

## Willkommen zum „IBM Informix Newsletter“

### Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Informix und NoSQL (8): Abfragen von NoSQL-Daten beschleunigen.....	2
TechTipp: TimeSeries – Pattern Search.....	5
TechTipp: DRDA Verbindungen über den ConnectionManager.....	8
TechDay und IUG in München: Eine Nachlese zu den Events.....	11
WebTipp: System Requirements für INFORMIX nach Plattform .....	11
Versionsinfo: 12.10.xC5W1 ist verfügbar.....	12
Anmeldung / Abmeldung / Anmerkung.....	12
Die Autoren dieser Ausgabe.....	13

### Aktuelles

Liebe Leserinnen und Leser,

die Veranstaltung INFORMIX TechDay in München zeigte, dass das Interesse an den neuen Themen bei INFORMIX sehr gross ist. Trotz Bahnstreik kamen knapp 70 Teilnehmer und dabei waren viele neue und junge Gesichter zu sehen. Die Veranstaltung selbst war auch aussergewöhnlich, denn so viel Kompetenz aus den Bereichen Labor, Support und Services trifft man nur selten an einem Fleck (\*). Der anschliessende Workshop der IUG war die perfekte Ergänzung, so dass es sich gelohnt hat beide Tage teilzunehmen.



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

Ihr TechTeam

(\*) ... ausser man geht im Sommer im Münchener Osten abends in einen Biergarten nahe der IBM Niederlassung, wo man sich mit vielen technischen Fragen dann aber wenig Freunde macht.

## TechTipp: Informix und NoSQL (8): Abfragen von NoSQL-Daten beschleunigen

[ Alle Beispiele wurden mit Informix Version 12.10.xC5 (auf Linux x86 64-bit) erprobt. ]

Das Mittel der Wahl zur Beschleunigung von Abfragen ist bei Informix natürlich der Informix Warehouse Accelerator (IWA). Nun ist IWA allerdings nicht in der Lage, die intern mit dem BSON-Datentyp abgespeicherten NoSQL-Daten direkt zu interpretieren und zu verarbeiten. Um trotzdem eine gewünschte Beschleunigung zu erreichen, nutzen wir die von IWA gebotene Möglichkeit, Views zu verarbeiten.

Für einen ersten Versuch können wir eigentlich den View "rating\_v" verwenden, den wir auf die NoSQL-Collection "rating" schon angelegt haben (siehe "SQL-Zugriff auf NoSQL-Daten" im Newsletter November 2014). Bei der Definition des Data Marts für IWA müssen wir bei der Angabe der Tabelle (bzw. des Views) auch das Schema, bzw. den Besitzer, angeben, um die Tabelle (den View) eindeutig zu identifizieren. Damit wir immer einen konsistenten Schemanamen haben, der sich mit der Angabe in der Data Mart Definition deckt, legen wir den View neu an und geben dabei den Schemanamen "dwa" mit an:

```
DROP VIEW rating_v;
CREATE VIEW dwa.rating_v (catalog_num, rating_value, comment) AS
SELECT
    data.catalog_num::INT,
    data.rating_value::INT,
    data.comment::LVARCHAR
FROM rating;
```

Nun erstellen wir die Data Mart Definition als XML-Datei. In unserem ersten Versuch definieren wir den Data Mart so einfach wie möglich - er enthält nur den View:

```
<?xml version="1.0" encoding="UTF-8" ?>
<aqtmart:martModel
  xmlns:aqtmart="http://www.ibm.com/xmlns/prod/dwa"
  version="1.0">
  <mart name="rating_mart">
    <table name="rating_v" schema="dwa">
      <column name="catalog_num" />
      <column name="rating_value" />
      <column name="comment" />
    </table>
  </mart>
</aqtmart:martModel>
```

