IBM DB2 Query Monitor for z/OS V3.2 Tech Doc Update

The following update applies to IBM DB2 Query Monitor for z/OS V3.2 User's Guide (SC19-4143-01)

Contents

Terms and Abbreviations

The following product names, terms, and abbreviations are used throughout this document.

Products that use the IBM DB2 Data Access Common Collector

The following products use the IBM DB2 Data Access Common Collector (CQC) Version 1.1:

- **IBM DB2 Query Monitor for z/OS** (CQM) Version 3.2
- **InfoSphere Guardium S-TAP for DB2 on z/OS** (ADH) Version 9.1
- **InfoSphere Optim Workload Replay S-TAP on DB2 on z/OS** (CQR) Version 2.1

Products that do not use the IBM DB2 Data Access Common Collector

The following products **do not** use the IBM DB2 Data Access Common Collector (CQC) Version 1.1:

- **IBM DB2 Query Monitor for z/OS** (CQM) Version 3.1
- **InfoSphere Guardium S-TAP for DB2 on z/OS** (ADH) Version 9.0
- **InfoSphere Guardium S-TAP for DB2 on z/OS** (ADH) Version 8.1
- **InfoSphere Optim Query Capture and Replay on DB2 on z/OS** (CQR) Version 1.1

Cross-Product Components

- **Support Services Address Space** – Also referred to as the Master Address Space. Specifying the same MASTER_PROCNAME value for multiple products causes those products to share that Support Services Address Space.
- **Shared Memory Objects** – The memory objects owned by the Support Services Address Space.
- **IBM DB2 Data Access Common Collector for z/OS** (CQC) Version 1.1

Product-Specific Started Tasks

- **DB2 Query Monitor Subsystem** – The started task used by IBM DB2 Query Monitor for z/OS. The DB2 Query Monitor Subsystem integrates the data collected by the CQC to provide DB2 Query Monitor users with a complete picture of query activity in a monitored system.
- **ADH Agent** – The started task used by InfoSphere Guardium S-TAP for DB2 on z/OS.
- **CQR Agent** – The started task used by InfoSphere Optim Workload Replay S-TAP for DB2 on z/OS. The CQR Agent collects workload SQL statement data in an InfoSphere Optim Workload Replay S-TAP for DB2 on z/OS environment. The CQR Agent filters and sends workload SQL data to a Guardium Appliance with installed support for InfoSphere Optim Workload Replay S-TAP for DB2 on z/OS. One CQR Agent is required for each DB2 that you intend to audit.

Product-Specific Monitoring Agents

- **DB2 Query Monitor Monitoring Agent** – The DB2 Query Monitor monitoring agent is the interface that DB2 Query Monitor installs within a DB2 subsystem to capture SQL performance data. When a Query Monitor subsystem collects data about a DB2 subsystem, a monitoring agent is at work collecting data about that DB2 subsystem.

Other Components

- **Guardium Appliance** - The Guardium Appliance (and its user interface) is an IBM appliance that is required by the following products that use CQC:
    - InfoSphere Guardium S-TAP for DB2 on z/OS (ADH) Version 9.1
    - InfoSphere Optim Workload Replay S-TAP on DB2 on z/OS (CQR) Version 2.1

    And by the following products that do not use CQC:

    - InfoSphere Guardium S-TAP for DB2 on z/OS (ADH) Version 9.0
    - InfoSphere Optim Query Capture and Replay on DB2 on z/OS (CQR) Version 1.1

  All of the products listed above plug into the Guardium Appliance. The user configures the products using the Guardium Appliance user interface. Then, data from the products is streamed to the appliance for the user to view, filter, store, etc.

Product-Specific Terms

DB2 Query Monitor (CQM) Components and Important Terminology

- **Consolidation and Analysis Engine** (CAE) – DB2 Query Monitor's Consolidation and Analysis Engine (CAE) consists of three components: the CAE Agent, the CAE Server, and the CAE Web Client. Together, these components provide enterprise-wide data consolidation, autonomic root cause analysis, and corrective actions.
- **CAE Agent** – The CAE Agent provides TCP/IP access to all of the DB2 Query Monitor Subsystems and monitored DB2 subsystems on the local z/OS image.
- **CAE Server** – The CAE Server consolidates data from one or multiple CAE Agents and performs additional analysis so that data can be presented by the CAE Web Client. It includes an alert system and web server.
- **CAE Web Client** – The CAE Web Client enables you to view data and exceptions for a single DB2 subsystem or more DB2 subsystems, regardless of z/OS and Sysplex boundaries. The CAE Web Client also provides you with powerful filtering and browsing capabilities for both data and alerts.
- **Performance History Database** – (Formerly referred to as “offload tables”) The DB2 Query Monitor Performance History Database is a set of DB2 tables to which you can offload data from DB2 Query Monitor’s Performance History Files. The uniqueness of DB2 Query Monitor data across z/OS systems, DB2 subsystems, and DB2 Query Monitor Subsystems is maintained when you offload data from DB2 Query Monitor’s Performance History Files to the DB2 Query Monitor Performance History Database.
- **Performance History Files** – (Formerly referred to as “backstore data sets”) Performance History Files are VSAM data sets that hold information collected by DB2 Query Monitor's collection points (SQL metrics, DB2 object access, SQL text, DB2 commands, negative SQLCODES). One Performance History File is created for each collection point (with the exception of the host variables collection point) on a per-interval basis. Additionally, information about the exceptions and notifications DB2 Query Monitor recognizes and sends are also written to Performance History Files (also on a per interval basis).

- **Interval** – An interval is a unit into which DB2 Query Monitor divides and stores data. Intervals have a start and end time as well as an interval number and other information that identifies the data for that interval.
- **Monitoring Profile** – Monitoring Profiles enable you to tailor how DB2 Query Monitor monitors specific SQL workloads. Monitoring Profiles control things such as summary reporting, negative SQLCODE reporting, exception limits, alert notifications thresholds, collection of host variable information, and OPTKEYS override settings. Profiles do not affect DB2 command reporting.
- **Monitoring Profile Line** – DB2 Query Monitor's profiles consist of one or more monitoring profile lines that can be created, inserted, updated, ordered, and deleted as needed to tailor a monitoring profile to fit your needs. Each monitoring profile line applies to one workload. Each monitoring profile line consists of the following elements:
    - Line type (Include or Exclude)
    - Miscellaneous flags
    - Workload definition
    - Exception thresholds
    - Exception limit
    - SQL codes excluded from exceptions
    - Alert thresholds
    - SQLCODEs excluded from alerts
    - SQLCODEs excluded from summary collection
    - OPTKEYS overrides
- **SQL Workload** - An SQL Workload provides a way of identifying a group of applications to DB2 Query Monitor so that performance data can be collected for SQL statements executed by those applications. Workloads are defined in the monitoring profile used by DB2 Query Monitor to monitor a given DB2 subsystem.

Legacy Terms and Components

The following components were used in previous releases of products:

- **Audit SQL Collector** – A predecessor to the IBM DB2 Data Access Common Collector (CQC) Version 1.1 that is used by IBM DB2 Query Monitor for z/OS (CQM) Version 3.1 and before. The Audit SQL Collector is replaced by the IBM DB2 Data Access Common Collector (CQC) Version 1.1 in IBM DB2 Query Monitor for z/OS (CQM) Version 3.2.

1. Overview of Shared Product Components

The following topics provide updated and expanded information for the overview section of the IBM DB2 Query Monitor User's Guide.

IBM DB2 Data Access Common Collector for z/OS (CQC)

The IBM DB2 Data Access Common Collector for z/OS V1.1 (CQC), is a separate no-charge product that contains the common collector technology. The advantage of using it with the following three products is the data is collected once through a 'collector' eliminating the need to collect the data multiple times for multiple products thus reducing CPU overhead. The savings can be substantial. This means reduced maintenance since there is a common place for the collector changes which only need to be applied once to keep the separate products synchronized.

