

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

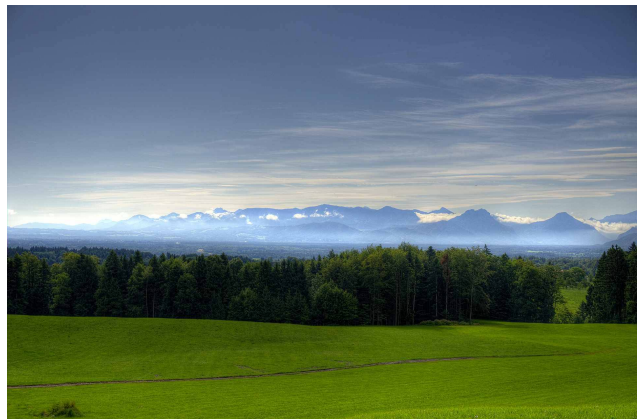
Aktuelles.....	1
TechTipp: Neue Features in der Version 11.70.xC3.....	2
TechTipp: Flexible Grid – Replikation der nächsten Generation (Teil 2).....	3
TechTipp: Grid Tabellen in der Datenbank syscdr.....	7
TechTipp: Optionen des ONSTAT (onstat -g smb).....	8
TechTipp: Optionen des ONCHECK (oncheck -cS / -pS).....	10
Termine: INFORMIX 11.70 – New Features (IX3020DE).....	12
WebTipp: INFORMIX IWA 11.70.FC3 – Enhancements.....	13
WebTipp: Informationen zum Informix Warehouse Accelerator.....	13
WebTipp: Free Informix – News zu INFORMIX.....	13
WebTipp: A smarter database for a smarter planet.....	13
Referenzen zu TimeSeries: Coldset Printing Partners	14
Anmeldung / Abmeldung / Anmerkung.....	14
Die Autoren dieser Ausgabe.....	15

Aktuelles

Liebe Leserinnen und Leser,

die Themen TimeSeries (besonders im Zusammenhang mit Smart Metering), der Warehouse Accelerator, die Grid Replikation und die vielen neuen Features der aktuellen Version liefern reichlich Stoff um Sie auch während der Sommerferien mit Tipps und Tricks zu versorgen, die Sie nach den Ferien im Alltag produktiv anwenden können.

Auch die Redaktion hat sich ein paar wenige Tage gegönnt und Urlaub in Bayern gemacht, wie sie am Foto rechts sehen können.



Wie immer haben wir für Sie eine Reihe an Tipps und Tricks zusammengestellt.

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

TechTipp: Neue Features in der Version 11.70.xC3

Das Release 11.70.xC3 hat eine ganze Reihe an nützlichen Features mit sich gebracht. Anbei wollen wir als Übersicht diese Features auflisten und kurz beschreiben. In den kommenden Ausgaben des INFORMIX Newsletters werden wir dann einige der neuen Features näher beleuchten und anhand praktischer Beispiele vorstellen.

