



## ASG-TMON<sup>®</sup> for DB2 Release Notes

### Version 4.0

March 18, 2009  
TD21100-40

This publication fully describes the modifications made to ASG-TMON for DB2 (herein called TMON for DB2) and the ASG-TMON for DB2 SQL Analyzer feature (herein called SQL Analyzer) since the release of Version 4.0. This set of release notes supersedes the November 21, 2008, version and is cumulative, with updated or new information marked with change bars.

## ISP Updates and Service Packs

ASG encourages you to visit the Intelligent Support Portal (ISP): <http://www.asg.com/support/support.asp>. The ISP allows you to verify whether any product or documentation revisions, new maintenance, or service packs apply to this product release.

## Overview

This table provides a summary of the product changes and documentation updates discussed in this set of release notes:

Product or Documentation Change	Page
ACCUMACC=NO recommendation for DB2 Version 8.1 and 9.1.	2
SQL Analyzer package log (D7) record limitations.	4
Resource limitation.	5
Enhanced partition number support for DB2 Version 8.1.	5
Improved operation of the lock contention exception monitor categories.	6
Documentation update to the <i>ASG-TMON for DB2 SQL Analyzer Reference Guide</i> with regards to SQL Analyzer Package or SQL Statement logging.	8

<b>Product or Documentation Change</b>	<b>Page</b>
Documentation updated for a TSS message reference.	9
Counts now provided by event, rather than by IFCID. Support is retained for IFCID-based counts. The EVENTCOUNTS parameter and command provide access to this support.	10
Improved nested activity support for Current Thread Activity and Collection Analysis screens.	11
The new DB2MAINT parameter and command, which provide better control over TMON for DB2's behavior when an unexpected IFCID data segment length is encountered.	53
Tips about navigating from TMON for CICS TS to TMON for DB2.	55
A description of the new IOTHROTTLER startup parameter.	55
An explanation of TMON for DB2 dump handling.	56
An overview of SQL Analyzer exclusion profiling.	61
A discussion of TMON for DB2 toleration support for DB2 Version 8.1.	64
A description of the separation of the CANCEL and KILL functions and the effects of the change on product screens, installation, security, and messages.	65
Instructions for changing the system clock and a discussion of the impact of the adjustment on TMON for DB2.	69

## **ACCUMACC=NO Recommendation and SQL Analyzer Rework**

The ACCUMACC<sup>1</sup> DSNZPARM introduced with DB2 Version 8.1 creates a data correlation problem within TMON for DB2 that goes beyond the limitations previously stated in the *ASG-TMON for DB2 Version 4.0 Release Notes* dated January 18, 2007. ASG now recommends that sites using TMON for DB2 to monitor a DB2 subsystem from Version 8.1 or later use a DSNZPARM parameter setting of ACCUMACC=NO.

---

1. ACCUMACC is a DB2 accounting parameter used to roll up distributed thread information for SMF record logging. It was introduced with DB2 Version 8.1 to limit the amount of information recorded for DDF and RRSF type threads. The default value provided by IBM is 10, which means that DB2 will basically combine the information accumulated from 10 distributed threads into a single summarized IFCID 003 thread accounting record. As stated by IBM, the result will be a consolidation of data from other transactions using the same user (as `USER` is defined with the ACCUMUID DSNZPARM) in exchange for logging less thread data to SMF.

In response to a growing number of SQL Analyzer issues reported by customers, ASG determined that a full analysis of the function was needed. Recent maintenance had failed to resolve a number of the reported problems. This analysis effort produced multiple resolutions for the reported SQL Analyzer problems. The effort also uncovered issues with the ACCUMACC DSNZPARM, which resulted in the recommendation stated above. These are some of the ACCUMACC-related problems that were uncovered:

- Threads out of date/time sequence order in the Thread Activity (Collection Analysis) screen (option 5.2).
- The SQL LOGR field displaying as zero when SQL statements are expected.
- SQL statement logging stopping for no apparent reason.
- SQD counts (an SQL Analyzer buffer) in the Current Global Statistics screen (option 8.2.3) reaching maximum values as a result of orphaned work areas.
- Buffer shortages of various kinds across the TMON for DB2 product.
- Incorrect thread counts displayed on the DB2 Subsystem Selection screen.
- Missing or erroneous data related to thread execution.
- Abends as a result of SQL Analyzer initialization or profiles.

Using a DSNZPARM setting of ACCUMACC=NO will affect only DDF/RRSAF threads in DB2 Version 8.1 and later. The result will be a one-for-one correlation of DDF/RRSAF threads to thread accounting records logged by TMON for DB2 (or SMF). This setting produces the same behavior found in DB2 Version 7.1 and is recommended by IBM to enable monitor thread detail accounting.

## **SQL Analyzer Fixes**

Sites using SQL Analyzer should apply PTF TE03001 and all related maintenance, and should read and adhere to all HOLDS. This will pull in critical maintenance, including PTFs TE02789, TE02887, TE02904, TE02905, and TE02906. In addition, TE03001 addresses some crucial setup and optimization information for SQL Analyzer with documentation (since some symptoms addressed by TE03001 also can be caused by other issues with setup, including resource constraints like buffer shortages).

**Note:**

Starting with PTF TE03077, the default numbers of SQL Analyzer buffers are maximized; therefore, there is no need to override these defaults.

Under rare circumstances (usually only on request from ASG Customer Support), you can reduce the number of buffers for diagnostic purposes using these startup parameters:

```
SQLASQDS=nnnnn (min=100; max=default=40252)
SQLASQRS=nnnnn (min=100; max=default=27581)
SQLASQPS=nnnnn (min=500; max=default=41227)
```

If a parameter is less than the allowed minimum or greater than the maximum, it is set to its maximum value, which is the TMON for DB2 default.

---

Also read “[SQL Analyzer Package Log \(D7\) Record Limitations.](#)” If you still have issues with SQL Analyzer after following the steps in this section (“[SQL Analyzer Fixes](#)”) and implementing *all* optimization recommendations outlined in the TE03001 HOLD DOC and “[SQL Analyzer Package Log \(D7\) Record Limitations.](#)” contact ASG Customer Support for assistance.

## SQL Analyzer Package Log (D7) Record Limitations

Since its inception, TMON for DB2 4.0 has had a limit on the number of SQL statements it can collect in an SQL Analyzer package log (D7) record. This section documents that limitation and provides guidelines for managing SQL Analyzer logging.

SQL Analyzer can log statistics for up to 120 SQL statements in the D7 record; either summarized by SQL statement or, if you are collecting package table statistics, by SQL statement and table. To ensure that no data is lost because of the D7 record size constraints, set LOG PKG TABLE STATISTICS? in the SQL Analyzer Controls screen (option 11.1) to N.

To obtain the most meaningful data from SQL Analyzer when you are collecting package information, be sure to create very specific, targeted SQL Analyzer profiles. ASG recommends that you include SQL *target* limits in your SQL Analyzer profiles when requesting package records (e.g., the number of GETPAGES to exceed to qualify for collection or the amount of DB2 CPU time to be exceeded to qualify for collection). Your specified limits will restrict the collection of package SQL statistics only to statements that are most costly and might require tuning, and will exclude collection of statements that are not significantly impacting DB2 performance. In addition, be sure to follow all other SQL Analyzer tuning recommendations (as detailed in PTF TE03001). One of the most important recommendations is ensuring that GROUP REPEATING FETCHES and GROUP REPEATING STATEMENTS are both set to Y on the SQL Analyzer Controls screen (option 11.1) to streamline and optimize package data collection.

## Resource Limitation

The internal monitor storage allocation for resources that a single thread accesses is managed by the INFLIGHT THREAD RESOURCES value, which you specify on the DB2 Monitor Internal Adjustments (Page 1 of 2) screen (accessed through option 8.3). If this storage allocation is insufficient, TMON for DB2 might not collect all the resources a thread accesses. There is a limit to the number of resources that TMON for DB2 collects—and, subsequently, displays or logs—which is set internally to 160. If a thread accesses more than 160 distinct resources, TMON for DB2 collects or logs only the first 160.

---

**Note:**

Each partition in a single tablespace is counted as a distinct resource.

---

## Enhanced Partition Number Support for DB2 Version 8.1

PTF TE02913, which corrects partition number notation on I/O for threads, provides DB2 Version 8.1 toleration support that will set the partition number value to 255 for partition numbers that are beyond the DB2 Version 7.1 limit of 254 (from 255 through 4096). Full DB2 Version 8.1 support for this DB2 feature is available in TMON for DB2 4.1. (TMON for DB2 Version 4.0 provides only toleration support for DB2 Version 8.1. See [“TMON for DB2 Support for DB2 Version 8.1” on page 64.](#))

Partition numbers are extracted (in DB2 Version 7.1 and earlier, and in Version 8.1 prior to PQ90547) from the high order bits of the page number obtained from IFCID006. The number of bits that contain the partition number varies depending on several factors (the number of partitions in the tablespace, the buffer pool size, the type of tablespace, and the DSSIZE specified at tablespace creation). These factors are obtained by TMON for DB2 at connection time and stored in an internal representation of the DB2 catalog. (This lookup exists already, to translate internal DBID/OBID information to external names.)

If, however, you have this DB2 catalog lookup disabled (at the direction of ASG Customer Support, with NOCTGTABLES<sup>2</sup> in the TMDBPRMS input parameter), partition numbers will not be collected, and all I/O will be attributable to the base pageset.

When heavy I/O occurs, the partition number lookup may become a performance issue. This may be the case if you start receiving this buffer full message:

```
TMDB31108S DETAIL BUFFER POOL ERROR errcde
```

---

2. Specifies whether to build table/index (TMON for DB2 internal catalog) entries. The default is N. This is a support option for customers with a large number of table entries relative to tablespace entries. This option will also disable SQL Analyzer table statistics collection, which relies on catalog table entries.

Increasing the size or number of DETAIL buffers in the Monitoring Options Detail (Page 1 of 2) screen (accessed using the EDI line command from the DB2 Monitor Controls screen [option 8.3]) will normally alleviate the condition.

If it does not (or the size and number of DETAIL buffers are already at maximum), you can bypass the partition number lookup by adding this parameter setting to TMDBPRMS input:

```
PAGESETPARTS=N
```

The PAGESETPARTS parameter specifies whether to process pageset partitions. The default is N.

## New Exception Monitor Categories

PTF TE02827 provides a fix for a situation in which text substitution was not occurring consistently for lock contention exceptions. This problem was occurring because TMON for DB2 was allowing fields from two different record types from the data dictionary (i.e., IFCIDs 172 and 196) to support lock contention exception reporting (which, in earlier releases, was provided by a single, now obsolete, IFCID 054).

To accommodate the two data dictionary record types, the fix replaces the existing lock contention exception monitor category (which was based on IFCID 054) with two new ones called Timeout and Deadlock. Applying PTF TE02827 and following the hold instructions will result in changes to both the Data Dictionary Record Directory screen (option 9.3) and the Exception Definitions screen (option 3.1).

On the Data Dictionary Record Directory screen, which is described in the *ASG-TMON Products for OS/390 & z/OS Common Components Reference Guide*, the EXCEPTION MON - LOCK TIMEOUT/DEADLCK record title will be replaced by two new record titles:

Record Title	Description
EXCEPTION MON - TIMEOUT	Lets you maintain, activate, and deactivate exception criteria related to lock timeout events for selected DB2 subsystems.
EXCEPTION MON - DEADLOCK	Lets you maintain, activate, and deactivate exception criteria related to deadlock events for selected DB2 subsystems.

---

This example shows the updated screen with the new fields highlighted in red:

```

JOBNAME: jobname          DATA DICTIONARY RECORD DIRECTORY      DATE: MM/DD/YY
PRODUCT: TMDB 042        TIME: HH:MM:SS
COMMAND:
LMRK02711I - CURSOR SELECT SCREEN FIELD FOR INQUIRY
<----- RECORD TITLE -----> <----- RECORD TITLE ----->
SET CURRENT SQLID          END OF IDENTIFY
END OF SIGNON              AUTHORIZATION FAILURE
EXPLICIT GRANT/REVOKE      DDL ON AUDITED OBJECT
1ST WRITE ON AUDITED OBJECT 1ST READ ON AUDITED OBJECT
DML ON AUDITED OBJECT      UTILITY START
UTILITY CHANGE             UTILITY END
DISTRIBUTED AUTHID TRANSLATION SQL TRACE RECORDS
INTERVAL STATISTICS RECORD  THREAD DETAIL RECORD
EXCEPTION LOG RECORD        EXCEPTION MON - SET SQLID
EXCEPTION MON - END OF IDENTIFY EXCEPTION MON - END OF SIGNON
EXCEPTION MON - AUTHORIZATION FAILUR EXCEPTION MON - EXPLICIT GRANT/REVOK
EXCEPTION MON - AUDITED OBJ DDL    EXCEPTION MON - AUDITED OBJ 1ST WRIT
EXCEPTION MON - AUDITED OBJ 1ST READ EXCEPTION MON - AUDITED OBJ DML
PAGE SET/DB2 CATALOG STATS  ACTIVE THREAD RECORD
PAGE SET STATISTICS         DB2/IRLM MESSAGE RECORD
EXCEPTION MON - EDM POOL FULL    EXCEPTION MON - TIMEOUT
HOURLY SUMMARY TABLE         EXCEPTION MON - DEADLOCK
CONNECTION SUMMARY TABLE      PLAN SUMMARY TABLE
HELP INFORMATION = PF1         applid          PF KEY ASSIGNMENTS = PA1

```

Similarly, on the Exception Definitions screen, which is described in Chapter 7 of the *ASG-TMON for DB2 Reference Guide*, the LOCK CONTENTION category has been removed and replaced with these two new categories:

Category	Description
DEADLOCKS	Lets you maintain, activate, and deactivate exception criteria related to deadlock events for selected DB2 subsystems.
TIMEOUTS	Lets you maintain, activate, and deactivate exception criteria related to lock timeout events for selected DB2 subsystems.

This example shows the updated screen with the new fields highlighted in red:

```
JOBNAME:jobname           Exception Definitions           DATE:MM/DD/YY
DB2      :db2              TIME:HH:MM:SS
COMMAND:
TMDB07201I - CURSOR SELECT EXCEPTION TYPE
<----- Exception ----->           <----- Exception ----->
INTERVAL STATISTICS                 IDENTIFY REQUEST
ACTIVE AND COMPLETED THREADS      SIGNON REQUEST
COMPLETED THREADS                 AUTHORIZATION FAILURE
PAGE SET ACTIVITY                 EXPLICIT GRANT/REVOKE
EDM POOL FULL                     FIRST READ ON AUDITED OBJECT
DEADLOCKS                          FIRST WRITE ON AUDITED OBJECT
TIMEOUTS                           DML ON AUDITED OBJECT
SET SQLID                          DDL ON AUDITED OBJECT
DB2/IRLM MESSAGES

HELP INFORMATION = PF1           applid           PF KEY ASSIGNMENTS = PA1
```

## Documentation Update for SQL Analyzer Package and SQL Statement Logging

This update to the *ASG-TMON for DB2 SQL Analyzer Reference Guide* points readers to the *ASG-TMON for DB2 Report Writer Guide* for details on the type of data logged for packages and SQL statements during SQL Capture.

SQL Capture is started from the SQL Analyzer Collection Profile Activation screen, which you access from the Thread Summary (Short Form), option 1.2, or the Thread Summary (Long Form) screen, by entering A next to the CONN ID field.

You specify logging values for the SQL Analyzer Collection Profile Activation screen using the SQL Analyzer Collection Profile screen. You access the SQL Analyzer Collection Profile screen from the SQL Analyzer Controls screen (option 11.1) either by entering ADD in the CMD field and overtyping the profile name, or by entering EDI in the CMD field.

Two fields on the SQL Analyzer Collection Profile screen let you set the type of logging:

- LOG PACKAGE RECORDS?
- LOG SQL STATEMENT RECORDS?

While these fields are described in the *ASG-TMON for DB2 SQL Analyzer Reference Guide*, they fail to note that the type of data being logged is documented in the *ASG-TMON for DB2 Report Writer Guide*. Here is how the field descriptions should read:

Field	Description
LOG PACKAGE RECORDS?	Indicates whether you want TMON for DB2 to log package records for any thread that satisfies your filtering criteria. Valid values are Y for yes and N for no. To see the type of data that will be logged (assuming filter criteria are met), see the section on the SQL Analyzer package log (D7) record in Chapter 3 of the <i>ASG-TMON for DB2 Report Writer Guide</i> .
LOG SQL STATEMENT RECORDS?	Indicates whether you want TMON for DB2 to log SQL statement records for any thread that satisfies your filtering criteria. Valid values are Y for yes and N for no. To see the type of data that will be logged (assuming filter criteria are met), see the section on the SQL Analyzer SQL statement log (D6) record in Chapter 3 of the <i>ASG-TMON for DB2 Report Writer Guide</i> .

