

## Willkommen zum „IBM Informix Newsletter“

### Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Bereinigung von „Pending Alter“ bei Tabellen.....	2
TechTipp: PAM – Remote – Users (error 950).....	3
TechTipp: Create Compression Dictionary before Compress.....	3
TechTipp: Optionen des ONSTAT (onstat -g cluster).....	4
TechTipp: dbimport -exists.....	4
TechTipp: ONCONFIG – deprecated parameters.....	5
TechTipp: Sind meine Statistiken aktuell ?.....	7
TechTipp: TimeSeries – Rolling Window Containers.....	8
TechTipp: TimeSeries – GetNextNonNull() mit Beispiel „Interpolation“.....	9
TechTipp: TimeSeries – GetNextValid().....	10
TechTipp: TimeSeries – GetPreviousValid().....	10
TechTipp: TimeSeries Loader – TSL_Flush() abgekündigt.....	10
TechTipp: TimeSeries Loader – TSL_Commit().....	10
TechTipp: INFORMIX in der Cloud - „Informix hosted“.....	11
TechTipp: Extendable Chunks – aus Fehlern lernen.....	12
TechTipp: SPL – Datentyp „List“ als Parameter einer Funktion (*G1*).....	13
TechTipp: Ermittlung des Platzbedarfs für Tabellen und Indices (*G2*).....	14
Anmeldung / Abmeldung / Anmerkung.....	16
Die Autoren dieser Ausgabe.....	16

### Aktuelles

Liebe Leserinnen und Leser,

das Thema INFORMIX ist inzwischen fest verankert in den Bereichen IoT / Industrie 4.0 / Cloud / Watson. Damit ist auch die Redaktion in all diesen Abteilungen tätig und trifft sich leider nicht mehr so oft wie bisher (was sich im Sommer Dank der Biergärten wieder bessern wird). Wie unschwer an der Themenliste zu erkennen ist, kommt das Feature TimeSeries immer öfter in Projekten zum Einsatz.

Vielen Dank an die Einsender der Gastbeiträge !  
Tipps aus der Praxis für die Praxis sind immer besonders wertvoll.



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

## TechTipp: Bereinigung von „Pending Alter“ bei Tabellen

Mit Version 12.10 wurde in der Datenbank „sysadmin“ ein Task eingeführt, mit dem „pending alter tables“ bereinigt werden können.

Zum Zustand „pending alter“ kommt es dann, wenn ein SQL-Befehl „alter table ...“ von der Instanz ausgeführt werden kann, ohne dass der Inhalt der Tabelle auf eine neue Struktur kopiert werden muss. Dies hat den Vorteil, dass auch auf sehr grossen Tabellen eine Änderung in Bruchteilen von Sekunden ausgeführt werden kann, und damit das System nicht lange durch die Sperre der Tabelle blockiert ist. Der Nachteil besteht darin, dass bei jedem Zugriff auf die Daten von Platte für die betroffenen Pages die Datenstruktur angepasst werden muss (so lange diese nicht neu geschrieben werden), was einen kleinen Overhead beim Lesen bedeutet. Informix in der aktuellen Version erlaubt bis zu 254 derartige Änderungen, bevor die Tabelle einmal komplett neu gespeichert wird.

Der Task „table update\_ipa“, der in Version 12.10 eingeführt wurde, ermöglicht es nun die Pages von Tabellen, die geändert wurden, in kleinen Transaktionen in das aktuelle Format zu überführen, ohne dass die Tabelle hierfür gesperrt werden muss.

Welche Tabellen im Zustand „pending alter“ sind, lässt sich ab Version 12.10 aus der Datenbank sysmaster schnell ermitteln, da hierfür zusätzliche Spalten eingeführt wurden.

Die Abfrage auf der Datenbank sysmaster könnte folgendermassen aussehen:

```
select n.dbsname as database, n.tabname as table, h.pta_totpgs as pending_pg
from sysptnhdr h, systabnames n
where h.partnum = n.partnum
and h.pta_totpgs > 0
order by 1,2
```

Ergebnis im Test:

database	stores
table	test
pending_pg	1141

Der „oncheck -pT stores:test liefert ebenfalls diese Information:

```
Home Data Page Version Summary
Version                Count
0 (oldest)            1141
1 (current)           51
```

Die Bereinigung kann nun mittels des Aufrufs eines Tasks durchgeführt werden:

```
database sysadmin;
EXECUTE FUNCTION task("table update_ipa", "test", "stores");
```

Anschliessend liefern die Abfragen keine Pages mehr, die sich im „pending alter mode“ befinden.

