



DB2 for z/OS Version 10

Enhanced Monitoring Support in DB2 10 for z/OS

*Baltimore/Washington DB2 Users Group
December 8, 2010
Mark Rader – IBM Advanced Technical Skills*

Information Management software

Disclaimer

© Copyright IBM Corporation 2010. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, CICS, DB2, IMS, z/OS, Omegamon and Parallel Sysplex and are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.

Agenda

- Overview
- Enhancements for problem determination and performance monitoring
 - Monitoring information for static and dynamic SQL
 - Introduce STMT_ID
 - Modified and new trace records
- Enhancements for monitoring of system level activities
 - Connections
 - Threads
 - Idle thread timeout
- General changes to support these enhancements

Overview

- Current performance and exception monitoring is primarily at the thread or package level
 - SQL statement level monitoring and application level monitoring is limited
 - Difficult to understand impact of a specific SQL statement across the DB2 environment
 - Deadlock and timeout resolution and avoidance can be challenging
- For distributed access, controls are specified at the DB2 subsystem level
 - DSNZPARM is the primary mechanism; single set of values control all distributed access
 - MAXDBAT – maximum number of active DBATs
 - CONDBAT – maximum number of connections
 - IDTHTOIN – idle thread timeout
 - Low priority or poorly behaving applications can monopolize DB2 resources

DB2 10 for z/OS Enhanced Monitoring Support

- **Enhanced problem determination and performance monitoring:**
 - Enable rapid problem diagnosis and resolution
 - Messages and traces are captured with statement level information for both dynamic and static statements.
 - Complete statistics performance metrics at statement level for both dynamic and static statements
 - Support performance monitoring tools
 - Provide system-wide statement level detail information in real time for tooling
- **Increased granularity of monitoring for system level activities**
 - Number of connections
 - Number of threads
 - Idle thread timeout
- **Note: both sets of enhancements require DB2 10 for z/OS NFM**

Enhanced problem determination and performance monitoring support

Problem determination and performance monitoring

- Statement identifier support
- Trace enhancements
- Message enhancements
- Monitoring enhancements through DRDA

Statement identifier support

- **STMT_ID** is defined at DB2 for z/OS server
 - Available for local and remote applications, static and dynamic SQL
 - Returned to DRDA application requester
 - Captured in IFCIDs for both static and dynamic SQL
 - Dynamic SQL: when dynamic statement cache is enabled
 - Unique STMT_ID is only generated for a dynamic SQL statement that qualifies for insertion into dynamic statement cache.
 - Static SQL: STMT_ID column in SYSIBM.SYSPACKSTMT
 - Pre-V10 packages must be rebound in V10 NFM
- **DRDA returns STMT_ID and**
 - 2-byte header which includes statement type (static or dynamic)
 - 10-byte compilation time: last bind time for static; compilation time for dynamic

Instrumentation

- DB2 9 for z/OS: Real-time statement level detail on per thread basis via IFCID 124, not including performance metrics
- DB2 10 for z/OS: Support for real-time statement level detail, including performance metrics
- Monitor applications, such as OMPE, can monitor in real-time:
 - All dynamic statements in dynamic statement cache
 - All static statements active in EDM pool
 - System-wide information rather than thread level granularity
- New Monitor Class 29 for system-wide statement level detail
- Changes to existing IFCIDs

Monitor Class 29

- New monitor class for system-wide statement level detail
- Starting Monitor Class 29 activates the following IFCIDs:
 - IFCID 318 – switch to produce greater detail about dynamic statements in the cache
 - Accounting or monitor class 3 must be enabled
 - Statement level detail can be externalized via READS for IFCID 316
 - IFCID 316 – enhanced for event-based trigger
 - When statement evicted from dynamic statement cache, IFCID 316 written for single statement and final cached statement level statistics
 - IFCID 400 – new IFCID that mirrors IFCID 318, but for static statements
 - Statement level detail for static statements via READS for IFCID 401
 - IFCID 401 – new IFCID that has IFI READS capability and event trigger
 - Externalizes statement level details for static statements in EDM pool
 - When statement evicted from EDM pool, IFCID will be written for single statement and final cached statement level statistics

