

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....1
 TechTipp: SQL - Connect by.....2
 TechTipp: INFORMIX 11.70 – dbschema (neue Optionen).....5
 TechTipp: INFORMIX 11.70 – „dbschema -c [-ns]“.....5
 TechTipp: INFORMIX 11.70 – „dbschema / dbexport ... -nw“.....6
 TechTipp: INFORMIX 11.70 – „dbschema ... -it <isolation> -l <time>“.....7
 TechTipp: Substring mit negativer Startadresse.....8
 TechTipp: Table Level Restore – Hinweis zu onbar.....9
 TechTipp: onlicinfo.....9
 Referenzkunden: Technische Universität Berlin.....9
 Termine: INFORMIX 11.70 – 3-tägiges Bootcamp mit Hands-on Übungen.....10
 WebTipp: Le village Informix.....11
 Termine: IUG Workshops 2011.....11
 Versionsinfo: 11.50.xC8W1 ist verfügbar.....11
 Anmeldung / Abmeldung / Anmerkung.....12
 Die Autoren dieser Ausgabe.....12

Aktuelles

Liebe Leserinnen und Leser,

wir hoffen, dass Sie gut in das Neue Jahr gestartet sind.
 Nach viel Schnee und Kälte kam dann im Januar gleich der Frühling in den Süden. Frühling bei INFORMIX bedeutet mit neuer Kraft und vielen Veranstaltungen ins Jahr zu starten.
 Mit dem „Hands-On“ Workshop zu INFORMIX 11.70 und den IUG Workshops stehen schon bald wichtige Termine an. In dieser Ausgabe stellen wir, wie angekündigt, weitere neue Features der Version 11.70 vor.



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.
 Ihr TechTeam

TechTipp: SQL - Connect by

Mittels „connect by“ kann die Selbstreferenz einer Tabelle mit hierarchischer Struktur einfach abgefragt werden. Ohne die Syntax von „connect by“ wäre die Ermittlung der Hierarchieebenen innerhalb der selben Tabelle wohl nur mittels einer Stored Procedure abbildbar gewesen.

Beispiel: Anzeige der Versionen und Seriennummern zu einer Installation mit allen vorgenommenen Updates.

```
create table informix_srn (
srn          char(12),
description  char(8),
version      char(12),
master_srn   char(12)
);

insert into informix_srn values ("AAA#J001122","ONLINE"," 5.03.UC6","");
insert into informix_srn values
      ("AAA#J131263","ONLINE"," 7.12.UC4","AAA#J001122");
insert into informix_srn values ("AAA#J231297","IDS"," 9.21.UC4","AAA#J131263");
insert into informix_srn values ("AAA#J160200","IDS","10.00.UC6","AAA#J231297");
insert into informix_srn values ("AAA#J240876","IDS","11.50.UC3","AAA#J160200");
insert into informix_srn values
      ("AAA#J180185","INFORMIX","11.70.UC2","AAA#J240876");

select srn, version, description,
master_srn, CONNECT_BY_ISCYCLE cycle, CONNECT_BY_ISLEAF leaf, level
from informix_srn
start with srn = "AAA#J160200"
connect by NOCYCLE prior srn = master_srn
order by version ;
```

Ergebnis:

srn	version	description	master_srn	cycle	leaf	level
AAA#J160200	10.00.UC6	IDS	AAA#J231297	0	0	2
AAA#J240876	11.50.UC3	IDS	AAA#J160200	0	0	3
AAA#J180185	11.70.UC2	INFORMIX	AAA#J240876	0	1	4

Will man die gesamte Hierarchie sehen (also auch die Vergangenheit), so hängt man die umgekehrte Abfrage per UNION an:

```
select srn, version, description, master_srn , CONNECT_BY_ISCYCLE cycle,
CONNECT_BY_ISLEAF leaf, level
from informix_srn
start with srn = "AAA#J160200"
connect by NOCYCLE prior srn = master_srn
union
select srn, version, description, master_srn , CONNECT_BY_ISCYCLE cycle,
CONNECT_BY_ISLEAF leaf, level
from informix_srn
start with srn = "AAA#J160200"
connect by NOCYCLE prior master_srn = srn
order by version ;
```

