

SQL Server 2008 und Performance Data Collection

Leistungsdaten sammeln

Data Collection ist ein Feature des SQL Server 2008. Damit sammeln und speichern Sie zentralisiert Performance-Daten aus dem SQL Server, dem SQL Profiler und dem Performance Monitor. **Fabian Schenker**

Auf einen Blick

Dieser Artikel beschreibt, wie Administratoren mit der neuen SQL-Server-2008-Funktionalität und mit T-SQL die Leistungsdaten eines Datenbanksservers überwachen und sammeln, der selbst SQL Server 2008 in der Edition Standard oder Enterprise nutzt.

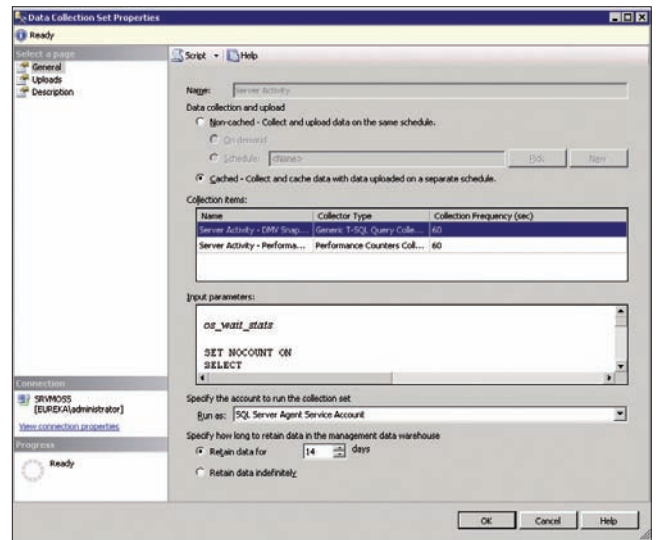
- **Plattform**
SQL Server 2008/2008 R2
- **Technik/Anwendung**
Administration, T-SQL
- **Voraussetzungen**
SQL Server 2008/2008 R2
Standard Edition
Microsoft SQL Server Management Studio
- **Autor**
Fabian Schenker ist seit 2008 bei Trivadis beschäftigt. Als zertifizierter Microsoft Database Administrator sowie System Administrator liegen seine Schwerpunkte in der Administration von Microsoft SQL Server, der Konfiguration von Microsoft SharePoint Server sowie dem Entwickeln von T-SQL-Skripten.

Performance Data Collection ist eine neue Funktionalität, die in den Standard- und Enterprise-Editionen des SQL Server 2008 verfügbar gemacht wird. Sie erlaubt das Sammeln sowie zentralisierte Speichern von (Performance-)Daten. Das Sammeln der Daten geschieht durch eine Verbindung der SQL Server Agent Jobs, Schedules sowie SSIS-Packages. Interessant ist das Feature vor allem deshalb, weil die Daten für alle SQL Server 2008 in einem zentralisierten Management Data Warehouse gespeichert werden.

Komponenten

Das Data-Collection-Feature ist aufgeteilt in mehrere Grundkomponenten:

- **Management Data Warehouse:** In der zentralisierten SQL-Server-2008-Instanz wird die Management-Data-Warehouse-Datenbank (MDW) erstellt. Diese enthält die drei Schemata *core*, *snapshots* und *custom_snapshots*. Das Schema *core* umfasst Konfigurationstabellen, Views, Stored Procedures und Functions; die anderen Schemata umfassen die gesammelten Daten.
- **Data Collection Sets:** Ein *Data Collection Set* ist eine Sammlung von *Collection Items*, welche zur gleichen Zeit, im gleichen Intervall gesammelt und abgelegt werden sollen. Microsoft stellt mit der Standard-Konfiguration des Data-Collection-Features drei sogenannte *System*



Sammlung von Collection Items im Data Collection Set (Bild 1)

Data Collection Sets zur Verfügung (Bild 1). Darüber hinaus können weitere Collection Sets bei Bedarf benutzerdefiniert in Form von *Custom Collection Sets* unter Verwendung von T-SQL-Befehlen erstellt werden.