The CQC collects a variety of data for the following products:

- IBM DB2 Query Monitor for z/OS
- InfoSphere Guardium S-TAP for DB2 on z/OS
- InfoSphere Optim Workload Replay S-TAP on DB2 on z/OS

The CQC integrates data for these products to provide a complete picture of query activity in the monitored systems. The sections that follow describe how the CQC interacts with these products and their components.

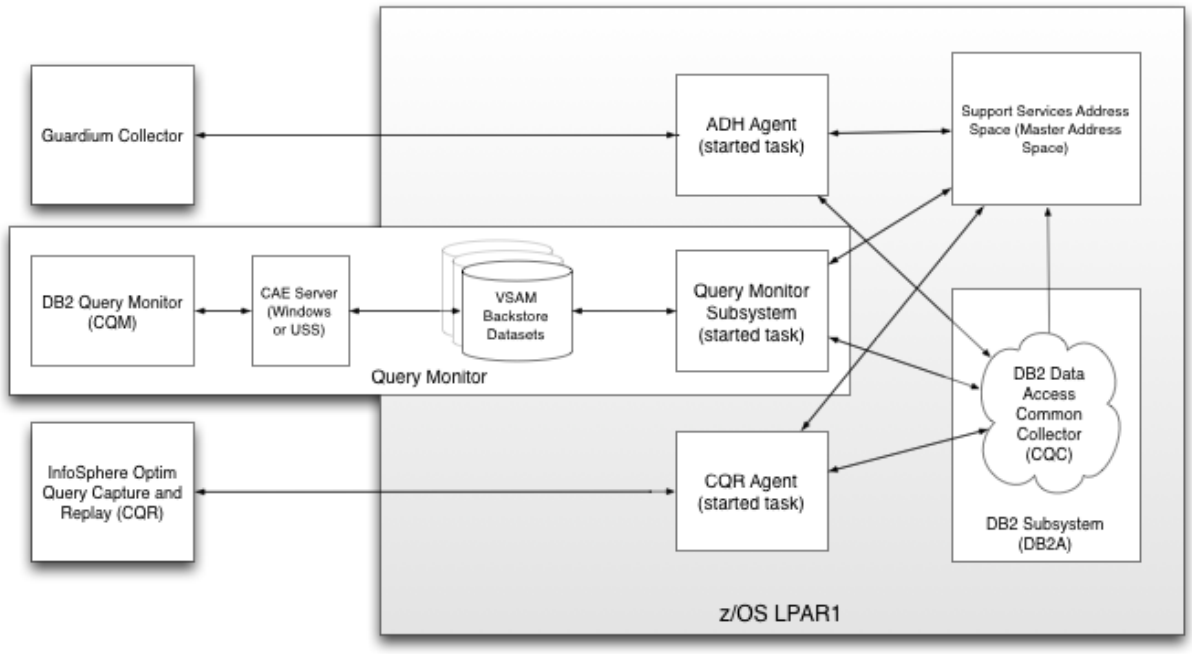

Figure 1. CQC Architecture

Note: *This diagram does not show every component associated with every product. Instead, it provides a simplified view of the system in order to highlight the components that interact with the CQC and the Support Services Address Space.*

The CQC collects the following data:

- SQL metrics
- DB2 object access
- SQL text
- DB2 commands
- Negative SQLCODEs
- Host variables

The CQC is not a stand-alone address space. The CQC provides the collection points for the DB2 address space. The physical code for the CQC runs in the DBM1 address space and collects the data necessary for CQM or ADH or CQR. The data collected by the CQC is stored in memory objects that are owned by the Support Services Address Space.

DB2 Query Monitor Monitoring Agent

DB2 Query Monitor enables you to work with the CQC through the use of monitoring agents. A monitoring agent is the interface that DB2 Query Monitor installs within a DB2 subsystem to capture SQL performance data. When a DB2 Query Monitor Subsystem collects data about a DB2 subsystem, a monitoring agent is at work collecting data about that DB2 subsystem.

If one DB2 Query Monitor Subsystem started task is performing data collections on three different DB2 subsystems, then there are three monitoring agents active for that DB2 Query Monitor Subsystem. Additionally, each agent can optionally be assigned a monitoring profile.

Support Services Address Space (Master Address Space)

For each MVS image, a Support Services Address Space (also referred to as the Master Address Space) is started automatically by CQM, CQR, or ADH. The first of these products that is launched on an MVS image automatically initiates the Master Address Space.

The Master Address Space is a service address space that owns the shared memory objects where the data that is collected by the CQC is staged. The Master Address Space acts as a placeholder for CQC resources and is similar to other master address spaces that are used throughout MVS (MVS and DB2, for example, have master address spaces).

The Master Address Space:

- Never shuts down
- Does not execute any code during the course of existence, except for its initialization routines so therefore does not have to be controlled by an installation
- Owns resources needed by the CQC
- Does not require a formal shutdown and should not be canceled or forced to shut down during the operation of IBM DB2 Query Monitor for z/OS, InfoSphere Guardium S-TAP for DB2 on z/OS, or

InfoSphere Optim Workload Replay S-TAP on DB2 on z/OS. Forcing the master address space to stop causes the abnormal termination of all IBM DB2 Query Monitor for z/OS, InfoSphere Guardium S-TAP for DB2 on z/OS, or InfoSphere Optim Workload Replay S-TAP on DB2 on z/OS started tasks on the LPAR.

About the MASTER_PROCNAME Parameter

The Master Address Space used by a CQM, CQR, or ADH installation is specified using the MASTER_PROCNAME. The MASTER_PROCNAME parameter is required, it must be specified for CQM, CQR, and ADH. When the same MASTER_PROCNAME parameter is specified among product installations (CQM, CQR, or ADH), this causes the product installations to use the same Monitor Address Space.

Master Address Space - Usage Considerations

Stopping the Master Address Space

- Stop the master address space only if directed to do so by IBM Software Support or by a ++HOLD(ACTION) in a PTF. To ensure product stability, the master address space should be stopped only by using the sample job provided in SCQMSAMP member CQMMSTR (for DB2 Query Monitor Version 3.2 and later, use the TCz customization panels to generate this job). As a safeguard, this job verifies that no CQM, CQR, or ADH installations are using the Master Address Space before stopping it.
- During installation, do not stop or start the Master Address Space unless required by product maintenance or instructed to do so by IBM Software Support.

Monitoring the same DB2 subsystem or multiple DB2 subsystems on the same LPAR

- **Monitoring the same DB2 subsystem** - If you use multiple DB2 Data Access Common Collector products (CQM, CQR, ADH) to monitor the same DB2 subsystem, each product must specify the same value for the MASTER_PROCNAME parameter.
- **Monitoring multiple DB2 subsystems** (that reside on the same LPAR) - If you use multiple DB2 Data Access Common Collector products (CQM, CQR, ADH) or multiple instances of the same product to monitor different DB2 subsystems that reside on the same LPAR, each product can have a different value for the MASTER_PROCNAME parameter. **Note:** This configuration is appropriate when running code at different maintenance levels on the same LPAR (for example, if you are testing new maintenance prior to upgrading your production system).

Configuring Multiple Master Address Spaces

A single Master Address Space is required for each LPAR. However, you can configure multiple Master Address Spaces on a single LPAR. Configuring multiple Master Address Spaces on a single LPAR enables you to, for example, configure TEST and PROD implementations of your CQM, CQR, or ADH environment.

