

Willkommen zum Informix Newsletter

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Features 14.10.xC8.....	2
TechTipp: Rolling Upgrade and Connection Manager.....	2
TechTipp: ONCONFIG - ARCHIVE_UNLOGGED_OBJECTS.....	3
TechTipp: ARCHECKER – Restore Filter beim Table-Level-Restore.....	7
TechTipp: ARCHECKER – Distributed Restore.....	8
TechTipp: LOCOPY() - Move Smart Large Objects to a different SBSpace.....	9
TechTipp: ONCONFIG - AUTO_READAHEAD.....	10
TechTipp: ONCONFIG - TLS_VERSION.....	11
TechTipp: CDR Replikation nach UTF-8.....	11
TechTipp: Auslieferung der Komponente JRE endet.....	11
Nutzung des INFORMIX Newsletters.....	12
Die Autoren dieser Ausgabe.....	12

Aktuelles

Liebe Leserinnen und Leser,

kurz vor den Sommerferien erhalten Sie die zweite Ausgabe des INFORMIX Newsletters für dieses Jahr.

Der Sommer verspricht sonnig und erholsam zu werden. Damit auch die Datenbank Administratoren unter den Lesern beruhigt in Urlaub gehen können, sollte noch überprüft werden, ob die Sicherungen eingerichtet und konsistent sind. Dies kann z.B. mit dem Archecker erfolgen.

Table Level Restore mit dem Archecker, sowie Neue Features der Version 14.10.xC8 bilden den Kern dieser Ausgabe.

Viel Spass beim Lesen und Gute Erholung !

Ihr TechTeam



TechTipp: Features 14.10.xC8

Mit der Version 14.10.xC8 kamen wie gewohnt einige Bereinigungen und Verbesserungen in das Produkt. Zudem sind auch neue Features mit dabei.

Hier eine kurze Übersicht der wichtigsten Änderungen:

- Die intern genutzte Java Version wurde auf 8.0.7.0 upgedatet.
Ausnahme: HP-UX, bei der weiterhin die Version 8.0.6.30 genutzt wird.
- Wichtiger Hinweis zum Rolling Upgrade (*)
- Neuer Parameter ARCHIVE_NOLOG_OBJECTS beim „onbar“ (*)
- Erweiterung beim Parameter AUTO_READAHEAD (*)
- Transport Layer Security (TLS) Version 1.0 / 1.1 werden nicht mehr unterstützt (*)
- In der Konfiguration des „Wire Listeners“ wird der Parameter „command.blacklist“ zukünftig nicht mehr unterstützt. Neu kann nun die Option „command.denylist“ verwendet werden.

(*) Details hierzu finden sich in einem TechTipp.

TechTipp: Rolling Upgrade and Connection Manager

Mit der Version 12.10.xC5 wurde der Rolling Upgrade eingeführt, bei dem die neue Version (nur das nächst höhere Patch-Level) oder ein FixPack zur selben Version bei Verwendung der (HDR/RSS/SDS) Replikation im laufenden Betrieb eingespielt werden können.

Hierbei werden nach und nach die Instanzen auf die neue Version gehoben, wobei die Verbindungen zur Datenbank über den ConnectionManager auf die restlichen Instanzen umgeleitet werden.

Zu beachten ist nun, dass mit Version 14.10.xC8 zuerst alle Instanzen des ConnectionManagers auf die neue Version gehoben werden (im Softwarepaket C-SDK 4.50.xC8 enthalten) müssen, bevor die Informix Instanzen mit der neuen Version versehen werden.

Als Reihenfolge zur Installation der neuen Software wird empfohlen:

1. Remote Standalone Server (RSS)
2. HDR Secondary Server
3. SDS Secondary Server
4. Primary Server

Soll die ursprüngliche Konstellation wieder hergestellt werden, so ist am Ende der Befehl „onmode -d make primary <instanz>“ auf dem früheren Primary Server notwendig, damit dieser wieder seine alte Rolle übernimmt.