Data Collection Sets und Data Collection Items

Die Collection Items definieren die zu sammeln den Daten und können selbst den Typ von vier verschiedenen *Collection Types* (Sammlungstypen) einnehmen (Tabelle 1).

Bei der Initialisierung der *Performance Data Collection* werden drei System Data Collection Sets erstellt:

- **Usage:** Das *Disk Usage Collection Set* verfolgt das Wachstum von Data- sowie Log-Dateien. Außerdem bietet dieses Collection Set eine datei-abhängige Statistik zum durchschnittlichen Wachstum eines Files pro Tag in Megabyte an. Das Collection Set besteht aus den zwei Collection Items *Disk Usage (Data Files)* und *Disk Usage (Log Files)*. Die Collection Items sammeln die Informationen Data-File-Größe, Log-File-Größe und I/O-Statistik.
- **Server Activity:** Das *Server Activity Collection Set* verbindet Performance-Daten aus den Kategorien CPU, Disk-I/O, Memory und Netzwerk mit SQL-Server-Aktivitäten, um Bottlenecks aufzuzeigen. Es sammelt Ressourcen und Performance-Daten über den SQL Server.

Tabelle 1: Collection Item Types

Collector Type	Beschreibung
Generic T-SQL Query	Durch diesen Collector Type werden Ergebnisse einer T-SQL-Abfrage gespeichert.
Generic SQL Trace	Verschiedene SQL-Server-Profiler-Elemente werden hier gespeichert.
Query Activity	Bei diesem Collector Type handelt es sich um einen Microsoft-eigenen Collector Type, der für eigene Collection Items nicht verwendet werden sollte. Er sammelt Daten aus bestimmten Dynamic Management Views wie zum Beispiel sys.dm_exec_requests.
Performance Counter	Collection Items mit diesem Collector Type können Daten aus Performance Counters sammeln.

■ **Statistics:** Dieses Collection Set sammelt Query-Statistiken und Informationen zur Ausführung von Abfragen.

Neben den vom System erstellten Performance Data Collection Sets können auch benutzerdefinierte *Custom Data Collection Sets* erstellt werden, die ihrerseits die Collector Types *Generic T-SQL Query Collector Type*, *Generic SQL Trace Collector Type* oder *Performance Counter Collector Type* verwenden. Dabei wird in XML definiert, was abgefragt wird und was die Ausgabe ist. Wie genau ein Custom Data Collection Set erstellt wird, wird nachfolgend gesondert behandelt.

Data Collection Sets erstellen

Microsoft SQL Server 2008 bietet ausschließlich die Möglichkeit, Custom Data Collection Sets via Skript zu erstellen und nicht via Wizard. Nachdem das Management Data Warehouse angelegt und das *Data Collector Feature* aktiviert wurde, können Custom Data Collection Sets erstellt werden. Mit dem hier exemplarisch erzeugten Performance Collection Set soll ein Überblick über die Prozessor- sowie Arbeitsspeicherauslastung eines Servers zurückgeliefert werden. Daher sind folgende Performance-Monitor-Metrics an dieser Stelle relevant:

■ **CPU:** Zur Überwachung des Prozessors werden die Leistungsindikatoren *Processor/% Processor Time*, *System/Processor Queue Length* und *Processor/% Privileged Time* berücksichtigt.

■ **RAM:** Zur Überwachung des Arbeitsspeichers kommen die Leistungsindikatoren *Memory/Pages/sec*, *Memory/Available Mbytes* und *Process/Page Faults/sec* zum Einsatz.

Um ein eigenes Custom Data Collection Set anzulegen, erstellen Sie zunächst das neue Performance Data Collection Set. Anschließend erzeugen Sie die Performance Data Collection Items und weisen diese dem Data Collection Set zu.