Product-Specific Started Tasks

Started tasks are used by CQM, ADH, and CQR to collect and integrate data pertinent to each product. These started tasks include the following:

Started task	Product that uses this started task
ADH Agent	InfoSphere Guardium S-TAP for DB2 on z/OS (ADH)

CQR Agent	InfoSphere Optim Workload Replay S-TAP on DB2 on z/OS (CQR)
DB2 Query Monitor Subsystem	IBM DB2 Query Monitor for z/OS (CQM)

ADH Agent and CQR Agent

The ADH Agent is a started task that collects audit data in a Guardium environment. The ADH Agent filters and sends SQL data to the CQR Agent. Users can then view reports on a workstation. One ADH Agent is required for each DB2 that you intend to audit.

The ADH Agent and the CQR Agent communicate with the Guardium Appliance using a TCP/IP connection. The ADH Agent and the CQR Agent use filtering policies created by you to determine what data to collect. The policy specifies filter information, such as which jobs and data sets to be monitored for data accesses.

DB2 Query Monitor Subsystem

The DB2 Query Monitor Subsystem is a started task that integrates the data collected by the CQC to provide a complete picture of query activity in a monitored system. DB2 Query Monitor uses an exception-processing layer to identify exceptions.

The DB2 Query Monitor Subsystem collects data, manages intervals, archives historical data, and installs/removes the required instrumentation components.

Data collection consists of the following components:

- Address space initialization
- Control block allocation
- Address space termination and cleanup
- DB2 discovery
- Instrumentation install and de-install
- First-level data summarization
- Performance history file population and management
- Exception recognition and notification
- Field diagnostic generation
- Interval processing
- Historical data archival
- Historical data archive history maintenance

A single DB2 Query Monitor Subsystem can monitor from 1 to 64 DB2 subsystems on a single LPAR. A minimal implementation will have one DB2 Query Monitor Subsystem per LPAR. An extremely granular implementation will have one DB2 Query Monitor Subsystem for each monitored DB2 subsystem.

DB2 Query Monitor Consolidation and Analysis Engine (CAE)

The Consolidation and Analysis Engine (CAE) is comprised of three sub-components, the CAE Agent, the CAE Server, and the CAE Web Client.

DB2 Query Monitor CAE Agent

The CAE Agent is a non-Java address space that runs under MVS. The CAE Agent does not use any ZFS/HFS facilities. The CAE Agent is required for the CAE Server to be able to access information from the DB2 Query Monitor subsystem on an LPAR. The CAE Agent can run as a started task or as a batch job under the control of JES.

DB2 Query Monitor CAE Server

The CAE Server interfaces to any number of DB2 Query Monitor Subsystems through any number of CAE Agents per CAE Server. A CAE Agent is required on every MVS image that host DB2 Query Monitor Subsystems. Any number of CAE Web Clients can connect to the CAE Server, and the CAE Server acts as a consolidator, looking at the DB2 Query Monitor Subsystem, no matter where that DB2 Query Monitor Subsystem resides, to give a consolidated view.

DB2 Query Monitor CAE Web Client

The CAE Web Client provides a user-friendly graphical interface that is menu driven and context sensitive enabling users to quickly and painlessly define scopes. It serves as a centralized browsing and configuration facility, and enables users to efficiently manage events, correlations, and responses.

2. DB2 Query Monitor concepts

The topics that follow provide information about DB2 Query Monitor concepts you need to understand before getting started with the product:

- Data collection
- SQL workloads
- Summary data, exception data, and alert data
- OPTKEYS
- MAX_SQLCODES and MAX_SQLCODE_DETAIL

Data collection

DB2 Query Monitor uses the CQMPARMS data set and the monitoring profile to determine what performance data to collect from any given DB2 subsystem. Performance data is collected from the monitored DB2 subsystem by a monitoring agent, and is for the most part stored in z/OS data spaces until a DB2 Query Monitor interval is completed, at which time the data is written to the Performance History Files.

SQL Workloads

An **SQL Workload** (also referred to as a "workload") provides a way of identifying a group of applications to DB2 Query Monitor so that performance data can be collected for SQL statements executed by those applications. Workloads are defined in the monitoring profile used by DB2 Query Monitor to monitor a given DB2 subsystem. Workloads can be said to have three characteristics:

- The workload name, which must be unique and is used purely to identify the workload.
- The workload filters, in which identifiers such as plan name, subsystem name, authorization id and so on are used to identify which applications to include or exclude from data collection.
- The thresholds for collecting exceptions and alerts, such as elapsed time, CPU time or GETPAGE requests.
- OPTKEY override settings to provide customized summary data collection specific to an individual workload.

Summary data, Exception data, and Alert data

The DB2 Query Monitor Subsystem maintains three basic types of performance data. The data types are summary, exceptions, and alerts.

Summary data

Summary data is performance data that is summarized for each unique SQL statement that is executed in a DB2 Query Monitor interval. The values collected are totals and averages. For example, the elapsed time values are averages. A unique SQL statement is represented by a unique value of:

Plan + Program + Section + Statement number + Statement Type

When DB2 Query Monitor is first installed, by default, partial summary data is collected for all SQL executed in the DB2 subsystem(s) being monitored. The default collection does not include the Statement Number or the Statement Type data elements. SQL from specific workloads can be excluded from summary collection by using the monitoring profiles.

Summary data that is not collected by default is the negative SQLCODE data. This collection must be activated using the appropriate CQMPARMS startup parameters.

Exception data

DB2 Query Monitor exception data is data for individual SQL calls that have exceeded user-defined thresholds (such as elapsed time or CPU time). These thresholds are defined in the monitoring profile; without a monitoring profile, no exception data is collected.

Alert data

DB2 Query Monitor alerts are SQL events that require immediate attention. Similar to exceptions, alert data is about individual SQL calls that have exceeded user-defined thresholds. Alerts can also be classified as exceptions – this is strongly recommended because the alert/exception data is written to Performance History Files and are therefore available for later analyses. It is possible, but not recommended, to have alerts that are only viewable through the CAE Web Client user interface, by only defining them as alerts, and not as exceptions.

Alert specifications can be identical to the corresponding exception specifications, thereby generating alerts and exceptions simultaneously, but this may mean you either generate large numbers of 'immediate attention' alerts or have too narrow a view of what constitutes an exception SQL event. When defining thresholds for alerts that the alert thresholds be higher than the exception thresholds. To reiterate, alerts are SQL events which require immediate attention, whereas this is not necessarily the case with exceptions.

Note: Alerts that do not also qualify as exceptions are not available in the "View Exceptions" area of the ISPF interface or the "Exceptions" perspective of the CAE Web Client. Such alerts are not stored in Performance History Files and therefore are not available in the DB2 Query Monitor Performance Database, if used. For this reason, every alert should also qualify as an exception.

OPTKEYS

DB2 Query Monitor collects and summarizes SQL activity using the following basic key:

Plan + Program + Section

The OPTKEYS parameter enables DB2 Query Monitor users to specify additional levels of summarization. Since the additional levels of summarization OPTKEYS offer can significantly increase both the volume of data that is collected by DB2 Query Monitor and the amount of data DB2 Query Monitor stores in data spaces, you should be careful when adding OPTKEYS.

NOTE: We recommend that you specify OPTKEYS in monitoring profiles, not in CQMPARMS.

Examples of how to evaluate the use of OPTKEYS

Using OPTKEYS for an ad-hoc query-based DB2 - For an ad-hoc DB2, OPTKEYS(TEXT) is probably not useful, because most of the SQL in the systems is unique and not reused. But the OPTKEY (PTEXT) might be very useful, because the literals will be removed from the ad-hoc SQL and the SQL may then summarize appropriately. The OPTKEYS(AUTHID) may be useful in this same system as the number of users is most likely relatively small. **Note:** Omitting TEXT from the OPTKEYS specification does not result in the text of exception dynamic SQL statements being lost. The SQL statement text associated with exception events is always recorded.

