

Willkommen zum „IBM DB2 Newsletter“

Liebe Leserinnen und Leser,

Wieder neigt sich das Jahr schnellen Schrittes dem Ende entgegen. Weihnachten steht für der Tür, Zeit für besinnliche Stunden.

Ich weiß zwar nicht, wie es bei Ihnen ist, aber bisher habe ich von besinnlicher Weihnachtszeit noch nichts gemerkt. Jahresendgeschäft und natürlich der alljährliche Weihnachtsstress, Weihnachtsfeiern, Weihnachtsmarkt, Sturm auf die Läden, um noch die letzten fehlenden Geschenke zu ergattern, Gedanken darüber, was bringe ich auf den Tisch, Das alles führt dazu, das noch keine Ruhe eingekehrt ist.

Für alle, die zwischen den Feiertagen das zweifelhafte Glück haben Arbeiten zu dürfen, ist hier eine Quelle zum Stöbern und vielleicht finden Sie Zeit einiges auszuprobieren. Dabei, wie immer viel Spaß.



Wir wünschen Ihnen von Herzen frohe Weihnachten und ein gesundes und glückliches Jahr 2012.

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

Ihr TechTeam

Inhaltsverzeichnis

TECHTIPP: DB2_ALL ANSTARTEN AKTIONEN FÜR NUR BESTIMMTE PARTITIONEN	2
TECHTIPP: ENTFERNEN LANGLÄUFER ABFRAGEN OHNE DBA RECHTE.....	3
TECHTIPP: ÜBERGABE VON MEHREREN KOMMANDOS AN DB2 AUF COMAND LINE EBENE.....	6
CHATS MIT DEM LABOR.....	6
SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG.....	6
NEWSLETTER ARCHIV.....	6
ANMELDUNG/ABMELDUNG.....	7
DIE AUTOREN DIESER AUSGABE.....	7

TECHTIPP: db2_all Anstarten Aktionen für nur bestimmte Partitionen

Hatten Sie auch schon mal das Problem in einer Multipartitionsdatenbank, das sie nur auf bestimmten Partitionen Aktivitäten durchführen wollten und das wenn möglich mit db2_all?

db2_all liefert die Möglichkeit entweder alle Partitionen, oder nur eine (Bsp: db2_all "<<+0< ..") oder alle außer der einen Partition (Bsp: db2_all "<<-0< ..").

Macht man es mit Shell-Mitteln, das man

```
for partNo in 1 3 5 7 9 ; do
  db2_all "<<+$partNo< echo \$HOSTNAME:\$DB2NODE" &
done
```

kann das Problem auftreten, das sich die Prozesse auf einer Maschine gegenseitig behindern und folgende Meldung erscheint:

```
ksh: /tmp/db2inst1/rahout: not found.
```

Will man das verhindern, muss vor dem db2_all RAHBUFNAME und/oder RAHBUFDIR gesetzt werden:

```
for partNo in 1 3 5 7 9 ; do
( export RAHBUFNAME=rahout.$partNo; db2_all "<<+$partNo< echo \$HOSTNAME:\$DB2NODE" ) &
done

db2inst1@server0 [/db2/db2inst1/djs/appl/datamove]
$ for partNo in 1 3 5 7; do ( export RAHBUFNAME=rahout.$partNo; db2_all "<<+$partNo<
echo \$HOSTNAME:\$DB2NODE" ) &
[1] 4485130
[2] 2801812
[3] 2822392
[4] 2506784
[8] Done for partNo in 1 3 5 7; do ( export RAHBUFNAME=rahout.$partNo; db2_all
"<<+$partNo< echo \$HOSTNAME:\$DB2NODE" ) & done
[7] Done for partNo in 1 3 5 7; do ( export RAHBUFNAME=rahout.$partNo; db2_all
"<<+$partNo< echo \$HOSTNAME:\$DB2NODE" ) & done
[6] Done for partNo in 1 3 5 7; do ( export RAHBUFNAME=rahout.$partNo; db2_all
"<<+$partNo< echo \$HOSTNAME:\$DB2NODE" ) & done
[5] Done for partNo in 1 3 5 7; do ( export RAHBUFNAME=rahout.$partNo; db2_all
"<<+$partNo< echo \$HOSTNAME:\$DB2NODE" ) & done

db2inst1@server0 [/db2/db2inst1/djs/appl/datamove]
$
server2:5
server2: echo \$HOSTNAME:\$DB2NODE" completed ok
server1:1
server1: echo \$HOSTNAME:\$DB2NODE" completed ok
server1:3
server1: echo \$HOSTNAME:\$DB2NODE" completed ok
server2:7
server2: echo \$HOSTNAME:\$DB2NODE" completed ok
```

