

Willkommen zum Informix Newsletter

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Identifizieren von „unused Indexes“ in der Datenbank.....	2
TechTipp: Abfrage, ob eine Transaktion bereits gestartet wurde.....	4
TechTipp: Grant EXTEND - Berechtigung auf externen Funktionen.....	5
TechTipp: ONCONFIG - IFX_EXTEND_ROLE.....	5
TechTipp: ONCONFIG - SECURITY_LOCALCONNECTION.....	5
TechTipp: Environments - IFX_LARGE_PAGES.....	6
TechTipp: Neue Features in Version 14.10.xC7.....	7
TechTipp: Test der Migration auf 14.10.xC7 (InPlace von 14.10.xC6).....	8
TechTipp: Optionen des ONCHECK (oncheck -w #sec).....	8
TechTipp: Neue Tabellen in der Datenbank sysmaster.....	9
TechTipp: Optionen des ONSTAT (onstat -I).....	9
TechTipp: Neue Features im IBM INFORMIX Health Check (V1.05).....	10
TechTipp: InfoSphere CDC 11.4.0.4 wieder mit INFORMIX Engine.....	11
Nutzung des INFORMIX Newsletters.....	11
Die Autoren dieser Ausgabe.....	12

Aktuelles

Liebe Leserinnen und Leser,

eigentlich wäre jetzt die Zeit der Weihnachtsmärkte ...
 ... besser nicht nachdenken was es nicht gibt, sondern erfreuen an dem, was man hat. Statt draussen in der Kälte frierend einen überteuerten Glühwein aus billigstem Fusel zu trinken, kann man sich daheim selbst aus Wein, Zimtstangen, Nelken, Orangen und Äpfeln ein leckeres Getränk zaubern. Die Wärme vom Ofen spüren, während draussen der Schnee fällt, Lebkuchen, Kekse, zur Abwechslung mal eine überbackene Seele geniessen ...
 Das Leben ist so, wie man es sich macht.

Sich selbst und seinen Lieben eine Freude machen, und das Leben geniessen, ist besser als nur denken was nicht geht.
 In diesem Sinne: Bleiben Sie gesund !

Ihr TechTeam



TechTipp: Identifizieren von „unused Indexes“ in der Datenbank

Eine Anfrage, die immer wieder gestellt wird, ist, wie man ungenutzte Indexe in der Datenbank identifizieren kann.

Indexe sollen generell dabei helfen, schneller und gezielter Datensätze in einer Tabelle zu finden. Ziel ist eine gute Performance für den Zugriff auf die Daten. Teilweise werden Indexe angelegt, um doppelte Einträge zu vermeiden, aber dies ist ein anderes Thema. Das Entfernen von „Unique“ Indexen oder „Primary Keys“ sollte immer sehr gut abgestimmt sein.

Immer wieder begegnet man Entwicklern, die nach dem Motto „Viel hilft viel“ deutlich mehr als die unbedingt notwendigen Indexe anlegen. Zudem ändern sich Abfragen über die Zeit, und so werden bisher oft genutzte Indexe nicht mehr benötigt.

Zu viele Indexe können der Performance beim Lesen schaden, da der Optimizer für alle möglichen Zugriffswege die Kosten berechnen und vergleichen muss. Bei allen Schreibaktivitäten kosten zusätzliche Indexe Zeit, da die neuen Datensätze in allen Indexen eingepflegt werden müssen.

Indexe belegen Plattenplatz und können, je nach Art und Anzahl der enthaltenen Spalten, den freien Platz in den DBSpaces deutlich reduzieren.

Bis zur Version 14.10.xC5 gab es im Prinzip zwei Wege um ungenutzte Indexe zu identifizieren:

1. Anzahl der lesenden Operationen auf den Partitionen der Indexe (sysptprof)
 - > Hier muss der Einfluss der Rücksetzung der Statistiken beim Restart oder durch „onstat -z“ berücksichtigt werden. Eine Zeitmarke ist nicht ersichtlich.
2. Der „oncheck -pt“, der zu jedem Index die Zeitmarke der letzten Nutzung ausgibt.
 - > Der Aufruf ist recht komplex, da der „oncheck -pt“ einzeln je Tabelle erfolgen muss, und die Information dann mittels „grep“ zu extrahieren ist.