Using OPTKEYS for an On-Line Transaction Processing (OLTP)-based DB2 - In an OLTP-based DB2, if the dynamic SQL is repeated, OPTKEYS(TEXT) might be very useful, whereas OPTKEYS(AUTHID) is probably not useful. If the number of distinct AUTHIDs is large, this will cause more overhead and minimal summarization. If there is a single AUTHID used for all DB2 SQL, then all of the summary data will be in a single bucket and there is no value in using the OPTKEYS. If there is no dynamic SQL, there is no benefit in using OPTKEYS(TEXT) or OPTKEYS(PTEXT), because the default summarization key is enough to identify each SQL statement being executed.

The number of summary buckets grows quickly and this can be exacerbated by specifying multiple OPTKEYS. For example, let's assume that a system has 1,000 distinct dynamic SQL statements and 1,000 users. Also assume that each user will execute every SQL statement at least once during each interval. Finally add into the assumptions that each SQL statement accesses 3 application objects plus the 12 objects needed to PREPARE a dynamic SQL statement for execution. For this example, the OPTKEYS will affect the summary collection as follows:

- Specifying OPTKEYS(TEXT) – This adds 1,000 summary buckets to the METR data and 15,000 (1,000*15) buckets to the OBJS data.
- Specifying OPTKEYS(AUTHID) – This adds 1,000 summary buckets to the METR data and 15,000 (1,000*15) buckets to the OBJS data.
- Specifying OPTKEYS(TEXT,AUTHID) – This adds 1,000,000 (1,000*1,000) summary buckets to the METR data and 15,000,000 (1,000*1,000*15) buckets to the OBJS data.

When you choose OPTKEYS settings, it is important to determine what categories of summarization are meaningful and useful in your environment. For example, with SAP, in contrast to the settings for the OLTP-based DB2 and the Data Warehouse-based DB2 discussed above, it is probably more appropriate to summarize by WSTRAN and TEXT. The reason for this is that there is only one AUTHID used by SAP (usually SAPR3), whereas it is WSTRAN that helps you identify the user. Since SAP only uses dynamic SQL which is subject to repeated execution, TEXT is needed to be able to summarize by SQL statement. Don't forget, however, that these both act as multipliers to the number of summary buckets, so a large number of users may mean that specifying WSTRAN is undesirable.

MAX_SQLCODES and MAX_SQLCODE_DETAIL

The MAX_SQLCODES and MAX_SQLCODE_DETAIL parameters control the summary data collection for negative SQLCODEs. The summary data is accessed using the "View SQLCODEs" option in the ISPF interface and the "SQLCODES" perspective in the CAE GUI.

The MAX_SQLCODES parameter sets the limit on the number of unique SQLCODES for which summary information is collected. The MAX_SQLCODE_DETAIL sets the limit on the number of detail records which is collected for each occurrence of a negative SQLCODE.

Note: Because we are dealing with summary information in this area, the detail collected is very limited. The detail collected consists of the SQLCA and the text of the SQL statement, if the statement text is available. There is no performance data or host variable information available in this part of the DB2 Query Monitor product. Host variables and performance metrics for statements which end with negative SQLCODES are kept with the exception record for the event.

A recommended starting value for MAX_SQLCODES is 250. This will most likely be larger than the number of distinct negative SQLCODES in a given interval.

A recommended starting value for MAX_SQLCODE_DETAIL is between 50 and 100. This will normally allow a DB2 Query Monitor user to determine which negative SQLCODES are being used as coding techniques by which programs. Once the codes being used as coding techniques are identified, they can be excluded from exception and alert processing in the monitoring profile. How to do this is discussed in detail in the section dealing with setting up monitoring profiles.

3. Creating an Implementation Strategy

The following sample scenarios help you to identify the most appropriate implementation strategy for DB2 Query Monitor at your site.

Configuration Example 1

Objective

To monitor one DB2 on a single LPAR.

Solution

To monitor one DB2 on a single LPAR in a non-data sharing environment, the following are required:

- A DB2 Query Monitor Subsystem
- One Master Address Space

Additionally, if you want to use the CAE Web Client, you must also install and configure the following CAE components:

- CAE Agent
- CAE Server

Configuration Example 2

Objective

To monitor two DB2s across two LPARS in a non-data sharing environment.

Solution

To monitor two DB2s across two LPARS in a non-data sharing environment, the following are required:

- One DB2 Query Monitor Subsystem installed on each LPAR
- One Master Address Space installed on each LPAR

Additionally, if you want to use the CAE Web Client, you must also install and configure the following CAE components:

- One CAE Agent installed on each LPAR
- One CAE Server installed on one of the LPARs

Configuration Example 3

Objective

To monitor two LPARS in a data sharing environment.

Solution

To monitor two LPARS in a data sharing environment, the following are required:

- One DB2 Query Monitor Subsystem installed on each LPAR

- One Master Address Space installed on each LPAR

Additionally, if you want to use the CAE Web Client, you must also install and configure the following CAE components:

- One CAE Agent installed on each LPAR
- One CAE Server installed on one of the LPARs

4. Getting started with DB2 Query Monitor

The sections that follow provide a series of steps you need to follow to configure your installation's DB2 Query Monitor Subsystem(s). These steps help you to ensure that DB2 Query Monitor uses a minimal amount of resources for its collection process yet still provides you with access to the data you need to tune your SQL.

IMPORTANT: The proper configuration of DB2 Query Monitor Subsystem to collect SQL data ensures that you do not incur excessive overhead when using DB2 Query Monitor.

The topics that follow also provide information about:

- **Step 1: Gather and analyze performance data** - Performance data provides you with a basis to set up monitoring profiles that use exception and alert thresholds appropriate for your site. Continue reading: "Step 1: Configuring the DB2 Query Monitor Subsystem to Collect Performance Data" for more information.
- **Step 2: Configure the CQMPARMS file** – The proper configuration of the CQMPARMS file enables you to define what data is collected as well as the level of detail at which data is collected. Continue reading: "Step 2: How to configure the CQMPARMS file and the DB2 Query Monitor parameters it uses to tailor data collection" for more information.
- **Step 3: Configure a monitoring profile** – The proper configuration of a monitoring profile requires that you perform the following substeps:
    - Step 3.1: Setting exception and alert thresholds
    - Step 3.2: Excluding negative SQL codes from analysis
    - Step 3.3: Creating profile lines

  Continue reading: "Step 3: Configure a monitoring profile" for more information.

Step 1: Gather and analyze performance data

This step describes how to gather performance data about your system and use that data to decide how to best setup monitoring profiles and define reasonable exception and alert thresholds.

QUESTION 1: Do you have an existing base of SQL Metrics?

> **YES:** If you already have existing base of SQL metrics (such as average elapsed time, average CPU time, number of SQL calls), then you can use these metrics to determine a good initial monitoring profile workload. You can often obtain a base of SQL metrics from SMF data using a tool such as IBM DB2 Performance Expert for z/OS. The base profile you derive from this data can be in place the first time the DB2 Query Monitor Subsystem is started.
>
> **NO:** If you do not have an existing base of SQL metrics, then you can use DB2 Query Monitor to gather information about SQL volume, average CPU use, average elapsed time, and negative SQLCODEs throughout your systems. This process requires at least 2.5 days, as described below:
>
> - **Customize DB2 Query Monitor and initiating the DB2 Query Monitor Subsystem (requires at least 0.5 days)** - The initial setup and configuration should have the DB2 Query Monitor Subsystem monitoring one or more DB2 subsystems without specifying a monitoring profile.