Ergebnis:

AAA#J001122	5.03.UC6	ONLINE		0	1	3
AAA#J131263	7.12.UC4	ONLINE	AAA#J001122	0	0	2
AAA#J231297	9.21.UC4	IDS	AAA#J131263	0	0	1
AAA#J160200	10.00.UC6	IDS	AAA#J231297	0	0	2
AAA#J240876	11.50.UC3	IDS	AAA#J160200	0	0	3
AAA#J180185	11.70.UC2	INFORMIX	AAA#J240876	0	1	4

Der „connect by“ kann zusätzliche Informationen mittels Aliasnamen liefern:

- > CONNECT_BY_ISCYCLE Eintrag referenziert sich selbst
- > CONNECT_BY_ISLEAF Eintrag ist Ende der Kette
- > level Anzahl der Ebenen vom Startknoten aus

Um den Ursprung der Ausgabezeilen anzugeben, kann der Alias „CONNECT_BY_ROOT“ ausgegeben werden, der in jeder Zeile den Ursprung angibt:

Im Beispiel:

```
select srn, version, description, master_srn, CONNECT_BY_ROOT master_srn as
master
from informix_srn
start with srn = "AAA#J231297"
connect by NOCYCLE prior srn = master_srn
```

Ergebnis:

srn	version	description	master_srn	master
AAA#J231297	9.21.UC4	IDS	AAA#J131263	AAA#J131263
AAA#J160200	10.00.UC6	IDS	AAA#J231297	AAA#J131263
AAA#J240876	11.50.UC3	IDS	AAA#J160200	AAA#J131263
AAA#J180185	11.70.UC2	INFORMIX	AAA#J240876	AAA#J131263

Die Abhängigkeiten können auch als Pfad dargestellt werden mit der Option „SYS_CONNECT_BY_PATH()“:

Beispiel:

```
create table sponsoring (
prn          char(6),
vorname     char(18),
jobrole     char(18),
manager_prn char(6)
);

insert into sponsoring values ("000001","Werner","Senior Sponsor","000000");
insert into sponsoring values ("000042","Gerd","Sponsor","000001");
insert into sponsoring values ("000043","Ursula","Sponsor","000001");
insert into sponsoring values ("000142","Carmen","Schueler","000042");
insert into sponsoring values ("000143","Carina","Schueler","000042");

select prn, SYS_CONNECT_BY_PATH(trim(vorname),'/')
from sponsoring
start with vorname = "Werner"
connect by NOCYCLE prior prn = manager_prn
order by prn
```

Das Ergebnis in Form von Pfaden:

```
prn          000001
expression_1 /Werner
prn          000042
expression_1 /Werner/Gerd
prn          000043
expression_1 /Werner/Ursula
prn          000142
expression_1 /Werner/Gerd/Carmen
prn          000143
expression_1 /Werner/Gerd/Carina
```

Die Reihenfolge im „order by“ kann nach jeder Spalte erfolgen.

```
select prn, vorname, manager_prn, level
from sponsoring_x
start with vorname = "Werner"
connect by NOCYCLE prior prn = manager_prn
order by prn
```

prn	vorname	manager_prn	level
000001	Werner	000000	1
000042	Gerd	000001	2
000043	Ursula	000001	2
000142	Carmen	000042	3
000143	Carina	000042	3

Hierarchische Abfragen erlauben zudem eine Sortierung innerhalb der Hierarchieebenen. Dies wird mit einem „order siblings by“ angegeben.

```
select prn, vorname, manager_prn, level
from sponsoring_x
start with vorname = "Werner"
connect by NOCYCLE prior prn = manager_prn
order siblings by prn
```

prn	vorname	manager_prn	level
000001	Werner	000000	1
000042	Gerd	000001	2
000142	Carmen	000042	3
000143	Carina	000042	3
000043	Ursula	000001	2

Die Funktion „connect by“ kann zudem genutzt werden, um eine Zahlenreihe zu generieren.

