

Willkommen zum „IBM DB2 Newsletter“

Liebe Leserinnen und Leser,

Die IBM Information on Demand Konferenz in Berlin ist seit einer Woche vorbei. Dazu gibt es einen kurzen Rückblick. Aber schon steht die nächste Tagung vor der Tür, dazu mehr im Abschnitt "Schulungen/Tagungen".

Die erste Hälfte des Jahres und somit das zweite Quartal geht zu Ende, die Urlaubszeit beginnt. Dafür wünschen wir Ihnen bestes Wetter.

Wir haben auch in dieser Ausgabe wieder versucht, Artikel zu interessanten Themen zu schreiben. Beim Lesen und Ausprobieren wünschen wir Ihnen viel Spaß.

Für Fragen und Anregungen unsere Kontaktadresse: db2news@de.ibm.com.

Ihr TechTeam

Inhaltsverzeichnis

IOD 2009 RÜCKBLICK.....	1
DB2 MONITORING MIT WLM (TEIL 2).....	2
WLM STATISTIKDATEN MONITORING	2
MONITORING INDIVIDUELLER AKTIVITÄTEN.....	2
VERWALTEN DER WLM MONITOR DATEN.....	6
ERZEUGEN HISTORISCHER REPORTS	7
DB2 AUDIT IN DB2 V9.5 (TEIL 1 - KONFIGURATION).....	7
AUDIT AUF INSTANZEBENE	7
AUDIT AUF DATENBANKEBENE.....	9
SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG.....	10
DB2 AKTUELL 2009.....	10
CHATS MIT DEM LABOR.....	11
NEWSLETTER ARCHIV.....	11
ANMELDUNG/ABMELDUNG.....	11
DIE AUTOREN DIESER AUSGABE:.....	11
A SMARTER PLANET:	12

IOD 2009 Rückblick

Letzte Woche fand in Berlin die Information on Demand Konferenz der IBM statt. Sie hatte ihren Schwerpunkt im Bereich Information Management und Mitarbeiter aus den IBM Labors berichteten über aktuelle Trends und neue Produkte. Am interessantesten waren jedoch die Präsentationen von Kunden und Partnern, die über ihre Erfahrungen beim Einsatz der IBM Produkte und Lösungen referierten. Nähere Informationen über die Inhalte der Konferenz können Sie im [IBM Blog](#) finden. Auch die nächste Version von DB2 war ebenfalls ein großes Thema auf der IOD. In

mehreren Sessions wurden die neuen Möglichkeiten von DB2 9.7 erörtert. Einfachere Verwaltung, höhere Komprimierungsraten und die Kompatibilität mit Funktionen anderer Hersteller waren die Hauptthemen. In den nächsten Ausgaben des DB2 Newsletters werden wir uns den einzelnen Verbesserungen von DB2 9.7 im Detail widmen.

DB2 Monitoring mit WLM (Teil 2)

Nach dem wir uns in der letzten Ausgabe ([DB2 NL - Ausgabe April](#)) mit dem Aufsetzen der nichtlizenzpflichtigen WLM Funktionen beschäftigt haben, wollen wir in dieser Ausgabe auf einige Beispiele zur Analyse und Auswertung, der durch diese Monitore gesammelten Daten, vorstellen.

WLM Statistikdaten Monitoring

In der letzten Aufgabe wurde ein Ereignismonitor für Statistikdaten `WLM_STATS` angelegt, und die Histogramm Darstellung wurde vorgestellt. Weitere WLM gesammelten Statistikdaten können durch Betrachtung der `WLM_STATS_sc` Tabelle analysiert werden.

Mit nachfolgender Abfrage wird der Durchsatz an Aktivitäten im System, Maximum an Connections, die durchschnittliche Laufzeit und maximale Kosten von Aktivitäten pro Tag verglichen:

```
select  date(STATISTICS_TIMESTAMP) DATUM,
        sum(COORD_ACT_COMPLETED_TOTAL) SUM_COMPLETED,
        max(CONCURRENT_CONNECTION_TOP) CONCURR_CONN,
        max(COORD_ACT_LIFETIME_TOP/1000) COORD_ACT_LTIME,
        max(ROWS_RETURNED_TOP) ROWS_RET,
        max(COST_ESTIMATE_TOP) COST_EST,
        substr(SERVICE_SUBCLASS_NAME,1,20) SERVICE_SUBCLASS,
  from  WLM_STATS_SC
 group by date(STATISTICS_TIMESTAMP), SERVICE_SUPERCLASS_NAME
 order by date(STATISTICS_TIMESTAMP), SERVICE_SUPERCLASS_NAME
```