## Documentation Update for TSS Message Reference

This section describes an erroneous message reference in the *ASG-TMON for DB2 System Administrator's Guide*. The section in Chapter 5 about changing the XCF group name refers to this TMON Strategic Services message:

```
TMHP52122I - XCFGROUPNAME ( name ) ; MODE ( RESOURCE ) ; SECURE ( YES )
```

All TMHP messages have been renumbered to start with a prefix of LMRK so this message should be documented in this manner:

```
LMRK52122I - XCFGROUPNAME ( name ) ; MODE ( RESOURCE ) ; SECURE ( YES )
```

Here is the full documentation of this message:

```
LMRK52122I XCFGROUPNAME ( group ) , MODE ( mode ) , SECURE ( option )
```

Explanation: This message displays the values of certain parameters relating to the operation of XCF.

Action: Review the values to ensure that they are correct. Refer to the documentation if you need to change them.

## EVENTCOUNTS Startup Parameter and Command

On the Thread Elapsed Times screens (both the Current Thread Activity screen, [page 15](#), and the Collection Analysis screen, [page 36](#)), many event counts that were historically expressed by IFCID (start/end event pairs) now display by event count. If you wish to continue displaying those fields by IFCID pairs, you can use the EVENTCOUNTS startup parameter or command. Affected fields are documented as *EVENTCOUNTS-enabled*.

### EVENTCOUNTS Startup Parameter

This example shows the syntax for the EVENTCOUNTS startup parameter:

```
EVENTCOUNTS=BYEVENT  
EVENTCOUNTS=BYIFCID
```

where:

BYEVENT takes the entry/exit event count and halves it to show the number of times the event occurred.

BYIFCID shows event counts as entry/exit events to a process.

### EVENTCOUNTS Command

You can issue the EVENTCOUNTS command anywhere in TMON for DB2. You can use it to query the current setting or to change it. You must have MONCNTL authorization to change the setting.

This example shows the format for the new command:

```
EVENTCOUNTS=BYEVENT | BYIFCID | ?
```

where:

BYEVENT tells TMON for DB2 to take the entry/exit event count and halve it to show the number of times the event occurred.

BYIFCID tells TMON for DB2 to show event counts as entry/exit events to a process. Fields were displayed in this manner prior to the TMON for DB2 nested activity enhancement.

? displays a message informing you of the current event count setting. All users, including those without MONCTL authorization, can issue EVENTCOUNTS=?.

## Thread Detail Accounting of Nested Activity / WLM Enclave Times

Changes have been made to the Current Thread Activity and Collection Analysis components to support nested activity statistics and to improve data presentation. The new nested activity statistics include event timing (i.e., for stored procedures, user-defined functions [UDFs], triggers, and other relevant events) and adjusted CPU times for threads running nested activity under a workload manager (WLM) enclave.

To distinguish Current Thread Activity screens from the corresponding screens in Collection Analysis, they each include “Current” either in the title or in parentheses following the title. The Collection Analysis screens each include “Collection Analysis” in parentheses following the title.

---

**Note:**

All sample screens in this section display as they will once you apply the TMON for DB2 Nested Activity enhancement.

---

Incorporating the new support into TMON for DB2 has required changing the titles and content of some existing screens (including adding new fields), adding a new screen, and adding new error messages.

These are the changes:

- The Current Thread Detail screen (known as Thread Detail in Collection Analysis) has had field modifications and additions. See [“Changes to the Current Thread Detail Screen” on page 13](#) and [“Changes to the Thread Detail \(Collection Analysis\) Screen” on page 34](#).

The next three screens are accessed from the Thread Detail screens by cursor-selecting the ELAPSED, TOT CPU, and NESTED fields, respectively.

- Two screens have been renamed and had significant field changes and additions:

Old Screen Title	New Screen Title	Page
Thread Event Detail (Current)	Thread Elapsed Times (Current)	15
Thread Event Detail (Collection Analysis)	Thread Elapsed Times (Collection Analysis)	36
Thread CPU Time Detail (Current)	Thread CPU/Elapsed Detail (Current)	25
Thread CPU Detail (Collection Analysis)	Thread CPU/Elapsed Detail (Collection Analysis)	45

See the page references in the table for more information about the changes made to the screens.

- The Thread Nested Activity screen is new. For more information, see [“New Thread Nested Activity \(Current\) Screen” on page 31](#) and [“New Thread Nested Activity \(Collection Analysis\) Screen” on page 51](#).
- The CONN ID field on the Thread Activity (Collection Analysis) screen now supports an additional color, turquoise, to indicate a thread that is involved in either rollup accounting or thread reuse. (As before, white indicates a normal terminating thread and yellow is a thread that participated in parallel activity as a child process.) By studying the different colored threads, you can get the full picture of the activity on the thread.
- The SRB field on the Thread IFI Detail screen has been changed so that its value is now N/A for all rows.
- The PROCEDURE NAME field value on the Thread Summary Short Form screen now displays the currently executing nested activity, whether it is a procedure or another type of nested activity.
- In Current Thread Activity screens, TMON for DB2 adjusts the values under the Elapsed and CPU columns more precisely during a screen refresh. In earlier releases, these values changed only if a higher value was calculated during refresh, resulting in a high-water mark report. Now they will display lower values as well, resulting in values going up *or* down. Note that downward fluctuation will not occur with respect to Overall Total Times as you watch a thread processing.

**Note:**

In Current Thread Activity screens that report trigger times, there is a known problem with DB2's reporting on IFCIDs 147 and 148 that prevents correct enclave times from being included in trigger-related fields under the CPU and Elapsed columns.

When IBM has fixed the DB2 reporting of these IFCIDs, then these fields will include correct enclave times. Correct times are included in the values displayed under corresponding Collection Analysis screens because IFCID 003 does correctly report values.

See IBM APAR PK55611 regarding IBM maintenance needed, as well as the PK55611 text for the status of the APAR.

## Changes to the Current Thread Detail Screen

Some fields and data values on the Current Thread Detail screen have been changed. This example shows the updated screen:

```

JOBNAME: jobname                Current Thread Detail                DATE: MM/DD/YY
DB2      : db2                   TIME: HH:MM:SS
COMMAND:                          CYCLE: MMSS

CONN ID : DB2CALL                PLAN      : TMDB014    CURRENT STATE: INDB2
CORR ID : TBTDB40T              AUTH ID   : LSCSTC    THREAD START : 12:21:08.8979
LOCATION  : RS91D71H              SQLID     : LSCSTC    CONN TYPE    : CALL ATTACH
RQST LOC:                        LUWID     : USASG.RS91D71H.BF2668497169.0001
PKG LOC  :                       ACCT TKN:
PKG NAME :

+----- Timings -----+ +----- Event Counts -----+
ELAPSED:      2.23H DB2 ELA:  5:34.39  WAIT      :      4  PACKAGES:    0
TOT CPU:  0:00.03  DB2 CPU:  0:00.02  IFI       :     291  PARA GRP:    0
I/O WT  :  0:00.00  LOCK WT:  0:00.00  RMT CALL:    0    PARA CPU:    0
SORT    :  0:00.00  TOT WT  :  0:00.00-  SORT      :    0    PARA MBR:    0
NESTED  :  0:00.00                                SQL LOGR:    0    DS OPENS:    0
                                                RID LIST:    0

+----- SQL Counts -----+ +----- Buffer Pool/Locking -----+
TOTAL      :      1  PREPARES :    0  GETPAGE:    0  MX PG LK:    0
SELECT    :    0  OPEN CSR  :    0  SYNC RD:    0  LOCKESCL:    0
FETCH     :    0  INCR BIND:    0  PREFTCH:    0  SUSPENDS:    0
COMMITTS  :    1  SECURITY  :    0  ASYN RD:    0  TIMEOUTS:    0
DML       :    0  DDL       :    0  PGS/IO  :    N/A  DEADLOCK:    0
HELP INFORMATION = PF1                applid                PF KEY ASSIGNMENTS = PA1
    
```

You can access this screen by entering =1 . 2 on the command line and from the Thread Summary (Short Form) screen, cursor-selecting any row from the roster.

This table describes the field changes:

Old Field Name	New Field Name	Explanation
<u>CICS TKN</u>	<u>CICS TKN</u> or ACCT TKN	<p>This field now toggles between CICS TKN and ACCT TKN, depending on the connection type. CICS TKN displays if the connection type is CICS ATTACH. For all other connection types, the field label will be ACCT TKN, which is short for <i>accounting token</i>. If an accounting token exists, it will be displayed as received in the DB2 data section that provides it.</p> <p>CICS TKN remains cursor-selectable to take you to the TMON for CICS TS Activity Monitor. ACCT TKN, however, is not cursor-selectable.</p>
DB2	DB2 ELA	The data value has changed from displaying only DB2 nonnested elapsed time to including elapsed time for all nested activity.
DB2 CPU	DB2 CPU	The data value has been changed from displaying only DB2 nonnested CPU time to including CPU time for all nested activity.
<u>TOTAL (Elapsed Times)</u>	ELAPSED	Even though this field has been renamed, the data value has not changed: it is still the total elapsed time. The new label, however, better represents the underlying data that displays when you cursor-select the field. The resulting screen used to be called Thread Event Detail (see <a href="#">page 15</a> ), but has been renamed Thread Elapsed Times.
Elapsed Times	Timings	The Elapsed Times section heading has been renamed Timings to better describe the broader scope of data the section now provides.
—	NESTED	This new field displays the total elapsed time for all DB2 nested activity for the thread. Cursor-selecting this field will display the new Thread Nested Activity (Current) screen, described on <a href="#">page 31</a> .
SORT (Event Counts)	SORT (Event Counts)	This field is now cursor-selectable. It displays the Thread Sort Detail (Current) screen, which is the same screen you access by cursor-selecting the SORT field under Timings.

Old Field Name	New Field Name	Explanation
<u>TOT CPU</u>	TOT CPU	The data value for this field has changed from displaying only application and DB2 nonnested CPU time to including CPU time for all nested activity. Also, cursor-selecting this field displays one of the renamed screens: the Thread CPU/Elapsed Times screen, which was formerly called the Thread CPU Detail (Current) screen (see <a href="#">page 25</a> ).
—	TOT WT	This new field displays the total wait time. This value includes I/O wait, lock wait, and all other Class 3 wait times that occur within DB2. For a breakdown of all wait times, cursor-select the <u>ELAPSED</u> field.

## Changes to the Thread Event Detail (Current) Screen

The Thread Event Detail (Current) screen has been renamed the Thread Elapsed Times (Current) screen. Due to the significant number of changes that have been made to the screen, the complete documentation for it is included in this section—see “[Thread Elapsed Times \(Current\)](#)” on [page 17](#).

Besides the name change, these changes also have been made:

- It is now a GRF screen.
- Three fields have been added:
  - DB2 Elapsed (This field has a different meaning and location than the field previously known as DB2 ELAPSED. That field has been renamed; see the table below.)
  - WAIT: USER DEF FUNC
  - WAIT: STORED PROC
- The names of these fields have been changed:

Old Field Name	New Field Name
COMMIT TOTAL	COMMIT 1PHS TOTAL
DB2 ELAPSED	DB2 NON-NESTED
INDOUBT	COMMIT 2PHS INDOUBT
PHASE1	COMMIT 2PHS PHASE 1
PHASE2	COMMIT 2PHS PHASE 2

Old Field Name	New Field Name
PAGE LATCH	WAIT: PAGE LATCH
LOCK/LATCH	WAIT: LOCK/LATCH
Global Messages	WAIT: GLOBAL MSGS
Global Locks	WAIT: GLOBAL LOCK
SPROC EXECUTIONS	STORED PROCEDURE
WAIT: I/O	WAIT: SYNC I/O

- The TIMEOUTS, REJECTS, and ABENDS fields have been moved to the new Thread Nested Activity (Current) screen, described on [page 31](#).
- Sorting has changed. Before, records were sorted on group (by type, for example, Commits) and on Elapsed time. Now, because the event names are more descriptive, the records are sorted only on Elapsed time.
- Values of greater than 100 percent are explained for the STORED PROCEDURE, TRIGGER, and USER DEF FUNCTION fields.
- By default, all fields now display their event counts by event; however, there are certain fields you can view by IFCID if desired. These are documented as *EVENTCOUNTS-enabled*. For more information, see “[EVENTCOUNTS Startup Parameter and Command](#)” on [page 10](#).
- A second screen (accessed by scrolling to the right) shows Average Elapsed times. This is derived by dividing the total Elapsed Time by the number of events (not by the number of entry/exit IFCIDs received).

## Thread Elapsed Times (Current)

```

JOBNAME: jobname          Thread Elapsed Times          DATE: MM/DD/YY
DB2      : db2            Current                    TIME: HH:MM:SS
COMMAND:                                CYCLE: MMSS
                                           _RIGHT 1
CONN ID: DB2CALL         PLAN   : TMDB014      CURRENT STATE: INDB2
CORR ID: TBTDB40T       AUTH ID: LSCSTC      THREAD START  : 12:21:08.8979

DB2 ELAPSED:      7:55.7595          TOTAL ELAPSED:  3:00:55.1763
DISPLAYED: 1 TO 12 OF 28          SORT COLUMN#: 03 (1-5)  ORDER: D (A/D)
Event              Count      Elapsed      % Total Elapsed      % DB2 Elapsed
*                  *          *
DB2 NON-NESTED    :    198      7:55.7595      4.38                100.00
WAIT: OTHER EUS   :     1         0.0003      0.00                 0.00
COMMIT 1PHS TOTAL :     1         0.0000      0.00                 0.00
WAIT: DEALLOC EUS :     1         0.0000      0.00                 0.00
SORT ELAPSED      :     0         0.0000      0.00                 0.00
COMMIT 2PHS PHASE 1:  0         0.0000      0.00                 0.00
COMMIT 2PHS PHASE 2:  0         0.0000      0.00                 0.00
ABORT ELAPSED     :     0         0.0000      0.00                 0.00
WAIT: OTHR READ I/O:  0         0.0000      0.00                 0.00
WAIT: OTHR WRIT I/O:  0         0.0000      0.00                 0.00
WAIT: LOCK/LATCH  :     0         0.0000      0.00                 0.00
WAIT: PAGE LATCH  :     0         0.0000      0.00                 0.00
HELP INFORMATION = PF1          applid          PF KEY ASSIGNMENTS = PA1

```

This is a GRF screen. See the section on GRF screens in Chapter 2 of the ASG-TMON for DB2 Reference Guide for a description of its characteristics.

The Thread Elapsed Times (Current) screen presents thread activity (e.g., commits, sorts, drain locks, claim locks, and remote calls) and corresponding thread and DB2 times for a specific thread, sorted by elapsed time. You can use this screen to pinpoint thread response problems to DB2.

The Thread Elapsed Times (Current) screen contains filter areas below some of the field headings in which you can enter criteria to limit the roster to specific images. For information on filtering data, see the section on GRF screens in Chapter 2 of the ASG-TMON for DB2 Reference Guide.

**Note:**

You can use GRF's PROFILE command to display the GRF Profile Display/Alter screen, which allows you to save the sort order, sort column, and filtering criteria for this screen and apply them to subsequent screens. For more information, see the description of PROFILE in the section on GRF commands in Chapter 2 of the ASG-TMON for DB2 Reference Guide and Chapter 2 in ASG-TMON Products for OS/390 & z/OS Common Components Reference Guide.

## Accessing this Screen

### *To access this screen*

- ▶ From the Current Thread Detail screen ([page 13](#)), cursor-select the ELAPSED field.

### Primary Commands

You can enter certain TMON for DB2 commands on every Current Thread data screen. These commands are documented under Current Thread Activity Menu in Chapter 4 of the *ASG-TMON for DB2 Reference Guide*.

You also can enter these commands on the command line:

Command	Description
LEFT	Scrolls the screen left to return to the previous display. You also can cursor-select the Left field to perform this function. This command is valid only if the Left field is displayed on the screen.
PROFILE	This is a GRF command, see the section on GRF commands in Chapter 2 of the <i>ASG-TMON for DB2 Reference Guide</i> .
RIGHT	Scrolls the screen right to display additional data. You also can cursor-select the Right field to perform this function. This command is valid only if the Right field is displayed on the screen.

Use the DOWN and UP commands to scroll through this screen. For syntax and descriptions of all commands, see the section on commands in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

### Fields

A field value that is highlighted and displayed in turquoise indicates an abbreviated variable-length character string, or *V-string*. You can cursor-select it to display a pop-up window showing the entire V-string data value. See “[DB2 Version 8.1 Variable-character String Long Identifiers](#)” on [page 64](#) for further information.

Because several fields in this component serve the same function on multiple screens, they are defined under the Current Thread Activity Menu in Chapter 4 of the *ASG-TMON for DB2 Reference Guide*.