Mit dieser Data Mart Definition erzeugen wir nun den Data Mart in IWA. (Dies setzt natürlich voraus, dass IWA konfiguriert und gestartet ist, und dass der Informix Server zu IWA verbunden ist, d.h. dass in IWA ein sogenannter 'accelerator' für den Informix Server erzeugt wurde. Eine Beschreibung dieser Schritte sprengt den Rahmen des Newsletters, daher sei an dieser Stelle auf das Handbuch zu IWA verwiesen.) Der Data Mart wird in zwei Schritten angelegt: zuerst wird er mit der Definition als leeres Gerüst erzeugt, und dann mit den Daten aus der Datenbank geladen. Falls die Schritte später wiederholt werden sollen, dann muss der schon existierende Data Mart zuerst entfernt werden. In unseren Beispiel verwenden wir die entsprechenden Stored Procedures folgendermassen:

```
-- execute function ifx_dropMart('dwa1', 'rating_mart');
execute function ifx_createMart('dwa1',
    filetoclob('/tmp/crea_mart01.xml', 'client'));
execute function ifx_loadMart('dwa1', 'rating_mart', 'NONE');
```

Hierbei ist "dwa1" der Name des 'accelerator', zu dem der Informix Server eine Verbindung hat. Die obige XML-Definition für den Data Mart wurde in der Datei "/tmp/crea\_mart01.xml" gespeichert. "rating\_mart" ist der Name des Data Mart, wie wir ihn auch in der XML-Definition spezifiziert haben. "NONE" bedeutet, dass für den Ladevorgang keine Sperrungen gesetzt werden sollen - was auf einem View sowieso nicht möglich wäre.

Bei Aufruf der Stored Procedures mittels "dbaccess" sollten wir für die erfolgreiche Durchführung jeweils eine Meldung folgender Form erhalten:

```
(expression)  The operation was completed successfully.
1 row(s) retrieved.
```

Damit können wir nun eine Abfrage auf die NoSQL-Daten beschleunigen. Die Abfrage muss hierzu nur den View verwenden, auf dessen Basis wir den Data Mart erzeugt haben. Zusätzlich muss die Abfrage spezifizieren, dass sie beschleunigt werden will. Dazu setzen wir die session environment variable "use\_dwa" entsprechend:

```
SET ENVIRONMENT use_dwa 'accelerate on';
SELECT MAX(rating_value) FROM rating_v;
```

Bei der vorhandenen Datenmenge in der Collection "rating" werden wir allerdings kaum eine wirkliche Beschleunigung feststellen können. Dennoch können wir kontrollieren, ob die Abfrage mithilfe von IWA ausgeführt wird.

Hierzu setzen wir zusätzlich die Option "debug on" der session environment variable "use\_dwa":

```
SET ENVIRONMENT use_dwa 'accelerate on';  
SET ENVIRONMENT use_dwa 'debug on';  
SELECT MAX(rating_value) FROM rating_v;
```

Nach Ausführung der Abfrage können wir nun im Messagelog des Informix Server's nachsehen. Eine erfolgreiche Beschleunigung wird wie folgt protokolliert:

```
14:15:36 SQDWA: select max(rating_value) from rating_v  
14:15:36 SQDWA: Identified 1 candidate AQTs for matching  
14:15:36 SQDWA: matched: aqt11fc1f87-0d62-4523-ba92-  
0868fd2ea506  
14:15:36 SQDWA: matching successful (0 msec) aqt11fc1f87-0d62-  
4523-ba92-0868fd2ea506  
14:15:43 SQDWA: offloading successful (6535 msec)
```

Bisher haben wir nur eine SQL-Abfrage der NoSQL-Daten zum beschleunigten Ablauf gebracht, indem wir den Informix SQL-Befehl "SET ENVIRONMENT use\_dwa ..." der Abfrage direkt vorangestellt haben. Mit einer NoSQL-Abfrage, z.B. über das REST-API, ist dies aber nicht wirklich realistisch. In so einem Fall, oder allgemein um die Verwendung von "SET ENVIRONMENT use\_dwa ..." in einer Applikation zu vermeiden, können wir die gewünschten Einstellungen von "use\_dwa" auch in der Stored Procedure "sysdbopen()" vornehmen.