DATUM	SUM_COMPLETED	CONCURR_CONN	COORD_ACT_LIME	ROWS_RET	COST_EST	SERVICE_SUPERCLASS
05/14/2009	0	0	0	-1	-1	SYSDEFAULTMAINTENANCECLA
05/14/2009	0	6	0	-1	-1	SYSDEFAULTSYSTEMCLASS
05/14/2009	28393	200	534	14206	482632	SYSDEFAULTUSERCLASS
05/15/2009	0	0	0	-1	-1	SYSDEFAULTMAINTENANCECLA
05/15/2009	0	6	0	-1	-1	SYSDEFAULTSYSTEMCLASS
05/15/2009	6422	100	215	14190	482632	SYSDEFAULTUSERCLASS

Monitoring individueller Aktivitäten

Die durch den WLM Aktivitäts-Ereignis-Monitor gesammelten Daten können durch Betrachtung der `WLM_EVENT` und `WLM_EVENT_STMT` Tabellen analysiert werden.

Um Informationen zu SQL-Statements und Abfragezeiten zu sammeln, muß der Statement-Monitor aktiviert werden.

```
$db2 attach to db2inst
$db2 UPDATE DBM CFG USING DFT_MON_STMT ON
```

Im Nachfolgenden wollen wir die Abfragen und Screenshots von 3 Szenarien betrachten:

Herausfinden der 10 Abfragen mit der längsten Ausführungszeiten

Folgende Abfrage ermitteln die SQL-Abfragen auf Partition 0 mit den längsten Laufzeiten:

```
SELECT SUBSTR(CHAR(AGENT_ID),1,10) AS AGENT_ID,
       SUBSTR(APPL_NAME, 1,15) AS APPL_NAME,
       SUBSTR(ACTIVITY_TYPE,1,10) AS TYPE,
       TIMESTAMPDIFF(2, CHAR(TIME_COMPLETED-TIME_STARTED)) AS TOTALTIME,
       CASE WHEN SQLCODE = 0 THEN 'NORMAL' ELSE CHAR(SQLCODE) END AS RESULT,
       SUBSTR(SESSION_AUTH_ID,1,8) AS USER,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,20) AS SERVICE_SUBCLASS_NAME
FROM WLM_EVENT AS A
WHERE PARTITION_NUMBER=0
ORDER BY TOTALTIME DESC
FETCH FIRST 10 ROWS ONLY
```

Die Ausgabe könnte wie folgt aussehen:

AGENT_ID	APPL_NAME	TYPE	TOTALTIME	RESULT	USER	SERVICE_SUBCLASS_NAME
65888	db2bp	READ_DML	310	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65937	db2batch	READ_DML	213	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65917	db2batch	READ_DML	12	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65897	db2batch	READ_DML	12	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65913	db2batch	READ_DML	11	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65929	db2batch	READ_DML	10	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65909	db2batch	READ_DML	10	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65927	db2batch	READ_DML	10	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65907	db2batch	READ_DML	10	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS
65901	db2batch	READ_DML	9	NORMAL	PERFPOL1	SYSDEFAULTSUBCLASS

10 record(s) selected.

Herausfinden des SQL-Statements für die Abfrage mit der längsten Laufzeit

Mit der vorherigen Abfrage, haben wir die AGENT_ID ermittelt. Mit Hilfe dieser ID, kann das entsprechende SQL-Statement ermittelt werden. Dazu kann folgendes SQL-Statement verwendet werden:

```
SELECT SUBSTR(CHAR(AGENT_ID),1,10) AS AGENT_ID,
       TIMESTAMPDIFF(2, CHAR(TIME_COMPLETED-TIME_STARTED)) AS TOTALTIME,
       SUBSTR(S.STMT_TEXT, 1,1000) AS STMT
FROM WLM_EVENT AS A,
     WLM_EVENT_STMT AS S
WHERE A.APPL_ID = S.APPL_ID
     AND A.ACTIVITY_ID = S.ACTIVITY_ID
     AND A.UOW_ID = S.UOW_ID
     AND A.AGENT_ID=65888
     AND A.ACTIVITY_ID=11;
```

Die Ausgabe dazu könnte wie folgt aussehen:

