TAFJ-Lock Manager

R14/R15

10/3/2015

Temenos

Amendment History:

Revision	Date Amended	Name	Description
1	1st April 2011	TAFJ team	Initial version
2	7st February 2012	H. Aubert	R12GA review
3	16th January 2013	R. Vincent	R13GA review
4	18th Feb. 2014	R. Vincent	R14GA review
5	11th April 2014	R. Vincent	Add MSQL Locking
6	15th April 2014	H. Aubert	R14GA review
7	10th March 2015	H. Aubert	R15 AMR review

Copyright

Errata and Comments

If you have any comments regarding this manual or wish to report any errors in the documentation, please document them and send them to the address below:
Technology Department

Temenos Headquarters SA
2 Rue de l'Ecole-de-Chimie,
CH - 1205 Geneva,
Switzerland

Tel SB: +41 (0) 22 708 1150
Fax: +41 (0) 22 708 1160

Please include your name, company, address, and telephone and fax numbers, and email address if applicable. TAFJdev@temenos.com

Table of Contents

TAFJ tLockManager

Introduction

Pessimistic locking is needed within T24. Pessimistic locking assumes that the lock is not taken and a specific request is needed to take the lock. This can be accomplished in TAFJ by either using tLockManager (a process outside of the database) or one of the mechanisms for database locking.

tLockManager is a Lock Manager designed for TAFJ needs. It handles multiple databases and offers a monitoring console.

Syntax

To launch the tLockManager, use this syntax:

```
tLockManager [-p<port>][-m 1|2] [--verbose][-r]
```

Options:

-i	: run interactively (not in background)
-p<port>	: port on which to connect to
-r	: accept remote command
-m	: 1 for normal 2 for hash
--verbose	: run in verbose mode

The default port it will listen on is 7425.

Example

To launch it, go in the bin directory of your TAFJ installation and just type

> tLockManager –i (if you want to run in background mode then you can remove -i)

The Output will be:

```
JMX LockManager set to = false
Lock server PROC is listening on  (-p) : 7425
```

Once done, you can configure your instance by editing your <project>.properties file with following fields:

Under TAFJ Locking mechanism setup:

```
#********************************************************************
#
#  TAFJ Locking mechanism
#
#********************************************************************
# Tafj locking mode
# Could be : JDBC | ORCL | MSQL | PROC | PROCHASH | MEM | NONE
# JDBC : locks managed in database uses LOCK_RECORDS table.
# ORCL : locks managed in database using DBMS_LOCK Oracle specific package. Recommended
solution for Oracle.
# MSQL : locks managed in database using DBMS_LOCK MS-SQL specific package. Recommended
solution for MS-SQL.
# PROC : TAFJ lock manager, external process, server accepting client connections on a port.
# PROCHASH : TAFJ lock manager, external process, server accepting client connections on a
port (for performance).
# MEM : Memory mode version of the TAFJ lock manager.

temn.tafj.locking.mode                                     = JDBC

# Host name or IP of where is the tLockManager daemon
#
temn.tafj.locking.hostname                                 = 127.0.0.1

# TCP port of the tLockManager daemon is listening on
#
temn.tafj.locking.port                                     = 7425

# Bring the current call stack over to the LockManager?
#
temn.tafj.locking.callstack                                = false

# if we want to override the URL as unique key for the DBInstance
# we have to give a name here (eg myDatabase). This is usefull
# if the DBServer has multiple adapters. The URL's are different
# but this is still the same Database.
# only for PROC and PROCHASH
#
temn.tafj.locking.name                                     =
```

TAFJ JDBC Lock Manager

TAFJ can take locks by using an underlying database table. This table has the format:

```
SQL> desc lock_records;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------------------
 RECID                                     NOT NULL VARCHAR2(255)
 SESSIONID                                          VARCHAR2(50)
```

This method works well within an application server context where a separate datasource can be set up for locking. However, it performs worse than tLockManager and ORCL locking. Edit properties file and set:

```
temn.tafj.locking.mode  = JDBC
```

TAFJ Oracle Lock Manager

This method of locking works only with the Oracle database and leverages the DBMS_LOCK package within Oracle to take TAFJ application locks. It is the preferred method when using Oracle database and performs slightly faster than tLockManager. Edit properties file and set:

```
temn.tafj.locking.mode  = ORCL
```