In this configuration the DB2 Query Monitor Subsystem task is gathering summary data only. There will not be any data in exceptions, alerts, or current activity. Set MAX_SQLCODES and MAX_SQLCODE_DETAIL in CQMPARMS to 250 and between 50 and 100 as recommended in the section on CQMPARMS below, ensuring that data about negative SQLCODEs is collected. The DB2 Query Monitor Subsystem issues the following message (where *ssid* is the DB2 subsystem name), which can be ignored:
CQM3302I **WARNING** MONITORING AGENT FOR *ssid* WILL NOT COLLECT EXCEPTION DATA OR CURRENT ACTIVITY

- **Run DB2 Query Monitor to collect summary data (requires 24 hours)** - After DB2 Query Monitor is configured to collect summary data, let it run for 24 hours, if possible.
- **Analyze and review data to determine appropriate thresholds (requires 1 day)** – After collecting summary data for 24 hours, you should have enough data to enable you to determine the appropriate thresholds. Review the data (using the Activity Summaries and SQLCODE options in DB2 Query Monitor) to determine the appropriate thresholds to use for the monitoring profile lines in your monitoring profile.

QUESTION 2: How many DB2 Query Monitor Subsystems does your site need?

RECOMMENDATION: A DB2 Query Monitor Subsystem can monitor up to 64 DB2 subsystems on a single z/OS LPAR. In general, one DB2 Query Monitor Subsystem for each z/OS LPAR is recommended.

EXCEPTIONS: The following scenarios describe situations in which the one DB2 Query Monitor Subsystem per z/OS LPAR does not apply. In the situations described below, more than one DB2 Query Monitor Subsystem per LPAR is appropriate:

- **Different DB2 subsystems might have different monitoring requirements.**

  A DB2 subsystem used by development might have a longer interval length or retention period than a DB2 subsystem that is used by QA or in a production environment. You can use the INTERVAL and RETAIN parameters to specify the interval length and retention period appropriate for these cases.

- **The DB2 Query Monitor user's view of DB2 subsystems being monitored should be limited.**

  Access to the DB2 Query Monitor data from a given DB2 Query Monitor Subsystem can be restricted using an external security system such as RACF. If an installation decides to setup a specific DB2 Query Monitor Subsystem for each DB2 subsystem to be monitored, users could be restricted to only being allowed to access DB2 Query Monitor Subsystems for DB2 subsystems where the user is also authorized. **Note:** You can use the CAE Web Client to access DB2 Query Monitor data on all DB2 subsystems on all LPARs that are connected to the CAE Server.

Step 2: Configure the CQMPARMS file

DB2 Query Monitor uses a set of startup parameters which define how DB2 Query Monitor is implemented, including the DB2 Query Monitor Subsystem name, the monitored DB2 subsystems, and the length of the interval. These parameters are stored in a data set allocated to the CQMPARMS DD

statement in the DB2 Query Monitor JCL. **This data set should be allocated as a partitioned data set (PDS) with a separate member for each set of startup parameter definitions.**

Each individual DB2 Query Monitor Subsystem must have its own set of startup parameters. Using a PDS allows all of the startup parameters for the various DB2 Query Monitor Subsystems to be stored in a single data set. In addition, the individual members can be edited while the DB2 Query Monitor Subsystem is active. If a sequential data set is used for each DB2 Query Monitor Subsystem, the parameters can only be changed while the DB2 Query Monitor Subsystem is shut down.

An example of the contents of the CQMPARMS file is shown below. Refer to the DB2 Query Monitor User's Guide for more information about the complete list of parameters that can be used in CQMPARMS.

```
AUTHID(DB2USER)          -
  MONITOR(DB2A,DB2APROF,DB2B,DB2BPROF) -
  SUBSYS(I71A)             -
  INTERVAL(60)             -
  RETAIN(96)               -
  OPTKEYS(TEXT)            -
  ALERT_LIMIT(100)         -
  MAX_SQLCODES(250)        -
  MAX_SQLCODE_DETAIL(100)  -
  STORCLAS(DB2TEMP)        -
  MGMTCLAS(DB2)            -
  DATACLAS(VSHAR33)        -
  EXCPDATA_DSN(CQMHLQ.I71A.EDATA.D&YYMMDD..T&LHR.&LMIN..&INTV.) -
  EXCPINDX_DSN(CQMHLQ.I71A.EINDX.D&YYMMDD..T&LHR.&LMIN..&INTV.) -
  METRDATA_DSN(CQMHLQ.I71A.METRD.D&YYMMDD..T&LHR.&LMIN..&INTV.) -
  OBJSDATA_DSN(CQMHLQ.I71A.OBJSD.D&YYMMDD..T&LHR.&LMIN..&INTV.) -
  TEXTDATA_DSN(CQMHLQ.I71A.TEXTD.D&YYMMDD..T&LHR.&LMIN..&INTV.) -
  SQLCDATA_DSN(CQMHLQ.I71A.SQLCD.D&YYMMDD..T&LHR.&LMIN..&INTV.) -
  DB2CDATA_DSN(CQMHLQ.I71A.DB2CD.D&YYMMDD..T&LHR.&LMIN..&INTV.) -
  EXCPDATA_SPACE_UNITS(TRKS) -
  EXCPINDX_SPACE_UNITS(TRKS) -
  METRDATA_SPACE_UNITS(CYLS) -
  OBJSDATA_SPACE_UNITS(CYLS) -
  TEXTDATA_SPACE_UNITS(CYLS) -
  SQLCDATA_SPACE_UNITS(TRKS) -
  DB2CDATA_SPACE_UNITS(TRKS) -
  EXCPDATA_PRIMARY(45)     -
  EXCPINDX_PRIMARY(15)     -
  METRDATA_PRIMARY(5)      -
  OBJSDATA_PRIMARY(5)      -
  TEXTDATA_PRIMARY(3)      -
  SQLCDATA_PRIMARY(15)     -
  DB2CDATA_PRIMARY(15)     -
```

Data Set Sizing

The allocation specifications for the VSAM back-store data sets are also in the CQMPARMS data set. Typically, once the parameters are set up, they are seldom if ever changed. Refer to the DB2 Query Monitor User's Guide for complete instructions on calculating the space for these data sets. You will

need to know the volume and mix of SQL being executed in order to perform the calculations. This can be determined from several sources such as:

- DB2 Performance Expert statistical reports
- Data from an SQL monitor product (such as DB2 Query Monitor)
- Other DB2 capacity planning products.

If the volume and mix of SQL in the workload cannot be easily determined, a simple method is to use the default allocations in whole cylinders. Track the number of extents used by the data sets during the first few days or weeks that DB2 Query Monitor is running. If the data sets take multiple extents during every interval, then increase the primary allocation until the data will fit in a single extent during intervals when the SQL volume is low.

During intervals when SQL volume is highest, the back-stores should always be allowed to take multiple extents. This helps to minimize the DASD used for the back-stores because it avoids over-allocation of the back-store data sets for intervals when SQL volumes are low.

SMS management is highly recommended for the VSAM back-store data sets. There are seven unique data sets created for each interval. Using a RETAIN parameter of 96 will result in 672 VSAM back-store data sets being created and retained.

Step 3: Configure a monitoring profile

Monitoring profiles have four basic functions in DB2 Query Monitor. These functions are:

- To define exception or alert thresholds for particular SQL workloads
- To activate the current activity display for particular SQL workloads
- To exclude particular SQL workloads from summary data collection
- To override the OPTKEYS settings for the summary data collection for a specific SQL workload

Step 3.1 Determine how many monitoring profiles to create

This step describes how to determine the number of monitoring profiles you should create to monitor your DB2 subsystem(s).

QUESTION: How many monitoring profiles should I configure for my environment?

> **RECOMMENDATION:** We recommended that you create a separate monitoring profile for each DB2 subsystem that you want to monitor. This setup will reduce the number of monitoring profile lines in each monitoring profile, and in doing so might reduce overhead.
>
> **EXCEPTION:** The exception to this recommendation is when you create a monitoring profile for a data sharing group. In this situation, you should use one monitoring profile for all members of the data sharing group, because thresholds, OPTKEYS settings, and negative SQLCODE exclusions will usually be the same for all data sharing group members.