The (filter area), DISPLAYED: *nnnnn* TO *nnnnn* OF *nnnnn*, ORDER, and SORT COLUMN# fields are GRF fields and are described in the section on GRF screens in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

This table lists fields unique to the Thread Elapsed Times (Current) screen:

Field	Description
% DB2 Elapsed	Displays a percentage that shows the ratio of the elapsed time of the event (the ELAPSED field) to total DB2 elapsed time (the ELAPSED field for the DB2 Elapsed event). See the individual field descriptions for an explanation of each event.
% Total Elapsed	Displays a percentage that shows the ratio of the elapsed time of the event (the ELAPSED field) to total thread elapsed time (the TOTAL ELAPSED field). See the individual field descriptions for an explanation of each event.
ABORT ELAPSED	Displays statistics for the number of times the allied thread processed an abort and the amount of time the allied thread has spent in abort processing.
Avg. Elapsed	Displays the average elapsed time for the associated event. This value is derived by dividing the total elapsed time by the number of events. It is <i>not</i> related to the number of entry/exit IFCIDs received.
COMMIT 1PHS TOTAL	Displays statistics for the total elapsed time spent for all aspects of single-phase commit processing.
COMMIT 2PHS INDOUBT	Displays statistics for the total time spent in INDOUBT during 2-phase commit processing. This time is spent waiting on the application in between phase 1- and phase 2-commit processing during a 2-phase commit. It could exceed the total DB2 elapsed time as it occurs in the application.
COMMIT 2PHS PHASE 1	Displays statistics for the total time spent in phase 1 of 2-phase commit processing.
COMMIT 2PHS PHASE 2	Displays statistics for the total time the allied thread spent in phase 2 of 2-phase commit processing.
Count	Displays the total number of occurrences of each event listed in the Event field. By default, all fields display their event counts by event, but there are certain fields you can view by IFCID if desired. For more information, see the EVENTCOUNTS command under <a href="#">“EVENTCOUNTS Startup Parameter and Command” on page 10</a> .  See the individual field descriptions for an explanation of each event and whether they can be displayed by IFCID using the EVENTCOUNTS command. These are referred to as being <i>EVENTCOUNTS-enabled</i> .
DB2 ELAPSED	Displays statistics for the number of DB2 entry and exit events processed to calculate the elapsed time in DB2 and the amount of time spent processing in DB2.

Field	Description
	<p>The count does not include SQL entry/exit events processed by a stored procedure. The time is calculated by subtracting the store-clock (STCK) time on exit from DB2 from the STCK time on entry to DB2. It does not include certain abend conditions or thread allocation processes.</p>
DB2 NON-NESTED	<p>Displays statistics for the time spent IN-DB2, but not in a nested activity.</p> <p>The count does not include SQL entry/exit events processed by a stored procedure. The time is calculated by subtracting the store-clock (STCK) time on exit from DB2 from the STCK time on entry to DB2. It does not include certain abend conditions or thread allocation processes.</p> <p>This field is EVENTCOUNTS-enabled.</p>
Elapsed	<p>Displays the amount of elapsed time (in <i>HH:MM:SS.tttt</i> format) associated with the event shown in the Event field. See the individual field descriptions for an explanation of each event.</p> <p><b>Note:</b></p> <hr/> <p>Do not expect the sum of the elapsed times for the events to equal the total DB2 elapsed time. It may be either greater (due to overlapping intervals for some events) or less (due to DB2 buffer pool processing that incurs no delays) than the total DB2 elapsed time. Also, be careful when correlating the I/O elapsed times on this screen to the I/O elapsed times on the Thread Resource Detail (Current) screen. Only the I/O WAIT field on this screen can be correlated to the SYNCREAD and SYNCWRIT fields on that screen.</p> <hr/> <p>Elapsed times reported for asynchronous I/O cannot be correlated. This screen provides only the time incurred by the thread while waiting for a page being retrieved through asynchronous I/O processing. The Thread Resource Detail (Current) screen contains the total asynchronous I/O elapsed time. It is possible for asynchronous I/Os to overlap, causing the total elapsed I/O times shown on the Thread Resource Summary (Current) screen to exceed the total DB2 elapsed time as well as the total thread elapsed time. For example, suppose a DB2 thread starts at 9:00:00 and ends at 9:00:10. Its total elapsed time is 10 seconds. Suppose that during the 10-second interval, two simultaneous asynchronous I/O requests start at 9:00:02 and end at 9:00:08. Each asynchronous I/O has an elapsed time of 6 seconds. When added together, the total elapsed I/O time is 12 seconds, which exceeds the thread's total elapsed time of 10 seconds.</p>

---

Field	Description
	For more information, see the section on DB2 elapsed time in Appendix B of the <i>ASG-TMON for DB2 Reference Guide</i> .
Event	Lists the event to which the information on the remainder of the row applies. The events incurring the longest elapsed times are displayed at the top of the list. Scroll forward to display the remaining events.
SORT ELAPSED	Displays statistics for the number of times DB2 has requested a sort and the amount of time the thread has spent sorting since it began processing.
STORED PROCEDURE	Displays statistics for the number of SQL entry/exit events performed by stored procedures and the elapsed time spent executing a stored procedure.
	<p><b>Note:</b></p> <p>The elapsed time for these metrics may show a value greater than 100 percent of the DB2 elapsed time. This is possible due to the fact that the elapsed value for these fields includes the total activity time (both time spent IN-DB2 and IN-APP, where the APP is the nested activity), whereas the DB2 elapsed time includes only the time actually spent processing in DB2. Refer to the Thread CPU/Elapsed Detail (Current) screen, described on <a href="#">page 25</a>, for a breakdown of where the elapsed time was spent for a given nested activity.</p>
TOTAL ELAPSED	Displays the amount of time that has elapsed since the thread began processing.
TRIGGER	Displays the amount of time that elapsed while the thread was executing DB2 triggers.
	This field is valid only if you are running DB2 6.1 or later.
	<p><b>Note:</b></p> <p>The elapsed time for these metrics may show a value greater than 100 percent of the DB2 elapsed time. This is possible due to the fact that the elapsed value for these fields includes the total activity time (both time spent IN-DB2 and IN-APP, where the APP is the nested activity), whereas the DB2 elapsed time includes only the time actually spent processing in DB2. Refer to the Thread CPU/Elapsed Detail (Current) screen, described on <a href="#">page 25</a>, for a breakdown of where the elapsed time was spent for a given nested activity.</p>

Field	Description
USER DEF FUNCTION	<p>Displays statistics for the number of entry/exit pairs from UDFs. This field also displays statistics for the elapsed time the thread waited for the function to complete.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p> <p><b>Note:</b></p> <hr/> <p>The elapsed time for these metrics may show a value greater than 100 percent of the DB2 elapsed time. This is possible due to the fact that the elapsed value for these fields includes the total activity time (both time spent IN-DB2 and IN-APP, where the APP is the nested activity), whereas the DB2 elapsed time includes only the time actually spent processing in DB2. Refer to the Thread CPU/Elapsed Detail (Current) screen, described on <a href="#">page 25</a>, for a breakdown of where the elapsed time was spent for a given nested activity.</p> <hr/>
WAIT: ARCHIVE LOG	<p>Displays statistics for the number of ARCHIVE LOG commands issued and the accumulated time the thread was suspended because of archive log requests. This is not the total time required to complete the command.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: ARCHIVE READ	<p>Displays statistics for the number of ARCHIVE LOG READ commands issued and the accumulated time the thread was suspended because of archive log reads.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: CLAIM	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate the elapsed time the thread spent waiting for claims to be released. This field also displays statistics for the accumulated time the thread waited for claims to be released.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: DB2 SRV EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for DBM1 services, including dataset define, dataset delete, dataset extend, dataset reset, and VSAM catalog access. This field also displays statistics for the elapsed time the thread waited for the process to complete.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p> <hr/>

Field	Description
WAIT: DEALLOC EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for DB2 commit, abort, or deallocation processing. This field also displays statistics for the elapsed time the thread waited for the process to complete.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: DRAIN	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate drain lock elapsed wait times. This field also displays statistics for the accumulated time the thread waited for a drain lock.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: GLOBAL LOCK	<p>Displays statistics for the number of wait trace events processed for waits for global lock contention. This field also displays statistics for the elapsed time the thread spent waiting for a suspension of an IRLM lock request due to contention with a global lock, which required intersystem communication to resolve.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: GLOBAL MSGS	<p>Displays statistics for the number of wait trace events processed for waits sending messages to other members in a data-sharing group (DSG). This field also displays statistics for the elapsed time the thread has spent waiting due to suspension for sending messages.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: LOCK/LATCH	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate lock and latch elapsed wait times. This field also displays statistics for the amount of time the thread has spent in lock wait since it began processing in DB2. The lock wait is caused by <i>local</i> lock contention, which does not require intersystem communication to resolve.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: LOG WRITE I/O	<p>Displays statistics for the number of entry/exit pairs from wait events processed for log write I/O. This field also displays statistics for the elapsed time the thread waited for the I/O to complete.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>

Field	Description
WAIT: OPEN/CLS EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for dataset open/close or HSM recall processing. This field also displays statistics for the elapsed time the thread waited for the process to complete.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: OTHER EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for other execution unit switch processing. This field also displays statistics for the elapsed time the thread waited for the process to complete.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: OTHR READ I/O	<p>Displays statistics for the number of wait trace events processed for waits for read I/O under another thread (e.g., a system agent) since this thread began processing in DB2. This field also displays statistics for the accumulated wait time for read I/O performed under a thread other than this one. This value is calculated for threads by subtracting the STCK time on entry to event wait from STCK on resume from the event.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: OTHR WRIT I/O	<p>Shows statistics for the number of wait trace events processed for waits for write I/O under another thread (e.g., a system agent) since this thread began processing in DB2. This field also displays statistics for the accumulated wait time for write I/O performed under a thread other than this one. This value is calculated for threads by subtracting the STCK time on entry to event wait from STCK on resume from the event.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: PAGE LATCH	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate page latch elapsed wait times and the accumulated time the thread waited for a page latch.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: STORED PROC	<p>Displays statistics for the number of times a SQL CALL statement had to wait for an available TCB and the elapsed time the thread spent waiting for an available TCB to schedule a stored procedure.</p>

---

Field	Description
WAIT: SYNC I/O	<p>Displays statistics for the number of entry/exit pairs from I/O events processed to calculate I/O elapsed wait times. This field also displays statistics for the amount of time the allied thread has spent waiting for synchronous read and write I/O since it began processing in DB2.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: SYSLGRNG EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for SYSLGRNG recording processing. This field also displays statistics for the elapsed time the thread waited for the process to complete. See the % DB2 Elapsed, % Total Elapsed, Count, and Elapsed field descriptions for more information.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: USER DEF FUNC	<p>Displays the wait time a UDF could have incurred while waiting to be run. If the UDF WAIT TIME is zero, this field displays a count of 0 (zero); otherwise, it will be 1.</p>

## Changes to the Thread CPU Time Detail (Current) Screen

The Thread CPU Time Detail (Current) screen has been renamed the Thread CPU/Elapsed Detail (Current) screen. Because of the significant number of changes that have been made to this screen, the complete documentation for it is included in this section—see [“Thread CPU/Elapsed Detail \(Current\)” on page 26](#).

Besides the name change, these changes also have been made:

- It is now a GRF screen.
- The TCB time and SRB time columns have been replaced with two graph fields, % CPU and % ELAPSED.
- The APPLICATION and DB2 fields that used to be under the % of Total heading are now covered under % CPU and % ELAPSED in conjunction with the TOTAL category.
- The number of fields has increased from 13 to 24 to provide additional metrics. (This count does not include these column headings: Elapsed, CPU, % ELAPSED, and % CPU.)
- Many of the fields have been renamed.
- The hierarchical information in the Times column is more clearly delineated with summation (-) and detail (..) characters preceding each row listed below the OVERALL TOTAL field.

## Thread CPU/Elapsed Detail (Current)

JOBNAME: <i>jobname</i>		Thread CPU/Elapsed Detail		DATE: <i>MM/DD/YY</i>	
DB2 : <i>db2</i>		Current		TIME: <i>HH:MM:SS</i>	
COMMAND:				CYCLE: <i>MMSS</i>	
CONN ID: DB2CALL	PLAN : TMDB014	CURRENT STATE: INDB2			
CORR ID: TBTDB40T	AUTH ID: LSCSTC	THREAD START : 12:21:08.8979			
DISPLAYED: 1 TO 15 OF 17					
Times	CPU	Elapsed	% CPU	% ELAPSED	
OVERALL TOTAL:	0.0347	3:01:28.1501	100.00	100.00	
.IN-APP :	0.0000	2:53:32.3908	0.00	95.63	
.IN-DB2 :	0.0347	7:55.7593	100.00	4.36	
-NON NESTED :	0.0347	3:01:28.1501	100.00	100.00	
..IN-APP :	0.0000	2:53:32.3908	0.00	95.63	
..IN-DB2 :	0.0347	7:55.7593	100.00	4.36	
-NESTED :	0.0000	0.0000	0.00	0.00	
..IN-APP :	0.0000	0.0000	0.00	0.00	
..IN-DB2 :	0.0000	0.0000	0.00	0.00	
-STORED PROC:	0.0000	0.0000	0.00	0.00	
..IN-APP :	0.0000	0.0000	0.00	0.00	
..IN-DB2 :	0.0000	0.0000	0.00	0.00	
-UDF :	0.0000	0.0000	0.00	0.00	
..IN-APP :	0.0000	0.0000	0.00	0.00	
..IN-DB2 :	0.0000	0.0000	0.00	0.00	
HELP INFORMATION = PF1		<i>applid</i>	PF KEY ASSIGNMENTS = PA1		

This is a GRF screen. See the section on GRF screens in Chapter 2 of the ASG-TMON for DB2 Reference Guide for a description of its characteristics.

The Thread CPU/Elapsed Detail (Current) screen presents detailed CPU information for a specific thread, based on level of detail and summation. You can use this information to pinpoint thread CPU resource consumption and thread response problems.

Information in the Times column is arranged in hierarchical fashion underneath the OVERALL TOTAL field. Fields preceded by dashes indicate summary values, and fields preceded by periods indicate a subset of the last field preceded by a dash. For a specific description of each row and column, see “Fields” on page 27.

The fields displayed on this screen are from the thread accounting and distributed accounting data sections of the IFCID 148 record. Refer to Chapter 3 in ASG-TMON for DB2 System Administrator’s Guide for detailed information on IFCID records.

**Note:**

You can use GRF’s PROFILE command to display the GRF Profile Display/Alter screen, which allows you to save the sort order, sort column, and filtering criteria for this screen and apply them to subsequent screens. For more information, see the description of PROFILE in the section on GRF commands in Chapter 2 of the ASG-TMON for DB2 Reference Guide and Chapter 2 in ASG-TMON Products for OS/390 & z/OS Common Components Reference Guide.

## Accessing this Screen

### To access this screen

- ▶ From the Current Thread Detail screen ([page 13](#)), cursor-select the TOT CPU field.

### Primary Commands

You can enter certain TMON for DB2 commands on every Current Thread data screen. These commands are documented under the section describing the Current Thread Activity Menu in Chapter 4 the *ASG-TMON for DB2 Reference Guide*. For syntax and descriptions of all commands, see the section describing commands in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

### Fields

A field value that is highlighted and displayed in turquoise indicates an abbreviated variable-length character string, or *V-string*. You can cursor-select it to display a pop-up window showing the entire V-string data value. See “[DB2 Version 8.1 Variable-character String Long Identifiers](#)” on [page 64](#) for further information.

The DISPLAYED: *nnnnn* TO *nnnnn* OF *nnnnn* field is a GRF field and is described under GRF screens in Chapter 2 the *ASG-TMON for DB2 Reference Guide*.

Because several fields in this component serve the same function on multiple screens, they are defined under Current Thread Activity Menu in Chapter 4 of the *ASG-TMON for DB2 Reference Guide*. This table lists fields unique to the Thread CPU/Elapsed Detail (Current) screen:

Field	Description
% CPU <sup>†</sup>	<p>Displays a graphical representation of the value shown in the CPU field for this category.</p> <p><b>Note:</b> _____            Non-total values in this column may be greater than 100 percent. This is possible due to the fact that threads are sometimes reported across multiple accounting records when they are involved in thread reuse, rollup accounting, and/or parallel activity.</p>
% ELAPSED <sup>†</sup>	<p>Displays a graphical representation of the value shown in the Elapsed field for this category.</p> <p><b>Note:</b> _____            Non-total values in this column may be greater than 100 percent. This is possible due to the fact that threads are sometimes reported across multiple accounting records when they are involved in thread reuse, rollup accounting, and/or parallel activity.</p>

Field	Description
CPU	Displays the total CPU time the thread has consumed since it began processing in the activity, in <i>HH:MM:SS.tttt</i> format. The field descriptions for some of the rows under this heading contain additional information about this column.
Elapsed	Displays the amount of time that has elapsed since the thread began processing in the activity, in <i>HH:MM:SS.tttt</i> format. The field descriptions for some of the rows under this heading contain additional information about this column.
ENCLAVE <sup>†</sup>	Displays the amount of time consumed while the thread was executing triggers under an enclave.
IN-APP (NESTED) <sup>†</sup>	Displays the amount of time the thread has spent executing in the application code within a nested activity (i.e., stored procedures and UDFs). Times are broken down in a hierarchy below this row, showing the details on where this total nested IN-APP time was spent.
IN-APP (NON NESTED) <sup>†</sup>	Displays the amount of time the thread has spent executing within a nonnested activity.
IN-APP (OVERALL TOTAL)	Displays the amount of time the thread has used since it began processing both nested and nonnested activities.
	<p><b>Note:</b></p> <hr/> <p>Overall total IN-APP time includes both nonnested and nested IN-APP time. From a user's application perspective, the IN-APP time might call a stored procedure and the time spent in the stored procedure could be viewed as all IN-DB2 time. However, the nested activity IN-APP time is classified as part of the value in OVERALL TOTAL IN-APP since it is indeed time spent outside of DB2.</p> <hr/>
IN-APP (STORED PROC)	<p>Displays the amount of time the thread spent executing within the stored procedure activity.</p> <p>The value in the Elapsed field is computed using the thread package segments that were logged by a stored procedure. If thread package segments are not logged, N/A is displayed.</p>
IN-APP (UDF)	<p>Displays the total amount of time the thread consumed while executing within UDF activities.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p>

Field	Description
IN-APP (zIIP) <sup>†</sup>	Displays the amount of time consumed while the thread was executing in an activity on an IBM System z9 Integrated Information Processor (zIIP).
IN-DB2 (NESTED) <sup>†</sup>	Displays the amount of time the thread has spent executing in DB2 within a nested activity (i.e., stored procedures, UDFs, and triggers). Times are broken down in a hierarchy below this row, showing the details about where this total nested IN-DB2 time was spent.
IN-DB2 (NON NESTED) <sup>†</sup>	Shows the amount of time the thread has spent executing nonnested activity in DB2.  The value in the Elapsed field includes DB2 processing time plus activity processing time.
IN-DB2 (OVERALL TOTAL)	Shows the amount of time the thread has used since it began processing in the local DB2.
IN-DB2 (STORED PROC)	Displays the amount of time the thread spent executing DB2 SQL from the stored procedure.  The value in the Elapsed field is computed using the thread package segments that were logged by a stored procedure. If thread package segments are not logged, N/A is displayed.
IN-DB2 (UDF)	Displays the total amount of time the thread consumed while executing DB2 SQL from UDF activities.  The value in the Elapsed field includes DB2 processing time plus activity processing time.  This field is valid only if you are running DB2 6.1 or later.
IN-DB2 (zIIP) <sup>†</sup>	Displays the total amount of time the thread consumed while executing DB2 SQL on an IBM zIIP processor.
NESTED <sup>†</sup>	Shows the amount of time the thread spent executing in nested activities.
NON-ENCLV <sup>†</sup>	Displays the amount of time the thread spent executing triggers, excluding time spent on a zIIP processor and time spent under an enclave. In other words, this is the time not accounted for in displayed enclave or zIIP times.
NON NESTED <sup>†</sup>	Shows the amount of time the thread spent executing in nonnested activities.

