

## Willkommen zum Informix Newsletter

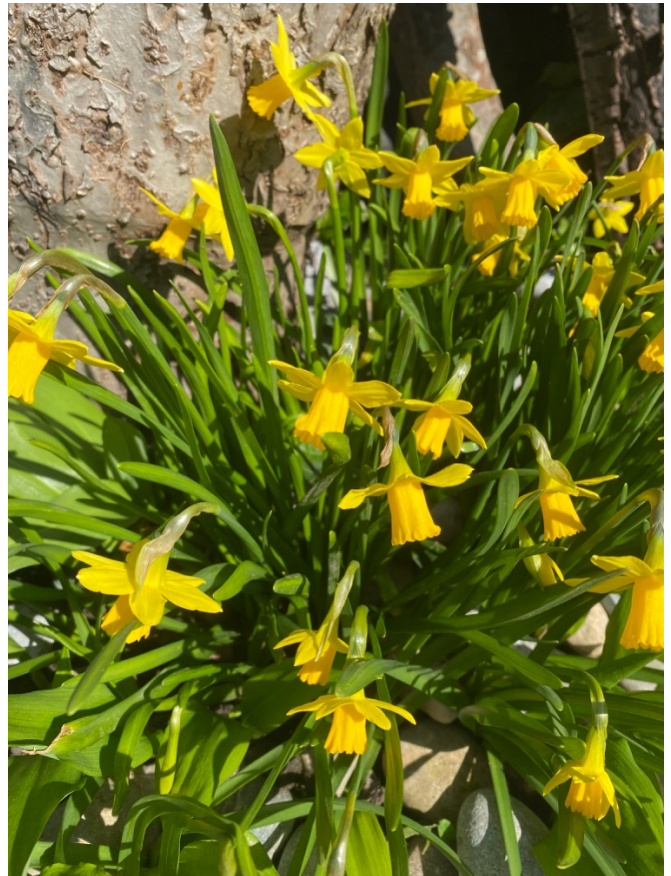
Liebe Leserinnen und Leser,

schon wieder ist ein Quartal vorüber, in dem sich sehr viel getan hat. Alle Ostereier sind inzwischen gefunden und verzehrt (Schokolade hat in der Redaktion eine sehr sehr kurze Haltbarkeitsdauer).

Auf weltweiter Ebene haben wir eine Kooperation mit dem "IIUG Insider" gestartet, der von der International Informix User Group (IIUG) erstellt wird. Bei einigen Themen gab es bereits Überschneidungen. Wir werden daher in Zukunft Artikel in beiden Richtungen austauschen (und übersetzen).

Wie bereits im letzten Quartal setzen wir die Reihe an nützlichen Datablades fort. Zudem stellen wir neue Features der Version 15.0.1, sowie bisher im Newsletter nicht beachtete Parameter und Funktionen vor.

Viel Spass mit dem aktuellen Newsletter !  
Ihr TechTeam



**Inhaltsverzeichnis**

TechTipp: ONCONFIG - DUMPSHMEM.....	3
TechTipp: ONCONFIG - DUMPCNT .....	3
TechTipp: ONCONFIG - STRING_TRUNCATE_ERROR .....	4
TechTipp: Environment - STRING_TRUNCATE_ERROR.....	5
TechTipp: CDR - list catalog .....	5
TechTipp: LOTOFILE - Prefix.....	6
TechTipp: AVG_LO_SIZE .....	7
TechTipp: Funktionen even() / odd() .....	8
TechTipp: SHM-Listener unter Ubuntu.....	8
TechTipp: DBMS_RANDOM Package .....	9
TechTipp: TEMP TABLES auf HDR/RSS Servern .....	10
TechTipp: Migration Tasks und Scheduler.....	11
TechTipp: ONCONFIG - TEMPTAB_NOLOG.....	12
TechTipp: dbinfo() - Edition und Hersteller .....	12
TechTipp: Version 15.0.1.4 ist verfügbar.....	12
In eigener Sache: HEALTH CHECK auf HDR/RSS Servern.....	13
Nutzung des INFORMIX Newsletters .....	13
Die Autoren dieser Ausgabe.....	14

## TechTipp: ONCONFIG - DUMPSHMEM

Der Parameter DUMPSHMEM wird kaum beachtet, und spielt bei einem stabilen System meist keine Rolle.

