.
TSAM-0501	Time Zone is set to {0}
TSAM-0502	Effective Time Zone is {0}
TSAM-0503	Unknown Transport {0}

Table 2-1 Oracle TSAM Error Messages

Message Code	Description
TSAM-0504	INFO: Nothing found
TSAM-0505	BIRT not initialized
TSAM-0506	BIRT engine not available
TSAM-0507	BIRT data source problem
TSAM-0508	BIRT encountered problem
TSAM-0509	Cleanup BIRT temporary image files periodically
TSAM-0510	INFO: No monitoring data
TSAM-0511	INFO: Monitoring policy {0} is applied to Tuxedo components {1}
TSAM-0512	INFO: Monitoring policy {0} is created for Tuxedo components {1}
TSAM-0513	INFO: Monitoring set {0} is created
TSAM-0514	INFO: Monitoring set {0} is edited
TSAM-0515	INFO: Please enter Correlation ID
TSAM-0516	ERROR: Call Path data for correlation id {0} in database is corrupted
TSAM-0517	INFO: Please enter Global Transaction ID (GTRID)