Listing 1: T-SQL-Skript: Neues Performance Collection Set

```

- Die Stored Procedure befindet sich in der msdb
USE msdb
GO
- Deklarieren von zwei OUTPUT-Variablen
DECLARE @collection_set_uid uniqueidentifier
DECLARE @collection_set_id int
- Generieren einer neuen GUID
SET @collection_set_uid = NEWID()
- Überprüfen, ob der Scheduler vorhanden ist mit 15 Minuten
SELECT COUNT(*)
  FROM msdb.dbo.sysschedules
 WHERE name = 'CollectorSchedule_Every_15min'
- Ausführen der Stored Procedure
EXEC dbo.sp_syscollector_create_collection_set
  @name          = N'Capacity',
  @days_until_expiration = 120,
  @schedule_name  = N'CollectorSchedule_Every_15min',
  @description    = N'Capacity Management Collection Set',
  @collection_set_id = @collection_set_id OUTPUT,
  @collection_set_uid = @collection_set_uid OUTPUT

- ID sowie GUID auslesen
SELECT @collection_set_id, @collection_set_uid
- Überprüfen, ob das Collection Set erstellt wurde
SELECT *
  FROM syscollector_collection_sets
 WHERE collection_set_uid = @collection_set_uid

```

gen Sie die Performance Data Collection Items und weisen diese dem Data Collection Set zu. Das Erstellen von Custom Performance Data Collection Sets geschieht über Stored Procedures. Zur Anlage des Data Collection Set verwenden Sie die Stored Procedure *sp_syscollector_create_collection_set*. Die zu definierenden Werte für das neue Data Collection Set sind **Tabelle 2** zu entnehmen. ▶

Tabelle 2: Custom-Performance-Collection-Set-Parameter

Parameter	Beschreibung	Wert
@name	Collection-Set-Name	Capacity
@target	Ziel (wird nicht gebraucht)	-
@collection_mode	0 = Cached (Default/Standard) 1 = Non-cached	- 120
@days_until_expiration	Retentiontime Default/Standard = 720 days	-
@proxy_id	SQL Server Agent Proxy Account	-
@proxy_name	SQL Server Agent Proxy Account	-
@schedule_uid	Scheduler-GUID (die sysschedules-System-Tabelle enthält die Informationen)	-
@schedule_name	Scheduler-Name (entweder uid oder Name angeben!)	CollectorSchedule_Every_15min
@logging_level	Logging Level Default/Standard = 1	-
@description	Beschreibung	Capacity Management Collection Set
@collection_set_id	OUTPUT (id)	@collection_set_id
@collection_set_uid	OUTPUT (uniqueidentifier)	@collection_set_uid

Alles NT: Systemmonitor und Leistungsindikatoren

Leistungsindikatoren (Performance Counter) werden von NT-basierten Windows-Systemen angeboten. Sie lassen sich mit dem Systemmonitor überwachen. Der Systemmonitor ist ein Snap-in für die Microsoft Management Console (MMC); er unterstützt Sie dabei zu erkennen, ob geänderte Systemeigenschaften positiven Einfluss auf das Leistungsverhalten eines Windows-Rechners oder -Servers haben oder ob es Schwachstellen im Bereich der Performance (Leistung) gibt.

Anzeige von Leistungsreserven oder Überlastung

Über den Systemmonitor erhalten Sie umfassende Informationen darüber, ob Ihr Rechner Leistungsreserven hat oder gegebenenfalls eine Aufrüstung im Bereich Hardware sinnvoll ist. Um Veränderungen im Leistungsverhalten beurteilen zu können, rufen Sie die gewünschten Leistungskennwerte über den Systemmonitor vor und nach der Änderung von Eigenschaftswerten ab und setzen sie in direkten Vergleich.