Ich gebe zu, auf Command Line Ebene ist es etwas nervig mit dem Maskieren, damit die Kommandos im db2_all auch so ankommen, wie man sie meint und nicht die Ausgabe des aufrufenden Hosts erscheint. Und daher zu empfehlen, das Kommando erst mal ausprobieren, bevor es in einem großen Skript versteckt wird.

Hinweis: schränkt man die Anzahl der Partitionen ein, ist auf keinen Fall das ; (wie bei db2_all "; ..") zu verwenden, das führt dazu, das rahwait auf Rückgaben von allen Maschinen wartet. Nur das auf diesen Maschinen gar nichts lief. Ich musste meine Prozesse von einer anderen Session aus killen, da die Session mit den Ausgaben

```
rah: waiting for 2646146, server8:echo
rah: waiting for 1577016, server5:echo
rah: waiting for 2199696, server6:echo
```

vollmüllte.

Bei meiner Suche nach noch laufenden Prozessen, bin ich dann über db2_all Prozessleichen gestolpert und sah dann einen Eintrag RAHOSTFILE=/home/db2inst1/sqlllib/db2nodes.cfg und da dachte ich mir, wenn ich RAHBUFNAME vorher verändern kann, geht das vielleicht auch mit

RAHOSTFILE.

Habe es gleich ausprobiert und es klappte. Prima. Hier mein Beispiel dafür:

```
export RAHOSTFILE=/db2/db2inst1/djs/appl/test/mynodes.cfg;
db2_all "|| echo \"\$HOSTNAME:\$DB2NODE\""
unset RAHOSTFILE

db2inst1@server0 [/db2/db2inst1/djs/appl/test]
$ cat mynodes.cfg
0 server0 0
1 server1 0

db2inst1@server0 [/db2/db2inst1/djs/appl/test]
$ export RAHOSTFILE=/db2/db2inst1/djs/appl/test/mynodes.cfg; db2_all "|| echo \"\$HOSTNAME:\$DB2NODE\""

rah: primary monitoring process for echo is 3023004

server0: :0
server0: echo "$HOSTNAME:$DB2NODE" completed ok

server1: :1
server1: echo "$HOSTNAME:$DB2NODE" completed ok

#-----

db2inst1@server0 [/db2/db2inst1/djs/appl/test]
$ cat mynodes.cfg
1 server1 0
32 server8 3

db2inst1@server0 [/db2/db2inst1/djs/appl/test]
$ export RAHOSTFILE=/db2/db2inst1/djs/appl/test/mynodes.cfg; db2_all "|| echo \"\$HOSTNAME:\$DB2NODE\""

rah: primary monitoring process for echo is 4190358

server1: :1
server1: echo "$HOSTNAME:$DB2NODE" completed ok

server8: :32
server8: echo "$HOSTNAME:$DB2NODE" completed ok
```

Hinweis: || wird benötigt, damit db2_all nicht sequentiell arbeitet.

Das Einzige was man jetzt also machen muss, ist ein File mit den gewünschten Partitionen zur Verfügung stellen. Dieses sollte das Layout der db2nodes.cfg haben (habe es nicht mit einem anderen ausprobiert) und schon kann es los gehen. Yippiee.

Bevor Sie aber selbst so ein File erstellen, empfehle ich dieses basierend auf dem Original zu erstellen, dann kann nichts schief gehen, von wegen Hostname, Zuordnung der Partitionen und Co.

```
egrep -e "^1|^32" ~/sqllib/db2nodes.cfg >mynodes.cfg
...
```

Hinweis: Und nicht vergessen, nach der Ausführung das RAHOSTFILE wieder zurückzusetzen. Mir hat das enorm geholfen und ich hoffe Ihnen auch.