- Automatic read-ahead operations
Statt der festen Einstellungen über RA_PAGES und RA_THRESHOLD kann das Verhalten des ReadAhead nun dynamisch erfolgen, falls AUTO_READAHEAD gesetzt wird.
- Verhalten des Servers bei einem Engpass im Bereich Speicher
LOW_MEMORY_RESERVE - Reservierung für systemkritische Aktivitäten
LOW_MEMORY_MGR – Beschränkung von Sessions bei Engpässen
- Connection Manager enhancements
HRD/RSS/SDS und ER können nun gemeinsam über einen ConnectionManager verwaltet werden
- Erweiterungen des OpenAdmin Tools
u.a. ein PlugIn zur Konfiguration von TimeSeries / Codesetwahl bei der CDR
- Ein Task für das Houskeeping beim Message Log steht zur Verfügung
- SQL Built-In Funktionen wurden im Bereich Stringoperationen und Trigonometrie erweitert :
 - * CHARINDEX() - Startposition eines Suchstrings ermitteln
 - * INSTR() - Anzahl Vorkommen eines Suchstrings ausgeben
 - * LEFT() - Erste x Zeichen des String ausgeben
 - * LEN() - Alias für Length()
 - * REVERSE() - Umkehrung der Schriftrichtung
 - * RIGHT() - Letzte x Zeichen ausgeben
 - * SPACE() - String aus Leerzeichen (blancs) erzeugen
 - * SUBSTRING_INDEX() - Teilstring nach Delimiter ausgeben
 - * DEGREES() - Gradzahl zu Kreissegment
 - * RADIANS () - Kreissegment zu einer Gradzahl
- Der Aufruf von „Grid Connect“ wurde nunmehr auch in der Prozedur sysdbopen() erlaubt.
- Code set conversion wurde für die Enterprise Replication eingeführt
Die Replikation zwischen unterschiedlichen Codesets ist nunmehr möglich.
Damit kann eine Migration auf UTF8 mittels Replikation erfolgen.
- Einschränkung der Rechte von Mapped Usern im Betriebssystem
- Automatisierungen bei der Nutzung von TimeSeries
- weitere Anpassungen

TechTipp: Flexible Grid – Replikation der nächsten Generation (Teil 2)

Nachdem wir in Teil 1 (in der Ausgabe Juli 2011) bereits den Aufbau und die Nutzung von DDL Statements im Grid vorgestellt haben, geht es im 2. Teil um die Ausführung von DML-Statements im Grid.

Die Prozedur „ifx_grid_execute“ ermöglicht es, eine SQL-Anweisung auf allen im Grid beteiligten Instanzen gleichzeitig auszuführen.

Beispiel - Eintrag der Session-ID und des DBServernamens in eine Tabelle (sesinfo):

```
execute procedure ifx_grid_execute("test_grid",
  "insert into sesinfo values (dbinfo('sessionid'),DBSERVERNAME)"
);
```

Die Prozedur „ifx_grid_execute()“ liefert nie Ausgaben zurück. Alle Aufrufe werden als „begin work without replication“ ausgeführt. Daher werden die Ergebnisse nicht an die anderen Server im Grid übertragen. Ergebnisse können nur an den jeweiligen Instanzen abgefragt werden oder mittels „cdr list grid -v“.

cdr list grid -v:

```
Node:test2_rep Stmtid:23 User:informix Database:stores 2011-07-24
11:08:20
```

```
insert into sesinfo values (dbinfo('sessionid'),DBSERVERNAME)
```

```
ACK test1_rep 2011-07-24 11:08:21
```

```
ACK test2_rep 2011-07-24 11:08:20
```

```
ACK test3_rep 2011-07-24 11:08:21
```

Ein „select * from sesinfo“ auf den einzelnen Servern ausgeführt ergibt danach:

Server test1:

```
f1                                f2
180                               test1
```

Server test2:

```
f1                                f2
137                               test2
```

Server test3:

```
f1                                f2
142                               test3
```

Die Abfrage über die Server hinweg:

```
execute procedure ifx_grid_execute("test_grid",
  "select * from sesinfo"
);
```

```
<<procedure executed>>
```

Die Ergebnisse des Prozeduraufrufs sind mit „cdr list grid -v“ zu sehen.

```
cdr list grid -v:
```

```
Node:test2_rep Stmtid:24 User:informix Database:stores 2011-07-24
12:12:19
```

```
select * from sesinfo
```

```
ACK test1_rep 2011-07-24 12:12:19
  '180','test1'
```

```
ACK test2_rep 2011-07-24 12:12:19
  '137','test2'
```

```
ACK test3_rep 2011-07-24 12:12:19
  '142','test3'
```

Befehle, die nicht erfolgreich waren, können nach Behebung des Problems erneut gestartet werden. Fehlgeschlagene Aktionen im „Grid“ können dazu mit REDO erneut ausgeführt werden.

