

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: SYSMaster:SYSSHMVAlS.....	2
TechTipp: IDS 11 – Migration.....	3
TechTipp: Btree Scanner (Teil 2).....	3
Hinweis.....	11
Termine.....	11
Anmeldung / Abmeldung.....	12
Wir über uns: Die Redaktion stellt sich vor – Sandor Szabo.....	12
Die Autoren dieser Ausgabe:.....	12

Aktuelles

Liebe Leserinnen und Leser,

die Begeisterung für INFORMIX ist voll in Fahrt. Neue und bestehende Kunden setzen auf die Version IDS 11 und deren Features. Im Consulting hatte dies bereits die Folge, dass wir einen Aufruf an externe Consultants machen mussten, die uns bei der Bewältigung der Anfragen unterstützen können.

Mit dieser Ausgabe des INFORMIX Newsletters wollen wir Ihnen weitere wertvolle Tipps geben wie Sie den Einsatz Ihrer INFORMIX Datenbank noch effizienter gestalten zu können.

Am Umfang der Tipps sehen Sie, dass wir den Sommer (auch wenn man die Tage zählen konnte) genutzt haben um neue Kraft zu tanken. Nun steht Ihnen das gesamte Team wieder gestärkt zur Verfügung.

Ihr TechTeam

TechTip: SYSMASTER:SYSSHMVALS

Die Datenbank sysmaster beinhaltet u.a. die Pseudo-/SMI-Tabelle sysshmvals, in der eine Reihe von Parametern stehen, die dem Administrator nützlich sein können.

Folgende Werte können hier abgefragt werden:

sh_mode	5	Man kommt aber nur im Mode=5 ran
sh_boottime	1188366395	Zeitpunkt der Instanzstarts
sh_pfclrtype	1188366395	Zeitpunkt des letzten onstat -z
sh_curttime	1188380756	Aktuelle Zeit
sh_bootstamp	6912384	Intern Timestamp beim Start
sh_stamp	6946188	Letzte vergebene Timestamp
sh_mainlooptcb	1168161760	
sh_sysflags	4097	
sh_maxchunks	32766	Limit Anzahl der Chunks
sh_maxdbspaces	2047	Limit Anzahl der Dbspaces
sh_maxuserthreads	32767	Limit der Userthreads
sh_maxtrans	32767	Limit maximal offener TAs
sh_maxlocks	20000	LOCKS (onconfig)
sh_maxlogs	16	LOGFILES (onconfig)
sh_nbufs	5000	BUFFERS (onconfig)
sh_pagesize	2048	Pagesize in Bytes
sh_nlrus	16	LRUS (onconfig)
sh_maxdirty	60.000000000000	LRU_MAX_DIRTY (onconfig)
sh_mindirty	50.000000000000	LRU_MIN_DIRTY (onconfig)
sh_ncleaners	8	CLEANERS (onconfig)
sh_longtx	0	Counter Long Transactions
sh_optstgbsnum	0	
sh_cpflag	0	
sh_rapages	42	RA_PAGES (onconfig)
sh_rathreshold	12	RA_THRESHOLD (onconfig)
sh_lastlogfreed	74	ID des zuletzt freigegebenen Log
sh_rmdlktout	60	DEADLOCK_TIMEOUT (onconfig)
sh_narchivers	0	
sh_maxpdqpriority	0	MARPDQPRIORITY (onconfig)
sh_curmaxcons	1	Max. Anzahl Connections seit Start
sh_ovlmaxcons	20	Max. Anzahl Connctions seit Initialize

Die Werte, die als Integer angegeben sind, wie z.B. Profile-Clear-time können mittels der Funktion dbinfo('UTC_TO_DATETIME', int_time) als DATETIME ausgegeben werden.