Dazu genügt es, die entsprechenden Befehle in der Stored Procedure "sysdbopen()" auszuführen - "sysdbopen()" selbst wird beim Verbindungsaufbau zur Datenbank automatisch aufgerufen. Wenn "sysdbopen()" noch nicht existiert, so kann der Administrator die Stored Procedure entsprechend anlegen. Im Bedarfsfall kann auch ein benutzerspezifisches "sysdbopen()" angelegt werden, so dass für unterschiedliche Nutzer verschiedene Einstellungen vorgenommen werden können. Mehr Information zu "sysdbopen()" finden Sie auch im Newsletter Oktober 2011.

Mit dem Mechanismus von "sysdbopen()" können wir Einstellungen von "use\_dwa" vornehmen, ohne dass eine Applikation den Mechanismus kennen und direkt benutzen muss. Dies funktioniert auch für Abfragen mittels NoSQL, z.B. über das REST-API.

## TechTipp: TimeSeries – Pattern Search

Mit der Version 12.10.xC5 kam die Möglichkeit hinzu, in Zeitreihen nach einem bestimmten Muster zu suchen. Unterstützt werden aktuell die numerischen Datentypen.

Der Sinn dieser Suche kann zum Beispiel darin liegen, dass sich aussergewöhnliche Ereignisse, wie zum Beispiel Ausfälle von Produktionsmaschinen, durch signifikante Messwerte von Sensoren einige Zeit vor dem Ausfall ankündigen. Damit könnte vor der dem eigentlichen Problem die Situation erkannt werden, um rechtzeitig entsprechende Massnahmen einzuleiten.

Das folgende Beispiel zeigt, wie die Suche nach einem Muster erfolgen kann und dass nicht nur exakte Treffer, sondern auch Treffer mit leichten Abweichungen gefunden werden können:

Zuerst erstellen wir eine Datenbank mit einer Tabelle für Zeitreihen:

```
create database tspattern in datadbs with log;
create row type kalu_row(
    tstamp datetime year to fraction(5),
    value1 real
);
execute procedure tscontainercreate(
    "kalu_cont", "datadbs", "kalu_row", "1024", "10240"
);
create table tsdata(
    id int primary key,
    ts1 timeseries(kalu_row)
);
```

Dann füllen wir zum Test einige Daten ein (z.B. was „sensors“ über die Temperatur des Prozessors liefert):

```
insert into tsdata values(1, "origin(2015-05-02 00:00:00.00000),
calendar(ts_1hour), container(kalu_cont), threshold(0), regular,
[(49), (51), (54), (59), (55), (54), (55), (54), (54), (55), (54), (54), (54),
(62), (75), (88), (92), (91), (95), (83), (74), (56), (54), (54), (53), (53),
(53), (55), (55), (56), (54), (54), (66), (75), (77), (54), (54), (54), (62),
(73), (88), (93), (90), (94), (83), (73), (56), (54), (54)]");
```

Zur Kontrolle erstellen wir noch eine virtuelle Tabelle, mit der wird die Einträge der Daten in gewohnter Form sehen können:

```
execute procedure TSCreateVirtualTab('tsdata_vt', 'tsdata',
'calendar(ts_1hour), origin(2015-05-01 00:00:00.00000)');
```

Damit lassen sich die Werte über SQL abfragen:

```
select *
from tsdata_vt
order by tstamp;
```

Ergebnis:

id	tstamp	value1
1	2015-05-02 00:00:00.00000	49.00000000000
1	2015-05-02 01:00:00.00000	51.00000000000
1	2015-05-02 02:00:00.00000	54.00000000000
1	2015-05-02 03:00:00.00000	59.00000000000
1	2015-05-02 04:00:00.00000	55.00000000000
1	2015-05-02 05:00:00.00000	54.00000000000
1	2015-05-02 06:00:00.00000	55.00000000000
1	2015-05-02 07:00:00.00000	54.00000000000
1	2015-05-02 08:00:00.00000	54.00000000000
1	2015-05-02 09:00:00.00000	55.00000000000
1	2015-05-02 10:00:00.00000	54.00000000000
1	2015-05-02 11:00:00.00000	54.00000000000
1	2015-05-02 12:00:00.00000	54.00000000000
1	2015-05-02 13:00:00.00000	62.00000000000
1	2015-05-02 14:00:00.00000	75.00000000000
1	2015-05-02 15:00:00.00000	88.00000000000
1	2015-05-02 16:00:00.00000	92.00000000000
	...	

