

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Rolling Upgrade.....	2
TechTipp: SEC2ER - Von der HDR/RSS zur Enterprise Replication.....	2
TechTipp: Flexible Grid – Replikation der nächsten Generation (Teil 1).....	6
TechTipp: Connection Manager (Gastbeitrag Bereos).....	10
Termine: INFORMIX 11.70 – New Features (IX3020DE).....	18
WebTipp: Artikel auf Developerworks zum Thema Grid Replication.....	18
WebTipp: INFORMIX integration (by Lester Knutsen).....	18
WebTipp: INFORMIX Redbook zur Grid Replikation.....	18
Versionsinfo: 11.70.xC3 ist verfügbar.....	18
Anmeldung / Abmeldung / Anmerkung.....	19
Die Autoren dieser Ausgabe.....	19

Aktuelles

Liebe Leserinnen und Leser,

der Sommer sorgt für heiße Tage, und auch INFORMIX bietet mit den Themen „Warehouse Accelerator“, „TimeSeries“ und „Flexible Grid“ brennende Themen. Das große Interesse am Markt zeigt, dass INFORMIX damit die Zeichen der Zeit erkannt und umgesetzt hat, was besonders durch viele neue Kunden bestätigt wird. Die Sommerferien stehen an bzw. haben teilweise bereits begonnen. Die Newsletter Redaktion macht jedoch angesichts der vielen Themen und neuen Features keine Pause, so dass sie auch im Juli und August die TechTipp im gewohnten Umfang genießen können. Wir gönnen uns nur hin und wieder ein Eis zur Abkühlung.



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

TechTipp: Rolling Upgrade

Eine geplante Downtime für eine Migration der Datenbank ist nicht immer einfach zu bekommen. Immer mehr Systeme müssen 24x7 verfügbar sein.

Mit dem Rolling Upgrade ist es möglich, die Downtime für die Migration auf einen Schwenk der Applikationen zu einem zweiten Server zu begrenzen.

Der Grundgedanke dabei ist, dass alle Applikationen für die Zeit des Upgrade zu einer zweiten Instanz verbunden werden, deren Daten mittels „Enterprise Replication“ identisch zu denen der ersten Instanz gehalten werden.

Sind alle Applikationen auf den zweiten Server umgeschaltet und die Sendqueue des Primary Servers ist leer, wird die Replikation der Daten auf den primären Server ausgesetzt.

Nun kann der primäre Server auf eine höhere Version migriert werden. Ist dies erfolgreich abgeschlossen, wird die Replikation wieder aufgenommen (RESUME). Werden nun die Applikationen zurück auf die primäre Seite geschwenkt, ist die Migration beendet.

Anschließend kann der zweite Datenbankserver wieder entfernt werden. Meist wird dieser jedoch auch noch im selben Verfahren migriert, so dass auch eine ungeplante Downtime (Serverausfall) in Zukunft mit nur einem Schwenk der Applikationen behoben werden kann.

Um eine Enterprise Replikation für alle Tabellen aller Datenbanken aufzusetzen, war bisher ein erheblicher Administrationsaufwand notwendig: CDR-Server definieren, alle Tabellen mit Primary Keys versorgen (falls noch nicht vorhanden), Replicates je Tabelle definieren.

Mit Version 11.70 wurde eine neue Funktionalität eingeführt, die eine bestehende HDR oder RSS Replikation in eine Enterprise Replikation umwandeln kann. Hierbei handelt es sich um die Funktion **SEC2ER**, die im nächsten Beitrag ausführlich beschrieben wird.

TechTipp: SEC2ER - Von der HDR/RSS zur Enterprise Replication

Eine bestehende HDR oder RSS Replikation kann mittels der neuen Funktion `sec2er` in eine Enterprise Replikation umgewandelt werden. Das Aufsetzen einer HDR mittels `ontape` nach `STDIO` wurde im *INFORMIX Newsletter 10/2006* beschrieben. In der Version 11.70 gibt es zudem die Möglichkeit die HDR oder RSS mittels `ifxclone` aufzubauen.

In der Datei `$INFORMIXDIR/etc/sqlhosts` müssen die beteiligten Instanzen, sowie Gruppen für die Replikation auf allen Rechnern eingetragen sein. Die gegenseitige Erreichbarkeit ohne Eingabe eines Passworts für den Benutzer „informix“ ist notwendig. Dies kann mittels „dbaccess“ über „connection“ getestet werden.

Beispiel:

<code>ifx1_rep</code>	<code>group</code>	<code>-</code>	<code>-</code>		<code>i=1</code>
<code>ifx1</code>	<code>onsoctcp</code>	<code>172.16.41.229</code>		<code>9081</code>	<code>g=ifx1_rep</code>
<code>ifx2_rep</code>	<code>group</code>	<code>-</code>	<code>-</code>		<code>i=2</code>
<code>ifx2</code>	<code>onsoctcp</code>	<code>172.16.41.229</code>		<code>9082</code>	<code>g=ifx2_rep</code>

Alternative Verbindungen zu einer Instanz (DBSERVERALIAS) können mit in die selbe Gruppe aufgenommen werden, indem der Gruppenname als Option angefügt wird.

Bei der Umwandlung der HDR/RSS in die Enterprise Replikation wird jeder Tabelle, die keinen Primary Key besitzt, mittels „alter table ... add ERKEY“ ein interner Schlüssel für die Replikation zugewiesen.

Der Inhalt des ERKEY kann mittels SQL abgefragt werden:

```
select id,ifx_erkey_1,ifx_erkey_2,ifx_erkey_3 from mx42;
```

Ergebnis:

id	ifx_erkey_1	ifx_erkey_2	ifx_erkey_3
42	5	2	1
23	5	3	1
13	5	4	1

Es wird auf beiden Datenbankservern die Enterprise Replikation eingerichtet, wodurch die Datenbank syscdr implizit aufgebaut wird.

Falls es sich bisher um eine HDR Replikation handelte, so wird diese zuerst in eine RSS Replikation umgewandelt, wobei der Secondary als „Updatable“ gekennzeichnet wird.

Danach werden je Tabelle (für alle Tabellen außer Systemtabellen) Replicates eingerichtet. Nach erfolgreicher Erstellung der Replicates je Tabelle wird die RSS Replikation deaktiviert und die Enterprise Replikation gestartet.

