

Willkommen zum „IBM Informix Newsletter“

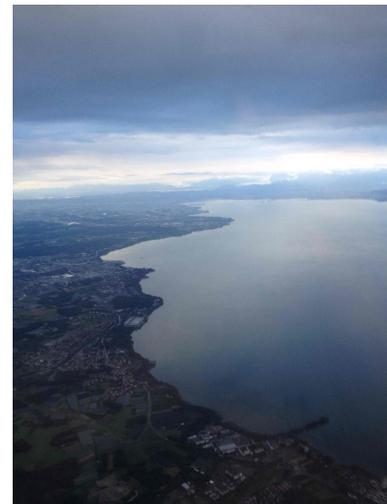
Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: New Features 12.10.xC9.....	2
TechTipp: ONCONFIG – DEFAULTESCCHAR / ESCAPE	2
TechTipp: REGEX Datablade – Reguläre Ausdrücke im SQL	3
TechTipp: regex_match() – Suche nach regulären Ausdrücken.....	5
TechTipp: regex_replace() – Austausch von Ausdrücken in Texten.....	6
TechTipp: regex_extract() – Extrahieren von Ausdrücken.....	8
TechTipp: regex_split() – Aufteilen von Texten.....	9
TechTipp: regex_set_trace() – Trace für regex_...() setzen.....	9
TechTipp: regex_release() – Version der Datablades regex.....	10
TechTipp: LATERAL – Derived table mit Referenz zum Aufruf.....	11
TechTipp: SQL Package Extension DBMS.....	12
TechTipp: DBMS_LOB_GETLENGTH().....	13
TechTipp: DBMS_LOB_SUBSTR().....	13
Termin: "What The Data!?"- Hackathon mit Informix in Hamburg.....	13
Termin: 70. IUG-Workshop am 24. Oktober 2017 in Hamburg.....	14
Versionsinfo: 12.10.xC9 ist verfügbar.....	14
Anmeldung / Abmeldung / Anmerkung.....	14
Die Autoren dieser Ausgabe.....	15

Aktuelles

Liebe Leserinnen und Leser,

der INFORMIX Newsletter feiert mit dieser Ausgabe seinen 11. Geburtstag. Vielen Dank, dass Sie uns so lange die Treue gehalten haben. In dieser langen Zeit blieb der Kern der Newsletter Redaktion weitgehend unverändert. Mit der Ausgabe Q4/2017 wird es einige Änderungen in der Redaktion geben, da das bisherige Team neue Aufgaben und Herausforderungen übernimmt. Die Leitung der Redaktion übernimmt Dr. Andreas Weininger. Der Newsletter mit aktuellen Informationen, sowie Tipps&Tricks bleibt in leicht modernisierter Form erhalten, und auch der Turnus mit einer Ausgabe je Quartal ist etabliert.



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

TechTipp: New Features 12.10.xC9

Die Version 12.10.xC9 beinhaltet eine Vielzahl an Fehlerbereinigungen und Verbesserungen. Zudem sind einige neue Features verfügbar. Das Highlight der Features sind sicher die **Smart Trigger** um Daten aktiv an eine Applikation zu liefern. Damit Applikationen nicht aktiv nach Änderungen von Daten suchen müssen, was unnötige Last erzeugt, können nun JDBC Smart Trigger erstellt werden, die Änderungen an Datensätzen erkennen und im Format JSON an den Datenbank-Client schicken. Zudem wurde im Bereich des **Tracking von moving objects**, das bereits in einer früheren Ausgabe des Informix Newsletters beschrieben wurde, um eine Vielzahl von Funktionen erweitert.