Performance trace record changes

- IFCIDs enhanced to include statement type and STMT_ID:
 - IFCID 58 (End SQL); also returns statement level performance metrics
 - IFCID 63 and 350 (SQL Statement); also return original source CCSID of the SQL statement
 - IFCID 124 (SQL Statement Record)
- IFCID 65 (Open Cursor) enhanced to trace cursor attribute on implicit commit request.
- IFCID 66 (Close Cursor) enhanced to trace implicit close.
 - New field indicates if close statement is implicit or explicit

Deadlock, timeout and lock escalation traces

- **IFCID 172 and IFCID 196**
 - Include statement type and STMT_ID for both dynamic and static statements when deadlock or timeout condition detected
- **IFCID 337**
 - Include statement type and STMT_ID for dynamic and static statements when lock escalation detected

Messages and codes

- THREAD-INFO token added to DSNL030I, DSNV436I
- THREAD-INFO token updated, for messages that already include it, to contain:
 - Statement type
 - STMT_ID
 - ROLE
- New messages
 - DSNT771I
 - DSNT772I
- New reason codes
- SQLCODE modified: -20142
- New resource type: 4000, profile exception threshold exceeded

Changes to EXPLAIN and Profile tables

- **DSN_STATEMENT_CACHE_TABLE** columns added to support 64 bit integers
 - STAT_EXECB BIGINT NOT NULL WITH DEFAULT,
 - STAT_INDXB BIGINT NOT NULL WITH DEFAULT,
 - STAT_RSCNB BIGINT NOT NULL WITH DEFAULT,
 - STAT_PGRPB BIGINT NOT NULL WITH DEFAULT,
 - STAT_RIDLIMTB BIGINT NOT NULL WITH DEFAULT,
 - STAT_RIDSTORB BIGINT NOT NULL WITH DEFAULT
- **DSN_STATEMENT_RUNTIME_INFO**
 - Same columns added

Example for deadlock message

- DSNT375I

DSNT375I >

PLAN=DSNTEP3 WITH CORRELATION-ID=RUNTEP1B

CONNECTION-ID=BATCH

LUW-ID=USIBMSY.SYEC1DB2.C6C38C788498=7

THREAD-INFO=ADMF001:*:*:*:DYNAMIC:7:AVERYLONGROLENAMEROLE1

IS DEADLOCKED WITH PLAN=DSNTEP3 WITH

CORRELATION-ID=RUNTEP1A

CONNECTION-ID=BATCH

LUW-ID=USIBMSY.SYEC1DB2.C6C38C76D2CD=5

THREAD-INFO=ADMF001:*:*:*:DYNAMIC:5:*

ON MEMBER VA1B

- For THREAD 1

- Dynamic statement with statement ID of 7

- For THREAD 2

- Dynamic statement with statement ID of 5

Deadlock message, continued

- Since both of the statements are dynamic statements, you can query against SYSIBM.DSN_STATEMENT_CACHE_TABLE using the provided statement ID in the message to find out more information about the involved statements.
- Before query against SYSIBM.DSN_STATEMENT_CACHE_TABLE
 - Execute EXPLAIN STMTCACHE ALL command to have all of the cached statements in the dynamic statement cache externalized to the SYSIBM.DSN_STATEMENT_CACHE_TABLE
 - Each row in the table contains information about each statement in the cache, as well as the statistics that reflect the execution of the statement
 - A sample query
 - SELECT PROGRAM_NAME, COLLID, STMT_TEXT FROM SYSIBM.DSN_STATEMENT_CACHE_TABLE WHERE STMT_ID=7;
- If statements are static, query against SYSIBM.SYSPACKSTMT

System Level Monitoring Enhancements

Enhancements for monitoring system level activities

- DB2 9 for z/OS profile table facility enhanced to provide improved threshold monitoring and filtering
 - New fields and keywords
- Monitor profile is extended to support monitoring of
 - Idle thread timeout
 - Number of threads
 - Number of connections
- Monitor profile can filter by ROLE and client product ID
 - For monitoring system level activities
- Purpose is to provide DBAs with greater flexibility and control in allocating distributed resources to particular clients, applications or users

Profile table facility – system level monitoring, DB2 10

- DB2 10 for z/OS adds support for filtering and threshold monitoring of system related activities
 - Number of threads
 - Number of connections
 - Idle thread timeout
- New scope filters
 - ROLE (available through Trusted Context support)
 - Product-specific identifier
- New keywords for threads, connections, idle thread timeout
- Note: these enhancements to not apply to statement level monitoring and tuning, only to system level monitoring and tuning for threads and connections

Managing distributed threads

- Three DSNZPARM parameters
 - MAXDBAT
 - Maximum number of concurrently active DBATs in the subsystem
 - DB2 10 for z/OS supports defining up to 19,999
 - CONDBAT
 - Maximum number of concurrent inbound connections in the subsystem
 - DB2 10 for z/OS supports defining up to 150,000
 - IDTHTOIN
 - Specifies the amount of time, in seconds, that an active server thread can remain idle
 - Single value across the subsystem
- Subsystem level values do not reflect varying priorities or requirements for different users or applications.