## TechTipp: PAM – Remote – Users (error 950)

In der Praxis kommt es immer häufiger vor, dass die Benutzer, die auf das Datenbanksystem zugreifen, über LDAP/PAM authentifiziert werden. Ist LDAP im Betriebssystem eingerichtet, werden in aktuellen Versionen von INFORMIX diese Benutzer von der Datenbank authentifiziert, indem die Abfrage an das Betriebssystem gestellt wird. Dabei weiss die Datenbankinstanz nicht unbedingt, ob ein Benutzer lokal oder mittels LDAP authentifiziert wird. Anders verhält es sich bei verteilten Transaktionen oder der Verbindung zu einem HDR oder RSS Server. Hierfür ist es notwendig die Benutzer, die solche Verbindungen nutzen dürfen, in der Tabelle „sysauth“ der Datenbank „sysuser“ einzutragen, um nicht den Fehler -950 zu erhalten.

**-950 User username is not known on the database server.**

Ein solcher Eintrag hat die Form:

```
insert into sysauth
  values (<username>, NULL,
         <remote_dbservername>, <hostname>);
```

Um auf der lokalen Instanz der Benutzerin „carmen“ den Zugriff vom Informixserver „ifx42“ auf dem Host „kalu.de.ibm.com“ aus zu erlauben, ist somit folgender Eintrag notwendig:

```
insert into sysauth values ("carmen", NULL,
                           "ifx42", "kalu.de.ibm.com");
```

## TechTipp: Create Compression Dictionary before Compress

Wird das „Storage Optimization Feature“ für eine Tabelle angewandt, so wird implizit aus den ersten gelesenen Datensätzen das „Compression Dictionary“ erstellt. Erst danach startet die eigentliche Komprimierung der Daten.

Dieser Schritt lässt sich vom eigentlichen Aufruf der Komprimierung trennen, indem das Dictionary zuvor erstellt wird. Dies ist nur sinnvoll, wenn die Tabelle bereits genügend Datensätze enthält, die repräsentativ für den Datenbestand sind.

Der Aufruf für die Beispieltabelle „test\_compress“ in der Datenbank „stores“ lautet:

```
execute function
  task("table create_dictionary",
       "test_compress", "stores", "informix");
```

Anschliessend kann die Tabelle anhand dieses Dictionarys komprimiert werden:

```
execute function
  task("table compress", "test_compress", "stores", "informix");
```

## TechTipp: Optionen des ONSTAT (onstat -g cluster)

Eine recht unbekannt Option des Onstat ist die Anzeige des aktuellen Status der Hochverfügbarkeit. Diese kann mittels „onstat -g cluster“ ausgegeben werden.

Beispiel:

```
IBM Informix Dynamic Server Version 12.10.FC7 -- On-Line (Prim)
```

```
Primary Server:test1
```

```
Current Log Page:289,1038
```

```
Index page logging status: Enabled
```

```
Index page logging was enabled at: 2015/07/01 17:13:22
```

Server	ACKed Log (log, page)	Applied Log (log, page)	Supports Updates	Status
test5	289,1038	289,1038	Yes	SYNC(SDS),Connected,Active
test7	289,1038	289,1038	Yes	SYNC(SDS),Connected,Active
test2	0,0	0,0	No	ASYNC(HDR),Disconnected,Off
test3	0,0	0,0	No	ASYNC(RSS),Disconnected,Defined

Neben der aktuellen Position im Transaktionslog, die zur Synchronisation genutzt wurde, ist zu sehen, ob die Remote-Server auch schreibend genutzt werden können (updatable). Im Beispiel sind sowohl der HDR-Server als auch der RSS-Server derzeit Offline, beide SDS-Server sind aktiv und sind im Modus „updatable“ definiert.