TechTipp: ONCONFIG - ARCHIVE_UNLOGGED_OBJECTS

Steht auf dem Remote Secondary Server (RSS) der Konfigurationsparameter `BAR_SEC_ALLOW_BACKUP` auf 1, so ist es möglich die Sicherungen mittels `ONTAPE` oder `ONBAR` nicht nur am Primary Server, sondern auch am RSS zu erstellen.

Voraussetzung ist, dass ein Temporärer DBSpace auf dem RSS existiert und als `DBSPACETEMP` in der Konfiguration eingetragen ist.

Der Parameter kann folgende Werte annehmen:

- 0 Eine Sicherung kann nicht erstellt werden wenn Objekte ohne Transaktionslogging existieren. (Default)
- 1 Eine Sicherung wird auch erstellt, falls es Objekte ohne Logging gibt. Es wird in diesem Fall eine allgemeine Warnung ausgegeben.
- 2 Eine Sicherung wird erstellt und es erfolgt eine detaillierte Warnung falls Objekte ohne Logging existieren.

Beispiel mit einem SBSPACE ohne Logging:

RSS: `ontape -s -L 0`

Das Backup bricht ab (beim Default Wert) mit:

```
„(-83380) An archive checkpoint could not be completed“
```

Im `online.log` steht:

```
The storage space, 'sbdbs', is preventing the backup on the secondary server.
```

```
(-83380) An archive checkpoint could not be completed in the secondary server.
```

Ändert man nun den Parameter auf 1 mittels:

```
onmode -wf ARCHIVE_UNLOGGED_OBJECTS=1
```

```
Value for ARCHIVE_UNLOGGED_OBJECTS (1) was saved in config file.
```

```
Value of ARCHIVE_UNLOGGED_OBJECTS has been changed to 1.
```

so läuft die Sicherung durch ohne weitere Meldungen:

```
ontape -s -L 0
100 percent done.
File created: /home/informix/BACKUP/kalu42_3_L0
```

```
Please label this tape as number 1 in the arc tape sequence.
```

```
This tape contains the following logical logs:
```

```
8
Program over.
```

Im `online.log` findet sich danach folgende Warnung:

```
WARNING: This archive contains unlogged objects such as raw tables
         or BLOBspace blobs and is therefore incomplete because these objects
         are not fully replicated. If restored from this archive these objects
         will be internally inconsistent and must be recreated before they
         can be used.
```

Wird eine der unterschiedlichen Möglichkeiten der Replikation verwendet, so wird dringend empfohlen den SmartBlobSpace mit Logging zu betreiben. Auch ohne Replikation wäre diese Option vorteilhaft für die Behandlung von Transaktionen.

Nun ändern wir den SmartBlobSpace auf Logging:

```
onspaces -ch sdbds -Df LOGGING=ON
```

```
You are turning on logging - REMEMBER TO DO A LEVEL 0 ARCHIVE !!!
```

```
These changes will become the new sbspace default values.
```

```
Do you really want to continue? (y/n)y
```

```
Changes have been made to the sbspace.
```

```
Now the utility will read and update 1000 smart large objects at a time and
```

```
commit each block of 100 smart large objects as a single transaction.This
```

```
utility may take a long time to complete.Do you really want to continue? (y/n)y
```

Der Befehl ist auch auf dem RSS notwendig, falls dieser nicht durch einen Neustart die Änderung des Parameters vom Primary übertragen bekommt !

```
onspaces -ch sdbds -Df LOGGING=ON
```

```
You are turning on logging - REMEMBER TO DO A LEVEL 0 ARCHIVE !!!
```

```
These changes will become the new sbspace default values.
```

```
Do you really want to continue? (y/n)y
```

```
Changing the sbspace characteristics has resulted in error: 0
```

Zweiter Test mit einer Datenbank ohne Logging:

```
The nonlogging database, 'nologdb', is preventing the backup on the  
secondary server.
```