Nun haben wir die Datenbasis, um mittels „Pattern Search“ nach ungewöhnlichen Ereignissen zu suchen, wie z.B. einem Temperaturanstieg mit steigenden Werten im Bereich 65->75->85 Grad.

Die Abfrage sieht dabei so aus:

```
select TSPatternMatch(
    ts1, 'tsdata', id, 'value1',
    '2015-05-02 00:00:00.00000',
    '2015-05-07 00:00:00.00000',
    'LIST{65,75,85}'::LIST(DOUBLE PRECISION NOT NULL), 0, 0.5, 5)
FROM tsdata
WHERE id = 1;
```

Die Prozedur TSPatternMatch hat dabei folgende Syntax:

```
TSPatternMatch(
    ts      TimeSeries,          # Der Name der TimeSeries
    table_name  LVARCHAR,      # Der Name der Tabelle
    primary_key  LVARCHAR,    # Der PrimaryKey der Suchtabelle
    field_name  LVARCHAR,      # Das Suchfeld der Tabelle
    begin_tstamp DATETIME YEAR TO FRACTION(5), # Startzeitpunkt der Suche
    end_tstamp  DATETIME YEAR TO FRACTION(5), # Endzeitpunkt der Suche
    pattern     LIST(DOUBLE PRECISION NOT NULL), # Suchmuster
    search_method INTEGER,     # Suchmethode
                                # 0 – Teilstringsuche
                                # 1 – gesamter Suchstring
    similarity  DOUBLE PRECISION, # Prozent der Übereinstimmung 0.0 – 1.0
                                # Default 0.8 (für 80%)
    unit_error  DOUBLE PRECISION # Erlaubte Abweichung (Default ist 0.1)
                                # Wir haben im Beispiel 5 erlaubt
)
returns LIST(SEARCHROW NOT NULL)
```

Das Ergebnis der Abfrage mit Pattern Match zeigt, dass der gesuchte Anstieg am 02.05. zwischen 13:00 und 15:00 und am 03.05. zwischen 14:00 und 16:00 aufgetreten ist. Zudem gibt es eine Struktur mit 66,7% Übereinstimmung am 03.05. zwischen 08:00 und 10:00:

```
(expression) LIST{
ROW('1', '2015-05-02 13:00:00.00000',
      '2015-05-02 15:00:00.00000', 1.000000000000),
ROW('1', '2015-05-03 08:00:00.00000',
      '2015-05- 03 10:00:00.00000', 0.666666666667),
ROW('1', '2015-05-03 16:00:00.0 0000',
      '2015-05-03 18:00:00.00000', 1.000000000000)}
```

Über die Werte „similarity“ und „unit\_error“ lässt sich justieren, wie sehr die Werte der Zweitreihe mit den „SuchPattern“ übereinstimmen müssen, um als „Treffer“ gewertet zu werden.

## TechTipp: DRDA Verbindungen über den ConnectionManager

Einige Verbindungen zur Datenbank bedingen eine DRDA-Verbindung. Ein IBM internes Beispiel hierfür ist Cognos, das für das Cognos Repository, wenn dies als Informix Datenbank angelegt ist, eine Verbindung über das DRDA Protokoll vorgibt. Der Trick, der hierbei zu beachten ist, ist ganz einfach: Es muss zu jeder DRDA-Verbindung zusätzlich eine Verbindung über Sockets zum Server existieren, über die der ConnectionManager mit dem Server kommuniziert und über die der Status abgeglichen werden kann.