## TechTipp: dbimport -exists

Dieser Parameter ermöglicht einen dbimport in eine bereits existierende, leere Datenbank. Für herkömmliche Datenbanken ist dieser Parameter nicht sinnvoll, da eine Ergänzung einer bestehenden Datenbank um weitere Tabellen nicht möglich ist, und der dbimport die leere Datenbank ohnehin anlegen würde. Anders verhält es sich bei Tenants, die im Bezug auf dbexport/dbimport wie eine Datenbank behandelt werden. Hier wird die leere Datenbank mittels des Tasks „TenantCreate“ erstellt, und anschliessend mittels dbimport -exists mit Tabellen und Daten des Tenants gefüllt.

## TechTipp: ONCONFIG – deprecated parameters

Wer bei einer Migration die bisherige Konfiguration als Template nimmt, der sollte einen Blick auf die Liste der abgekündigten, oder bereits entfernten Konfigurationsparameter werfen, und diese dann aus dem Template entfernen. Hierunter fallen Parameter wie PHYSDBS oder NUMAIOVPS, die durch andere Parameter oder Aufrufe ersetzt wurden. Um Ihnen einen Überblick zu verschaffen, haben wir die aktuelle Liste hier mit aufgenommen. Die Information, ab welcher Version der Parameter nicht mehr fortgeführt wurde, ist zudem ein guter Hinweis bei Migrationen.

Parameter	Deprecated in	Discontinued in	Explanation
DRAUTO		9.3	You must manually transition from HDR secondary to standard mode.
DRNODE		12.10.xC1	This parameter is not granular enough for high-availability clusters and the secondary servers in such clusters.
FAST_RESTART_CKPT_FUZZYLOG		11.10.xC1	RTO_SERVER_RESTART configuration parameter eliminates fuzzy checkpoints, and it uses interval checkpoints instead.
FAST_RESTART_PHYSLOG		11.10.xC1	RTO_SERVER_RESTART configuration parameter eliminates fuzzy checkpoints, and it uses interval checkpoints instead.
JDKVERSION	11.5	12.10.xC1	Discontinued.
JVPHOME	11.5	TBD	Deprecated.
JVPJAVAHOME	11.5	TBD	Deprecated.
JVPJAVALIB	11.5	TBD	Deprecated.
JVPJAVAVM	11.5	TBD	Deprecated.
LBU_PRESERVE		9.3	onarchive utility is out of support.
LOGSMAX		9.3	The fixed-size logfile array was replaced with a log list that can be added dynamically with the DYNAMIC_LOGS

LRU_MAX_DIRTY	10	12.10.xC1	Replaced by BUFFERPOOL
LRU_MIN_DIRTY	10	12.10.xC1	Replaced by BUFFERPOOL
LRUPOLICY		12.10.xC1	Replaced by AUTO_LRU_TUNING
LRUS	10	12.10.xC1	Replaced by BUFFERPOOL
NOAGE	9.3	12.10.xC1	Replaced by VPCLASS
NOFUZZYCKPT		11.10.xC1	RTO_SERVER_RESTART configuration parameter eliminates fuzzy checkpoints, and it uses interval checkpoints instead.
NUMAIOVPS	9.3	12.10.xC1	Replaced by VPCLASS
NUMCPUVPS	9.3	12.10.xC1	Replaced by VPCLASS
OPCACHEMAX		12.10.xC1	Optical storage is not supported since the optical subsystem was discontinued.
OPTICAL_LIB_PATH		12.10.xC1	Optical storage is not supported since the optical subsystem was discontinued.
PHYSDBS	11.10 .xC1	11.50.xC1	You must move the physical log with the server online.
RA_PAGES	12.10 .xC1	TBD	Replaced by AUTO_READAHEAD
RA_THRESHOLD		12.10.xC1	Replaced by AUTO_READAHEAD
SPINCNT	7.3	12.10.xC1	Use the MULTIPROCESSOR configuration parameter to specify whether the database server performs locking that is suitable for a multiprocessor computer.
STAGEBLOB		12.10.xC1	Optical storage is not supported since the optical subsystem was discontinued.

Die Liste und zusätzliche Informationen finden Sie unter:

[https://www.ibm.com/support/knowledgecenter/SSGU8G\\_12.1.0/com.ibm.mig.doc/ids\\_mig\\_288.htm](https://www.ibm.com/support/knowledgecenter/SSGU8G_12.1.0/com.ibm.mig.doc/ids_mig_288.htm)

## TechTipp: Sind meine Statistiken aktuell ?

In den meisten Systemen werden die Statistiken der Datenbank, die dem Optimizer als Basis der Entscheidungen dienen, periodisch im Cron aktualisiert (bzw. mit einem Task der Datenbank sysadmin).