Sie starten das Programm zur Leistungsüberwachung je nach Windows-System über *Start/Verwaltung/Leistung* oder *Start/Verwaltung /Leistungsüberwachung*. Um die zu überwachenden Informationen zu bestimmen, rufen Sie zum grafischen Ausgabebereich des Systemmonitors den Kontextmenübefehl *Leistungsindikatoren hinzufügen* auf. Wählen Sie anschließend, ob Sie lokale oder Leistungsindikatoren eines bestimmten Rechners im Netzwerk überwachen wollen. Wählen Sie dann im Dialogfenster *Leistungsindikatoren hinzufügen* ein Leistungsobjekt (z. B. DNS, Logischer Datenträger, Webdienst, Auslagerungsdatei, Indexdienst, Netzwerkschnittstelle, Prozessor, Speicher, System, TCPv4, TCPv6) an. Haben Sie den SQL Server eingerichtet, finden Sie hier auch Leistungsindikatoren für den Datenbankservers. Legen Sie nach Bedarf fest,

ob Sie nur einen oder sämtliche Leistungsindikatoren eines Objekts überwachen wollen. Mit *Hinzufügen* nehmen Sie die gewünschte Verfolgung des Leistungsindikators in den Systemmonitor auf. Um nähere Informationen zu den verfügbaren Leistungsindikatoren zu erhalten, müssen Sie lediglich je nach Windows-Version im Dialog die Schaltfläche *Erklärung* anwählen oder das Kontrollfeld *Beschreibung anzeigen* markieren. Auf Wunsch stellen Sie mehrere Indikatoren unterschiedlicher Objekte nach Bedarf zusammen.

Die überwachten Werte werden in regelmäßigen Abständen aktualisiert. Anhand der Ergebnisse erkennen Sie beispielsweise, ob ein Rechner im Netzwerkbetrieb überbeansprucht ist und ob ausreichend Rechenzeit für die Arbeit mit Anwendungen zur Verfügung steht. Ist der verfügbare Speicher dauerhaft zu niedrig, die Speicherauslastung permanent zu hoch, ist das ein Zeichen dafür, dass entweder der Arbeitsspeicher des Rechners aufgestockt oder aber weniger Programme gleichzeitig ausgeführt werden sollten. Sind die physikalischen und logischen Laufwerke permanent überbeansprucht, sollten Sie den Rechner entweder um eine schnellere oder um mehrere zusätzliche Festplatten erweitern.

Kritische Werte im Blick

Kritische Werte werden besonders bei der grafischen Anzeige erkennbar. Immer dann, wenn Balken in der Maximalgröße aufgezeigt werden, sind die zugehörigen Ressourcen erschöpft. Je mehr Leistungsindikatoren sich im roten Bereich befinden, umso leistungsschwächer und instabiler ist das System. Sind Leistungsindikatoren nur kurzfristig und nicht dauerhaft maximal ausgelastet, besteht hingegen kein Grund zur Sorge. Kritisch sind lediglich die Ressourcen, die längere Zeit oder dauerhaft überbeansprucht sind. **[am]**

Tabelle 3: Custom-Performance-Collection-Item-Parameter

Parameter	Beschreibung	Wert
@collection_set_id	Collection Set ID	@collection_set_id
@collector_type_uid	GUID für den Collector Type	@collector_type_uid
@name	Name des Collection Item	Perfmon
@frequency	Frequenz der Abfrage, Default = 5	60
@parameters	XML mit dem Schema des Collector Type	XML nach Schema
@collection_item_id	OUTPUT ID für das Item	@collection_item_id

Tabelle 5: Die Tabellen zur Sammlung der Leistungskennwerte

Tabellenname (inkl. Schema)	Funktion
Core.snapshots_internal	Bei jedem Upload wird hier eine neue Zeile geschrieben
Snapshots.performance_counter_instances	Enthält Informationen zu den Performance Counters (Object Name, Counter Name, Instance ...)
Snapshots.performance_counter_values	Enthält die Werte der einzelnen Counter zu den jeweiligen Snapshots

Daraus ergibt sich das Skript für das neue Collection Set (**Listing 1**).

Als Nächstes müssen Sie die Collection Items, die die Daten von Prozessor und RAM sammeln, zum Collection Set hinzufügen. Auch hierfür bietet Microsoft eine gesonderte Stored Procedure an, die auch hier genutzt wird. Die Parameter für diese Stored Procedure *sp_syscollector_create_collection_item* sind **Tabelle 3** zu entnehmen.