Beispiel:

```
execute procedure ifx_grid_connect('test_grid', 1);
create table kalu (f1 char(42));
alter table kalu add (f2 char(23));
alter fragment on table kalu init fragment by expression
  f1[1] <= "D" in rootdbs,
  f1[1] <= "R" in datadbs,
  remainder in data_time ; <- DBSpace existiert nur auf test1
execute procedure ifx_grid_disconnect();
```

Da der DBSpace „time_data“ nicht auf allen Instanzen angelegt war, konnte der Befehl „alter fragment“ nicht im „Grid“ ausgeführt werden.

Die restlichen Befehle wurden von allen beteiligten Instanzen ausgeführt und bestätigt:

```
cdr list grid -v
```

Grid	Node	User
test_grid	test1_rep*	informix kalu
	test2_rep*	informix kalu
	test3_rep*	informix kalu

Details for grid test_grid

Node:test1_rep Stmtid:1 User:informix Database:stores 2011-07-24
11:33:18

create table kalu (f1 char(42))

ACK test1_rep 2011-07-24 11:33:18

ACK test2_rep 2011-07-24 11:33:19

ACK test3_rep 2011-07-24 11:33:19

Node:test1_rep Stmtid:2 User:informix Database:syscdr 2011-07-24
11:33:18

Define Repl G65567_1_2_kalu for stores:informix.kalu

ACK test1_rep 2011-07-24 11:33:19

ACK test2_rep 2011-07-24 11:33:19

ACK test3_rep 2011-07-24 11:33:19

Node:test1_rep Stmtid:3 User:informix Database:stores 2011-07-24
11:33:19

alter table kalu add (f2 char(23))

ACK test1_rep 2011-07-24 11:33:19

ACK test2_rep 2011-07-24 11:33:19

ACK test3_rep 2011-07-24 11:33:19

Node:test1_rep Stmtid:4 User:informix Database:syscdr 2011-07-24
11:33:19

Remaster Repl : Primary Repl=G65567_1_2_kalu, Shadow

Repl=Shadow_4_G65567_1_2_kalu_1311759199_1_4 for

stores:informix.kalu

ACK test1_rep 2011-07-24 11:33:19

ACK test2_rep 2011-07-24 11:33:19

ACK test3_rep 2011-07-24 11:33:19

Node:test1_rep Stmtid:5 User:informix Database:stores 2011-07-24
11:34:39

alter fragment on table kalu init fragment by expression

f1[1] <= "D" in rootdbs,

f1[1] <= "R" in datadbs,

remainder in time_data

ACK test1_rep 2011-07-24 11:34:39

**NACK test2_rep 2011-07-24 11:34:39 SQLERR:-261 ISAMERR:-130
Grid Apply Transaction Failure**

**NACK test3_rep 2011-07-24 11:34:39 SQLERR:-261 ISAMERR:-130
Grid Apply Transaction Failure**

Die fehlgeschlagenen Statements können mittels „**ifx_grid_redo**“ erneut auf den Servern ausgeführt werden, auf denen der Fehler aufgetreten ist. Erfolgreiche Befehle werden dabei nicht erneut ausgeführt.

Nach Anlage des fehlenden DBSpace auf den Instanzen wird dazu folgender Befehl ausgeführt:

```
execute procedure ifx_grid_redo('test_grid');
```

```
cdr list grid -v
```

```
Node:test1_rep Stmtid:5 User:informix Database:stores 2011-07-27  
11:34:39
```

```
alter fragment on table kalu init fragment by expression  
  f1[1] <= "D" in rootdbs,  
  f1[1] <= "R" in datadbs,  
  remainder in time_data
```

```
ACK test1_rep 2011-07-27 11:34:39
```

```
ACK test2_rep 2011-07-27 13:24:35
```

```
ACK test3_rep 2011-07-27 13:24:35
```

```
NACK test2_rep 2011-07-27 13:24:35 SQLERR:0 ISAMERR:0  
Grid Apply Transaction Failure
```

```
NACK test3_rep 2011-07-27 13:24:35 SQLERR:0 ISAMERR:0  
Grid Apply Transaction Failure
```

Hinweis: Die bisherigen Fehlermeldungen werden nochmals mit aufgeführt, (nun jedoch mit SQLERR 0 und ISAMERR 0) damit zu erkennen ist, welche Befehle nun erfolgreich ausgeführt werden konnten.