TechTipp: ONCONFIG – DEFAULTESCCHAR / ESCAPE

Zeichen für Wildcards in Vergleichsoperationen werden z.B. beim Operator „LIKE“ mit „_“ für ein beliebiges Zeichen, „%“ für beliebig viele Zeichen interpretiert. Wer diese Zeichen nicht als Operatoren nutzen will, sondern diese als Zeichen benötigt, der muss diese mittels „ESCAPE“ entwerten. Der DEFAULTESCCHAR in der Konfigurationsdatei \$ONCONFIG ist das Zeichen „\“ und kann auf jedes Zeichen, das mit einem Byte gespeichert werden kann, gesetzt werden. Im SQL-Statement ist es möglich, mittels Angabe der ESCAPE-Clause ein Zeichen als Escape-Character festzulegen, das vom Default abweicht. Dies soll das folgende Beispiel veranschaulichen:

```
create table testtab ( pathname varchar(255) );

insert into testtab values
    ("/opt/informix_12.10.FC9/tmp/online.log");
insert into testtab values
    ("/opt/informixy12.10.FC9/tmp/online.log");
```

Ergebnis: Nur der Datensatz mit „_“ im Namen vor der 12:

```
SELECT pathname FROM testtab
    WHERE pathname like @%informix#_12%@ ESCAPE @#@;
```

```
pathname /opt/informix_12.10.FC9/tmp/online.log
```

Ergebnis: Beide Datensätze, da „_“ als Wildcard interpretiert wird:

```
SELECT pathname FROM testtab WHERE pathname like @%informix_12%@;
pathname /opt/informix_12.10.FC9/tmp/online.log
pathname /opt/informixy12.10.FC9/tmp/online.log
```

TechTipp: REGEX Datablade – Reguläre Ausdrücke im SQL

In dieser Ausgabe des Newsletters wollen wir die Funktionen zu Regulären Ausdrücken in SQL näher betrachten. Zum Verständnis haben wir jeweils ein Beispiel erstellt, das die Funktionsweise anschaulich darstellen soll.

Hier nochmals die Auflistung der Funktionen, die seit Version 12.10.FC8 in jeder Datenbank zur Verfügung stehen. Voraussetzung ist ein SmartBlobSpace.

- Funktion [regex_match](#) Prüft, ob ein String einen Ausdruck enthält
- Funktion [regex_replace](#) Tausche einen Ausdruck durch einen anderen aus
- Funktion [regex_extract](#) Liefert eine Liste von Treffern zu einem Ausdruck
- Funktion [regex_split](#) Teilt einen String in Teilstrings anhand eines Delimiters
- Prozedur [regex_set_trace](#) Debugging für regex Funktionen
- Funktion [regex_release](#) Versionsinformation zu Regex Funktionen

Die Funktionen sind sowohl für Single-Byte, als auch auf UTF8 Daten nutzbar.

Als Metazeichen in regulären Ausdrücke können verwendet werden:

^	Beginning of line
\$	End of line
	Or Not applicable to basic regular expressions.
[abc]	Match any character enclosed in the brackets
[^abc]	Match any character not enclosed in the brackets
[a-z]	Match the range of characters specified by the hyphen

[<i>cclass</i> :]	Use the character list that is specified by <i>cclass</i> : (nur für Single-Byte) <ul style="list-style-type: none"> • <i>alnum</i> = Uppercase and lowercase alphabetic characters and numbers: [A-Za-z0-9] • <i>alpha</i> = Uppercase and lowercase alphabetic characters: [A-Za-z]
--------------------	--

	<ul style="list-style-type: none"> • blank = Whitespace and tab characters • cntrl = Control characters • digit = Numbers: [0-9] • graph = Visible characters (the alnum class plus the punct class) • lower = Lowercase alphabetic characters: [a-z] • print = Printable characters (the graph class plus whitespace) • punct = Punctuation marks: !"#\$%&'()*+,-./:;<=>?@[\\]^_`{ }~ • space = Whitespace characters: tab, newline, carriage-return, form-feed, and vertical-tab • upper = Uppercase alphabetic characters: [A-Z] • xdigit = Hexadecimal characters: [0-9a-fA-F]
[=cname=]	Substitute the character name that is specified by <i>cname</i> with the corresponding character code. siehe „ Regex character names “.
.	Match any single character.
()	Group the regular expression within the parentheses.
?	Match zero or one of the preceding expression. Not applicable to basic regular expressions.
*	Match zero, one, or many of the preceding expression.
+	Match one or many of the preceding expression. Not applicable to basic regular expressions.
\	Use the literal meaning of the metacharacter. For basic regular expressions, treat the next character as a metacharacter.

Die Wildcards sind identisch denen des „matches“ im SQL.

Komplexe Ausdrücke sind durch die Länge von 32,739 Bytes des LVARCHAR begrenzt.