Nun ändern wir den Parameter ARCHIVE_UNLOGGED_OBJECTS auf den Wert 2.
Das Backup läuft durch, und im online.log wird eine ausführliche Meldung ausgegeben:

```
The nonlogging database, 'nologdb', would normally prevent  
a backup on the secondary server.
```

```
These restrictions have been overridden by the user.
```

```
WARNING: This archive contains unlogged objects such as raw tables
```

```
or BLOBspace blobs and is therefore incomplete because these objects  
are not fully replicated. If restored from this archive these objects  
will be internally inconsistent and must be recreated before they  
can be used.
```

```
Level 0 Archive started on rootdbs, datadbs, logdbs, sdbds
```

```
Archive on rootdbs, datadbs, logdbs, sdbds Completed.
```

Hinweis:

Eine Replikation (HDR/RSS) verhindert, dass Tabellen aus dem Logging genommen werden, was auch implizit bei der Nutzung des HighPerformanceLoaders (onpladm) genutzt wird.

Die Meldung lautet:

```
alter table test1 type (raw);
```

```
19845: You cannot alter the logging mode of a table in a logged database on a  
primary
```

TechTipp: SPL - Globale Variablen in Stored Procedures

In der Informix Stored Procedures Language (SPL) gibt es die Möglichkeit lokale Variablen zu definieren. Diese sind nur innerhalb der Prozedur verfügbar und können nicht nach ausserhalb der Prozedur übergeben werden.

Die Parameter werden als „call by value“ übergeben, wobei nur der Wert, nicht aber der Parameter in die Prozedur übergeben wird. Dies gilt ebenso für die Rückgabewerte. Bei lokalen Variablen kann ein Defaultwert angegeben werden, dies ist jedoch nicht verpflichtend.

Beispiel:

```
define var1 int default 42;
define var2 char(42);
define var3 date default today;
```

Um Variablen in mehreren Prozeduren der selben Session gemeinsam nutzen zu können, gibt es die Option Variablen GLOBAL zu definieren. Globale Variablen müssen mit einem Defaultwert definiert werden.

Beispiel:

```
define GLOBAL gvar1 int DEFAULT 42;
```

Der Default wird nur bei der ersten Definition der Globalen Variablen gesetzt. Wird die Variable anschliessend wieder verwendet, sei es in der selben oder einer anderen Prozedur, so bleibt der aktuell Wert erhalten.

Das folgende Beispiel verdeutlicht die Funktionsweise:

```
create procedure proc_1()
returning int, char(42), date
define global gvar1 int default 42;
define global gvar2 char(42) default '1';
define global gvar3 date default today;
return gvar1, gvar2, gvar3 with resume;
let gvar1 = gvar1 + 3;
let gvar2 = trim(gvar2)||'-'||gvar1;
let gvar3 = gvar3 - 20 units day;
return gvar1, gvar2, gvar3;
end procedure;
```

Der Aufruf dieser Prozedur mehrfach nacheinander setzt nur beim ersten Aufruf den Defaultwert.

```
execute procedure proc_1();
(expression) (expression) (expression)
42 1 18.05.2022
45 1-45 28.04.2022
```

Der Defaultwert 42 wurde um +3 erhöht, daher 45.

Der Character String wurde mit dem Default „1“ belegt und es wurde die „45“ angefügt.

Der Default „today“ wurde als Datum um 20 Tage zurück gesetzt.

Wird die Prozedur wiederholt aufgerufen, so werden die bereits belegten globalen Variablen verwendet und es findet keine neue Definition statt:

```
execute procedure proc_1();
(expression) (expression) (expression)

      45 1-45                28.04.2022
      48 1-45-48            08.04.2022

execute procedure proc_1();
(expression) (expression) (expression)

      48 1-45-48            08.04.2022
      51 1-45-48-51        19.03.2022
```

Hätten wir die Variablen lokal definiert, so wäre bei jedem Aufruf der Default neu vergeben worden, da die Variablen nur innerhalb der Prozedur existieren.

