

TM_MIB(5)

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SYNOPSIS	1
2	DESCRIPTION	2
3	REQUEST BUFFERS	3
4	RESPONSE STATUS	4
5	TA_ERROR error codes	5
6	T_CLIENT CLASS	6
7	T_DOMAIN CLASS	7
8	T_MACHINE CLASS	8
9	T_QUEUE CLASS	9
10	T_SERVER CLASS	10
11	T_SERVICE CLASS	11
12	T_SVCGRP CLASS	12
13	EXAMPLE SESSION OF INFORMATION FETCHING	13
14	BUGS	15
15	SEE ALSO	16
16	COPYING	17

Chapter 1

SYNOPSIS

Management classes:

- **T_CLIENT** - Client process
 - **T_DOMAIN** - Domain
 - **T_MACHINE** - Machines in domain
 - **T_QUEUE** - IPC Queues
 - **T_SERVER** - XATMI server processes
 - **T_SERVICE** - XATMI service definitions
 - **T_SVCGRP** - XATMI service instances
-

Chapter 2

DESCRIPTION

This is management information interface for Enduro/X. Currently only information reading is available, setting or changing system characteristics via this interface is not possible. Basically TM MIB interface in GET mode is designed, so that user application can call in standard way ".TMIB" service with requesting the information for particular object. These object types are encoded as classes. Thus user requests information about some class, and TMIB service provided by **tpadmsv(8)** will respond with objects found in system for particular class.

The information fetching is performed with standard **tpcall(3)** service invocations. Request and response is encoded in UBF buffer format. At request certain fields are filled (indicating the class and operation), at response the error fields are filled, object information, number of objects found and number of objects fetched. For each request cursor is open at particular **tpadmsv**, if objects fetched are less than found, then cursor_id with GETNEXT operation code shall be sent to .TMIB, to receive the next portion of data. When all data is fetched, the cursor is deleted in particular process. Cursor is valid for configured period of time, which is configured for **tpadmsv**. The page size also is configured in **tpadmsv** server configuration parameters.

Chapter 3

REQUEST BUFFERS

To initiate request/open cursor for fetching particular class of information, following UBF fields shall be filled (defined in tpadm.h/Excompat.h):

- **TA_OPERATION**: Currently supported modes are only **GET** and **GETNEXT**.
- **TA_CLASS**: Shall be set to one of the supported class which is "T_CLIENT", "T_DOMAIN", "T_MACHINE", "T_QUEUE", "T_SERVER", "T_SERVICE" or "T_SVCGRP".
- **TA_CURSOR**: In case of **GETNEXT** field must be present in order to fetch next page of the information. The **TA_CURSOR** value must be set from **GET** response same field **TA_CURSOR**.

The filled UBF buffer must be sent to **.TMIB** service.

Chapter 4

RESPONSE STATUS

General status information is encoded in following fields UBF fields:

- **TA_ERROR**: Error code, or **0** for success. See section below for detailed error codes.
- **TA_STATUS**: Status message, **OK** in case of success.
- **TA_BADFLD**: Complementary problematic fields ID, in case if error associated with request data.

If requests succeeds, then data paging information is encoded in following fields:

- **TA_OCCURS**: Number of objects loaded in response.
- **TA_MORE**: Number of objects left for fetching in other pages. I.e. if having greater than **0**, then **TA_CURSOR** shall be copied to UBF buffer requesting **GETNEXT** command which is sent to same **.TMIB** service.

Chapter 5

TA_ERROR error codes

List of error codes which .TMIB interface returns is:

- **-3 (TAEINVAL)**: Invalid fields passed to service. For example invalid cursor format. Problematic field ID is returned in **TA_BADFLD** field.
- **-9 (TAEREQUIRED)**: Field is required in request, but not present in UBF buffer. The field ID is returned in **TA_BADFLD**.
- **-10 (TAESUPPORT)**: Operation or class is not supported. Field which encoded the identifier is set in **TA_BADFLD**.
- **-11 (TAESYSTEM)**: System failure occurred during collection of information or setting the fields (memory error). See Enduro/X logs for more details.
- **-13 (TAELIMIT)**: Number of open cursors limit reached. Limits is configured in **tpadmsv(8)** configuration. Basically this means that some problem exists i.e. requesters have made several requests for information in short period of time with out fetching cursors till the end.