TechTipp: regex_match() – Suche nach regulären Ausdrücken

Mit der Funktion `regex_match()` kann in regulären Ausdrücken gesucht werden. Durchsucht werden können hierbei Felder die Strings speichern können mit Ausnahme des alten Datentyps TEXT.

Die Syntax:

```
regex_match(  
    str      clob,  
    re       lvarchar,  
    copts    integer DEFAULT 1)  
returns boolean
```

Wobei **str** für den zu durchsuchenden String steht, der vom Datentyp er CHAR, NCHAR, VARCHAR, NVARCHAR, LVARCHAR oder CLOB sein kann.

Der Ausdruck „**re**“ kann wiederum vom Datentyp CHAR, NCHAR, VARCHAR, NVARCHAR oder LVARCHAR sein (not null) und kann einen regulären Ausdruck enthalten (siehe Beispiele).

Das dritte Argument „**copts**“ ist optional und bestimmt den Typ der Suche:

- 0 = Basic regex
- 1 = Default. Extended POSIX regex
- 2 = Basic regex and ignore case
- 3 = Extended POSIX regex and ignore case

Beispiel:

Wir laden uns in ein CLOB die INFORMIX Newsletter des letzten Jahres incl. der Ausgabe Q1_2017 und suchen darauf mit regulären Ausdrücken:

```
create database regex_test in datadbs with log;  
create table documents (  
    txt      varchar(14),  
    doc      clob,  
    tstamp   datetime year to second  
            default current year to second  
) put doc in (sdbds);  
  
insert into documents values ("INFORMIX_NL_201601",  
filetoclob("/tmp/TMP/INFORMIX_NL_201601.txt", "server"),  
current year to second);
```

¼

```

select case
  when regex_match(doc,"Scripts zur Erstellung",3) = @t@
    then "TREFFER_1"
  when regex_match(doc,"DBSpace encryption on Disk",3) = @t@
    then "TREFFER_2"
  when regex_match(doc,"Performance|Encryption",3) = @t@
    then "TREFFER_3"
  else "----"
  end search
  ,txt , tstamp
from documents

```

Das Ergebnis sieht dann wie folgt aus:

search	txt	tstamp
----	INFORMIX_NL_201601	2017-05-18 16:18:54
TREFFER_3	INFORMIX_NL_201602	2017-05-18 16:18:54
TREFFER_1	INFORMIX_NL_201603	2017-05-18 16:18:54
----	INFORMIX_NL_201604	2017-05-18 16:18:54
TREFFER_3	INFORMIX_NL_2016_Q4	2017-05-18 16:18:54
TREFFER_2	INFORMIX_NL_2017_Q1	2017-05-18 16:18:54

In den Dokumenten NL_201602 und NL2016_Q4 ist somit eines der Worte „Performance“ oder „Encryption“ enthalten, im Dokument NL_201603 ist der String „Scripts zur Erstellung“ enthalten und im Dokument NL_2017_Q1 der Ausdruck „DBSpace encryption on Disk“. Da mit der Option 3 gesucht wurde, ist das Ergebnis Case-Insensitiv.

TechTipp: regex_replace() – Austausch von Ausdrücken in Texten

Mit der Funktion regex_replace() können Ausdrücke in Texten ausgetauscht werden. Die Syntax und die Regeln für das Auffinden der Ausdrücke ist bei str, re, limit und copts identisch mit der von regex_match (siehe vorheriger Artikel).

Hinzu kommt der String für den Ersatz (der auch wieder den Suchstring enthalten darf), sowie ein Limit wie viele Ersetzungen vorgenommen werden sollen (default ist 0=unlimited).

```

regex_replace(
  str      clob,
  re       lvarchar,
  rep      lvarchar,
  limit    integer DEFAULT 0,
  copts    integer DEFAULT 1)
returns clob

```

Als zusätzliche Funktionalität sind in den Ausdrücken folgende Zeichen möglich:

