

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....1
 TechTipp: ONCONFIG - DBSERVERALIAS.....2
 TechTipp: ONCONFIG - HA_ALIAS.....3
 TechTipp: ONCONFIG - DRINTERVAL.....4
 TechTipp: ONCONFIG - HDR_TXN_SCOPE.....4
 TechTipp: ONCONFIG - TEMPTAB_NOLOG.....5
 TechTipp: Optionen des ONSTAT (onstat -g ipl).....5
 TechTipp: Optionen des ONSTAT (onstat -g dri xxx).....6
 TechTipp: Time-Moving-Objects (STS_TrajectoryCross) Teil 8.....9
 TechTipp: Time-Moving-Objects (STS_TrajectoryIntersect) Teil 9.....10
 TechTipp: Time-Moving-Objects (STS_GetIntersectSet) Teil 10.....10
 TechTipp: IBMCHFWATSON – Kreatives Kochen.....12
 Termin: CDC Bootcamp in Ehningen von 26. bis 29. Januar.....13
 Anmeldung / Abmeldung / Anmerkung.....14
 Die Autoren dieser Ausgabe.....14

Aktuelles

Liebe Leserinnen und Leser,

die ersten Weihnachtsmärkte haben geöffnet, der erste Schnee ist gefallen und die Redaktion wurde mit Nüssen versorgt - Danke !

Ein Schwerpunkt dieser Ausgabe behandelt Randthemen der Replikation (HDR/RSS/SDS) und auch die Serie zu Time-Moving-Objects geht weiter.

Als Anregung für die Freizeit möchten wir Sie auf IBM Watson aufmerksam machen, der in immer mehr Bereichen eingesetzt werden kann.

Die Redaktion hat ausführlich die Kochvorschläge von IBMCHFWATSON getestet und für sehr kreativ und gut befunden.



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

Ihr TechTeam

TechTipp: ONCONFIG - DBSERVERALIASES

Der Parameter DBSERVERALIASES in der Konfigurationsdatei \$ONCONFIG ermöglicht es, bis zu 32 Aliasnamen zu vergeben, unter denen die Informix Instanz angesprochen werden kann. Die Auflistung erfolgt ohne Leerzeichen mit Komma getrennt.

Aliasnamen können verwendet werden, um unterschiedliche Optionen für die Verbindung zu Datenbankserver zu spezifizieren. So sind für die Anbindung mittels TCP, SharedMemory, Pipe oder DRDA unterschiedliche Einträge notwendig. Jeder dieser Einträge muss in der Datei „sqlhosts“ in einer eigenen Zeile eingetragen werden und kann unterschiedliche Optionen (fünfte Spalte der sqlhosts) nutzen. Für verschiedene Verbindungen zur selben Instanz kann der gleiche Hostname bzw. die selbe IP-Adresse genutzt werden, wobei sich die Ports je Verbindung unterscheiden müssen.

Jeder Eintrag der Kommunikationsart „onsoctcp“ oder „onimcsoc“ kann in DBSERVERALIASES optional zusätzlich einen Parameter erhalten, der die Anzahl der Listenerthreads bestimmt. Diese Angabe erfolgt, indem an den Aliasnamen mit „-<num>“ die Anzahl angefügt wird.

Beispiel: „ifxnl-3“ würde für den INFORMIXSERVER ifxnl drei Listenerthreads starten.

Ein Aliasname darf maximal 128 Zeichen umfassen. Die Gesamtlänge der Einträge in DBSERVERALIASES (incl. Trenner und Angabe der Listenerthreads) darf 512 Zeichen nicht überschreiten.

Nicht erlaubt im Servernamen bzw. Alias sind:

- Grossbuchstaben
- Leerzeichen
- Tabulator
- Newline
- # (Kommentarzeichen)
- - (Minuszeichen, da für die Definition der Listenerthreads genutzt)
- @ (das „at“)

Der Parameter ist **nicht dynamisch änderbar**, so dass für Anpassungen ein Neustart der Instanz notwendig ist.

Im den meisten Fällen macht es keinen Unterschied, ob eine Instanz über den DBSERVERNAME oder einen der DBSERVERALIASES angesprochen wird. Soll der Server in einer Replikation (HDR/RSS/SDS) eingebunden werden, so muss der DBSERVERNAME verwendet werden, oder einer der ALIASNAMEN als HA_ALIAS eingetragen sein (siehe nachfolgender Artikel).