Chapter 6

T_CLIENT CLASS

This class returns information about client processes found in the system. Client processes are gathered from system in two different ways:

1. XATMI client information is read and decoded from open XATMI queue names.
2. Client information (including XATMI and non XATMI/generic process) is provided by **cpmsrv(8)** shared memory block.

In case if process is started by **cpmsrv** client process monitor and it operates as XATMI client, the information is merged, so that identifier of process is copied from shared memory (tag/subsection) rather than queue id.

Object fields returned for **T_CLIENT** class are following:

- Field **TA_LMID** (string..30): Cluster node ID on which client process runs. Format is number 0..255.
- Field **TA_CLIENTID** (string..128): Client process ID. **This is object key** Identifier consists of XATMI client queue, or if process is standard from Client Process Monitor (cpmsrv), then client identifier is built as **<Cluster Node Id>/<CPM Tag>/<CPM Sub Section>**. The benefit of CPM tag id is that this can be used for monitoring as there are no dynamic data in the identifier. In opposite of queue identifier, which contains the process PID. For example **"/dom2,clt,reply,atmiclt68,28765,3"**. In case if for CPM process XATMI queue exists, then ATMI context ID is extracted from queue and added to via slash context number to the id string, thus **<Cluster Node Id>/<CPM Tag>/<CPM Sub Section>/<Context ID>**.
- Field **TA_CLTNAME** (string..30): Client process name. This is name of executable, if process name is longer than 30 chars, then it is cut to 30 chars from the left.
- Field **TA_STATE** (string..15): Client process state. **ACT** - active, **DEA** - process dead.
- Field **TA_PID** (long): Client process PID.
- Field **TA_CURCONV** (long): Current number of conversations open. For CPM processes which are not part of XATMI sub-system, the value is set to **-1**.
- Field **TA_CONTEXTID** (long): Current XATMI client context ID. For non XATMI processes, the value is set to **-1**.
- Field **TA_CURTIME** (long): Unix epoch time (seconds since January 1, 1970, UTC) when process was started. This information is available only for CPMSRV booted processes.

Chapter 7

T_DOMAIN CLASS

Class of these object describes general information about current domain, that is Enduro/X instance. Information includes such information as number of queues, servers and services.

Object fields returned for **T_DOMAIN** class are following:

- Field **TA_DOMAINID** (string..30): Cluster node ID.
 - Field **TA_STATE** (string..3): Domain state: **ACT** - active, **DEA** - dead.
 - Field **TA_CURQUEUES** (long): Number of IPC queues open.
 - Field **TA_CURSERVERS** (long): Current number of XATMI servers running.
 - Field **TA_CURSERVICES** (long): Current number of services available in shared mem.
-

Chapter 8

T_MACHINE CLASS

Objects from this class describes current machine and linked machines for the application cluster. The information about other machines are gathered from the **tpbridge(8)** processes.

Object fields returned for **T_MACHINE** class are following:

- Field **TA_LMID** (string..30): Cluster node ID.
- Field **TA_CURACCESSERS** (long): Number of XATMI clients and servers present on particular cluster node. Information is available only for current node. No information is present from remote machines, and for them **-1** is returned.
- Filed **CURCONV** (long): Number of XATMI conversational sessions open. Information is only present for local machine, for remote machines **-1** is returned.
- Filed **TA_STATE** (string..3): Cluster node state. **ACT** - active, **INA** - tpbridge connector is not running, **PEN** - tpbridge connector is running, but connection is not established yet.

Chapter 9

T_QUEUE CLASS

These objects describe the IPC queues open in the system. Note as Enduro/X is capable of working with different kind of queues, either Posix on System V, the output might be different. But basically all queues within Enduro/X are represented as strings. For System V queues queue id is returned too.

Object fields returned for **T_QUEUE** class are following:

- Field **TA_LMID** (string..30): Cluster node ID.
 - Field **TA_RQADDR** (string..128): Queue name. For Posix Queues, this is queue name. For other approaches, this is Enduro/X representation of the queue. **This is key of object.**
 - Field **TA_STATE** (string..3): Queue state. **ACT** - active.
 - Field **TA_NQUEUED** (long): Number of messages currently enqueued.
 - Field **TA_RQID** (long): System V msqid, as from `ipcs -q` output.
-

Chapter 10

T_SERVER CLASS

This class describes XATMI servers currently defined on current cluster node.

