

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Ablage von Dokumenten in BLOBs.....	2
TechTipp: Monitoring der Logdateien.....	3
TechTipp: Nagios als Überwachungstool für INFORMIX.....	5
TechTipp: IPv6 Adressen.....	9
Versionsinfo: Version 10.00 ist nicht mehr verfügbar.....	9
WebTipp: Rawdevices oder Filesystem und wenn JA, welches ?.....	10
WebTipp: INFORMIX Referenz - DHL in Mexico und viele mehr.....	10
Tipp: Weihnachtsgeschenk für den INFORMIX Administrator.....	11
Anmeldung / Abmeldung / Anmerkung.....	11
Die Autoren dieser Ausgabe.....	12

Aktuelles

Liebe Leserinnen und Leser,

nun hat uns der Winter im Griff.
Bei schönem Wetter ist jede (Daten-)Bank nutzbar. Erst bei schwierigen Situationen zeigt sich was es Wert ist, (sich) auf eine zuverlässige Variante zu setzen.
In diesem Sinne hoffen wir, dass Sie mit Ihren Lieben die Feiertage im Warmen vor dem Kamin geniessen können und den vielen Schnee nur für sportliche Aktivitäten aufsuchen müssen.

**Frohe Festtage ! Feliz Navidad !
Boas Festas ! Merry Christmas !
Joyeux Noël ! Prettige Kerstdagen !**



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.
Ihr TechTeam

TechTipp: Ablage von Dokumenten in BLOBs

Immer öfter werden Dokumente, Bilder, Töne und Anleitungen in der Datenbank abgelegt. Ein Vorteil dabei ist, dass mit der Datenbanksicherung auch die zugehörigen Dateien mit dem selben Stand gesichert werden. Ein weiterer Vorteil besteht z.B. darin, dass mit der Volltextsuche (Excalibur oder BTS) in den Dokumenten über Suchindexe (schnell) und Kontext (kontext sensitiv) gesucht werden kann. Die Ablage der Dokumente kann in der Datenbank in BLOBs oder CLOBs erfolgen. CLOBs dürfen dabei nur Zeichen des aktuellen Zeichensatzes der Datenbank enthalten, BLOBs können beliebige binäre Dateien sein.

Im folgenden Beispiel finden Sie die Syntax für die Verwendung von Spalten vom Typ BLOB und CLOB.

```
create table documents (
doc_id      serial,
docname     char(38),
doc         blob,
description clob
) put doc in (sbdb), description in (sbdb);

insert into documents values (0,"carmen.jpg",
filetoblob("/home/kalu/FOTOS/FOTOS_2010/carmen.jpg","server"),
filetoclob("/home/kalu/DOC/carmen.txt","server")
);

select * from documents
doc_id      1
docname     carmen.jpg
doc         <SBlob Data>
description
Aufgenommen bei der Schulparty 2010 im Bodenseegymnasium.
Zu sehen sind ...
```

Um die Dateien wieder unter eindeutigen Namen zu speichern, kann der „docname“ als Teil des Dateinamens verwendet werden. Das „!“ am Ende des Dateinamens verhindert, dass eine eindeutige ID an den Dateinamen angehängen wird.

Beispiel:

```
select docname, lotofile(doc,"/tmp/"||trim(docname)||"!","server"),
lotofile(description,"/tmp/"||trim(docname)||".txt!","server")
from documents

docname     carmen.jpg
(expression) /tmp/carmen.jpg
(expression) /tmp/carmen.jpg.txt

docname     carina.jpg
(expression) /tmp/carina.jpg
(expression) /tmp/carina.jpg.txt
```

TechTipp: Monitoring der Logdateien

Bisher werden die historischen Daten „wann wurden wie viele Logs geschrieben“ nicht gespeichert. Wer das „online.log“ archiviert, der könnte hierin ermitteln wie viele Logs je Stunde oder je Tag genutzt werden.

Mittels eines Tasks, der stündlich die Unique-ID des aktuellen Logs notiert, kann die Nutzung der Logs auch in der Datenbank protokolliert werden.