Wichtig wird dieser Parameter dann, wenn ein Problem auftritt, und der Support im Shared Memory Dump nach der Ursache suchen will.

Die möglichen Werte des Parameters sind, dass bei einem ernsthaften Problem:

- 0 - keinen Abzug (dump) des Shared Memory erzeugt wird.
- 1 - ein Abzug des gesamten Speichers der Instanz erstellt wird. (Default)
- 2 - ein Abzug des Speichers der Instanz (ausser der Bufferpools) erstellt wird.
- 5 - nur bei einem Absturz (Crash der Instanz) wiederholt ein Speicherabzug erstellt wird.  
Die Anzahl wird über den Parameter DUMPCNT bestimmt.
- 6 - nur bei einem Absturz (Crash der Instanz) wiederholt ein Speicherabzug erstellt wird, der jedoch nicht die Bufferpools beinhaltet.  
Die Anzahl wird über den Parameter DUMPCNT bestimmt.

Da ein Abzug des gesamten Speicherbereiches zu Problemen mit dem Plattenplatz führen kann, und dieser zudem einige Minuten dauern kann, bis danach die Instanz Offline geht, wird üblicherweise der Parameter in der Produktionsumgebung auf den Wert 0 gesetzt.

Zur Sicherheit könnte der Wert 2 gewählt werden, damit im Ausnahmefall ein Speicherabzug ohne Bufferpools zur Verfügung steht.

Die Werte 1, 5 und 6 werden üblicherweise erst gesetzt, wenn der Support bei einem wiederholbaren Problem mit einbezogen werden musste, und dieser nach einem Memory Dump gefragt hat.

Die Werte 5 und 6 sind offiziell erst mit Version 14.10 dokumentiert.

(Dieser Artikel wurde vom IIUG Insider übernommen und ergänzt)

## TechTipp: ONCONFIG - DUMPCNT

Der Parameter DUMPCNT bestimmt, wie oft bei einem ernsthaften Problem ein Speicherabzug erstellt wird. Der Default Wert ist 1.

Zu beachten ist, dass abhängig vom Wert DUMPSHMEM der Speicherabzug sehr gross werden kann, so dass hier der benötigte Plattenplatz in die Betrachtung mit einfließen sollte.

## TechTipp: ONCONFIG - STRING\_TRUNCATE\_ERROR

Ein neuer Parameter der ONCONFIG wurde mit Version 15.0.1.0 eingeführt. Dieser gibt einen Fehler zurück, wenn versucht wird eine Zeichenkette in ein Feld vom Typ CHAR, VARCHAR, LVARCHAR einzufügen, dessen Inhalt beim Insert abgeschnitten würde.

Das Abschneiden ohne Warnung und Information ist besonders fatal, wenn dies auf einer verschlüsselten Spalte erfolgt. Nach dem Abschneiden der Daten ist der Datensatz so "sicher verschlüsselt", dass selbst Anwender mit dem richtigen Schlüssel diesen nicht mehr lesen können:

Beispiel:

```
create table users_passwd (
    who char(18),
    passwd char(24)
);

SET ENCRYPTION PASSWORD 'Test_in_V15.0.1';
insert into users_passwd values ('kalu',encrypt_aes('informix42'));

select who, passwd, decrypt_char(passwd) as password
from users_passwd;

who                passwd                password
26012: The internal base64 decoding function failed.
Error in line 24
Near character position 15
```

Wird die Umgebungsvariable gesetzt, um das Abschneiden zu vermeiden, sieht der Aufruf so aus:

```
onmode -wf STRING_TRUNCATE_ERROR=1
Value for STRING_TRUNCATE_ERROR (1) was saved in config file.
Value of STRING_TRUNCATE_ERROR has been changed to 1.

insert into users_passwd values ('kalu',encrypt_aes('informix42'));
1279: Value exceeds string column length.
```