TechTipp: ONCONFIG - HA_ALIAS

Die Replikation zu einem Remote Standalone Secondary Server (RSS) ist nur für den Instanznamen unterstützt, der als DBSERVERNAME in der Konfigurationsdatei eingetragen ist. Versucht man die Replikation zwischen den Servern über die Einträge aus der Liste der DBSERVERALIASES aufzubauen, so bricht die Replikation direkt nach der Definition mit einer Fehlermeldung ab.

Beispiel (DBSERVERNAME=test3, DBSERVERALIASES=test3_rss):

```
onmode -d RSS test3_rss
```

```
...
```

```
DR: new type = RSS
```

```
smx creates 3 transports to server test1
```

```
Updates from secondary allowed
```

```
Unexpected RSS Authentication Failure during Initialization Phase
```

```
Server test3 is not defined as an RSS node.
```

```
    This can occur because the server has never been define on the  
    primary or the primary was restored from a backup.
```

```
DR:Shutting down the server.
```

```
DR:Terminating the master daemon to bring system down immediately.
```

```
ERROR:can not fork secondary Server thread (MACH11 Shutdown)
```

Soll ein Name aus DBSERVERALIASES für die Replikation genutzt werden, so muss dieser als HA_ALIAS in der Konfigurationsdatei definiert werden. Der Parameter kann dynamisch mittels „onmode -wm/-wf“ gesetzt werden und ist damit sofort wirksam.

```
onmode -wf HA_ALIAS=test3_rss
```

Der erneute Aufruf zur Initialisierung der Replikation ist danach erfolgreich:

```
onmode -d RSS test3_rss
```

```
DR: new type = RSS
```

```
smx creates 3 transports to server test1
```

```
Updates from secondary allowed
```

```
Peer node test1 has version 65538
```

```
RSS Server test1 - state is now connected
```

```
setting version information for test1 65538
```

```
Logical Recovery Started.
```

```
10 recovery worker threads will be started.
```

```
Start Logical Recovery - Start Log 255, End Log ?
```

```
Starting Log Position - 260 0x41018
```

```
DR: Cleared 9868 KB of logical log in 0 seconds.
```

```
Defragmenter cleaner thread now running
```

```
Defragmenter cleaner thread cleaned:0 partitions
```

```
B-tree scanners disabled.
```

```
Started processing open transactions on secondary during startup
```

```
Finished processing open transactions on secondary during startup.
```

```
DR: RSS secondary server operational
```

TechTipp: ONCONFIG - DRINTERVAL

Der Parameter DRINTERVAL gibt die maximale Zeit (in Sekunden) an, die zwischen dem Ausschreiben der Replikationsbuffers liegen darf. Im Zusammenspiel mit dem Parameter HDR_TXN_SCOPE bestimmt dieser damit die Synchronisierung der HDR Replikation.

Für DRINTERVAL sind folgende Werte möglich:

- -1 Sync-Mode, der Primary Server wartet auf das Commit am Secondary
- 0 HDR_TXN_SCOPE entscheidet (Default)
- >0 Async-Mode. Maximale Zeitspanne in Sekunden bis zur Übertragung des Commit auf den Secondary, unabhängig von HDR_TXN_SCOPE

Der Wert ist mittels „onmode -wm/wf“ dynamisch änderbar.

Die Optionen für den Parameter HDR_TXN_SCOPE erfahren Sie im folgenden Artikel.

TechTipp: ONCONFIG - HDR_TXN_SCOPE

Der Parameter HDR_TXN_SCOPE bestimmt bei DRINTERVAL „0“ wie synchron die Replikation erfolgt. Dies hängt davon ab, ob die Datenbank mit buffered oder unbuffered Logging betrieben wird.

Der Parameter kann folgende Werte annehmen:

- FULL_SYNC Synchrone Replikation, der Primary wartet auf das Commit sowohl bei buffered, als auch bei unbuffered Logging
- NEAR_SYNC Synchrone Replikation für Datenbanken mit unbuffered Logging (analog FULL_SYNC).
- ASYNC Asynchrone Replikation

Um bei einer Replikation möglichst keine Verzögerung bei der Übertragung von Transaktionen zu erhalten, empfiehlt es sich die Datenbank mit „**unbuffered logging**“ zu betreiben. Nur so kann im Notfall eine Übernahme mit allen committeten Werten gewährleistet werden.