Field	Description
OVERALL TOTAL	<p>Shows the total amount of time the thread has used since it began processing in the activity.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p>
REMOTE DB2	<p>Displays the amount of time the thread has used since it began processing at the remote DB2(s).</p> <p>The values in the CPU and Elapsed fields are available only for threads using system-directed access.</p> <p>When a thread is executing remotely using DRDA protocols, no remote CPU times are available. If the thread executed remotely using only DRDA protocols, the remote DB2 CPU time will contain a value of zero.</p> <p>For more information on DRDA and private protocols, see IBM's DB2 Administration Guide for distributed environment information on remote access types.</p>
STORED PROC	<p>Shows the amount of time the thread spent executing a stored procedure.</p>
TRIGGER	<p>Displays the amount of time the thread consumed while executing DB2 triggers.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p>
UDF	<p>Displays the total amount of time the thread consumed while executing UDFs.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p> <p>This field is valid only if you are running DB2 6.1 or later.</p>
zIIP <sup>†</sup>	<p>Displays the total amount of time the thread consumed while executing on an IBM zIIP processor in all environments.</p>
zIIP ELIGIBLE <sup>†</sup>	<p>Displays the total amount of time the thread spent on work that was zIIP-eligible, but was not executed on a zIIP processor.</p>
zIIP (TRIGGER) <sup>†</sup>	<p>Displays the amount of time the thread consumed while executing triggers on an IBM zIIP processor.</p>

<sup>†</sup> This field is new.

## New Thread Nested Activity (Current) Screen

The Thread Nested Activity (Current) screen is new. This section contains a sample of the screen and its complete documentation.

JOBNAME: <i>jobname</i>	Thread Nested Activity	DATE: <i>MM/DD/YY</i>		
DB2 : <i>db2</i>	Current	TIME: <i>HH:MM:SS</i>		
COMMAND:		CYCLE: <i>MMSS</i>		
CONN ID: DB2CALL	PLAN :	START DATE: 07/24/06		
CORR ID:	AUTH ID:	START TIME: 16:21:08.8979		
DB2 ELAPSED: 0.0124		TOTAL ELAPSED: 29:39:25.0798		
Activity	CPU	Elapsed	Invocations	
STORED PROCEDURES :	0.0000	0.0000	0	
USER DEFINED FUNCS :	0.0000	0.0000	0	
TRIGGERS :	0.0000	0.0000	0	
TOT NESTED ACTIVITY:	0.0000	0.0000	0	
Exceptions	Stored Proc	UDF	Trigger Info	Count
ABENDS :	0	0	STATEMENT TRGRS :	0
REJECTS :	0	0	ROW TRIGGERS :	0
TIMEOUTS :	0	0	SQL ERRORS :	0
Max Nesting Levels :	0			
HELP INFORMATION = PF1	<i>applid</i>	PF KEY ASSIGNMENTS = PA1		

The Thread Nested Activity (Current) screen presents nested activity for a single thread, as far as it has completed. The bottom of the screen contains these sections:

- The Activity section shows the CPU and elapsed times spent by stored procedures, UDFs, and triggers, including times spend under an enclave. It also shows the number of times each activity was invoked.
- Information in the Exceptions section is presented only for stored procedures and UDFs.
- The Trigger Info section gives additional information pertaining to triggers.

### Accessing this Screen

#### *To access this screen*

- From the Current Thread Detail screen ([page 13](#)), cursor-select the NESTED field.

### Primary Commands

You can enter certain TMON for DB2 commands on every Current Thread data screen. These commands are documented under the section describing the Current Thread Activity Menu in Chapter 4 of the *ASG-TMON for DB2 Reference Guide*. For syntax and descriptions of all commands, see the section describing commands in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

## Fields

A field value that is highlighted and displayed in turquoise indicates an abbreviated variable-length character string, or *V-string*. You can cursor-select it to display a pop-up window showing the entire V-string data value. See [“DB2 Version 8.1 Variable-character String Long Identifiers” on page 64](#) for further information.

Because several fields in this component serve the same function on multiple screens, they are defined in the section on the Current Thread Activity Menu in Chapter 4 of the *ASG-TMON for DB2 Reference Guide*. This table lists fields unique to the Thread Nested Activity (Current) screen:

Field	Description
ABENDS	Displays the number of times a stored procedure or UDF terminated abnormally.
CPU	Displays the total CPU time the thread consumed while processing in the activity, in <i>HH:MM:SS.tttt</i> format.
DB2 ELAPSED	Shows the total amount of time the thread consumed while processing in DB2; includes nested and nonnested time.
Elapsed	Displays the amount of time that the thread spent processing in the activity, in <i>HH:MM:SS.tttt</i> format. The field descriptions for some of the rows under this heading contain additional information about this column.
Invocations	Displays the number of times the activity was called. N/A means your installation does not use that DB2 nested activity (or function).
Max Nesting Levels	Displays the depth of nesting that occurred with all stored procedures, UDFs, and DB2 triggers. The maximum nesting depth is 16.
REJECTS	Displays the number of SQL CALL statements rejected because the stored procedure or UDF was in STOP ACTION(REJECT) state.
ROW TRIGGERS	Displays the number of DB2 row triggers that were executed by the thread.
SQL ERRORS	Displays the number of SQL errors the thread encountered while executing DB2 triggers.
STATEMENT TRGRS	Displays the number of DB2 statement triggers that were executed by the thread.
STORED PROCEDURES	Shows the amount of time the thread spent in nested activity while executing stored procedures.

---

Field	Description
TIMEOUTS	Displays the number of SQL CALL statements for stored procedures and UDFs that timed out while waiting to be scheduled.
TOT NESTED ACTIVITY	Displays the total CPU time, elapsed time, and invocation count for the stored procedures, UDFs, and DB2 trigger activities.
TOTAL ELAPSED	Shows the total amount of time the thread consumed while processing in the activity. This includes DB2 processing time plus activity processing time.
TRIGGERS	Displays the total amount of time the thread spent in nested activity while executing DB2 triggers; this includes both enclave and non-enclave times. This field is valid only if you are running DB2 6.1 or later.  <b>Note:</b> <hr/> At the time of production, a known problem with IBM's reporting on IFCIDs 147 and 148 prevents correct enclave times from being included in the values in the CPU and Elapsed fields. When IBM has fixed the DB2 reporting of these IFCIDs, then these fields will include correct enclave times. Correct times are included in the values displayed under Collection Analysis because IFCID 003 does correctly report values. To see those values, see the TRIGGERS field on the Thread Nested Activity (Collection Analysis) screen, described on <a href="#">page 51</a> . <hr/>
USER DEFINED FUNCS	Displays the total amounts of time the thread spent in nested activity while executing UDFs. This field is valid only if you are running DB2 6.1 or later.

---

## Changes to the Thread Detail (Collection Analysis) Screen

Some fields and data values on the Thread Detail (Collection Analysis) screen have been changed. This example shows the updated screen:

```

JOBNAME:jobname                      Thread Detail                      DATE:MM/DD/YY
DB2      :db2                        Collection Analysis                TIME:HH:MM:SS
COMMAND:

CONN ID : DB2CALL                    PLAN      : TMDB324                DATE       : 06/29/06
CORR ID : TBTDB40T                   AUTH ID   : LSCSTC                 START TIME : 13:33:12.7956
LOCATION:  RS91D71H                    OPID      : LSCSTC                 END TIME   : 13:33:16.1745
RQST LOC:                             SQLID     : LSCSTC                 ACE TOKEN  : 1ED01E08
REASON   : NORMAL                     LUWID     : USASG.RS91D71H.BF0709C4718C.0001
CONN TYP: CALL ATTACH                 ACCT TKN:

+----- Timings -----+          +----- Event Counts -----+
ELAPSED: 0:03.37 DB2 ELA: 0:03.34   WAIT      :          8   PACKAGES:    1
TOT CPU:  0:00.03 DB2 CPU: 0:00.02   IFI       :          0   PARA GRP:    0
I/O WT  : 0:00.00 LOCK WT: 0:00.00-   RMT CALL:          0   PARA CPU:    0
SORT    : 0:02.09 TOT WT : 0:00.00-   SORT      :          7   PARA MBR:    0
NESTED  : 1:25.98                    SQL LOGR:          0   DS OPENS:    0
                                           RID LIST:         158

+----- SQL Counts -----+       +----- Buffer Pool/Locking -----+
TOTAL   :          395  PREPARES :          2   GETPAGE:         1897  MX PG LK:    4
SELECT  :          8   OPEN CSR  :          4   SYNC RD:          0   LOCKESCL:    0
FETCH   :          375  INCR BIND:          0   PREFTCH:          55  SUSPENDS:    0
COMMIT  :          2   SECURITY  :          0   ASYN RD:          66  TIMEOUTS:    0
DML     :          0   DDL       :          0   PGS/IO  :          28.7  DEADLOCK:    0
HELP INFORMATION = PF1                    applid                    PF KEY ASSIGNMENTS = PA1
    
```

You can access this screen by entering =5 . 2 on the command line and, from the Thread Activity (Collection Analysis) Screen, cursor-selecting any row from the roster.

This table describes the field changes:

Old Field Name	New Field Name	Explanation
<u>CICS TKN</u>	<u>CICS TKN</u> or ACCT TKN	This field now toggles between CICS TKN and ACCT TKN, depending on the connection type. CICS TKN displays only if the connection type is CICS ATTACH. For all other connection types, the field label will be ACCT TKN, which is short for <i>accounting token</i> . If an accounting token exists, it will be displayed as received in the DB2 data section that provides it.  CICS TKN remains cursor-selectable to take you to the TMON for CICS TS Activity Monitor; however, ACCT TKN is not cursor-selectable.
DB2	DB2 ELA	The data value has changed from displaying only DB2 nonnested elapsed time to also including elapsed time for all nested activity.

Old Field Name	New Field Name	Explanation
DB2 CPU	DB2 CPU	The data value has been changed from displaying only DB2 nonnested CPU time to also including CPU time for all nested activity.
<u>TOT ELAP</u>	ELAPSED	Even though this field has been renamed, the data value has not changed: it is still the total elapsed time. The new label, however, better represents the underlying data that displays when you cursor-select the field. The resulting screen is the Thread Event Detail screen (see <a href="#">page 36</a> ), which has been renamed Thread Elapsed Times.
Elapsed Times	Timings	The Elapsed Times section heading has been renamed Timings to better describe the broader scope of the data the section now provides.
—	NESTED	This new field displays the total elapsed time for all DB2 nested activity for the thread. Cursor-selecting this field will display the new Thread Nested Activity (Collection Analysis) screen, described on <a href="#">page 51</a> .
<u>SORT (Event Counts)</u>	SORT (Event Counts)	This field is now cursor-selectable. It displays the Thread Sort Detail (Collection Analysis) screen, which is the same screen you access by cursor-selecting the SORT field under Timings.
<u>TOT CPU</u>	TOT CPU	The data value for this field has changed from displaying only application and DB2 nonnested CPU time to also including CPU time for all nested activity. Also, cursor-selecting this field displays the Thread CPU/Elapsed Times screen, which was formerly called the Thread CPU Detail (Collection Analysis) screen (see <a href="#">page 45</a> ).
—	TOT WT	This new field displays the total wait time. This value includes I/O wait, lock wait, and all other Class 3 wait times that occur within DB2. For a breakdown of all wait times, cursor-select the ELAPSED field.

The three screens described in the next three sections are accessed from the Thread Detail (Collection Analysis) screen by cursor-selecting the ELAPSED, TOT CPU, and NESTED fields, respectively.

## Changes to the Thread Event Detail (Collection Analysis) Screen

The Thread Event Detail (Collection Analysis) screen has been renamed the Thread Elapsed Times (Collection Analysis) screen. Due to the significant number of changes that have been made to the screen, the complete documentation for it is included in this section—see [“Thread Elapsed Times \(Collection Analysis\)” on page 37](#).

Besides the name change, these changes also have been made:

- It is now a GRF screen.
- Three fields have been added:
  - DB2 Elapsed (This field has a different meaning and location than the field previously known as DB2 ELAPSED. That field has been renamed; see the table below.)
  - WAIT: USER DEF FUNC
  - WAIT: STORED PROC
- The names of these fields have been changed:

Old Field Name	New Field Name
COMMIT TOTAL	COMMIT 1PHS TOTAL
DB2 ELAPSED	DB2 NON-NESTED
INDOUBT	COMMIT 2PHS INDOUBT
PHASE1	COMMIT 2PHS PHASE 1
PHASE2	COMMIT 2PHS PHASE 2
PAGE LATCH	WAIT: PAGE LATCH
LOCK/LATCH	WAIT: LOCK/LATCH
Global Messages	WAIT: GLOBAL MSGS
Global Locks	WAIT: GLOBAL LOCK
SPROC EXECUTIONS	STORED PROCEDURE
WAIT: I/O	WAIT: SYNC I/O

- The TIMEOUTS, REJECTS, and ABENDS fields have been moved to the new Thread Nested Activity (Collection Analysis) screen, described on [page 51](#).

- Sorting has changed. Before, records were sorted on group (by type, for example, Commits) and on Elapsed time. Now, because the event names are more descriptive, the records are sorted only on Elapsed time.
- Values of greater than 100 percent are explained for the STORED PROCEDURE, TRIGGER, and USER DEF FUNCTION fields.
- By default, all fields now display their event counts by event; however, there are certain fields you can view by IFCID if desired. These are documented as *EVENTCOUNTS-enabled*. For more information, see “[EVENTCOUNTS Startup Parameter and Command](#)” on page 10.
- A second screen (accessed by scrolling to the right) shows Average Elapsed times. This is derived by dividing the total Elapsed Time by the number of events (not by the number of entry/exit IFCIDs received).

## Thread Elapsed Times (Collection Analysis)

```

JOBNAME:jobname          Thread Elapsed Times          DATE:MM/DD/YY
DB2      :db2            Collection Analysis          TIME:HH:MM:SS
COMMAND:

CONN ID: DB2CALL        PLAN   : TMDB324          DATE       : 06/29/06
CORR ID: TBTDB40T      AUTH ID: LSCSTC         START TIME: 13:33:12.7956
REASON  : NORMAL        OPID   : LSCSTC         END TIME   : 13:33:16.1745
DB2 ELAPSED:          3.3448          TOTAL ELAPSED:          3.3789
DISPLAYED: 1 TO 12 OF 28          SORT COLUMN#: 03 (1-5)  ORDER: D (A/D)
Event          Count      Elapsed      % Total Elapsed      % DB2 Elapsed
*              *          *          <----->          <----->
DB2 NON-NESTED :    396        3.3448          98.99          99.99
SORT ELAPSED   :     7         2.0936          61.96          62.59
COMMIT 1PHS TOTAL : 1         0.0359          1.06           1.07
WAIT: LOCK/LATCH : 3         0.0001          0.00           0.00
WAIT: DEALLOC EUS : 1         0.0000          0.00           0.00
WAIT: I/O       : 0         0.0000          0.00           0.00
COMMIT 2PHS PHASE 2: 0         0.0000          0.00           0.00
COMMIT 2PHS PHASE 1: 0         0.0000          0.00           0.00
WAIT: OTHR READ I/O: 0         0.0000          0.00           0.00
WAIT: OTHR WRIT I/O: 0         0.0000          0.00           0.00
ABORT ELAPSED  : 0         0.0000          0.00           0.00
WAIT: PAGE LATCH : 0         0.0000          0.00           0.00
HELP INFORMATION = PF1          applid          PF KEY ASSIGNMENTS = PA1

```

*This is a GRF screen. See the section on GRF screens in Chapter 2 of the ASG-TMON for DB2 Reference Guide for a description of its characteristics.*

The Thread Elapsed Times (Collection Analysis) screen presents thread activities (e.g., SQL, commits, sorts, drain and claim locks, and remote calls) and corresponding thread and DB2 times for a specific thread, sorted by elapsed time.