TAFJ MSQL Lock Manager

This method of locking works only with the MS Sql Server database and leverages packages within Sql Server to take TAFJ application locks. It is the preferred method when using Sql Server database and performs slower than tLockManager but keeps all locking within the database. Edit properties file and set:

```
temn.tafj.locking.mode  = MSQL
```

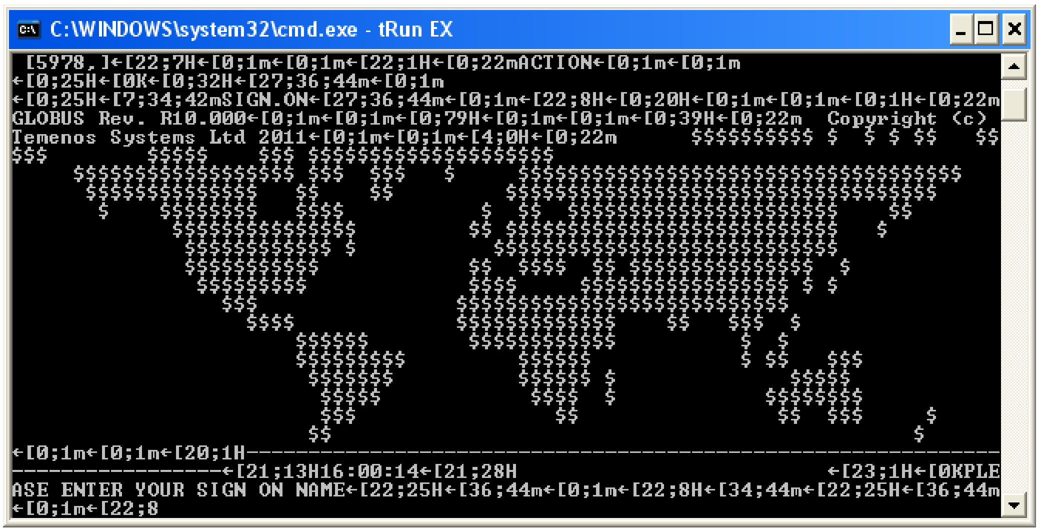
Monitoring

The lock manager offers several ways to monitor locks. One easy way of seeing what is going on is to run the EX Basic File using tRun