Definieren wir eine weitere Prozedur, die die selben globalen Variablen definiert und nutzt, so werden trotzdem die bestehenden Variablen auch hier genutzt.

```
create procedure proc_2()
returning int, char(42), date
define global gvar1      int default 11;
define global gvar2      char(42) default 'XXX';
define global gvar3      date default "21.02.1965";

return gvar1, gvar2, gvar3 with resume;

let gvar1 = gvar1 + 3;
let gvar2 = trim(gvar2)||'+'||gvar1;
let gvar3 = gvar3 + 2 units day;

return gvar1, gvar2, gvar3;
end procedure;
```

Aufruf der Prozeduren im Wechsel (proc1–proc2–proc1–proc2)

```
(expression) (expression) (expression)
      42 1                18.05.2022
      45 1-45            28.04.2022
(expression) (expression) (expression)
      45 1-45            28.04.2022
      48 1-45+48        30.04.2022
(expression) (expression) (expression)
      48 1-45+48        30.04.2022
      51 1-45+48-51    10.04.2022
```

Damit ist die Reihenfolge wichtig, welche Prozedur zuerst aufgerufen wird, da hiermit bestimmt wird, welche Defaultwerte gesetzt werden.

Im Beispiel ist zu sehen dass der Wert 11 als Default für gvar1 nie verwendet wird, da die Prozedur proc1() die Definition durchgeführt hat.

TechTipp: ARCHECKER – Restore Filter beim Table-Level-Restore

In letzter Zeit kam es mehrfach vor, dass Kunden aus Versehen zu viel Daten gelöscht oder irrtümlicherweise zu viele Datensätze mit einem Update geändert hatten. Dank des Table-Level-Restore im Tool „archecker“ konnten wir in allen Fällen helfen, sind aber auf eine Restriktion gestossen, die keinem der Kollegen bewusst war: „Das Filtern der Datensätze beim Table-Level-Restore ist nur ohne Logical Restore erlaubt“.

Dies bedeutet, dass man entweder den Restore einer Tabelle für einen exakten Zeitpunkt definieren kann (z.B. „2022-05-18 23:42:02“) und dann Alle Datensätze bekomme, oder man nutzt eine Where-Bedingung, die aber nur ohne Log Recovery erlaubt ist, und damit den Zustand zum Start der Sicherung bietet.

Beispiel eines AC_SCHEMA File:

```
database stores;
create table customer
(
  customer_num serial not null ,
  fname char(15),
  lname char(15),
  ...
  phone char(18),
  primary key (customer_num)
);
create table customer_neu
(
  customer_num integer not null ,
  fname char(15),
  lname char(15),
  ...
  phone char(18)
);

insert into customer_neu select * from customer where customer_num < 110;
restore to current with no log;
```

Dies ermöglicht den gezielten Restore weniger Datensätze auf den Stand der Sicherung.

```
insert into customer_neu select * from customer;
restore to "2022-05-18 23:42:02";
```

Dieser Aufruf holt alle Datensätze der Tabelle in die neue Tabelle zurück, und fährt nach dem Restore der Sicherung noch alle Transaktionen auf der Tabelle bis zum angegebenen Zeitpunkt nach.

Leider ist die Kombination aus beiden Optionen (derzeit) nicht unterstützt.

TechTipp: ARCHECKER – Distributed Restore

Der Archecker kann dazu genutzt werden, um Daten einer Sicherung auf entfernte Informix Instanzen zu übertragen. Dabei können sich die Remote Instanzen in der Version, und beim Betriebssystem, von der lokalen Installation unterscheiden.

Das Vorgehen ist dabei ganz einfach:

Es wird ein Table-Level-Restore definiert, bei dem die Zieltabellen in einer entfernten Datenbank aufgebaut werden sollen. Dies wird durch die Angabe der Datenbank mittels „database <dbname>“ vor der Tabellendefinition in der AC_SCHEMA Datei angegeben. Dass diese Datenbank REMOTE ist, wird erst beim „insert“ Befehl durch die Angabe der Distributed Syntax angegeben.