Der Wert für HDR_TXN_SCOPE kann dynamisch mittels „onmode -wm/wf“ geändert werden.

TechTipp: ONCONFIG - TEMPTAB_NOLOG

Die Datenbankinstanz nutzt DBSpaces, die als DBSPACETEMP in der \$ONCONFIG eingetragen sind (bzw. als Umgebungsvariable angegeben werden) für implizite temporäre Tabellen, wie sie z.B. für Subqueries und komplexe Joins benötigt werden. DBSpaces für diesen Zweck sollten mit dem Flag „-t“ als temporär erstellt werden. Diese werden dann weder gesichert, noch mit Transaktionssicherung beschrieben.

Werden explizit temporäre Tabellen erstellt („create temp table ...“ bzw. für die Zwischenablage von Ergebnissen „select ... into temp ...“), so muss in der Anweisung explizit der Zusatz „WITH NO LOG“ erfolgen, damit die TempSpaces genutzt werden können.

Der Parameter TEMPTAB_NOLOG in der \$ONCONFIG bewirkt, dass das „WITH NO LOG“ als Default für explizite, temporäre Tabellen gesetzt wird und diese damit in den TempSpaces gespeichert werden.

Beim Einsatz von Features der Hochverfügbarkeit (HDR, RSS oder SDS) sollte der Parameter TEMPTAB_NOLOG auf 1 gesetzt werden, damit ein Rollback von temporären Daten auf der Zielinstanz vermieden wird.

TechTipp: Optionen des ONSTAT (onstat -g ipl)

Für Instanzen, die an einer Replikation teilnehmen, muss der Parameter LOG_INDEX_BUILDS in der \$ONCONFIG ab dem Start der Replikation gesetzt sein. Der Parameter wurde im Newsletter Februar 2009 bereits vorgestellt.

Der „onstat -g ipl“ zeigt an, ob in einer laufenden Instanz der Parameter gesetzt ist und wann die Aktivierung.

Beispiel:

```
IBM Informix Dynamic Server Version 12.10.FC5 -- On-Line (Prim) ...  
Index page logging status: Enabled  
Index page logging was enabled at: 2014/03/12 09:21:24
```

Ist hingegen Index Logging nicht aktiviert, so erscheint:

```
Index page logging status: Disabled
```

TechTipp: Optionen des ONSTAT (onstat -g dri xxx)

Wer bereits Erfahrung mit der HDR-Replikation hat, dem wird der „onstat -g dri“ sicher geläufig sein. Dieser zeigt, ob der Primary Server und der Secondary Server miteinander verbunden sind, wann die letzte Kommunikation stattgefunden hat incl. der Position im Log, die zuletzt gesendet und empfangen wurde.

Beispiel:

```
onstat -g dri
```

Data Replication at 0x498a4028:

| Type | State | Paired server | Last DR CKPT (id/pg) | Supports Proxy Writes |
|--|--------------------------------|---------------|----------------------|-----------------------|
| primary | on | test2 | 874791 / 13748 | NA |
| DRINTERVAL | 0 | | | |
| DRTIMEOUT | 30 | | | |
| DRAUTO | 3 | | | |
| DRLOSTFOUND | /opt/informix/tmp/dr.lostfound | | | |
| DRIDXAUTO | 0 | | | |
| ENCRYPT_HDR | 0 | | | |
| Backlog | 0 | | | |
| Last Send | 2015/11/20 22:26:54 | | | |
| Last Receive | 2015/11/20 22:26:54 | | | |
| Last Ping | 2015/11/20 22:26:26 | | | |
| Last log page applied(log id,page): 874791,13772 | | | | |

bzw. am Secondary mit den Werten:

Data Replication at 0x498a6028:

| Type | State | Paired server | Last DR CKPT (id/pg) | Supports Proxy Writes |
|---|--------------------------------|---------------|----------------------|-----------------------|
| HDR Secondary | on | test1 | 874791 / 13931 | N |
| DRINTERVAL | 0 | | | |
| DRTIMEOUT | 30 | | | |
| DRAUTO | 3 | | | |
| DRLOSTFOUND | /opt/informix/tmp/dr.lostfound | | | |
| DRIDXAUTO | 0 | | | |
| ENCRYPT_HDR | 0 | | | |
| Backlog | 0 | | | |
| Last Send | 2015/11/20 22:26:07 | | | |
| Last Receive | 2015/11/20 22:26:07 | | | |
| Last Ping | 2015/11/20 22:25:45 | | | |
| Last log page applied(log id,page): 0,0 | | | | |

Weitere Informationen sind mit einem zusätzlichen Parameter zu erhalten.

