

## Willkommen zum „IBM Informix Newsletter“

### Inhaltsverzeichnis

|  |    |
|--|----|
| Aktuelles.....   | 1  |
| TechTipp: JDBC – Nutzung der Datei INFORMIXSQLHOSTS.....               | 2  |
| TechTipp: Scheduler Stoppen / Starten.....                             | 3  |
| TechTipp: Global Security Kit (gskit).....                             | 3  |
| TechTipp: onpassword – Login und Password verschlüsselt speichern..... | 4  |
| TechTipp: cdr check repl --repair repl / cdr sync repl.....            | 5  |
| TechTipp: IWA 12.10 Feature: Enhanced Query Support.....               | 7  |
| TechTipp: First steps mit NoSQL – Gastbeitrag der CURSOR AG.....       | 8  |
| WEBTIPP: Lizenzinformationen - Firma OMTCO.....                        | 11 |
| Versionsinfo: 11.70.xC8 ist verfügbar.....                             | 12 |
| Anmeldung / Abmeldung / Anmerkung.....                                 | 12 |
| Die Autoren dieser Ausgabe.....  | 13 |

### Aktuelles

Liebe Leserinnen und Leser,

auch in diesem Jahr wollen wir Sie Monat für Monat mit Tipps und Tricks rund um Informix versorgen.

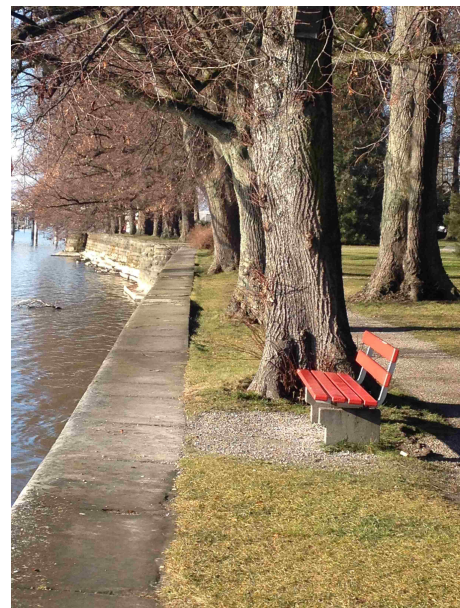
Obwohl der Winter bisher eher einem Frühjahr gleicht, starten wir mit der Ausgabe Januar und hoffen, dass der Schnee nicht pünktlich zum Frühlingsanfang doch noch kommt.

Vielen Dank besonders an die Neuanmeldungen, die gleich im Januar unsere Gemeinschaft verstärken !

Informix startet auch schon in den ersten Wochen voll durch. Bereits im Februar findet das Informix Bootcamp in Düsseldorf statt, weitere Veranstaltungen sind aktuell in der Vorbereitung, so dass Sie sich auf einige interessante Events freuen dürfen.

Wie immer haben wir für Sie eine Reihe an Tipps und Tricks zusammengestellt. Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam



## TechTipp: JDBC – Nutzung der Datei INFORMIXSQLHOSTS

Soll über JDBC auf einen Informix Server zugegriffen werden, so werden die Verbindungsinformationen im „Connection String“ angegeben.

Dieser hat üblicherweise die Form:

```
jdbc:informix-sqli://<host>:<port>/<database>:informixserver=<dbserver>
```

Dabei wird der Informix Treiber **com.informix.jdbc.IfxDriver** genutzt.

Damit die Verbindung zur Datenbank zustande kommt, müssen zudem die Umgebungsvariablen INFORMIXDIR und PATH korrekt gesetzt sein.

Dem „Connection String“ können mit Semikolon getrennt weitere Parameter mitgegeben werden wie z.B. die Spracheinstellung der Datenbank:

```
jdbc:informix-sqli://<host>:<port>/<database>:informixserver=<dbserver>;  
CLIENT_LOCALE=en_us.utf8;DB_LOCALE=en_us.utf8
```

Da bei der Verbindung ein konkreter Rechner (als Hostname oder IP) angegeben wird, kann die Verbindung nicht flexibel darauf reagieren, wenn z.B. in einer Clusterumgebung (HDR/RSS/ER) das System auf den Secondary Server geschwenkt wird.