Beispiel:

```
database remote_database_xxx
create table kunden
  ( columns );

database test1;
create table customer
  ( columns ) in dbspace1;

insert into remote_database_xxx@remote_srv.kunden
  select * from customer;
```

Der Table Level Restore holt nun die Daten aus der Sicherung die zur Tabelle „customer“ gehören, und speichert diese in der Tabelle „kunden“ auf dem Remote Server in der Remote Datenbank.

Einzigste Voraussetzung ist, dass die entfernte Instanz (hier „remote_srv“) in der Datei sqlhosts steht, und ein passwortfreier Zugriff für den Benutzer „informix“ möglich ist.

TechTipp: LOCOPY() - Move Smart Large Objects to a different SBSpace

Im INFORMIX Newsletter Q1 aus 2021 hatten wir beschrieben, wie man herausfindet, in welchen SBSpace die eingefügten SmartLargeObjects gespeichert wurden.

Stellt sich nun die Aufgabe eines oder mehrere dieser Objekte in einen anderen SBSpace zu verschieben, so kann dies mit der Funktion LOCOPY() erfolgen.

Das Ändern der PUT Clause in der Tabellendefinition wirkt nur auf neu eingefügte Datensätze, nicht aber auf bereits bestehende Objekte.

Beispiel:

```
create table if not exists pictures (  
id      serial,  
txt     varchar(255),  
img     blob  
) put img in (sbdbs);
```

Ein Ändern des Speicherortes für das Feld "img" wirkt sich nur auf Objekte aus, die neu eingefügt werden.

```
alter table pictures put img in (sbdbs2);
```

Sollen auch die bestehenden Daten in den neu definierten Bereich verschoben werden, um z.B. den bestehenden SBSpace zu löschen, so ist der Aufruf der Funktion LOCOPY() notwendig:

```
update pictures set img = LOCOPY(img,"pictures","img") where 1=1;
```

Beispiel:

```
insert into pictures values  
  (0,"Reutin",filetoblob("/home/kalu/FOTOS/Reutin1953.jpg","server"));  
insert into pictures values  
  (0,"Reutin",filetoblob("/home/kalu/FOTOS/Reutin1954.jpg","server"));  
insert into pictures values  
  (0,"Reutin",filetoblob("/home/kalu/FOTOS/Reutin1955.jpg","server"));  
insert into pictures values  
  (0,"Reutin",filetoblob("/home/kalu/FOTOS/Reutin1956.jpg","server"));
```

Die Abfrage auf die Location der Objekte ergibt:

id	space_nr	spacename
1	3	sbdbs
2	3	sbdbs
3	3	sbdbs
4	3	sbdbs

Nun ändern wir die PUT Clause der Tabelle pictures:

```
alter table pictures put img in (sbdbs2);
```

... und fügen einen neuen Datensatz ein:

```
insert into pictures values
  (0, "NEU", filetoblob("/home/kalu/FOTOS/Reutin1957.jpg", "server"));
```

Die Abfrage auf die Location der Objekte ergibt:

id	space_nr	spacename
1	3	sdbds
2	3	sdbds
3	3	sdbds
4	3	sdbds
5	6	sdbds2

Um alle Objekte zu verschieben, wenden wir die Funktion LOCOPY() an:

```
update pictures set img = LOCOPY(img, "pictures", "img") where 1=1;
```

Die Abfrage auf die Location der Objekte ergibt:

id	space_nr	spacename
1	6	sdbds2
2	6	sdbds2
3	6	sdbds2
4	6	sdbds2
5	6	sdbds2

TechTipp: ONCONFIG - AUTO_READAHEAD

Der Parameter AUTO_READAHEAD wurde um eine weitere Optionen ergänzt.

Es wurde die Option hinzugefügt einen Threshold anzugeben.