Beispiel für Option 2.:

```
oncheck -pt health_check:check_tabnames | grep -e Index -e Lookup
```

Ergebnis:

```

Index check_tabnames_i_001 fragment partition datadbs in
DBspace datadbs
Last Lookup/Scan Thu Nov 4 09:19:50 2021
Index check_tabnames_i_002 fragment partition datadbs in
DBspace datadbs
Last Lookup/Scan Wed Sep 1 12:48:53 2021
Index check_tabnames_i_003 fragment partition datadbs in
DBspace datadbs
Last Lookup/Scan Fri Oct 22 10:16:03 2021

```

Ab **Version 14.10.xC6** steht in der Tabelle „sysptnhdr“ das Feld „acc_or_mod_tm“ zur Verfügung, in dem eine Zeitmarke für den letzten Zugriff abgelegt wird.

An einem Beispiel wird gezeigt, wie einfach und übersichtlich diese Analyse dank dem neu eingeführten Feld in der Datenbank sysmaster nun ist.

Beispiel:

```
select
  n.dbsname,
  n.tabname,
  dbinfo('utc_to_datetime',p.acc_or_mod_tm) as wann,
  p.npused as pages
from sysptnhdr p, systabnames n
where p.acc_or_mod_tm != 0
  and p.partnum = n.partnum
  and n.dbsname[1,3] != 'sys'
  and n.dbsname = 'health_check'
  and p.lockid != p.partnum
  and nkeys = 1
order by wann desc
```

Ergebnis:

dbsname	tabname	wann	pages
health_check	check_tabnames_i_001	2021-11-04 09:19:50	7581
health_check	check_vplst_i_001	2021-11-04 09:19:49	9
health_check	check_cdrs_i_001	2021-11-04 09:19:49	3
health_check	check_seglst_i_001	2021-11-04 09:19:49	5
health_check	check_cluster_i_001	2021-11-04 09:19:49	3
health_check	check_shmvals_i_001	2021-11-04 09:19:40	4
health_check	check_cfg_i_001	2021-11-04 09:19:31	118
health_check	check_ha_nodes_i_001	2021-11-02 16:20:28	3
health_check	check_cdrtx_i_001	2021-10-22 14:34:35	3
health_check	check_cdr_rqm_i_001	2021-10-22 14:34:35	3
health_check	check_cdrprog_i_001	2021-10-22 14:34:35	3
health_check	check_cdr_state_i_001	2021-10-22 14:34:35	3
health_check	check_shmhdr_i_001	2021-10-22 14:34:07	92
health_check	check_tabnames_i_003	2021-10-22 10:16:03	53340
health_check	check_tabnames_i_002	2021-09-01 12:48:53	7838

Durch diese Abfrage erhält man schnell eine Übersicht über die Nutzung der Indexe.

Im obigen Beispiel ist zu sehen, dass der Index „check_tabnames_i_002“ schon seit mehr als zwei Monaten nicht mehr genutzt wurde (Abfrage am 04.11.2021).

In der Praxis spielen periodische Ereignisse eine Rolle. So kann es vorkommen, dass ein Index speziell für den Monatsabschluss oder sogar für Jahresabschluss oder Inventur angelegt wurde, der dann hier als „lange nicht genutzt“ erscheint. Diese Sonderfälle sollte abgeklärt werden, bevor ein Index gelöscht wird.

TechTipp: Abfrage, ob eine Transaktion bereits gestartet wurde

Im INFORMIX Newsletter Juni 2013 haben wir eine SPL-Procedure vorgestellt, mit deren Hilfe man herausfinden konnte, ob man sich bereits in einer offenen Transaktion befindet. Die Lösung damals war praktikabel, allerdings als „Tial&Error“ implementiert, bei der anhand der Rückmeldung beim „begin work“ entschieden wurde, ob eine Transaktion bereits offen ist.

Nun hat der HCL Advanced Support eine elegante Lösung beschrieben, wie man die Abfrage ausführen kann. Die Lösung basiert ebenfalls auf einer Funktion, nutzt aber interne Informationen.