TechTipp: IDS 11 – Migration

Die Migration auf IDS11 kann wie gewohnt sowohl mittels Datenmigration (dbexport/dbimport, unload/load, ...) oder InPlace erfolgen. Bei der Planung der Migration InPlace sollte darauf geachtet werden, dass die neue Version zusätzliche Ressourcen benötigt. Folgende Ressourcen sollten daher vor einer Migration InPlace geprüft werden, bzw. bei einer Neuinstallation gleich in der Planung berücksichtigt werden:

- Der ROOTDBS bekommt mit sysadmin und eventuell sysha weitere Datenbanken die je nach Nutzung der neuen Features erheblichen Platzbedarf haben können. Der ROOTDBS sollte daher bei einer Migration InPlace noch einige MB als Reserve haben. Bei Neuinstallationen sollte der ROOTDBS nicht zu knapp kalkuliert werden. Eine Größe von mindestens 500 MB ist daher für Neuinstallationen empfehlenswert.
- Soll das Feature des Systemtracing genutzt werden, dann empfiehlt es sich die stark wachsenden Tabellen der Datenbank sysadmin mittels „alter fragment“ aus dem ROOTDBS in einen eigens dafür angelegten Dbspace zu verschieben.
- Das PHYSLOG sollte nicht zu klein gewählt werden, da die neue Checkpointstrategie hier je nach Einstellungen mehr Platz benötigt. Zudem kann das automatische Checkpointtuning nur dann aktiviert werden, wenn das PHYSLOG nicht zu klein gewählt wurde. Ist das Physlog kleiner als 10MB, dann wird dieses Feature implizit ausgeschaltet. Je nach Aktivität der User ist ein PHYSLOG von 500MB bis zu 2 GB sinnvoll.
- Die Logs sollten nicht zu klein gewählt werden. Auch dieser Parameter kann das automatische Tuning verhindern. Als Mindestwert sollten mindestens 100MB für die Logical Logs verfügbar sein. Mehr Logs erleichtern zudem dem Administrator die Arbeit, sei es um längere Transaktionen ungehindert zu einem erfolgreichen Ende zu bringen oder wenn das Logbackup einmal Probleme machen sollte. Die Größe der einzelnen Logs hängt dabei stark von der Anwendung ab. Werden meist lesende Aktivitäten durchgeführt und kaum Änderungen, die im Log landen, dann können Logfiles mit 1MB sinnvoll sein. Sind viele Schreibaktivitäten im System, dann empfiehlt es sich Logs zwischen 10 MB bis zu 100MB zu erstellen. Die Abwägung der Übersichtlichkeit gegen granulare Sicherung der Logfiles fließt hier in die Entscheidung ein.

TechTipp: Btree Scanner (Teil 2)

Im vergangenen Teil dieses Artikels wurden Ihnen die grundlegenden Konzepte des Btree Scanners erläutert. In diesem zweiten und letzten Teil beschäftigen wir uns mit dem neuen „ALICE“ Scan Modus, sowie weiteren Tipps zur Konfiguration und Monitoring des Btree Scanners.

ALICE

... ist etwa nicht der Name der Unbekannten von nebenan, sondern eine in der IT-Welt so wunderbar gebräuchlichen Abkürzungen: „Adaptive Linear Index Cleaning“.

Wie im letzten Teil dieses Artikels bereits erwähnt, gilt es bei einem Index einen Kompromiss zu finden, der einerseits für einen möglichst kleinen, kompakten (und daher performanten) Indexbaum sorgt, gleichzeitig aber die Reorganisation eines Index auf ein Minimum reduziert (und somit den „regulären“ Datenbankbetrieb nicht stört).

Was sind die Vorteile von ALICE Scan gegenüber „Leaf Scan“ und „Range Scan“?

Der „Leaf Scan“ zeichnet sich durch ein Überprüfen sämtlicher Index Pages aus. Wenn ein Index reorganisiert werden soll, werden alle Index Pages in den Bufferpool geladen und dort gegebenenfalls reorganisiert. Nachteile dieses Verfahrens – bei großen Indexbäumen – sind das hohe I/O Aufkommen durch Lesen der Index Pages und die CPU Belastung durch Überprüfung aller Index Pages im Bufferpool, ob sich dort gesetzte delete flags befinden.

Der „Range Scan“ versucht den Nachteilen des „Leaf Scans“ entgegen zu treten, indem nicht mehr alle Pages eines Indexbaums gelesen werden, sondern nur noch ein Teil (Bereich). Dazu werden Positionen im Indexbaum vermerkt, an denen sich ein delete flag an der niedrigsten, bzw. höchsten „Position“ befindet.