Wer überprüfen will, wie der aktuelle Status der Statistiken in seiner Datenbank ist, kann dies mit einer einfachen Abfrage ermitteln (ab Version 11.5):

```
select "Statistics Aktuell:" as flag, 1 as order,
       count(*) as anzahl
from systables
where tabtype = "T" and tabid > 99
and ustlowts >= current year to hour - 24 units hour
UNION
select "Statistics this week:" as flag, 2 as order,
       count(*)
from systables
where tabtype = "T" and tabid > 99
and ustlowts < current year to hour - 24 units hour
and ustlowts >= current year to hour - 7 units day
UNION
select "Statistics Old:" as flag, 3 as order,
       count(*)
from systables
where tabtype = "T" and tabid > 99
and ustlowts < current year to hour - 7 units day
order by order
```

Das Ergebnis der Abfrage zeigt z.B. folgende Ausgabe:

flag	order	anzahl
Statistics Aktuell:	1	2138
Statistics this week:	2	712
Statistics Old:	3	3

## TechTipp: TimeSeries – Rolling Window Containers

Die Ablage von Daten in Zeitreihen erfolgt in Containern. Ein Container kann sich über einen oder mehrere DBSpaces erstrecken. Da ein Container nur eine bestimmte Anzahl an Pages aufnehmen kann, ist es in vielen Fällen sinnvoll „Rolling Window Container“ zu erstellen, bei denen nur Daten einer bestimmten Zeitperiode in einem Container gespeichert werden. Werden Daten zu einer neuen Periode eingetragen, so wird hierfür implizit ein weiterer Container erstellt. Diese Container können wiederum auf mehrere DBSpaces verteilt werden.

Die Erstellung eines solchen Rolling Window Containers wird anhand des folgenden Beispiels erklärt:

Beispiel:

```
EXECUTE PROCEDURE TSContainerCreate (  
  'performance_rolling',  
  'tsdbs',  
  'performance_data1',  
  50000,  
  5000,  
  '2016-01-01 00:00:00.00000'::datetime year to fraction(5),  
  'month',0,0,  
  'ts_001dbs,ts_002dbs,ts_003dbs',  
  0,  
  20000,2000  
);
```

Der Container **“performance\_rolling”** wird im DBspace **“tsdbs”** erstellt. Er kann Daten vom Row-Type **“performance\_data1”** aufnehmen. Für die Zeitreihen im Container wurde eine Extent Size von **50'000** und eine Next Size von **5'000** definiert (analog der Erstellung von Tabellen).

Die Initialisierung des Containers erfolgte zur Timestamp 1. Januar 2016 **“2016-01-01 00:00:00.00000”**. Die Window Container werden in den DBSpaces **“ts1dbs”, “ts2dbs”, “ts3dbs”** erstellt und umfassen jeweils das Intervall eines Monats (**month**) der Daten. Die Flags **0, 0** nach dem Zeitintervall geben an, dass es keine Begrenzung der Anzahl der aktiven Container und ShadowContainer gibt. Die **0** nach der Liste der DBSpaces gibt an, dass keine Container nach einer bestimmten Zeit automatisch gelöscht werden sollen.

Für die Basisdaten im tsdbs wurde eine Extent Size von **20'000** und eine Next Size von **2'000** angegeben.

Shadow Container sind Container, die aus den aktiven Containern (Anzahl kann definiert werden), herausgenommen wurden, aber noch nicht gelöscht sind. Details dazu in einem der kommenden Newsletter.



## TechTipp: TimeSeries – GetNextNonNull() mit Beispiel „Interpolation“

Die TimeSeries, die zur effizienten Ablage von Sensordaten dienen, bieten eine Vielzahl an Funktionen zur ebenso effizienten Abfrage. Nachdem wir bereits vor einiger Zeit (im August 2015) die Funktionen „GetOrigin()“ und „GetFirstElementStamp()“ vorgestellt haben, setzen wir nun diese Reihe fort.