TechTipp: Entfernen Langläufer Abfragen ohne DBA Rechte

Da unsere Entwickler keine DBA-Rechte haben, mussten immer wieder "unschöne" Transaktionen aus dem DB2 entfernt werden.

Dies wurde bisher über eine selbstgeschriebene Prozedur abgewickelt.

Der Benutzer konnte in eine Tabelle die HandleID eintragen und alle 5 Minuten wurde dann die Prozedur ausgeführt. War in der Tabelle ein Eintrag, wurde diese Transaktion beendet. Soweit so gut.

Eine andere Möglichkeit bietet die Funktion `WLM_CANCEL_ACTIVITY`. Damit ein User Anwendungen beenden kann, benötigt er nur Ausführungsrechte auf diese Funktion.

```
db2 grant execute on procedure SYSPROC.WLM_CANCEL_ACTIVITY to user <USERID>
```

Mit der folgende View, kann der Entwickler nun die laufende Transaktionen finden und mit dem Aufruf der o.g. Prozedur die Transaktion stoppen:

```
DROP VIEW CURRENT_SQL;

CREATE VIEW CURRENT_SQL ("APPLICATION_HANDLE", "UOW_ID", "ACTIVITY_ID", "LOCAL_START_TIME",
"STMT_TEXT", "ACTIVITY_STATE", "ACTIVITY_TYPE", "TOTAL_CPU_TIME", "ROWS_READ", "ROWS_RETURNED",
"QUERY_COST_ESTIMATE", "DIRECT_READS", "DIRECT_WRITES", "TOTAL_ACT_TIME", "TOTAL_ACT_WAIT_TIME") AS
WITH A1 AS
  (SELECT * FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97(null, -1))
   WHERE activity_id > 0 )
SELECT A1.application_handle, A1.uow_id, A1.activity_id, A1.local_start_time,
       actmetrics.stmt_text,
       A1.activity_state, A1.activity_type, A1.total_cpu_time,
       A1.rows_read, rows_returned, A1.query_cost_estimate,
       A1.direct_reads, A1.direct_writes, actmetrics.total_act_time,
       actmetrics.total_act_wait_time
FROM A1,
     TABLE(MON_GET_ACTIVITY_DETAILS(A1.application_handle, A1.uow_id,A1.activity_id, -1)) AS
ACTDETAILS,
     XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
              '$actmetrics/db2_activity_details'
              PASSING XMLPARSE(DOCUMENT ACTDETAILS.DETAILS) AS "actmetrics"
              COLUMNS "STMT_TEXT" VARCHAR(32000) PATH 'stmt_text',
                      "TOTAL_ACT_TIME" INTEGER PATH 'activity_metrics/total_act_time',
                      "TOTAL_ACT_WAIT_TIME" INTEGER PATH 'activity_metrics/total_act_wait_time' ) AS
ACTMETRICS;
```

Nachdem der View angelegt wurde, benötigt der User/Gruppe die Select-Rechte für diesen View, sowie die EXECUTE Rechte für die verwendeten Table-Funktionen.

```
db2 grant select on <SCHEMA>.current_sql to user <USERID>
db2 grant execute on function SYSPROC.MON_GET_ACTIVITY_DETAILS to user <USERID>
db2 grant execute on function SYSPROC.WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 to user <USERID>

SELECT * FROM <SCHEMA>.CURRENT_SQL;
```

Die View zeigt die laufenden SQL an - und noch mehr.

```
$ db2 "select * from current_sql" | tr -s " " | tr -s '-' ' '

APPLICATION_HANDLE UOW_ID ACTIVITY_ID LOCAL_START_TIME STMT_TEXT ACTIVITY_STATE ACTIVITY_TYPE
TOTAL_CPU_TIME ROWS_READ ROWS_RETURN
ED QUERY_COST_ESTIMATE DIRECT_READS DIRECT_WRITES TOTAL_ACT_TIME TOTAL_ACT_WAIT_TIME
-----
873 1 1 2011-07-27-22.53.11.330998 select * from current_sql EXECUTING READ_DML 0 0 0 62959 0 0 0 0
1 record(s) selected.
```

Mit Prozedur CALL `WLM_CANCEL_ACTIVITY` (APPHANDLE, UOWID, ACTIVITYID), können dann "unschöne" laufende Verbindungen schnell abgebrochen werden.

