



*The World's Largest Community  
of SQL Server Professionals*

# Introducing the SQL Server 2008 Performance Data Collector

Brad M. McGehee  
Microsoft SQL Server MVP  
Director of DBA Education  
Red Gate Software  
[www.bradmcgehee.com](http://www.bradmcgehee.com)

# My Assumptions About You

- You have been a SQL Server DBA or developer for six months or longer.
- You have a basic understanding of SQL Server internals and performance tuning.
- You want to learn how to use the SQL Server 2008 Data Collector to help make your job of a being DBA a little easier.

# Here is What We are Going to Learn Today

- Introduction to the Data Collector
- Data Collector Architecture
- How to Install & Configure the Data Collector
- How to Read & Interpret Data Collector Reports

# Introduction to the Data Collector

- Finally, a Solution all DBAs Need
- What is the Data Collector
- Benefits of the Data Collector
- Limitations of the Data Collector
- How Does the SQL Server 2008 Data Collector Fit Into the Overall Picture of SQL Server Tools

# Finally, a Solution All DBAs Need

- For years, it has been suggested to DBAs that they perform **performance baselines** on their servers so they can better understand what is going on with their SQL Server instances over time, and to plan for the future.
- In addition, DBAs often **hear about performance problems after the fact**, but have no way to go back to see what was happening with the server when the problem occurred.
- While it has always been **possible to create your own performance data warehouse**, it has not been practical.
- **Third-party tools have also been available**, but often they have been expensive.
- To deal with the problem, Microsoft has introduced the **Data Collector in SQL Server 2008**.

# What is the Data Collector

- The Data Collector is an optional feature of SQL Server 2008 that collects and stores a lot of pre-defined information about SQL Server's 2008's status and performance.
- Once data is collected and stored in a database, it can be used for historical baselining, planning, troubleshooting, and much more.
- Built-in analysis reports are included that help the DBA become a more proactive DBA.
- Quick demo from SSMS

# Benefits of the Data Collector

- Easy to install and configure, and comes with 2008.
- Provides **most of the key data** needed by a DBA to resolve many different SQL Server-related problems, especially performance-related ones.
- Allows you to **customize** what data is collected, in case you don't like the default data collected.
- Allows you to **manually query, or create Reporting Services reports**, on the historical data stored in the management data warehouse.

# Limitations of the Data Collector

- It can only collect data from SQL Server 2008 (all editions, except Express).
- Customizing data collection and reports is not particularly easy.
- In most cases, a dedicated SQL Server instance is required, if many instances are to be monitored.
- Memory, CPU and disk I/O resources are consumed on the SQL Server instances being monitored.
- Once installed, it can't be uninstalled, only disabled.

# How Much Overhead is Used by the Data Collector

- According to the latest Microsoft testing, the Data Collector will add about 4% to the current CPU load of the SQL Server instance being monitored.
- The above does not include the overhead of the instance where the MDW resides.
- Using typical data collection settings, the Data Collector can generate about 250-300 MB of data *every day* per SQL Server instance being monitored.
- Each dcexec.exe process uses 30-100 MB memory each, and 2+ are running all the time.

# How Does the SQL Server 2008 Data Collector Fit Into the Overall Picture of SQL Server Performance Tuning

- The Data Collector is **just one more tool**, among the many performance-related tools that come with SQL Server, that can be used by DBAs to collect and analyze performance information.
- In its current form, **it will not fully replace other tools**, although it makes a good attempt.
- The Data Collector is a foundation that Microsoft will expand on in future versions of SQL Server.
- Think of it as a SQL Server data warehouse for storing any SQL Server internal data.

# Data Collector Architecture

- Essential Data Collector Terminology
- Data Collection Sets
- Data Collection Types
- Data Collection Modes
- More Detailed Architecture

# Essential Terminology #1

- **Target:** SQL Server instance being monitored.
- **MDW:** Management data warehouse.
- **Collection Set:** A group of one or more *collection types* that are collected together as a single entity. There are **three default** collection sets:
  - Disk Usage
  - Query Activity
  - Server Activity
- You can create your own custom collection sets.

# Essential Terminology #2

- **Collection Type:** The mechanism for collecting and uploading specific types of data. There are four:
  - T-SQL Query
  - Performance Counter
  - Query Activity
  - SQL Trace
- **Collection Item:** Is the data gathered by a collector type, such as a query or trace event.
- **Collection Mode:** How data is collected and moved to the MDW. There are two modes: 1) Cached Mode and 2) Non-Cached Mode.