Die Abfrage bedingt, dass eine Funktion erstellt wird, die eine bereits vorhandene C-Funktion aufruft, welche im Bereich der Datablades zur Verfügung steht:

```
create function if not exists txstate()
  returns int
  external name '(mi_transaction_state)'
  language C;
```

Befindet man sich in einer impliziten Transaktion, so wird der Wert 2 zurückgegeben:

```
select txstate() txstate;
Ergebnis: txstate 2
```

Befindet man sich in einer expliziten Transaktion, so erhält man den Rückgabewert 1:

```
begin work;
select txstate() txstate;
Ergebnis: txstate 1
```

Befindet man sich nicht in einer Transaktion, so wird der Wert 0 zurückgegeben:

```
rollback work;
execute function txstate();
Ergebnis: (expression) 0
```

Hinweis:

Da diese Funktion nicht offiziell dokumentiert ist, könnte sich der Aufruf in zukünftigen Versionen ändern.

TechTipp: Grant EXTEND - Berechtigung auf externen Funktionen

Wer die Funktion aus dem vorigen Artikel als „normaler User“ erstellen will, der wird erst einmal mit einer Fehlermeldung konfrontiert:

```
25785: Cannot create external routine (txstate) without the
EXTEND role.
```

Als Default ist das Recht Datablades zu registrieren, oder externe Funktionen in eine Datenbank einzubinden, nur an den Benutzer „informix“ vergeben.

Die Kontrolle darüber wird über den Parameter IFX_EXTEND_ROLE der \$ONCONFIG gesteuert. (siehe folgender Artikel).

Zudem kann das Recht vom Benutzer „informix“ an weitere Datenbankuser weitergegeben werden:

```
grant EXTEND to kalu
```

Danach lässt sich auch die Funktion erstellen:

```
>>> Routine created.
```

TechTipp: ONCONFIG - IFX_EXTEND_ROLE

Die Berechtigung, DataBlades in einer Datenbank zu registrieren oder externe Funktionen zu erstellen, wird über den Parameter IFX_EXTEND_ROLE gesteuert.

Der Parameter kann folgende Werte annehmen:

- | | |
|-------------|---|
| 0 | Jeder Benutzer darf externe Funktionen in der Datenbank erstellen und Datablades registrieren. |
| 1 (Default) | Nur der Benutzer „informix“ und die Benutzer, denen das Recht „EXTEND“ gegeben wurden, dürfen externe Erweiterungen in der Datenbank erstellen bzw. registrieren. |

Der Parameter ist dynamisch mittels „onmode -wf“ bzw. „onmode -wm“ änderbar.

TechTipp: ONCONFIG - SECURITY_LOCALCONNECTION

Der Parameter steuert, ob die Informix Instanz bei einem lokalen Connect eine Überprüfung des Benutzers durchführt.

Mögliche Werte sind:

- | | |
|---|--|
| 0 | Keine Sicherheitsüberprüfung wird vorgenommen |
| 1 | Die Informix Instanz überprüft ob die ID, mit der die Verbindung aufgenommen werden soll, der des Benutzers entspricht. |
| 2 | wie 1., zusätzlich wird der Port kontrolliert, ob dieser tatsächlich dem PeerPort der Applikation entspricht, die die Verbindung erstellen will. |

Der Parameter ist nur durch einen Neustart der Informix Instanz änderbar.

TechTipp: Environments - IFX_LARGE_PAGES

Auf aktuellen Linux Plattformen ist das Feature der HUGE PAGES im Kernel (ab 2.6) bereits enthalten. Der Default für die Pagesize auf Linux ist 4kB. Wird viel Memory z.B. durch die Datenbank genutzt, so werden sehr viele Einträge beim Memory Mapping benötigt. Bei Large Pages (Huge Pages) ist die Default Size 2MB. Damit reduziert sich die Anzahl der Mapping Einträge, und es werden grössere Blöcke in einer Aktion gelesen und geschrieben.

Um die HUGE PAGES nutzen zu können, müssen diese vom Administrator bereitgestellt werden. Dies geschieht mittels:

```
sysctl -w vm.nr_hugepages=<Anzahl der Huge Pages>
```

Ob die Huge Pages bereits eingerichtet sind, lässt sich ermitteln durch:

```
cat /proc/meminfo | grep -i huge
```

```
AnonHugePages:      0 kB
ShmemHugePages:    0 kB
FileHugePages:     0 kB
HugePages_Total:   2371
HugePages_Free:    2142
HugePages_Rsvd:    1991
HugePages_Surp:    0
Hugepagesize:      2048 kB
Hugetlb:           4855808 kB
```

Hier ist die PageSize von 2048 kB zu sehen und die Anzahl an konfigurierten Pages. Damit sind knapp 5 GB für die Nutzung durch INFORMIX verfügbar.