Folgende Optionen sind dokumentiert:

- que Ausgabe der Warteschlangen für Send und Acknowledge
- ckpt Informationen zur Verarbeitung der Checkpoints am Secondary
- sta Letzte Statusänderungen der Replikation
- idx Informationen zur Replikation der Indexe.
- Msg Letzte Kommunikationen zwischen Primary und Secondary

„**ckpt**“ zeigt am Secondary die Informationen zum letzten Checkpoint:

DR Checkpoint processing:

| | |
|-------------------------------|---------------------|
| Save State | N |
| Pages Saved | 0 |
| Save Area | none |
| Received log id, page | 874791,15043 |
| Saved log id, page | 0,0 |
| Drain log id, page | 0,0 |
| Processed log id, page | 874791,15043 |
| Pending checkpoints | 0 |

Wobei „Save State“ folgende Werte annehmen kann:

- **B** (buffering) wenn Logs vom Primary in die Zwischenablage (Staging) geschrieben werden
- **D** (draining) wenn Logs aus der Zwischenablage gelöscht werden
- **N** (normal) wenn die Zwischenablage leer ist und alle Logs verarbeitet sind

Saved Log ID, Drained Log ID beziehen sich auf die Zwischenspeicherung der Logs. Pending Checkpoints gibt an wie viele Checkpoints bei der Synchronisierung noch ausstehen.

„**sta**“ listet die letzten Statusänderungen (max. 10) der Replikation auf:

Past 10 HDR states:

```
State           : On (0x20)
Failure recovery flag: FRCHECK FRCONN (0x1001040)
Thread id      : 102
Thread name    : dr_prsend
Time           : 2015-11-19.22:38:17
```

```
State           : Connecting (0x40)
Failure recovery flag: FRCONN (0x1001000)
Thread id      : 102
Thread name    : dr_prsend
Time           : 2015-11-19.22:38:16
```

```
State           : Off (0x10)
Failure recovery flag: (0x1000000)
Thread id      : 102
Thread name    : dr_prsend
Time           : 2015-11-19.22:37:36
```

„**idx**“ zeigt den Status der Synchronisierung der Indexe:

```
DR Primary Index sync logid/logpos:    0:0x      0
DR Primary Active Index Transfer: none.
DR Primary Pending Index Transfers:    none.
```

bzw.

```
DR Secondary Index sync logid/logpos:  874791:0x   4b71
DR Secondary Active Index Transfer:    none.
DR Secondary Pending Index Transfers:  none.
```

„**que**“ zeigt zusätzlich den Queue-Status:

```
Pending Msg to Send 0
ACK QUEUE 874791:3687fff
Applied QUEUE 874791:3687240
```

Wobei jeweils die Unique-ID und die Position innerhalb des Logs angegeben wird, wo die Transaktion zu finden ist.

„**msg**“ listet die letzten Kommunikationen zwischen Primary und Secondary auf:

```
Past 10 DR messages:
DR message sent at 1448027661      : DR_MSG_REQACK
DR message sent at 1448027660      : DR_MSG_REQACK
DR message sent at 1448027659      : DR_MSG_REQACK
DR message sent at 1448027658      : DR_MSG_GETWORKLOAD
DR message sent at 1448027657      : DR_MSG_REQACK
DR message sent at 1448027656      : DR_MSG_REQACK
DR message sent at 1448027655      : DR_MSG_REQACK
DR message sent at 1448027654      : DR_MSG_REQACK
DR message sent at 1448027653      : DR_MSG_REQACK
DR message sent at 1448027652      : DR_MSG_GETWORKLOAD
```

bzw.

```
Past 10 DR messages:
DR message sent at 1448027359      : DR_MSG_RETWORKLOAD
DR message sent at 1448027359      : DR_MSG_ACK
DR message sent at 1448027358      : DR_MSG_ACK
DR message sent at 1448027357      : DR_MSG_ACK
DR message sent at 1448027356      : DR_MSG_ACK
DR message sent at 1448027356      : DR_MSG_ACK
DR message sent at 1448027355      : DR_MSG_REPREQS
DR message sent at 1448027355      : DR_MSG_ACK
DR message sent at 1448027355      : DR_MSG_ACK
DR message sent at 1448027354      : DR_MSG_ACK
```