&	<p>Reference the entire matched text for string substitution.</p> <p>For example, the statement execute function <code>regex_replace('abcdefg', '[af]', '&')</code> replaces 'a' with '.a.' and 'f' with '.f.' to return: <code>'.a.bcde.f.g'</code>.</p>
\n	<p>Reference the subgroup <i>n</i> within the matched text, where <i>n</i> is an integer 0-9.</p> <p><code>\0</code> and <code>&</code> have identical actions.</p> <p><code>\1 - \9</code> substitute the corresponding subgroup.</p> <p>For example, the statement execute function <code>regex_replace('abcdefg', '[af]', '\0')</code> replaces 'a' with '.a.' and 'f' with '.f.' to return: <code>'.a.bcde.f.g'</code>.</p> <p>For example, the statement execute function <code>regex_replace('abcdefg', '([af])([bg])', '.p1-\1.p2-\2.')</code> replaces 'ab' with <code>'.p1-a.p2-b'</code> and 'fg' with <code>'.p1-f.p2-g.'</code> to return: <code>'.p1-a.p2-b.cde.p1-f.p2-g.'</code>.</p> <p>Not applicable to basic regular expressions.</p>
\	<p>Use the literal meaning of the metacharacter, for example, <code>\&</code> escapes the Ampersand symbol and <code>\\</code> escapes the backslash.</p> <p>For basic regular expressions, treat the next character as a metacharacter.</p>

Besonders interessant ist hier die Option mehrere Untergruppen gefundener Patterns unterschiedlich zu behandeln bzw. zu ersetzen.

Ein einfaches Beispiel zeigt wie man Suchbegriffe im Ergebnis mittels HTML-Tags highlighten kann:

```
execute function regex_replace (
    ©We love INFORMIX, we use INFORMIX, promote INFORMIX.
    INFORMIX is the best database in the world.©,
    ©INFORMIX|database©,
    ©<b>&</b>©);
```

Ergebnis:

```
(expression) We love <b>INFORMIX</b>, we use <b>INFORMIX</b>,
promote <b>INFOR MIX</b>. <b>INFORMIX</b> is the best
<b>database</b> in the world
```

Beispiel zu Untergruppen:

```
execute function regex_replace (
    ©We love INFORMIX !©,©(.*) (.*) (.*) (.*)©,©\3 \1 \2 \4©,0,3);
```

```
(expression) INFORMIX We love !
```

TechTipp: regex_extract() – Extrahieren von Ausdrücken

Mit der Funktion regex_extract() können Ausdrücke aus Texten extrahiert werden. Die Syntax und die Regeln für das Auffinden der Ausdrücke ist bei str, re, limit und copts identisch mit der von regex_match (siehe vorheriger Artikel).

```
regex_extract(
    str      clob,
    re      lvvarchar,
    limit   integer DEFAULT 0,
    copts   integer DEFAULT 1)
returns lvvarchar
```

```
execute function regex_extract (
    ©Informix, der Information Server und InfosSphere sind alle
    vom selben Hersteller©,©Info[a-z]*©,0,3);
```

Ergebnis:

```
(expression) Informix
(expression) Information
(expression) InfosSphere
```

TechTipp: regex_split() – Aufteilen von Texten

Mit der Funktion `regex_split()` können Texte anhand regulärer Ausdrücke aufgeteilt werden. Die Syntax und die Regeln für das Auffinden der Ausdrücke ist bei `str`, `re`, `limit` und `copts` identisch mit der von `regex_match` (siehe vorheriger Artikel).

```
regex_split(
    str      clob,
    re      lvvarchar,
    limit    integer DEFAULT 0,
    copts    integer DEFAULT 1)
returns lvvarchar
```

```
execute function regex_split (
    ©Informix, der Information Server, InfosSphere und viele
weitere Produkte sind von IBM©,
    ©,©
,0,3
);
```

Ergebnis:

```
(expression)  Informix
(expression)   der Information Server
(expression)   InfosSphere und viele weitere Produkte sind von IBM
```

TechTipp: regex_set_trace() – Trace für regex_...() setzen

Mit der Prozedur `regex_set_trace` wird das Trace Level für alle folgenden Aufrufe einer der Funktionen zu `regex_...` gesetzt. Als Parameter werden eine Datei (die für den Datenbankserver schreibbar ist) und das Trace Level übergeben.

