Applix TM1 Technical Bulletin

Using the TM1 Top Utility to Monitor TM1 Server Activity

Date: October 30, 2006

Relevant TM1 Versions: 8.4.4, 9.0 SP2

This technical bulletin describes TM1 Top, a new utility that allows dynamic monitoring of threads running in an instance of the TM1 server.

Overview of TM1 Top

The TM1 Top utility is supported in versions 8.4.4 and 9.0 SP2 of the TM1 server, allowing you to dynamically monitor the threads running in an instance of the TM1 server. This utility is similar to the Unix “top” utility which allows dynamic monitoring of the processes running on a given system.

Currently, the TM1 Top utility is not distributed with the standard TM1 installation. It is available only from Customer Support by request, on an as-needed basis.

TM1 Top is a stand-alone utility that runs within a console (command) window on a Windows machine. It is designed to make minimal demands on the TM1 server and the supporting network and system. With the exception of a user-initiated verification process, TM1 Top does not use any cube or dimension resources in the TM1 server, and does not use or interact with the data or metadata locks on the TM1 server. The server-side support code runs in a separate light thread to allow TM1 Top to report server state even if the server is frozen for some reason.

The Tm1top.ini File

Before running TM1 Top, edit the initialization file Tm1top.ini. The Tm1top.ini file is an ASCII file that specifies environment information for the TM1 Top utility. This file should be located in the same directory as the TM1 Top executable file.

The version of the Tm1top.ini file which ships with the product contains comments explaining all of the fields.

A sample of the Tm1top.ini file is shown below.

```
servername=sdata
adminhost=localhost
refresh=5
logfile=c:\temp\tm1top.log
logperiod=10
logappend=T
```

The parameters in the Tm1top.ini file are described in the following table.

Parameter	Description
ServerName	TM1 server name. This is the ServerName from the server configuration file, Tm1s.cfg.
AdminHost	Machine name or IP address of the computer running the Admin Server. Default value is localhost.
Refresh	Time interval between refreshing of the TM1 Top display. Format is: refresh=*nn* where *nn* is the time interval in seconds. Default value is 2 seconds.

Parameter	Description
LogFile	Specifies the path and file to which the log information is written. As TM1 Top is running, the status lines can be continuously written to this ASCII file so that the activity over time can be monitored. Format is: logfile=*path-to-log-file* where *path-to-log-file* must specify the complete path, filename, and file extension. A default filename and file extension is not provided. **NOTE:** You must also specify a value for the LogPeriod parameter to enable logging.
LogPeriod	Specifies the time interval between updates being written to the log file. This value should be a multiple of the Refresh time parameter. The format for this parameter is: logperiod=*nn* where *nn* is the number of elapsed seconds between updates to the log file. Setting this to zero disables logging. For example, if the screen Refresh is set to 2 seconds, LogPeriod could be set to 10 seconds so that every fifth screen display will be output to decrease the amount of data written to the file. The default value is zero for no logging.

Parameter	Description
LogAppend	Specifies if log data is appended to the log file that is set with the LogFile parameter, or if the file is overwritten when a new session is started. If the LogAppend parameter is not set, the existing log file will be over-written. Valid values are T and F as shown in the following examples: • If you set LogAppend=T, log data is appended to the log file specified by the LogFile parameter. • If you set LogAppend=F, the existing log file will be over-written. The default value is F, which overwrites the existing log file.

Running TM1 Top

TM1 Top should be started in a console (command) window. Obtain a copy of the TM1 Top utility from Customer Support, and then configure and run the utility as follows:

1. Edit the initialization file, Tm1top.ini, as described in section “The Tm1top.ini File”.
2. Place the Tm1top.exe and Tm1top.ini files in the TM1 <install_dir>\bin directory. For example, C:\Program Files\Applix\bin.
3. Run the Tm1top.exe executable file from a command prompt to start the TM1 Top utility.