AGENT_ID	TOTALTIME	STMT
65888	281	SELECT L_RETURNFLAG, L_LINESTATUS, SUM(L_QUANTITY) AS SUM_QTY, SUM(L_EXTENDEDPRI CE) AS SUM_BASE_PRICE, SUM(L_EXTENDEDPRI CE * (1 - L_DISCOUNT)) AS SUM_DISC_PRICE, SUM(L_EXTENDEDPRI CE * (1 - L_DISCOUNT) * (1 + L_TAX)) AS SUM_CHARGE, AVG(L_QUANTITY) AS AVG_QTY, AVG(L_EXTENDEDPRI CE) AS AVG_PRICE, AVG(L_DISCOUNT) AS AVG_DISC, COUNT BIG(*) AS COUNT_ORDER FROM TPCD.LINEITEM WHERE L_SHIPDATE <= DATE ('1998-12-01') - 90 DAY GROUP BY L_RETURNFLAG, L_LINESTATUS ORDER BY L_RETURNFLAG, L_LINESTATUS

1 record(s) selected.

Monitoren der Statements aller Langläufer-Abfragen für Performance-Zwecke

Mit nachfolgendem SQL-Statement werden alle SQL-Statements angezeigt, die gegen die Datenbank abgesetzt wurden und deren Laufzeit größer als 5 Minuten war, zusammen mit der Häufigkeit der Ausführung. Die Ausgabe erfolgt im DB2 Advisor kompatiblen Format:

```
$ db2 -x "SELECT '--#SET FREQUENCY ' || CHAR(COUNT(*)) || CHR(10) ||
          CHR(13), CHAR(RTRIM(S.STMT_TEXT)) || ';'
FROM WLM_EVENT AS A, WLM_EVENT_STMT AS S
WHERE A.APPL_ID = S.APPL_ID
      AND A.ACTIVITY_ID = S.ACTIVITY_ID
      AND A.UOW_ID = S.UOW_ID
      AND TIMESTAMPDIFF(2, CHAR(A.TIME_COMPLETED-A.TIME_STARTED)) > 300
GROUP BY CHAR(RTRIM(S.STMT_TEXT)) " > workload.sql
```

Das erzeugte workload.sql File kann für Performance Tuning verwendet werden. Zum Beispiel können mittels dem Design Advisor empfohlene Indizes, MDC und Verteilungs-Schlüssel ermittelt werden.

```
db2advis -d sampleDB -m ICP -i workload.sql
```

Monitoren der CPU Benutzung einzelner Abfragen

Nachfolgende Schritte beschreiben, wie die Systemnutzung für den Workload eines Tages gemessen werden kann.

Als ersten überprüfen wir einzelne Aktivitäten für folgende Abfrage:

```
SELECT SQLCODE AS SQLC, UOW_ID AS UOW, ACTIVITY_ID AS ACT_ID,
       PARTITION_NUMBER AS PART,
       (USER_CPU_TIME+SYSTEM_CPU_TIME) AS TCPU,
       ROWS_MODIFIED AS ROWM, ROWS_FETCHED AS ROWF,
       TOTAL_SORT_TIME AS TTSORT,
       TIME_STARTED, TIME_COMPLETED
FROM WLM_EVENT
WHERE DATE(TIME_STARTED) = CURRENT DATE
      AND PARTIAL_RECORD =0
```

Hinweis: Durch "PARTIAL_RECORD=0" werden nur Abfragen angezeigt, die bereits beendet sind.

Die Ausgabe sieht wie folgt aus:

SQLC	UOW	ACT_ID	PART	TCPU	ROWM	ROWF	TSORT	TIME_STARTED	TIME_COMPLETED
0	1	2	1	0	-1	-1	-1	2009-04-11-15.36.50.542760	2009-04-11-15.36.50.545551
...									

In der obigen Ausgabe sind u.a. folgende Informationen enthalten, wie ACTIVITY_ID, UOW_ID, Zeitstempel für Start und Ende der Abfrage. Wie zu erkennen ist, sind die Informationen aus ROWS_FETCHED, ROWS_MODIFIED, und TOTAL_SORT_TIME nicht aussagekräftig. Damit darin Werte enthalten sind, müssen die Monitor-Switches für Statement und Sort eingeschaltet werden. (Den daraus resultierende Overhead beschreiben wir später):

```
$db2 attach to db2inst
$db2 UPDATE DBM CFG USING DFT_MON_STMT ON
$db2 UPDATE DBM CFG USING DFT_MON_SORT ON
```

Hinweis: Auf Linux-Plattformen muß zusätzlich auch noch folgende Registry-Variable zur Anzeige der CPU Benutzung aktiviert werden :

```
db2set DB2_SYSTEM_MONITOR_SETTINGS=OLD_CPU_USAGE:TRUE;DISABLE_CPU_USAGE:FALSE
```

"OLD_CPU_USAGE:TRUE" ermöglicht sowohl das sammeln von System als auch

User Zeit, erzeugt aber u.U. Overhead. Daher ist die Registry Variable nur zu setzen, wenn notwendig. Zusätzliche Informationen dazu finden Sie [hier](#).