Ob Sie nun tatsächlich vom Range Scan profitieren, zeigt Ihnen ein Blick in die Ausgabe des onstat -C range:

```
$ onstat -C range
```

```
IBM Informix Dynamic Server Version 11.10.FC1      -- On-Line -- Up 00:00:13 --
29696 Kbytes
```

```
Btree Cleaner Info
```

```
Cleaning Range Statistics
```

```
=====
```

Partnum	Key	Low	High	Size	Saving
0x00100005	1	37	37	48	100.0 %
0x00100006	2	7	13	32	81.2 %
0x00100014	1	16	16	40	100.0 %
0x0010001b	1	13	14	24	95.8 %
0x001000b5	2	7	7	16	100.0 %
0x001000c3	1	16	18	56	96.4 %
0x001000d2	1	9	9	24	100.0 %
0x00100112	1	1	1	4	100.0 %

Die letzte Spalte (savings) berechnet, wie viele Indexpages im Verhältnis zu einem Leaf Scan beim Reorganisieren des Index nicht gelesen werden mussten. Höhere Werte besagen, dass die Instanz sich Arbeit sparen konnte. Sicherlich werden Sie aber auch Indices finden, bei denen die Einsparungen nicht so deutlich oder gar überhaupt nicht vorzufinden sind. Genau hier tritt dann die Schwachstelle des Range Scans zu Tage: Wenn beispielsweise der erste (niedrigster) und letzte (höchster) Eintrag im Index als zu löschen markiert sind, hat dies zur Folge, dass auch hier wieder der komplette Index verarbeitet werden muss – und das, obwohl vielleicht in dem Bereich dazwischen kein weiterer Indexeintrag als gelöscht markiert ist. Ob dies nun wahrscheinlich ist oder nicht: genau dieses Szenario, nämlich den kompletten Index zu lesen, sollte das Konzept des Range Scans verhindern.

Hier schafft ALICE Abhilfe.

Wird ALICE als Scanverfahren konfiguriert, wird der Index für die Reorganisation in mehrere Abschnitte unterteilt. Jeder dieser Abschnitte wird durch ein Bit in einer eigens eingerichteten Bitmap repräsentiert (siehe Abbildung 1). Wird in solch einem Abschnitt ein delete flag entdeckt, wird das für den Abschnitt entsprechende Bit gesetzt. Das bedeutet, dass der Btree Scanner nun gezielt einen Abschnitt des Indexbaums reorganisieren kann. Bezogen auf das obige Negativbeispiel mit dem Range Scan, würde in diesem Fall nur der erste und der letzte Abschnitt des Index weiter untersucht – Abschnitte dazwischen blieben unberührt.

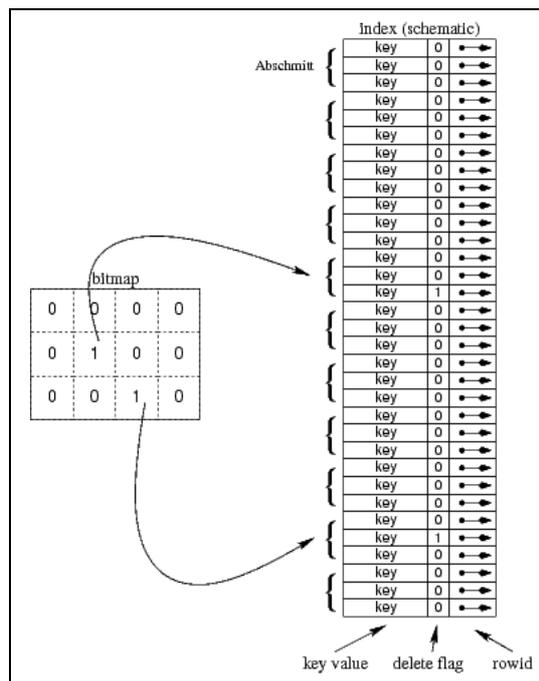


Abbildung 1 Bitmapverwaltung eines Index

ALICE können Sie mit

```
$ onmode -C alice <mode>
```

online aktivieren oder folgenden Parameter in der \$ONCONFIG setzen:

```
BTSCANNER num=1,threshold=50000,alice=<mode>
```

Beachten Sie bitte, dass sich Range Scan und ALICE ausschließen, daher taucht in der BTSCANNER Zeile der Parameter `rangesize` nicht auf.