Vergleicht man den Inhalt mit und ohne Kürzung, so wird klar wo der Fehler liegt:

```
select encrypt_aes('informix42'), encrypt_aes('informix42')::char(24)
from systables where tabid = 1;

(constant)                (constant)
0Mc7/AAAAEAKaFxxN1pCc9LjS+93UxhStgiZXk/J15+ 0Lth/AAAAEAVmAsrI4CP4hq+
```

### Anmerkung:

Derzeit wirkt der Check nur auf CHAR, VARCHAR, NCHAR und NVARCHAR. Beim Datentyp LVARCHAR wird derzeit noch kein Fehler ausgegeben. Dieser Defekt wird in einer der nächsten Versionen behoben sein. (Defekt ist in 15.0.1.7 noch vorhanden)

## TechTipp: Environment - STRING\_TRUNCATE\_ERROR

Der Parameter STRING\_TRUNCATE\_ERROR kann auch als Session Environment gesetzt werden. Allerdings sind hier nicht die Werte 0 und 1 wie in der \$ONCONFIG gültig, sondern:

Zur aktivierung:

```
set environment STRING_TRUNCATE_ERROR 'error';
```

Zur Deaktivierung:

```
set environment STRING_TRUNCATE_ERROR 'ignore';
```

## TechTipp: CDR - list catalog

Um die Befehle zu sehen, mit der die Enterprise Replikation erstellt wurde, kann der Katalog abgefragt werden:

```
cdr list catalog [-c server] [-s] [-r] [-e] [-t] [-z] [-g] [-q] [-a]
-c server --connect=server      connect to server
-s          --server            print define server commands
-r          --replicate         print define replicate commands
-e          --replicateset      print define replicateset commands
-t          --template          print define template commands
-z          --realizetemplate    print realize template commands
-g          --grid              print define grid commands
-a          --all               print all commands
```

Die Option -a gibt sämtliche Definitionen der Enterprise Replikation aus. Dies kann sehr gut zur Dokumentation genutzt werden.

Die Definition der Server, die an der Replikation beteiligt sind, sieht z.B. so aus:

```
cdr list catalog -s

#
# cdr define server commands.
#

cdr define server --ats=/opt/informix_14.10.FC8/ATS
  --ris=/opt/informix_14.10.FC8/RIS --init x1

cdr define server --connect=x2 --ats=/opt/informix_14.10.FC2/ATS
  --ris=/opt/informix_14.10.FC2/RIS --sync=x1 --init x2

cdr define server --connect=x3 --ats=/opt/informix_14.10.FC13/ATS
  --ris=/opt/informix_14.10.FC13/RIS --sync=x1 --init x3
```

Ein Blick auf die Replikate zeigt, wie diese erstellt wurden.  
Dabei wird ein "select \*" in die vollständige Liste der Spalten aufgelöst.

```
cdr list catalog -r

#
# cdr define replicate commands.
#

cdr define replicate --conflict=ignore --firetrigger --ats --ris rep_kunden \
"deutschland@x1:informix.kunden" "select t.kdnr, t.name, t.vorname, t.plz,
t.ort, t.land, t.bundesland, t.flag from kunden t where flag<20" \
"R deutschland@x2:informix.kunden" "select t.kdnr, t.name, t.vorname, t.plz,
t.ort, t.land, t.bundesland, t.flag from kunden t" \
"europa@x3:informix.customer" "select t.cnr, t.lname, t.fname, t.zip, t.city,
t.contry, t.county, t.flag from customer t where flag!=100"
```

## TechTipp: LOTOFILE - Prefix

Um SmartBlobs aus der Datenbank auszulesen, gibt es die Funktion lotofile().  
Wird der Name der Datei, in die der Inhalt der blob/clob Spalte ausgegeben werden soll,  
mit einem Ausrufezeichen beendet, so wird der Dateiname unverändert genutzt.  
Fehlt dieses Ausrufezeichen, wird eine eindeutige ID an den Dateinamen angefügt.  
Somit können mehrere Datensätze jeweils in einer eigenen Datei ausgegeben werden.