Nach nochmaliger Ausführung der obigen Abfrage, sind nun die Werte für gelesenen Datasets (ROWS_FETCHED) und verwendete CPU und Sortierzeit gesetzt.

```
SQLC UOW ACT_ID PART TCPU ROWM ROWF TSORT TIME_STARTED TIME_COMPLETED
-----
0      2      2      1 70000 0      400591 0      2009-06-04-15.39.50.998717 2009-06-04-15.40.05.184900
...
1 record(s) selected....
```

Für die Daten auf allen Partitionen sieht dies dann wie folgt aus:

```
SQLC UOW ACT_ID PART TCPU ROWM ROWF TSORT TIME_STARTED TIME_COMPLETED
-----
0      2      2      0 10000 7      7      0      2009-06-04-15.39.50.591698 2009-06-04-15.40.01.594172
0      2      2      1 70000 0      400591 0      2009-06-04-15.39.50.998717 2009-06-04-15.40.05.184900
0      2      2      2 50000 0      399411 0      2009-06-04-15.39.50.999316 2009-06-04-15.40.05.177098

3 record(s) selected.
```

Um die Auswertung nach CPU Zeit zu sortieren kann folgendes SQL-Statement verwendet werden. Dabei wird die Ausgabe um die AGENT_ID erweitert, um später die SQL-Statements für die Abfragen zu ermitteln:

```
SELECT AGENT_ID, ACTIVITY_ID, UOW_ID,
SUM(SORT_OVERFLOWS) AS SORT_OVERFLOW, SUM(TTLSORT) AS SORT_TIME, SUM(TTLCPU) AS CPU_TIME,
TIMESTAMPDIFF(2, CHAR( MAX(TIME_COMPLETED)-MIN(TIME_STARTED))) AS EXE_TIME
FROM (
SELECT AGENT_ID, A.ACTIVITY_ID AS ACTIVITY_ID, A.UOW_ID AS UOW_ID,
PARTITION_NUMBER AS PART_NUM,
(USER_CPU_TIME+SYSTEM_CPU_TIME)/1000000 AS TTLCPU,
TOTAL_SORT_TIME/1000000 AS TTLSORT,
SORT_OVERFLOWS, TIME_COMPLETED, TIME_STARTED
FROM WLM_EVENT AS A
WHERE DATE(TIME_STARTED) = CURRENT DATE
AND PARTIAL_RECORD =0
)
GROUP BY AGENT_ID, ACTIVITY_ID, UOW_ID
ORDER BY CPU_TIME DESC
FETCH FIRST 20 ROWS ONLY;
```

Die Ausgabe kann wie folgt aussehen:

```
AGENT_ID ACTIVITY_ID UOW_ID SORT_OVERFLOWS SORT_TIME CPU_TIME EXE_TIME
-----
65625      2          1          2          1          308          209
65605      1          1          1          0          137          223
65623      3          1          0          0           0           0

3 record(s) selected.
```

Monitor Temp Table Activities

db2top and *db2pd* ermöglichen das Monitoren der SMS and DMS Tempespace Aktivitäten. Jedoch liefern diese Tools die Informationen auf Systemlevel und nicht auf Basis der Abfrage Aktivitäten. DB2 WLM monitored die Benutzung des Tempspaces auf einem feineren Level.

Mit der DB2 WLM Lizenz kann man den Tempespace Schwellwert (Threshold) setzen und die Abfragen monitoren, die den Schwellwert verletzen.

In diesem Szenario betrachten wir nur lesende Abfragen, die den größten TempSPACE brauchten (ohne einen TempSPACE Schwellwert kann WLM nicht mitteilen, ob DB2 temporäre oder echte Tabellen geschrieben hat. Für nur-Lese Abfragen würden alle Schreib-Aktivitäten schreiben in temporäre Tabellen bedeuten):

