

## Willkommen zum „IBM Informix Newsletter“

### Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: TimeSeries Datablade (Teil 1: Erstellen).....	2
TechTipp: TimeSeries Datablade (Teil 2: Einfügen der Daten).....	4
TechTipp: TimeSeries Datablade (Teil 3: Abfrage der Daten mit SQL).....	5
Schulung: INFORMIX 11.70 – New Features (IX3020DE).....	7
Termine: INFORMIX 11.70 - 3-tägiges Bootcamp mit Hands-on Übungen.....	8
WebTipp: freeinformix.com.....	9
WebTipp: Redbook - Embedding IBM Informix.....	9
WebTipp: Redbook - Migrating from Microsoft SQL Server to Informix.....	10
Anmeldung / Abmeldung / Anmerkung.....	10
Die Autoren dieser Ausgabe.....	11

### Aktuelles

Liebe Leserinnen und Leser,

ein Beispiel wie eine Datenbank dabei helfen kann, unseren Planeten smarter zu machen, ist das TimeSeries Datablade. Der optimierte Einsatz von Ressourcen wird erst möglich, wenn z.B. intelligente Stromzähler den Verbrauch so steuern, dass Strom dann genutzt wird, wenn dieser günstig zur Verfügung steht. Um Ihnen das Thema näher zu bringen, haben wir diesen Newsletter zum großen Teil dem Thema TimeSeries gewidmet. Lassen Sie uns gemeinsam den Planeten ein wenig smarter machen.



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

## TechTipp: TimeSeries Datablade (Teil 1: Erstellen)

Seit Version 11.70 ist das TimeSeries Datablade im Bundle des INFORMIX Servers enthalten.

Immer öfter besteht die Notwendigkeit, Daten zeitabhängig zu sammeln und zu speichern. Eines der bekanntesten Beispiele in diesem Bereich sind die intelligenten Stromzähler, die es ermöglichen, den Verbrauch zeitlich zuzuordnen und als Folge den günstigen Strom dann zu nutzen, wenn dieser im Überfluss vorhanden ist.

Ein anderes Beispiel sind intelligente Verkehrsleitsysteme die helfen, lange Staus zu verhindern und so helfen die Umwelt zu schonen.

Dies und viele andere Projekte im Rahmen der Aktivitäten rund um das Thema „[Smarter Planet](#)“ der IBM basieren auf Daten, die zur Auswertung in der Datenbank gespeichert werden.

Vor diesem Hintergrund wollen wir an einem praktischen Beispiel die Funktionsweise und die Vorteile des TimeSeries Datablade demonstrieren.

Der Vorteil dieses Datablades liegt einerseits an der schnelleren und effizienteren Speicherung der Daten im Gegensatz zu üblicher Ablage in Tabellen. Hierbei wird der Overhead so weit als möglich reduziert, um schneller auf die Nutzdaten zugreifen zu können. Zudem steht eine Fülle an Funktionen bereit, die den Zugriff auf die gespeicherten Daten vereinfacht.

Im Beispiel betrachten wir die Verwendung des TimeSeries Datablades bei regulären Zeitreihen. Hierbei handelt es sich um Werte die jeweils nach einem festen Zeitintervall anfallen. Für irreguläre Zeitreihen, bei denen die Werte in ungleichmäßigen Zeitabständen erfasst werden, wird jeweils eine Timestamp mit gespeichert. Einige Funktionen zeigen ein leicht abweichendes Verhalten (wie z.B. `putelem()`).

Da bei irregulären Zeitreihen der Timestamp immer mit gespeichert wird, ist der Vorteil des reduzierten Overhead (erzielt durch Eliminierung der redundanten Informationen) geringer. Es bleibt jedoch der Vorteil des komfortablen Zugriffs auf die Daten mittels optimierter Funktionen aus dem Datablade erhalten.

Beispiel: Stündliche Wetterwerte von Messstationen sammeln:

Das Timeseries Datablade muss in der Datenbank registriert werden, damit die Funktionen und Datentypen in der Datenbank (im Beispiel „wetter“) zur Verfügung stehen.

Die Version und den Namen des Datablades sehen Sie an dem Verzeichnis, das unter `$INFORMIXDIR/extend` bei der Installation des Datablades erstellt wurde.