Z.B. bei der Datei "foto.jpg" die folgenden Dateien:

```
foto.jpg.0000000069933b6b
foto.jpg.0000000069933b6c
foto.jpg.0000000069933b6d
foto.jpg.0000000069933b6e
...
```

Um eine Beziehung zwischen Key und Ausgabedatei herzustellen, besteht die Option,  
diesen als Prefix für die Ausgabe zu verwenden:

```
select lotofile(f3, "/tmp/" || rpad(f1, 4, '_') || "foto.jpg!", "server") from
smart_test

1___foto.jpg
2___foto.jpg
3___foto.jpg
4___foto.jpg
...
```

## TechTipp: AVG\_LO\_SIZE

Über den \$ONCONFIG Parameter AVG\_LO\_SIZE lässt sich die Grösse des MetaDaten Bereichs im SmartBlobSpace beeinflussen.

Wird der Parameter nicht angegeben, so erfolgt die Berechnung der Speichergrösse der Metadaten unter der Annahme von 8k Objekten (Default).

Werden viele kleine Objekte in den SmartBlobs abgelegt, so kann dies dazu führen, dass der Bereich der MetaDaten vollläuft, obwohl im SmartBlobSpace noch reichlich Platz für weitere Einträge vorhanden ist.

Im online.log erscheint dann folgende Meldung:

```
WARNING: partition <sb spacename>:informix.LO_hdr_partn': no more pages
```

Der Check mit "oncheck -cs bzw. cS" zeigt, dass im Bereich LO\_hdr\_partn kein freier Platz mehr vorhanden ist.

Wir empfehlen daher, bei der Erstellung der SmartBlobSpaces den Wert für den Parameter AVG\_LO\_SIZE so anzugeben, dass ausreichend Platz für die MetaDaten reserviert wird.

Beispiel für die Erfassung vieler kleiner Objekte (2-5k):

```
onspaces -c -S sbtest -p /IFXDBS/sbtest -o 0 -s 1000000  
-Df LOGGING=ON,AVG_LO_SIZE=4
```

In der Dokumentation findet sich ein Task, der nachträglich den Wert für AVG\_LO\_SIZE ändert:

```
EXECUTE FUNCTION task("set sb space avg_lo_size","sbtest1","2");
```

Der Wert wird dabei in der Definition des SmartBlobSpaces geändert, die Grösse des MetaDaten Bereichs wird dadurch jedoch nicht verändert.

Sind Smart Blobs in einem "falsch angelegten" DBSpace gespeichert, so können diese nicht mittels UPDATE der Tabellendefinition zu verschoben werden:

```
alter table smart_test PUT f3 in (sbtest3,sbtest4);
```

Dieser Befehl wirkt sich **ausschliesslich auf neue Einträge** aus. Bereits gespeicherten Einträge bleiben in den bisherigen SmartBlob Spaces gespeichert.

Abhilfe bringt hier nur der LOCOPY, der die Daten umspeichert.  
(siehe Informix Newsletter 2022-Q2).

## TechTipp: Funktionen even() / odd()

In einigen relationalen Datenbanken stehen die Funktionen odd() und even() zur Verfügung. Bei Informix sind diese nicht als Default verfügbar, können aber durch einfache Funktionen bereitgestellt werden:

```
create procedure if not exists odd(x int)
returning boolean;
define res boolean;
let res = 'f';
if mod(x,2) = 1
    then let res = 't';
end if;
return res;
end procedure;
```

```
create procedure if not exists even(x int)
returning boolean;
define res boolean;
let res = 'f';
if mod(x,2) = 0
    then let res = 't';
end if;
return res;
end procedure;
```

```
select tabid, odd(tabid), even(tabid)
from systables
where tabid <20
or even(tabid)
order by 1;
```

## TechTipp: SHM-Listener unter Ubuntu

Im Linux Kernel ab Version 4.19 wurde der schreibende Zugriff von "root" auf Dateien geändert. Um mögliche Sicherheitslücken durch „attacker-controlled FIFO or regular file“ einzugrenzen, kann root manche Dateien nicht schreiben (durchaus aber löschen und neu anlegen).

Damit kann es Probleme geben, wenn eine lokale Shared Memory oder Pipe Verbindung konfiguriert wurde.

Um dieses Problem zu umgehen, sollten die "Kernel Settings" entsprechend angepasst werden.