- Setzen der Event Monitore wie in der letzten Ausgabe beschrieben.
- Mit folgendem SQL-Statement werden SQL-Abfragen ermittelt:

```
SELECT SUBSTR(APPL_NAME,1,10) APPNAME,
       SUBSTR(A.AGENT_ID,1,15) AS AGENT_ID,
       A.ACTIVITY_ID, A.UOW_ID, A.PARTITION_NUMBER AS PARTITION,
       ROWS_MODIFIED AS TEMP_ROWS,
       SUBSTR(S.STMT_TEXT, 1, 1000) AS STMT
FROM WLM_EVENT AS A, WLM_EVENT_STMT AS S
WHERE A.APPL_ID = S.APPL_ID
      AND A.ACTIVITY_ID = S.ACTIVITY_ID
      AND A.UOW_ID = S.UOW_ID
      AND WORKLOADID = WORKLOAD_ID
      AND (A.ACTIVITY_TYPE='OTHER' OR A.ACTIVITY_TYPE = 'READ_DML')
ORDER BY ROWS_MODIFIED DESC;
```

Die Ausgabe sieht wie folgt aus:

```
APPNAME AGENT_ID ACTIVITY_ID UOW_ID PARTITION TEMP_ROWS STMT
-----
db2bp      65937          1      1          2 60001417 select c_name, c_custkey, o_orderkey,
o_orderdate, o_totalprice, sum(l_quantity) from tpcd.customer, tpcd.orders, tpcd.lineitem where
o_orderkey in (select l_orderkey from tpcd.lineitem group by l_orderkey having sum(l_quantity) >
300) and c_custkey = o_custkey and o_orderkey = l_orderkey group by c_name, c_custkey, o_orderkey,
o_orderdate, o_totalprice order by o_totalprice desc, o_orderdate fetch first 100 rows only
```

Verwalten der WLM Monitor Daten

Wenn Sie die Workload-Management- Ereignismonitore verwenden, müssen Sie Ihre Ereignismonitordateien oder -tabellen regelmäßig bereinigen. Der Ereignismonitor bereinigt die erfassten Daten nicht automatisch, sodass Ihre Dateien oder Tabellen im Lauf der Zeit durch die automatische Erfassung gefüllt werden.

```
delete from wlm_event where time_completed
< (current timestamp - 10 days);

delete from wlm_event_stmt where ACTIVATE_TIMESTAMP
< (current timestamp - 10 days);

delete from wlm_event_vals where ACTIVATE_TIMESTAMP
< (current timestamp - 10 days);

delete from wlm_event_control where MESSAGE_TIME
< (current timestamp - 10 days);

delete from WLM_STATS_SC where STATISTICS_TIMESTAMP
< (current timestamp - 10 days);

delete from wlm_stats_WC where STATISTICS_TIMESTAMP
< (current timestamp - 10 days);

delete from wlm_stats_WL where STATISTICS_TIMESTAMP
< (current timestamp - 10 days);

delete from wlm_stats_q where STATISTICS_TIMESTAMP
< (current timestamp - 10 days);

delete from Wlm_stats_HISTOGRAM where STATISTICS_TIMESTAMP
< (current timestamp - 10 days);

delete from wlm_stats_CONTROL where MESSAGE_TIME
< (current timestamp - 10 days);
```

Alternativ können die Ereignismonitortabellen regelmäßig mit *drop table* gelöscht werden und dann mit „create event monitor“ wiedererstellt.

```
DROP EVENT MONITOR WLM_EVENT;  
DROP EVENT MONITOR WLM_STATS;  
DROP TABLE WLM_EVENT ;  
DROP TABLE WLM_EVENT_STMT ;  
DROP TABLE WLM_EVENT_VALS ;  
DROP TABLE WLM_EVENT_CONTROL ;  
DROP TABLE WLM_STATS_SC ;  
DROP TABLE WLM_STATS_WC ;  
DROP TABLE WLM_STATS_WL ;  
DROP TABLE WLM_STATS_Q ;  
DROP TABLE WLM_STATS_HISTOGRAM ;  
DROP TABLE WLM_STATS_CONTROL ;
```

Erzeugen Historischer Reports

WLM Historical Analysis Tool zeigt welche Tabellen und Indexe und wie oft benutzt wurden (die gleiche Information wie Query Patroller historical analysis).

Dies wurde in Perl implementiert. Die Skripte dazu sind zu finden unter `<instance>/sqllib/samples/perl`.

Zusätzlich zu den DB2 Perl Modulen werden die Explain-Tabellen benötigt. Ebenso muß der Aktivitäten-Ereignis-Monitor (activity event monitor) vorhanden sein.

Mehr Informationen hierzu sind im [Information Center](#) zu finden.