Für Applikationen, die ihre Verbindungsinformationen aus der Datei sqlhosts beziehen, besteht die Option die Verbindung mittels Gruppeneintrag zu realisieren, und damit im Fall des Schwenks automatisch auf einen anderen Server zu verweisen.

Um diese Option bei der Verbindung mittels JDBC zu nutzen, muss die Datei sqlhosts statt des Hosts und des Ports als Parameter in den Verbindungsinformationen angegeben werden:

```
jdbc:informix-sqli:/<database>:informixserver=<dbserver>;SQLH_TYPE=FILE;  
SQLH_FILE=/opt/ifx/etc/sqlhosts
```

Als „dbserver“ wird die Gruppe angegeben, die dann in der Datei \$INFORMIXDIR/etc/sqlhosts auf die zur Verfügung stehende Instanz aufgelöst wird. Hierbei kann es sich direkt um den INFORMIXSERVER oder um ein SLAs eines Connection Managers handeln.

Beispiel für die Datei sqlhosts:

```
ifx_kalu    group      -           -       i=42  
cm1_kalu   onsoctcp    kalu42     9042    g=ifx_kalu  
cm2_kalu   onsoctcp    kalu43     9042    g=ifx_kalu
```

Im Beispiel würde die Gruppe „ifx\_kalu“ im „Connection String“ angegeben, die dann nach Verfügbarkeit auf cm1\_kalu bzw. cm2\_kalu aufgelöst wird.

## TechTipp: Scheduler Stoppen / Starten

Der Scheduler der Informix Instanz, der dafür sorgt dass Tasks ausgeführt werden, und dass Alarme verarbeitet werden, läuft ständig. Der Scheduler ist Bestandteil der Datenbank sysadmin, greift aber auch immer wieder auch auf die anderen Datenbanken der Instanz zu, um z.B. die Notwendigkeit der Aktualisierung der Statistiken zu prüfen und Kennzahlen über die Tabellen zu ermitteln (je nach Aktivierung der entsprechenden Task). Sollen diese Zugriffe unterbunden werden, um sicherzustellen, dass keine Zugriffe auf eine bestimmte Datenbank erfolgen, so kann der Scheduler mittels Funktionsaufruf gestoppt und auch wieder gestartet werden.

Beispiel:

```
DATABASE sysadmin;  
EXECUTE FUNCTION task("scheduler shutdown");
```

Soll der Scheduler wieder gestartet werden, kann dies über folgenden Aufruf erfolgen:

```
DATABASE sysadmin;  
EXECUTE FUNCTION task("scheduler start");
```

Beide Aufrufe müssen als User „informix“ erfolgen.

Hinweis: Bei gestopptem Scheduler lässt sich auch die Datenbank sysadmin exportieren.

## TechTipp: Global Security Kit (gskit)

Das IBM Global Security Kit (GSKit) installiert Mechanismen für die Verschlüsselung und die SSL Kommunikation zwischen Informix Datenbanken bzw. Tools.

Das Package ist unter \$INFORMIXDIR/gskit nach der Installation verfügbar.

Unter Linux werden zwei RPM-Packages mitgeliefert :

```
gskcrypt64-8.0.50.10.linux.x86_64.tar  
gskssl64-8.0.50.10.linux.x86_64.tar
```

Die Installation unter Linux erfolgt mit mit rpm (was z.B. auf Ubuntu eigens installiert werden muss) mittels:

```
./installgskit  
Performing GSKit installation for Linux ...
```

Anschliessend sind die beiden Pakete installiert:

```
rpm -qa | grep gsk  
gskssl64-8.0-50.10.x86_64  
gskcrypt64-8.0-50.10.x86_64
```

Die Installation des gskit ist die Voraussetzung um z.B. mit onpassword die Verbindungsinformationen für login und password für einen User zu hinterlegen.

## TechTipp: onpassword – Login und Password verschlüsselt speichern

Das Programm onpassword setzt voraus, dass das aktuelle gskit installiert wurde. Damit können die Login- und Password-Informationen zu Benutzern verschlüsselt abgelegt werden. Dies ist z.B. dann erforderlich, wenn ein Connection Manager im LDAP-Umfeld zum Einsatz kommt. Bei LDAP ist es notwendig, dass sich die Benutzer mit Login und Password am Server anmelden. Da der Connection Manager beim Start Abfragen auf allen beteiligten Instanzen absetzt, um den Status und Informationen zur Rolle im Cluster und zur Verfügbarkeit absetzt, benötigt dieser für den Connect den Benutzernamen und das Password. Diese werden mittels des Programms „onpassword“ verschlüsselt in der Datei \$INFORMIXDIR/etc/passwd\_file abgelegt.