System level monitoring for threads and connections

- Define the following profile tables and indexes (if they do not already exist)
- To monitor system level activities:
 - 1. Create a profile to specify what activities to monitor
 - Four exclusive filtering categories
 - 2. Define the type of monitoring
 - Three keywords, two attribute columns
 - 3. Load or reload profile tables
 - START PROFILE
 - 4. Optional: Stop Monitoring

Define profile tables and indexes

- Create four profile tables and necessary indexes provided in the DSNTIJSJG job in SDSNSAMP
 - SYSIBM.DSN_PROFILE_TABLE
 - SYSIBM.DSN_PROFILE_HISTORY
 - SYSIBM.DSN_PROFILE_ATTRIBUTES
 - SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY
 - SYSIBM.DSN_PROFILE_TABLE_IX_ALL
 - SYSIBM.DSN_PROFILE_TABLE_IX2_ALL
 - SYSIBM.DSN_PROFILE_ATTRIBUTES_IX_ALL
- [Sample DDL in Appendix](#)

1. Create a profile

- Insert profile row(s) to SYSIBM.DSN_PROFILE_TABLE
 - New columns, ROLE and PRDID, provide additional filtering
 - ROLE filters by the role assigned to the user associated with the thread
 - PRDID filters by product-specific id, e.g. 'JCC03570'
- Four filtering categories are supported:
 - IP address or DNS name (IPADDR column)
 - Product Identifier (PRDID column)
 - Role and/or Authorization Identifier (ROLE, AUTHID columns)
 - Collection ID and/or Package Name (COLLID, PKGNAME columns)
- Filtering criteria from different categories cannot be specified on the same profile row.
 - E.g. IPADDR and PRDID cannot be specified on the same row. One must be null.

1. Create a profile, cont.

- If a category has more than one filtering scope, either can exist without the other. For example,
 - ROLE or AUTHID can be the only filtering column specified in a profile row, or they can both be specified.
 - COLLID or PKGNAME can appear in the same row, or either can be specified alone.
- These rules on filtering categories only apply to system level monitoring functions/keywords for threads and connections.
- ROLE and PRDID filtering only apply to system level monitoring functions/keywords for threads and connections.
- GROUP MEMBER applies to data sharing, allowing different values for different members:
 - If this column is blank, the profile row can apply to any member in the data sharing group.
 - There is only one SYSIBM.DSN_PROFILE table, which all members share, but this column allows a profile to be applied to only one member.

1. Create a profile: 8 filtering modes

- Four separate filtering categories support eight filtering modes
 - A. Client IP address
 - 1. IPADDR column
 - Domain name format, dotted-decimal (IPV4) format or colon-hex (IPV6) format
 - B. Client Product ID
 - 2. PRDID column
 - C. Role name, Authorization ID
 - 3. Both ROLE and AUTHID columns
 - 4. ROLE column
 - 5. AUTHID column
 - D. Collection ID, Package name
 - 6. Both COLLID and PKGNAME columns
 - 7. COLLID column
 - 8. PKGNAME column

Unqualified profiles in *SYSIBM.DSN_PROFILE_TABLE*

- Profiles 11 and 15 are invalid – different filtering categories in same row

ROLE	AUTHID	IPADDR	PRDID	COLLID	PKGNAME	PROFILEID	PROFILE_ENABLED
DB2DEV	Tom					10	Y
DB2DEV			SQL09071			11	Y
	Tom					12	Y
			JCC03570			13	Y
		TEST.SVL. IBM.COM				14	Y
		TEST.SVL. IBM.com		COLL1		15	Y

- SYSIBM.DSN_PROFILE_HISTORY* for rows 11, 15 include the text 'REJECTED' and the reason for rejection in the STATUS column