Die Replicates sind als „update anywhere“ erstellt und replizieren daher in beiden (allen) Richtungen.

Bevor die Umwandlung einer HDR/RSS Replikation in eine CDR Replikation gestartet wird sollte geprüft werden, ob dies problemlos möglich ist. Werden Spalten vom Typ Serial gefunden und ist in der ONCONFIG der Parameter CDR_SERIAL nicht gesetzt, so wird eine Warnung ausgegeben.

Der Aufruf:

```
cdr check sec2er ifx2
```

The version of the server you are using has full ERKEY support.

WARNING: CDR_SERIAL value on ifx1 can cause collisions.

WARNING: Database space logdbs is becoming full.

WARNING: Using the same values for CDR_SERIAL can cause collisions.

Secondary conversion to ER is possible.

Sollten Fehler „ERROR“ gemeldet werden, so können diese behoben werden um die Umwandlung erneut zu versuchen.

Gibt man bei der Prüfung der Umwandlung zusätzlich den Schalter „-p“ an, so werden die Befehle ausgegeben, die zur Erstellung der CDR Server und Replicates ausgeführt werden müssten:

```

cdr define serv -c server1_rep -I ifx1_rep
dbaccess - - <<EOF

database mx42;
alter table 'informix'.warehouses add ERKEY;
EOF

cdr define repl --connect=ifx1_rep sec2er_1_1302602697_call_type      \
--master=ifx1_rep --conflict=always --scope=row                    \
      \ "mx42@ifx1_rep:'informix'.call_type"                       \
      \ "select * from 'informix'.call_type"cdr start repl
--connect=ifx1_rep sec2er_1_1302602697_call_type

```

Diese Befehle könnten genutzt werden, um die Replikation manuell Schritt für Schritt einzurichten.

Der Start der automatischen Umwandlung (nach erfolgreichem Check) erfolgt mittels:

```
cdr start sec2er ifx2
```

Dabei wird in der ONCONFIG der betroffenen Instanzen der Parameter LOG_INDEX_BUILDS auf 1 geändert, wenn dies bisher nicht der Fall war.

Im online.log des Primary Servers ist die Umwandlung zu sehen als:

```

'syscdr' database built successfully.
'sysha' database built successfully.
RSS ifx2 added RSS Server ifx2 - state is now connected
CDR shutdown complete
RSS Server ifx2 - state is now disconnected
Internal stopping and restarting of ER was successful
ER clone: Successfully created new ER node with id 2
      ('server2_rep')
CDR: Re-connected to server, id 2, name <ifx2_rep>
...

```

Am ehemaligen Secondary Server sieht dies im online.log so aus:

```

DR: new type = RSS
DR: RSS secondary server operational
ER clone: Starting split from primary server 'ifx1'.
ER clone: Joining this server into the ER domain.
ER clone: Determining Grid participation.
ER clone: Adding this server to ER replicates.
ER clone: Waiting for ER control queues to drain
      at the progenitor server.
ER clone: Restarting ER queues at progenitor server and peers.
ER clone: Waiting for open transactions to complete
      at the progenitor server.
ER clone: Waiting for LSN to advance to 63:0x0x3f90dc
ER clone: LSN successfully advanced to 63:0x0x3f90dc
ER clone: Converting this server to standard mode.
...

```

```
Shutting down SDS/RSS threadsSMX thread is exiting
DR: Turned off on secondary server
DR: new type = standard
ER clone: Adjusting Enterprise Replication parameters.
On-Line Mode
ER clone: Starting Enterprise Replication.
ER checkpoint started
ER clone: Successfully completed split
         from primary server 'ifx1'
```

Danach sind die Server in der Enterprise Replikation aktiv.

cdr list server

SERVER	ID	STATE	STATUS	QUEUE	CONNECTION	CHANGED
ifx1_rep	1	Active	Local	0		
ifx2_rep	2	Active	Connected	0	May 31 12:36:05	

Die Liste der Replicates kann ebenfalls angezeigt werden:

cdr list repl brief

REPLICATE	TABLE	SELECT
sec2er_1_1309159217_call_type	stores@ifx1_rep:informix.call_type	select call_code, code_descr from 'informix'.call_type
sec2er_1_1309159217_call_type	stores@ifx2_rep:informix.call_type	select call_code, code_descr from 'informix'.call_type
sec2er_2_1309159217_catalog	stores@ifx1_rep:informix.catalog	select catalog_num, stock_num, manu_code, cat_descr, cat_picture, cat_advert from 'informix'.catalog
sec2er_2_1309159217_catalog	stores@ifx2_rep:informix.catalog	select catalog_num, stock_num, manu_code, cat_descr, cat_picture, cat_advert from 'informix'.catalog
sec2er_3_1309159217_classes	stores@ifx1_rep:informix.classes	select classid, class, subject, ifx_erkey_1, ifx_erkey_2, ifx_erkey_3 from 'informix'.classes
...		

Die Namen der Replicates werden automatisch vergeben. Diese beginnen mit „sec2er“, also dem Befehl der die Erstellung veranlasst hat. Danach kommt die SERVERNUM der Servers, auf dem der Befehl aufgerufen wurde. Nach eine generierten ID folgt der Tabellenname, wie am Beispiel „sec2er_1_1309159217_call_type“ zu sehen ist.

TechTipp: Flexible Grid – Replikation der nächsten Generation (Teil 1)

Die Enterprise Replikation bot bisher bereits eine Vielzahl an Varianten, um Daten zu replizieren. Alle Änderungen an Tabellenstrukturen (DDL) waren jedoch mit teilweise erheblichem administrativem Aufwand verbunden.

Viele Kunden nahmen für einen „alter table“ die betroffene Tabelle aus der Replikation und erstellten anschließend nach der Änderung auf allen Servern das Replikat für die Tabelle neu. Alternativ dazu konnte mit Master Replicates gearbeitet werden, wobei nach dem „alter“ der Tabelle ein Remaster auf allen Servern notwendig war.

Für neu angelegte Tabellen mussten explizit Replicates erstellt werden.

Das „Flexible Grid“ ist eine Erweiterung der Enterprise Replikation, die es ermöglicht Änderungen an Tabellen ohne zusätzlichen administrativen Aufwand auf allen beteiligten Server durchzuführen. Neue Tabellen werden sofort in die Replikation mit eingebunden. DDL und DML Befehle können auf allen Servern des Grid gleichzeitig ausgeführt werden.