Die Syntax lautet nun:

```
AUTO_READAHEAD      <mode>, <batch_size>, <threshold>
```

Mögliche Parameter für dem Mode sind:

- 0 Der AUTO_READAHEAD ist deaktiviert
- 1 Der AUTO_READAHEAD ist aktiviert und lädt, falls bei einem SQL-Statement ein I/O-Wait auftritt.
- 2 Der AUTO_READAHEAD startet mit dem Start der Abfrage.

Die „batch_size“ wird in Pages angegeben, die in den Cache kopiert werden, in der Hoffnung dass diese benötigt werden. Dieser Parameter kann die Wert 4 bis 4096 annehmen (der Default ist 128).

Der Threshold bestimmt, wann weitere Pages nachgelesen werden, und wird in Prozent angegeben. Ein Threshold von 20 (%) bedeutet, dass ein Nachlesen startet, wenn weniger als 20% der bisher gelesenen Pages noch nicht verarbeitet wurden.

Der Defaultwert für diesen Parameter ist 50.

Eine Anpassung ist dynamisch mittels „onmode -wf“ möglich.

TechTipp: ONCONFIG - TLS_VERSION

Für die sichere Verbindung zwischen dem Datenbankserver und den Clients (bzw. anderen Instanzen) wird oft eine verschlüsselte Verbindung mittels SSL (Secure Socket Layer) verwendet.

Der Default für die TLS_VERSION ist 1.2.

Optional kann die TLS_VERSION 1.3 gewählt werden, das jedoch bei Java Clients mindestens Java11 voraussetzt.

TLS_VERSION 1.0 und 1.1 werden mit der Version 14.10.xC8 nicht mehr unterstützt.

TechTipp: CDR Replikation nach UTF-8

Die Enterprise Replikation (CDC) ist bereits seit Version 7.23 Bestandteil des Datenbankservers. Eine der Eigenschaften der CDR ist, dass eine Replikation in andere Versionen von INFORMIX und auf andere Betriebssysteme möglich ist. So wurden viele Migrationen z.B. von HP-UX auf Linux mittels Enterprise Replikation durchgeführt, ohne dass die Datenbank mit langer Downtime exportiert und importiert werden musste.

Weniger bekannt ist, dass seit Version 12.10 die CDR Replikation bereits im Default bei der Übertragung mit UTF-8 arbeitet.

Beim „cdr create replicate ...“ kann über die Option „--UTF8=y“ angegeben werden, ob die Daten beim Kopieren in den Übertragungsbereich bereits in UTF-8 konvertiert werden. Wird der Parameter nicht angegeben, so erfolgt laut Dokumentation die Konvertierung (Default=y).

Soll dies nicht geschehen, z.B. um die Performance zu optimieren, so kann diese Umwandlung explizit mit „--UTF8=n“ unterbunden werden.

Am Zielsystem wird versucht, die Daten in den entsprechenden Codeset der Zieldatenbank umzuwandeln. Ist das Ziel eine Datenbank mit Codeset utf-8, so ist keine Anpassung mehr notwendig.

Dieser Umstand kann dazu genutzt werden, um eine bestehende Datenbank mittels Replikation in eine Datenbank mit Codeset UTF8 zu konvertieren.

Hinweis: Eine Umwandlung erfolgt (laut Dokumentation) nicht bei TimeSeries, UserDefineDatatypes, CLOBs, BLOBs, sowie den Datentypen TEXT und BYTE.

TechTipp: Auslieferung der Komponente JRE endet

Ab der kommenden Version wird die Komponente JRE, die bisher im \$INFORMIXDIR/extend/krakatoa/jre zu finden war, nicht mehr mitgeliefert.

Damit wird für die Installation eine bereits installierte JavaRuntimeEnvironment als Voraussetzung erwartet, damit die Informix Produkte installiert werden können.

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

INFORMIX Advanced Support
HCL Technologies

Martin Fuerderer

INFORMIX Development
HCL Technologies

Gerd Kaluzinski

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:

Blick B12 Weissensberg in Richtung Bodensee - Mai 2022

(Gerd Kaluzinski)