Die Installation aus der Kommandozeile kann mittels „Blademanager“ erfolgen:

```
blademgr  
>register TimeSeries.4.01.UC8 wetter
```

Alternativ kann die Registrierung auch als SQL-Befehl erfolgen:

```
EXECUTE FUNCTION sysbldprepare('TimeSeries.4.01.UC8', 'create');
```

**Achtung:** Die Registrierung mittels „`sysbldprepare()`“ ist nur in Datenbanken mit Logging möglich.

**Anmerkung:** Bei Datenbanken mit Logging ist eine Registrierung von Standard Datablades nicht mehr notwendig und erfolgt implizit bei erster Nutzung der Funktionalität.

Um die Daten als TimeSeries zu speichern, muss ein Rowtype erstellt werden, der einen Timestamp enthält:

```
create row type r_wetter_time (  
    t_stamp datetime year to fraction(5),  
    temperatur decimal(6,2),  
    regen int,  
    druck int,  
    hygro decimal(6,2),  
    wind int,  
    windrichtung int  
);
```

Nun müssen noch die Kalenderfunktionen des TimeSeries Datablades mit Werten gefüllt werden.

Für die stündliche Erfassung (im Beispiel) wird folgendes Pattern erstellt:

```
insert into CalendarPatterns values ( 'All_Hours', '{24 on}, hour');
```

In diesem Fall wird rund um die Uhr erfasst. Soll z.B. nur tagsüber erfasst werden, so könnte dies mittels {6 off, 12 on, 6 off} angegeben werden.

Dies hätte zur Folge dass die ersten 6 Stunden nicht in der TimeSeries sind, dann 12 Stunden erfasst werden und dann wieder 6 Stunden keine Daten enthalten.

Der Startzeitpunkt und der Zeitpunkt der ersten Erfassung müssen in die Kalendertabelle eingetragen werden:

```
insert into CalendarTable ( c_name, c_calendar)  
values ('Wetter Tag',  
    'startdate(2011-03-14 00:00:00.00000),  
    pattstart(2011-03-14 00:00:00.00000), pattname(All_Hours)')
```

Nun kann der TimeSeries Container erstellt werden, in dem die Daten abgelegt werden:

```
execute procedure TSContainerCreate  
    ("Wetter_Day_Cont", "datadbs", "r_wetter_time", 42000, 13000);
```

Der Container im Beispiel wurde mit 42 MB (42000 kB) im DBSpace „datadbs“ angelegt. Die Nextsize für eine Erweiterung wurde auf 13 MB gesetzt.

Schließlich fehlt noch die Tabelle, die die Werte der TimeSeries aufnehmen soll:

```
create table tageswetter (  
    location char(12),  
    kfz char(3),  
    wetterdaten timeseries(r_wetter_time)  
);
```

## TechTipp: TimeSeries Datablade (Teil 2: Einfügen der Daten)

Das Einfügen der Daten in TimeSeries erfolgt über spezielle Funktionen, die das TimeSeries Datablade zur Verfügung stellt. Dies sind vor Allem die Funktionen „**putelem()**“ und **putelemnodups()**“, deren Verhalten sich bei irregulären Zeitreihen unterscheidet. Zuvor muss jedoch noch die TimeSeries selbst erstellt werden, was mittels eines Inserts in die Tabelle und der Funktion „**tscreate()**“ erfolgt.

Mit einem Startwert, der die festen Bestandteile der Information einfügt (z.B. den Standort) sowie den Startzeitpunkt, wird die Tabelle initial gefüllt.

```
insert into tageswetter values
('Lindau', 'LI', tscreate('Wetter Tag',
    '2011-03-14 00:00:00.00000', 24, 0, 1, 'Wetter_Day_Cont'));
```

Nun können die Messwerte aufgenommen werden:

```
update tageswetter
set wetterdaten = putelem(wetterdaten, row(NULL::datetime year to fraction(5),
6.6, 0, 1024, 34, 13, 142)::r_wetter_time) where kfz = 'LI';
update tageswetter
set wetterdaten = putelem(wetterdaten, row(NULL::datetime year to fraction(5),
7.6, 0, 1026, 32, 11, 141)::r_wetter_time) where kfz = 'LI';
...
update tageswetter
set wetterdaten = putelem(wetterdaten, row('2011-03-14 17:42:00.00000'::datetime
year to fraction(5),
16.3, 0, 1022, 30, 5, 137)::r_wetter_time) where kfz = 'LI';
```