Beispiel:

cmsm.cfg.cog:

```
#####
# Global configuration parameters
#####

NAME          cm_drcog1
LOGFILE       /home/informix/LOG/cm_drcog.log

#####
# Settings for a high-availability cluster
#####

CLUSTER drcog1
{
  INFORMIXSERVER      cog1,dr_cog1,cog2,dr_cog2
  SLA      cog_cognos      DBSERVERS=primary,HDR
  SLA      drcog_cognos    DBSERVERS=primary,HDR

  FOC      ORDER=DISABLED PRIORITY=3 TIMEOUT=10 RETRY=2
  LOG      1
}
#####
```

Die rot markierten Namen sind stehen für die Verbindung über DRDA, die Namen in schwarzer Schrift stehen für die Socket-Verbindung zu den jeweiligen Servern. Wichtig hierbei ist die Reihenfolge. Die Socket-Verbindung sollte immer vor der DRDA-Verbindung stehen um unnötige Fehler zu vermeiden.

Im Log des ConnectionManagers sind folgende Informationen zu finden:

```
09:16:30 IBM Informix Connection Manager
09:16:30 IBM Informix CSDK Version 4.10, IBM Informix-ESQL Version 4.10.FC2
09:16:30 Build Number:    N217
09:16:30 Build Host:      lxvm-l172
09:16:30 Build OS:   Linux 2.6.18-128.el5
09:16:30 Build Date:     Thu Aug 29 08:19:06 CDT 2013
09:16:30 GLS Version:   glslib-6.00.FC2
09:16:30
09:16:30 SLA drcog_cognos listener mode REDIRECT
09:16:30 SLA cog_cognos listener mode REDIRECT
09:16:30 Connection Manager name is cm_drcog1
09:16:30 Starting Connection Manager...
09:16:30 Warning: Password Manager failed; working in trusted node mode
09:16:30 dbservername = cog1
09:16:30 nettype = onsoctcp
09:16:30 hostname = 192.168.23.42
09:16:30 servicename = 9091
09:16:30 options =
09:16:30 dbservername = dr_cog1
09:16:30 nettype = drsoctcp
09:16:30 hostname = 192.128.23.42
09:16:30 servicename = 9089
09:16:30 options =
09:16:30 dbservername = cog2
09:16:30 nettype = onsoctcp
09:16:30 hostname = 192.128.23.44
09:16:30 servicename = 9091
09:16:30 options =
09:16:30 dbservername = dr_cog2
09:16:30 nettype = drsoctcp
09:16:30 hostname = 192.128.23.44
09:16:30 servicename = 9089
09:16:30 options =
09:16:30 listener cog_cognos initializing
09:16:30 listener drcog_cognos initializing
09:16:30 Listener drcog_cognos DBSERVERS=primary,HDR is active with 4 worker
threads
09:16:30 Listener cog_cognos DBSERVERS=primary,HDR is active with 4 worker threads
09:16:30 Connection Manager successfully connected to cog1
09:16:30 The server type of cluster drcog1 server cog1 is Primary.
```

09:16:30 Cluster drcog1 Arbitrator FOC ORDER=DISABLED PRIORITY=3 TIMEOUT=10  
 RETRY=2  
 09:16:30 The server type of cluster drcog1 server cog2 is HDR.  
 09:16:30 Connection Manager successfully connected to cog2  
 09:16:53 Connection Manager started successfully  
 09:17:31 SLA drcog\_cognos redirect DRDA client to dr\_cog1 192.128.23.42:9089  
 ...

Im „onstat -g cmsm“ am Server ist dann zu sehen, wie viele Verbindungen der  
 ConnectionManager bereits bearbeitet hat:

Unified Connection Manager: cm\_drcog1  
 Hostname: kalutest1.lindau

CLUSTER	drcog1	LOCAL	Informix Servers: cog1,dr_cog1,cog2,dr_cog2
Rule	SLA	Connections	Service/Protocol
	cog_cognos	0	9289/onsoctcp
DBSERVERS=primary,HDR	drcog_cognos	97054	9290/drsoctcp
DBSERVERS=primary,HDR			

Failover Arbitrator: Failover is disabled  
 ORDER=HDR TIMEOUT=10 RETRY=2 PRIORITY=3