# Collection Sets, Types, & Items

Collection Sets

- Disk Usage
- Query Activity
- Server Activity

Collection Types

- T-SQL Query
- Performance Counter
- Query Activity
- SQL Trace

Collection Items

- Performance Counters
- DMV Stats
- Profiler Events
- And many more

# Data Collection Sets

- By default, the Data Collector includes these three **data collection sets**, which are used to collect data from targets and put it in the MDW:
  - Disk Usage Collection Set
  - Server Activity Collection Set
  - Query Statistics Collection Set

# Disk Usage Collection Set

- Tracks the growth of database and log files and provides file-related statistics on a daily basis.
- Captures this data:
  - Snapshots of data file sizes obtained from `sys.partitions` and `sys.allocation_units`.
  - Snapshots of log file sizes obtained from `DBCC SQLPERF (LOGSPACE)`.
  - Snapshots of I/O statistics from `sys.dm_io_virtual_file_stats`.
- Quick demo of property screen

# Server Activity Collection Set

- Provides an overview of SQL Server activity, SQL Server resource utilization and SQL Server resource contention.
- Captures this data:
  - sys.dm\_os\_wait\_stats
  - sys.dm\_os\_latch\_stats
  - sys.dm\_os\_schedulers
  - sys.dm\_exec\_sessions , sys.dm\_exec\_requests, sys.dm\_os\_waiting\_tasks (using a joined query)
  - sys.dm\_os\_process\_memory
  - sys.dm\_os\_memory\_nodes
  - Additional system and SQL Server performance counters
- Quick demo of property screen

# Query Statistics Collection Set

- Gathers data about query statistics and individual query text, query plans, and specific queries.
- Captures this data:
  - `sys.dm_exec_query_stats` view
  - The text of selected batches and queries
  - The plan of selected batches and queries
  - The normalized text of selected batches
- Quick demo of property screen

# Data Collector Types

- Data Collection Types include:
  - T-SQL Query Collector Type
  - Performance Counters Collector Type
  - Query Activity Collector Type
  - SQL Trace (Profiler) Collector Type

# T-SQL Collector Type

- Literally runs most any T-SQL you want to run, and stores the results in the MDW.
- Can be used to create custom data collectors of DMVs, and much more.

# Performance Counters Collector Type

- Used to collect any Performance Monitor counter.
- You can create a custom Collector Set to collect any counters you want and store them in the MDW.

# Query Activity Collector Type

- Used to collect information on the queries running on your server, including:
  - Query text.
  - Estimated execution plan.
  - Query activity (how often it runs, number of compilations).
  - Query statistics (how many resources does it use).
- Is used by the default Query Statistics Collection Set, and cannot be used to create your own custom collection set.

# SQL Trace Collection Type

- Can be used to collect any SQL Trace (Profiler) event and store it in the MDW.
- This collection type is not used by any of the three default collection sets.
- If you want to use this type, you will have to create your own custom collection set.

# Key Components Used by the Data Collector

- **SSMS & SPs**: To access and configure the data collector.
- **Local Cache**: Used to store collected data temporarily before being moved to the MDW.
- **msdb**: Stored data collector configuration info.
- **MDW**: Relational database used to store data.
- **SSIS Packages**: Used to collect data from targets, and to move data to the MDW.
- **SQL Server Agent Jobs**: Used to run the scheduled SSIS packages. All activities are scheduled.
- **dcexec.exe Process or the Data Collector Run-time component**: Each collection set has its own process, and it runs only during the collection process. This process runs outside of sqlserver.exe.