Die zugehörige Tabelle, in der die Werte gespeichert werden, könnte folgende Form haben:

```
create table mon_logs (
    id integer,                -- ID des Task-Calls
    log_id integer,           -- Logid des Current Logs
    log_cnt integer,         -- Logs seit letztem Aufruf
    log_time interval day to minute, -- Zeit seit letztem Aufruf
    rec_time datetime year to second -- Zeitpunkt des Aufrufs
);
```

Um den Aufruf im Task einfach zu gestalten, wurde eine Prozedur erstellt, die die Log-ID, sowie den Zeitstempel in die Tabelle einträgt. Diese sieht im Beispiel so aus:

```
database sysadmin;
CREATE FUNCTION "informix".record_logids(task_id INTEGER, ID INTEGER)
    RETURNING INTEGER
    DEFINE cur_log_id    INTEGER;
    DEFINE cur_rec_time  datetime year to second;
    DEFINE last_log_id   INTEGER;
    DEFINE last_rec_time datetime year to second;
    LET cur_log_id = NULL;
    select log_id, rec_time
    INTO last_log_id, last_rec_time
    FROM sysadmin:mon_logs
    where rec_time = (select max(rec_time) from sysadmin:mon_logs);
    select max(uniqid), current year to second
    INTO cur_log_id, cur_rec_time
    FROM sysmaster:syslogfil;
    INSERT INTO mon_logs VALUES( ID, cur_log_id,
        cur_log_id - last_log_id,
        cur_rec_time - last_rec_time,
        cur_rec_time );
    return 0;
END FUNCTION;
```

Die Prozedur kann separat mit „execute function record_logids(0,0)“ getestet werden. Die Definition des Tasks im Beispiel als unload-Format (ph_task_logid.unl):

```
0|mon_logs|Collect information about database server's logical logs. Only the
maximum logid is collected.|SENSOR|0|mon_logs|create table mon_logs(id
integer,log_id integer,log_cnt integer,log_time interval day to minute,
rec_time datetime year to second);|sysadmin|record_logids||00:00:00||0 01:00:00|
2010-12-12 13:42:00|0|0.0|t|t|t|t|t|t|t|t|405|SERVER|t|0|
```

(Details zum Erstellen eines Tasks siehe INFORMIX Newsletter Mai 2008)

Nun muss noch der Task in die Tabelle ph_task eingetragen werden (als User informix):

```
load from ph_task_logid.unl insert into ph_task
```

Ab diesem Zeitpunkt werden die Daten protokolliert.

Die Auswertung kann dann ganz einfach über eine Abfrage in der Tabelle „mon_logs“ erfolgen:

Beispiel1 - Logs je Stunde:

```
select * from mon_logs
```

Ergebnis:

id	log_id	log_cnt	log_time	rec_time
120	1304		0 01:00	2010-12-13 13:42:27
121	1307	3	0 01:00	2010-12-13 14:42:27
122	1313	6	0 01:00	2010-12-13 15:42:27
123	1342	29	0 01:00	2010-12-13 16:42:27
125	1343	1	0 01:00	2010-12-13 17:42:27
126	1356	13	0 01:00	2010-12-13 18:42:27
127	1368	12	0 01:00	2010-12-13 19:42:27

Beispiel2 – Logs je Tag:

```
select sum(log_cnt) as anz_logs, date(rec_time) as datum
from mon_logs
group by 2 order by 2
```

Ergebnis:

anz_logs	datum
144	09.12.2010
132	10.12.2010
11	11.12.2010
13	12.12.2010
79	13.12.2010
42	14.12.2010

Oft wird uns die Frage gestellt, wie viele Logs für den Datenbankserver eingerichtet werden sollen.

Als Faustformel für einen Betrieb, der auch einen Ausfall des Log Backups für einen Tag verkraftet, lautet die Empfehlung „So viele Logs, dass diese nicht innerhalb eines Tages überschrieben werden“.

Wer von Ihnen weiss, wie viele Logdateien maximal je Tag geschrieben werden und wie viele dies durchschnittlich je Tag sind ?