Beispiel:

```
SELECT level FROM sysmaster:sysdual CONNECT BY LEVEL <= 13
```

Ergebnis:

```
level
1
2
...
13
```

TechTipp: INFORMIX 11.70 – dbschema (neue Optionen)

Mit Version 11.70 wurde die Funktion „dbschema“ um nützliche Optionen erweitert. Die Erweiterungen umfassen zahlreiche Bereiche, wie z.B. die Berechtigungen und die Angabe der Eigentümer der Tabellen und anderer Objekte. Der Aufbau der DBSpaces und Chunks kann als Script erstellt werden. Die Konfiguration des Physical Log und der Logical Logs kann in unterschiedlicher Darstellung als Skript erstellt werden.

```
dbschema { [-q] [-t tablename] [-s user] [-p user] [-r rolename] [-f procname]
           [-hd tablename] -d dbname [-w passwd] [-seq sequence] [-l [num]]
           [-u [ia] udtname [all]] [-it [Type]] [-ss [-si]] [filename]
           [-sl length] } | [-nw]
           { [-q] -c [-ns] [filename] }
```

In den nachfolgenden Artikeln finden Sie Beispiele zu einigen neuen Parametern.

TechTipp: INFORMIX 11.70 – „dbschema -c [-ns]“

Diese Option kann mit und ohne Angabe einer Datenbank genutzt werden. Die Option „-c“ gibt die Befehle aus, mit denen die Konfiguration der DBSpaces und Chunks, sowie des Physical Log und der Logical Logs neu erstellt werden könnte.

Beispiel:

```
dbschema -c
DBSCHEMA Schema Utility          INFORMIX-SQL Version 11.70.UC2
-- Dbspace 1 -- Chunk 1
-- EXECUTE FUNCTION TASK
('create dbspace', 'rootdbs', '/DBS/rootdbs', '1000000', '128', '2', '500', '200');
-- Dbspace 2 -- Chunk 2
EXECUTE FUNCTION TASK
('create dbspace', 'logdbs', '/DBS/logdbs', '1000000', '128', '2', '100', '100');
-- Dbspace 3 -- Chunk 3
EXECUTE FUNCTION TASK
('create dbspace', 'datadbs', '/DBS/datadbs', '2000000', '128', '2', '100', '200');
...
-- Physical Log
EXECUTE FUNCTION TASK ('alter pllog', 'rootdbs', '100000');
-- Store pre-existing logical logs information before create new logical logs
DATABASE sysadmin;
CREATE TABLE llog (log smallint, flags smallint);
INSERT INTO llog SELECT number, flags FROM sysmaster:syslogfil;
-- Logical Log 7
EXECUTE FUNCTION TASK ('add log', 'logdbs', '10000');
```

...
Diese Ausgabe beinhaltet den Aufbau der Instanz und könnte z.B. im dbaccess eine neue Instanz als Kopie der existierenden Instanz erstellen.

Wird dem Aufruf „-ns“ angefügt, so werden die Befehle in „Non-SQL“ erstellt. Dies sind Aufrufe von „onspaces“, „onparams“ und „onmode“.

Das Beispiel hat dann folgende Ausgabe:

```
dbschema -c -ns
```

```
DBSCHEMA Schema Utility          INFORMIX-SQL Version 11.70.UC2
# Dbspace 1 -- Chunk 1
# onspaces -c -d rootdbs -k 2 -p /DBS/rootdbs -o 128 -s 1000000 -ef 500 -en 200
# Dbspace 2 -- Chunk 2
onspaces -c -d logdbs -k 2 -p /DBS/logdbs -o 128 -s 1000000 -ef 100 -en 100
# Dbspace 3 -- Chunk 3
onspaces -c -d datadbs -k 2 -p /DBS/datadbs -o 128 -s 2000000 -ef 100 -en 200
...
# Physical Log
onparams -p -s 100000 -d rootdbs -y
# Store pre-existing logical logs information before create new logical logs
dbaccess sysadmin << END
CREATE TABLE llog (log smallint, flags smallint);
INSERT INTO llog SELECT number, flags FROM sysmaster:syslogfile;
END
# Logical Log 7
onparams -a -d logdbs -s 10000
...
```

TechTipp: INFORMIX 11.70 – „dbschema / dbexport ... -nw“

Sowohl beim Erstellen des Datenbankschemas, als auch beim Exportieren einer Datenbank werden die Ersteller der Objekte (Tabellen, Indizes, Prozeduren, ...) mit ausgegeben. Soll die Datenbank auf einem neuen System mit einem einheitlichen Eigentümer der Objekte erstellt werden, musste bisher der „Owner“ im SQL-Script an allen Vorkommen gelöscht bzw. ersetzt werden. Da der Owner z.B. beim „create index“ nicht nur bei Indexnamen, sondern auch beim Namen der Tabelle vorangestellt wird, kann dies sehr aufwändig sein.