Die Redo-Befehle könnten auch einzeln für die jeweiligen Instanzen ausgeführt werden. Dabei wird der Replikations-Server angegeben, auf dem der fehlgeschlagene Befehl wiederholt ausgeführt werden soll, sowie die Statement-ID des Kommandos, das fehlgeschlagen ist.

Syntax:

```
execute procedure
```

```
ifx_grid_redo('<Gridname>', '<TargetServer>', '<SourceServer>', <tag_<br>identifizier>', '<Statement_ID>');
```

```
execute procedure
```

```
ifx_grid_redo('test_grid', 'test2_rep', 'test1_rep', NULL, '5');
```

```
execute procedure
```

```
ifx_grid_redo('test_grid', 'test3_rep', 'test1_rep', NULL, '5');
```

In der Command History (cdr list grid -v) sammeln sich immer mehr erfolgreiche und erfolglose Befehle an. Diese Einträge können nach Abarbeitung gelöscht werden:

```
execute procedure
  ifx_grid_purge('test_grid',NULL,NULL,NULL,NULL,'force');
```

Grid Replikation ermöglicht es, auf einer (virtuellen) Instanz, die aus einer Reihe von Servern besteht, SQL-Kommandos wahlweise lokal oder auf einer Gruppe von Servern (Grid) auszuführen. Alle unterstützten SQL, also DDL und DML Befehle können nach einem Connect zum Grid gleichzeitig auf allen Servern ausgeführt werden, wobei sich die Strukturen, die für die Enterprise Replikation notwendig sind, automatisch anpassen.

TechTipp: Grid Tabellen in der Datenbank syscdr

Wer weiter in die Tiefen des Systems einsteigen will, der kann in der Datenbank syscdr Abfragen durchführen. Hier sind z.B. in der Tabelle „grid_cmd_errors“ die Commands zu finden, die einen Fehler bei der Ausführung zurückgeliefert haben.

In der Tabelle „grid_redo“ findet sich die Information, welche der Commands auf welchen Instanzen mittels redo() erneut aufgerufen wurden.

Anbei die Liste der Tabellen:

Tabelle	View	Kommentar
grid_cdr_node_tab	grid_cdr_node	Grid Node Information
grid_def_tab	grid_def	Repl.Sets im Grid
grid_part_tab	grid_part	Participants im Grid
grid_users_tab	grid_users	User für Grid Nutzung
grid_cmd_tab	grid_cmd	Grid Befehlsheader
grid_cmd_part_tab	grid_cmd_part	Grid Befehle
grid_cmd_errors_tab	grid_cmd_errors	Fehler bei Grid-Befehlen
grid_cmd_ack_tab	grid_cmd_ack	Ausgeführte Grid-Befehle
grid_redo_tab	grid_redo	Ausgeführte Redo-Befehle
grid_repl_tab		Replicates im Grid

Für das Zusammenspiel der Grid Replikation mit dem Connection Manager wurden folgende Tabellen eingeführt:

grid_cm_nodes	Grid-Nodes, die vom CM überwacht werden.
grid_cm_sla	Service Level Agreements der Connection Manager.
grid_cm_er_serv	Replikationsserver je SLA der Connection Manager.

TechTipp: Optionen des ONSTAT (onstat -g smb)

SmartLargeObjects zur Speicherung von Texten, Bildern oder anderen Dokumenten gewinnen immer mehr an Bedeutung. Der Vorteil besteht darin, dass die Datenbank als zentraler Speicherort alle Informationen beherbergt, und diese bei einer Sicherung in einem konsistenten Stand in Relation zu den relationalen Daten speichern kann.