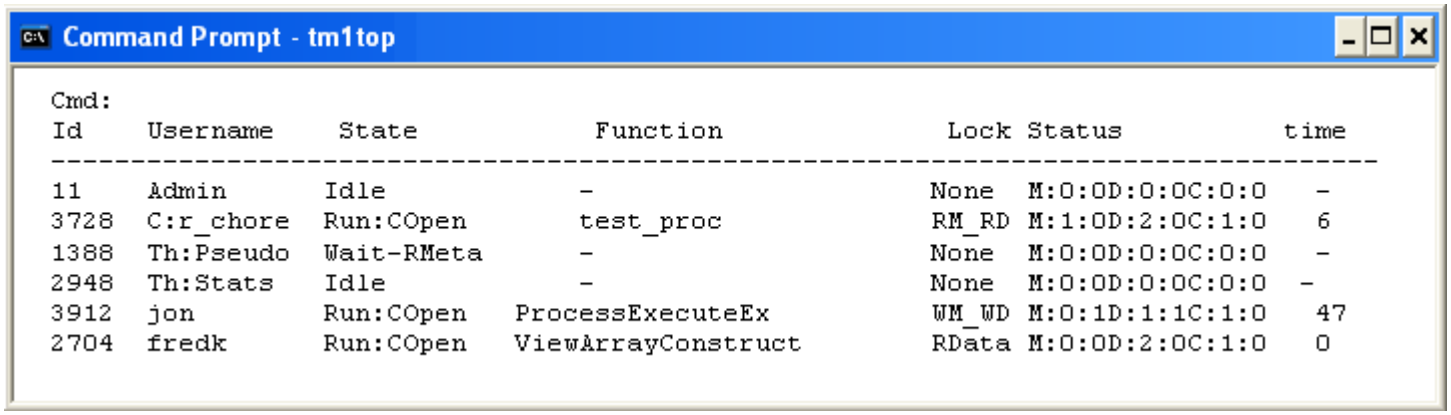
   `tm1top.exe -i` *path-to-the-initialization-file*

   Use the –i command line parameter to specify the location of the Tm1top.ini file. By default, the Tm1top.ini file is read from the same directory that contains the TM1 Top executable.

TIP: To see more lines or a wider display, re-size the console window and use a smaller font size *before* starting the TM1 Top utility. TM1 Top will utilize the size of the window as it exists when TM1 Top is started. You can scroll the view up and down when the utility is running by using the up and down arrow keys, but it is more convenient to see all threads in a single window if possible.

Viewing TM1 Top Status

When TM1 Top is running, the display looks something like the following.



If the display fills the entire height of the console window, you can use the up and down arrow keys on your keyboard to go to the next or previous page within the console window.

The status fields displayed by the TM1 Top utility are described in the following table.

Field Name	Description
Id	The thread ID of the underlying system thread in the TM1 server.
Username	This can be either the name of a user logged into the TM1 server or the name of a chore or system thread running on the TM1 server. • **User** - Name of an actual user that is logged into the TM1 server. • **Chore Thread** - If the thread is a chore thread, the Username field displays the name of the chore in the format C:*chore-name* • **System Thread** - If the thread is a system thread, then the Username field displays the name of the system thread in the following format: Th:*name of system thread* The system thread can be one of two possible types: Pseudo or Stats. For example: Th:Pseudo, or Th:Stats. The Pseudo thread is used to clean up UDC objects, and the Stats thread is the performance monitor thread.

Field Name	Description
State	The current processing state of a specific user or thread. This consists of two sub-fields in the following format: *general processing state* : *calculation cube state* The general processing state can be either Idle, Processing or a state of waiting for a lock as follows: • Idle – thread is idle • Processing – thread is processing • Wait-RData – thread is waiting for a read lock on the data • Wait-RMeta – thread is waiting for a read lock on the metadata • Wait-WData – thread is waiting for a write lock on the data • Wait-WMeta – thread is waiting for a write lock on the metadata If the thread is holding or using the calculation cube, the "cac" state is appended to the general processing state. The cac state can be one of following values: • CWaitRD – waiting for read database lock • CWaitRC – waiting for read Clock • CWaitWC – waiting for write Clock • COpen – the cac is open • CSave – about to do a cac save • CPub – in the process of "publish"

Field Name	Description
Function	The current API function that the thread is executing. If the thread is running a chore, as opposed to a function from an interactive user, the Function field will display the currently executing process under that chore.