### Usage: onpassword -k access\_key -[e|d] pw\_file

Die Informationen zu Login und Password je Datenbankinstanz werden dazu in einer Datei abgelegt, die nach dem Aufruf von „onpassword“ wieder gelöscht werden kann.

Die Passwortdatei muss folgenden Aufbau besitzen (Beispiel):

| <dbserver> | <dbserver/alias> | <user>   | <passwd>  |
|------------|------------------|----------|-----------|
| ifxserv1   | ifxswg           | informix | Passlfx42 |

Der „access\_key“ ist frei wählbar und dient lediglich zur Verschlüsselung bzw. zur Entschlüsselung, wenn die Informationen aus der Datei passwd\_file ausgelesen werden sollen.

Der Aufruf

### onpassword -k access\_key -e pw\_file

dient zur Verschlüsselung der Daten, der Aufruf mit dem Argument -d zeigt die Informationen an, die verschlüsselt abgelegt sind.

Hinweis:

Kann der Connection Manager nicht auf die Datenbankserver zugreifen, da kein password\_file erstellt wurde, so zeigt sich dies in der Logdatei der Connection Managers mit dem Fehler -1829 direkt nach dem Start beim Versuch die Serverinformationen auszulesen..

## TechTipp: `cdr check repl --repair repl / cdr sync repl`

Die Synchronisierung der Daten beim Aufsetzen der Enterprise Replikation kann über verschiedene Wege erfolgen. Ein Weg ist es, die Daten vom „Master“ mittels „dirty read“ in externe Tabellen zu entladen, und diese dann auf der Zielseite wieder einzuspielen. Anschliessend können die in der Zwischenzeit entstandenen Unterschiede mittels „`cdr check repl –repair ...`“ nachgezogen werden.

Soll der Datenbestand ohne Filetransfer übertragen werden, so gibt es die Möglichkeit entweder „`cdr check repl`“ mit der Option „`repair`“ zu nutzen, oder den „`cdr sync repl`“ einzusetzen.

Der grundsätzliche Unterschied zwischen den beiden ist:

'`cdr check repair`' gleicht (nur) die zu reparierenden Rows ab, und zwar weitgehend auf dem Wege herkömmlicher Dummy-Updates, die dann ganz regulär, allerdings über ein Shadow Replicate, repliziert werden. Hierzu vergleicht es die Tabellen zunächst Row für Row, per Primary Key (bzw. heutzutage auch ERKEY oder beliebigem Unique Index) und Row Checksums.

'`cdr sync`' dagegen überträgt den kompletten Master-Bestand an Rows an die Ziele, über die reguläre Replikation, jedoch an den logischen Logs vorbei direkt über die Send Queue, also Master-seitig ohne jeden Logging, Snooper und Grouper Aufwand. Dies geschieht ebenso in Primary Key (oder Alternativen) Ordnung, d.h. der gewählte Index wird, wie beim '`cdr check`', zunächst benutzt, um "extra target rows" zu identifizieren und je nach "`--extratargetrows`" Option zu behandeln (wird keine solche Option angegeben, gilt der Default "delete"). Um diesen Extra-Target-Rows Schritt zu vereinfachen, empfiehlt es sich evtl., die Tabelle target-seitig vorher zu leeren mittels „`truncate table`“.

Das '`cdr check repair`' ist daher eher für kleinteilige Reparaturen vorgesehen, entsprechend ist es noch mit zusätzlichen, begrenzenden Optionen "`--since`" und "`--where`" ausgestattet, um gezielt nur bestimmte Regionen einer Tabelle zu checken bzw. zu reparieren (während die Tabelle auf allen Knoten weiter voll zur Verfügung steht).