Aus Performancegesichtspunkten ist es nun interessant, wie groß die Abschnitte sind, die durch der Bitmap verwaltet werden. Je größer die Bitmap, desto mehr Abschnitte können verwaltet werden. Oder anders: Je mehr Bits in der Bitmap, desto kleiner die Abschnitte, in die ein Index unterteilt wird.

Die initiale Größe der Bitmap wird durch **mode** bestimmt, wobei hier ein ganzzahliger Wert zwischen 1 und 12 zulässig ist (0=OFF). Je größer die Indices in einer Instanz sind, desto höher sollte der Wert sein. Als Faustregel gilt, dass wenn die Indizes in einer Instanz kleiner 1GB sind, ein Wert von 6 ausreichend ist (Default ist 0). Ein größerer **mode** bedeutet eine größere Bitmap. Eine weitere Komponente, welche die Größe der Bitmap bestimmt, ist die Größe des Index. Es lässt sich also nicht vorhersagen, wie viel Platz eine Bitmap im Memory beansprucht, ohne die Größe eines Index zu kennen. Welchen Speicherbedarf die Bitmaps einnehmen, können Sie in der Ausgabe von `onstat -C alice` sehen:

```
$ onstat -C alice
```

```
IBM Informix Dynamic Server Version 11.10.FC1      -- On-Line -- Up 00:29:49 --
326656 Kbytes
```

```
Btree Cleaner Info
```

```
ALICE Cleaning Statistics
=====
```

```
System ALICE Info: Mode =      8, Eff = 30 %, Adj =      5
```

Partnum	Mode	BM_Sz	Used_Pg	Examined	Dirty_Pg	#I/O	Found	Eff	Adj
0x00100002	8	64	4	3	1	1	1	100.0 %	1
0x00100112	8	64	2	0	0	0	0	0.0 %	0
0x002000b6	8	64	89	0	0	0	0	0.0 %	0
0x002000b8	8	64	41	40	3	4	1	25.0 %	1
0x002000c7	8	64	23	5	0	0	0	0.0 %	0

Die Spalte "Partnum" listet wieder die einzelnen Indices auf. „Mode“ zeigt Ihnen den aktuellen ALICE-Mode der jeweiligen Partition. Die dritte Spalte „BM_Sz“ gibt Ihnen die Größe der Bitmap in Byte an.

Autonomic Computing

Das spannende an ALICE ist das „A“ im Namen („Adaptive“): ALICE verfügt über einen Automatismus, der die Größe der Bitmap anpassen kann. Dazu werden statistische Daten zu einem Index bzw. Indexbereichen gesammelt.

Um herauszufinden, ob eine Bitmap vergrößert werden soll, wird ein Verhältnis aus Anzahl der Indexpages zu Pages mit gesetztem delete-flag innerhalb eines Abschnitts berechnet (effectiveness: *eff*). Wird ein definierter Wert unterschritten, wird die Bitmap vergrößert.

Dieser zunächst seltsam anmutende Algorithmus verfolgt folgende Idee: Wenn bei der Reorganisation eines Index festgestellt wird, dass in einem Index-Abschnitt nur wenige gesetzte delete-flags gefunden wurden, bedeutet das, dass „viele“ pages aus dem Index-

Abschnitt umsonst gelesen wurden (=> +IO / +CPU). Das heißt, die Effektivität *eff* dieses Index-Abschnitt-Scans sinkt.

Nach anfänglichen <Adj> Scans wird dann nach jedem Scan die gerade gemessene Effektivität mit einer vorgegebenen Mindesteffektivität „*ALICE_EFF*“ verglichen. *ALICE_EFF* hat eine Standardgröße von 30%, kann jedoch dynamisch durch

```
onmode -C eff n
```

verändert werden.

Wird *ALICE_EFF* unterschritten, gilt für den gesamten Index der nächsthöhere ALICE-Mode.