Die neue Option „-nw“ bietet nun die Möglichkeit den „Owner“ der Objekte zu unterdrücken, so dass dieser beim Erstellen neu vergeben wird.

Anmerkung:

In Version 11.70.xC1 werden mit der Option -nw nur die Owner der Tabellen unterdrückt. Bei den Indizes und Berechtigungen bleibt der Owner bisher noch erhalten. Dieses Problem soll mit Version 11.70.xC2 behoben sein.

Beispiel:

```
dbschema -d kalu -t mx42
```

```
DBSCHEMA Schema Utility          INFORMIX-SQL Version 11.70.UC2
{ TABLE "kalu".mx42 row size = 301 number of columns = 6 index size = 103 }
create table "kalu".mx42
(
  id serial not null ,
  vorname char(42),
  name char(42),
  gebdat date,
  tel char(80),
  mail varchar(128)
);
```

Mit der Option „-nw“:

```
dbschema -d kalu -t mx42 -nw
```

```
DBSCHEMA Schema Utility      INFORMIX-SQL Version 11.70.UC2
{ TABLE mx42 row size = 301 number of columns = 6 index size = 103 }
create table mx42
(
  id serial not null ,
  vorname char(42),
  name char(42),
  gebdat date,
  tel char(80),
  mail varchar(128)
);
```

TechTipp: INFORMIX 11.70 – „dbschema ... -it <isolation> -l <time>“

Der Aufruf von „dbschema“ erfolgte bisher immer in der Default Isolationsstufe der Datenbank. Bei Datenbanken ohne Logging war dies der „Dirty Read“, bei Datenbanken mit Logging der „Committed Read“ (auch bei Datenbanken im Mode ANSI).

Wurde während des Aufrufs von „dbschema“ eine Strukturänderung vorgenommen (create oder drop von Tabellen, Indices, temporären Tabellen, ...), so brach der Aufruf mit einer Fehlermeldung ab. Um auf diesen Ablauf Einfluss nehmen zu können, wurden die Parameter „-it“ und „-l“ eingeführt.

Mittels „-it“ kann das Isolations Level des Aufrufs beeinflusst werden. Hierbei sind folgende Werte möglich:

- DR Dirty Read
- CR Committed Read
- CS Cursor Stability
- CRU Committed Read with RETAIN UPDATE LOCKS
- CSU Cursor Stability with RETAIN UPDATE LOCKS
- DRU Dirty Read with RETAIN UPDATE LOCKS
- RR Repeatable Read
- LC Read Last Committed

Der Parameter „-l“ ermöglicht zudem das Defaultverhalten bei Sperrern von „NOT WAIT“ auf eine bestimmte Wartezeit in Sekunden zu setzen.

In den folgenden Beispielen wurden unterschiedliche Werte für die Isolationsstufe beim Aufruf von „dbschema“ auf eine Datenbank mit Logging genutzt.

Dazu ist jeweils die Anzeige des „onstat -g sql“ aufgelistet:

```
dbschema -d ibm -ss
Id      Stmt type      Database      Lvl Mode      ERR  ERR  Vers Expl.
62      SELECT          ibm           CR  Not Wait    0    0    9.24 Off

dbschema -d ibm -ss -it DR
64      SELECT          ibm           DR  Not Wait    0    0    9.24 Off

dbschema -d ibm -ss -it CS
67      SELECT          ibm           CR  Not Wait    0    0    9.24 Off

dbschema -d ibm -ss -it RR
69      SELECT          ibm           RR  Not Wait    0    0    9.24 Off

dbschema -d ibm -ss -it LC
73      SELECT          ibm           LC  Not Wait    0    0    9.24 Off

dbschema -d ibm -ss -it CR -l 42
74      SELECT          ibm           CR  Wait 42     0    0    9.24 Off
```

TechTip: Substring mit negativer Startadresse

Um aus Daten die letzten x Stellen zu ermitteln, musste man bisher trickreich mit `substring()` und `length()` arbeiten. Die Funktion `substr()` erlaubt mittlerweile auch negative Werte für die Startposition. Diese wird dann vom Ende der Zeichenkette her ermittelt. Damit lassen sich z.B. die letzten x Zeichen extrahieren.