Dabei hat der Entwickler jedoch auch die Möglichkeit Anwendungen anderer Entwickler oder DBA's zu entfernen.

<pre>> db2 "select * from syscat.tables t, syscat.tablespaces, syscat.views, syscat.bufferpools" >/dev/null</pre>	<pre>> db2 "select * from db2inst1.current_sql" tr -s " " grep -v -- "--" APPLICATION_HANDLE UOW_ID ACTIVITY_ID LOCAL_START TIME STMT TEXT ACTIVITY STATE</pre>
---	---

```

ACTIVITY_TYPE TOTAL_CPU_TIME ROWS_READ
ROWS_RETURNED QUERY_COST_ESTIMATE
DIRECT_READS DIRECT_WRITES TOTAL_ACT_TIME
TOTAL_ACT_WAIT_TIME
 23 13 1 2011-07-28-10.59.49.484217 select *
from syscat.tables t, syscat.tablespaces,
syscat.views, syscat.bufferpools IDLE
READ_DML 1692000 18145 18063 1064 60308 0
1876 1336
 8 13 1 2011-07-28-10.59.55.636844 select *
from db2inst1.current_sql EXECUTING READ_DML
2000 0 0 64609 0 0 0 0

2 record(s) selected.

> db2 "call wlm_cancel_activity(23,13,1)"

Return Status = 0

db2inst1@ibmclass:/home/db2inst1
> echo $?
4

```

Durch Erweiterung der oberen Abfrage um die USER-ID und Einschränkung der Ausgabe der Applications auf den abfragenden USER, kann man minimieren, das „falsche“ Anwendungen, d.h. Anwendungen des Administrators oder anderer Entwickler, beendet werden.

```
$ db2 "select * from current_sql" | tr -s " " | grep -v -- "--"
```

```

APPLICATION_HANDLE UOW_ID ACTIVITY_ID LOCAL_START_TIME STMT_TEXT ACTIVITY_STATE ACTIVITY_TYPE
TOTAL_CPU_TIME ROWS_READ ROWS_RETURN
ED QUERY_COST_ESTIMATE DIRECT_READS DIRECT_WRITES TOTAL_ACT_TIME TOTAL_ACT_WAIT_TIME USERID
1169 1 1 2011-07-28-10.33.56.516535 select * from current_sql EXECUTING READ_DML 0 0 0 64333 0 0 0 0
DOREEN

1 record(s) selected.

```

```
$ db2 "select * from user_current_sql" | tr -s " " | grep -v -- "--"
```

```

APPLICATION_HANDLE UOW_ID ACTIVITY_ID LOCAL_START_TIME STMT_TEXT ACTIVITY_STATE ACTIVITY_TYPE
1180 1 1 2011-07-28-10.37.55.791400 select * from user_current_sql EXECUTING READ_DML

1 record(s) selected.