Validating profile rows

- SYSIBM.DSN_PROFILE_HISTORY table

PROFILEID	:	STATUS (DSN_PROFILE_HISTORY)
10	:	ACCEPTED
11	:	REJECTED - INVALID SCOPE SPECIFIED. FOR SYSTEM LEVEL MONITORING, ONLY IP ADDR, PRDID, ROLE AND/OR AUTHID, COLLECTION ID AND/OR PACKAGE NAME CAN BE SPECIFIED
12	:	ACCEPTED
13	:	ACCEPTED
14	:	ACCEPTED - DOMAIN NAME IS RESOLVED INTO IP ADDRESS ::FFFF:9.1.2.3
15	:	REJECTED - INVALID SCOPE SPECIFIED. FOR SYSTEM LEVEL MONITORING, ONLY IP ADDR, PRDID, ROLE AND/OR AUTHID, COLLECTION ID AND/OR PACKAGE NAME CAN BE SPECIFIED

Multiple profiles

- Multiple qualified filtering profiles from different categories can apply if the filtering criteria match the thread or connection.
 - In previous example, profile 10 and 14 might both apply to Tom's thread or connection.
- Within the same filtering category, only one qualified profile is applied.
 - Most specific profile is chosen
 - ROLE has precedence over AUTHID
 - COLLID has precedence over PKGNAME
 - In previous example, if profile 10 and 12 both apply to Tom's thread or connection for the same keyword, row 10 will apply, as ROLE takes precedence.

Multiple profiles example

- Multiple profiles that filter on different categories can apply ★
- Multiple profiles that filter on one category – most specific applies ★
 - If Tom uses ROLE DB2DEV, PROFILEID 10 applies

ROLE	AUTHID	IPADDR	PRDID	COLLID	PKGNAME	PROFILEID	PROFILE_ENABLED
DB2DEV ★	Tom					10 ★ ★	Y
DB2DEV			SQL09071			11	Y
	Tom					12 ★	Y
			JCC03570			13	Y
		TEST.SVL. IBM.COM				14 ★	Y
		TEST.SVL. IBM.COM		COLL1		15	Y

2. Define type of monitoring

- **Insert row(s) into SYSIBM.DSN_PROFILE_ATTRIBUTES**
 - PROFILEID – specify the profile that defines the activities to monitor
 - KEYWORDS – specify one of the following:
 - MONITOR THREADS – cannot exceed MAXDBAT
 - MONITOR CONNECTIONS (IPADDR filtering only) – active plus inactive
 - Cannot exceed CONDBAT
 - MONITOR IDLE THREADS – applied independently of IDTHTOIN
 - ATTRIBUTE1 – specify WARNING or EXCEPTION and specify the messaging level
 - DIAGLEVEL1 – DSNT771I message with minimal information
 - DIAGLEVEL2 – DSNT772I message with detailed information
 - ATTRIBUTE2 – specify threshold value (zero or positive value)
 - Number of active threads or total connections
 - Seconds for idle threads
 - ATTRIBUTE3 – leave blank
 - Values in KEYWORDS and ATTRIBUTE1 columns must be all upper case

WARNING or EXCEPTION behavior

- **WARNING** – processing continues
 - DSNT771I or DSNT772I issued, depending on DIAGLEVEL
- **EXCEPTION** – issue DSNT771I or DSNT772I and
 - MONITOR THREADS – queued or suspended
 - If IPADDR filtering: request is suspended until the next thread is available
 - If PROPID, ROLE and/or AUTHID, COLLID and/or PKGNAME filtering: the thread is suspended until the next thread (DBAT) is available. The number of threads suspended can not exceed the profile threshold value. If suspending a thread would cause the number of suspensions to exceed the profile threshold, then DB2 fails the connection request and returns SQLCODE -30041
 - MONITOR CONNECTIONS – (IPADDR filtering only)
 - New connection requests rejected
 - MONITOR IDLE TREADS – thread terminated