Die Größe der angelegten DBSpaces für SmartBlobs ist im „onstat -d“ zu sehen, wobei hier der Bereich der Objektablage und der Bereich für die Metadaten (Zusatzinformationen zu den Objekten) gesondert aufgelistet wird.

Beispiel:

```
IBM Informix Dynamic Server Version 11.70.UC3 -- On-Line -- Up 01:17:41 ...
```

Dbspaces

address	number	flags	fchunk	nchunks	pgsize	flags	owner	name
...								
572b6c30	6	0x48001	6	1	2048	N SBA	informix	sbdbbs
5a253148	8	0x68001	8	1	2048	N SBA	informix	sbdbbs2

8 active, 2047 maximum

Chunks

address	chunk/dbs	offset	size	free	bpages	flags	pathname
...							
59c5b7b8	6	64	25000	23240	23241	POSB-D	./sbdbbs
		Metadata	1706	1269	1706		
5a2532b8	8	0	50000	46558	46558	POSB-D	./sbdbbs2
		Metadata	3389	2522	3389		

8 active, 32766 maximum

SmartBlobSpaces können beim Anlegen unterschiedliche Eigenschaften als Default Parameter erhalten. Dies sind **LOGGING** {ON|OFF}, was beeinflusst, wie das Loggingverhalten bei BLOBs ist, **ACCESSTIME** {ON|OFF} was bestimmt, ob die letzten Zugriffszeiten gespeichert werden, sowie die durchschnittlich geschätzte Größe der Large Objects **AVG_LO_SIZE** {1 – 2097152} in kB.

Nach der Anlage sind diese Optionen im „onstat -d“ nicht mehr zu sehen. Wer wissen will, mit welchen Optionen die SBSpaces angelegt wurden, der muss auf den „onstat -g smb“ zurückgreifen.

```
onstat -g smb
```

Smart Blob Usage:

```
onstat -g smb s | c | h | e | t
smb          usage
smb t        print timing statistics for enabled server (*)
smb s        smartblob spaces
smb c        smartblob chunks
smb fdd      LO file descriptor table
smb lod      LO header table
(*= internal use only)
```


Der Aufruf mit dem Parameter „s“ zeigt die Defaultwerte, mit denen der SBSpace angelegt wurde:

```
onstat -g smb s
IBM Informix Dynamic Server Version 11.70.UC3 -- On-Line -- Up 01:49:42 ...
```

Sbspace Summary:

```
sbnum 6      address 572b6c30
  Space      : flags      nchk      owner      sbname
                ----- 1          informix   sbdbs
  Defaults   : LO_LOG

  LO         : ud b/pg  flags      flags      avg s/kb max lcks
                2048    0          ----- -1          -1
  Ext/IO     : 1st sz/p  nxt sz/p  min sz/p  mx io sz
                0        0          0          -1
  HdrCache  : max      free
                512      1

sbnum 8      address 5a253148
  Space      : flags      nchk      owner      sbname
                ----- 1          informix   sbdbs2
  Defaults   : LO_LOG LO_KEEP_LASTACCESS_TIME

  LO         : ud b/pg  flags      flags      avg s/kb max lcks
                2048    0          ----- -1          -1
  Ext/IO     : 1st sz/p  nxt sz/p  min sz/p  mx io sz
                0        0          0          -1
  HdrCache  : max      free
                512      0
```

Details über die Chunks der SBSpaces zeigt der „onstat -g smb c“:

```
onstat -g smb c
IBM Informix Dynamic Server Version 11.70.UC3 -- On-Line -- Up 01:53:58 --
380452 Kbytes
```

Chunk Summary:

```
sbnum 6      chunk 6
Chunk: address flags      offset  size      orig fr  usr pgs  free pg
        59c5b7b8 FR----- 64      25000    23241   23241   13657
path: ./sdbbs
        start pg npages
  Ud1   : 53      11620
  Md    : 11673   1706
  Ud2   : 13379   11621

sbnum 8      chunk 8
Chunk: address flags      offset  size      orig fr  usr pgs  free pg
        5a2532b8 FR----- 0       50000    46558   46558   41508
path: ./sdbbs2
        start pg npages
  Ud1   : 53      23279
  Md    : 23332   3389
  Ud2   : 26721   23279
```