Das Beispiel oben erlaubt eine exakte Analyse des Logverhaltens der letzten Monate mit SQL-Mitteln und hilft somit bei der Abschätzung der einzustellenden Werte.

TechTipp: Nagios als Überwachungstool für INFORMIX

Das OpenAdminTool (OAT) eignet sich sehr gut um die Aktivitäten innerhalb einer INFORMIX Instanz zu analysieren. Eine Überwachung ob die Instanz überhaupt läuft ist dadurch allerdings nicht möglich. Zudem fragen immer wieder unsere Leser nach der Einbindung von INFORMIX in andere Überwachungstools nach, die auch zur Überwachung des Rechners und des Netzwerks eingesetzt werden. Dies hat uns veranlasst, die Einbindung von INFORMIX in das kostenlose Überwachungstool Nagios näher zu betrachten.

Als Beispiel soll überwacht werden:

- Status der Instanz (OnLine / Qiescent / Single-User / Offline / ...)
- Füllungsgrad der DBSpaces (mit Vorwarnung bei Erreichen eines Limits)

Die Einbindung von INFORMIX in Nagios erfolgt über die Nagios Konfigurationsdatei, in der die Informix Überwachung als weitere Services hinzugefügt werden können.

```
/etc/nagios3/conf.d/localhost_nagios2.cfg:
...
define service{
    use                generic-service
    host_name          localhost
    service_description Informix State
    check_command      check_informix
}

define service{
    use                generic-service
    host_name          localhost
    service_description Informix Dbspaces
    check_command      check_ifx_spaces
}
```

Bei den PlugIns müssen die Aufrufe definiert werden, die in der Konfig angegeben sind:

```
/etc/nagios-plugins/config/informix.cfg:
# 'check_informix' command definition
define command{
    command_name      check_informix
    command_line      /usr/lib/nagios/plugins/check_informix
    --warning=' $ARG1$, $ARG2$, $ARG3$' --critical=' $ARG4$, $ARG5$, $ARG6$'
}

define command{
    command_name      check_ifx_spaces
    command_line      /usr/lib/nagios/plugins/check_ifx_spaces
    --warning=' $ARG1$, $ARG2$, $ARG3$' --critical=' $ARG4$, $ARG5$, $ARG6$'
}
```

Das Script „check_informix“ wurde als Shellprogramm erstellt, die Überwachung des Füllungsgrades der DBSpaces wurde im Beispiel als Abfrage auf die Datenbank „sysmaster“ implementiert.

Das Script zur Überwachung des Status könnte folgendermassen aussehen:

```

/usr/lib/nagios/plugins/check_informix:
#!/bin/bash
. /home/informix/ifxini

# stderr needs to be closed or redirected,
exec 2>&1

ONSTAT=$(onstat - | grep -v '^$')

if [[ -z $ONSTAT ]]; then
    printf "UNKNOWN:\tNo onstat output for Informix server %s\n" \
        $INFORMIXSERVER
    rc=3
elif echo $ONSTAT | egrep -qi 'On-Line'; then
    printf "OK:\t%s\n" "$ONSTAT"
    rc=0
elif echo $ONSTAT | egrep -qi 'Quiescent'; then
    printf "WARN:\t%s\n" "$ONSTAT"
    rc=1
elif echo $ONSTAT | egrep -qi 'Single'; then
    printf "WARN:\t%s\n" "$ONSTAT"
    rc=1
else
    printf "CRITICAL:\t%s\n" "$ONSTAT"
    rc=2
fi

exit $rc

```

Da es sich hier um ein Shellscript handelt, können weitere Modi einfach ergänzt werden.