Ein weiterer Parameter, „*ALICE_ADJ*“, steuert, ab wann überhaupt ein autonomes Anpassen des ALICE-Modes vorgenommen werden soll. Ein Index muss zunächst eine definierte Anzahl an Reorganisationen vollzogen haben, so dass ein Wechsel in den nächsthöheren ALICE-Mode in Betracht gezogen werden soll. *ALICE_ADJ* hat einen Standardwert von 5 (in aktuellen IDS 10/11 Versionen).

Auch *ALICE_ADJ* ist dynamisch konfigurierbar:

```
onmode -C adj n
```

ALICE_EFF und *ALICE_ADJ* können durch folgende Beispielzeile in der ONCONFIG dauerhaft übernommen werden:

```
BTSCANNER num=1,priority=low,threshold=50000,alice=8,adj=50,eff=10
```

Die aktuelle ALICE Konfiguration können Sie mit `onstat -C alice` erfahren:

```
$ onstat -C alice
```

```
IBM Informix Dynamic Server Version 11.10.FC1      -- On-Line -- Up  
00:29:49 -- 326656 Kbytes
```

```
Btree Cleaner Info
```

```
ALICE Cleaning Statistics  
=====
```

```
System ALICE Info: Mode =      8, Eff =  30 %, Adj =      5
```

```
[...]
```

Bitte beachten Sie, dass die Angaben für „Eff“ und „Adj“ aufgrund eines Defektes im `onstat` derzeit falsch angegeben werden können.

ALICE_EFF und *ALICE_ADJ* sind wohl nur in Ausnahmesituationen zu verändern und eher für Analysen im Supportbereich gedacht.

Eine automatische Verringerung des ALICE Modes findet übrigens nicht statt.

Monitoring

Durch die drei Reorganisationsverfahren – insbesondere durch ALICE – bieten sich vielfältige Parameter, die überwacht werden können. Wenn ein Index reorganisiert wird, stellen sich unterschiedliche Fragen, zu denen onstat Antworten liefern kann:

„Welcher Index wird aktuell reorganisiert?“

```
$ onstat -C [prof]
```

```
IBM Informix Dynamic Server Version 11.10.FC1      -- On-Line -- Up 05:22:26 --
256000 Kbytes
```

```
Btree Cleaner Info
```

```
BT scanner profile Information
```

```
=====
```

```
Active Threads                1
Global Commands                2000000    Building hot list
Number of partition scans      250
Main Block                    0x0000000011907fec8
BTC Admin                     0x0000000011858e870
```

BTS info	id	Prio	Partnum	Key	Cmd	Yield N
0x119080250	0	High	0x00000000	0	40	
Number of leaves pages scanned						44
Number of leaves with deleted items						31
Time spent cleaning (sec)						0
Number of index compresses						14
Number of deleted items						1423
Number of index range scans						0
Number of index leaf scans						2
Number of index alice scans						3

Das „Profile“ aus `onstat -C prof` zeigt einen Überblick über die Aktivitäten des Btree Scanners. Unter „Partnum“ finden Sie die partition number des Index, der aktuell reorganisiert wird. Eine partnum = 0x00000000 bedeutet, dass aktuell kein Index reorganisiert wird. Finden Sie dort eine echte partnum (z.B.: 0x0030003F), können Sie folgendermaßen den Namen und Ort des Index finden:

```
select * from sysmaster:systabnames
where hex (partnum) = "0x0030003F";
```

Output:

=====

```
partnum  3145791
dbsname  keyonly
owner    uweber
tabname  coll_idx
collate  en_US.819
```

So wissen Sie, in welcher Datenbank (Spalte „dbsname“) der aktuell reorganisierte Index sich befindet. Der Name des **Index** steht in der Spalte „tabname“ (= Objektname). Hier gibt es nun zwei Möglichkeiten: entweder es handelt hierbei um den tatsächlichen Namen des Index (detached Index), oder Sie finden hier den Namen der Tabelle (attached Index), die den Index beinhaltet. Unter Umständen sehen Sie dem Namen des Objektes nicht an, ob es sich um einen Index oder um eine Tabelle handelt. Ein Blick in den Datenbank-Systemkatalog oder in das Datenbankschema sollte das klären.