Object fields returned for **T_SERVER** class are following:

- Field **TA_LMID** (string..30): Cluster node ID.
- Field **TA_SRVID** (long): Server instance id, as defined in <srvid> tag in **ndrxconfig.xml(5)** with min/max settings applied. **This is the key of object.**
- Field **TA_RQADDR** (string..128): Request address queue. Used only in System V mode.
- Field **TA_STATE** (string..3): **ACT** - Server process is working. **RES** - Server process is starting. **CLE** - Server process is stopping. **DEA** - Server process is dead / stopped.
- Field **TA_TIMERESTART** (long): **ndrxd(8)** sanity cycles from last state change. Thus if server **DEA**, it will be sanity cycles since **CLE** state change.
- Field **TA_PID** (long): This is Unix process PID. If process is dead, then pid number is left from last session it was running.
- Field **TA_SERVERNAME** (string..78): This is server name as defined in "<server>" tag in **ndrxconfig.xml(5)**.
- Field **TA_CLOPT** (string..256): This is actual binary name which is booted for the server process. The binary name either matches the **TA_SERVERNAME** or is extracted from "<cmdline>" tag.
- Field **TA_GENERATION** (long): This is number of consecutive restarts performed of the process. I.e. if processes is in dead stat, then number of attempts was made to boot it up till the current moment.

Chapter 11

T_SERVICE CLASS

This class describes the services defined and available in the system, with out dynamic information.

Object fields returned for **T_SERVICE** class are following:

- Field **TA_SERVICENAME** (string..30): Service name as defined in shared memory. **This is object key.**
 - Filed **TA_STATE** (string..3): **ACT** - service is available, **INA** - service is not available (was advertised, but currently any server is shutdown).
-

Chapter 12

T_SVCGRP CLASS

This class describes dynamic information currently available about the services in the cluster system. Information is retrieved as request from **ndrxd(8)** process about it's visibility and knowledge of the current system setup.

Object fields returned for **T_SVCGRP** class are following:

- Field **TA_LMID** (string..30): Cluster node ID.
- Field **TA_SERVICENAME** (string..30): Service name as defined in shared memory.
- Field **TA_SRVGRP** (string..30): **This is key of the object.** Format <Cluster node id>/Server id which provides the service.
- Field **TA_STATE** (string..3): **ACT** - service is available.
- Field **TA_SRVID** (long): Server ID which provides the service.
- Field **TA_SVCRNAM** (string..30): Service routine name (function) associated with service.
- Field **TA_NCOMPLETED** (long): Number of requests completed. Includes succeed and fail.
- Field **TA_TOTSUCCNUM** (long): Total number of succeed requests processed.
- Field **TA_TOTSFAILNUM** (long): Total number of failed requests processed.
- Field **TA_LASTEXEETIMEUSEC** (long): Number of microseconds spent for last service call.
- Field **TA_MAXEXEETIMEUSEC** (long): Max microseconds spent for particular service instance call.
- Field **TA_MINEXEETIMEUSEC** (long): Minimum microseconds spent for particular service instance call.

Chapter 13

EXAMPLE SESSION OF INFORMATION FETCHING

The example test is performed with "ud" utility.

Request (initial):

```
$ ud < test.ud
SENT pkt(1) is :
TA_CLASS      T_SERVICE
TA_OPERATION   GET
```

Response:

```
RTN pkt(1) is :
TA_ERROR      0
TA_MORE      8
TA_OCCURS     10
TA_CLASS      T_SERVICE
TA_CURSOR     .TMIB-1-2660_SC000000001
TA_OPERATION   GET
TA_SERVICENAME RETSOMEDATA
TA_SERVICENAME UNIX2
TA_SERVICENAME UNIXINFO
TA_SERVICENAME .TMIB-1-2660
TA_SERVICENAME @CPMSVC
TA_SERVICENAME @TM-1-1-810
TA_SERVICENAME @TPRECOVER
TA_SERVICENAME @TM-1-1-1650
TA_SERVICENAME ECHO
TA_SERVICENAME .TMIB
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATE      ACT
TA_STATUS     OK
```

Request next (initial):

```
$ ud < test.ud
SENT pkt(1) is :
TA_CLASS          T_SERVICE
TA_CURSOR          .TMIB-1-2660_SC000000001
TA_OPERATION       GETNEXT
```

Response next (and complete):

[illegible]

Chapter 14

BUGS

Report bugs to support@mavimax.com

Chapter 15

SEE ALSO

tpadmsv(8) cpmsrv(8) ndrxd(8) xadmin(8)

Chapter 16

COPYING

© Mavimax, Ltd