Unified Connection Manager: cm\_drcog2  
 Hostname: kalutest2.lindau

CLUSTER	drcog2	LOCAL	Informix Servers: cog1,dr_cog1,cog2,dr_cog2
Rule	SLA	Connections	Service/Protocol
	cog_cognos	0	9289/onsoctcp
DBSERVERS=primary,HDR	drcog_cognos	146088	9290/drsoctcp
DBSERVERS=primary,HDR			

Failover Arbitrator: Failover is disabled  
 ORDER=HDR TIMEOUT=10 RETRY=2 PRIORITY=4

## TechDay und IUG in München: Eine Nachlese zu den Events

Der INFORMIX TechDay am 5. Mai erwies sich als Besuchermagnet. Mit knapp 70 Teilnehmern wurden unsere Erwartungen weit übertroffen. Besonders erfreulich war, dass sich auch viele neue und junge Gesichter unter die Reihen der „alten Informixer“ gemischt hatten. Das grosse Aufgebot an Sprechern aus den Bereichen Support, Lab und Services konnte in dieser Form nur in München geboten werden, an der Heimatbasis dieser Instanzen. Dass auch bei der abschliessenden Fragerunde ab 17:00 noch alle Stühle belegt waren, zeigt das grosse Interesse an den gebotenen Informationen. Abgerundet wurde der Tag durch eine aussergewöhnliche Leistung der Küche der IBM Lokation, die die Mittagspause zum Genuss werden liess.

Vielen Dank an dieser Stelle an alle Besucher und Mitwirkenden für diesen rundum gelungenen Tag.

Der IUG Workshop am Folgetag bot vom Thema her eine perfekte Ergänzung des TechDays. Schwerpunkt waren diesmal Demos aus der Praxis. Beeindruckend war die Livevorführung von Alexander Körner, der ausgehend von einem Multisensor über einen RaspberryPi2 die gesamte Kette von der Erfassung der Messwerte, die lokale Zwischenspeicherung bis hin zur Übertragung in die Cloud, sowie die graphische Aufbereitung der Daten veranschaulichte. Auch die weiteren Themen drehten sich um Cloud, NoSQL und die Zukunft von INFORMIX.

Da der zweite Tag nur für IUG Mitglieder offen war, reduzierte sich die Teilnehmerzahl leider stark gegenüber dem ersten Tag. Wer beide Tage dabei war, der bekam durch die Kombination ein abgerundetes Bild, wohin sich INFORMIX entwickelt und welche Stärken neue Kunden und Einsatzgebiete erschliessen.

## WebTipp: System Requirements für INFORMIX nach Plattform

Immer wieder erreichen uns Anfragen zu den Systemvoraussetzungen für INFORMIX auf den unterschiedlichen Betriebssystemen und Windows.

Die Folgenden Links bieten eine gute Übersicht darüber, was ggf. vor der Installation von Informix vorbereitet werden muss. Für Linux sei noch ergänzt, dass der „bc“ und die „ksh“ vorausgesetzt werden, so dass diese auf jeden Fall vor INFORMIX installiert werden sollten.

### Die Matrix zum Server:

<http://www.ibm.com/support/docview.wss?uid=swg27013343>

Auch für **4gl** gibt es eine ähnliche Seite:

<http://www.ibm.com/support/docview.wss?uid=swg27018046>

## Versionsinfo: 12.10.xC5W1 ist verfügbar

Seit einigen Tagen ist die Version 12.10.xC5W1 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.

Die Liste der mit W1 behobenen Probleme finden Sie hier:

<http://www.ibm.com/support/docview.wss?uid=swg27045761>

Auf der selben Seite finden Sie auch die in der Version 12.10.xC5 behobenen Probleme früherer Versionen.

## 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](mailto: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.iiug.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/index.php/aktuelles/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)

Markus Holzbauer IBM Informix Advanced Support  
IBM Software Group, Information Management Support  
[holzbauer@de.ibm.com](mailto:holzbauer@de.ibm.com)

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

(Köchlinweiher Lindau)