DB2 Audit in DB2 V9.5 (Teil 1 - Konfiguration)

Für die Protokollierung/Überwachung sicherheitsrelevanter Zugriffe auf das DB2 System steht das Auditing zur Verfügung. Diese Funktionalität wurde gegenüber Version 8 deutlich ausgeweitet und verbessert. Das Auditing kann sowohl auf Instanz- als auch auf Datenbank-Ebene eingerichtet werden.

Da das Auditing der Instanz als eigenständiges Programm gestartet werden muss, kann dieser autonome Prozess auch die Aufrufe von db2start bzw. db2stop protokollieren.

In diesem Artikel soll eine kurze Übersicht über die notwendigen Schritte gegeben werden, die notwendig sind um das Auditing zu nutzen.

Audit auf Instanzebene

Für Auditing auf Instanz-Ebene wird der Befehl `db2audit` genutzt. Dieser stand bereits in Version 8 zur Verfügung. Die Aufrufe haben sich jedoch teilweise geändert, so dass Scripts überarbeitet werden müssen.

Das Auditing auf Instanz-Ebene wird vom Instanz-Owner konfiguriert und gestartet. In der DBM Konfiguration sollte der Audit Buffer vergrößert werden. Der Defaultwert 0 würde bedeuten, dass jede Aktion darauf warten muss, bis der zugehörige Auditrecord geschrieben wurde. Wird ein Wert >0 verwendet, werden die Auditrecords in diesem Cache gesammelt und ohne Beeinflussung der Antwortzeiten der DB2-Aufrufe geschrieben.

```
db2 update dbm cfg using audit_buf_sz 512
```

Nach dem Ändern der AUDIT_BUF_SZ muss die Instanz neu starten werden.

```
db2stop / db2start
```

Das Audit aller Aktivitäten der Applikationen kann sehr umfangreich und unübersichtlich werden. Daher ist es ratsam ein Konzept zu erstellen welche Aktivitäten protokolliert werden soll, ob dies auf Instanz-Ebene oder innerhalb einer Datenbank erfolgen soll und ob das Auditing z.B. auf ändernde oder fehlgeschlagene Aufrufe beschränkt werden kann.

Für die Kategorien des Audit gibt es jeweils die Möglichkeit alle Aufrufe, nur die erfolgreichen oder nur die erfolglosen (z.B. auch Permission denied) zu protokollieren. Auf Instanz-Ebene erfolgt die Konfiguration mit dem Aufruf "db2audit configure".

Beispiel:

```
db2audit configure scope objmaint status both, sysadmin status both
```

Im Beispiel sind die Scopes "objmaint" und "sysadmin" jeweils mit both, also der erfolgreichen und erfolglosen Ausführung (error) als Auditevents erstellt.

Folgende Scopes sind möglich:

- ALL (alle folgenden)
- AUDIT (Änderungen im Auditing sowie Auslesen der Auditevents)
- CHECKING (Prüfung der Berechtigungen auf Objekten)
- CONTEXT (Context des Aufrufs)
- OBJMAINT (Erstellen oder Löschen von Objekten)
- SECMAINT (Rechtevergabe)
- SYSADMIN (Aktionen die SYSADM, SYSMAIN oder SYSCTRL-Rechte bedingen)
- VALIDATE (Prüfung der Authentication)

Für jede Kategorie kann individuell der Scope

```
BOTH, FAILURE, SUCCESS, NONE
```

definiert werden.

NONE bedeutet, dass diese Kategorie nicht im Auditing berücksichtigt wird.

Nach der Konfiguration muss das Audit explizit gestartet werden:

```
db2audit start
```

Das Auditing kann mittels

```
db2audit stop
```

gestoppt werden.

Die Konfiguration des Audit kann mittels "db2audit describe" ermittelt werden:

```
db2audit describe
DB2 AUDIT SETTINGS:
Audit active: "TRUE "
Log audit events: "FAILURE"
Log checking events: "FAILURE"
Log object maintenance events: "BOTH"
Log security maintenance events: "FAILURE"
Log system administrator events: "BOTH"
Log validate events: "BOTH"
Log context events: "NONE"
```



```
Return SQLCA on audit error: "FALSE "  
Audit Data Path: ""  
Audit Archive Path: ""
```

Audit auf Datenbankebene

Neu ab DB2 Version 9.5 ist die Möglichkeit der Auditkonfiguration je Datenbank. Diese muss durch den "SECADM" vorgenommen werden. Der Instanz-Owner hat nicht per Default die notwendigen Rechte und kann daher in diesem Fall auch nicht SECADM für das Auditing werden. Damit ist ein zusätzlicher Benutzer notwendig, dem der Instanz-Owner die Rechten SECADM vergibt.