Der „onstat -g smb lod“ zeigt die einzelnen Objekte an. Hier ist neben der Größe auch der SBSpace und Chunk zu sehen, in dem das Objekt gespeichert wurde:

```
onstat -g smb lod
IBM Informix Dynamic Server Version 11.70.UC3 -- On-Line -- Up 01:57:52 ...
```

Lo Header Table Header

Table_Name	Flags	# Primary	# Sec.	Entry_Size	Capacity
# Buckets	# Heads	# Entries	# Extends	# Reused	
sb_loheader	0x5c	512	80	508	80
512	409	27	0	1	

Lo Header Table Entries

opns	refs	size	ha_status	h_status	[sbs,chk,seq(rid),oid]	address	lsn
0	1	988496	0x00000004	0x00000000	[6,6,2(0x103),1306138840]	0x57196030	0,0
0	1	953903	0x00000005	0x00000000	[8,8,1(0x102),1311836568]	0x57196470	0,0
0	1	1122146	0x00000004	0x00000000	[6,6,1(0x102),1306138841]	0x57196690	0,0

Im Beispiel sind drei Dokumente zu sehen. Das zweite Dokument ist im SBSpace mit der ID 8 enthalten. Das erste und dritte Dokument im SBSpace mit der ID 6.

Die angezeigte Größe stimmt mit der Größe der Dateien im Filesystem überein:

Der „ls -l“ zeigt:

```
988496      /home/kalu/IFX/IFX_11.70/Autolocate.odp
953903      /home/kalu/IFX/IFX_11.70/Autonomic.odp
1122146     /home/kalu/IFX/IFX_11.70/AutoRegistrationPanther.odp
```

TechTipp: Optionen des ONCHECK (oncheck -cS / -pS)

Die im vorige Artikel beschriebenen Informationen über SmartBlobs können auch über den oncheck verifiziert und ausgelesen werden.

Der „oncheck -cS“ prüft dabei die Konsistenz der Ablage der Objekte in den SBSpaces und gibt Informationen über die Größe und den Zeitpunkt der letzten Änderung aus.

oncheck -cS

```
Validating space 'sdbds' ...
```

sbspace	Metadata	Partition	Partnum	Used	Free
sdbds:	'informix'	.TBLSpace	0x600001	6	44
sdbds:	'informix'	.sbspace_desc	0x600002	2	2
sdbds:	'informix'	.chunk_adjunc	0x600003	2	2
sdbds:	'informix'	.LO_ud_free	0x600004	2	63
sdbds:	'informix'	.LO_hdr_partn	0x600005	3	361

Large Objects

ID	Sbs#	Chk#	Seq#	Ref Cnt	Size (Bytes)	Allocced Pages	Extns	Creat Flags	Last Modified
	6	6	1	1	1122146	556	1	L-N-H Sun Jul 24 10:44:48 2011	
	6	6	2	1	988496	490	1	L-N-H Sun Jul 24 10:44:47 2011	

...

Validating space 'sdbds2' ...

sbspace Metadata Partition	Partnum	Used	Free
sdbds2:'informix'.TBLSpace	0x800001	6	44
sdbds2:'informix'.sbspace_desc	0x800002	2	2
sdbds2:'informix'.chunk_adjunc	0x800003	2	2
sdbds2:'informix'.LO_ud_free	0x800004	2	127
sdbds2:'informix'.LO_hdr_partn	0x800005	1	729

Large Objects

ID	Ref	Size	Alloced	Creat	Last			
Sbs#	Chk#	Seq#	Cnt	(Bytes)	Pages	Extns	Flags	Modified
8	8	1	1	953903	473	1	L-K-H	Sun Jul 24 10:44:47 2011

...

Hier finden sich die im vorigen Beispiel eingelesenen Dokumente wieder.