Voraussetzung für das Erstellen eines Grid ist eine aktive Enterprise Replikation, bei der zumindest alle Instanzen als CDR-Server definiert sind.

Zur Verwaltung des Grid stehen folgende Befehle zur Verfügung:

- **cdr define grid** # Erstellen eines Grid
- **cdr delete grid** # Löschen eines Grid
- **cdr change grid** # Ändern der Struktur eines Grid
(Hinzufügen oder entfernen von Instanzen und Berechtigungen)
- **cdr enable grid** # Aktivieren eines Grid
- **cdr disable grid** # Deaktivieren eines Grid
- **cdr list grid** # Anzeige eines Grid

Ist das Grid aufgebaut und aktiv, dann können die User, die für das Grid berechtigt wurden folgende Funktionen ausführen:

- **ifx_grid_connect()** # Zum Grid verbinden
- **ifx_grid_disconnect()** # Verbindung zum Grid trennen
- **ifx_grid_execute()** # Ausführen von Befehlen im Grid
- **ifx_grid_procedure()** # Ausführen einer Procedure im Grid
- **ifx_grid_function()** # Ausführen einer Funktion im Grid
- **ifx_grid_redo()** # Erneuter Aufruf eines Gridbefehls
- **ifx_grid_purge()** # Löschen bisheriger Grid Aufrufe
- **ifx_set_erstate()** # Aktivieren/Deaktivieren der Replikation
- **ifx_get_erstate()** # Abfragen des Replikations Status

Beispiel:

Erstellen eines Grid auf bestehenden CDR-Servern:

```
cdr define grid test_grid test1_rep test2_rep test3_rep
```

oder (wenn alle CDR-Server ins Grid sollen):

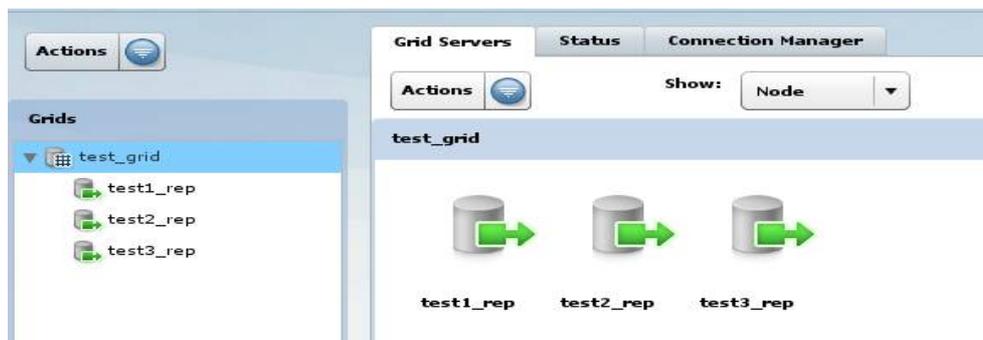
```
cdr define grid test_grid --all
```

Anzeige des Grid:

```
cdr list grid -v
```

Grid	Node	User
test_grid	test1_rep	
	test2_rep	
	test3_rep	

Das Grid ist auch im OAT sichtbar und kann dort auch erstellt und verwaltet werden:



Beim Aktivieren eines Grid können auch gleich die User angegeben werden, die die Grid Funktionalitäten nutzen dürfen:

```
cdr enable grid -g test_grid -n test1_rep -n test2_rep -n test3_rep -u informix -u kalu
```

Anzeige:

```
cdr list grid -v
```

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

DDL-Aufrufe ohne Verbindung zum Grid werden weiterhin lokal ausgeführt und es werden keine Replikate für die neuen Tabellen erstellt.

Verbindet man sich zum Grid mittels der Procedure „ifx_grid_connect()“, so werden die DDL-Befehle im Grid ausgeführt und Replikate erstellt.

```
execute procedure ifx_grid_connect('test_grid', 1);
create table kalu (f1 char(42));
execute procedure ifx_grid_disconnect();
```

ACHTUNG:

Sie sollten sich nach einer Grid Aktion immer vom Grid trennen (ifx_grid_disconnect) um nicht unbewusste Aktionen auf allen Grid Servern auszuführen.

Was beim Grid im Hintergrund geschieht, sieht man wieder im cdr list grid:

```
cdr list grid -v
```

```
Node:test1_rep Stmtid:41 User:informix Database:stores 2011-06-01 09:54:15
create table kalu (f1 char(42))
ACK test1_rep 2011-06-01 09:54:15
ACK test2_rep 2011-06-01 09:54:16
ACK test3_rep 2011-06-01 09:54:16
```

```
Node:test1_rep Stmtid:42 User:informix Database:syscdr 2011-06-01 09:54:15
Define Repl G65579_1_42_kalu for stores:informix.kalu
ACK test1_rep 2011-06-01 09:54:15
ACK test2_rep 2011-06-01 09:54:16
ACK test3_rep 2011-06-01 09:54:16
```

Ebenso beim „alter table“:

```
execute procedure ifx_grid_connect('test_grid', 1);
alter table kalu add (f2 char(23));
execute procedure ifx_grid_disconnect();
```