In den MACHINE SPECIFIC NOTES steht dazu:

```
On Ubuntu 20.04 LTS the kernel setting 'fs.protected_regular' defaults to '2'. This could cause initialization errors on instances configured with a SHM listener. To support SHM connections this parameter needs to be set to '0' (e.g., 'sudo sysctl fs.protected_regular=0')
```

## TechTipp: DBMS\_RANDOM Package

Das Task Random Package verrät bereits im Namen, dass es um die Generierung von Zufallszahlen geht.

Damit das Package in der Datenbank genutzt werden kann, muss die Erweiterung "Excompat" einmalig in der Datenbank registriert werden:

```
EXECUTE FUNCTION sysbldprepare('excompat.*','create');
```

Ist dies erfolgt, stehen die Random Funktionen zur Verfügung:

- `dbms_random_initialize` # Initialisierung der Zufallsgenerierung
- `dbms_random_seed` # Neue Basis (Seed) zur Erzeugung der Zufallszahlen
- `dbms_random_random` # Ausgabe der Zufallszahl
- `dbms_random_terminate` # Beenden der Generierung der Zufallszahlen

Beispiel (5 Würfel mit jeweils 10 Würfeln):

```
execute procedure dbms_random_initialize (42);

select
  abs(mod(dbms_random_random(),6))+1 as wuerfel1,
  abs(mod(dbms_random_random(),6))+1 as wuerfel2,
  abs(mod(dbms_random_random(),6))+1 as wuerfel3,
  abs(mod(dbms_random_random(),6))+1 as wuerfel4,
  abs(mod(dbms_random_random(),6))+1 as wuerfel5
from systables
where tabid <= 10
;
```

Ergebnis:

wuerfel1	wuerfel2	wuerfel3	wuerfel4	wuerfel5
5	4	3	5	3
1	5	1	3	2
5	1	6	4	2
4	2	3	1	6
1	6	2	3	1
6	1	4	2	6
1	6	2	3	4
1	2	2	5	3
2	1	5	2	3

Die generierten Zufallszahlen sind nicht für eine Verschlüsselung bzw. Anonymisierung gedacht, da sich die Reihenfolge nach einiger Zeit wiederholen kann.

Um dies zu vermeiden, sollte die Seed (Startwert) regelmässig gewechselt werden.

Die Reihenfolge ist pseudozufällig: Wenn die gleiche Zahl als Seed verwendet wird, ist auch die Folge der anschließend gelieferten Zahlen identisch. Um das abzuschwächen, können von sich aus schwankende Werte als Seed verwendet werden. Beispielsweise die Session-ID und die Nachkommastellen der aktuellen Zeit. Damit diese für Seed verwendet werden können, müssen sie von einer Funktion geliefert werden z.B.:

```
CREATE FUNCTION current_subseconds ()
  -- Returns CURRENT FRACTION TO FRACTION(5)A
  -- so it can be used by dbms_random_initialize()
  -- and dbms_random_seed().
  -- Requires USEOSTIME=1 for subsecond precision.
RETURNING integer;

DEFINE subseconds integer;
select (current
  fraction to fraction(5))::VARCHAR(10)::DECIMAL
  *100000 into subseconds
  FROM sysmaster:sysdual;
RETURN subseconds;
END FUNCTION;

execute procedure dbms_random_seed (
  current_subseconds()
  +dbinfo("utc_current")
  +dbinfo("sessionid")
);
```

## TechTipp: TEMP TABLES auf HDR/RSS Servern

Ist ein Secondary Server vom Typ HDR oder RSS als "Read-Only" eingerichtet, so besteht trotzdem die Möglichkeit, Daten in temporäre Tabellen zu schreiben.

Sollen Daten dauerhaft geschrieben werden, so geht dies nur falls der \$ONCONFIG Parameter "UPDATABLE\_SECONDARY" auf 1 gesetzt wird, oder Remote auf den Primary Server geschrieben wird.