Zur Größe des Dokuments wird hier noch die Anzahl der belegten Pages angezeigt, sowie das letzte Änderungsdatum.

Die Aufrufe des „oncheck“ mit dem Argument „-c“ dienen vornehmlich der Prüfung der Konsistenz von Objekten.

Will man mehr Details über die analysierten Objekte erfahren, so sollte die Option „-p“ eingesetzt werden, die prüft und zusätzlich die Informationen ausgibt.

Die Details zu den Zugriffszeiten (Created, Last Accessed, Last Modified) können mittels oncheck -pS gesehen werden.

oncheck -pS

TBLSpace Report for sdbds:'informix'.TBLSpace

TBLspace Flags	2801	Page Locking
		TBLspace use 4 bit bit-maps
		Permanent System TBLspace

Partition partnum	0x600001
Rowsize	136
Number of special columns	0

...

Number of data pages	0
Number of rows	4
Partition lockid	6291457
Optical Cluster Partnum	-1
Current SERIAL8 value	1
Current REFID value	1
Created	Mon May 23 10:12:17 2011

Zuerst werden Informationen über den SBSpace selbst ausgegeben, so z.B. das Erstellungsdatum. Anschließend werden die einzelnen Objekte (blobs) aufgelistet:

```

Space Chunk Page = [6,6,1]  Object ID = 1306138841
  LO SW Version                4
  LO Object Version            1
  Created by Txid              13
  Flags                        0x31  LO_LOG  LO_NOKEEP_LASTACCESS_TIME
LO_HIGH_INTEG
  Data Type                    0
  Extent Size                  -1
  IO Size                      0
  Created                      Sun Jul 24 10:44:48 2011
  Last Time Modified           Sun Jul 24 10:44:48 2011
  Last Time Accessed           Sun Jul 24 10:44:48 2011
  Last Time Attributes Modified Sun Jul 24 10:44:48 2011
  Ref Count                    1
  Create Flags                 0x31  LO_LOG  LO_NOKEEP_LASTACCESS_TIME
LO_HIGH_INTEG
  Status Flags                 0x0  LO_FROM_SERVER
  Size (Bytes)                 1122146
  Size Limit                   -1
  Total Estimated Size         -1
  Deleting TxId                -1
  LO Map Size                  200
  LO Map Last Row              -1
  LO Map Extents               2
  LO Map User Pages            556

```

Zu sehen ist, dass das Objekt mit Logging verwaltet wird (LO_LOG) und die Zeiten der letzten Zugriffe protokolliert werden (LO_NOKEEP_LASTACCESS_TIME). Zudem steht das Flag für strikte Prüfung der Datenkonsistenz auf „high“ (LO_HIGH_INTEG).

Termine: INFORMIX 11.70 – New Features (IX3020DE)

Unsere Schulungsabteilung hat einen neuen 3-tägigen Kurs ins Angebot aufgenommen:

Informix 11.7 New Features

Der Kurs beinhaltet praktische Übungen zu vielen der neuen Features und ist daher die ideale Gelegenheit, sich mit diesen vertraut zu machen.

Aktueller Termin:

19.-21. September 2011 München

(ZMCG)

WebTipp: INFORMIX IWA 11.70.FC3 – Enhancements

Nicht nur bei der Datenbank Instanz selbst, sondern auch beim Warehouse Accelerator gibt es Neuerungen mit der Version 11.70.xF3

Hervorzuheben sind hier besonders folgende Neuerungen:

- Das „Smart Mart Tool“ das auf Grund eines ermittelten Lastprofils die Statements für die Erstellung der Tabellen des Data Mart erzeugen kann (probe2mart()). Mit Hilfe dieses Tools können auch die notwendigen XML-Strukturen erstellt werden (genmartdef()).
- Multi Locale Support im IWA. Die Einschränkung auf Datenbanken einer Locale sind damit aufgehoben.
- Die „Informix Ultimate Warehouse Time-Limited Version“ um die Funktionalität und Performance des IWA zu testen.