Demgegenüber ist der '`sync`' Mechanismus für komplette Neuübertragung und Initiierung eines Target-Datenbestandes vorgesehen und hierbei entsprechend performanter. Er wird entweder über '`cdr sync replicate[set]`' oder aber über die '`--syncdatasource`' Option beim Starten von Replikaten (oder Realisieren von Templates) aktiviert. Wegen des hohen Durchsatzes durch die ER NIF Verbindungen hängt er in besonderem Maß von ausreichendem Queue Memory, primäre auf Sender-, aber auch auf Empfängerseite, ab, weswegen er jeweils auch über eine '`--memadjust`' Option verfügt, um das Send Queue Memory temporär zu vergrößern,

Um sowohl 'cdr sync' als auch 'cdr check [repair]' zu parallelisieren, benutzt man an Stelle von 'cdr sync/check replicate' die 'cdr sync/check replset' Alternative, Besonders hervorzuheben ist dabei die Fähigkeit, RI Beziehungen (referenzielle Integrität) zwischen replizierten Tabellen, bei 'cdr check' bis hin zu zirkulären Zusammenhängen, zu berücksichtigen. Hierfür wären zusammengehörige Tabellen jeweils in einem Replikat Set zusammenzufassen. Dies stellt sicher, daß die Tabellen in der richtigen Reihenfolge, also Hierarchie-Ebene für Hierarchie-Ebene übertragen werden, wobei nur innerhalb einer Ebene parallelisiert wird. 'cdr realize template' würde, mit der '--syncdatasource' Option, implizit ein 'cdr sync replset' ausführen.

Wer Trigger benutzt, die target-seitig von der ER ausgelöst werden sollen, sollte sich genau überlegen, ob bzw. wie er dies auch bei einem 'cdr sync' oder 'cdr check repair' Lauf haben möchte. Per Default geschieht dies nicht, es kann aber per '--firetrigger' Option unterdrückt ("off"), erzwungen ("on") bzw. erwünscht werden ("follow", sprich "richte dich nach dem jeweiligen Replikat").

Prinzipiell ist der große Vorteil dieser Mechanismen gegenüber manuellem Laden, daß Applikationen und Replikation hier weiterlaufen können und Inkonsistenzen durch den Ladevorgang weitgehend ausgeschlossen sind: ein geänderter/neuer Datensatz liegt entweder vor der aktuellen Synchronisations-Position, war also vor der Änderung schon synchronisiert, oder er ist noch nicht synchronisiert, was u.U. zu zweiseitigen Fehlern führen kann, die aber durch die Synchronisierung bereinigt werden, sobald diese den Satz erreicht.

Performance-Vergleich bei 100'000 Rows:

```
cdr check repl -v -R -e delete -m test1_cdr -r test_cdr_rep test2_cdr
Jan 19 2014 15:07:33 ----- Table scan for test_cdr_rep start -----
Node           Rows      Extra      Missing  Mismatch  Processed
-----
test1_cdr      100000    0          0         0         100000
test2_cdr      0         0         100000    0         0
The repair operation completed. Validating the repaired rows ...
Validation completed successfully.
Jan 19 2014 15:10:59 ----- Table scan for test_cdr_rep end -----
```

```
cdr sync repl -m test1_cdr -r test_cdr_rep test2_cdr
Jan 19 2014 15:13:02 ----- Table scan for test_cdr_rep start -----
Starting synchronization scan for replicate
      test_cdr_rep
Ending synchronization scan for replicate
      test_cdr_rep
Jan 19 2014 15:13:22 ----- Table scan for test_cdr_rep end -----
```

Das bedeutet, dass der Aufruf von „cdr sync“ um nahezu Faktor 10 schneller war (20.989 Sekunden statt 3 Minuten, 25.435 Sekunden ).

## TechTipp: IWA 12.10 Feature: Enhanced Query Support

Mit der neuen Version des Informix Warehouse Accelerators wurde die Unterstützung von SQL-Abfragen um folgende Konstrukte erweitert:

- UNION und UNION ALL
- OLAP-Window Abfragen
- NULLS FIRST und NULLS LAST

Kurze Erläuterung zu den Erweiterungen:

### UNION und UNION ALL

Der UNION oder UNION ALL Mengenoperator vereint die Ergebnisse zweier SELECT-Abfragen. Kann eine der SELECT-Abfragen beschleunigt werden, so führt IWA diese aus, der Informix Server nimmt das Ergebnis entgegen, kombiniert es mit dem Ergebnis der anderen SELECT-Abfrage, und übergibt dann das Gesamtergebnis an den Client.

Beispiel 1:

```
SELECT COUNT(*) FROM sales UNION ALL SELECT COUNT(*) FROM sales_returns;
```

Beide SELECT-Abfragen können beschleunigt werden und werden von IWA nacheinander ausgeführt.