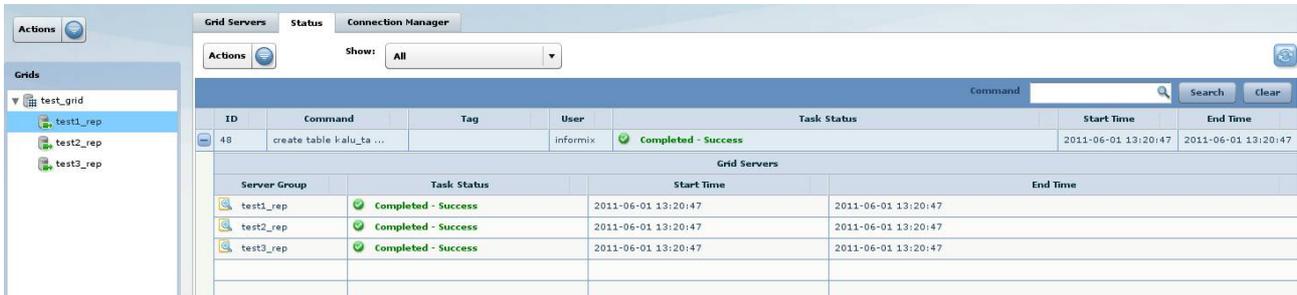
```
cdr list grid -v
```

```
Node:test1_rep Stmtid:43 User:informix Database:stores 2011-06-01 09:55:40
alter table kalu add (f2 char(23))
ACK test1_rep 2011-06-01 09:55:40
ACK test2_rep 2011-06-01 09:55:42
ACK test3_rep 2011-06-01 09:55:41
```

```
Node:test1_rep Stmtid:44 User:informix Database:syscdr 2011-06-01 09:55:40
Remaster Repl : Primary Repl=G65579_1_42_kalu, Shadow
Repl=Shadow_4_G65579_1_42_kalu_1306914940_1_44 for stores:informix.kalu
ACK test1_rep 2011-06-01 09:55:40
ACK test2_rep 2011-06-01 09:55:42
ACK test3_rep 2011-06-01 09:55:41
```

```
Node:test1_rep Stmtid:1 User:informix Database:ibm 2011-05-23 15:27:54
create table kalu_tab (f1 int, f2 char(42))
ACK test1_rep 2011-05-23 15:27:54
ACK test2_rep 2011-05-23 15:27:54
ACK test3_rep 2011-05-23 15:27:54
```

Im OAT ist je Befehl zu sehen, ob dieser auf den im Grid enthaltenen Instanzen erfolgreich oder fehlerhaft ausgeführt werden konnte:



ID	Command	Tag	User	Task Status	Start Time	End Time
48	create table kalu_ta ...		informix	Completed - Success	2011-06-01 13:20:47	2011-06-01 13:20:47

Server Group	Task Status	Start Time	End Time
test1_rep	Completed - Success	2011-06-01 13:20:47	2011-06-01 13:20:47
test2_rep	Completed - Success	2011-06-01 13:20:47	2011-06-01 13:20:47
test3_rep	Completed - Success	2011-06-01 13:20:47	2011-06-01 13:20:47

Auch die SQL Syntax der Befehle ist im OAT im Klartext zu sehen:



Im 2. Teil zum Flexible Grid in den nächsten Ausgabe des INFORMIX Newsletters erfahren Sie, wie man SQL-Statements auf allen am Grid beteiligten Servern ausführen kann.

TechTipp: Connection Manager**(Gastbeitrag Bereos)**

Um einen schnellen und zuverlässigen Zugriff auf die benötigten Daten gewährleisten zu können, erfreuen sich Hochverfügbarkeitscluster (auch MACH11-Cluster genannt) immer größerer Beliebtheit. Häufig ist es jedoch schwierig festzustellen, welcher Server über ausreichende Ressourcen verfügt oder gerade auf ein Problem stößt. Abhilfe schafft hierbei der Connection Manager, welcher als Bestandteil des Client SDK in den Versionen 3.50 oder höher mitgeliefert wird.

Folgendes Szenario ist gegeben:

- node1 (primary) → Hauptserver
- node2 (HDR Secondary) → Read Only
- node3 (RS Secondary) → Updateable Secondary

Aufsetzen der HDR:

node1	node2
ontape -s -l 0 -t stdio rsh node2 „. /opt/ifxini; ontape -p -t stdio“	
onmode -d primary n2_tcp	
	onmode -d secondary n1_tcp

Aufsetzen des RSS:

node1	node3
ontape -s -l 0 -t stdio rsh node3 „. /opt/ifxini; ontape -p -t stdio“	
onmode -d add RSS n3_tcp	
	onmode -d RSS n1_tcp

Zustände der einzelnen Server:

```
informix@node1:~$ onstat -
```

```
IBM Informix Dynamic Server Version 11.70.UC2GE -- On-Line (Prim) -- Up 00:32:10 -- 160540 Kbytes
```

```
informix@node2:~$ onstat -
```

```
IBM Informix Dynamic Server Version 11.70.UC2GE -- Read-Only (Sec) -- Up 00:29:07 -- 160540 Kbytes
```

```
informix@node3:~$ onstat -
```

```
IBM Informix Dynamic Server Version 11.70.UC2GE -- Updatable (RSS) -- Up 00:19:43 -- 168732 Kbytes
```

Einträge in der sqlhosts auf allen Nodes:

```
cluster      group      -      -      i=30
n1_tcp      onsoctcp  node1   1521   g=cluster
n2_tcp      onsoctcp  node2   1521   g=cluster
n3_tcp      onsoctcp  node3   1521   g=cluster
```

Installation des Connection Manager:

Den Connection Manager installieren wir mittels Client SDK in ein separates Verzeichnis. Um auch hier Redundanz zu erhalten, packen wir diesen auf alle 3 Nodes. Dies könnten natürlich auch komplett separate Server sein.

Konfiguration des Connection Manager:**node1:**

```
informix@node1:~$ cat /opt/csdk/etc/sqlhosts
```

```
cm1_read      onsoctcp      node1  1531
cm1_write     onsoctcp      node1  1532
```

```
n1_tcp        onsoctcp      node1  1521
n2_tcp        onsoctcp      node2  1521
n3_tcp        onsoctcp      node3  1521
```

```
informix@node1:~$ cat /opt/csdk/etc/cmsm.cfg
```

```
NAME      cm1
SLA       cm1_read=(HDR+RSS)
SLA       cm1_write=(primary+RSS)
FOC       HDR,10
LOG       1
LOGFILE   /opt/csdk/cm1.log
```

node2:

```
informix@node2:~$ cat /opt/csdk/etc/sqlhosts
```

```
cm2_read      onsoctcp      node2  1531
cm2_write     onsoctcp      node2  1532
```

```
n1_tcp        onsoctcp      node1  1521
n2_tcp        onsoctcp      node2  1521
n3_tcp        onsoctcp      node3  1521
```

```
informix@node2:~$ cat /opt/csdk/etc/cmsm.cfg
```

```
NAME      cm2
SLA       cm2_read=(HDR+RSS)
SLA       cm2_write=(primary+RSS)
FOC       HDR,10
LOG       1
LOGFILE   /opt/csdk/cm2.log
```

node3:

```
informix@node3:~$ cat /opt/csdk/etc/sqlhosts
```

```
cm3_read      onsoctcp      node3  1531
cm3_write     onsoctcp      node3  1532
```

```
n1_tcp        onsoctcp      node1  1521
n2_tcp        onsoctcp      node2  1521
n3_tcp        onsoctcp      node3  1521
```

```
informix@node3:~$ cat /opt/csdk/etc/cmsm.cfg
```

```
NAME      cm3
SLA       cm3_read=(HDR+RSS)
SLA       cm3_write=(primary+RSS)
FOC       HDR,10
LOG       1
LOGFILE   /opt/csdk/cm3.log
```