Die Werte für die Parameter sind bis auf *@parameters* ohne Probleme festzulegen. *@parameters* ist wie erwähnt im XML-Format gemäß **Tabelle 4** anzugeben. Mithilfe des Schemas aus **Tabelle 4** wird auch der letzte Parameter *@parameters* bestimmt und das T-SQL-Skript generiert (**Listing 2**).

Identifizieren und auswerten

Nachdem das Skript erfolgreich ausgeführt wurde, kann das neue Collection Set nun gestartet werden.

Das neue Collection Set mit dem zugehörigen Collection Item ist damit erstellt und sammelt Daten. Nun müssen die Daten im Management Data Warehouse identifiziert werden, um anschließend ausgewertet zu werden. Die in **Tabelle 5** aufgelisteten Tabellen im Management Data Warehouse enthalten gesammelte Daten. Wie der Tabelle zu entnehmen ist, beinhalten die Tabellen *snapshots.performance_counter_values* sowie *snapshots.perormance_counter_instances* Daten aus dem neuen Data Collection Set.

Nachdem die Performance-Daten identifiziert wurden, können diese ausgewertet werden. Hierzu bieten sich Microsoft SQL Server 2008 Reporting Services an, um einen Report zu erstellen, welcher die Daten darstellt und der auch von Drittpersonen verwendet werden kann, die mit SQL Server nicht vertraut sind. Für die System Data Collection Sets sind bereits Reports eingerichtet (**Bild 1**). **[am]**

Tabelle 4: XML-Schema Collector Type (Auszug)

Performance Counters Collector Type	<pre><xs:schema xmlns:xs="http://.../2001/XMLSchema" targetNamespace="DataCollectorType"> <xs:element name="PerformanceCountersCollector"> <xs:complexType> <xs:sequence> <xs:element minOccurs="0" maxOccurs= "unbounded" name="PerformanceCounters"> <xs:complexType> <xs:attribute name="Objects" type="xs:string" use="required" /> ... </xs:complexType> </xs:element> </xs:sequence> ... </xs:complexType> </xs:element> </xs:schema></pre>
---	---

Listing 2: T-SQL-Skript: Neues Performance Collection Item (Auszug)

```
- Die Stored Procedure befindet sich in der msdb
USE msdb
GO
- Deklarieren der Variablen und Auslesen der Collection Set ID unseres Collection Set
DECLARE @collection_item_id int
DECLARE @collection_set_id int
DECLARE @collector_type_uid uniqueidentifier
DECLARE @parameters xml
SET @collection_set_id = (SELECT collection_set_id
  FROM syscollector_collection_sets
  WHERE name = N'Capacity')
- Auslesen der Collector Type GUID unseres Collector Type
SET @collector_type_uid = (SELECT collector_type_uid
  FROM syscollector_collector_types
  WHERE name = N'Performance Counters Collector Type')
- Erstellen unseres Parameters nach dem XML Schema
SET @parameters = CONVERT(xml,N'
<ns:PerformanceCountersCollector xmlns:ns="DataCollectorType">
  <PerformanceCounters Objects="Processor" Counters="% Processor Time" Instances="*" />
  <PerformanceCounters Objects="Processor" Counters="% Privileged Time" Instances="*" />
  ...
</ns:PerformanceCountersCollector>
')
- Erstellen des Collection Item und Überprüfung, ob das Collection Item erstellt wurde
EXEC sp_syscollector_create_collection_item
  @collection_set_id = @collection_set_id,
  @collector_type_uid = @collector_type_uid,
  @name = N'Perfmon',
  @frequency = 60,
  @parameters = @parameters,
  @collection_item_id = @collection_item_id OUTPUT
SELECT *
  FROM syscollector_collection_items
  WHERE name = N'Perfmon'
- Nach dem nächsten Upload sollten die Counter erscheinen (MDW mit Name des Management Data Warehouse ersetzen)
SELECT *
  FROM MDW.snapshots.performance_counters
```