Beispiel 2:

```
SELECT COUNT(*) FROM sales UNION ALL SELECT COUNT(*) FROM systables;
```

Der erste SELECT-Abfrageblock auf die Tabelle "sales" kann beschleunigt werden und wird von IWA ausgeführt. Der zweite SELECT-Abfrageblock kann nicht beschleunigt werden und wird daher vom Informix Server ausgeführt.

### OLAP-Window Abfragen

Sogenannte OLAP-Window Funktionen werden für Online Analytical Processing (OLAP) benutzt, um Numerierung, Rangfolge und Aggregation auf Teilbereiche eines Abfrageergebnisses anzuwenden. Zur Beschleunigung solcher Abfragen kann die zugrundeliegende SELECT-Abfrage von IWA ausgeführt werden. Der Informix Server nimmt das Ergebnis von IWA entgegen, wendet darauf die OLAP-Window Funktionen an und gibt dann das Endergebnis an den Client.

Beispiel:

```
SELECT e.emp_name,  
       RANK() OVER (PARTITION BY region ORDER BY total_sales desc),  
       AVG(sales) OVER (PARTITION BY region, year)  
FROM employee e, sales s  
WHERE e.emp_id = s.emp_id;
```

IWA führt die Abfrage der Daten aus den Tabellen "employee" und "sales" mit dem zugehörigen Join durch. Auf dieses Ergebnis wendet der Informix Server die OLAP-Window Funktionen RANK und AVG an.

## **NULLS FIRST und NULLS LAST**

Wird in einer SQL-Abfrage das Ergebnis mittels ORDER BY sortiert, dann kann mit NULLS FIRST oder NULLS LAST spezifiziert werden, wie NULL-Werte in der Sortierung behandelt werden sollen. Frühere Versionen von Informix betrachteten NULL immer als den niedrigsten Wert und sortierten entsprechend. IWA unterstützt NULLS FIRST und NULLS LAST genauso wie der Informix Server.

## **TechTipp: First steps mit NoSQL – Gastbeitrag der CURSOR AG**

Einrichten der NoSQL Funktionalität auf einer bestehenden Instanz

Mit der Version 12.10.xC2 hat IBM die Möglichkeit geschaffen, mit MongoDB-Clients auf die Informix Datenbank zuzugreifen und die Informix NoSQL Funktionalität zu nutzen.

Leider bekommt man diese fantastische Feature nur, wenn man eine neue Instanz aufbaut. Alle Beschreibungen der IBM besagen, das man den Installer der xC2 Version eine neue Instanz anlegen lassen muss. Dann kann die Funktionalität, basierend auf dem Kopieren und Anpassen von 2 Dateien auf weitere neue Instanzen übernommen werden, die manuell erstellt werden.



[How do I install and configure NoSQL?](#)

[How do I configure an additional instance of IDS with NoSQL support?](#)

Leider gibt es keine Beschreibung, wie man auf einer bestehenden Instanz die Funktionalität einschalten kann.

Technisch ist die NoSQL Funktionalität als UDR gelöst. Die notwendigen Funktionen, Casts usw. sind im bootstrap-Script der Version 12.10.xC2 enthalten. Der Listener für MongoDB-Clients steckt nicht im oninit, sondern ist als extra Java-Service implementiert.