DIAGLEVEL1 or DIAGLEVEL2

- **DIAGLEVEL1** – DSNT771I message at most once every 5 minutes for any profile threshold exceeded.
 - DSNT771I csect-name A MONITOR PROFILE **condition-type** CONDITION OCCURRED numberof-time TIME(S).
 - Minimum console activity and limited message information
 - Refer to statistics trace records for accumulated occurrences of specific profile threshold exceeded under a specific profile ID.
- **DIAGLEVEL2** – DSNT772I message at most once every 5 minutes for each unique occurrence of the profile threshold exceeded in a specific profile ID
 - DSNT772I csect-name A MONITOR PROFILE **condition-type** CONDITION OCCURRED numberof-time TIME(S) IN PROFILE ID=**profile-id** WITH PROFILE FILTERING SCOPE=**filtering-scope** WITH REASON=**reason**
- **Statistics class 4 IFCID 402 record written at statistics interval**

System level monitoring - Example

- Sample rows from SYSIBM.DSN_PROFILE_ATTRIBUTES table.
 - 'REMARKS' column not pictured

ProfileID	Keywords	Attribute1	Attribute2	Attribute3	Attribute Timestamp
1	MONITOR THREADS	EXCEPTION_ DIAGLEVEL2	10	10 active threads Queue 11-20 *Reject 21st	2009-12-19..
2	MONITOR CONNEC- TIONS	WARNING	50		2009-12-19..
3	MONITOR IDLE THREADS	EXCEPTION_ DIAGLEVEL1	300		2009-12-21..

* If monitoring for this PROFILEID is based on IPADDR, then do not reject, but queue until reach CONDBAT

3. Load or reload profile tables

- Issue START PROFILE command
 - Loads or reloads SYSIBM.DSN_PROFILE_TABLE into memory
- Any valid rows with 'Y' in the PROFILE_ENABLED column are in effect.
- Once START PROFILE is issued, DB2 starts to monitor the next thread or connection that matches profiles in effect.
- START PROFILE command is member-specific

4. *Stop monitoring*

- To disable monitoring for a specific profile:
 - DELETE that profile's row from SYSIBM.DSN_PROFILE_TABLE
OR
 - Update PROFILE_ENABLED column to 'N' for that profile
 - Then issue START PROFILE
- To disable all monitoring for both system related and statement related monitoring that has been specified in the profile tables
 - Issue STOP PROFILE

DB2 Commands





- START PROFILE
- STOP PROFILE
- One of the following authorities required:
 - SYSOPR
 - SYSCTRL
 - SYSADM
- Issued from:
 - z/OS console
 - Batch job

Multiple profile example

SYSIBM.DSN_PROFILE_TABLE

ROLE	AUTHID	IPADDR	PRDID	COLLID	PKGNAME	PROFILEID	PROFILE_ENABLED
	Tom					20	Y
			JCC03570			21	Y
		TEST.SVL.IBM.COM				22	Y

SYSIBM.DSN_PROFILE_ATTRIBUTES

ProfileID	Keywords	Attribute1	Attribute2	Attribute3	Attribute Timestamp
20	MONITOR THREADS	EXCEPTION	10 		2009-12-19...
21	MONITOR IDLE THREADS	WARNING	180 		2009-12-19...
22	MONITOR CONNECTIONS	EXCEPTION	45 		2009-12-19...
22	MONITOR THREADS	EXCEPTION	20  *		2009-12-21...

* If 22 MONITOR CONNECTIONS row deleted, then queue requests over 21 up to CONDBAT

Multiple profile example – details #1

■ **SYSIBM.DSN_PROFILE_TABLE**

- PROFILEID 20 row defines a profile that matches all the threads that are run by user Tom.
- PROFILEID 21 row defines a profile that matches all the threads that are using client driver level JCC03570.
- PROFILEID 22 row defines a profile that matches all the threads and connections that are connecting from IP Address TEST.SVL.IBM.COM

■ **SYSIBM.DSN_PROFILE_ATTRIBUTES**





- All threads that are executed under user Tom connecting from IP address TEST.SVL.IBM.COM using client driver level JCC03570 are monitored by the profiles identified by PROFILEID 20, 21, and 22.
- PROFILEID 20, 21, and 22 are all active and qualified profiles from different filtering categories that may match the current thread execution environment, therefore all three profiles are applied.