Temporäre Tabellen können hingegen lokal genutzt werden, wobei hierzu einige Voraussetzungen erfüllt sein müssen:

1. Die Temporäre Tabelle wird ohne Transaktionslog erstellt, indem in der \$ONCONFIG der Parameter "TEMPTAB\_NOLOG" auf 1 gesetzt wird, oder beim "create table" der Zusatz "with no log" verwendet wird.
2. Der Parameter DBSPACETEMP in \$ONCONFIG muss auf einen vorhandenen TempDBSpace verweisen, oder die Umgebungsvariable DBSPACETEMP muss auf einen existierenden TempDBSpace verweisen.

Sind diese Voraussetzungen nicht erfüllt, so wird das Erstellen einer Temporären Tabelle mit der Meldung "**not allowed on Secondary Server**" abgewiesen.

## TechTipp: Migration Tasks und Scheduler

Der Scheduler, der in der Datenbank sysadmin läuft, steuern nicht nur die Aufrufe der Tasks, die Aufgaben in der laufenden Instanz ausführen.

Bei einer Migration (Im Beispiel 15.0.1.3 auf 15.0.1.4) werden zudem die Migration Tasks aufgerufen, um die Datenbanken auf die neue Version umzustellen, und die Aufrufe der Datablades zu aktualisieren.

Auch wenn ansonsten die Scheduler deaktiviert sein sollten, macht es Sinn diese zumindest nach der Migration für einige Zeit aktiv zu lassen.

Hier der Auszug aus dem online.log, der zeigt, dass die Tasks nach der Migration der Version die Datenbanken angepasst haben:

```
2026-03-30 08:02:38.261 On-Line Mode
2026-03-30 08:02:40.147 SCHAPI: Started dbScheduler thread.
2026-03-30 08:02:40.441 Booting Language <spl> from module <>
2026-03-30 08:02:40.441 Loading Module <SPLNULL>
2026-03-30 08:02:40.546 Auto Registration is synced
2026-03-30 08:02:40.566 SCHAPI: Started 2 dbWorker threads.
2026-03-30 08:02:43.135 Defragmenter cleaner thread now running
2026-03-30 08:02:43.135 Defragmenter cleaner thread cleaned:0 partitions
2026-03-30 08:03:40.186 Loading Module <${INFORMIXDIR/ex-
tend/ifxmngr/ifxmngr.bld}>
2026-03-30 08:03:40.211 The C Language Module </opt/informix/ex-
tend/ifxmngr/ifxmngr.bld> loaded
2026-03-30 08:03:40.288 INFO (migrate) Begin migrating database stores
2026-03-30 08:03:40.290 INFO (migrate) Begin migrating database health_check
2026-03-30 08:03:40.322 INFO (migrate) Begin migrating database movin_obj
2026-03-30 08:03:40.325 INFO (migrate) Begin migrating database moving_obj
2026-03-30 08:03:40.333 INFO (migrate) Begin migrating database stores_ts
2026-03-30 08:03:41.212 INFO (migrate) stores_ts: TimeSeries.6.01.FC2 is up-to-
date
2026-03-30 08:03:41.213 INFO (migrate) stores_ts: spatial.8.22.FC4 is up-to-
date
2026-03-30 08:03:41.213 INFO (migrate) Finished migrating database stores_ts
2026-03-30 08:03:41.250 INFO (migrate) test: binaryudt.1.0 is up-to-date
2026-03-30 08:03:41.274 INFO (migrate) Finished migrating database stores
2026-03-30 08:03:41.281 INFO (migrate) moving_obj: TimeSeries.6.01.FC2 is up-
to-date
2026-03-30 08:03:41.282 INFO (migrate) moving_obj: spatial.8.22.FC4 is up-to-
date
2026-03-30 08:03:41.282 INFO (migrate) moving_obj: sts.2.00.FC2 is up-to-date
2026-03-30 08:03:41.282 INFO (migrate) Finished migrating database moving_obj
2026-03-30 08:03:41.296 INFO (migrate) Finished migrating database health_check
2026-03-30 08:03:41.329 INFO (migrate) movin_obj: TimeSeries.6.01.FC2 is up-to-
date
2026-03-30 08:03:41.329 INFO (migrate) movin_obj: spatial.8.22.FC4 is up-to-
date
2026-03-30 08:03:41.329 INFO (migrate) movin_obj: sts.2.00.FC2 is up-to-date
2026-03-30 08:03:41.330 INFO (migrate) Finished migrating database movin_obj
2026-03-30 08:08:07.458 Checkpoint Completed: duration was 0 seconds.
```