Example:

```
tRun EX
```

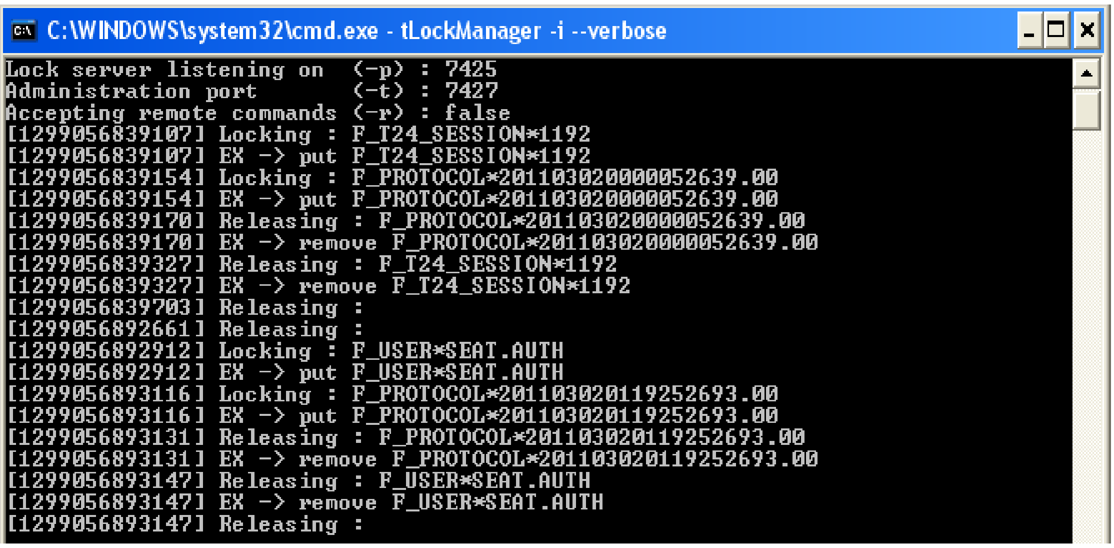
The output will be:



```
 [5978.]←[22;7H←[0;1m←[0;1m←[22;1H←[0;22mACTION←[0;1m←[0;1m
←[0;25H←[0K←[0;32H←[27;36;44m←[0;1m
←[0;25H←[7;34;42mSIGN.ON←[27;36;44m←[0;1m←[22;8H←[0;20H←[0;1m←[0;1m←[0;1H←[0;22m
GLOBUS Rev. R10.000←[0;1m←[0;1m←[0;79H←[0;1m←[0;1m←[0;39H←[0;22m  Copyright (c)
Temenos Systems Ltd 2011←[0;1m←[0;1m←[4;0H←[0;22m      $$$$$$$$$$ $  $ $ $$   $$
$$$        $$$$$    $$$ $$$$$$$$$$$$$$$$$$$$
     $$$$$$$$$$$$$$$$$$ $$$  $$$   $     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      $$$$$$$$$$$$$$   $$    $$         $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
       $    $$$$$$$$   $$$$           $  $$  $$$$$$$$$$$$$$$$$$$$$$    $$
             $$$$$$$$$$$$$$$         $$ $$$$$$$$$$$$$$$$$$$$$$$$$$$   $
              $$$$$$$$$$$$ $           $$$$$$$$$$$$$$$$$$$$$$$$$$$$
              $$$$$$$$$$$            $$  $$$$  $$ $$$$$$$$$$$$$$$  $
               $$$$$$$$$             $$$$     $$$$$$$$$$$$$$$$$ $ $
                 $$$                $$$$$$$$$$$$$$$$$$$$$$$$$$$
                   $$$$             $$$$$$$$$$$$$    $$   $$$  $
                       $$$$$$        $$$$$$$$$$$$          $  $
                       $$$$$$$$$         $$$$$$            $ $$   $$$
                        $$$$$$$          $$$$$$ $              $$$$$
                         $$$$$            $$$$  $            $$$$$$$$
                         $$$                $$                $$  $$$     $
                        $$                                               $
←[0;1m←[0;1m←[20;1H-------------------------------------------------------------
-----------------←[21;13H16:00:14←[21;28H                         ←[23;1H←[0KPLE
ASE ENTER YOUR SIGN ON NAME←[22;25H←[36;44m←[0;1m←[22;8H←[34;44m←[22;25H←[36;44m
←[0;1m←[22;8
```

If you’re running tLockManager in **verbose** mode (ie tLockManager --verbose -i) then, you will get some information while running **EX** basic.

In Verbose Mode:



```
Lock server listening on  (-p) : 7425
Administration port       (-t) : 7427
Accepting remote commands (-r) : false
[1299056839107] Locking : F_T24_SESSION*1192
[1299056839107] EX -> put F_T24_SESSION*1192
[1299056839154] Locking : F_PROTOCOL*2011030200000052639.00
[1299056839154] EX -> put F_PROTOCOL*2011030200000052639.00
[1299056839170] Releasing : F_PROTOCOL*2011030200000052639.00
[1299056839170] EX -> remove F_PROTOCOL*2011030200000052639.00
[1299056839327] Releasing : F_T24_SESSION*1192
[1299056839327] EX -> remove F_T24_SESSION*1192
[1299056839703] Releasing :
[1299056892661] Releasing :
[1299056892912] Locking : F_USER*SEAT.AUTH
[1299056892912] EX -> put F_USER*SEAT.AUTH
[1299056893116] Locking : F_PROTOCOL*20110302011925 2693.00
[1299056893116] EX -> put F_PROTOCOL*20110302011925 2693.00
[1299056893131] Releasing : F_PROTOCOL*20110302011925 2693.00
[1299056893131] EX -> remove F_PROTOCOL*20110302011925 2693.00
[1299056893147] Releasing : F_USER*SEAT.AUTH
[1299056893147] EX -> remove F_USER*SEAT.AUTH
[1299056893147] Releasing :
```

With the TAFJTrace.properties

In <TAFJ_HOME>/conf/TAFJTrace.properties there is a log4j logger setup for LOCKING. To set maximum level of debugging, set the below line:

```
log4j.logger.LOCKING=DEBUG,  locking
```

The log file is <TAFJ_HOME>/log/locking.log

With Eclipse:

By right clicking on a TAFJ Eclipse project, choose « Lock Management View ». From this screen, you can monitor locks.