Step 3.2: Create a monitoring profile

Refer to the '*Working with profiles*' chapter of the *IBM DB2 Query Monitor User's Guide* for a description of how to create a monitoring profile. After you have created a monitoring profile, refer to the sections below for information about how to configure monitoring profile lines to a monitoring profile.

Step 3.3: Configure one or more monitoring profile lines

A **monitoring profile** consists of one or more **monitoring profile lines**. Each monitoring profile line applies to one **workload**. Monitoring profiles are made up of one or more monitoring profile lines. Each monitoring profile line consists of the following elements:

- Line type (Include or Exclude)
- Miscellaneous flags
- Workload definition
- Exception thresholds
- Exception limit
- SQL codes excluded from exceptions
- Alert thresholds
- SQLCODEs excluded from alerts
- SQLCODEs excluded from summary collection
- OPTKEYS overrides

A sample of the Update Profile Line Panel is shown below. For further explanation of the fields on this panel, see “Field descriptions for Monitoring Profile Lines” on page 30.

```
---------------------- Update Profile Line for PROF1   -----------------
Option  ===> _____________________________________________________ Scroll
===> PAGE

More:     +

  INCLUDE/EXCLUDE             I     (I=Include, E=Exclude)
  Disable Summary Reporting   N  (Y/N)  Gather Host Variables  Y  (Y/N)
  DB2 Subsystem               *___        Plan Name      *________
  Program Name                *________
  AUTHID                      *________   JOBNAME        *________
  Connection ID               *________   CORRID         *____________
                                          CORRNAME       *____________
  Workstation User            *________________
  Workstation Trans           *_____________________________________
  Workstation Name            *____________________
  Workload Name               _____________________________________
  Exception CPU               00 : 00 : 00 . 000000
  Exception Elapsed           00 : 00 : 00 . 000000
  Exception Getpages          0_____________
  Exception SQL Calls         0_____________
  Exception Limit             0_____________
  Generate SQLCODE Exceptions Y  (Y/N)
  Exclude Exception SQLCODEs  N  (Y/N)
  Alert CPU                   00 : 00 : 00 . 000000
  Alert Elapsed               00 : 00 : 00 . 000000
  Alert Getpages              0_____________
  Alert SQL Calls             0_____________
  Generate SQLCODE Alerts     N  (Y/N)
  Exclude Alert SQLCODEs      N  (Y/N)
  Exclude Summary SQLCODEs    N  (Y/N)
```

```
  Override OPTKEYS            N  (Y/N)
     OPTKEYS(TEXT)            N  (Y/N)
     OPTKEYS(AUTHIDS)         N  (Y/N)
     OPTKEYS(CORRID)          N  (Y/N)
     OPTKEYS(CORRNAME)        N  (Y/N)
     OPTKEYS(WSUSER)          N  (Y/N)
     OPTKEYS(WSTRAN)          N  (Y/N)
     OPTKEYS(WSNAME)          N  (Y/N)
     OPTKEYS(CALLS)           N  (Y/N)
     OPTKEYS(PTEXT)           N  (Y/N)
```

The techniques described below enable you to efficiently create individual monitoring profile lines.

Step 3.3.1: Excluding SQLCODEs from exception and alert processing

Most negative SQLCODEs will be excluded from exception and alert processing. Only the first matching profile line is used for evaluating the SQLCODEs to be excluded. This means that the list of exclude alert SQLCODEs must be included on every profile line in order to ensure that the SQLCODEs are excluded for all workloads. In general, the following rules apply to the alert and exception processing of negative SQLCODEs

- In general, negative SQLCODEs that are used as application coding techniques should be excluded from exception processing (they should not be allowed to produce exceptions).
- In general, almost all negative SQLCODEs should be excluded from alert processing (they should not be allowed to produce alerts). Alerts should only be generated for negative SQLCODEs that are important enough for a DBA to take immediate action. An exception to this would be the case where the DB2 Query Monitor user wants to use the alert message board as a management technique for storing negative SQLCODEs.

QUESTION: What negative SQLCODEs should be excluded from exception and alert processing?

RECOMMENDATION: Collect summary data about the negative SQLCODEs at your site and build an SQLCODE exclusion list for use in your monitoring profile lines.

The simplest way to build the SQLCODE exclusion list is as follows:

1. Allow DB2 Query Monitor to collect summary negative SQLCODE data for 24 hours.
2. Load the collected SQLCODE data into the QM DB2 Performance database.
3. Query the QM Performance database to get a list of all the distinct negative SQL codes collected.
4. Build the most generic line profile line first. In the example shown above, that means the last line.
   4.1. Specify “Y” for “Exclude Exception SQLCODEs” and for “Exclude Alert SQLCODEs”.
   4.2. Enter the list of SQLCODES from item 3 on the panel headed “Exception SQL Code Exclusion List”, and then hit F3.
   4.3. Enter the same list on the panel headed “Alert SQL Code Exclusion List”.
   4.4. Save the workload definition line

When you have created a single monitoring profile line that has the SQLCODE exclusion list, you can copy and edit that line as needed:

1. Replicate the generic line.
2. Edit the new line and change the workload definition, thresholds and exclude SQLCODEs as appropriate. Remember, you started with a complete list of excludes when the line was replicated.
3. Move the new line to the appropriate place in the monitoring profile.

Step 3.3.2: Determining OPTKEYS overrides

QUESTION: What is the intended use of the monitoring profile?

SANDBOX DB2 SUBSYSTEM: Profiles intended to be used for a "sandbox DB2 subsystem" are typically created to verify the installation of the DB2 Query Monitor. As such, they typically consist of a single profile line with very low exception and alert thresholds. In addition, normally all OPTKEYS override settings are specified as "Y". This will result in summarizing all the SQL by all possible OPTKEYS and generating exceptions for most, if not all, of the SQL executed in the DB2 subsystem. This is typically not a problem as the sandbox DB2 subsystem is an extremely well controlled and the low-volume environment.

DEVELOPMENT DB2 SUBSYSTEM: In a development DB2 subsystem, DB2 Query Monitor is typically used in a problem determination mode, instead of a true performance profile mode. Since the volume is typically much lower than production, OPTKEYS settings may be dramatically different than a production environment.

- CALLS is almost universally specified in a development DB2 environment.
- TEXT or PTEXT is typically activated in a development DB2 environment where dynamic SQL is used by the applications. TEXT is used when the application uses parameter markers in their SQL, and PTEXT is used when literal values are used in dynamic SQL.
- AUTHIDS is frequently used in a development environment in order to attract what SQL each programmer is executing.
- CORRID is not normally used in the development DB2 system.
- CORRNAME is usually specified even in development for CICS applications.
- The OPTKEYS specific to distributed applications (WSUSER, WSTRAN, WSNAME) are typically used if they are being coded by the applications. If the application is not supplying those fields, the OPTKEYS are typically not specified.

QA DB2 SUBSYTEM: Monitoring profiles that are intended for use with a QA DB2 subsystem are typically mirror images of the production profiles. An exception may be that the exception and alert thresholds would be set lower than production. This lower setting may be used to compensate for a lower transaction rate or volume of data in a QA DB2 subsystem.

PRODUCTION DB2 SUBSYSTEM – SINGLE APPLICATION: For a production DB2 subsystem with a single application, the monitoring profile, often consists of a single profile line. An exception to this might be the use of one monitoring profile line for batch work and a second monitoring profile line for online transactions. There might also be good reason to add additional monitoring profile lines to the monitoring profile when different exceptions/alert thresholds and/or when using OPTKEYS overrides.

PRODUCTION DB2 SUBSYSTEM – MULTIPLE APPLICATIONS: For a production DB2 subsystem shared by multiple applications, there will usually be at least one monitoring profile line per application. This

allows for different exceptions/alert thresholds, based on the individual application requirements. It also allows for tailoring the OPTKEYS overrides to the individual applications.