The Thread Elapsed Times (Collection Analysis) screen contains filter areas below some of the field headings in which you can enter criteria to limit the roster to specific images. For information on filtering data, see the section on GRF screens in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

---

**Note:**

You can use GRF's PROFILE command to display the GRF Profile Display/Alter screen, which allows you to save the sort order, sort column, and filtering criteria for this screen and apply them to subsequent screens. For more information, see the description of the PROFILE command in the section on GRF commands in Chapter 2 of the *ASG-TMON for DB2 Reference Guide* and Chapter 2 in *ASG-TMON Products for OS/390 & z/OS Common Components Reference Guide*.

---

## Accessing this Screen

### To access this screen

- ▶ From the Thread Detail (Collection Analysis) screen ([page 34](#)), cursor-select the ELAPSED field.

### Primary Commands

You can enter the AVERAGE, SUMMARY, VIEWNEXT, and VIEWPREV commands on the command line. See the section on primary commands under Thread Detail (Collection Analysis) in Chapter 9 of the *ASG-TMON for DB2 Reference Guide* for a description of these commands.

You also can enter these commands on the command line:

Command	Description
LEFT	Scrolls the screen left to return to the previous display. You also can cursor-select the Left field to perform this function. This command is valid only if the Left field is displayed on the screen.
PROFILE	This is a GRF command, see its description under GRF commands in Chapter 2 of the <i>ASG-TMON for DB2 Reference Guide</i> .
RIGHT	Scrolls the screen right to display additional data. You also can cursor-select the Right field to perform this function. This command is valid only if the Right field is displayed on the screen.

---

Use the DOWN and UP commands to scroll through this screen. To learn about the various ways to scroll using these commands and for syntax and descriptions of all commands, see the section on commands in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

## Fields

A field value that is highlighted and displayed in turquoise indicates an abbreviated variable-length character string, or *V-string*. You can cursor-select it to display a pop-up window showing the entire V-string data value. See [“DB2 Version 8.1 Variable-character String Long Identifiers” on page 64](#) for further information.

Because several fields in this component serve the same function on multiple screens, they are defined in Collection Analysis in Chapter 9 of the *ASG-TMON for DB2 Reference Guide*.

The (filter area), DISPLAYED: *nnnnn* TO *nnnnn* OF *nnnnn*, ORDER, and SORT COLUMN# fields are GRF fields and are described in the section on GRF screens in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

This table lists fields unique to the Thread Elapsed Times (Collection Analysis) screen:

Field	Description
% DB2 Elapsed	Displays a percentage that shows the ratio of the elapsed time of the event (the Elapsed field) to total DB2 elapsed time (the Elapsed field for the DB2 ELAPSED event). See the individual field descriptions for an explanation of each event.
% Total Elapsed	Displays a percentage that shows the ratio of the elapsed time of the event (the Elapsed field) to total thread elapsed time (the TOTAL ELAPSED field). See the individual field descriptions for an explanation of each event.
ABORT ELAPSED	Displays statistics for the number of times the allied thread processed an abort and the amount of time the allied thread has spent in abort processing.
Avg. Elapsed	Displays the average elapsed time for the associated event. This value is derived by dividing the total elapsed time by the number of events. It is <i>not</i> related to the number of entry/exit IFCIDs received.
COMMIT 1PHS TOTAL	Displays statistics for the total elapsed time spent for all aspects of single-phase commit processing.
COMMIT 2PHS INDOUBT	Displays statistics for the total time spent in INDOUBT during 2-phase commit processing. This time is spent waiting on the application in between phase 1- and phase 2-commit processing during a 2-phase commit. It could exceed the total DB2 elapsed time as it occurs in the application. (DB2 does not provide statistics on how many times the event occurred, so the count will always be zero.)

Field	Description
COMMIT 2PHS PHASE 1	Displays statistics for the total time spent in phase 1 of 2-phase commit processing. (DB2 does not provide statistics on how many times the event occurred, so the count will always be zero.)
COMMIT 2PHS PHASE 2	Displays statistics for the total time the allied thread spent in phase 2 of a 2-phase commit processing. (DB2 does not provide statistics on how many times the event occurred, so the count will always be zero.)
Count	<p>Displays the total number of occurrences of each event listed in the Event field. By default, all fields display their event counts by event, but there are certain fields you can view by IFCID if desired. For more information, see the EVENTCOUNTS command under <a href="#">“EVENTCOUNTS Startup Parameter and Command”</a> on page 10.</p> <p>See the individual field descriptions for an explanation of each event and whether they can be displayed by IFCID using the EVENTCOUNTS command. These are referred to as being <i>EVENTCOUNTS-enabled</i>.</p>
DB2 ELAPSED	Displays the sum of all DB2 nonnested and nested time.
DB2 NON-NESTED	<p>Displays statistics for the time spent IN-DB2, but not in a nested activity.</p> <p>The count does not include SQL entry/exit events processed by a stored procedure. The time is calculated by subtracting the store-clock (STCK) time on exit from DB2 from the STCK time on entry to DB2. It does not include certain abend conditions or thread allocation processes.</p> <p>This field is EVENTCOUNTS-enabled.</p>
Elapsed	<p>Displays the amount of elapsed time (in <i>HH:MM:SS.tttt</i> format) associated with the event shown in the Event field. See the individual field descriptions for an explanation of each event.</p> <p><b>Caution!</b> Do not expect the sum of the elapsed times for the events to equal the total DB2 elapsed time. It may be either greater (due to overlapping intervals for some events) or less (due to DB2 buffer pool processing that incurs no delays) than the total DB2 elapsed time. Also, be careful when correlating the I/O elapsed times on this screen to the I/O elapsed times on the Thread Resource Detail (Collection Analysis) screen. Only the I/O WAIT field on this screen can be correlated to the SYNCREAD and SYNCWRIT fields on that screen.</p>

Field	Description
	<p>Elapsed times reported for asynchronous I/O cannot be correlated. This screen provides only the time incurred by the thread while waiting for a page being retrieved through asynchronous I/O processing. The Thread Resource Detail (Collection Analysis) screen contains the total asynchronous I/O elapsed time. It is possible for asynchronous I/Os to overlap, causing the total elapsed I/O times shown on the Thread Resource Summary (Collection Analysis) screen to exceed the total DB2 elapsed time as well as the total thread elapsed time. For example, suppose a DB2 thread starts at 9:00:00 and ends at 9:00:10. Its total elapsed time is 10 seconds. Suppose that during the 10-second interval, two simultaneous asynchronous I/O requests start at 9:00:02 and end at 9:00:08. Each asynchronous I/O has an elapsed time of 6 seconds. When added together, the total elapsed I/O time is 12 seconds, which exceeds the thread's total elapsed time of 10 seconds.</p> <p>See the section on DB2 elapsed time in Appendix B of the <i>ASG-TMON for DB2 Reference Guide</i> for more information.</p>
Event	Lists the event to which the information on the remainder of the row applies. The events incurring the longest elapsed times are displayed at the top of the list. Scroll forward to display the remaining events.
SORT ELAPSED	Displays statistics for the number of times DB2 has requested a sort and the amount of time the thread has spent sorting since it began processing.
STORED PROCEDURE	Displays statistics for the number of SQL entry/exit events performed by stored procedures and the elapsed time spent executing a stored procedure.
	<p><b>Note:</b></p> <p>The elapsed time for these metrics may show a value greater than 100 percent of the DB2 elapsed time. This is possible due to the fact that the elapsed value for these fields includes the total activity time (both time spent IN-DB2 and IN-APP, where the APP is the nested activity), whereas the DB2 elapsed time includes only the time actually spent processing in DB2. Refer to the Thread CPU/Elapsed Detail (Collection Analysis) screen, described on <a href="#">page 45</a>, for a breakdown of where the elapsed time was spent for a given nested activity.</p>
TOTAL ELAPSED	Displays the amount of time that has elapsed since the thread began processing.

Field	Description
TRIGGER	<p>Displays the amount of time that elapsed while the thread was executing DB2 triggers. This field is valid only if you are running DB2 6.1 or later.</p> <p><b>Note:</b></p> <p>The Elapsed time for these metrics can show a value greater than 100 percent of the DB2 ELAPSED value. This is due to the fact that the Elapsed value for these fields includes the total activity time (both time spent IN-DB2 and IN-APP, where the APP is the nested activity), whereas the DB2 ELAPSED time includes only the time actually spent processing in DB2. Refer to the Thread CPU/Elapsed Detail (Collection Analysis) screen, described on <a href="#">page 45</a>, for a breakdown of where the Elapsed time was spent for a given nested activity.</p>
USER DEF FUNCTION	<p>Displays statistics for the number of entry/exit pairs from UDFs. This field also displays statistics for the elapsed time the SQL statement waited for the function to complete. This field is valid only if you are running DB2 6.1 or later.</p> <p><b>Note:</b></p> <p>The elapsed time for these metrics may show a value greater than 100 percent of the DB2 elapsed time. This is possible due to the fact that the elapsed value for these fields includes the total activity time (both time spent IN-DB2 and IN-APP, where the APP is the nested activity), whereas the DB2 elapsed time includes only the time actually spent processing in DB2. Refer to the Thread CPU/Elapsed Detail (Collection Analysis) screen, described on <a href="#">page 45</a>, for a breakdown of where the elapsed time was spent for a given nested activity.</p>
WAIT: ARCHIVE LOG	<p>Displays statistics for the number of ARCHIVE LOG commands issued and the accumulated time the thread was suspended because of archive log requests. This is not the total time required to complete the command.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: ARCHIVE READ	<p>Displays statistics for the number of ARCHIVE LOG READ commands issued and the accumulated time the thread was suspended because of archive log reads.</p> <p>This field is EVENTCOUNTS-enabled.</p>

---

Field	Description
WAIT: CLAIM	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate the elapsed time the thread spent waiting for claims to be released. This field also displays statistics for the accumulated time the thread waited for claims to be released.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: DB2 SRV EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for DBM1 services, including dataset define, dataset delete, dataset extend, dataset reset, and VSAM catalog access. This field also displays statistics for the elapsed time the SQL statement waited for the process to complete. This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: DEALLOC EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for DB2 commit, abort, or deallocation processing. This field also displays statistics for the elapsed time the SQL statement waited for the process to complete. This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: DRAIN	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate drain lock elapsed wait times. This field also displays statistics for the accumulated time the thread waited for a drain lock.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: GLOBAL LOCK	<p>Displays statistics for the number of wait trace events processed for waits for global lock contention. This field also displays statistics for the elapsed time the thread spent waiting for a suspension of an IRLM lock request due to contention with a global lock, which required intersystem communication to resolve. The count and time values are derived from the QWACARNJ and QWACAWTJ fields, respectively, of the IFCID 003 record.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: GLOBAL MSGS	<p>Displays statistics for the number of wait trace events processed for waits sending messages to other members in a data-sharing group (DSG). This field also displays statistics for the elapsed time the thread has spent waiting due to suspension for sending messages. The count and time values are derived from the QWACARNJ and QWACAWTG fields, respectively, of the IFCID 003 record.</p> <p>This field is EVENTCOUNTS-enabled.</p>

Field	Description
WAIT: LOCK/LATCH	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate lock and latch elapsed wait times. This field also displays statistics for the amount of time the thread has spent in lock wait since it began processing in DB2. The lock wait is caused by <i>local</i> lock contention, which does not require intersystem communication to resolve.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: LOG WRITE I/O	<p>Displays statistics for the number of entry/exit pairs from wait events processed for log write I/O. This field also displays statistics for the elapsed time the SQL statement waited for the I/O to complete. This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: OPEN/CLS EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for dataset open/close or HSM recall processing. This field also displays statistics for the elapsed time the SQL statement waited for the process to complete. This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: OTHER EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for other execution unit switch processing. This field also displays statistics for the elapsed time the SQL statement waited for the process to complete. This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: OTHR READ I/O	<p>Displays statistics for the number of wait trace events processed for waits for read I/O under another thread (e.g., a system agent) since this thread began processing in DB2. This field also displays statistics for the accumulated wait time for read I/O performed under a thread other than this one. This value is calculated for threads by subtracting the STCK time on entry to event wait from STCK on resume from the event.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: OTHR WRIT I/O	<p>Shows statistics for the number of wait trace events processed for waits for write I/O under another thread (e.g., a system agent) since this thread began processing in DB2. This field also displays statistics for the accumulated wait time for write I/O performed under a thread other than this one. This value is calculated for threads by subtracting the STCK time on entry to event wait from STCK on resume from the event.</p> <p>This field is EVENTCOUNTS-enabled.</p>

---

Field	Description
WAIT: PAGE LATCH	<p>Displays statistics for the number of entry/exit pairs from wait events processed to calculate page latch elapsed wait times and the accumulated time the thread waited for a page latch.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: STORED PROC	<p>Displays statistics for the number of times a SQL CALL statement had to wait for an available TCB and the elapsed time the thread spent waiting for an available TCB to schedule a stored procedure. The count and time values are derived from the QWACCANM and QWACCAST fields, respectively, of the IFCID 003 record.</p>
WAIT: SYNC I/O	<p>Displays statistics for the number of entry/exit pairs from I/O events processed to calculate I/O elapsed wait times. This field also displays statistics for the amount of time the allied thread has spent waiting for synchronous read and write I/O since it began processing in DB2.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: SYSLGRNG EUS	<p>Displays statistics for the number of entry/exit pairs from wait events processed for SYSLGRNG recording processing. This field also displays statistics for the elapsed time the SQL statement waited for the process to complete. This field is valid only if you are running DB2 6.1 or later.</p> <p>This field is EVENTCOUNTS-enabled.</p>
WAIT: USER DEF FUNC	<p>Displays the wait time a UDF could have incurred while waiting to be run. If the UDF WAIT TIME is zero, this field displays a count of 0 (zero); otherwise, it will be 1.</p>

## Changes to the Thread CPU Detail (Collection Analysis) Screen

The Thread CPU Detail (Collection Analysis) screen has been renamed the Thread CPU/Elapsed Detail (Collection Analysis) screen. Because of the significant number of changes that have been made to the screen, the complete documentation for it is included in this section—see [“Thread CPU/Elapsed Detail \(Collection Analysis\)” on page 46](#).

Besides the name change, these changes also have been made:

- It is now a GRF screen.
- The TCB time and SRB time columns have been replaced with two graph fields, % CPU and % ELAPSED.
- The APPLICATION and DB2 fields that used to be under the % of Total heading are now covered under % CPU and % ELAPSED in conjunction with the TOTAL category.

- The number of fields has increased from 13 to 24 to provide additional metrics. (This count does not include these column headings: Elapsed, CPU, % ELAPSED, and % CPU.)
- Many of the fields have been renamed.
- The hierarchical information in the Times column is more clearly delineated with summation (-) and detail (..) characters preceding each row listed below the OVERALL TOTAL field.

## Thread CPU/Elapsed Detail (Collection Analysis)

```

JOBNAME: jobname          Thread CPU/Elapsed Detail          DATE: MM/DD/YY
DB2      : db2            Collection Analysis              TIME: HH:MM:SS
COMMAND:
                                     _VNXT?
CONN ID: DB2CALL          PLAN   : TMDB324          DATE       : 07/21/06
CORR ID: ROTDB41T        AUTH ID: LSCSTC          START TIME: 11:43:48.7186
REASON  : NORMAL         OPID   : LSCSTC          END TIME  : 11:43:54.5373
DISPLAYED: 1 TO 14 OF 17
Times          CPU          Elapsed          % CPU          % ELAPSED
OVERALL TOTAL: 1.0433      5.8187           100.00         100.00
..IN-APP      : 1.0127      3.2061           97.06          55.10
..IN-DB2      : 0.0307      2.6126          2.93           44.89
-NON NESTED   : 1.0433      5.8187           100.00         100.00
..IN-APP      : 1.0127      3.2061           97.06          55.10
..IN-DB2      : 0.0307      2.6126          2.93           44.89
-NESTED       : 0.0000      0.0000          0.00           0.00
..IN-APP      : 0.0000      0.0000          0.00           0.00
..IN-DB2      : 0.0000      0.0000          0.00           0.00
-STORED PROC : 0.0000      0.0000          0.00           0.00
..IN-APP      : 0.0000      0.0000          0.00           0.00
..IN-DB2      : 0.0000      0.0000          0.00           0.00
-UDF          : 0.0000      0.0000          0.00           0.00
..IN-APP      : 0.0000      0.0000          0.00           0.00
HELP INFORMATION = PF1          applid          PF KEY ASSIGNMENTS = PA1
    
```

This is a GRF screen. See the section on GRF screens in Chapter 2 of the ASG-TMON for DB2 Reference Guide for a description of its characteristics.