TechTipp: Time-Moving-Objects (STS_TrajectoryCross) Teil 8

Die Funktion **STS_TrajectoryCross** gibt an, ob ein bewegtes Objekt eine Region durchquert. Hierbei kann eine Distanz um eine Region angegeben werden. Region ist dabei eine Geographie, die entweder ein Punkt, eine Linie, ein Polygon oder ein Multipolygon sein kann. Anwendungsgebiete gibt es hier reichlich. Z.B. die Auslage, dass Mietwägen nicht nach Osteuropa gefahren werden dürfen, könnte so überwacht und ggf. als Alarm zurückgegeben werden.

Die Syntax:

```
STS_TrajectoryCross(  
    ts_tabname LVARCHAR,  
    obj_id LVARCHAR,  
    ts_colname LVARCHAR,  
    starttime DATETIME YEAR TO FRACTION(5),  
    endtime DATETIME YEAR TO FRACTION(5),  
    geometry ST_Point,  
    max_distance REAL)  
returns Boolean
```

In unserem Beispiel nutzen wir einen Punkt (z.B. der Glühweinstand am Weihnachtsmarkt) und analysieren welche Objekte sich näher als x Meter an diesen Stand annähern:

```
SELECT id,STS_TrajectoryCross('moves_ts',id, 'data',  
    '2015-09-21 08:42:50.00000',  
    '2015-09-21 08:43:50.00000',  
    '4326 point(47.551740 9.696854)', 400) as crosses_400m  
FROM moves_ts;
```

```
SELECT id,STS_TrajectoryCross('moves_ts',id, 'data',  
    '2015-09-21 08:42:50.00000',  
    '2015-09-21 08:43:50.00000',  
    '4326 point(47.551740 9.696854)', 500) as crosses_500m  
FROM moves_ts;
```

Ergebnis:

id crosses_400m

| | |
|---|---|
| 1 | f |
| 2 | f |

id crosses_500m

| | |
|---|---|
| 1 | t |
| 2 | f |

TechTipp: Time-Moving-Objects (STS_TrajectoryIntersect) Teil 9

Die Funktion **STS_TrajectoryIntersect()** gibt für bewegte Objekte an, ob diese eine Region betreten. Hierbei kann eine Distanz um eine Region angegeben werden. Region ist dabei eine Geographie, die entweder ein Punkt, eine Linie, ein Polygon oder ein Multipolygon sein kann. Auch hier kann eine Reihe von Anwendungen gefunden werden, z.B. die Sperrzonen im Naturschutzgebiet, die mit wassergefährdenden Stoffen nicht befahren werden dürfen.

```
SELECT id, STS_TrajectoryIntersect('moves_ts',id,'data',
  '2015-09-21 08:42:50.00000',
  '2015-09-21 08:43:50.00000',
  '4326 point(47.551740 9.696854)', 500) as crosses_400m
FROM moves_ts;
```

```
SELECT id, STS_TrajectoryIntersect('moves_ts',id,'data',
  '2015-09-21 08:42:50.00000',
  '2015-09-21 08:43:50.00000',
  '4326 point(47.551740 9.696854)', 1000) as crosses_1000m
FROM moves_ts;
```

Ergebnis:

```
id crosses_500m
1          t
2          f
```

```
id crosses_1000m
1          t
2          t
```

TechTipp: Time-Moving-Objects (STS_GetIntersectSet) Teil 10

Die Funktion **STS_GetIntersectSet()** gibt ein Set von bewegten Objekten zurück, die eine Region betreten. Hierbei kann eine Distanz um eine Region angegeben werden. Region ist dabei eine Geographie, die entweder ein Punkt, eine Linie, ein Polygon oder Multipolygon sein kann.