Multiple profile example – details #2

- **What are the results?**
 - If Tom attempts to execute more than 10 concurrent threads, DB2 suspends up to 10 more requests based on PROFILEID 20 with EXCEPTION
 - NOT by PROFILEID 22 because PROFILEID 20 has a lower threshold value in the MONITOR THREADS keyword.
 - The 21st concurrent connection from AUTHID “Tom” receives -30041.
 - If a thread that stays idle for more than 3 minutes, DB2 issues a DSNT771I console message and lets the thread continue to stay idle based on PROFILEID 21 with a WARNING action defined.
 - All connections from IP address TEST.SVL.IBM.COM are monitored by the profiles identified by PROFILEID 22 as the connection is subject to the filtering on IP Address only.
 - When the number of connections exceeds 45, DB2 rejects any subsequent connection request, since PROFILEID 22 has an EXCEPTION action defined.
 - The lower green star signifies that the total number of threads from this IPADDR are limited to 20, plus 25 connections (see 45 in row immediately above); the 46th concurrent connection from this IPADDR is rejected. Since Tom is limited to 10 active threads, what is the meaning of PROFILEID 22 with MONITOR THREADS keyword? Other users, other than Tom, can have active threads, up to a total of 20. So, if Tom already has 10 active threads (limited by PROFILEID 20), Lisa can have another 10 active threads from the same IP address. Between Tom and Lisa (and any others from this IP address) there can be 20 active threads and 25 suspended requests (inactive connections).
 - Note that if the third row of the attribute table (PROFILEID 22, MONITOR CONNECTIONS) were not present, then the fourth row (PROFILEID 22, MONITOR THREADS) allows inactive connections up to the number in CONDBAT, because the filtering category is IPADDR.

Multiple profile example – DB2 Data Sharing

ROLE	AUTHID	IPADDR	PRDID	GROUP MEMBER	PROFILEID	PROFILE_ENABLED
	Tom				20	Y
			JCC03570		21	Y
		TEST.SVL.IBM.COM		DB2A	22	Y
		TEST.SVL.IBM.COM		DB2B	23	Y

ProfileID	Keywords	Attribute1	Attribute2	Attribute3	Attribute Timestamp
20	MONITOR THREADS	EXCEPTION	10 		2009-12-19...
21	MONITOR IDLE THREADS	WARNING	180 		2009-12-19...
22	MONITOR CONNECTIONS	EXCEPTION	45 		2009-12-19...
23	MONITOR THREADS	EXCEPTION	20 	Queue up to CONDBAT	2009-12-21...

Data Sharing example - details

- The result for the data sharing example are:
 - PROFILEID 22 only applies to DB2A
 - All connections from IP address TEST.SVL.IBM.COM to member DB2A are monitored by PROFILEID 22 (MONITOR CONNECTIONS is subject to filtering on IP Address only.)
 - When the number of connections to DB2A exceeds 45, DB2A rejects any subsequent connection request, based on 'EXCEPTION' in Attribute1 in PROFILEID 22.
 - PROFILEID 23 only applies to DB2B.
 - DB2B can have 20 active threads from IPADDR TEST.SVL.IBM.COM.
 - Additional thread requests are queued until one of the 20 active DBATS ends and a DBAT is available.
 - The number of requests that are queued is limited only by CONDBAT, since the filtering category on PROFILEID 23 is IPADDR.
 - From the IP address TEST.SVL.IBM.COM,
 - DB2A: Tom can have 10 active threads and 10 suspended threads
 - Tom and Lisa (and/or other AUTHIDs) can combine to have 20 active threads and 25 inactive connections from this IP address on DB2A.
 - DB2B: Tom can have another 10 active threads and 10 suspended threads.
 - Tom and Lisa (and/or other AUTHIDs) can combine to have 20 active threads on DB2B.
 - On DB2B the combined inactive connections from this IP address is limited by CONDBAT.

Benefit

- **Monitoring system level activities**
 - Provides granularity of control for warning and exception handling based on filtering capability for:
 - Idle threads
 - Number of connections
 - Number of threads
- **Now thread and connection resources can be monitored and controlled by filters to correspond to business priorities.**

Questions?