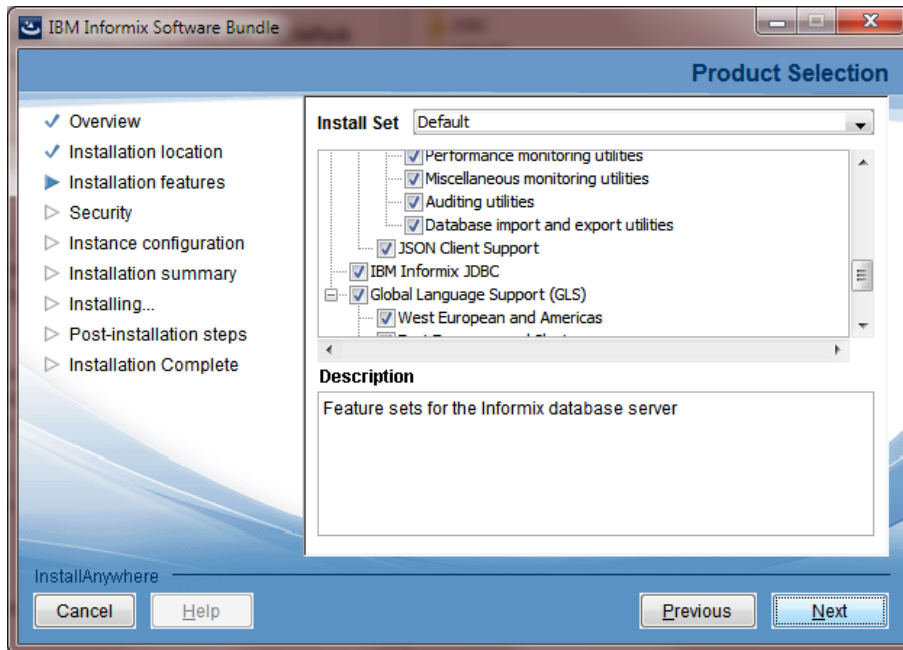
Gestartet wird dieser Listener vom Informix internen Scheduler, der das Vorhandensein der Datei \$INFORMIXDIR/etc/jsonListener.properties prüft und den Listener startet, wenn die Datei existiert.



Der Name der Datei jsonListener.properties wird während der Initialisierung der Instanz im Start-Task hinterlegt. Dadurch können auch mehrerer Instanzen innerhalb einer Installation die Json-Funktionalität nutzen.

Dieser Artikel ist als Anleitung gedacht, wie man in einer bestehenden Instanz die NoSQL Funktionalität einschaltet.

Dann erfolgt ein Update auf die entsprechende 12.10.xC2 Version. Dabei gibt man den Pfad seiner vorherigen Installation an und achtet darauf, das die Option "Json Client Support" an gehakt ist.



Nachdem die Installation durchgelaufen ist, braucht man noch einige Dateien. Ein Paket Informix-NoSQL-Update-Dateien der notwendigen Dateien steht für Sie zum Download bereit (siehe \*1).

Alle folgenden Aktionen müssen als Benutzer informix ausgeführt werden!

Als erstes muss man die UDR Funktionen erstellen. Dazu startet man eine Shell mit einer Informix-Umgebung. Geht in das Verzeichnis \$INFORMIXDIR/etc und führt folgendes Kommando aus

```
dbaccess sysadmin boot1210.sql
```

Das Script sollte fehlerfrei durchlaufen.

Jetzt benötigt man die Datei \$INFORMIXDIR/etc/jsonListener.properties Diese ist im Paket enthalten, muss aber entsprechend editiert werden.

Beispiel für Linux:

```
listener.port=27017
url=jdbc:informix-sqli:/<hostname/IP des DB-Servers>:<Port des Informix SQLI
Listeners>/sysmaster:INFORMIXSERVER=<INFORMIXSERVER-Name>;USER=<Username
Default: ifxjson>;PASSWORD=<Passwort>
```

Die komplette Dokumentation der jsonListener.properties Datei ist unter (siehe \*2) zu finden.

Nachdem diese Datei korrekt erstellt wurde, muss noch der Scheduler-Eintrag erzeugt werden, der den Service startet und auf den Unix-Plattformen ein spezieller User für den Listener erstellt werden.

Dazu muss der Benutzer daemon in den surrogates cache eingetragen werden. Es muss eine Datei "allowed.surrogates" in /etc/informix angelegt werden. Die Rechte müssen sein:

```
-rw-r--r-- 1 root root 14 Oct 31 13:35 allowed.surrogates
```

In dieser Datei muss mindestens stehen:

```
USERS: daemon
```

Sie können prüfen, ob diese Datei korrekt angelegt ist, indem Sie:

```
onmode -cache surrogates
```

aufrufen. Dieser Befehl muss fehlerfrei durchlaufen. Gibt es Fehler, so steht im online.log (onstat -m) die Ursache.

Jetzt muss je nach Plattform noch die im Pack enthaltenen Datei sch\_init\_<plattform>.sql ausgeführt werden.

Besonderheit bei Unix/Linux ist, dass hier der User ifxjson angelegt wird. Das dazugehörige Passwort 'extremgeheim' muss in der Datei natürlich durch ein sinnvolles Passwort ersetzt werden, das mit dem in der jsonListener.properties übereinstimmen muss.