Syntax:

```
STS_GetIntersectSet(
  ts_tabname LVARCHAR,
  ts_colname LVARCHAR,
  starttime DATETIME YEAR TO FRACTION(5),
  endtime DATETIME YEAR TO FRACTION(5),
  geometry LVARCHAR,
  max_distance REAL)
returns Set
```

Beispiel:

```
SELECT STS_GetIntersectSet('moves_ts','data',
  '2015-09-21 08:42:50.00000',
  '2015-09-21 08:43:50.00000',
  '4326 point(47.551740 9.696854)', 500) as crosses_500m
FROM moves_ts;
```

```
SELECT STS_GetIntersectSet('moves_ts','data',
  '2015-09-21 08:42:50.00000',
  '2015-09-21 08:43:50.00000',
  '4326 point(47.551740 9.696854)', 1000) as crosses_1000m
FROM moves_ts;
```

Ergebnis:

```
crosses_500m  SET{'1'}
crosses_1000m SET{'1','2'}
```

TechTipp: Time-Moving-Objects (STS_TrajectoryWithin) Teil 11

Die Funktion **STS_TrajectoryWithin()** gibt für bewegte Objekte an, ob diese im angegebenen Zeitraum innerhalb einer Region bleiben. Hierbei kann eine Distanz um eine Region angegeben werden. Region ist dabei eine Geographie, die entweder ein Punkt, eine Linie, ein Polygon oder ein Multipolygon sein kann.

```
SELECT STS_TrajectoryWithin('moves_ts',id,'data',
  '2015-09-21 08:42:50.00000',
  '2015-09-21 08:43:50.00000',
  '4326 point(47.551740 9.696854)', 500) as crosses_500m
FROM moves_ts;
```

```
SELECT STS_TrajectoryWithin('moves_ts',id,'data',
  '2015-09-21 08:42:50.00000',
  '2015-09-21 08:43:50.00000',
  '4326 point(47.551740 9.696854)', 1000) as crosses_1000m
FROM moves_ts;
```

Ergebnis:

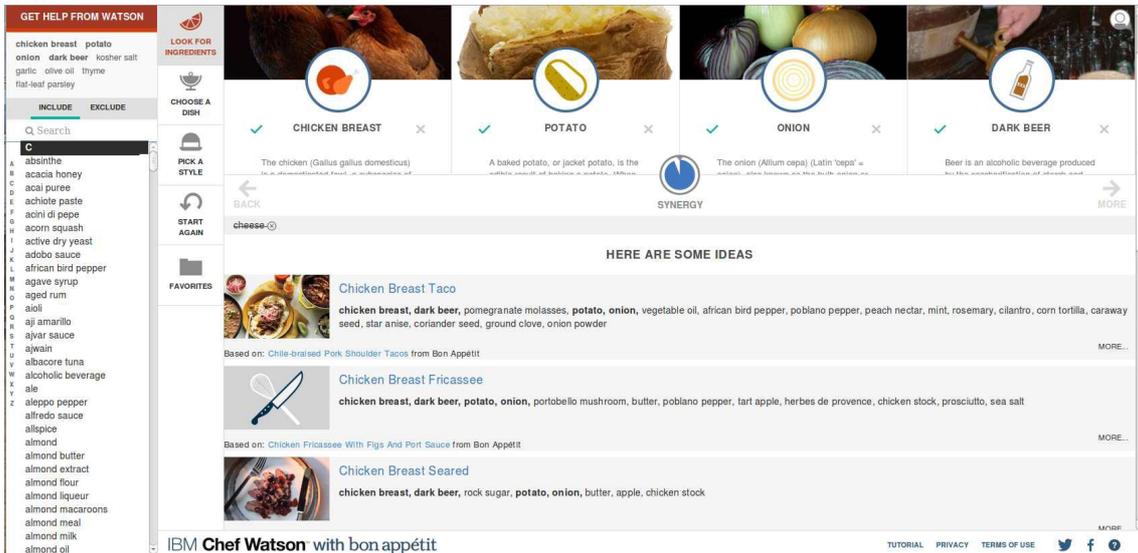
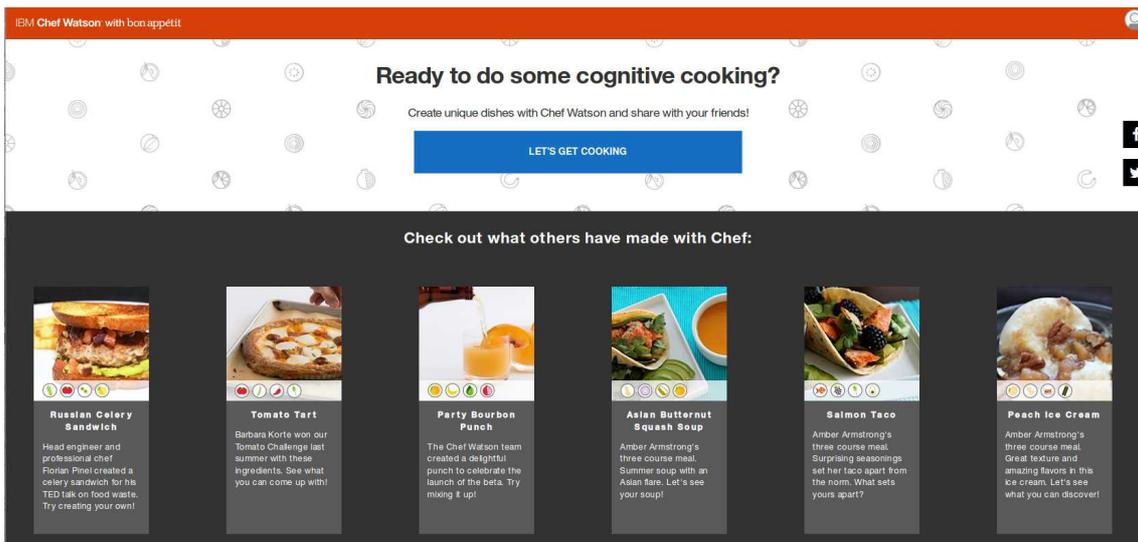
```
id crosses_500m
1             f
2             f
```

```
id crosses_1000m
1             t
2             f
```