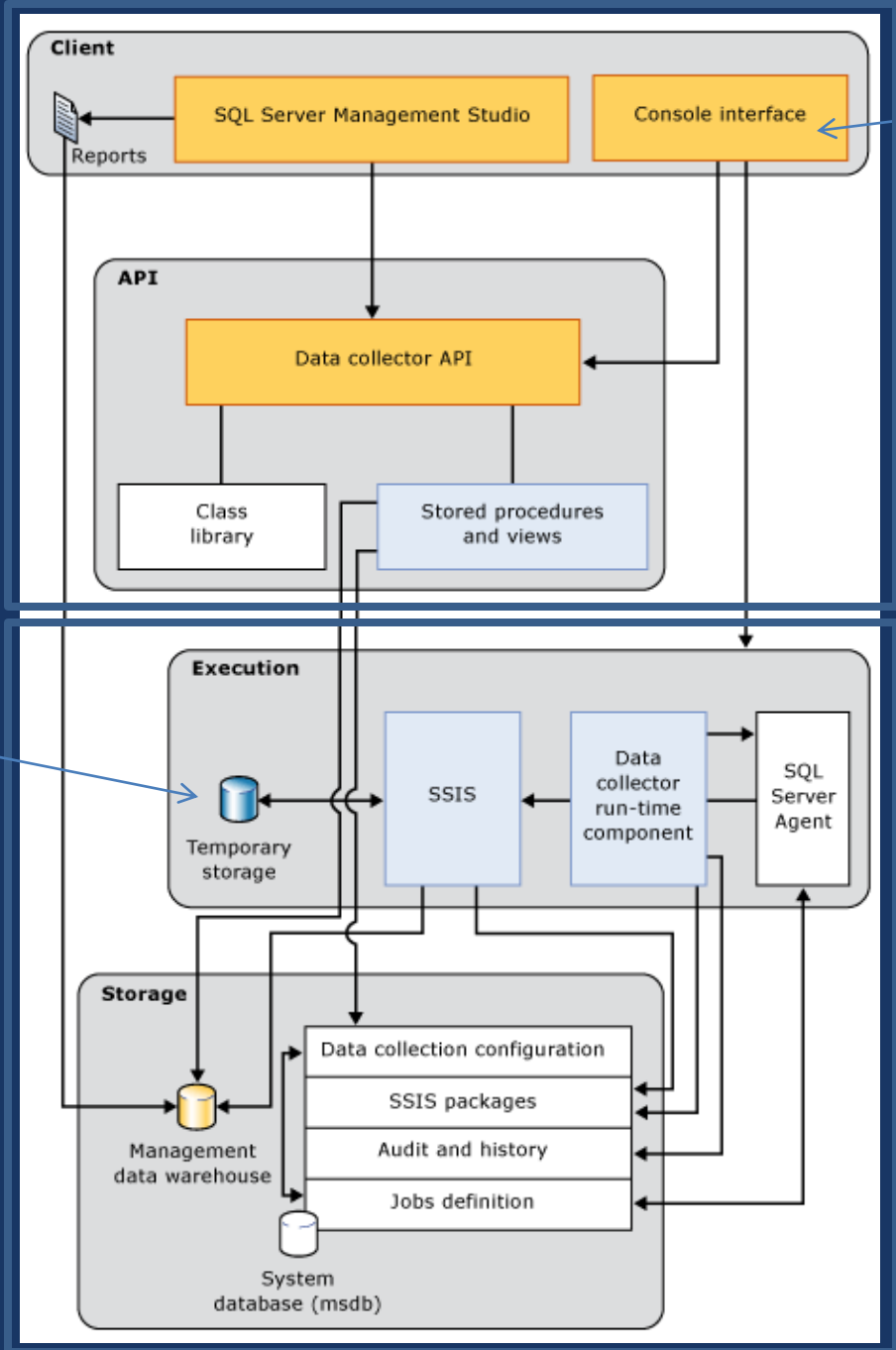
# Cached vs Non-Cached Collection Sets

- A collection set can be designated to collect and send data to the MDW in **two** different ways.
- **Cached**: Data is collected on one schedule, and stored in the local cache. Another schedule runs to move it from the local cache to the MDW. The *Server Activity* and *Query Statistics* collection sets use this option. Best for frequent data collection (more often than every five minutes) to reduce performance hit.
- **Non-Cached**: Data is collected and sent to the MDW as a single job, in a single step. Job runs continuously. The *Disk Usage* collector set uses this method. Best used when data is collected in periods of 5 minutes or greater to reduce performance hit.
- **Quick SSMS Demo**

# Data Collector Architecture

Used for setup, configuration, and reporting.

Local Cache  
Used for scheduled data collection.



PowerShell

Most work occurs here.