Voraussetzung für das Audit auf Datenbank-Ebene ist, dass das Auditing in der Instanz gestartet ist („db2audit start“siehe Audit auf Instanz-Eben).

Als SECADM wird das Audit einer Datenbank in zwei Stufen konfiguriert:
Zuerst muss eine **Audit Policy** für die Datenbank erstellt werden:

```
db2 "create audit policy <policy_name> categories <category> status <status> error type normal"
```

Anmerkung: Der Abschnitt "<category> status <status>" kann dabei mit Komma getrennt mehrfach hintereinander für alle benötigten Kategorien aufgeführt werden.

Die Audit_Policy kann folgende CATEGORIES enthalten:

- ALL - Alle folgenden Kategorien (EXECUTE ohne Daten)
- AUDIT - Änderungen an den AUDIT-Einstellungen
- CHECKING - Berechtigung beim Zugriff auf Objekte
- CONTEXT - Context von Datenbankoperationen (ab V9.5 in EXECUTE enthalten)
- EXECUTE - Ausführung von SQL-Statements (optional WITH DATA, WITHOUT DATA ist DEFAULT)
- OBJMAINT - Erstellen oder Löschen eines Datenbankobjekts
- SECMAINT - Vergabe oder Wegnahme von Rechten auf Datenbankobjekten
- SYSADMIN - Ausführung von Aktionen die SYSADM, SYSMAIN oder SYSCTL als Recht benötigen
- VALIDATE - Auswertung von User-Authentications

Der STATUS kann dabei jeweils auf

- BOTH - Success und Failure werden protokolliert
- FAILURE - Nur fehlgeschlagene Aktionen werden protokolliert
- SUCCESS - Nur erfolgreiche Aktionen werden protokolliert
- NONE - Keine Protokollierung dieser Kategorie (genutzt zum Ausschalten)

Die bisher erstellten Audit Policies können mittels SQL-Statement aus der Tabelle SYSAUDITPOLICIES abgefragt werden:

```
db2 "select * from SYSCAT.AUDITPOLICIES"
```

Dann kann diese Policy für das Auditing eines Datenbankobjekts verwendet werden:

```
db2 "audit database using policy <policy_name>"
```

Welche Policy aktiv ist, kann mittels SQL abgefragt werden:

```
db2 "select * from syscat.AUDITUSE"
```

Soll die Policy ersetzt werden, dann muss zuerst eine neue Policy erstellt werden, dann kann diese mittels „REPLACE“aktiviert werden:

```
db2 "audit database replace policy <new_policy_name>"
```

Policy löschen:

```
db2 "audit database remove policy <policy_name>"
```

Objekte, die mittels einer Audit Policy überwacht werden können:

- DATABASE
(Alle Ereignisse in der Datenbank)
- TABLE table-name
(Ereignisse auf einer Tabelle - Kein View)
- TRUSTED CONTEXT context-name
(Ausführung von Trusted context)
- USER authorization-name
(Ereignisse einer Benutzerkennung)
- GROUP authorization-name
(Ereignisse einer Gruppe)
- ROLE authorization-name
(Ereignisse in einer Role)
- SYSADM, SYSMANT, SYSCTRL, SYSMON, DBADM or SECADM
(Ereignisse, die die spezielle Berechtigung bedingen)

Im 2. Teil zu DB2 Auditing wird die Archivierung und Auswertung der Auditprotokolle beschrieben. Dort finden Sie dann auch eine Reihe von Beispielen.

Weiterführende Information sind zu finden im Information Center beim [audit](#) Kommando zu finden.

Schulungen / Tagungen / Informationsveranstaltung

DB2 Aktuell 2009

DB2 Aktuell 2009
15.-16. September
in Münster



Diese technische Konferenz bietet Ihnen neben fachlich hochkarätigen Vorträgen einen direkten Zugang zu den DB2 Experten aus Labors der IBM. Aktuelle Fragen, Probleme und Ideen können Sie außerhalb der offiziellen Agenda mit den Sprechern im Detail diskutieren und in Gesprächen mit den anwesenden Fachleuten anderer Firmen wertvolle Anregungen und Tipps erfahren.