Starten des Connection Manager:

Den Connection Manger starten wir auf allen 3 Nodes mit folgendem Befehl.
INFORMIXDIR sollte hierbei auf das Installationsverzeichnis des Client SDK zeigen.

```
oncmsm -c /opt/ids/etc/cmsm.cfg
```

Überwachung:

```
informix@node1:~$ onstat -g cmsm
```

```
IBM Informix Dynamic Server Version 11.70.UC2GE -- On-Line (Prim) -- ...
```

```
Connection Manager Name: cm1
```

```
  Hostname: node1
```

SLA	Connections	Service/Protocol	Rule
cm1_read	0	1531/onsoctcp	(HDR+RSS)
cm1_write	1	1532/onsoctcp	(primary+RSS)

```
Connection Manager Name: cm2
```

```
  Hostname: node2
```

SLA	Connections	Service/Protocol	Rule
cm2_read	0	1531/onsoctcp	(HDR+RSS)
cm2_write	0	1532/onsoctcp	(primary+RSS)

```
Connection Manager Name: cm3
```

```
  Hostname: node3
```

SLA	Connections	Service/Protocol	Rule
cm3_read	1	1531/onsoctcp	(HDR+RSS)
cm3_write	1	1532/onsoctcp	(primary+RSS)

```
Failover Configuration:
```

Connection Manager name	Rule	Timeout	State
cm1	HDR	10	Active Arbitrator, Primary is up
cm2	HDR	10	Primary is up
cm ³	HDR	10	Primary is up

```
informix@node1:~$ onstat -g rss
```

```
IBM Informix Dynamic Server Version 11.70.UC2GE -- On-Line (Prim) -- ...
```

```
Local server type: Primary
```

```
Index page logging status: Enabled
```

```
Index page logging was enabled at: 2011/06/29 05:51:42
```

```
Number of RSS servers: 1
```

```
RSS Server information:
```

RSS Srv name	RSS Srv status	Connection status	Next LPG to send (log id,page)	Supports Proxy Writes
n3_tcp	Active	Connected	10,1494	Y

```
informix@node1:~$ netstat -tln
```

```
...
tcp        0      0 172.16.60.1:1531 0.0.0.0:*      LISTEN
tcp        0      0 172.16.60.1:1532 0.0.0.0:*      LISTEN
...
```

Sollte der Connection Manger nicht ordentlich starten, wird dieser durch den Parameter DEBUG <num> in der cmsm.cfg etwas gesprächiger.

Client Konfiguration:

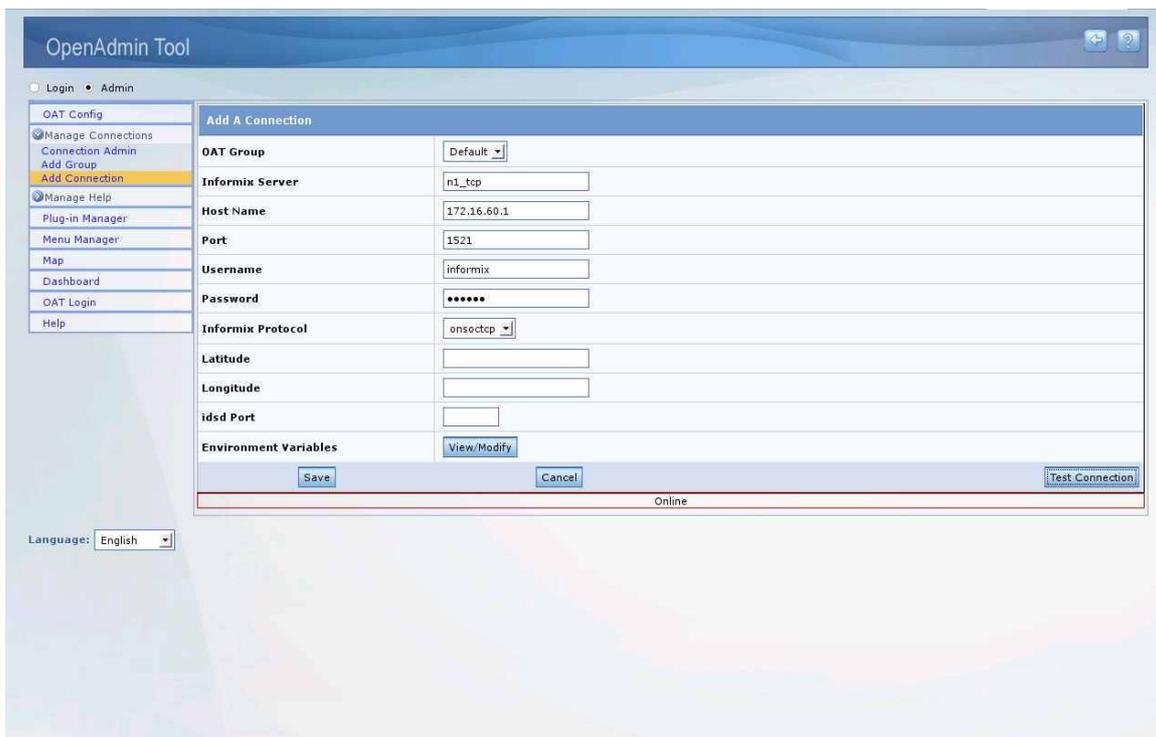
Die Konfiguration lässt sich am einfachsten mit einem weiteren Informix Client und dbaccess testen. Hierzu tragen wir Folgendes in die sqlhosts ein:

```
informix@client1:~$ cat /opt/ids/etc/sqlhosts
```