Beispiel:

```
create table subtest ( f1 varchar(20) );
insert into subtest values ("mx42");
insert into subtest values ("carmen23");
insert into subtest values ("carinal6");
insert into subtest values ("julia17");
```

Die Abfrage „`select f1, substr(f1,-2,2) from subtest`“ liefert:

```
mx42          42
carmen23      23
carinal6      16
julia17       17
```

Ohne negative Startadresse wäre folgende Syntax notwendig:

```
select f1, substr(f1,length(f1)-1,2) from subtest;
```

TechTipp: Table Level Restore – Hinweis zu onbar

Der Parameter AC_TAPEBLOCK in der Konfigurationsdatei AC_CONFIG lässt vom Namen her eigentlich vermuten, dass dieser nur auf Sicherungen mit „ontape“ wirkt. Beim Table Level Restore aus einer Sicherung mit „ontape“ wird der Parameter, falls er nicht explizit gesetzt ist, auf den Wert von TAPEBLK der ONCONFIG gesetzt.

Bei einem Table Level Restore einer Sicherung die mit „onbar“ gezogen wurde, muss dieser explizit auf die Blockung gesetzt werden, die bei der Sicherung eingestellt war. Diese ist durch den Parameter **BAR_XFER_BUF_SIZE** der ONCONFIG und die System Pagesize bestimmt.

Damit muss in der Konfigurationsdatei der Wert AC_TAPEBLOCK auf **BAR_XFER_BUF_SIZE*2** bei System-Pagesize 2k bzw. **BAR_XFER_BUF_SIZE*4** bei System-Pagesize 4k gesetzt werden.

Der Default für den Wert BAR_XFER_BUF_SIZE steht auf 31.

Somit müsste z.B. für einen Table Level Restore auf Linux(32) in der AC_CONFIG für AC_TAPEBLOCK der Wert 62 eingetragen werden.

TechTipp: onlicinfo

Ein neues Programm namens „onlicinfo“ ist unter \$INFORMIXDIR/bin zu finden. Dahinter steckt nichts als eine Ausgabe aller Informationen, die sich in der Tabelle syslicenseinfo befinden:

```
secho "select * from syslicenseinfo" | dbaccess sysmaster
```

Die Tabelle syslicenseinfo wurde bereits im INFORMIX Newsletter Januar 2009 ausführlich beschrieben.

Referenzkunden: Technische Universität Berlin

Die Technische Universität Berlin hat in mit dem Einsatz von **IBM Informix 11.5** mit Continuous Availability, dem Informix Online Connection Manager sowie Server Monitor und IBM I-Spy eine Umgebung geschaffen, die performant und ausfallsicher alle Anforderungen an eine hochmoderne Datenbank erfüllt. Hierbei wurden mehrere Datenbanken unterschiedlicher Hersteller in zwei INFORMIX Instanzen zusammengefasst. Mit Hilfe des OpenAdminTools kann die Administration nach Aussagen der Administratoren in der halben Zeit wie bisher erledigt werden.

Die sehr gute Performance bei hoher Ausfallsicherheit soll in den nächsten Jahren auf weitere Systeme ausgeweitet werden.

Mehr Details finden Sie unter folgendem Link:

http://www.ibm.com/software/success/cssdb.nsf/CS/JHAL-8BZGAA?OpenDocument&Site=software&cty=en_us

Termine: INFORMIX 11.70 – 3-tägiges Bootcamp mit Hands-on Übungen

Im Rahmen der Veranstaltungsreihe INFORMIX 11.70 Bootcamp läuft derzeit die Anmeldung zum kostenlosen dreitägigen Hands-on Workshop.
Im Anschluss an den Workshop kann eine Zertifizierung durchgeführt werden.