Für die Huge Pages wird ein separater Bereich gemountet:

```
mount | grep huge
cgroup on /sys/fs/cgroup/hugetlb type cgroup
(rw,nosuid,nodev,noexec,relatime,hugetlb)
hugetlbfs on /dev/hugepages type hugetlbfs
(rw,relatime,pagesize=2M)
```

Die Huge Pages sind auch im Memory Mapping zu sehen:

```
grep huge /proc/*/numa_maps
/proc/97297/numa_maps:44000000 default
file=/SYSV52574801\040(deleted) huge dirty=3 mapmax=12 N0=3
kernelpagesize_kB=2048
/proc/97297/numa_maps:44600000 default
file=/SYSV52574802\040(deleted) huge dirty=22 mapmax=12 N0=22
kernelpagesize_kB=2048
/proc/97297/numa_maps:4c400000 default
file=/SYSV52574803\040(deleted) huge dirty=5 mapmax=12 N0=5
kernelpagesize_kB=2048
/proc/97298/numa_maps:44000000 default
file=/SYSV52574801\040(deleted) huge dirty=3 mapmax=12 N0=3
kernelpagesize_kB=2048
/proc/97298/numa_maps:44600000 default ...
```

Um bei INFORMIX nun diese HUGE PAGES zu nutzen, muss vor dem Start der INFORMIX Instanz der Parameter IFX_LARGE_PAGES auf 1 gesetzt werden (wobei der Wert 1 für enable und 0 für disable steht). Zudem muss in der ONCONFIG der Parameter RESIDENT auf einen Wert ungleich 0 gesetzt werden. Empfohlen wird hier der Wert -1 für „alle Segmente“ oder ein Wert der höher ist, als die Anzahl der Memory Segmente, die die Instanz allokiert.

Wird die Instanz mit diesen Parametern gestartet, so ist die Nutzung bereits im online.log beim Start zu sehen:

```
15:33:09 IBM Informix Dynamic Server Started.
15:33:09 Requested shared memory segment size rounded from 4308KB
to 6144KB
15:33:09 Shared memory segment will use huge pages.
15:33:11 Requested shared memory segment size rounded from
44826KB to 45056KB
15:33:11 Shared memory segment will use huge pages.
15:33:11 Segment locked: addr=0x4c400000, size=46137344
15:33:11 Successfully added a bufferpool of page size 2K.
```

Ist nicht genügend Platz in den Huge Pages verfügbar, wenn die Instanz weitere Virtuelle Memory Segmente allokiert, so werden die Pages aus dem normalen Linux Memory Pool genommen und es wird im online.log eine Warnung ausgegeben, dass nicht genug huge pages verfügbar sind:

```
16:37:27 IBM Informix Dynamic Server Version 14.10.FC7AEE
16:37:27 Warning: Server is unable to lock huge pages in memory.
Switching to normal pages.
16:37:27 Segment locked: addr=0x49200000, size=8388608
16:37:27 Dynamically allocated new virtual shared memory segment
(size 8192KB)
```

Für Solaris hatten wir dieses Feature LARGE PAGES bereits im INFORMIX Newsletter Q4-2016 beschrieben.

TechTipp: Neue Features in Version 14.10.xC7

Im November wurde die Version 14.10.xC7 bereitgestellt.

Wie in jedem neuen Release sind viele Bereinigungen und Verbesserungen enthalten, die auf das vorhergehende Releases basieren.

Die Neuen Features in diesem Release sind:

- Die interne IBM Java Version wurde aktualisiert auf 8.0.6.36 (SunOS: 8.0.6.35)
- Die GSKit Installation wurde auf 8.0.55.25 erhöht.
- Neue Option im ONCHECK: oncheck -w [#seconds]
- Reverse Scan metrics wurden dem onstat -g tpf hinzugefügt.
- Informationen über die Verwendung der Indexe ist nun mit „onstat -l“ verfügbar.
- Details zum ReadAhead wurden im „onstat -g rah“ hinzugefügt.
- Der „deployment assistant (DA)“ ist nicht mehr unterstützt ab Version 14.10.xC7.