Ziele des Symposiums:

- Vertiefen Sie Ihr Wissen zu Technologie und strategischen Ansätzen von DB2 für LUW und z/OS
- Knüpfen Sie Kontakte zu anderen DB2 Spezialisten und Entwicklern aus den IBM Laboren
- Nutzen Sie die Möglichkeit zur kostenlosen technischen Zertifizierung

Zielgruppe:

- IT Architekten
- DB2 Systemadministratoren und -designer

- Datenbankadministratoren
- DB2 Anwendungsentwickler und -designer

Sprecher:

Folgende Sprecher kommen für Sie nach Münster:

- Dwaine Snow, Produkt Manager DB2 für LUW, Toronto Lab
- Dr. Matthias Nicola, Technischer Leiter für DB2 XML Performance, Silicon Valley Lab
- Serge Rielau, SQL Architekt DB2 für LUW, Toronto Lab
- Terence Purcell, Performance Specialist DB2 für z/OS, Silicon Valley Lab
- Maryela Weihrauch, Distinguished Engineer, Silicon Valley Lab
- Michael Tiefenbacher, Data Management und Data Warehouse Consultant, LIS.TEC GmbH

Anmeldungs- und andere Veranstaltungsinformationen erhalten sie [hier](#).

Chats mit dem Labor

Der letzte Chat mit dem Lab zum Thema "[DB2 9.7 Overview](#)" fand am 6. Mai statt.

Eine Liste der bereits durchgeführten Chats ist [hier](#) zu finden.

Die Präsentationen der Chats, können angeschaut und heruntergeladen werden.

Analog zu den DB2 Chats gibt es auch noch BI Chats mit dem Labor. Die Präsentationen und Replays der Chats können [hier](#) angeschaut und heruntergeladen werden.

Der nächste BI Chat findet am 26.6 zum Thema "The Journey to Trusted Information - Enhanced Monitoring in DB2 9.7". Die Anmeldung erfolgt unter folgendem [Link](#).

Newsletter Archiv

Alte Ausgaben vom DB2-NL sind nun zum Nachlesen in den Archiven zu finden von:

- BYTEC : https://www.bytec.de/de/software/ibm_software/newsletter/db2newsletter/
- DRAP Solutions: <http://www.drap.de/link/db2>
- Lis.Tec: http://www.listec.de/DB2_Newsletter/View_category.html
- Cursor Software AG : <http://www.cursor-distribution.de/index.php/aktuelles/db2-newsletter>

Anmeldung/Abmeldung

Sie erhalten diesen Newsletter bis zur 3ten Ausgabe ohne Anmeldung. Wenn Sie weiterhin diesen Newsletter empfangen wollen, schicken Sie Ihre Anmeldung mit dem Subjekt „ANMELDUNG“ an db2news@de.ibm.com.

Die Autoren dieser Ausgabe:

Sollten Sie Anfragen zu den Artikeln haben, können Sie sich entweder direkt an den jeweiligen Autor wenden oder stellen Ihre Frage über den DB2 NL, denn vielleicht interessiert ja die Antwort auch die anderen DB2 NL Leser.

Doreen Stein	IT-Spezialist für DB2 LUW, IBM SWG; djs@de.ibm.com
Gerd Kaluzinski	IT-Specialist Informix Dynamic Server und DB2 UDB IBM Software Group, Information Management gerd.kaluzinski@de.ibm.com DB2 Audit in DB2 V9.5 (Teil 1 - Konfiguration)

Nela Krawez	IBM SWG, InfoSphere Balanced Warehouse Development, krawez@de.ibm.com DB2 Monitoring mit WLM (Teil 2)
-------------	--

Reviewer und Ideenlieferanten:

Nela Krawez	IBM SWG, InfoSphere Balanced Warehouse Development
Wilfried Hoge	IBM SWG, Technical Sales IM

A Smarter Planet:

IBM SWG IM Services in enger Zusammenarbeit mit Business Partnern und ISV's.

SWG Information Management Services bietet seit Anfang des Jahres IBM Business Partnern, Distributoren und ISV's eine enge Zusammenarbeit in allen Bereichen rund um die IBM Information Management Produkte an.

Im Rahmen der "A Smarter Planet" Initiative soll diese Zusammenarbeit unseren Partnern die Möglichkeit geben neue Themen zu entdecken und Lösungen zu implementieren, welche über den bisherigen Standard hinausgehen: um unsere Welt ein bisschen intelligenter und transparenter zu machen.

Durch unser perfekt aufgestelltes Team von über 120 Personen, in den Bereichen Technik, Architektur und Projektleitung, können wir unseren Partnern genau die Skills und Erfahrungen zur Seite stellen, die es ihnen ermöglichen die neuen Wege zu beschreiten.

Möchten auch Sie mithelfen unsere Welt "smarter" zu machen?
Dann schreiben Sie eine kurze Email an volker.fraenkle@de.ibm.com.