Das Trace Level kann zwischen 0 (OFF=default) und 100 gewählt werden. Höhere Trace Level ergeben detailliertere Ausgaben im Trace.

```
execute procedure regex_set_trace('/tmp/rx.log', 42);
```

Ausgabe:

```
14:23:37 ifxregex_regexec: exit
14:23:37 regex_split: str (85) 'Informix, der Information Server, InfosSphere und viele
weitere Produkte sind von IBM'
14:23:37 regex_split: re (1) ','
14:23:37 regex_split: subs[0] [8,9] (1) ','
```

```
14:23:37 regex_split: Return (8) 'Informix'  
14:23:37 regex_split: exit  
14:23:37 regex_split: enter  
14:23:37 regex_split: SET_RETONE  
14:23:37 ifxregex_regexec: enter  
14:23:37 ifxregex_regexec: exit  
14:23:37 regex_split: str (85) 'Informix, der Information Server, InfoSphere und viele  
weitere Produkte sind von IBM'  
14:23:37 regex_split: re (1) ','  
14:23:37 regex_split: subs[0] [23,24] (1) 'o'  
14:23:37 regex_split: Return (23) ' der Information Server'  
14:23:37 regex_split: exit  
14:23:37 regex_split: enter  
14:23:37 regex_split: SET_RETONE
```

TechTipp: regex_release() – Version der Datablades regex

Mit dem Aufruf von `regex_release()` wird die eingesetzte Version des Datablades ausgegeben:

```
(expression) ifxregex release 1.00 (Build 114)
```

```
Compiled on Mon Apr 24 18:09:16 CDT 2017 with:  
IBM Informix Dynamic Server Version 12.10.FC8W2  
gslib-6.00.UC10
```

TechTipp: LATERAL – Derived table mit Referenz zum Aufruf

Wird in einer Abfrage eine „derived table“ genutzt, so kann in der Abfrage, mit der diese Tabelle erstellt wird, nicht auf die anderen Tabellen Bezug genommen werden, die zuvor im selben SQL Statement genutzt wurden.

Beispiel:

```
SELECT d.deptno, d.deptname, empinfo.avgsal, empinfo.empcount
FROM department d,
     TABLE ((SELECT AVG(e.salary) AS avgsal, COUNT(*) AS empcount
              FROM employee e
              WHERE e.workdept=d.deptno
             )) AS empinfo;
```

217: Column (**deptno**) not found in any table in the query (or SLV is undefined).

Ist dies erforderlich, so muss die „derived table“ als LATERAL gekennzeichnet werden:

```
SELECT d.deptno, d.deptname, empinfo.avgsal, empinfo.empcount
FROM department d,
     LATERAL (SELECT AVG(e.salary) AS avgsal, COUNT(*) AS empcount
              FROM employee e
              WHERE e.workdept=d.deptno
             ) AS empinfo;
```

TechTipp: SQL Package Extension DBMS

Die „SQL packages extension“ beinhaltet eine Vielzahl an hilfreichen Erweiterungen zum SQL-Umfang des Datenbankservers.

Im Gegensatz zu den mitgelieferten Datablades muss die SQL Extension explizit in der Datenbank registriert werden mittels:

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

Dabei muss die Datenbank folgende Bedingungen erfüllen, damit die Erweiterungen genutzt werden können:

- Die Datenbank muss mit Transaktionslog betrieben werden.
- Die Datenbank darf nicht im Mode ANSI betrieben werden..

Die folgenden Pakete sind verfügbar:

- DBMS_ALERT
- DBMS_LOB
- DBMS_OUTPUT
- DBMS_RANDOM
- UTL_FILE

Im Paket DBMS_LOB sind z.B. folgende Funktionen enthalten:

APPEND procedure	Ein Large Object zu einem anderen LOB hinzufügen.
COMPARE function	Zwei LOBs vergleichen
COPY procedure	Ein LOB in ein anderes LOB kopieren.
ERASE procedure	Ein LOB löschen.
GETLENGTH function	Die Länge eines LOBs bestimmen.
INSTR function	Die Position des n-ten Vorkommens eines Suchstring ausgeben
READ procedure	Ein LOB auslesen
SUBSTR function	Einen Teil eines LOB ausgeben
TRIM procedure	Ein LOB auf die gewünschte Grösse reduzieren
WRITE procedure	Daten in ein LOB schreiben

TechTipp: DBMS_LOB_GETLENGTH()

Die Funktion DBMS_LOB_GETLENGTH() gibt die Länge eines Large Objects zurück, was der Anzahl der Zeichen entspricht.