TechTipp: Test der Migration auf 14.10.xC7 (InPlace von 14.10.xC6)

Die Migration InPlace verlief bei allen Tests problemlos und schnell. Die Konfigurationsdatei onconfig.std unterscheidet sich nur in den Beschreibungen, nicht bei den Parametern. In der Beschreibung wurde zur IBM nunmehr auch HCL aufgenommen. Zudem erfolgte die Bereinigung vieler Tippfehler oder syntaktische Korrekturen an der Satzstellung.

Eine Beobachtung während der Migration war, dass der Parameter **MIRRORPATH** zum Spiegel des ROOTDBS nun auch beim Wert MIRROR = 0 beachtet wird, und der bestehende Eintrag entfernt wird falls keine Spiegelung eingerichtet ist. Während der Migration wird dies als „ERROR“ ausgegeben, was aber nicht zu einem Abbruch führt:

```
Could not open mirror root chunk. Errno 2.  
succeeded  
11/08/2021 11:05:00 The ROOT DBspace is unmirrored. Unsetting  
MIRRORPATH.
```

Ursache dieses Verhaltens ist, dass der Parameter MIRROR nicht mehr erforderlich ist, falls eine Spiegelung des ROOTDBS gewünscht wird. Es reicht den Spiegel mittels „onspaces“ einzurichten.

Eine weitere gute Nachricht der Tests:

Eine bestehende Replikation HDR bzw. RSS bleibt auch nach der Migration erhalten und repliziert zuverlässig weiter.

TechTipp: Optionen des ONCHECK (oncheck -w #sec)

Der oncheck zur Überprüfung der Konsistenz unterschiedlichster Teile der Datenbank konnte auf stark frequentierten Systemen nicht komplett genutzt werden, da oft Sperren im System den Check blockierten und dieser daher mit einer Fehlermeldung abgebrochen ist. Mit der Version 14.10.xC7 wurde ein neuer Parameter -w eingeführt, mit dessen Hilfe man dem Check mitteilen kann, dass bei Sperren statt eines Abbruchs gewartet werden soll. Hierbei gibt es die Optionen „unendliches Warten“, wie es z.B. bei der Überprüfung des Systemkataloges sinnvoll ist:

```
oncheck -cc health_check -w
```

sowie die Einschränkung der Wartezeit auf eine vorgegebene Anzahl an Sekunden:

```
oncheck -cDI health_check:check_tabnames -w 42
```

wobei im Beispiel maximal 42 Sekunden gewartet wird, ob der Check ohne durch eine Sperre blockiert zu werden, den Zugriff auf die Daten und Indices erhält.

TechTipp: Neue Tabellen in der Datenbank sysmaster

Für die detaillierte Ausgabe der Informationen zum ReadAhead wurden neue Tabellen in die Datenbank sysmaster aufgenommen:

```
{ Read Ahead Queues }           Tabelle: sys_ra_act_queues
{ Read Ahead Threads }         Tabelle: sys_ra_act_daemons
{ Partition Read Ahead Stats } Tabelle: sys_ra_ptn_stats
{ Thread Read Ahead Stats }    Tabelle: sys_ra_thrđ_stats
```

Auf die Analyse der Inhalte werden wir in einer der kommenden Ausgaben des INFORMIX Newsletters näher eingehen.

TechTipp: Optionen des ONSTAT (onstat -l)

Die neue Option „-l“ (Grosses -l, nicht klein L) im „onstat“ bietet einen Überblick über die Aktivitäten auf Indexen aller Datenbanken einer Informix Instanz.

Da die Daten für eine gesamte Instanz zusammengefasst sind, werden die Werte für Search, Additem und Delitem sicher sehr schnell anwachsen.

Bei der Performance Analyse wird der Augenmerk eher auf der Zeile der Splits liegen, in der die Anzahl unter „root“ einen besonderen Augenmerk verdient. Jeder „Root-Split“ ist Ergebnis der Erweiterung eines Indexbaumes um ein Level.