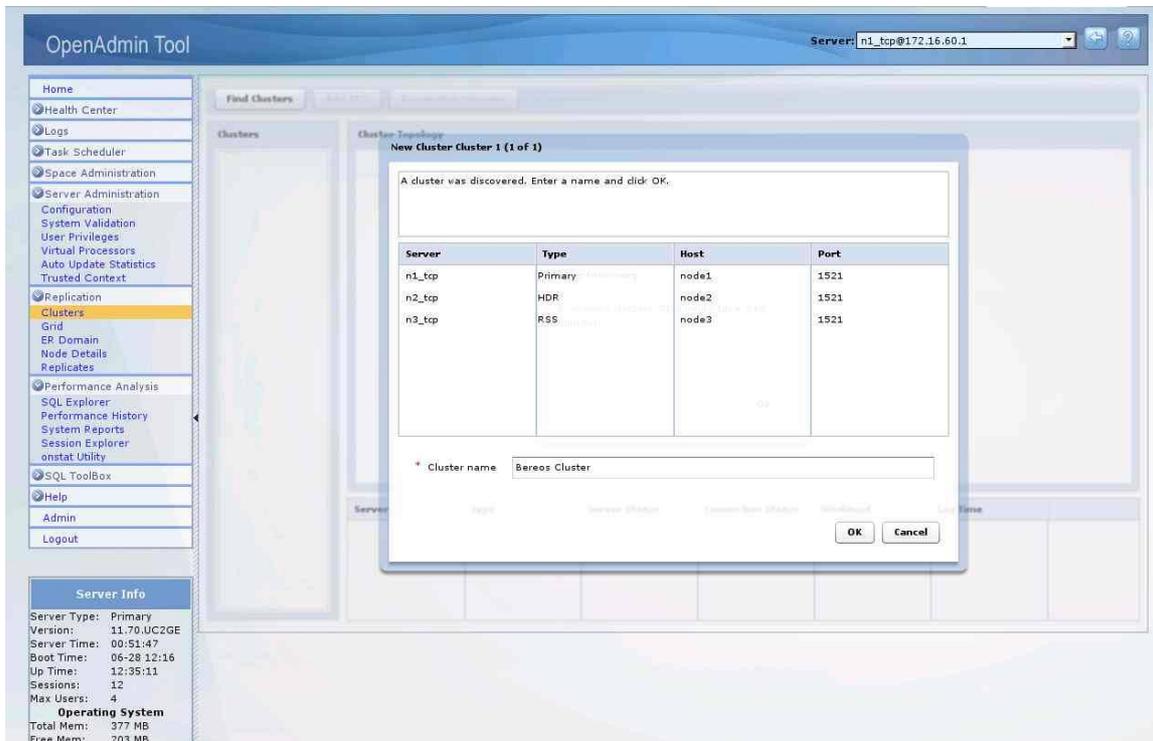
```
cm_read      group      -      -      i=10,c=1
cm1_read     onsoctcp   node1   1531   g=cm_read
cm2_read     onsoctcp   node2   1531   g=cm_read
cm3_read     onsoctcp   node3   1531   g=cm_read
cm_write     group      -      -      i=10,c=1
cm1_write    onsoctcp   node1   1532   g=cm_write
cm2_write    onsoctcp   node2   1532   g=cm_write
cm3_write    onsoctcp   node3   1532   g=cm_write
```

Integration in OAT:

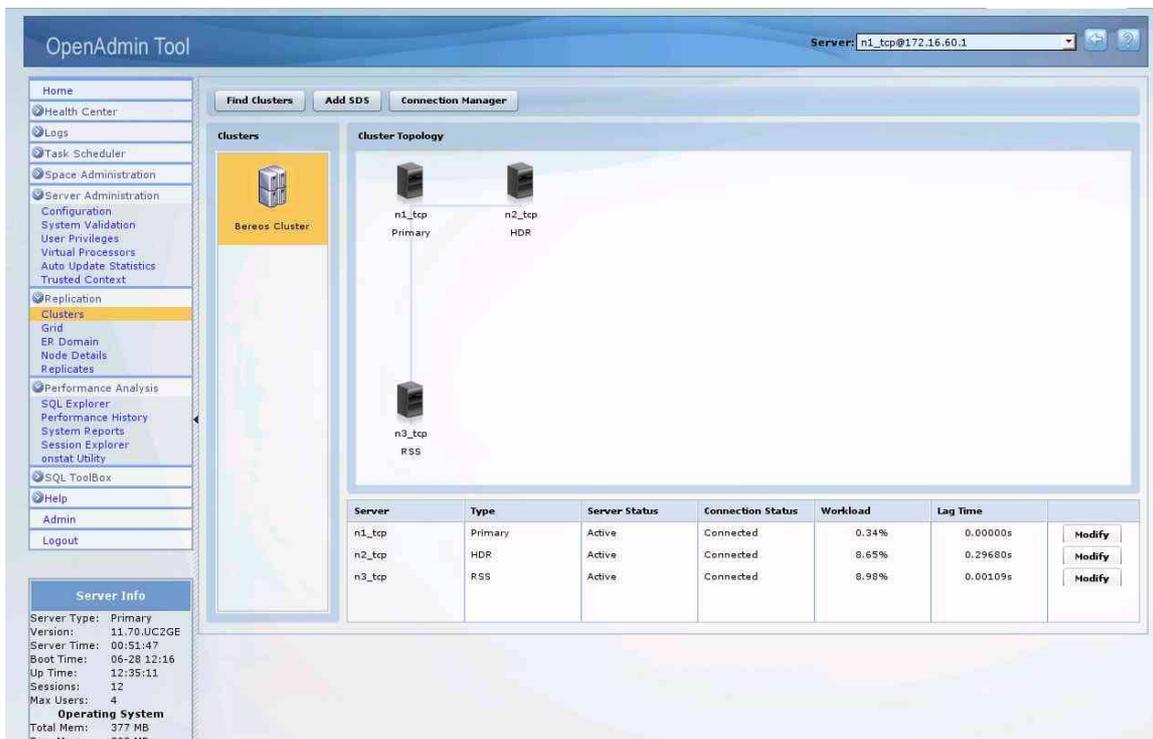
Zunächst richten wir einen unserer Server im OAT ein. In unserem Beispiel den node1.