Handelt es sich um einen attached index, wissen Sie, dass ein Index in dieser Tabelle reorganisiert wird – allerdings nicht welcher. Die Spalte „Key“ in der Ausgabe von

```
onstat -C prof
```

zeigt Ihnen an, der wievielte Index in der Tabelle gerade in Bearbeitung ist. Die Reihenfolge der Indices wird auf der „partition page“ der Tabelle verwaltet. Es gibt keine Möglichkeit, diese Struktur direkt abzufragen. Die Indexreihenfolge können Sie aber zuverlässig erfahren, wenn Sie entweder die Ausgabe obigen onstats weiter verfolgen bis die Indices aufgelistet werden, oder Sie sich z.B. im dbaccess im Infomenü für Tabellen die Indices betrachten. Beide „Ausgaben“ berichten die Indices in der Reihenfolge, wie die Struktur auf der partition page abgelegt ist.

Warum wird aktuell kein Index reorganisiert, obwohl der Threshold erreicht wird?

Der Btree Scanner überprüft nur von Zeit zu Zeit die Hot-List. Die Ausgabe von onstat -g ath zeigt Ihnen, wann der Btree-Scanner das nächste Mal die Hot-List schaut, ob es zu reorganisierende Indices gibt:

```
$ onstat -g ath
```

```
IBM Informix Dynamic Server Version 11.10.FC1      -- On-Line -- Up
02:27:17 -- 256000 Kbytes
```

Threads:

tid	tcb	rstcb	prty	status	vp-class	name
*2	118f7a268	0	1	IO Idle	3lio	lio vp 0
11	1190b1488	0	1	sleeping secs: 1	1cpu	sm_discon
13	1190de2a8	11858e058	2	sleeping secs: 1	1cpu	aslogflush
14	1190fe028	11858e870	1	sleeping secs: 64	1cpu	
btscanner_0						
*30	119786600	1185900b8	3	sleeping secs: 1	1cpu	onmode_mon
*37	11974d528	11858f8a0	1	sleeping forever	1cpu	dbWorker1

Der Status des Btree-Scanner Threads "btscanner_0" gibt an, wann die nächste Überprüfung der Hot-List ansteht (hier: 64 Sekunden). Das Überprüfungsintervall ist aber nicht statisch: Wenn zum Beispiel die Instanz gerade erst gestartet wurde, ist das Intervall 5 Sekunden. Sollte bei der nun anstehenden Überprüfung der Hot-List keine Einträge gefunden werden, wird das Intervall um 5 Sekunden erhöht. Finden sich in der nächsten Überprüfung wieder keine Einträge, wird das Intervall um weitere 5 Sekunden erhöht usw. Die Erhöhung des Intervalls wird aber nicht unendlich fortgesetzt. Das Intervall kann maximal 180 Sekunden betragen.

Wird hingegen bei einer Überprüfung der Hot-List ein zu reorganisierender Index gefunden, wird das Intervall auf 5 Sekunden zurück gesetzt und das Spiel beginnt von vorne.

Welches Verfahren wird benutzt?

Es können durchaus unterschiedliche Reorganisationsverfahren benutzt werden, auch wenn Sie z.B. ALICE konfiguriert haben. ALICE und der „Range Scan“ setzen voraus, dass es sich bei dem zu reorganisierenden Index um einen detached Index handelt. Handelt es sich um einen attached index, kommt nur der „Leaf Scan“ in Frage. Welches Verfahren für den derzeit verarbeiteten Index benutzt wird, können Sie auch durch `onstat -C prof` überprüfen.

Wie groß ist der Index?

`oncheck -pT <datenbank>:<tabelle>`

(alternative kann auch die partition number des Index angegeben werden.)