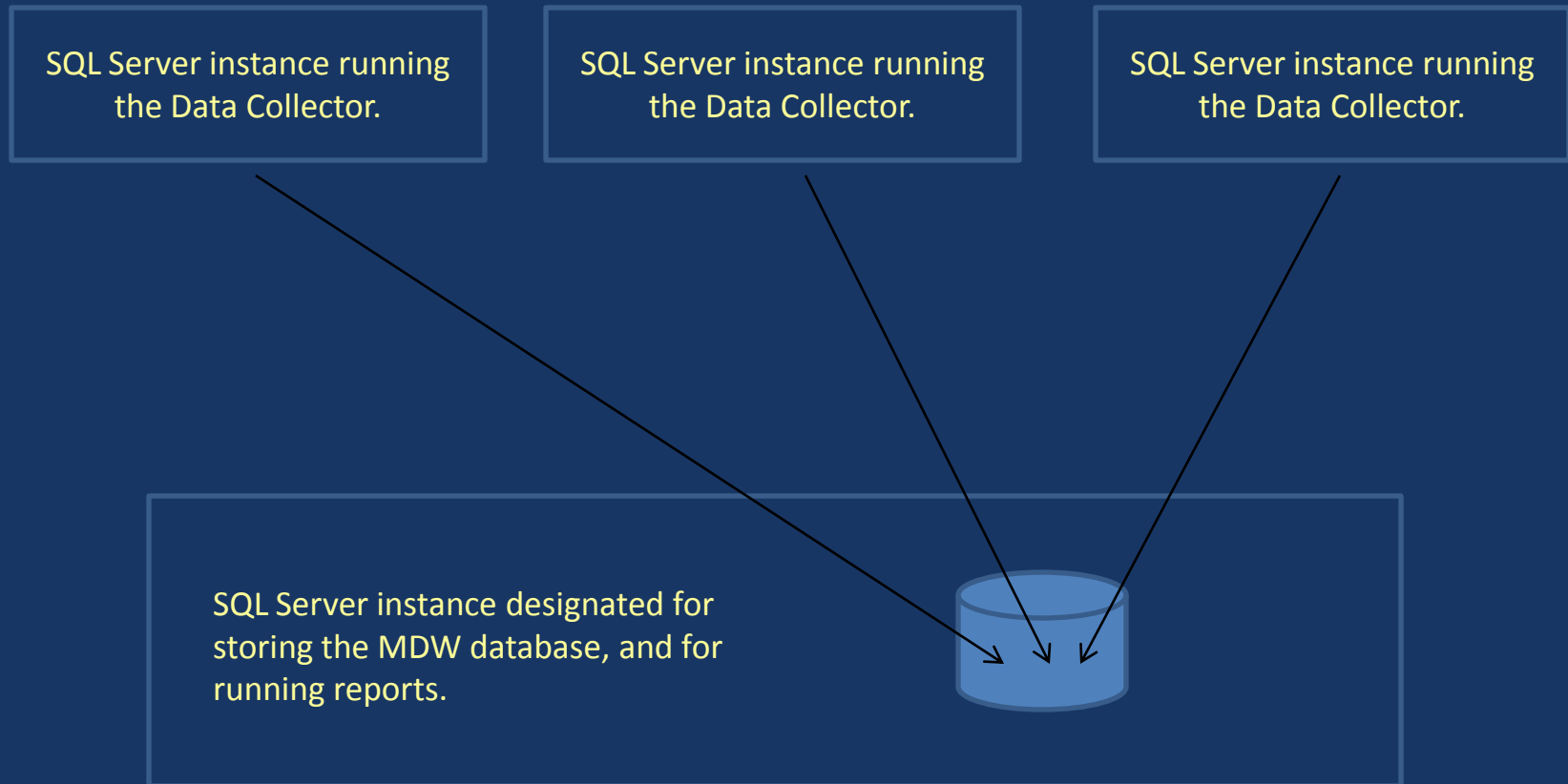
# How to Install & Configure the Data Collector

- Planning Your Installation
- Installing and Configuring the Data Collector

# Planning Your Installation

- Decide what instances you want to monitor. They must all be in the same domain.
- Decide if you will be using a separate SQL Server instance for the MDW and reporting.
- Determine where you want the temp folder to be located.
- Consider what the physical impact on your servers will be after turning on the Data Collector.

# High Level Overview of Data Collector



# Installing and Configuring the Data Collector

- Installing the Data Collector is a multistep process:
  - Create the MDW (a one time step).
  - Add the Data Collector to each SQL Server instance to be monitored, and point it to the MDW. If you want to monitor 25 servers, then you perform this step 25 times.
  - Once each instance is added, the Data Collector begins to collect data immediately.
- Demo

# How to Read & Interpret Data Collector Reports

- Three key reports (with drill-down):
  - Disk Usage Report
  - Query Statistics Report
  - Server Activity Report
- Demo reports

# Take Homes for Today

- The SQL Server 2008 Data Collector has a lot of potential for helping DBAs track performance data over time.
- To take full advantage of the tool, you will need to take some time and effort to master it, especially if you want to create your own data collection sets.
- We still don't fully understand the amount of resources needed by the Data Collector and how this will affect production SQL Server instances. Don't put into production until you have tested it.

# Q & A

- Please ask your questions clearly and loudly.
- If you don't get your questions answered now, see me after the session, or e-mail me.

# Find Out More

## Free E-Books:

[www.sqlservercentral.com/Books](http://www.sqlservercentral.com/Books)

## Check these out:

[www.SQLServerCentral.com](http://www.SQLServerCentral.com)

[www.Simple-Talk.com](http://www.Simple-Talk.com)

## Contact me at:

[bradmcgehee@hotmail.com](mailto:bradmcgehee@hotmail.com)

## Blogs:

[www.bradmcgehee.com](http://www.bradmcgehee.com)

[www.twitter.com/bradmcgehee](http://www.twitter.com/bradmcgehee)

[Click Here for a free 14-day trial of the Red Gate SQL Server Toolbelt](#)