Folgende Termine werden derzeit angeboten:

22.-24. Februar 2011	Österreich	Wien
05.-07. April 2011	Deutschland	Düsseldorf

Service für unsere Leser im Rest der Welt:

14.-16. Februar 2011	Australien	Sydney
21.-23. Februar 2011	Australien	Melbourne
28. Februar-02. März 2011	Mexico	Mexico City

Die Agenda:

1. Tag:

- 1.0 Welcome (5 min)
- 1.1 Informix Technical Survival Guide (15 min)
- 1.2 Informix Fundamentals (60 min)
- 1.3 Informix Database Administration (75 min)
- 1.4 Informix Database Administration (Hands-on) (60 min)
- 1.5 Informix Performance Tuning and Troubleshooting (75 min)
- 1.6 Informix Performance Tuning (Hands-on) (60 min)

2. Tag:

- 2.1 Informix Availability and Scalability (75 min)
- 2.2 Informix Availability and Scalability (Hands-on) (75 min)
- 2.3 Informix Security (45 min)
- 2.4 Informix Security (Hands-on) (45 min)
- 2.5 Upgrading and Migrating Informix Databases (45 min)
- 2.6 Upgrading and Migrating Informix Databases (Hands-on) (45 min)

3. Tag:

- 3.1 Informix Feature Spotlight (75 min)
- 3.2 Informix Embeddability (45 min)
- 3.3 Informix Embeddability (Hands-on) (60 min)
- 3.4 Informix Application Development Overview (60 min)
- 3.5 Informix Application Development (Hands-on) (60 min)
- 3.6 Summary (15 min)
- 3.7 Informix Mastery Test (Hands-on) (75 min)

Weitere Informationen, sowie den Link zur Anmeldung finden Sie unter:

<http://www.ibm.com/developerworks/wikis/display/im/Informix+11.7+Bootcamp>

WebTipp: Le village Informix

Die Gallier formieren sich und haben einen neuen Blog ins Leben gerufen. Wer entsprechende Sprachkenntnisse hat, sollte mal rein schauen was die Gallier da schreiben:

<http://levillageinformix.blogspot.com/>

Bereits das Logo dieses Blogs lässt auf große Taten hoffen:



Le village Informix

Seit einiger Zeit sind mehr und mehr Aktivitäten zu INFORMIX in Frankreich zu beobachten. Nach dem sehr informativen französischen INFORMIX Newsletter sind einige Blogs rund um INFORMIX in Frankreich entstanden.

Termine: IUG Workshops 2011

Die IUG plant für 2011 wieder drei Workshops in unterschiedlichen Regionen Deutschlands. Reservieren Sie sich diese Termine gleich im Kalender. Ein Besuch lohnt sich, sowohl um aktuelle technische Informationen zu erhalten, als auch um Kontakte zu knüpfen oder aufzufrischen. Wie immer findet am Vorabend der beliebte IUG Stammtisch statt.

Für IUG-Mitglieder sind die Workshops kostenfrei. Wer noch nicht IUG Mitglied ist, kann sich unter info@iug.de über die Vorteile einer Mitgliedschaft informieren.

Als Termine sind vorgesehen:

ORT	WANN	THEMENSCHWERPUNKT
➤ Stuttgart	KW 11	Aktuelle Lizenzmodelle
➤ Wiesbaden	KW 21	Hochverfügbarkeit / Replikation / Cloud Computing Monitoring / Performance
➤ Hamburg	KW 42	Applikationsentwicklung Backend zu Business Lösungen

Weitere Informationen und Anmeldung unter:

http://www.iug.de/index.php?option=com_content&task=view&id=5&Itemid=17

Versionsinfo: 11.50.xC8W1 ist verfügbar

Seit einigen Tagen ist die Version 11.50.xC8W1 für alle unterstützten Plattformen und Editionen verfügbar.

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.

Das Archiv der bisherigen Ausgaben finden Sie zum Beispiel unter:

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: Gerd Kaluzinski (Redaktionsgarten)