## TechTipp: ONCONFIG - TEMPTAB\_NOLOG

Der ONCONFIG Parameter TEMPTAB\_NOLOG kann neben den Werten

0 Deaktiviert (Temporäre Tabellen werden im Default mit Logging erstellt, wenn nicht explizit "with no log" angegeben wird)

1 Aktiviert (Temporäre Tabellen werden ohne Logging erstellt)

zusätzlich noch den Wert

2 Server Specific configuration parameter

Am Primary Server : Verhalten wie bei Wert 0

Am Secondary Server : Verhalten wie bei Wert 1

annehmen.

Dies kann z.B. in einer HDR oder RSS Replikation angewendet werden, falls am Primary auch Temporäre Tabellen mit Transaktionslogging genutzt werden sollen.

## TechTipp: dbinfo() - Edition und Hersteller

Die dbinfo() Funktion wurde um weitere Argumente erweitert.

Besonders hilfreich ist dabei die Möglichkeit der Abfrage der Edition.

Hierüber könnte z.B. gesteuert werden, ob PDQ verwendet werden darf oder nicht.

Zusätzlich kann der Hersteller abgefragt werden (hier kommen HCL und IBM als Ausgabe in Frage).

Beispiel:

```
select dbinfo('version','edition') as edition, dbinfo('version','vendor') as vendor
from systables
where tabid = 1
```

```
edition vendor
AEE IBM
```

```
1 row(s) retrieved.
```

## TechTipp: Version 15.0.1.7 ist verfügbar

Die Version 15.0.1.7 steht in FixCentral zum Download bereit.

Wie immer wurden Verbesserungen der letzten Wochen in dieses Release integriert.

## **In eigener Sache: HEALTH CHECK auf HDR/RSS Servern**

Auf vielfachen Wunsch unserer Kunden haben wir den Informix Health Check nun auf die Instanzen HDR und RSS (derzeit nur ein RSS) erweitert.

Da je nach Lizenz auf dem HDR und RSS auch gearbeitet werden darf, macht es Sinn auch hier die Cache Rate, die Deadlocks und weitere Performance Metriken zu analysieren. Zudem werden die Meldungen im online.log ausgewertet, die ggf. auf Probleme hinweisen könnten.

Das neue Installationspackage des Health Checks (Version 1.15) enthält diese Option und kann von allen Bestandskunden des Health Checks als Update auf den bisherigen Check angefordert werden. Der Check des HDR/RSS erfolgt zusätzliche Kosten.

Unterstützt werden alle Informix Versionen ab 14.10.FC6 (auf Unix/Linux).

## **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 Webseiten 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.

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.

Wir würden uns über eine Information freuen, wann und wo unsere Inhalte weiterverbreitet werden.

## Die Autoren dieser Ausgabe

<b>Andreas Legner</b>	<b>INFORMIX Development</b> <b>HCL Software</b>
<b>Martin Fuerderer</b>	<b>Database Development</b> <b>HCL Software</b>
<b>Gerd Kaluzinski</b>	<b>Delivery Consultant</b> <b>IBM Expert Labs</b> <a href="mailto:gerd.kaluzinski@de.ibm.com">gerd.kaluzinski@de.ibm.com</a>
<b>Manuel Müller</b>	<b>Delivery Consultant</b> <b>IBM Expert Labs</b> <a href="mailto:manuel.mueller@de.ibm.com">manuel.mueller@de.ibm.com</a>

Über die Kooperation mit dem IIUG Insider wurden Artikel von dort übernommen.  
Vielen Dank an **Gary Ben-Israel** (IIUG Insider Editor)

Herzlichen Dank an die vielen Helfer im Hintergrund.

Nicht zu vergessen der Dank an die Informix User Group, ohne die es den INFORMIX Newsletters heute nicht mehr geben würde, und die dankenswerterweise die Verteilung übernimmt.

Foto Nachweis: Osterglocken im Redaktionsgarten

(Gerd Kaluzinski)