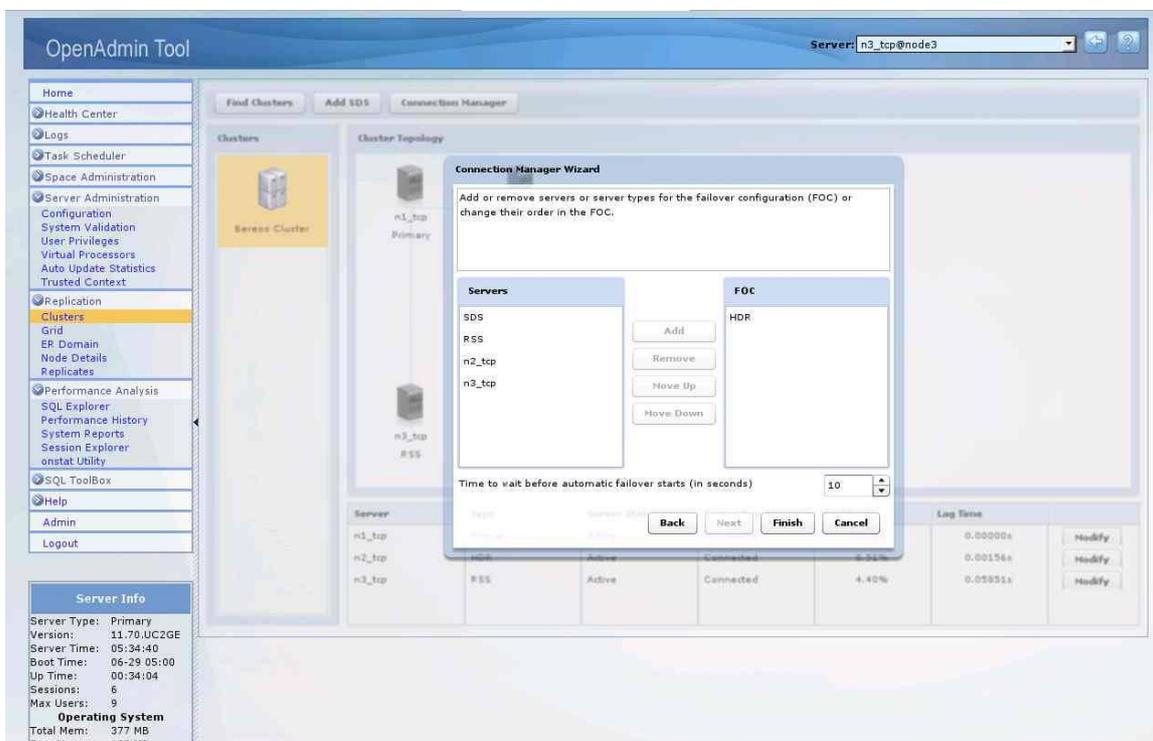
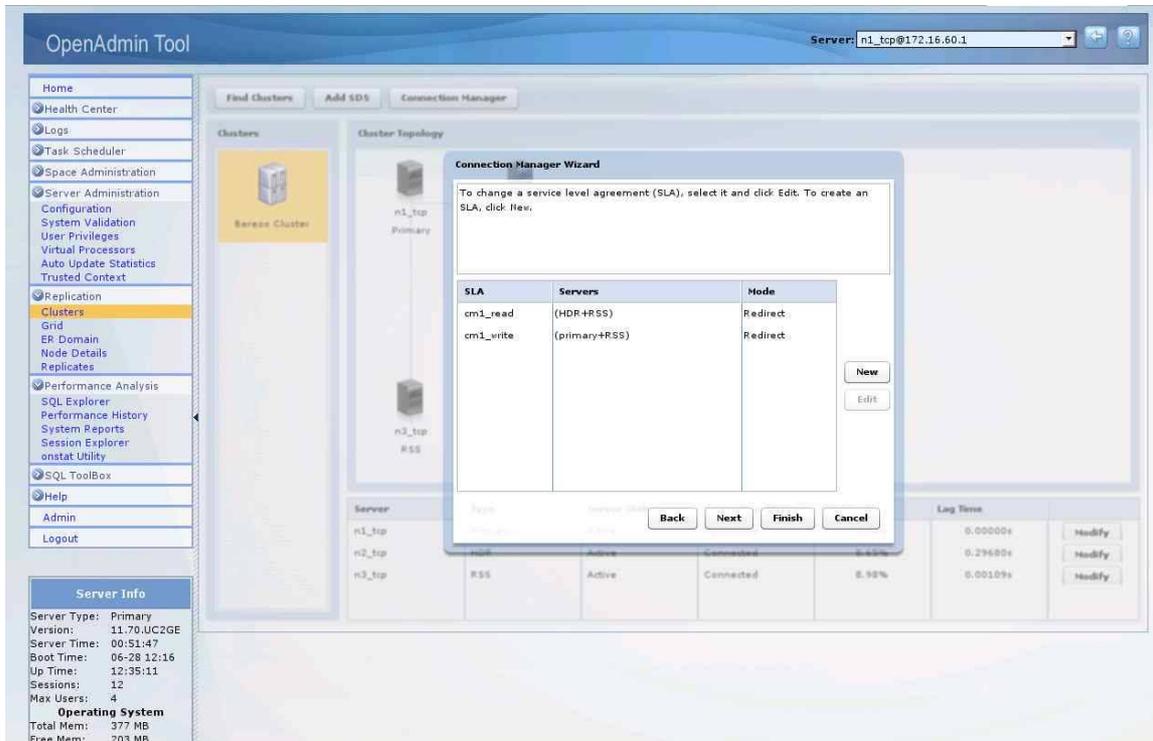
Nachdem wir uns auf diesen Server verbunden haben, können wir über Replication / Cluster nach unserem Cluster suchen (Find Clusters) und diesem einen Namen geben.



Alle weiteren Nodes werden automatisch gefunden und als Verbindungen mit eingetragen. Hier muss ggf. nochmals das Passwort mitgegeben werden.



Über den Reiter Connection Manager finden wir alle Connection Manger Instanzen, die mit diesem Cluster verbunden sind. Dort lassen sich die Service Level Agreements (SLA) oder Failover Configuration (FOC) bearbeiten, hinzufügen oder löschen.