Werden Werte mit expliziter Zeitangabe eingetragen, so werden die Intervalle, die zwischen dem letzten gültigen Wert und dem neuen Eintrag liegen als „NULL“ gekennzeichnet. Wird ein bereits bestehender Wert neu eingetragen, so wird der bisherige Wert mit den neuen Daten aktualisiert. (Die Erklärung des unterschiedlichen Verhaltens bei irregulären Zeitreihen würde den Rahmen dieses Newsletters sprengen.)

Kommen Werte anderer Stationen dazu, muss vor diesen wiederum ein initialer Eintrag gestellt werden:

```
insert into tageswetter values
('Ravensburg', 'RV', TSCreate('Wetter Tag',
    '2011-03-14 00:00:00.00000', 24, 0, 1, 'Wetter_Day_Cont'));
update tageswetter
    set wetterdaten = PutElem(wetterdaten, row(NULL::datetime year to
fraction(5), 3.6, 0, 1024, 50, 13, 120)::r_wetter_time) where kfz = 'RV';
update tageswetter
    set wetterdaten = PutElem(wetterdaten, row(NULL::datetime year to
fraction(5), 4.6, 0, 1025, 52, 3, 121)::r_wetter_time) where kfz = 'RV';
```

## TechTipp: TimeSeries Datablade (Teil 3: Abfrage der Daten mit SQL)

Zur Formatierung der Ausgabe der Daten mittels SQL stehen zahlreiche Funktionen im TimeSeries Datablade zur Verfügung. Auch ohne diese Funktionen können die Daten mittels SQL ausgelesen werden. Die Informationen aus dem Feld vom Typ TimeSeries werden dabei als „Startzeitpunkt“ und einem Array von Werte ausgegeben. Jeder Eintrag im Array ist dabei genau ein Zeitintervall nach dem vorhergehenden Eintrag (immer unter der Voraussetzung regulärer Zeitreihen).

Eine Abfrage (select \*) auf die Tabelle „tageswetter“ liefert Daten in der Form:

```
select * from tageswetter
```

Ergebnis:

```
location      Lindau
kfz           LI
wetterdaten  origin(2011-03-14 00:00:00.00000), calendar(Wetter Tag), container
              (Wetter_Day_Cont), threshold(24), regular, [(6.60      ,0
              1024      ,34.00  ,13      ,142      ), (7.60      ,0
              ,1026      ,32.00  ,11      ,141      ), (8.60      ,0
              ,1026      ,31.00  ,9      ,141      ), (9.60      ,0
              ,1026      ,31.00  ,4      ,140      ), (10.50     ,
              0      ,1025      ,31.00  ,4      ,149      ), (11.80
              ,0      ,1025      ,31.00  ,4      ,143      ), (11.
              90      ,0      ,1025      ,31.00  ,4      ,143      ), (
              12.60     ,0      ,1025      ,31.00  ,4      ,143      )
              , (13.30     ,0      ,1024      ,31.00  ,4      ,143
              ), (14.10     ,0      ,1024      ,31.00  ,4      ,148
              ), (14.30     ,0      ,1024      ,32.00  ,4      ,140
              ), (14.60     ,0      ,1024      ,33.00  ,7
              ,147      ), (14.90     ,0      ,1024      ,33.00  ,7
              ,147      ), (15.20     ,0      ,1024      ,33.00  ,7
              ,141      ), (15.70     ,0      ,1025      ,33.00  ,11
              ,141      ), NULL, NULL, (16.30     ,0      ,1022
              ,30.00     ,5      ,137      )]
```

```
location      Ravensburg
kfz           RV
wetterdaten  origin(2011-03-14 00:00:00.00000), calendar(Wetter Tag), container
              (Wetter_Day_Cont), threshold(24), regular, [(3.60      ,0
              1024      ,50.00  ,13      ,120      ), (4.60      ,0
              ,1025      ,52.00  ,3      ,121      )]
```