Field Name	Description
Lock Status	The current lock held by the thread. This field displays a main value, followed by counters that are used primarily to debug deadlock situations. Format for the Lock Status field is: *<main lock value> <read and write counters>* Possible values for the main lock value include: • None – no lock currently held • RMeta – a read-metadata lock is held • WMeta – a write-metadata lock is held • RData – a read database lock is held • WData – a write database lock is held • WM_WD – write metadata and a write database locks are held • RM_RD – read metadata and read database locks are held • WD_bat – a write database batch-mode lock is held • WM_RD – write metadata and read database locks are held The read and write counters make up the second part of the Lock Status field. This sub-field displays the read and write counters for each of the three main locks for a specific user or thread. The format is: *M:r:w D:r:w C:r:w* Where M = metadata lock, D = database lock, C = calculation cube (cac) lock, r = read count, w = write count For example, the string M:0:1D:1:1C:1:0 indicates that this user has counts of 1 for the metadata write count, 1 for both the read and write database counts, and a 1 count for the read cac counter.

Field Name	Description
Time	The current time, in seconds, that the API call has been processing.

TM1 Top Commands

You can enter the following commands at the command prompt:

Command	Description
X	Exits the TM1 Top utility.
W	Writes the current display to a file.
H	Displays help text.
V	Verifies the current session so that subsequent commands to cancel processing of other threads may be made.
C	Cancels processing in another thread. **NOTE:** Before using the Cancel command, the session must first be "verified" by using the V (Verify) command. For more information on using the Cancel command, see the section "Canceling a User's Thread Processing".
Up, Down arrow keys	Use the up and down arrow keys to scroll the status display if there are more lines in the status than the height of the console window.

Canceling a User's Thread Processing

A system administrator can use TM1 Top to request the cancellation of certain thread operations or functions that are processing-intensive. Currently, TM1 Top can attempt to cancel the following types of thread functions:

- TI processing
- Constructing or calculating a view

This functionality provides a way to cancel an action that is negatively impacting other threads by consuming too much of the server's processing time. For example, a TI process that gets stuck in a loop, or when a user tries to compute a very large and demanding view.

To cancel a thread, you must use the Cancel and Verify commands together, as described below.

Understanding the Cancel Process

The Cancel process requires that the thread to be canceled is actively running. A running thread will check to see if its "exit indicator" has been set, and if set, the thread will exit out of the current operation. The cancel operation is actually implemented by the thread itself. Therefore, you can not use the Cancel command to cancel processing of a thread that is deadlocked against another thread, both of which are waiting for a lock resource, because the thread will not be running and will not be checking its exit indicator.

Understanding the Verify Process

TM1 Top was designed to run without requiring a user to log on to the server. This allows TM1 Top to run against a server which might be resource deadlocked. If a login procedure *was* required, then the login process would also hang because it requires access to cubes and other data that are manipulated under the resource locks.

However, it is not desirable to allow just any user running TM1 Top to cancel user threads. As a compromise between a full login procedure and no security at all, TM1 Top requires that the TM1 Top user must first be "verified" as having administrator access before the Cancel command can be used. This verification process is initiated using the V (verify) command. The process can be done in advance and is maintained for the life of the TM1 Top connection.

Using the Verify and Cancel Commands

To verify a TM1 Top session and cancel a thread:

1. Enter V to display the following prompt to verify a TM1 Top session:

   ```
   Verify Admin Access
   Admin User Name:
   ```

2. Enter the login name of an admin user.

   The following prompt appears:

   ```
   Verify Admin Access
   Admin Password:
   ```

3. Enter the password for the admin user.

   If you entered a valid admin user name and password, then the following confirmation appears:

   ```
   Admin access verified
   Press any key to continue:
   ```

4. After the session has been verified, enter C to initiate the Cancel process.

   This brings up the prompt:

   ```
   Cancel processing of a thread
   Thread-id:
   ```

5. Enter the ID of the thread to be canceled. You can find the thread ID displayed in the first column of the TM1 Top display.