The Thread CPU/Elapsed Detail (Collection Analysis) screen presents detailed CPU information for a specific thread, based on level of detail and summation. You can use this information to pinpoint thread CPU resource consumption problems as well as thread resource problems.

Information in the Times column is arranged in hierarchical fashion underneath the OVERALL TOTAL field. Fields preceded by dashes indicate summary values, and fields preceded by periods indicate a subset of the last field preceded by a dash. For a specific description of each row and column, see [“Fields” on page 47](#).

## Accessing this Screen

### To access this screen

- ▶ From the Thread Detail (Collection Analysis) screen ([page 34](#)), cursor-select the TOT CPU field.

### Primary Commands

You can enter the AVERAGE, SUMMARY, VIEWNEXT, and VIEWPREV commands on the command line. See the section on primary commands in the section on the Thread Detail (Collection Analysis) screen in Chapter 9 of the *ASG-TMON for DB2 Reference Guide* for a description of these commands.

For syntax and descriptions of all commands, see the section on commands in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

### Fields

A field value that is highlighted and displayed in turquoise indicates an abbreviated variable-length character string, or *V-string*. You can cursor-select it to display a pop-up window showing the entire V-string data value. See “[DB2 Version 8.1 Variable-character String Long Identifiers](#)” on [page 64](#) for further information.

The DISPLAYED: *nnnnn* TO *nnnnn* OF *nnnnn* field is a GRF field and is described in the section on GRF screens in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

Because several fields in this component serve the same function on multiple screens, they are defined in the section on Collection Analysis in Chapter 9 of the *ASG-TMON for DB2 Reference Guide*. This table lists fields unique to the Thread CPU/Elapsed Detail (Collection Analysis) screen:

Field	Description
% CPU <sup>†</sup>	<p>Displays a graphical representation of the value shown in the CPU field for this category.</p> <p><b>Note:</b> _____            Non-total values in this column may be greater than 100 percent. This is possible due to the fact that threads are sometimes reported across multiple accounting records when they are involved in thread reuse, rollup accounting, and/or parallel activity.</p>

Field	Description
% ELAPSED <sup>†</sup>	<p>Displays a graphical representation of the value shown in the Elapsed field for this category.</p> <p><b>Note:</b></p> <p>Non-total values in this column may be greater than 100 percent. This is possible due to the fact that threads are sometimes reported across multiple accounting records when they are involved in thread reuse, rollup accounting, and/or parallel activity.</p>
CPU	Displays the total CPU time the thread consumed when processing in the activity, in <i>HH:MM:SS.tttt</i> format.
Elapsed	Displays the amount of time that elapsed while the thread processed in the activity, <i>HH:MM:SS.tttt</i> format. The field descriptions for some of the rows under this heading contain additional information about this column.
ENCLAVE <sup>†</sup>	Displays the amount of time the thread consumed while executing triggers under an enclave.
IN-APP (NESTED) <sup>†</sup>	Displays the amount of time the thread spent executing within nested activities.
IN-APP (NON NESTED) <sup>†</sup>	Displays the amount of time the thread spent executing within nonnested activities.
IN-APP (OVERALL TOTAL)	<p>Displays the amount of time the thread used for processing both nested and nonnested activities.</p> <p><b>Note:</b></p> <p>OVERALL TOTAL IN-APP time includes both nonnested and nested IN-APP time. From a user's application perspective, the IN-APP time might call a stored procedure and the time spent in the stored procedure could be viewed as all IN-DB2 time. However, the nested activity IN-APP time is classified as part of the value in OVERALL TOTAL IN-APP since it is indeed time spent outside of DB2.</p>
IN-APP (STORED PROC)	<p>Displays the amount of time the thread spent executing within stored procedure activities.</p> <p>The value in the Elapsed field is computed using the thread package segments that were logged by a stored procedure. If thread package segments are not logged, N/A is displayed.</p>

Field	Description
IN-APP (UDF)	<p>Displays the total amount of time the thread spent executing within UDF activities. This field is valid only if you are running DB2 6.1 or later.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p>
IN-APP (zIIP) <sup>†</sup>	Displays the amount of time consumed while executing in an activity on an IBM zIIP processor.
IN-DB2 (NESTED) <sup>†</sup>	Displays the amount of time the thread spent executing in DB2 within a nested activity (i.e., stored procedures, UDFs, and triggers). Times are broken down in a hierarchy below this row, showing the details on where this total nested IN-DB2 time was spent.
IN-DB2 (NON NESTED) <sup>†</sup>	<p>Shows the amount of time the thread spent executing nonnested activity in DB2.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p>
IN-DB2 (OVERALL TOTAL)	Shows the amount of time the thread used in the local DB2.
IN-DB2 (STORED PROC)	<p>Displays the amount of time the thread spent executing DB2 SQL from the stored procedure.</p> <p>The value in the Elapsed field is computed using the thread package segments that were logged by a stored procedure. If thread package segments are not logged, N/A is displayed.</p>
IN-DB2 (UDF)	<p>Displays the total amount of time the thread spent executing DB2 SQL from UDF activities. This field is valid only if you are running DB2 6.1 or later.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p>
IN-DB2 (zIIP) <sup>†</sup>	Displays the total amount of time the thread consumed while executing DB2 SQL on an IBM zIIP processor.
NESTED <sup>†</sup>	Shows the amount of time the thread spent executing in nested activities.
NON-ENCLV <sup>†</sup>	Displays the amount of time spent executing triggers, but excluding time spend on a zIIP processor and time spent under an enclave. In other words, this is the time not accounted for in displayed enclave or zIIP times.
NON NESTED <sup>†</sup>	Shows the amount of time the thread spent executing in nonnested activities.

Field	Description
OVERALL TOTAL	<p>Shows the total amount of time the thread consumed while processing in the activity.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p>
REMOTE DB2	<p>Displays the amount of time the thread spent processing at the remote DB2(s).</p> <p>The values in the CPU and Elapsed fields are only available for threads using system-directed access.</p> <p>When the thread is executing remotely using DRDA protocols, no remote CPU times are available. If the thread executed remotely using only DRDA protocols, the remote DB2 CPU time will contain a value of zero.</p> <p>For more information on DRDA and private protocols, see IBM's DB2 Administration Guide for distributed environment information on remote access types.</p>
STORED PROC	<p>Shows the amount of time the thread spent executing a stored procedure.</p>
TRIGGER	<p>Displays the amount of time the thread consumed while executing DB2 triggers.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time. This field is valid only if you are running DB2 6.1 or later.</p>
UDF	<p>Displays the total amount of time the thread consumed while executing UDFs. This field is valid only if you are running DB2 6.1 or later.</p> <p>The value in the Elapsed field includes DB2 processing time plus activity processing time.</p>
zIIP <sup>†</sup>	<p>Displays the total amount of time the thread consumed while executing on an IBM zIIP processor in all environments.</p>
zIIP ELIGIBLE <sup>†</sup>	<p>Displays the total amount of time the thread spent on work that was zIIP-eligible, but was not executed on a zIIP processor.</p>
zIIP (TRIGGER) <sup>†</sup>	<p>Displays the amount of time the thread consumed while executing triggers on an IBM zIIP processor.</p>

<sup>†</sup> This field is new.

## New Thread Nested Activity (Collection Analysis) Screen

The Thread Nested Activity (Collection Analysis) screen is new. This section contains a sample of the screen and its complete documentation.

```

JOBNAME: jobname          Thread Nested Activity          DATE: MM/DD/YY
DB2      : db2            Collection Analysis          TIME: HH:MM:SS
COMMAND:
                                _VNXT?
CONN ID: DB2CALL          PLAN   : TMDB017          DATE       : 07/24/06
CORR ID: TBTDB40T        AUTH ID: LSCSTC          START TIME: 10:28:45.1008
REASON  : NORMAL         OPID   : LSCSTC          END TIME   : 10:28:45.1010
DB2 ELAPSED:              0.0001          TOTAL ELAPSED:          0.0002

Activity                    CPU                    Elapsed                Invocations
STORED PROCEDURES :          0.0000                0.0000                  0
USER DEFINED FUNCS :          0.0000                0.0000                  0
TRIGGERS           :          0.0000                0.0000                  0
TOT NESTED ACTIVITY:          0.0000                0.0000                  0

Exceptions   Stored Proc      UDF      Trigger Info          Count
ABENDS      :                0            0      STATEMENT TRGRS :          0
REJECTS     :                0            0      ROW TRIGGERS   :          0
TIMEOUTS    :                0            0      SQL ERRORS     :          0
Max Nesting Levels :          0

HELP INFORMATION = PF1          applid          PF KEY ASSIGNMENTS = PA1

```

The Thread Nested Activity (Collection Analysis) screen presents nested activity for a completed thread. The bottom of the screen contains these sections:

- The Activity section shows the CPU and elapsed times spent by stored procedures, UDFs, and triggers (including times spent under an enclave). It also shows the number of times each activity was invoked.
- Information in the Exceptions section is presented only for stored procedures and UDFs.
- The Trigger Info section gives additional information pertaining to triggers.

### Accessing this Screen

#### To access this screen

- From the Thread Detail (Collection Analysis) screen ([page 34](#)), cursor-select the NESTED field.

### Primary Commands

You can enter the AVERAGE, SUMMARY, VIEWNEXT, and VIEWPREV commands on the command line. See the section on primary commands in the section on Thread Detail (Collection Analysis) in Chapter 9 of the *ASG-TMON for DB2 Reference Guide* for a description of these commands.

For syntax and descriptions of all commands, see the section on commands in Chapter 2 of the *ASG-TMON for DB2 Reference Guide*.

## Fields

A field value that is highlighted and displayed in turquoise indicates an abbreviated variable-length character string, or *V-string*. You can cursor-select it to display a pop-up window showing the entire V-string data value. See “[DB2 Version 8.1 Variable-character String Long Identifiers](#)” on page 64 for further information.

Because several fields in this component serve the same function on multiple screens, they are defined in the section on Collection Analysis in Chapter 9 of the *ASG-TMON for DB2 Reference Guide*. This table lists fields unique to the Thread Nested Activity (Collection Analysis) screen:

Field	Description
ABENDS	Displays the number of times a stored procedure or UDF terminated abnormally.
CPU	Displays the total CPU time the thread consumed while processing in the activity, in <i>HH:MM:SS.tttt</i> format.
DB2 ELAPSED	Shows the total amount of time the thread consumed while processing in DB2; includes nested and nonnested time.
Elapsed	Displays the amount of time that the thread spent processing in the activity, in <i>HH:MM:SS.tttt</i> format. The field descriptions for some of the rows under this heading contain additional information about this column.
Invocations	Displays the number of times the activity was called.
Max Nesting Levels	Displays the depth of nesting that occurred with all stored procedures, UDFs, and DB2 triggers. The maximum nesting depth is 16.
REJECTS	Displays the number of SQL CALL statements rejected because the stored procedure or UDF was in STOP ACTION(REJECT) state.
ROW TRIGGERS	Displays the number of DB2 row triggers that were executed by the thread.
SQL ERRORS	Displays the number of SQL errors the thread encountered while executing DB2 triggers.
STATEMENT TRGRS	Displays the number of DB2 statement triggers that were executed by the thread.

---

Field	Description
STORED PROCEDURES	Shows the amount of time the thread spent in nested activity while executing stored procedures.
TIMEOUTS	Displays the number of SQL CALL statements for stored procedures and UDFs that timed out while waiting to be scheduled.
TOTAL ELAPSED	Shows the total amount of time the thread consumed while processing in the activity. This includes DB2 processing time plus activity processing time.
TOT NESTED ACTIVITY	Displays the total CPU time, elapsed time, and invocation count for the stored procedures, UDFs, and DB2 trigger activities.
TRIGGERS	Displays the total amount of time the thread spent in nested activity while executing DB2 triggers; this includes both enclave and non-enclave times. This field is valid only if you are running DB2 6.1 or later.
USER DEFINED FUNCS	Displays the total amount of time the thread spent in nested activity while executing UDFs. This field is valid only if you are running DB2 6.1 or later.  The value in the ELAPSED field includes DB2 processing time plus activity processing time.

## DB2MAINT Startup Parameter and Command

The DB2MAINT startup parameter and command give you greater control over how TMON for DB2 should react when it encounters a DB2 IFCID data section at run time that is not of the minimum expected length.

TMON for DB2 will issue a TMDB35x01E message when this type of problem (i.e., a *DB2MAINT problem*) is occurring (see your TMON for DB2 job log).

DB2MAINT problems can be caused by problems in either DB2 or TMON for DB2. They occur due to differences between the version of IBM macros that were available when TMON for DB2 was assembled and the current level of DB2 you are running. This situation results in TMON for DB2 expecting a different-sized IFCID data section than DB2 provides.

For certain functions, the TMON for DB2 screens might contain erroneous or incomplete data. In other situations, TMON for DB2 might not be able to continue processing.

The DB2MAINT parameter and command both provide runtime integrity checking of IBM macros that was not available before. Now, TMON for DB2 compares the expected IBM trace record lengths against documented lengths. This check ensures that a change in either DB2 or TMON for DB2 will not regress trace record collection.

See the explanation and recommended action under the TMDB35101E message in the *ASG-TMON for DB2 Messages and Codes Guide*. ASG also recommends that you contact ASG Customer Support.

## DB2MAINT Startup Parameter

This example shows the syntax for the DB2MAINT startup parameter:

```
DB2MAINT=USEDATA  
DB2MAINT=IGNOREDATA
```

where:

USEDATA (the default) makes TMON for DB2 use the IFCID data and continue as if there were no problem. Note these ramifications:

- If DB2 maintenance has shortened an IFCID data section, the data that TMON for DB2 expects may not exist.
- If TMON for DB2 is using a newer copy of DB2 macros than the level of DB2 being used, the data that TMON for DB2 expects will not be present. Check with ASG Customer Support, however, as not all IFCID information received by TMON for DB2 from DB2 trace records is used.

IGNOREDATA makes TMON for DB2 ignore the IFCID data as if it were not returned by DB2. Depending on the recovery logic, processing might not continue. For certain functions, TMON for DB2 screens might contain erroneous or incomplete data.

## DB2MAINT Command

You can issue the DB2MAINT command from anywhere in TMON for DB2. You can use it to either query the current setting or change it. If you are making a change, then you must have MONCNTL ALTR authority.

This example shows the DB2MAINT command:

```
DB2MAINT=? | IGNOREDATA | USEDATA
```

where:

? displays a message informing you of the current setting. No authorization is needed to use this parameter.

IGNOREDATA causes TMON for DB2 to treat the IFCID data section as if it was not returned by DB2. TMON for DB2 might not continue processing, depending on the recovery logic of the TMON for DB2 component requesting the IFCID data.

USEDATA makes TMON for DB2 use the information in the IFCID data section as if there were no problem. These are the ramifications of using this setting:

- If DB2 maintenance has shortened an IFCID data section, the data that TMON for DB2 expects might not exist.
- If TMON for DB2 is using a newer copy of DB2 macros than the level of DB2 being used, the data that TMON for DB2 expects will not be present; however, check with ASG Customer Support, as not all IFCID information received by TMON for DB2 from DB2 trace records is used.

## Tips about Navigating from TMON for CICS TS

If you wish to navigate from TMON for CICS TS to TMON for DB2, you must include a specification in the CICS DB2CONN definition. You must specify UOW for the ACCOUNTREC parameter. Any other setting will change the timestamps of DB2 threads so they are out of sync with CICS transactions and will result in failed searches for the unit of work token thread (cursor-selecting the UOW field on the TMON for CICS TS Detail Transaction Data screen).

## New IOTHROTTLE Parameter

DB2 produces I/O IFCIDs that TMON for DB2 uses in monitoring performance. When DB2 is saturated with physical I/O, however, the benefit of collecting this amount of data can be lost due to increased overhead in DB2 and TMON for DB2, resulting in buffer pool full failures and data loss.

A new TMON for DB2 startup parameter, IOTHROTTLE, allows TMON for DB2 to recognize extreme spikes of I/O and react intelligently, suspending collection of detailed I/O statistics until the I/O IFCID arrival rate drops. IOTHROTTLE has two accompanying optional threshold parameters, IOTHRESH1 and IOTHRESH2, which allow you to change the default values to adjust the thresholds based on your environment.

This is the syntax of the new IOTHROTTLE parameter:

```
IOTHROTTLE  
IOTHRESH1=nnnn  
IOTHRESH2=nnnn
```

where *nnnn* is a user-defined threshold I/O IFCID arrival rate count per minute that overrides the default.

## IOTHROTTL Parameter Actions

Specifying the IOTHROTTL parameter activates these actions in TMON for DB2:

- When the I/O IFCID arrival rate is over 1,000 per minute (an internal TMON for DB2-defined threshold), the PAT/CTG update for I/O counts on the Page Set Activity screen (option 2.3) is temporarily bypassed until a 1-minute interval with an I/O IFCID arrival rate count below the 1,000 threshold occurs.

Message TMDB05004I is issued to the console or Terminal Monitor Program (TMP) log when the 1,000 threshold is exceeded. Message TMDB05006I is issued to the TMP log when the I/O IFCID arrival rate drops below the 1,000 per minute threshold. (For further information, see the error message descriptions for TMDB05004I and TMDB05006I in the *ASG-TMON for DB2 Messages and Codes Guide*.)

