

TPCACHESV(8)

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SYNOPSIS	1
2	DESCRIPTION	2
3	DEFINITIONS	3
4	EXIT STATUS	4
5	BUGS	5
6	SEE ALSO	6
7	COPYING	7

Chapter 1

SYNOPSIS

`tpcachesv`

Chapter 2

DESCRIPTION

This is special XATMI server used for supporting cluster operations when receiving event of message adding or removing from other cluster nodes caches. Also this server is backend for command line tools such as *xadmin cs* (cache show), *xadmin cd* (cache dump) and *xadmin ci* (cache invalidate).

For cluster operations, cache server subscribes to events specified in *subscr* parameter for database. Per cluster node service name are fixed. Thus servers shall be booted with all CC tags set, so that it would have a visibility of all databases. Full visibility is required for command line tools processing and incoming event processing. Multiple copies of server can be booted. They all will subscribe to same events. The **tpesrv** will ensure the single event delivery is performed to the same service name.

During the event processing, following events can be consumed (by regexp mask they can be combined):

1. @CPNNN/F*/S+ - event is received if new data is put to cache. The data buffer includes full message which is being cached.
2. @CDNNN/F*/S+ - event is received when data is being deleted from cache. The delete is generated either from **tpcached** (when records reaches limit, expires or service is not available. In this case full or subset of message is broadcasted (according to *delete* parameter of service cache definition). The delete message is processed according to local cache definition. Service is found and then record is matched according to rules. This event also can be received in cases when "invalidate their" cache performs invalidate. If flags (F*) contains letter G, then if record is part of keygroup (it is key item), then whole group is deleted.
3. @CKNNN/F*/D+ - drop (kill) the database. This event is published when from xadmin tool **xadmin ci -d D+*, the database is being dropped. The event notification contains NULL buffer. The *D+* is database name. Note that in case of keygroup (between group db and key item db) there no linkage. So drop will clean up the database specified, and not any linked record or db.
4. @CMNNN/F*/D+ - delete database (D+) records by regular expression mask over the cached keys. The data buffer sent with this event is **UBF** typed, which contains one field *EX_CACHE_OPEXPR* - regular expression mask to match keys with. If key is matched, then record is deleted.
5. @CENNN/F*/D+ - delete records by key. The database name is encoded in *D+*. The key to delete is encoded in event buffer, which is **UBF** type. Field *EX_CACHE_OPEXPR* keeps the key to delete. Before deleting record, it is read from database. The flags are checked for record. If record is part of the keygroup and it is keyitem. Then data buffer is recovered from data and group key is built. Key group is updated (key removed) and key item is removed from item db.

tpcachesv server advertises following services:

1. @CACHEEVNNN - event processing target.
2. @CACHMGNNN - *xadmin* command back-end.

The above *NNN* stands for cluster node id, prefixed with zeros.

Chapter 3

DEFINITIONS

Keygroup is separate database where records can be grouped. For example by user id. The group record holds the **UBF** buffer with string key occurrences which are linked in the group.

Keyitem is linked record to the group.

Chapter 4

EXIT STATUS

0 Success

-1 Failure

Chapter 5

BUGS

Report bugs to support@mavimax.com

Chapter 6

SEE ALSO

`ex_env(5)` `ex_adminman` `tpcachebtsv(8)` `xadmin(8)` `tpcached(8)`

Chapter 7

COPYING

© Mavimax, Ltd