Im Falle von detached Indices finden Sie in dieser Ausgabe für jeden Index einen separaten Bericht:

TBLspace Usage Report for stores7:uweber.customer

Type	Pages	Empty	Semi-Full	Full	Very-Full
Free	3				
Bit-Map	31				
Index	121626				
Data (Home)	0				
Total Pages	121660				

Unused Space Summary

Unused data slots	0
Unused bytes per data page	88
Total unused bytes in data pages	0

Home Data Page Version Summary

Version	Count
0 (current)	0

Index Usage Report for index i_lname_fname on stores7:uweber.customer

Level	Total	Average No. Keys	Average Free Bytes
1	1	3	1936
2	3	18	1333
3	55	46	270
4	2533	46	235
5	119034	46	226
Total	121626	46	226

Zum einen ist hier zu sehen, wie viele Pages für den Index (-Extent) allokiert wurden (hier: 121626 pages), zum anderen erfahren Sie etwas über die Struktur des Indexbaums, wobei Letzteres hier weniger von Interesse ist.

Sollte es sich um einen attached index handeln, findet sich in dieser Ausgabe kein eigener Bericht zum Index. Stattdessen finden Sie die Anzahl der allokierten Pages im Tabellenreport.

Hinweis

Die Plandaten für **End Of Service** für die IDS Versionen 7.31, 9.40 und 10.00 wurden veröffentlicht:

- **IDS 7.31:** 30. September 2009
- **IDS 9.40:** 30. April 2009
- **IDS 10.0:** 30. September 2010

Die Ausführliche Liste mit Verfügbarkeit seit, EndOfMarketing und EndOfService steht unter: <http://www-306.ibm.com/software/data/support/lifecycle>

Termine

04./05. Oktober 2007

Das **Informix Partner Camp** findet im Rahmen des IBM Partner Camp, in Frankfurt statt. Die Einladung dazu ging bereits als Mailing vor längerer Zeit raus. Unter den folgenden Links finden Sie nochmals wichtige Informationen zusammengefasst:

www.ibm.com/de/partnercamp

http://www.iug.de/veranstaltung/partner_tref07/partner_camp.htm

09.Oktober 2007

IUG Stammtisch in Hamburg

10.Oktober 2007

46. IUG-Workshop in Hamburg

Thema: [Datenbanken und Entwicklungswerkzeuge](#)

Anmeldung / Abmeldung

Der Newsletter wird ausschliesslich an angemeldete Adressen verschickt. Die Anmeldung erfolgt, indem Sie das Subject „ANMELDUNG“ an ifmxnews@de.ibm.com senden. Im Falle einer Abmeldung senden Sie dies ebenfalls an diese Adresse.

Das Archiv der bisherigen Ausgaben finden Sie zum Beispiel unter:

http://www.iug.de/Informix/techn_news.htm

<http://www.informix-zone.com/informix-german-newsletter>

http://www.nsi.de/index.php?option=com_content&task=view&id=36&Itemid=87

<http://www.bytec.de/prd/sft/ibm/ibm-nlarchiv.php>

Wir über uns: Die Redaktion stellt sich vor – Sandor Szabo

Gründungsmitglied des Newsletters



Sandor Szabo

Manager IBM Informix Database Development

IBM Certified Professional

SWG Information Management Services Germany

sandor.szabo@de.ibm.com

Sandor Szabo ist ehemaliger Informix Mitarbeiter, der auch in den vergangenen Jahren immer die „Informix-Flagge“ aus Überzeugung ganz hoch gehalten hat.

Sandor Szabo führt die IBM Informix Entwicklungsmannschaft in München. Er hat Informix Produkte seit 1987 entwickelt. Die Hauptfokus-Gebiete von Sandor sind IBM Informix Dynamic Server unter Linux, Backup & Restore und Hochverfügbarkeit.

Die Autoren dieser Ausgabe:

Uwe Weber IT Specialist for Informix Dynamic Server
uwe.weber@de.ibm.com +49-89-4504-1461

Gerd Kaluzinski IT-Specialist Informix Dynamic Server und DB2 UDB
gerd.kaluzinski@de.ibm.com +49-175-228-1983

Sandor Szabo Manager IBM Informix Database Development
sandor.szabo@de.ibm.com +49-89-4504-1429

Thomas Simoner OEM Sales Manager, Informix und DB2
thomas.simoner@de.ibm.com +49-0160-907-59034

Andreas Legner IBM Informix Advanced Support
andreas.legner@de.ibm.com