TechTipp: DBMS_LOB_SUBSTR()

Die Funktion SUBSTR() bietet die Möglichkeit bei den Datentypen CHAR und VARCHAR einen definierten Bereich auszuschneiden und zurückzugeben. Auf dem Datentyp CLOB kann dies mittels der Funktion „DBMS_LOB_SUBSTR()“ erfolgen.

```
create table if not exists tickets_texts (  
id_ticket    int,  
txt_text     clob  
) put txt_text in (sdbds);
```

```
insert into tickets_texts values (42,filetoclob("/home/kalu/tmp","server"));
```

Der Select:

```
select DBMS_LOB_SUBSTR(txt_text, 200, 20)  
from tickets_texts
```

gibt die ersten 200 Zeichen ab der Position 20 aus.

Termin: "What The Data!?"- Hackathon mit Informix in Hamburg

Unter dem Motto "What The Data!?" findet vom 15.9. bis 17.9.2017 ein Hackathon mit Informix Beteiligung in Hamburg statt.

Der IBM Businesspartner HARTING Deutschland GmbH & Co. KG wird sich dieses Jahr erneut an dem "What The Data!?"-Hackathon der Cybus GmbH mit seinen Mica Boxen beteiligen. Da Informix auf der Analytics Mica eine wichtige Rolle zur Verarbeitung von industriellen Sensordaten spielt, werden die Hackathon Teilnehmer die Möglichkeit bekommen die optimierten Eigenschaften der Informix Zeitreihentechnologie in einer realitätsnahen Umgebung zu testen. Die IBM unterstützt Harting dabei mit vor Ort Expertise zu den Themen IBM Bluemix (Andreas Entgelmeier) und Informix (Alexander Körner).

Cybus bietet für den Hackthon auf ihrem Campus eine ideale Umgebung mit realen, industrieerprobten Hardware- und Softwarekomponenten, damit die Teilnehmer Sensordaten so einmal richtig 'anfassen' können. Für weitere Details und zur Registrierung klicken Sie bitte hier: <https://www.what-the-data.com/>
(es gibt noch ein paar freie Plätze!)

Termin: 70. IUG-Workshop am 24. Oktober 2017 in Hamburg

Nach einer längeren Pause findet im Oktober in Hamburg wieder ein IUG Workshop statt.

Das Thema lautet: Informix - Jetzt erst recht!

Neben vielen interessanten technischen Vorträgen, sowie Präsentationen über die zukünftige Entwicklung bei INFORMIX, sind die IUG Workshops immer ein beliebter Treffpunkt um sich mit anderen Anwendern zu vernetzen. Die Agenda ist derzeit noch nicht ganz finalisiert, so dass es noch die Möglichkeit gibt, dass Sie hier ihre Lösung im Umfeld von Informix vorstellen, und einen Vortrag halten.

Informationen zur Anmeldung, die Möglichkeit sich als Referent vorzustellen, und in einigen Wochen die endgültige Agenda finden sie unter:

http://www.iug.de/index.php?option=com_content&task=view&id=327&Itemid=429

P.S.: Vertreter des Informix Newsletters werden ebenfalls anwesend sein.

Versionsinfo: 12.10.xC9 ist verfügbar

Seit einigen Wochen ist die Version 12.10.xC9 für alle unterstützten Plattformen und Editionen verfügbar. Hierin sind einige Patches enthalten, die Fehler beheben und neue Features verbessern.

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.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, DB2, InfoSphere CDC, DataStage
IBM Analytics
gerd.kaluzinski@de.ibm.com +49-175-228-1983

Martin Fuerderer Informix Development Germany
HCL Technologies Germany GmbH
martin.furderer@hcl.com

Alexander Körner Analytics Architect
IBM Cloud, Industry 4.0 / Internet of Things & Informix
Consulting IT-Specialist
akoerner@de.ibm.com +49 89 4504 1423

Dr. Andreas Weininger Leading Technical Sales Professional
andreas.weininger@de.ibm.com +49-172-7565266

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

(Bodensee nach Start ab FDH)