Die Übersicht über die überwachten Services hat in Nagios die Form:

Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
Current Load	OK	2010-12-14 09:39:24	22d 18h 8m 53s	1/4	OK - load average: 0.82, 0.98, 1.05
Current Users	OK	2010-12-14 09:40:50	76d 1h 17m 41s	1/4	USERS OK - 2 users currently logged in
HTTP	OK	2010-12-14 09:42:16	7d 17h 42m 48s	1/4	HTTP OK: HTTP/1.1 200 OK - 454 bytes in 0.001 second response time
Informix Dbspaces	OK	2010-12-14 09:38:41	4d 4h 23m 48s	1/4	OK: datadbs:88 Free
Informix State	OK	2010-12-14 09:40:07	8d 1h 42m 19s	1/4	OK: IBM Informix Dynamic Server Version 11.70.UC1 -- On-Line -- Up 01:30:39 -- 300372 Kbytes
SSH	OK	2010-12-14 09:41:33	76d 1h 18m 24s	1/4	SSH OK - OpenSSH_5.1p1 Debian-6ubuntu2 (protocol 2.0)
Total Processes	OK	2010-12-14 09:42:59	69d 12h 22m 0s	1/4	PROCS OK: 251 processes

Das Script zur Überwachung des Füllungsgrades der DBSpaces wertet die Rückgabewerte eines Prozedur-Aufrufes aus, dem die Limits für die Warnung und Critical mitgegeben werden.

```

/usr/lib/nagios/plugins/check_ifx_spaces:
#!/bin/bash
. /home/informix/ifxini

# stderr needs to be closed or redirected,
exec 2>&1

```

```

lim1=25;
lim2=15;
C=`dbaccess sysmaster <<EOF 2>/dev/null
execute procedure check_ifx_space($lim1,$lim2);
EOF`

xx=`echo $C | grep 'flag' | cut -f2 -d ' '`
info=`echo $C | grep '_' | cut -f2 -d '_'`

if [[ $xx -eq 0 ]]; then
    printf "OK:$info"
    rc=0
elif [[ $xx -eq 1 ]]; then
    printf "WARN:$info"
    rc=1
elif [[ $xx -eq 2 ]]; then
    printf "CRITICAL:$info"
    rc=2
else
    printf "CRITICAL:$info"
    rc=3
fi
exit $rc

```

Anbei ein Beispiel für eine SPL, die den Füllungsgrad der DBSpaces ermittelt. Der ROOTDBS, LOGDBS, TempDBSpaces und BLOBSpaces werden hierbei explizit von der Überwachung ausgenommen:

```

create procedure check_ifx_space(limit1 int, limit2 int)
returning int as flag, varchar(255) as _
define xx varchar(255);
define name varchar(255);
define prc_free int;
define flag int;
let xx='';
let name='';
let prc_free='';
let flag = 0;
foreach xx_cur for
select trim(d.name),
    round(100 * sum(d.pagesize*c.nfree)/sum(d.pagesize*c.chksize),0)::char(4)
into name, prc_free
from sysdbspaces d, syschunks c, sysshmvals s
where d.dbsnum = c.dbsnum and d.is_temp = 0 and d.is_sbspace = 0
    and d.is_blobspace = 0 and d.name not in ("rootdbs","logdbs","sbdbs")
group by 1
order by 2
if prc_free < limit1 and flag = 0 then let flag = 1; end if;
if prc_free < limit2 and flag < 2 then let flag = 2; end if;
let xx=xx||" "||trim(name)||":"||prc_free||" Free";
end foreach
return flag, trim(xx);
end procedure

```


Die Überwachung zeigt zu den Services weitere Details, so z.B. wann die letzte Zustandsänderung war und wann der nächste Lauf des Scripts stattfindet:

Service State Information

Current Status:	OK (for 4d 4h 47m 15s)
Status Information:	OK: datadbs:88 Free
Performance Data:	
Current Attempt:	1/4 (HARD state)
Last Check Time:	2010-12-14 10:03:41
Check Type:	ACTIVE
Check Latency / Duration:	0.105 / 0.041 seconds
Next Scheduled Check:	2010-12-14 10:08:41
Last State Change:	2010-12-10 05:19:42
Last Notification:	N/A (notification 0)
Is This Service Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	2010-12-14 10:06:49 (0d 0h 0m 8s ago)