```

Die dazugehörige DDL könnte dann wie folgt aussehen:

```

drop view current_sql;
CREATE VIEW CURRENT_SQL ("APPLICATION_HANDLE", "UOW_ID", "ACTIVITY_ID", "LOCAL_START_TIME",
"STMT_TEXT", "ACTIVITY_STATE", "ACTIVI
TY_TYPE", "TOTAL_CPU_TIME", "ROWS_READ", "ROWS_RETURNED", "QUERY_COST_ESTIMATE", "DIRECT_READS",
"DIRECT_WRITES", "TOTAL_ACT_TIME"
, "TOTAL_ACT_WAIT_TIME", USERID ) AS

WITH A1 AS
  (SELECT *
   FROM TABLE(wlm_get_workload_occurrence_activities_v97(null, -1))
   WHERE activity_id > 0 )

SELECT A1.application_handle, A1.uow_id, A1.activity_id, A1.local_start_time, actmetrics.stmt_text,
A1.activity_state, A1.activity_type, A1.total_cpu_time,
A1.rows_read, rows_returned, A1.query_cost_estimate,
A1.direct_reads, A1.direct_writes, actmetrics.total_act_time,
actmetrics.total_act_wait_time
, a2.authid
FROM A1,
TABLE(MON_GET_ACTIVITY_DETAILS(A1.application_handle, A1.uow_id,A1.activity_id, -1)) AS
ACTDETAILS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
'$actmetrics/db2_activity_details'
PASSING XMLPARSE(DOCUMENT ACTDETAILS.DETAILS) AS "actmetrics"
COLUMNS "STMT_TEXT" VARCHAR(32000) PATH 'stmt_text',
"TOTAL_ACT_TIME" INTEGER PATH 'activity_metrics/total_act_time',
"TOTAL_ACT_WAIT_TIME" INTEGER PATH 'activity_metrics/total_act_wait_time' ) AS
ACTMETRICS
, sysibmadm.applications A2
where A1.application_handle=A2.agent_id
;
```

```
create view user_current_sql (application_handle, uow_id, activity_id, local_start_time, stmt_text,
activity_state, activity_type)
as
select application_handle, uow_id, activity_id, local_start_time, stmt_text, activity_state,
activity_type
from current_sql
where userid = upper(user)
;
```

Der User bekommt nun Execute Rechte auf die WLM_CANCEL_ACTIVITY Funktion und Lese-Rechte auf den USER_CURRENT_SQL. Da er nur seine Anwendung sieht, kann er auch nur seine beenden.

Nachteil dieser Methode ist, das für den Zugriff auf SYSIBMADM.APPLICATION mindestens SYSMON Rechte benötigt werden. Rechte auf den Funktionen unterhalb des Administrationsviews, reichen nicht aus.

Mehr dazu im Information Center (Scenario: [How to cancel activities queued for more than an hour](#)).

TechTipp: Übergabe von mehreren Kommandos an db2 auf Comand Line Ebene

In diesem Artikel wollen wir zeigen, wie man mehrere Kommandos an db2 übergeben kann, ohne ein separates File, das mit `db2 -tvf <file>` aufgerufen wird, anzulegen.

Ruft man db2 mit Semikolon getrennten Kommandos auf, tritt folgender Fehler auf:

```
$ db2 "connect to MIGR; select count(*) from syscat.tables"
SQL0104N  An unexpected token "select" was found following "<identifier>".
Expected tokens may include:  "USER".  SQLSTATE=42601
```

Verwendet man nun die Pipe, wie bereits in Ausgabe 2010/03 beschrieben, sieht das Ergebnis wie folgt aus:

```
$ echo "connect to MIGR \n select count(*) from syscat.tables " | db2 +p
```

```
Database Connection Information
```

```
Database server          = DB2/AIX64 9.7.4
SQL authorization ID     = DB2MIGR
Local database alias     = MIGR
```

```
1
```

```
-----
11178
```

```
1 record(s) selected.
```

Chats mit dem Labor

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

Die Präsentationen der Chats können dort angeschaut und heruntergeladen werden.

Schulungen / Tagungen / Informationsveranstaltung

Eine Liste der anstehenden Konferenzen ist [hier](#) zu finden.

Newsletter Archiv

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

- [Lis.Tec](#)
- [Bytec](#)
- [Drap](#)
- [Cursor Software AG](#)

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 Subject „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	Certified IT-Spezialist für DB2 LUW, IBM SWG; Chief-Editor DB2NL, djs@de.ibm.com
David Hültenschmidt	REWE Group Artikel: TechTipp: Übergabe von mehreren Kommandos an db2 auf Comand Line Ebene
Joachim Müller	Douglas Artikel: TechTipp: Entfernen Langläufer Abfragen ohne DBA Rechte

Reviewer und Ideenlieferanten:

Dirk Fechner	IBM SWG
Nela Krawez	IBM SWG
Wilfried Hoge	IBM SWG
Peter Schurr	IBM SWG