Die Abfrage einzelner Werte kann mit der Funktion „**getelem()**“ durchgeführt werden. Beispiel:

```
select kfz, getelem(wetterdaten, '2011-03-14 17:00:00.00000'::datetime year to
fraction(5)) from tageswetter where kfz = 'LI';
```

Ergebnis:

```
kfz          LI
(expression) ROW('2011-03-14 17:00:00.00000',16.30      ,0
              30.00      ,5      ,137      )
```

Ebenso ist die Abfrage einzelner Elemente der Wertereihe möglich. Wird bei der Abfrage eine Uhrzeit angegeben, die nicht genau dem Intervall entspricht, so wird der letzte gültige Wert vor diesem Zeitpunkt zurückgeliefert:

Beispiel: Abfrage der Temperatur am 14.03.2011 um 17:42 Uhr in Lindau.

```
select kfz,
getelem(wetterdaten, '2011-03-14 17:42:13.00000'::datetime year to
fraction(5)).t_stamp ,
getelem(wetterdaten, '2011-03-14 17:42:13.00000'::datetime year to
fraction(5)).temperatur
from tageswetter where kfz = 'LI';
```

Ergebnis:

```
kfz t_stamp                temperatur
LI  2011-03-14 17:00:00.00000    16.30
```

Sollen die Werte eines begrenzten Bereichs ausgegeben werden, so kann dies mit der Funktion „clip()“ erfolgen.

Beispiel: Wetterwerte der Station „LI“ 4 Stunden vor 14.03.2011 17:42 Uhr:

```
select clip(wetterdaten,
'2011-03-14 17:42:00.00000'::datetime year to fraction(5)
- interval(4) hour to hour,
'2011-03-14 17:42:00.00000'::datetime year to fraction(5) )
from tageswetter where kfz = 'LI';
```

Ergebnis:

```
(expression)  origin(2011-03-14 13:00:00.00000), calendar(Wetter Tag), containe
r(Wetter_Day_Cont), threshold(24), regular,
[(15.20 ,0 ,1024 ,33.00 ,7 ,141 ),
(15.70 ,0 ,1025 ,33.00 ,11 ,141 ),
NULL, NULL,
(16.30 ,0 ,1022 ,30.00 ,5 ,137 )]
```

Die Abfrage kann auch auf die letzten n Werte vor (n<0) oder nach (n>0) einem Zeitpunkt erfolgen. Hierzu dient die Funktion „clipcount()“:

Beispiel:

```
select kfz, "clipcount", clipcount(wetterdaten, '2011-03-14 04:00:00.00000',
-2 ) from tageswetter where kfz = 'LI';
```

Ergebnis:

```
kfz      LI
(constant) clipcount
(expression)  origin(2011-03-14 03:00:00.00000), calendar(Wetter Tag), containe
r(Wetter_Day_Cont), threshold(24), regular, [(9.60 ,0
,1026 ,31.00 ,4 ,140 ), (10.50 ,0
,1025 ,31.00 ,4 ,149 )]
```

Die Ermittlung des letzten gültigen Eintrags je Zeitreihe kann mittels „**getlastvalid()**“ erfolgen:

```
select kfz, "getlastvalid", getlastvalid(wetterdaten,
    current year to fraction(5)) from tageswetter;
```

Ergebnis:

```

kfz          LI
(constant)   getlastvalid
(expression) ROW('2011-03-14 17:00:00.00000',16.30 ,0      ,1024      ,
33.00      ,7      ,141      )

```

```

kfz          RV
(constant)   getlastvalid
(expression) ROW('2011-03-14 01:00:00.00000',4.60 ,0      ,1025      ,
52.00      ,3      ,121      )

```

Es gibt eine Vielzahl weiterer Funktionen, die den Zugriff auf die Werte vereinfachen. Testen Sie selbst z.B. `getfirstelem()`, `getnthvalid()`, `getlastnonnull()`, ...

Im den weiteren Artikeln zum TimeSeries Datablade (vorgesehen für die Ausgabe April 2011) wird die Möglichkeit der Ausgabe über Listen, Collection-Derived Tables und virtuelle Tabellen behandelt.