Zusätzlich wird die Historie der Alarme protokolliert:

localhost	Informix Dbspaces	OK	2010-11-18 18:02:32	root	notify-service-by-email	OK: datadbs:89 Free
localhost	Informix Dbspaces	CRITICAL	2010-11-18 08:56:28	root	notify-service-by-email	(Service Check Timed Out)
localhost	Informix Dbspaces	OK	2010-11-06 12:41:46	root	notify-service-by-email	OK: datadbs2:100 Free datadbs:89 Free
localhost	Informix Dbspaces	CRITICAL	2010-11-06 12:27:46	root	notify-service-by-email	(Service Check Timed Out)
localhost	Informix Dbspaces	OK	2010-11-03 10:29:02	root	notify-service-by-email	OK: datadbs2:100 Free datadbs:89 Free
localhost	Informix Dbspaces	WARNING	2010-10-29 10:53:12	root	notify-service-by-email	WARN: datadbs2:20 Free datadbs:89 Free
localhost	Informix Dbspaces	CRITICAL	2010-10-29 10:48:12	root	notify-service-by-email	CRITICAL: datadbs2:0 Free datadbs:89 Free
localhost	Informix Dbspaces	WARNING	2010-10-29 09:53:12	root	notify-service-by-email	WARN: datadbs2:20 Free datadbs:89 Free
localhost	Informix Dbspaces	CRITICAL	2010-10-29 08:53:12	root	notify-service-by-email	CRITICAL: datadbs2:2 Free datadbs:89 Free
localhost	Informix Dbspaces	OK	2010-10-29 08:40:12	root	notify-service-by-email	OK: datadbs:89 Free datadbs2:99 Free
localhost	Informix Dbspaces	CRITICAL	2010-10-29 08:00:12	root	notify-service-by-email	CRITICAL: datadbs2:0 Free datadbs:89 Free
localhost	Informix Dbspaces	CRITICAL	2010-10-28 17:13:40	root	notify-service-by-email	(Service Check Timed Out)
localhost	Informix Dbspaces	OK	2010-10-28 16:39:20	root	notify-service-by-email	OK: datadbs2:57 Free datadbs:89 Free
localhost	Informix Dbspaces	CRITICAL	2010-10-28 16:29:20	root	notify-service-by-email	CRITICAL: datadbs2:11 Free datadbs:89 Free

Setzen Sie Nagios zur Überwachung von INFORMIX ein und haben selbst Scripts dazu geschrieben ?

Senden Sie uns diese zu !

Wir würden diese gerne in einem Repository den Lesern des INFORMIX Newsletters zur Verfügung stellen.

TechTipp: IPv6 Adressen

Auf allen von INFORMIX unterstützten Plattformen können seit Version 10.00.xC4 neben den Ipv4 Adressen (die 32-Bit Länge haben) auch Ipv6 Adressen (mit 120 Bit Länge) verwendet werden.

INFORMIX prüft immer zuerst, ob das Betriebssystem Ipv6 unterstützt und versucht dann dieses zu nutzen. Ist dies erfolglos, so wird nach Ipv4 gesucht.

Bei der Verwendung in der Datei \$INFORMIXDIR/etc/sqlhosts bieten Ipv6 Adressen die selben Möglichkeiten wie die Ipv4 Adressen, also auch (*hostname).

Teilweise ist die Namensauflösung über Ipv6 aber noch relativ langsam, so dass diese explizit ausgeschaltet werden sollte.

Um die Suche nach Ipv6 Adressen explizit auszuschalten, gibt es die Möglichkeit dies am Server und am Client festzulegen.

Um für alle Instanzen an einem Datenbankserver oder an einem INFORMIX Client die Nutzung der Ipv6 Adressen zu deaktivieren, muss eine leere Datei mit Namen \$INFORMIXDIR/etc/IFX_DISABLE_IPV6 erstellt werden. Diese Datei muss zumindest für den Benutzer INFORMIX Leseberechtigung haben. Der Inhalt wird nicht ausgewertet.

Soll die Nutzung der Ipv6 Adressen nur für eine Instanz deaktiviert werden, so kann vor dem Start der Instanz mit „oninit“ die Umgebungsvariable „IFX_DISABLE_IPV6=yes“ gesetzt werden. Um die Ipv6 Adressen für einen Client zu deaktivieren, muss die Variable „IFX_DISABLE_IPV6=yes“ vor dem Start der Applikation gesetzt werden.