Konfigurationsparameter: DRAUTO

Ist DRAUTO auf 3 gesetzt, übernimmt der Connection Manager die Aufgabe des automatischen Umschaltens. Es wird so verhindert, dass mehrere Primary's in einem Hochverfügbarkeitscluster existieren können. Über die FOC Einstellungen wird festgelegt, wer beim Ausfall des Primary zum neuen Primary gemacht wird. In unserem Falle übernimmt also node2, unser HDR Secondary, die Aufgabe des Primary. Die Übernahme ist auch im online.log schön zu sehen.

```
informix@node2:~$ onstat -m
07:07:53 DR: Receive error
07:07:53 ASF Echo-Thread Server: asfcode = -25582: oserr = 4: errstr = :
Network connection is broken.
System error = 4.
07:07:53 DR_ERR set to -1
07:07:53 SMX thread is exiting
07:07:54 DR: Turned off on secondary server
07:08:09 Skipping failover callback.
07:08:10 Logical Recovery has reached the transaction cleanup phase.
07:08:10 Checkpoint Completed: duration was 0 seconds.
07:08:10 Wed Jun 29 - logunig 10, logpos 0x60d018, timestamp: 0x3cf5c Interval:
54
07:08:10 Maximum server connections 4
07:08:10 Checkpoint Statistics - Avg.Txn Block Time 0.000, #Txns blocked 0,Plog
used 0, Llog used 1
07:08:10 Logical Recovery Complete.
428 Committed, 0 Rolled Back, 0 Open, 0 Bad Locks

07:08:10 Logical Recovery Complete.
07:08:11 Quiescent Mode
07:08:11 Checkpoint Completed: duration was 0 seconds.
07:08:11 Wed Jun 29 - logunig 10, logpos 0x60f018, timestamp: 0x3cf8f Interval:
55
07:08:11 Maximum server connections 4
07:08:11 Checkpoint Statistics - Avg.Txn Block Time 0.000, #Txns blocked 0,Plog
used 5, Llog used 2
07:08:11 B-tree scanners enabled.
07:08:11 DR: Reservation of the last logical log for log backup turned on
07:08:11 DR: new type = primary, secondary server name = n1_tcp
07:08:11 Starting BldNotification
07:08:11 DR: Trying to connect to secondary server = n1_tcp
07:08:11 DR: Cannot connect to secondary server
07:08:11 DR: Turned off on primary server
07:08:11 RSS Server n3_tcp - state is now connected
07:08:12 On-Line Mode

informix@node3:~$ onstat -m
07:07:52 SMX thread is exiting
07:07:52 SMX thread is exiting
07:07:52 RSS: Lost connection to n1_tcp
07:07:52 Updates from secondary currently not allowed
07:07:52 Updates from secondary currently not allowed
07:08:10 RSS Source Node Alias Changed from (n1_tcp) to (n2_tcp)
07:08:10 RSS: Reconnected to n2_tcp
```

Im OpenAdminTool ist der Ausfall des Primary ebenfalls sichtbar:

The screenshot shows the OpenAdmin Tool interface for a cluster. The left sidebar contains navigation options like Home, Health Center, Logs, Task Scheduler, Space Administration, Server Administration, Configuration, Replication, Clusters, Performance Analysis, SQL Toolbox, and Help. The main area displays the 'Cluster Topology' with three nodes: n2_tcp (Primary), n1_tcp (HDR), and n3_tcp (RSS). Below the topology is a table with the following data:

Server	Type	Server Status	Connection Status	Workload	Lag Time	
n2_tcp	Primary	Active	Connected	0.22%	0.00000s	Modify
n1_tcp	HDR	Inactive	Disconnected	-	-	Modify
n3_tcp	RSS	Active	Connected	8.23%	0.00142s	Modify

Below the table, the 'Server Info' section provides details for the Primary server (n2_tcp):

```

Server Type: Primary
Version: 11.70.UC2GE
Server Time: 09:34:40
Boot Time: 06-29 05:00
Up Time: 00:34:04
Sessions: 6
Max Users: 9
Operating System
Total Mem: 377 MB
Free Mem: 183 MB
  
```

Dieser Gastbeitrag wurde uns freundlicherweise von Herrn Michael Spinnenhirn der Firma Bereos OHG zugesandt. Bereos ist ein Systemhaus in Tettnang am Bodensee, dessen Kernkompetenzen u.a. in der Administration, im Consulting und in der Durchführung von Schulungen liegen. Bereos bietet zudem selbst erstellte Lösungen im OpenSource Umfeld an wie z.B. den „Bereos Hotspot“.

Michael Spinnenhirn hat mehr als 12 Jahren praktische Erfahrung als Administrator, Consultant und Trainer im INFORMIX und Linux Umfeld. Er ist IBM Informix Certified System Administrator Version 10, Version 11, sowie IBM Certified Database Associate Informix 11.70 Fundamentals.

Weitere Informationen finden sie unter: <http://www.bereos.eu/>

Vielen Dank !

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.

Die Termine sind:

19.-21. September 2011 München

(ZMCG)

WebTipp: Artikel auf Developerworks zum Thema Grid Replication

Auf Developerworks ist ein sehr guter Artikel zum Thema Grid Replication erschienen. An Beispielen wird die Funktionsweise des Grid erklärt. Nutzen Sie den Artikel als Ergänzung zu den Beispielen hier im Newsletter und testen Sie selbst.

Der Link:

<http://www.ibm.com/developerworks/data/library/techarticle/dm-1106informixflexgrid/index.html>

WebTipp: INFORMIX integration (by Lester Knutsen)

Auf YouTube findet man einen Beitrag von Lester Knutsen, einem langjährigen und erfahrenen INFORMIX DBA, über die Vorteile der INFORMIX Integration:

http://www.youtube.com/watch?v=dW3Y70moMoc&feature=player_embedded

WebTipp: INFORMIX Redbook zur Grid Replikation

Seit einigen Tagen gibt es ein ausführliches Redbook zur Grid Replikation.

Wer sich näher mit dem Thema beschäftigen will oder die Grid Replikation in naher Zukunft einsetzen möchte, sollte sich dieses Redbook einmal ansehen:

<http://www.redbooks.ibm.com/abstracts/sg247937.html?Open>

Versionsinfo: 11.70.xC3 ist verfügbar

Seit einigen Tagen ist die Version 11.70.xC3 für alle unterstützten Plattformen und Editionen verfügbar.

Eine Vorstellung der neuen Features der 11.70.xC3 finden Sie in der Ausgabe August des INFORMIX Newsletters. Seien sie gespannt z.B. auf die Möglichkeiten der Migration einer 8-Bit-Datenbank auf UTF8.

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> **NEU !!!**
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: Carina Kaluzinski