Sollen mehrere Instanzen einer Installation den Listener starten, so ist in dieser Datei, in der Zeile:

```
LET json_listener = 'jsonListener.properties';
```

bei jeder Instanz ein anderer Dateiname anzugeben. Dementsprechend müssen diese Dateien dann auch angelegt und mit den richtigen Werten versehen werden.

```
dbaccess sysadmin sch_init_<Plattform>.sql
```

Jetzt kann man den Server neu starten. Sollte der Listener nicht starten, sollten sie nochmal überprüfen, ob der Task "json listener" in der sysadmin.ph\_task auf enabled ='t' steht.

Ist der Listener gestartet, kann man mit MongoDB-Clients auf die NoSQL-Funktionalität der Informix Datenbank zugreifen. Test-Client kann z.B. Robomongo sein.

Dieser Artikel wurde uns freundlicherweise von **Andreas Seifert**, Mitarbeiter der **CURSOR Software AG ([www.cursor-distribution.de](http://www.cursor-distribution.de))** zur Verfügung gestellt. Unter dem Folgendem Link ist dieser Artikel im Original zu finden und enthält zusätzliche Hinweise für ein Troubelshooting:

<http://www.cursor-distribution.de/de/ibm-software/informix-software/support-informix/support-fuer-techniker-informix/470-einrichten-der-nosql-funktionalitaet-auf-einer-bestehenden-instanz>

(Anmerkung \*1) [www.cursor-distribution.de/de/community/community-db2-aktuell/db2-newsletter/db2-newsletter-ibm/doc\\_download/1513-informix-nosql-update-dateien](http://www.cursor-distribution.de/de/community/community-db2-aktuell/db2-newsletter/db2-newsletter-ibm/doc_download/1513-informix-nosql-update-dateien)

(Anmerkung \*2) [http://pic.dhe.ibm.com/infocenter/informix/v121/index.jsp?topic=%2Fcom.ibm.json.doc%2Fids\\_json\\_006.htm](http://pic.dhe.ibm.com/infocenter/informix/v121/index.jsp?topic=%2Fcom.ibm.json.doc%2Fids_json_006.htm)

## WEBTIPP: Lizenzinformationen - Firma OMTCO

Auf den Internetseiten der Firma OMTCO sind eine Reihe an interessanten Dokumentationen rund um das Thema Lizenzierung, Metrik und Audit zu finden, die teilweise zum download bereit stehen.

Beispiel hierfür sind:

- IBM Informix – Product Editions, Metrics And Licensing Restrictions (Released October 2013)
- Top 12 IBM Licensing Metrics And Licensing Restrictions (Released March 2013)
- Top 100 IBM SAM Buzzwords – A Glossary Of Selected IBM Terms From A(udit) To Z(series) (Released June 2012, Updated February 2013)
- Top 100 IBM Acronyms In Licensing – From AUSI To XUVU (Released July 2012, Updated October 2012/February 2013)

Der Link zu den Dokumentationen ist: <http://omtco.eu/references/ibm/>

## Versionsinfo: 11.70.xC8 ist verfügbar

Seit einigen Tagen ist die Version 11.70.xC8 für alle unterstützten Plattformen und Editionen verfügbar. Da es in jeder Version eine Reihe an Verbesserungen gibt, sollte immer eine der neueren Versionen eingesetzt werden.

## 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.org/intl/deu>

[http://www.iug.de/index.php?option=com\\_content&task=view&id=95&Itemid=149](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.cursor-distribution.de/de/community/community-informix/informix-newsletter>

<http://www.listec.de/Newsletter/IBM-Informix-Newsletter/View-category.html>

<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](mailto:gerd.kaluzinski@de.ibm.com) +49-175-228-1983

Martin Fuerderer IBM Informix Entwicklung, München  
IBM Software Group, Information Management  
[martinfu@de.ibm.com](mailto:martinfu@de.ibm.com)

Andreas Legner IBM Informix Advanced Support  
IBM Software Group, Information Management  
[andreas.legner@de.ibm.com](mailto:andreas.legner@de.ibm.com)

Gastbeitrag:

Andreas Seifert Mitarbeiter der CURSOR Software AG  
([www.cursor-distribution.de](http://www.cursor-distribution.de))

Die Versionsinfo stammt aus dem Versions-Newsletter der CURSOR Software AG  
<http://www.cursor-distribution.de/download/informix-vinfo>

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Gerd Kaluzinski

(Winter am Bodensee)