Das das Objekt mit ID 2 im angegebenen Zeitraum das Gebiet betritt und nicht immer im Umkreis von 1000m war, ist das Ergebnis „false“.

TechTipp: IBM CHEFWATSON – Kreatives Kochen

Wer gerne kocht, der wird sich über die Ideen von WWW.IBMCHEFWATSON.COM freuen. Nach schneller Anmeldung steht eine Ideenküche zur Verfügung, die keine Wünsche offen lässt. Das Vorgehen ist simpel: Man trägt ein, was man im Kühlschrank oder in der Küche so findet, nimmt die Dinge auf eine „Exclude-List“, die jemand aus der Familie nicht mag, und schon erhält man kreative Vorschläge mit Fotos, was aus den Zutaten werden kann. Folgt man dann dem Link, so kommt man zum ausführlichen Rezept.



Viel Spass beim Testen !

Termin: CDC Bootcamp in Ehningen von 26. bis 29. Januar

In einigen Artikeln haben wir bereits auf die Option hingewiesen, mittels **Infosphere Data Replication** (besser bekannt als **Change Data Capture**) Daten unterschiedlicher Datenbanken (z.B. INFORMIX, DB2, ORACLE, ...) auszutauschen. Auch die Option eines fortlaufenden Audit mittels CDC auf eine geschützt installierte Datenbank könnte für eine Reihe an Informix Kunden interessant sein.

Wir möchten Sie daher zu einem CDC Bootcamp im Januar in unserem Innovation Center in Ehningen einladen. Wie immer ist der Workshop als Hands-On konzipiert, damit Sie selbst anhand von Beispielen die vielfältigen Möglichkeiten dieses Produktes erfahren können. Um jedem Teilnehmer die Möglichkeit der praktischen Übungen zu bieten ist die Anzahl der Teilnehmer begrenzt. Melden Sie sich bitte daher bald möglichst an um noch einen Platz zu sichern. Die Teilnahme ist kostenlos.

Das Bootcamp richtet sich an Entwickler und Administratoren, die sich mit Datenreplikation beschäftigen. Vorkenntnisse sind nicht unbedingt erforderlich, wären aber von Vorteil. Als Vorbereitung könnten erste Erfahrungen mit einer Demoversion nützlich sein.

Themenübersicht:

- IBM Data Replication portfolio
- Introduction to IBM Data Replication, Change Data Capture (CDC) and Data Event Publishing technologies
- Different replication scenarios
- Data auditing, data distribution, data consolidation and bi-directional replication solutions
- Use of Change Data Delivery integrated with DataStage
- Installation and configuration of the product
- Platform specific issues

Anmeldung bitte per Mail an: tim.horstkotte@de.ibm.com

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.iiug.org/intl/deu>

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 Nachricht falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

Die Autoren dieser Ausgabe

Gerd Kaluzinski IT-Specialist INFORMIX, Infosphere CDC, 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

Markus Holzbauer IBM Informix Advanced Support
 IBM Software Group, Information Management Support
holzbauer@de.ibm.com

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Gerd Kaluzinski (Care-Paket für die Redaktion aus dem Osten)