Step 3.3.3: Selecting production OPTKEYS

All of the OPTKEYS described in this section are optional keys. None of them are required for the proper function of QM. All of them will increase the volume of data collected by QM. Therefore all of the optional keys should be used with appropriate caution. Each of the available OPTKEYS will be discussed in detail in this section.

STEP 3.3.3.1: Setting OPTKEYS: TEXT and PTEXT

QUESTION: Does this application use dynamic SQL?

YES: Does the SQL use parameter markers?

YES: Set the override OPTKEYS(TEXT) to "Y".

NO: Set the override of OPTKEYS(PTEXT) to "Y".

NO: Neither the OPTKEYS setting of TEXT or PTEXT will have any effect on the data collection.

STEP 3.3.3.2: Setting OPTKEYS: AUTHIDS

The AUTHIDS OPTKEY should be used with caution. It has the potential of causing DB2 Query Monitor to exponentially expand the volume of detail collected. There are situations where the AUTHIDS OPTKEY can provide a useful navigation path into the application SQL. (i.e when a single AUTHID represents an entire application). Distributed applications which use an application gateway server are typically good candidates for using the AUTHIDS override.

If the application uses a unique AUTHID for each end user, then the AUTHIDS OPTKEY will most likely generate excessive data volumes. Examples of this usage are typically TSO based applications.

Note: Exception and alert records will always contain the primary AUTHID regardless of the AUTHIDS OPTKEY setting for the workload causing the exception/alert to be generated.

STEP 3.3.3.3: Setting OPTKEYS: CORRID and CORRNAME

The CORRID OPTKEY is the unaltered Correlation ID used by DB2 for the SQL statement.

The CORRNAME OPTKEY indicates whether or not the CORRNAME field will be added to the uniqueness criteria for all future DB2 SQL Statements. The CORRNAME OPTKEY directs DB2 Query Monitor to move only certain subsets of bytes from the originating DB2 correlation ID to the target summarization record during the collection process. These subsets of bytes vary depending on the type of connection to DB2 (for example, TSO, BATCH, RRSAF, CICS, IMS, etc.).The bytes that will be moved for the various connection types are shown below (the remaining right-most bytes will be space padded with EBCDIC blanks):

- TSO, CAF, RRSAF - Bytes 1-8 of the originating correlation ID.
- CICS - Bytes 5-8 of the correlation ID (Transaction ID).
- IMS - Bytes 5-8 of the correlation ID (IMS PST#).

Note: OPTKEYS(CORRNAME) and OPTKEYS(CORRID) are mutually exclusive (only one or the other can be specified at any time). If OPTKEYS(CORRID) is used, the regular CORRID is collected, if

OPTKEYS(CORRNAME) is coded, the field is filled in according to the TSO/CAF/RRSAF/CICS/IMS descriptions above.

QUESTION: Is this a CICS application?

> **YES:** The override of OPTKEYS(CORRNAME) should be set to "Y". In the CICS environment, the CORRNAME OPTKEY is a translation of the CORRID to the 4 character CICS transaction code. This allows for CICS transactions which use pool threads to summarize into a single bucket based on the transaction code.
>
> **NO:** Set the override OPTKEYS(CORRNAME) to "N".

STEP 3.3.3.4: Setting OPTKEYS: AUTHID

QUESTION: Is this DB2 subsystem a development subsystem or a production subsystem?

> **DEVELOPMENT SUBSYSTEM:** If development, consider setting the override OPTKEYS(AUTHID) to "Y". This will provide a summary of all of the SQL executed by each individual user. This level of summarization can be useful in a development environment for debugging purposes.
>
> **PRODUCTION SUBSYSTEM:** In a production environment consider setting the override OPTKEYS(AUTHID) to "N". Unless there is a compelling reason to collect a summary of the SQL statements for each AUTHID, this OPTKEY should be used with extreme caution in a production environment. In some production environments there may be literally thousands of AUTHIDs all executing the same SQL statements. This can result in very high memory usage in DB2 Query Monitor and excessive DASD utilization in the Performance History Files. On the other hand, in some DB2 subsystems, this can be a very useful OPTKEY. If a single unique AUTHID is used for each application then the AUTHID OPTKEY can be very useful.

STEP 3.3.3.5: Setting OPTKEYS: CALLS

QUESTION: Do you need the ability to measure the performance metrics of the component parts of an individual SQL statement (e.g. PREPARE, OPEN, FETCH. Etc.)?

> **YES:** If yes, then you will need to set the override OPTKEYS(CALLS) to "Y". This enables the collection of performance metrics down to the individual call component of every unique SQL statement.
>
> **NO:** Set the override OPTKEYS(CALLS) to "N".

STEP 3.3.3.5: Setting OPTKEYS that are unique to distributed applications: WSTRAN, WSUSER, WSNAME

WSTRAN - The name of the workstation submitting the SQL. This OPTKEY is the equivalent of the CORRNAME OPTKEY on a CICS transaction. However, this OPTKEY applies to distributed transactions

WSUSER - This OPTKEY is the DISTSERV equivalent to the AUTHIDS OPTKEY. As with the AUTHID OPTKEY, selecting this OPTKEY may cause QM to collect and excessive amount of summary data. Care should be used when specifying this OPTKEY.

WSNAME - This OPTKEY is the name of the workstation a.k.a. terminal from which the SQL was submitted. As with the WSUSER OPTKEY, selecting this OPTKEY may cause QM to collect and excessive amount of summary data.

Step 3.4: Arrange monitoring profile lines in the proper sequence

Monitoring profile lines are evaluated in the sequence they appear in the monitoring profile. The first matching monitoring profile line (and only the first matching monitoring profile line), is used to evaluate what to do with the SQL statement. Therefore the sequence of the lines in the monitoring profile is very important. Only fields in the workload definition section of the profile line are used for matching criteria.

The search is ended once a workload definition line is matched, whether or not the SQL statement qualifies for an exception or alert based on the thresholds in that profile line. In other words an SQL statement will be evaluated against the thresholds on one and only one workload definition line.

The workload definition lines should be placed in the profile with the most frequently matched line first and in order by decreasing frequency of use. There are exceptions to this rule. If there is a generic "catch all" line with an asterisk ('*') for all matching criteria, that line must be placed last in the sequence. If there are lines with more specific criteria, then these should be placed before those with less specific criteria. For example, a profile line that matches on plan name DSNTEP71 should be placed before one that matches on the more generic plan name of DSN*. Or, a profile line that matches on program name DSN* and AUTHID FRED should be placed before one that matches only on program name DSN*. Once a workload definition line in the monitoring profile is matched, the search is over and subsequent lines in the profile will not be checked for the SQL statement.

A sample profile is shown below:

```
2014/10/07 14:20:06  ----- Update Monitoring Profile ----  Row 1 of 10
Option  ===>                                               Scroll ===> PAGE
Profile Name: SAMPLE

C:I-Insert,U-Update,R-Repeat,D-Delete,C-Copy,M-Move,B-Before,A-After
CMD  WORKLOAD NAME                  INCL\EXCL SSID JOBNAME  Plan     Program
  -  --------------------------     -      ---- -------- -------- --------
  _  CICS Transactions                  I      *    CICS*    *        *
  _  IMS TM Work                        I      *    IMS*     *        *
  _  Human Resources batch work         I      *    HR*      *        *
  _  Accounts Payable batch work        I      *    AP*      *        *
  _  Exclude DB2 Performance Monitor    E      *    *        *        DGO*
  _  QMF work                           I      *    *        QMF*     *
  _  QMF for Windows work               I      *    *        RAA*     *
  _  All other work                     I      *    *        *        *
 ***************************** Bottom of Data ******************************
```

Using the monitoring profile shown above the following will be true:

- Work coming in with a job name beginning with "CICS" will use thresholds set in the first profile line.
- Work coming in with a job name beginning with "IMS" will use thresholds set in the second profile line.
- Work coming in with a job name beginning with "HR" will use thresholds set in the third profile line.
- Work coming in with a job name beginning with "AP" will use thresholds set in the fourth profile line.