Beispiel der Ausgabe:

```
IBM Informix Dynamic Server Version 14.10.FC7AEE -- On-Line (Prim)
-- Up 04:04:19 -- 115048 Kbytes
2021-11-08 15:09:12
```

```
Index statistics
search      additem    delitem    retry      rev_scans  rs_retries
101275      46707     7019      4           4           0

splits      page       slot       root       copyback
511         477       16         102        0

compress    merges     shuffles   root
17          0          0          0
```

TechTipp: Neue Features im IBM INFORMIX Health Check (V1.05)

Der Informix Health Check bekam mit der Version 1.05 neue Features

- Connection Manager Information
- DBSpaces erweitert um ExtendSize und MaxSize
- GLS Information der Datenbanken hinzugefügt
- Server Environment Variablen werden mit ausgegeben

Connection Manager Information:

HDR/RSS/SDS Replikation

ACTIVE								
ixdb1	PRIMARY	Updates: Y	Status: Active	Connection:	DelayApply:	LOG Diff: 0	Acknowledge: 2021-10-26 13:54:39	OK
ixdb1_n	HDR	Updates: Y	Status: on	Connection: Connected	DelayApply:	LOG Diff: 0	Acknowledge:	OK
STATUS								
Server: ixdb1	Typ: Primary		Status: Active					OK
Server: ixdb1_n	Typ: HDR		Status: On					OK

CDR Replikation

Keine CDR Replikation

CONNECTION MANAGER

CONNECTION MANAGER						
NAME	UNIT	HOST	TYPE	SERVERS	FOC	
sdemoB_CM	g_ixdb	sdemo1.demo-gmbh.de	CLUSTER	ixdb1,ixdb2,ixdb1_n	ORDER=HDR,RSS PRIORITY=3 TIMEOUT=1	
	UNIT	SLA	SLA Define	Connections		
	g_ixdb	ixdb	DBSERVERS=primary	55598		
	g_ixdb	ixdbread	DBSERVERS=(primary,HDR)	0		
NAME	UNIT	HOST	TYPE	SERVERS	FOC	
sdemoB_n_CM	g_ixdb	sdemo2.demo-gmbh.de	CLUSTER	ixdb1,ixdb1_n	ORDER=HDR,RSS PRIORITY=1 TIMEOUT=1	
	UNIT	SLA	SLA Define	Connections		
	g_ixdb	ixdb	DBSERVERS=primary	0		
	g_ixdb	ixdbread	DBSERVERS=(primary,HDR)	0		

Um die Informationen zum AutoExtend der DBSpaces zu sehen, kann in der Datenbank sysmaster die Tabelle sysdbspaces abgefragt werden.

In der Auswertung ist dies in der letzten Spalte der DBSpaces zu sehen:

DBSPACES	(Schwellenwerte: 30%, 15%)			
DBSPACE: 001 - rootdbs	(pagesize: 2048) - Size in GB: 0,29	FREE %: 67,81	OK	
DBSPACE: 002 - logdbs	(pagesize: 2048) - Size in GB: 0,95	FREE %: 0,99	OK	LOGSPACE
DBSPACE: 003 - sbdbs	(pagesize: 2048) - Size in GB: 1,91	FREE %: 5,01	OK	BLOBSPACE
DBSPACE: 004 - tmpdbs	(pagesize: 2048) - Size in GB: 0,95	FREE %: 99,92	OK	TEMPSPACE
DBSPACE: 005 - datadbs	(pagesize: 2048) - Size in GB: 2,90	FREE %: 12,58	WARN	EXTENDABLE ExtendSize (MB): 20 MaxSize(MB): 7812
DBSPACE: 006 - sbdbs2	(pagesize: 2048) - Size in GB: 0,01	FREE %: 5,36	OK	BLOBSPACE
DBSPACE: 007 - data4k	(pagesize: 4096) - Size in GB: 0,02	FREE %: 26,20	WARN	
CHUNKS	(Schwellenwerte: tbd%, tbd%)			
CHUNK: 001:001 - rootdbs	Pfad: /DBS/rootdbs	FREE %: 67,81	ONLINE	CONTAINS PHYSLOG
CHUNK: 002:002 - logdbs	Pfad: /DBS/logdbs	FREE %: 0,99	ONLINE	CONTAINS LOGS
CHUNK: 003:003 - sbdbs	Pfad: /DBS/sbdbs	FREE %: 5,01	ONLINE	
CHUNK: 004:004 - tmpdbs	Pfad: /DBS/tmpdbs	FREE %: 99,92	ONLINE	
CHUNK: 005:005 - datadbs	Pfad: /DBS/datadbs	FREE %: 12,58	ONLINE	EXTENDABLE
CHUNK: 006:006 - sbdbs2	Pfad: /DBS/sbdbs2	FREE %: 5,36	ONLINE	
CHUNK: 007:007 - data4k	Pfad: /DBS/data4k	FREE %: 26,20	ONLINE	