In beiden Fällen ist es ohnehin ratsam, das Name Caching des Betriebssystems zu aktivieren (/usr/sbin/nscd), unabhängig von der Nutzung der Datenbank.

Versionsinfo: Version 10.00 ist nicht mehr verfügbar

Die seit 30.09. aus dem regulären Support genommene Version 10.00 wurde auch aus dem Downloadbereich entfernt, so dass dieses Produkt nicht mehr auf den üblichen Wegen verfügbar ist.

Als nächstes reguläres Release steht die Freigabe der Version 11.70.xC2 auf dem Plan, was für das Frühjahr 2011 geplant ist.

Bis da hin werden wir Sie über ggf. verfügbare W-Releases informieren.

WebTipp: Rawdevices oder Filesystem und wenn JA, welches ?

Die Frage nach der optimalen Einrichtung von Plattenplatz für INFORMIX beschäftigt seit langer Zeit Generationen von Administratoren. Rawdevices, EXT2, EXT3, EXT4, JFS oder andere Optionen kommen hierbei immer wieder ins Gespräch. Auch RAID wird in diesem Zusammenhang gerne diskutiert.

Zu diesem Thema gibt es im Web einen sehr guten Artikel von Art S. Kagel, der die Hintergründe verdeutlicht und klare Empfehlungen gibt:

<http://informix-myview.blogspot.com/2010/07/new-journaled-file-system-rant.html>

WebTipp: INFORMIX Referenz - DHL in Mexico und viele mehr

Mexico war schon in früheren Jahren eine INFORMIX Hochburg. In vielen Bereichen wie Fernsehen, Telekommunikation und Retail war und ist INFORMIX die gesetzte Datenbank. Hier finden Sie zwei von vielen interessanten Videos zu INFORMIX in der Welt:

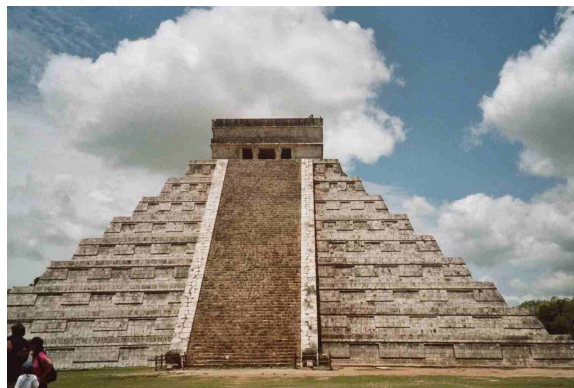
<http://www.youtube.com/watch?v=kZoiGptEqO8>

<http://www.youtube.com/watch?v=3Kgj-U7Ozds&NR=1>

Dies könnte nicht zuletzt ein weiterer Grund sein, das wunderschöne und interessante Land Mexico zu besuchen. Aus eigener Erfahrung kann ich sagen: Es lohnt sich !

P.S.: An unsere Leserinnen und Leser in Mexcio:

¡ Feliz Navidad y Próspero Año Nuevo !



Tipp: Weihnachtsgeschenk für den INFORMIX Administrator

Damit die Überwachung der Datenbank auch ohne Präsenz Vorort erfolgen kann, und der INFORMIX Administrator im Kreise seiner Lieben feiern kann, könnte ein Smartphone als Weihnachtsgeschenk hilfreich sein. Die Alarmer des OAT können so überall kontrolliert 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.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.cursor-distribution.de/index.php/aktuelles/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 +49-175-228-1983

Martin Fuerderer IBM Informix Entwicklung, München
 IBM Software Group, Information Management
martinfu@de.ibm.com

Sowie unterstützende Teams im Hintergrund.

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

Fotonachweis:	Winter (Lindau):	Gerd Kaluzinski
	Mexico (Chichén Itzá):	Carmen Kaluzinski
	OAT am Handy:	Carmen Kaluzinski