You can override the 1,000 per minute I/O IFCID arrival rate threshold by specifying an alternate value in the IOTHRESH1=*nnnn* threshold parameter.

- When the I/O IFCID arrival rate is over 4,000 per minute (also an internal TMON for DB2-defined threshold), no I/O IFCIDs are processed and collection of I/O IFCIDs and resource segments is temporarily suppressed until the next 1-minute interval with an I/O IFCID arrival rate count below the 4,000 threshold occurs.

Message TMDB05005I is issued to the console or TMP log when the 4,000 threshold is exceeded. Message TMDB05007I is issued to the TMP log when the I/O IFCID arrival rate drops below the 4,000 per minute threshold. (For further information, see the error message descriptions for TMDB05005I and TMDB05007I in the *ASG-TMON for DB2 Messages and Codes Guide*.)

You can override the 4,000 per minute I/O IFCID arrival rate threshold by specifying an alternate value in the IOTHRESH2=*nnnn* parameter.

- Threads collected while pageset collection is bypassed cause message TMDB13001W to be displayed on the Thread Resource Summary screen. (For further information, see the error message description for TMDB13001W in the *ASG-TMON for DB2 Messages and Codes Guide*.)
- If pageset segment collection is suspended, PAT/CTG I/O update also is suspended.

## Enhanced Dump Handling

Although software is designed to run without errors or problems, events can occur or environmental issues can arise that cause logic errors. The accepted method of dealing with these errors is to take a diagnostic dump, and TMON for DB2 is no exception to this industry-wide standard. Since, however, some TMON for DB2 customer sites require

uninterrupted service and are reluctant to diagnose logic errors that can occur in a volatile system environment, TMON for DB2 now includes enhancements to eliminate controlled or diagnostic abends and dumps.

The dump handling modifications are delivered in two PTFs. The first is delivered in PTF TE02504 and addresses the SQL Analyzer feature. The second is delivered in PTF TE02559 and addresses the TMON for DB2 TMP.

## SQL Analyzer Modification

SQL Analyzer was designed to produce a complete diagnostic dump for any failure, regardless of whether it occurred because of incomplete control blocks in the DB2 environment, abends in DB2 or TMON for DB2, or detected internal errors. Because SQL Analyzer is a production-ready, high-volume data collection tool for first-time problem diagnosis, more inconsistencies than usual can occur, resulting in frequent and sometimes unnecessary or unwanted dumps.

SQL Analyzer now provides a new field on the SQL Analyzer Controls screen (option 11.1) that allows you to accept and log all SQL or DB2 errors without producing dumps for already known issues. An updated message issued at shutdown provides statistics for all the errors, regardless of the dump options specified.

### New ACTIVATE DIAGNOSTIC DUMPS? Field

This new field replaces the former ACTIVATE ASG DIAGNOSTICS field and allows you to change your dump options without ASG assistance or direction.

```

JOBNAME: jobname                SQL Analyzer Controls                DATE: MM/DD/YY
DB2      : db2                  TIME: HH:MM:SS
COMMAND:

Global Collection Controls      SQL/Package Collection Options
ACTIVATE AT NEXT CONNECT? : Y  LOG PKG TABLE STATISTICS?: Y
SQL RECORDS LOGGING DD      : 04  LOG BUFFER POOL STATISTICS?: Y
PACKAGE RECORDS LOGGING DD: 05  SQL Only Collection Options
                                GROUP REPEATING STATEMENTS?: Y
SERVICEABILITY CODES: _____  GROUP REPEATING FETCHES?   : Y
ACTIVATE DIAGNOSTIC DUMPS? Y  LOG INPUT HOST VARIABLES?  : Y
# OF ERRORS BEFORE DUMP:   050  LOG SQL TEXT?              : Y

LINE CMDS: ACT,DEA,STA,STO,ADD,DEL,EDI      PROFILE 01 TO 07 OF 08
      Profile  Profile                      +-- Profile Status --+
      Name    Description                   Current  Next Connect
$DEFAULT  SQL ANALYZER SAMPLE PROFILE      STOPPED  INACTIVE
TEST0001  SQL ANALYZER SAMPLE PROFILE      STOPPED  INACTIVE
TEST0002  SQL ANALYZER SAMPLE PROFILE      STOPPED  ACTIVE
TEST0003  SQL ANALYZER SAMPLE PROFILE      STOPPED  INACTIVE
THEWORLD  SQL ANALYZER SAMPLE PROFILE      STOPPED  INACTIVE
TST00001  SQL ANALYZER SAMPLE PROFILE      STOPPED  ACTIVE
TST00002  SQL ANALYZER SAMPLE PROFILE      STOPPED  INACTIVE
HELP INFORMATION = PF1                    applid                PF KEY ASSIGNMENTS = PA1
    
```

Valid values are Y and N:

- Specify Y to display a second field, # OF ERRORS BEFORE DUMP, where you can specify the number of suppressions to occur between dumps. Valid values are 0 through 999. A value of 0 provides dumps for all errors with unique PSWs. The default value is 50, which causes SQL Analyzer to wait until 50 errors have occurred before testing PSWs and taking a dump. Regardless of the number you specify, only one dump per unique PSW encountered will occur. Shutdown statistics indicating that these errors are occurring is printed in the monitor log.
- If you specify N in the ACTIVATE DIAGNOSTIC DUMPS? field, SQL Analyzer produces no dumps under any circumstances; however, shutdown statistics to inform you that these errors are occurring are still printed in the monitor log.

### Updated Message

Messages produced at shutdown document any errors that have occurred, regardless of the dump option settings you specify in the ACTIVATE DIAGNOSTIC DUMPS? field. These TMDB62201I messages are used to document the abend recoveries at shutdown along with other SQL Analyzer shutdown statistics:

```
TMDB62201I - SQL ANALYZER: DISTRIBUTED ABEND RECOVERIES.....n
TMDB62201I - SQL ANALYZER: LOCAL ABEND RECOVERIES.....n
```

where *n* is the number of abends encountered for the associated category.

## TMON for DB2 Monitor Modification

Like SQL Analyzer, the TMON for DB2 monitor was designed to abend and take a dump when it encounters bad data, logic errors occur, or other unpredictable events affect a normal code processing path. Unfortunately, users have not had the option of ignoring occasional or insignificant errors that might never affect normal processing.

In some cases, dumping is the default reaction to a normal loss of some functionality that can restrict a portion of monitor operation. In many of these situations (e.g., when the TMON Log File Services address space is not yet initialized or has been shut down), the code should be able to intelligently interpret the situation, bypass taking a dump, issue a message indicating the potential error, and continue operations.

Acknowledging this preferred method of dump handling, ASG has introduced two new fields, three new messages, and one updated message in TMON for DB2 to control and document internal diagnostics and dumping.

## New Fields

Two new fields are available on the Current Monitor Options and Modify Monitor Options screens (options 8.2.1 and 8.2.2, respectively) that allow you to control the TMON for DB2 dumping options:

```

JOBNAME: jobname          Current Monitor Options          DATE: MM/DD/YY
DB2      : db2            TIME: HH:MM:SS
COMMAND:

+----- Parameter Descriptions -----+
LFS CONTROL REGION ID      : ROTDB40
NUMBER OF READS BUFFERS    : 23
READS BUFFER SIZE IN K     : 0256
CURRENT SYSTEM STATISTICS AUTOSTART?: N
EXCEPTION AUTOSTART        : N
ACTIVATE DIAGNOSTIC DUMPS? : Y
# OF ERRORS BEFORE DUMP    : 000
+----- DB2 Subsystems Monitored -----+
DB73  D71H  D8AH  D8BH
Supertrace Storage Areas   Slots   Size
LOG BUFFER POOL           :    0100  800K
TRANSLATED OBJECT TABLE  :    1000   24K
UNIT OF WORK TABLE       :    0300   31K
SQL STATEMENT EVENT TABLE :    0500  101K
CONDENSED EVENT TABLE    :    0512   66K
DATA OBJECT TABLE        :    0500   81K
LOCK HISTORY ELEMENT TABLE:    3000   94K
HELP INFORMATION = PF1      applid      PF KEY ASSIGNMENTS = PA1

```

```

JOBNAME: jobname          Modify Monitor Options          DATE: MM/DD/YY
DB2      : db2            TIME: HH:MM:SS
COMMAND:
TMDB00104I - PRESS ENTER THEN END TO SAVE; CANCEL TO VOID CHANGES

+----- Parameter Descriptions -----+
LFS CONTROL REGION ID      : ROTDB40
NUMBER OF READS BUFFERS    : 23
READS BUFFER SIZE IN K     : 0256
CURRENT SYSTEM STATISTICS AUTOSTART?: N
EXCEPTION AUTOSTART        : N
ACTIVATE DIAGNOSTIC DUMPS? : Y
# OF ERRORS BEFORE DUMP    : 000
+----- DB2 Subsystems Monitored -----+
DB73  D71H  D8AH  D8BH
Supertrace Storage Areas   Slots   Size
LOG BUFFER POOL           :    0100  800K
TRANSLATED OBJECT TABLE  :    1000   24K
UNIT OF WORK TABLE       :    0300   31K
SQL STATEMENT EVENT TABLE :    0500  101K
CONDENSED EVENT TABLE    :    0512   66K
DATA OBJECT TABLE        :    0500   81K
LOCK HISTORY ELEMENT TABLE:    3000   94K
HELP INFORMATION = PF1      applid      PF KEY ASSIGNMENTS = PA1

```

### ACTIVATE DIAGNOSTIC DUMPS?

Allows you to specify whether you want TMON for DB2 to take a diagnostic dump when a data or internal logic error occurs. Valid values are Y and N.

- If you specify Y, a 1-time internal diagnostic dump can be taken to assist ASG in diagnosing the error, depending on the threshold and number of diagnostics encountered.
- Specify N to disable all diagnostic dumping, including any requested through monitor input parameters or commands.

### # OF ERRORS BEFORE DUMP

Allows you to set a threshold for the number of errors that need to occur before a diagnostic dump is taken. Valid values are 000 through 999; the default is 010. For example, if you specify a value of 005 and the ACTIVATE DIAGNOSTIC DUMPS? flag also is set to Y, TMON for DB2 will produce a diagnostic dump when the sixth error occurs.

Specify a value of 000 to enable diagnostic dumping for all errors or when a diagnostic PTF has been provided and the TMDBDIAG command is required.

The Current Global Statistics screen (option 8.2.3) lists the dump types and number of occurrences of each.

```
JOBNAME: jobname          Current Global Statistics          DATE: MM/DD/YY
DB2      : db2              TIME: HH:MM:SS
COMMAND:

                                STATISTICS 52 TO 62 OF 62
+----- Current Global Monitor Statistics -----+
COUNT OF LMS ABENDS           :                               0
COUNT OF QMG ABENDS           :                               0
COUNT OF TMR ABENDS           :                               0
COUNT OF TRD ABENDS           :                               0
COUNT OF PSF ABENDS           :                               0
COUNT OF DB2 ABENDS           :                               0
COUNT OF SET ABENDS           :                               0
COUNT OF FLT ABENDS           :                               0
COUNT OF CPOOL ABENDS         :                               0
COUNT OF CPALLOC ABENDS       :                               0
COUNT OF OTHER UNCATEGORIZED ABENDS :                               0

HELP INFORMATION = PF1          applid          PF KEY ASSIGNMENTS = PA1
```

## Updated Message

These TMDB33503I messages are issued at TMP shutdown detailing any abends that have occurred, regardless of the dump option settings specified:

```

TMDB33503I COUNT OF VSF ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF LMS ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF QMG ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF TMR ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF TRD ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF PSF ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF DB2 ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF SET ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF FLT ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF CPOOL ABENDS ENCOUNTERED.....n
TMDB33503I COUNT OF CPALLOC ABBENDS ENCOUNTERED.....n
TMDB33503I COUNT OF UNCATEGORIZED ABENDS ENCOUNTERED.....n

```

where *n* is the number of abends encountered for the associated category.

## SQL Analyzer Exclusion Profiling

*Exclusion profiling* allows you to specify additional criteria for *not* collecting SQL Analyzer data, providing an exception to the rule for profiles that are set for collection. First, exclusion profiling evaluates SQL to see if it should be excluded from collection (through active exclusion profiles). If SQL is excluded, it is *never* considered for collection. If it is not excluded, it is compared with collection profiles to see if it should be collected.

This performance improvement allows you to set profiles to bypass collection at the front of the SQL Analyzer path for certain applications (e.g., those with known large fetch loops) while still collecting the critical application data required for performance analysis. It also reduces the number of profiles required to target specific data for logging, allowing more intelligent data collection. You can specify a more generic collection profile with specific exceptions to the collection rule(s) customized to your site-specific needs.

You implement exclusion profiling by defining individual profiles that specify exceptions to the criteria contained in the active collection profiles. These exceptions can be based only on fields that are available without entering the SQL Analyzer collector. To clearly indicate the fields that are eligible for exclusion, the STATEMENT #, STATEMENT TYPE, and DYN SQL ONLY? fields have been moved to the left side of the SQL Analyzer Collection Profile screen (see [“Updated Screen” on page 62](#)). TMON for DB2 applies an exclusion profile prior to any other collection profile type (i.e., static, dynamic, or exception).

**Note:**

Specifying N in both the EXCLUDE PACKAGE RECORDS? and EXCLUDE SQL STATEMENT RECORDS? fields forces the specification for statement records to Y (i.e., indicating that they should be excluded). A message noting the change is displayed.

## Updated Screen

This example of the updated SQL Analyzer Collection Profile screen excludes from entering the collector all threads running as plan DB2TEST1 (i.e., all records will be excluded from collection):

```
JOBNAME: jobname          SQL ANALYZER COLLECTION PROFILE          DATE: MM/DD/YY
DB2      : db2              TIME: HH:MM:SS
COMMAND: _____
TMDB64011I - CAUTION: DIAG DUMPS SHOULD ONLY BE USED WITH ASG GUIDANCE
PROFILE DESCRIPTION : EXCLUDE PLAN DB2TEST1
PROFILE NAME       : EXCLUDE          EXCLUDE PACKAGE RECORDS?      : Y
PROFILE LAST UPDATED:                EXCLUDE SQL STATEMENT RECORDS?: Y

THREAD TARGETS FOR PKG/SQL RECORDS      THREAD LIMITS FOR SQL RECORDS
JOBNAME          : *                    LOG RECORD COUNT      :
CONN ID         :                      COLLECTION DURATION   :
CORR ID        :                      STATEMENT ELAPSED TIME:
OPER ID        :                      STATEMENT CPU TIME   :
AUTH ID        :                      GETPAGE REQUESTS    :
PLAN           : DB2TEST1              LOCK REQUESTS        :
PROGRAM        :
STATEMENT #     :
STATEMENT TYPE:
DYN SQL ONLY?  : N                    SQLCODE OPERATOR     :
                                           VALUE              :

EXCLUSION PROFILE? (Y/N): Y           SERVICEABILITY CODES : _____
                                           _____

HELP INFORMATION = PF1                applid                PF KEY ASSIGNMENTS = PA1
```

This example excludes from entering the collector all FETCH statements for plan DB2BPTST (where STATEMENT TYPE 4 is FETCH, as identified in DB2's DSNWMSGs field RDISTYPE):

```

JOBNAME: jobname          SQL ANALYZER COLLECTION PROFILE      DATE: MM/DD/YY
DB2      : db2            TIME: HH:MM:SS
COMMAND: _____
TMDB64011I - CAUTION: DIAG DUMPS SHOULD ONLY BE USED WITH ASG GUIDANCE
PROFILE DESCRIPTION : EXCLUDE FETCHES IN PROGRAM DB2BPTST
PROFILE NAME       : EXCLUDEP          EXCLUDE PACKAGE RECORDS? : N
PROFILE LAST UPDATED:                EXCLUDE SQL STATEMENT RECORDS?: Y

THREAD TARGETS FOR PKG/SQL RECORDS      THREAD LIMITS FOR SQL RECORDS
JOBNAME      : *                        LOG RECORD COUNT      :
CONN ID     :                          COLLECTION DURATION   :
CORR ID     :                          STATEMENT ELAPSED TIME:
OPER ID     :                          STATEMENT CPU TIME    :
AUTH ID     :                          GETPAGE REQUESTS     :
PLAN        : DB2BPTST                 LOCK REQUESTS        :
PROGRAM     :
STATEMENT #  :
STATEMENT TYPE: 4                      SQLCODE OPERATOR     :
DYN SQL ONLY? : N                      VALUE                :

EXCLUSION PROFILE? (Y/N): Y            SERVICEABILITY CODES : _____
                                         _____

HELP INFORMATION = PF1                  applid                          PF KEY ASSIGNMENTS = PA1

```

With the EXCLUDE PACKAGE RECORDS? field set to N in the *exclusion* profile, any otherwise matching *collection* profile that has LOG PACKAGE RECORDS? set to Y will log all packages in D7 records for plan DB2BPTST. Since the exclusion profile discards FETCH statements prior to collection, these package records will contain all performance metrics except those for the FETCH statements that were bypassed. (Keep this in mind when viewing these “incomplete” D7 package records.)