TechTipp: InfoSphere CDC 11.4.0.4 wieder mit INFORMIX Engine

Die Datenreplikation mittels InfoSphere CDC, die als Erweiterung zur Replikation zwischen INFORMIX Instanzen eine Replikation in andere Datenbanksysteme, das Filesystem oder KAFKA ermöglicht, ist bei immer mehr Kunden im Einsatz.

Als bei CDC die Version 11.4 veröffentlicht wurde, kamen Fragen auf, warum in 11.4 die INFORMIX Komponente nicht mehr im Bundle dabei ist. Da bei CDC eine Replikation zwischen Komponenten unterschiedlicher Versionen unterstützt ist, war dies kein akutes Problem.

Der reale Hintergrund war, dass sich beim internen Layout der Transaktionslogs in Informix kaum etwas geändert hatte, so dass hier keine Anpassungen notwendig waren. Warum die ältere Version nicht einfach mit neuer Version ins Bundle aufgenommen wurde, ist unklar.

In Version 11.4 war der Hauptaugenmerk auf der Engine für Kafka, die in den Subreleases weiter verbessert wurde. Zudem mussten bei der Komponenten für Oracle Anpassungen eingeführt werden, um die Verschlüsselung der Logs mit einzubeziehen.

Mit der aktuell freigegebenen Version 11.4.0.4 von InfoSphere CDC ist auch die Informix Engine wieder im Bundle enthalten und hat einige Verbesserungen erfahren.

```
./dmshowversion
```

```
VERSIONSINFORMATIONEN
```

```
Produkt: IBM InfoSphere Data Replication (IBM Informix Dynamic Server)
```

```
Version: 11.4.0.4
```

```
Build: master_5611
```

```
Datum: 30.11.2021
```

Wer mehr zu InfoSphere CDC erfahren will, der findet in den Newsletter-Ausgaben 11/2009, 03/2012 und 10/2014 hierzu weitere Hintergründe. Falls Interesse besteht, dann könnte diese Variante der Replikation hier im Newsletter nochmals genauer vorgestellt werden.

Für aktuelle Fragen einfach eine Mail an gerd.kaluzinski@de.ibm.com senden.

Nutzung des INFORMIX Newsletters

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit.

Die IUG hat sich dankenswerterweise dazu bereit erklärt, den INFORMIX Newsletter auf ihren Web Seiten zu veröffentlichen.

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.

Rückmeldungen hierzu sind herzlich Willkommen !

Die gefundenen Tippfehler dürfen zudem behalten und nach Belieben weiterverwendet werden.

Eine Weiterverbreitung in eigenem Namen (mit Nennung der Quelle) oder eine Bereitstellung auf der eigenen HomePage ist ausdrücklich erlaubt. Alle hier veröffentlichten Scripts stehen uneingeschränkt zur weiteren Verwendung zur Verfügung.

Die Autoren dieser Ausgabe

Andreas Legner	(Technische Unterstützung) INFORMIX Advanced Support HCL Technologies	
Martin Fuerderer	(Technische Unterstützung und Fehlerkorrektur) INFORMIX Development HCL Technologies	
Gerd Kaluzinski	(Zusammenstellung und Technische Artikel) DataOps Application Consultant IBM Software gerd.kaluzinski@de.ibm.com	+49-175-228-1983

Dank auch an die vielen Helfer im Hintergrund.

Nicht zu vergessen der Dank an die Informix User Group, ohne die es keinen Neuanfang des INFORMIX Newsletters gegeben hätte und die dankenswerter Weise die Verteilung übernimmt.

Foto Nachweis:

Kekse für die Redaktion ... Danke an Thekla Schneider !

(Gerd Kaluzinski)