- Work coming in with a program (package/DBRM) name beginning with “DGO” and (a job name not beginning with (CICS, IMS, HR, and AP) will use thresholds set in the fifth profile line.
- Work coming in with a PLAN name beginning with “QMF” and (a job name not beginning with (CICS, IMS, HR, and AP) and a program (package/DBRM) not beginning with DGO) will use thresholds set in the sixth profile line.
- Work coming in with a PLAN name beginning with “RAA” and (a job name not beginning with (CICS, IMS, HR, and AP) and a program (package/DBRM) not beginning with DGO) will use thresholds set in the seventh profile line.
- All other work will use thresholds set in the eighth profile line.

Another sample profile is shown below:

```
2014/10/07 14:20:06  ----- Update Monitoring Profile ----  Row 1 of 10
Option  ===>                                               Scroll ===> PAGE
Profile Name: SAMPLE

C:I-Insert,U-Update,R-Repeat,D-Delete,C-Copy,M-Move,B-Before,A-After
CMD  WORKLOAD NAME                  INCL\EXCL SSID JOBNAME  Plan     Program
  -  ----------------------------    -      ---- -------- -------- --------
  _   All other work                    I      *    *        *        *
  _   Exclude DB2 Performance Monitor   E      *    *        *        DGO*
  _   QMF work                         I      *    *        QMF*     *
  _   QMF for Windows work             I      *    *        RAA*     *
 ***************************** Bottom of Data ********************************
```

In the profile above, only the first line of the profile is used. Because every SQL statement will match the workload definition specified in the first line, the search will always end with that line.

Recommendation when activating/refreshing monitoring profiles

When you have updated and activated or refreshed a monitoring profile, then it is advised to start a new interval.

NOTE: Refer to the ‘*Working with profiles*’ chapter of the *IBM DB2 Query Monitor User’s Guide* for additional information about working with monitoring profiles.

Appendix A. Additional Information

Field descriptions for Monitoring Profile Lines

INCLUDE/EXCLUDE

The **INCLUDE/EXCLUDE** flag indicates what DB2 Query Monitor is to do with SQL which matches the workload definition for this profile line. INCLUDE indicates that matching SQL should be included in further exception/alert processing. EXCLUDE indicates that matching SQL should be excluded from further exception/alert processing. **NOTE:** Excluding SQL from exception/alert processing will also exclude the matching SQL from the current activity display.

Disable Summary Reporting

The **Disable Summary Reporting** flag is only used for profile lines which are defined as EXCLUDE lines. Setting this flag to **Y** will exclude matching SQL from summary, exception, alert processing and the current activity display. In other words, DB2 Query Monitor will not perform any monitoring of any SQL which matches the workload definition in an exclude line with this flag set to **Y**.

NOTE: The **Disable Summary Reporting** flag only applies to EXCLUDE profile lines. The **Disable Summary Reporting (N)** setting does not, by itself, affect current activity reporting. However, if **Disable Summary Reporting (N)** is defined for an EXCLUDE profile, current activity tracking, exception, and alert processing is turned off. If **Disable Summary Reporting (Y)** is defined for an EXCLUDE profile, all tracking within DB2 Query Monitor is turned off.

Gather Host Variables

The **Gather Host Variables** flag is used to tell DB2 Query Monitor that host variables should be gathered when an SQL statement is tracked in current activity. If host variables are being gathered, they will be written with the resulting exception and/or alert record if appropriate.

Workload Definition

The workload definition section of the profile consists of the following fields:

- DB2 Subsystem
- Plan Name
- Program name
- AUTHID
- JOBNAME
- Connection ID
- CORRID
- Workstation User
- Workstation Trans
- Workstation Name
- Workload Name

These fields are described in the DB2 Query Monitor User's Guide.

When creating a workload definition, the attributes of the SQL statement must match all of the values specified in fields a through j listed above. In other words, the fields are “anded” together to evaluate if the SQL statement match the workload definition. It works something like this:

If **DB2 Subsystem** is a match and **Plan Name** is a match and **Program Name** is a match and **AUTHID** and **JOBNAME** is a match and **Connection ID** is a match and **CORRID** is a match and **Workstation User** is a match and **Workstation Trans** is a match and **Workstation Name** is a match, then the SQL statement matches this workload definition. **Once a match is made, no additional workload definition lines are searched.** For all these fields, the asterisk (‘*’) is a wild-card character.

The Workload Name entry is not used in evaluating whether the SQL matches the workload definition. However, the value in the Workload Name field is included in any exception and/or alert records created for SQL statements matching the workload definition. This field should always be filled in as it helps identify which monitoring profile line matched the characteristics of the SQL statement.

Exception Thresholds

Four user-defined thresholds are used to determine if DB2 Query Monitor should consider an SQL statement an exception:

- Exception CPU
- Exception Elapsed
- Exception Getpages
- Exception SQL Calls

These thresholds are evaluated independently. Therefore, if any one of the thresholds is exceeded, the SQL statement is considered an exception. Setting the value of any threshold to zero **turns exception checking off** for the specified threshold.

Exception Limit

The exception limit defines the maximum number of exceptions which will be captured for SQL matching this lines’ workload definition during a specific interval. The exception counters are automatically reset during interval processing. For example, a value of 100 will limit the number of exceptions captured for this particular workload to 100 for each interval.

Note: If exception limit is set to zero, then no exceptions will be captured for this workload. This is not a ‘no limit’ setting.

The exception limit is useful for preventing QM from capturing an excessive volume of exceptions which might be caused by some outside influence. In theory, once a certain volume of exceptions is captured for a given workload, the QM user should be able to determine and fix the problem with the available data. Therefore, no additional notification of these exceptions is necessary.

Exclude Exception SQLCODEs

By default, all SQL statements which complete with a negative SQLCODE are treated as exceptions (and also alerts). If an installation is using some negative SQL codes as coding techniques, those negative SQL codes can be excluded from exception processing.

Note: If negative SQL codes are excluded from exception processing, the same codes should also be excluded from alert processing. This is important as alerts are sent to the CAE Server and **not** retained in the QM VSAM back-store data sets unless, as recommended, the alert also qualifies as an exception.

Alert Thresholds

There are four user defined thresholds which determine if DB2 Query Monitor should generate an alert for a SQL statement. The thresholds are:

- Alert CPU
- Alert Elapsed
- Alert Getpages
- Alert SQL Calls

These thresholds are evaluated independently. Therefore, if any one of the thresholds is exceeded, an alert is generated and passed to the CAE Agent for the SQL statement. Setting the value of any threshold to zero **turns alert checking off** for the specified threshold.

Exclude Alert SQLCODEs

By default, alerts are generated for all SQL statements which complete with a negative SQL code. If an installation is using some negative SQL codes as coding techniques, those negative SQL codes can be excluded from alert processing.

Notes:

- If negative SQL codes are excluded from exception processing, the same codes should also be excluded from alert processing. This is important as alerts are sent to the CAE Server and **not** retained in the QM VSAM back-store data sets unless as recommended the alert also qualifies as an exception.
- Unlike the Exception thresholds, there is no limit to control the number of alerts written in a given interval.

Exclude Summary SQLCODEs

By default, data is collected on all negative SQLCODEs that occur within the monitored subsystem. SQLCODEs that are, for example, used as coding techniques (that is, SQLCODEs for which you do not want to collect data), can be added to this list and thereby excluded from DB2 Query Monitor data collection.

IMPORTANT: If you add an SQLCODE to this list, no data is collected for that SQLCODE; there will be no indication that this negative SQLCODE is occurring in the system. This option also enables you to turn off summary collection for specified negative SQLCODEs.