- If you think of a question later, you can e-mail me:
 - mrader@us.ibm.com

Appendix

SYSIBM.DSN_PROFILE_TABLE

```
CREATE TABLE SYSIBM.DSN_PROFILE_TABLE
  ("AUTHID"          VARCHAR(128)
  ,"PLANNAME"       VARCHAR(24)
  ,"COLLID"         VARCHAR(128)
  ,"PKGNAME"       VARCHAR(128)
  ,"IPADDR"        VARCHAR(254)
  ,"PROFILEID"     INTEGER      NOT NULL PRIMARY KEY
                        GENERATED BY DEFAULT AS IDENTITY
  ,"PROFILE_TIMESTAMP"  TIMESTAMP NOT NULL WITH DEFAULT
  ,"PROFILE_ENABLED"  CHAR(1)    NOT NULL DEFAULT 'Y'
  ,"GROUP_MEMBER"    VARCHAR(24)
  ,"REMARKS"         VARCHAR(762)
  ,"ROLE"            VARCHAR(128)
  ,"PRDID"           CHAR(8)
  )
IN DSNOPTDB.DSNOPTTS
CCSID UNICODE;
```

SYSIBM.DSN_PROFILE_HISTORY

```
CREATE TABLE SYSIBM.DSN_PROFILE_HISTORY
  ("AUTHID"          VARCHAR(128)
  ,"PLANNAME"       VARCHAR(24)
  ,"COLLID"         VARCHAR(128)
  ,"PKGNAME"        VARCHAR(128)
  ,"IPADDR"         VARCHAR(254)
  ,"PROFILEID"      INTEGER      NOT NULL
  ,"PROFILE_TIMESTAMP"  TIMESTAMP  NOT NULL
  ,"PROFILE_ENABLED" CHAR(1)     NOT NULL
  ,"GROUP_MEMBER"   VARCHAR(24)
  ,"STATUS"         VARCHAR(254)
  ,"ROLE"           VARCHAR(128)
  ,"PRDID"          CHAR(8)
  )
IN DSNOPTDB.DSNOPTTS
CCSID UNICODE;
```

SYSIBM.DSN_PROFILE_ATTRIBUTES

```
CREATE TABLE SYSIBM.DSN_PROFILE_ATTRIBUTES
  ("PROFILEID"          INTEGER          NOT NULL
   REFERENCES SYSIBM.DSN_PROFILE_TABLE
   ON DELETE CASCADE
  ,"KEYWORDS"          VARCHAR(128)     NOT NULL
  ,"ATTRIBUTE1"        VARCHAR(1024)
  ,"ATTRIBUTE2"        INTEGER
  ,"ATTRIBUTE3"        FLOAT
  ,"ATTRIBUTE_TIMESTAMP"  TIMESTAMP      NOT NULL WITH DEFAULT
  ,"REMARKS"           VARCHAR(762)
  )
IN DSNOPTDB.DSNOPTTS
CCSID UNICODE;
```

SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY

```
CREATE TABLE SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY
( "PROFILEID"          INTEGER          NOT NULL
, "KEYWORDS"          VARCHAR(128)      NOT NULL
, "ATTRIBUTE1"        VARCHAR(1024)
, "ATTRIBUTE2"        INTEGER
, "ATTRIBUTE3"        FLOAT
, "ATTRIBUTE_TIMESTAMP"  TIMESTAMP          NOT NULL
, "STATUS"            VARCHAR(254)
)
IN DSNOPTDB.DSNOPTTS
CCSID UNICODE;
```

SYSIBM.DSN_PROFILE_TABLE Indexes

```
CREATE UNIQUE INDEX SYSIBM.DSN_PROFILE_TABLE_IX_ALL
  ON SYSIBM.DSN_PROFILE_TABLE
  ( "PROFILEID"
  );
```

```
CREATE INDEX SYSIBM.DSN_PROFILE_TABLE_IX2_ALL
  ON SYSIBM.DSN_PROFILE_TABLE
  ( "PROFILE_ENABLED"
  , "AUTHID"
  , "PLANNAME"
  , "COLLID"
  , "PKGNAME"
  , "IPADDR"
  , "PRDID"
  , "ROLE"
  , "GROUP_MEMBER"
  , "PROFILE_TIMESTAMP" DESC
  );
```

SYSIBM.DSN_PROFILE_ATTRIBUTES Index

```
CREATE UNIQUE INDEX SYSIBM.DSN_PROFILE_ATTRIBUTES_IX_ALL
ON SYSIBM.DSN_PROFILE_ATTRIBUTES
( "PROFILEID"
, "ATTRIBUTE_TIMESTAMP" DESC
, "KEYWORDS"
, "ATTRIBUTE1"
, "ATTRIBUTE2"
, "ATTRIBUTE3"
)
USING STOGROUP DSNOPTSG;
```