## Schulung: INFORMIX 11.70 – New Features (IX3020DE)

Unsere Schulungsabteilung hat einen neuen, 3-tägigen Kurs ins Angebot aufgenommen:  
**„Informix 11.7 New Features“**

Der Kurs beinhaltet praktische Übungen zu vielen der neuen Features der Version 11.70 und ist daher die ideale Gelegenheit, sich mit diesen vertraut zu machen.

Termine:

23.-25. Mai 2011                      Frankfurt am Main                      (ZMCB )

04.-06. Juli 2011                      Stuttgart-Degerloch                      (ZMCE )

19.-21. September 2011              München                                      (ZMCG)

Anmeldung und weitere Informationen unter:

[http://www.ibm.com/jct03001c/services/learning/ites.wss/de/de?pageType=course\\_description&includeNotScheduled=y&courseCode=IX3020DE](http://www.ibm.com/jct03001c/services/learning/ites.wss/de/de?pageType=course_description&includeNotScheduled=y&courseCode=IX3020DE)

Der Kurs wendet sich an Administratoren, die Erfahrung mit INFORMIX in einer der Versionen 9.40/10.00/11.10/11.50 haben und die Vorteile der neuen Features kennenlernen und ggf. nutzen wollen. SQL- und Linuxkenntnisse sind keine Voraussetzung, könnten aber bei den Übungen hilfreich sein.

## 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 3-tägigen Hands-on Workshop.  
Im Anschluss an den Workshop kann eine Zertifizierung durchgeführt werden.

Termin: 04.-06. April 2011 Düsseldorf – **Achtung: Termin geändert !!!**

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: freeinformix.com**

Passend zum Thema TimeSeries gibt es einen interessanten Artikel im Web, der die Kosten und den Platzbedarf zwischen Speicherung in traditionellen Tabellen und Timeseries vergleicht.

Überzeugen Sie sich selbst:

<http://www.freeinformix.com/telco-billing-with-timeseries.html>

Auch die anderen Beiträge auf freeinformix.com sind lesenswert:

<http://www.freeinformix.com/>

**WebTipp: Redbook - Embedding IBM Informix**

Im Februar ist ein Redbook zum Thema „Embedding IBM Informix“ erschienen.

Inhalt:

Chapter 1. Introduction

Chapter 2. Embeddability basics

Chapter 3. Preparing to embed IBM Informix

Chapter 4. Installation strategies

Chapter 5. Deployment

Chapter 6. IBM Informix configuration for embeddability

Chapter 7. The SQL administration API

Chapter 8. Memory and storage management

Chapter 9. Automating management through tasks and sensors

Chapter 10. Administration of an embedded IBM Informix system

Chapter 11. Embedding high availability IBM Informix configurations

Chapter 12. IBM Informix appliances

Appendix A. SQL administration commands and scripts

Appendix B. IBM Informix Version 11.50 installation and configuring client connectivity

Appendix C. Event classes and event ids

Das Redbook ist zu finden unter:

<http://www.redbooks.ibm.com/abstracts/sg247666.html?Open>

Viel Spaß beim Lesen und Testen !

## WebTipp: Redbook - Migrating from Microsoft SQL Server to Informix

Ein weiteres Redbook wurde aktuell überarbeitet. Es beschreibt, wie der Umstieg von Microsoft SQL Server zu INFORMIX einfach und schnell bewerkstelligt werden kann.

Inhalt:

- Chapter 1. Introduction
- Chapter 2. Architecture overview
- Chapter 3. Migration methodology
- Chapter 4. SQL considerations
- Chapter 5. Database schema and data migration
- Chapter 6. Application conversion
- Chapter 7. Informix configuration and administration
- Appendix A. Terminology mapping
- Appendix B. Data types
- Appendix C. Function mapping
- Appendix D. Operator mapping
- Appendix E. Administrative and monitoring task mapping
- Appendix F. Database server utilities
- Appendix G. SQL limits

Das komplette Redbook ist zu finden unter:

[http://www.redbooks.ibm.com/abstracts/sg247847.html?Open&cm\\_sp=MTE16457](http://www.redbooks.ibm.com/abstracts/sg247847.html?Open&cm_sp=MTE16457)

## 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.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.bytec.de/de/software/ibm\\_software/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.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](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)

Sowie unterstützende Teams im Hintergrund.

Fotonachweis:                    Gerd Kaluzinski (Redaktionsgarten)