Mehr dazu findet man im Artikel:

https://www.ibm.com/developerworks/mydeveloperworks/blogs/newInformix/entry/informix_11_70xc3_and_warehouse_accelerator_enhancements30?lang=en

WebTipp: Informationen zum Informix Warehouse Accelerator

Im Bereich INFORMIX Developerworks findet sich ein BLOG, der nach und nach die Besonderheiten des Informix Warehouse Accelerators beleuchtet.

Jeder, der auch nur am Rande mit Datawarehouse zu tun hat, wird hier nützliche Informationen finden. Der Blog lebt und wird etwa einmal wöchentlich aktualisiert.

Der Blog ist zu finden unter:

<http://tinyurl.com/the-iwa-blog>

WebTipp: Free Informix – News zu INFORMIX

Die Seite freeinformix.com informiert über Nachrichten und Events rund um INFORMIX. Ein regelmässiger Blick auf die Seiten lohnt sich:

<http://www.freeinformix.com/news.html>

WebTipp: A smarter database for a smarter planet

Wer sich über die Vorteile von INFORMIX Timeseries interessiert, die bei den Lösungen zu Smart Metering eingesetzt werden, der sollte folgende Seite besuchen:

<http://smarterquestions.org/2011/06/using-informix-to-capture-timeseries-data-that-overwhelms-commodity-databases/>

Referenzen zu TimeSeries: Coldset Printing Partners

Coldset Printing Partners ist es gelungen dank INFORMIX Timeseries den Energieverbrauch um mehr als 10% zu senken. Mit Hilfe der Abfrage des Energieverbrauchs an allen Druckpressen im Intervall von 15 Minuten konnten die Spitzenverbraucher und die Energiefresser identifiziert und umgebaut werden.

Vor dem Einsatz von INFORMIX Timeseries waren nur die kumulierten Werte je Quartal verfügbar, die keinerlei Rückschluss auf die Details erlaubt haben.

Lesen Sie mehr dazu unter:

<http://public.dhe.ibm.com/common/ssi/ecm/en/imc14670usen/IMC14670USEN.PDF>

Anmeldung / Abmeldung / Anmerkung

Der Newsletter wird ausschließlich an angemeldete Adressen verschickt. Die Anmeldung erfolgt, indem Sie eine Email mit dem Betreff „**ANMELDUNG**“ an ifmxnews@de.ibm.com senden.

Im Falle einer Abmeldung senden Sie „**ABMELDUNG**“ an diese Adresse.

Neu hinzugekommen ist ein Archiv der INFORMIX Newsletters bei der International Informix User Group.

Das Archiv der bisherigen Ausgaben finden Sie zum Beispiel unter:

<http://www.iiug.org/intl/deu>

http://www.iug.de/index.php?option=com_content&task=view&id=95&Itemid=149

<http://www.informix-zone.com/informix-german-newsletter>

<http://www.drap.de/link/informix>

<http://www.nsi.de/informix/newsletter>

http://www.bytec.de/de/software/ibm_software/newsletter/

<http://www.cursor-distribution.de/index.php/aktuelles/informix-newsletter>

http://www.listec.de/Informix_Newsletter/

<http://www.bereos.eu/software/informix/newsletter/>

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit. Da uns weder Tippfehler noch Irrtümer fremd sind, bitten wir hier um Nachsicht falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

Die Autoren dieser Ausgabe

Gerd Kaluzinski IT-Specialist Informix Dynamic Server und DB2 UDB
 IBM Software Group, Information Management
gerd.kaluzinski@de.ibm.com +49-175-228-1983

Martin Fuerderer IBM Informix Entwicklung, München
 IBM Software Group, Information Management
martinfu@de.ibm.com

Sowie unterstützende Teams im Hintergrund.

Die Versionsinfo stammt aus dem Versions-Newsletter der CURSOR Software AG
<http://www.cursor-distribution.de/download/informix-vinfo>

Fotonachweis: Markus Holzbauer – Irschenberg, Bayern