With EXCLUDE SQL STATEMENT RECORDS? set to Y, no FETCH statements (D6 records) will be logged from plan DB2BPTST, but other statements from plan DB2BPTST will be logged by any otherwise matching collection profile that has LOG SQL STATEMENT RECORDS? set to Y.

Since the intent of exclusion profiling is to allow you to reduce SQL Analyzer overhead by excluding from collection and logging a subset of SQL activity (e.g., large FETCH loops), your specified field combinations and values must make sense. If they do not, an error message will appear and EXCLUDE settings may be reset. For example, you cannot specify a STATEMENT TYPE or STATEMENT # for exclusion and then set EXCLUDE SQL STATEMENT RECORDS? to N. If you do, the EXCLUDE option will be reset to Y, and a message noting the inconsistency and correction will be displayed. Also, an exclusion profile cannot specify all threads. For example, you cannot specify an asterisk (\*) in the JOBNAME field and leave the other thread identification fields blank.

## TMON for DB2 Support for DB2 Version 8.1

TMON for DB2 4.0 provides toleration support for DB2 Version 8.1. This includes toleration support for variable-character string long identifiers and Unicode-formatted fields. In addition, support has been added that allows TMON for DB2 to recognize new input parameters for the DSNZPARM display.

### DB2 Version 8.1 Variable-character String Long Identifiers

The term *v-string* generally is used in the TMON for DB2 documentation to refer to the VARCHARs (variable-characters), long identifiers, and Unicode fields that are defined as 0CL130 (i.e., half word length plus CL128) in the DB2 IFCIDs. The general layout of the data displayed on TMON for DB2 screens remains the same as in earlier product releases, and the affected fields are truncated to the field lengths already defined in TMON for DB2 4.0 record layouts.

### Unicode-formatted Fields

TMON for DB2 4.0 logs all DB2 IFCID fields in EBCDIC format. The Unicode-formatted fields (UTF-8) available in DB2 8.1 are converted to EBCDIC format before they are moved to the TMON Log File Services data record and made available for reporting and online displays.

**Note:**

---

TMON for DB2 4.0 does not support the conversion of UTF-16 fields.

---

### DSNZPARM Requirements

New DSNZPARM options may be either tolerated or not supported at all. Some DSNZPARM settings change the behavior of DB2 and, by design, cannot be supported in any monitored environment. The DSNZPARM setting `UIFCIDS=YES` is not supported by TMON for DB2 for performance reasons (`UIFCIDS=NO` is the only supported value). IBM has indicated that DDF/RRSAF rollups are not supported by detail monitoring. See [“ACCUMACC=NO Recommendation and SQL Analyzer Rework” on page 2](#). Since DSNZPARMs are frequently added to DB2 between releases, newer DSNZPARM values may not be supported or tolerated.

### New Bind Jobs

In support of DB2 Version 8, these new plan and package bind jobs have been added to the TMON for DB2 installation library:

- TDBBIND8 contains JCL to bind TMON for DB2 plans to DB2 (Version 8 systems only).

- TDBBNP8 contains JCL to bind TMON for DB2 packages to DB2 (Version 8 systems only).

## Separation of the CANCEL and KILL Functions

The Current Thread Activity component (option 1) now provides two separate screens you can use to issue thread CANCEL and thread KILL requests. This section discusses the impact of this change on product screens, installation, security, and messages.

### Note:

Use the thread CANCEL and KILL functions with caution. Although they can be very helpful in certain situations, unpredictable results can occur depending on the type of activity the thread was performing and the stage at which it was executing at the time of the CANCEL or KILL request.

## Updated and New Screens

In support of the separation of the thread CANCEL and KILL functions, the Current Thread Cancel screen has been redesigned with new functionality and a new Current Thread \*KILL\* screen has been added to the product.

### Current Thread Cancel

JOBNAME: <i>jobname</i>	Current Thread Cancel	DATE: MM/DD/YY
DB2 : <i>db2</i>		TIME: HH:MM:SS
COMMAND: _____		CYCLE: MMSS
CONN ID : SPUFI	PLAN : DSNESPRR	CURRENT STATE: INAPP
CORR ID : DVRXW4	AUTH ID : DVRXW4	THREAD START : 10:24:38.3297
LOCATION: DVGRP1	SQLID : DVRXW123	CONN TYPE : TSO
RQST LOC:	LUID : USASG.RS91D8AH.BDFB1FC43DFB.0001	
PKG LOC : DVGRP1	CICS TKN:	
PKG NAME: DSNESPRR.DSNESM68.149EEA901A79FE48		
ACE TOKN: 1F26D198	CANCEL# : 90	
	CANCEL THIS THREAD? (Y/N): N	
	CANCEL WITH DUMP? (Y/N): N	
	CANCEL W/NOBACKOUT? (Y/N): N	
NOTES: CANCEL USES STANDARD DB2 CANCEL THREAD COMMANDS AND IS BOUND BY THOSE LIMITATIONS. ADDITIONAL ACTION MAY BE REQUIRED FOR DISTRIBUTED THREADS (SEE DB2 COMMAND REFERENCE FOR DETAILS). APPLICATIONS THAT USE THE DB2 CALL ATTACHMENT FACILITY MAY ENTER AN ENDLESS WAIT STATE. IF THIS OCCURS, APPLICATION JOBS CAN BE CANCELLED USING THE MVS CANCEL COMMAND.		
HELP INFORMATION = PF1	<i>applid</i>	PF KEY ASSIGNMENTS = PA1

### Note:

The access path to the Current Thread Cancel screen is unchanged.

The Current Thread Cancel screen allows you to confirm your decision to cancel a selected thread with a DB2 CANCEL THREAD or CANCEL DDF THREAD command. It also allows authorized users to request the cancel with or without the NOBACKOUT and DUMP options enabled.

You must have authority to cancel threads. This authorization is governed by three levels within the TMON for DB2 CANCEL secured function: READ, UPDATE, and CONTROL. (See [“Updated CANCEL Secured Function” on page 68](#) for new information related to the CANCEL secured function.) To cancel a DDF or local thread, you also must have authority to issue the DB2 cancel commands (i.e., CANCEL DDF THREAD and CANCEL THREAD), which is governed by the TMON for DB2 DB2CMD secured function.

If you perform a thread cancel using the CANCEL DDF THREAD or CANCEL THREAD command, authority to do so also is required within DB2. When the cancel is performed, a DB2 thread is opened to pass the command to DB2. The user ID that issues the command to DB2 depends on whether the DB2 authorization ID was modified (as discussed in Chapter 4 in *ASG-TMON for DB2 System Administrator’s Guide*). If it was not modified, the TMON for DB2 user ID is the same as the user ID of the process and it must have the proper authority within DB2 to issue the command.

Once the thread cancel is performed, a thread detail record is produced with a plan name of TMDB346. You can view this record in the Collection Analysis Thread Activity option (option 5.2) to determine the user ID of the process (displayed in the AUTHID field). If there was a DB2 authorization problem with the cancel, you can review the authorization ID within DB2 to determine the cause of the failure.

This table describes the new fields on the Current Thread Cancel screen. See Chapter 4 in *ASG-TMON for DB2 Reference Guide* for descriptions of other fields that appear on this screen.

Field	Description
ACE TOKN	Displays the ACE token of the thread. This information is taken from the DB2 IFCID148 record.
CANCEL#	Displays the numeric thread token to be used in the DB2 CANCEL THREAD command. This information is taken from the DB2 IFCID148 record.
CANCEL WITH DUMP?	Confirms that you want the cancel issued with the DUMP option. Valid values are N and Y.

---

Field	Description
CANCEL W/NOBACKOUT?	Confirms that you want the cancel issued with the NOBACKOUT option. Valid values are N and Y.
	<p><b>Note:</b> _____</p> <p>This field is displayed only if the NOBACKOUT monitor input parameter is enabled.</p> <p>_____</p>

### Current Thread \*KILL\*

```

JOBNAME: jobname                Current Thread *KILL*                DATE: MM/DD/YY
DB2      : db2                  TIME: HH:MM:SS
COMMAND: _____            CYCLE: MMSS

CONN ID : TSO                   PLAN      :                CURRENT STATE: IDENT
CORR ID : DVRXW4                AUTH ID   : DVRXW4         THREAD START : 00:00:00.0000
LOCATION: DVGRP1                  SQLID    : DVRXW4          CONN TYPE   : TSO
RQST LOC:                        LUWID    : USASG.RS91D8AH.BDFB1FC43DFB.0001
PKG LOC  :                       CICS TKN:
PKG NAME:
ACE TOKN: 1F26D198             CANCEL#  : 0

                                KILL THIS THREAD?   (Y/N): N
                                ABEND WITH DUMP?     (Y/N): N

NOTES: USING THE KILL FUNCTION WILL CAUSE THE THREAD TO ABEND S222.
        KILL SHOULD ONLY BE USED IN EXTRAORDINARY CIRCUMSTANCES
        AND ONLY WHEN NORMAL CANCEL FAILS OR CANCEL# IS UNAVAILABLE.
        USING THE KILL FUNCTION WHEN THE SYSTEM IS UNSTABLE CAN
        PRODUCE CATASTROPHIC RESULTS AS THE CALLRTM IS ISSUED BASED
        ON CONTROL BLOCK INFORMATION THAT MAY CHANGE OR BE CORRUPTED.

HELP INFORMATION = PF1          applid                PF KEY ASSIGNMENTS = PA1
    
```

The new Current Thread \*KILL\* screen allows you to confirm your decision to terminate a selected thread's task control block (TCB) using the MVS CALLRTM macro. Termination causes the TCB within the ASID (to which the DB2 ACE token points) to abend with a code of S222. You also have the option to request that a dump be produced when the thread abends.

**Note:** \_\_\_\_\_  
 You must attempt to issue a CANCEL command before issuing the KILL command.

#### *To access this screen*

- ▶ From the Thread Summary (Short), Thread Summary (Long), Thread Lock Contention, or Systemwide Lock Contention screen, enter the K line command next to a selected thread.

The majority of the fields on this screen are the same as the fields on the Current Thread Cancel screen and are described in Chapter 4 in *ASG-TMON for DB2 Reference Guide* and “[Current Thread Cancel](#)” on page 65. This table lists the fields that are unique to the Current Thread \*KILL\* screen:

Field	Description
ABEND WITH DUMP?	Confirms that you want the KILL command issued with the DUMP option to the CALLRTM macro. Valid values are N and Y.
KILL THIS THREAD?	Confirms that you want to terminate the thread’s TCB. Valid values are N and Y.

## New Installation Parameter

A TMON for DB2 startup parameter (documented in Appendix B in *ASG-TMON Products for OS/390 & z/OS Mainframe Installation Guide*) has been added in support of the separate CANCEL and KILL functions. The new parameter enables the NOBACKOUT option used in conjunction with the Current Thread Cancel screen (see “[Current Thread Cancel](#)” on page 65).

- Enter either NOBACKOUT or NOBACKOUT=YES to enable the option. Specifying NOBACKOUT (or NOBACKOUT=YES) allows you to issue the DB2 CANCEL command with the NOBACKOUT option. User authorization (granted through the TMON for DB2 CANCEL secured function) is required.
- Enter NOBACKOUT=NO or omit the parameter entirely to disable it. The option of cancelling a thread with NOBACKOUT does not appear on the Current Thread Cancel screen.

## Updated CANCEL Secured Function

The TMON for DB2 CANCEL secured function (documented in Chapter 5 in *ASG-TMON Products for OS/390 & z/OS Common Components Reference Guide*) now supports read, update, and control access.

- READ (or higher) allows you to issue the C line command or the DB2 CANCEL THREAD command.
- UPDATE (or higher) allows you to issue the C line command with the NOBACKOUT option (if it has been enabled via the input parameter).
- CONTROL (or higher) allows you to issue the K line command or KILL command to terminate thread TCBs.

## Changing the System Clock

The need to change the system clock typically occurs when daylight saving time either goes into effect or changes back to standard time, requiring that you adjust the clock either ahead or back. This section discusses data considerations for this adjustment and two methods for making the change.

### *To obtain more information about time change procedures for ASG-TMON Products*

- 1 Go to [www.asg.com](http://www.asg.com) or use <http://isp.asg.com>.
- 2 Sign onto the ISP from the Support pull-down menu.
- 3 Under Items of Special Interest at the bottom of the page, click Change to US Daylight Savings Time.

**Note:**

---

For information specific to TMON for DB2, click ASG-TMON for DB2 under this heading: ASG Products not impacted by the change to US Daylight Saving time - additional details are available.

---

## Data Considerations

Before changing the system clock, ensure that procedures are in place to maintain the integrity and usability of data logged by TMON for DB2. Be especially careful if you are setting the clock back; otherwise, you can receive data that is unreliable or potentially inaccessible online.

Data logged by TMON for DB2 is managed by time stamps, which are crucial to the system clock (STCK) fields used for keying data records. If records are logged with overlapping time stamps, some data in the TMON for DB2 collection files may be inaccessible to online displays. Setting the system clock back can cause new records to be logged with STCK times earlier than those for existing records, causing display of that time range to potentially return incomplete results.

Another adverse effect of setting the system clock back is the potential for TMON for DB2 to report twice on the same time period. If this occurs, you still will be able to access and report on the data, but reports for some time periods will combine data from two different intervals into one set of summary statistics.

## Instructions for Changing the System Clock

You can use two methods to change the system clock:

- Re-IPLing the system

- Using the z/OS SET CLOCK command to change the Greenwich mean time (GMT) adjustment factor

Each of these methods raises different considerations for TMON for DB2 data. ASG recommends, therefore, that you use the method and employ the TMON for DB2 procedures that best meet the needs of your system. The two methods also vary depending on whether you are setting the clock ahead (skipping a time period) or back (repeating a time period).

## Re-IPLing the System

- Using this method to set the system clock *ahead* does not affect TMON for DB2 data because no activity is caught in flight when you re-cycle z/OS.
- Using this method to set the system clock *back*, however, may cause TMON for DB2 to record two sets of data for the same range of STCK times.

### *To reset the system clock back using the safest procedure*

- 1 Bring down all TMON Log File Services (LFSs), TMON for DB2 Terminal Monitor Programs (TMPs), and DB2 regions that are being monitored by TMON for DB2.
- 2 Re-IPL the system, resetting the system clock back.
- 3 When the reset time is later than the last logged time before the reset, restart the TMON for DB2 TMP and LFS regions.

The disadvantage of this procedure is that it interrupts the use of your system for the amount of time you have set the system clock back (e.g., one hour in the United States). If an interruption of this length of time is unacceptable at your site, you can shorten it by replacing step 3 with this alternate 2-step procedure:

### *To shorten the length of system-interrupt time*

- 3 Dump the LFS files that contain data you want to process.
- 4 Reinitialize your LFS files to ensure data integrity in the collection files.

Dumping the LFS files saves the data for later reporting by TMON Report Writer. Reinitializing them ensures that only records logged after the time change will appear in the LFS files and on TMON for DB2 Collection Analysis screens.

**Note:**

Keep in mind that the data integrity you achieve by reinitializing the files applies only to the usability of the collection files. Your installation procedures still must take into consideration that reporting completed before the reset must be segregated from reporting completed after the reset to prevent duplicate reporting of the time interval during which the clock was reset. You also must either restart history (or monthly) accumulation or take into consideration reporting on records logged during the period of time overlap before and after the clock reset (see [“Data Considerations” on page 69](#)).

## Changing the GMT Adjustment Factor

You can use the SET CLOCK command to reset the system’s time-of-day clock dynamically without re-cycling z/OS. This method changes the interpretations of STCK times (the GMT offset), but does not change the actual STCK times used by the TMON for DB2 collection modules. You can implement this method at any time without an IPL.

- If you bring down the DB2 and TMON for DB2 regions while the clock reset is performed, the data considerations are the same as those for the re-IPL method.
- If you do not bring down any of the regions while the clock reset is performed, the data considerations are more complex.

To understand the effects of this method on TMON for DB2 data collection, you must first understand how it uses the system clock. The interpretation of the STCKs affects the collection of TMON for DB2 data in that all displayed STCK times for TMON for DB2 records are adjusted to reflect local time. This time zone adjustment is computed at startup, so TMON for DB2 must be re-cycled in order to change this adjustment value.

Setting the system clock ahead using the SET CLOCK command does not affect data collection; however, the GMT offset is not recomputed during TMON for DB2 execution. Until you force revision of the adjustment factor by cycling TMON for DB2, records continue to report the time of day as though no changes have occurred.

Setting the system clock back using the SET CLOCK command does not cause any problems for LFS since the STCK times that it uses remain consistent. The local times reported in the data records continue to represent unadjusted times until a revision of the time adjustment occurs. At that point, regardless of how the reset is accomplished, a possibility exists that TMON for DB2 will report on the same time periods twice. Data reports will intermingle separate activities that carry the same local times.

The safest approach to this method is to wait to restart monitoring until the new local time is later than the last local time before the change. This procedure avoids the difficult task of interpreting the data.

If you choose not to wait, however, be aware that data reporting for the period in which the reset is done could lead to confusion in interpretation of that data. This condition could persist as long as the TMP stays up and even longer when displaying data for the

period in which the reset was done. Also be aware that, if the adjustment factors for the TMON for DB2 and DB2 regions are reset at different times, the records they produce will not match. Exceptions generated by TMON for DB2 will show different times of day than the corresponding events in the DB2 region describing the same activity.