Den Anfang macht die Funktion GetNextNonNull(), die aus einer Zeitreihe nach einem angegebenen Startzeitpunkt den ersten eingetragenen Wert zurückliefert. Bei regulären Zeitreihen müssen nicht immer für alle periodischen Zeitabstände Messwerte vorhanden sein, somit kommt es in der Realität vor, dass sporadisch Werte fehlen. Um durchgehende Charts z.B. für die Temperatur zu erhalten, werden oft fehlende Werte mittels Interpolation ergänzt. Das folgende Beispiel zeigt, wie hierbei aus den vorhandenen Werten die Lücken als Mittelwert bei einer regulären Zeitreihe ergänzt werden können:

```
create procedure interpolate()
returning datetime year to second as time,
       decimal(9,2) as orig_val,   decimal(9,2) as pre_val,
       decimal(9,2) as post_val,   decimal(9,2) as result
define elem measure_row;
define next_elem measure_row;
define pre_value decimal(9,2);    define post_value decimal(9,2);
define value decimal(9,2);        define i int;
let i = 0;
foreach EXECUTE FUNCTION Transpose ((
  select data
  from measures_ts)::timeseries(measure_row),
  "2016-08-18 01:00:00", "2016-08-18 06:00:00", 32)
into elem
if elem.value1 is null
then
  if i=0 then continue foreach; end if
  let next_elem =
    GetNextNonNull((select data
                    from measures_ts)::timeseries(measure_row),
                    elem.data_dtm);
  let post_value = next_elem.value1;
  let value = (pre_value + post_value) / 2;
else let i = 1;
  let pre_value = elem.value1; let post_value = elem.value1;
  let value = elem.value1;
end if
return elem.data_dtm,elem.value1,pre_value,post_value,value with resume;
end foreach
end procedure
;
```

Optional kann als zweites Argument noch der Spaltenname aus dem Datensatz (im Beispiel „data“) angegeben werden, um das nächste Element auszugeben, das in dieser Spalte der Reihe einen Wert besitzt, der nicht „null“ ist.

### **TechTipp: TimeSeries – GetNextValid()**

Die Funktion GetNextValid() hat eine ähnliche Funktionsweise wie die zuvor vorgestellte Funktion GetNextNonNull(). Allerdings wird hier nicht anhand der Werte in der Row entschieden, ob ein Wert belegt ist, sondern allein anhand der Tatsache, ob eine Zeitmarke vorhanden ist. Daher findet diese Funktion Anwendung auf irregulären TimeSeries, um hier den nächsten Eintrag nach einem gegebenen Zeitpunkt auszugeben. Als Argumente werden eine Zeitreihe und ein Zeitpunkt erwartet.

### **TechTipp: TimeSeries – GetPreviousValid()**

Die Suche nach dem nächsten, gültigen Eintrag ist auch für die zeitlich zuvor in der TimeSeries stehenden Werte möglich. Die Funktion „GetPreviousValid()“ gibt das Element aus, das vor dem angegebenen Zeitpunkt als letztes in der TimeSeries steht. Als Argument muss wie auch bei den anderen Funktionen dieses Bereichs eine Zeitreihe und eine TimeStamp angegeben werden.

### **TechTipp: TimeSeries Loader – TSL\_Flush() abgekündigt**

Die Funktion TSL\_Flush(), die Werte, die mittels TSL\_Put() eingelesen wurden, in einem einzigen Schreibvorgang auf Platte kopiert hat, wurde abgekündigt. Stattdessen sollte der Befehl TSL\_Commit() verwendet werden, der zusätzliche Optionen bietet.

### **TechTipp: TimeSeries Loader – TSL\_Commit()**

Die Funktion TSL\_Commit() schreibt die mittels TSL\_Put() eingelesenen Werte in die TimeSeries auf Platte. Wird nur die TimeSeries und die Spalte angegeben, so werden bei irregulären Zeitreihen neue Werte zu Zeitpunkten, die bereits in der TimeSeries vorhanden sind, nicht überschrieben, sondern der neue Wert wird mit einem Offset von 0,00001 Sekunden eingefügt. Sollen die Werte explizit überschrieben werden, so muss als drittes Argument das Flag „5“ eingetragen werden (Flag „1“ steht für Anhängen). Zusätzlich kann ein vierter Parameter angegeben werden, der das Commit Intervall angibt.

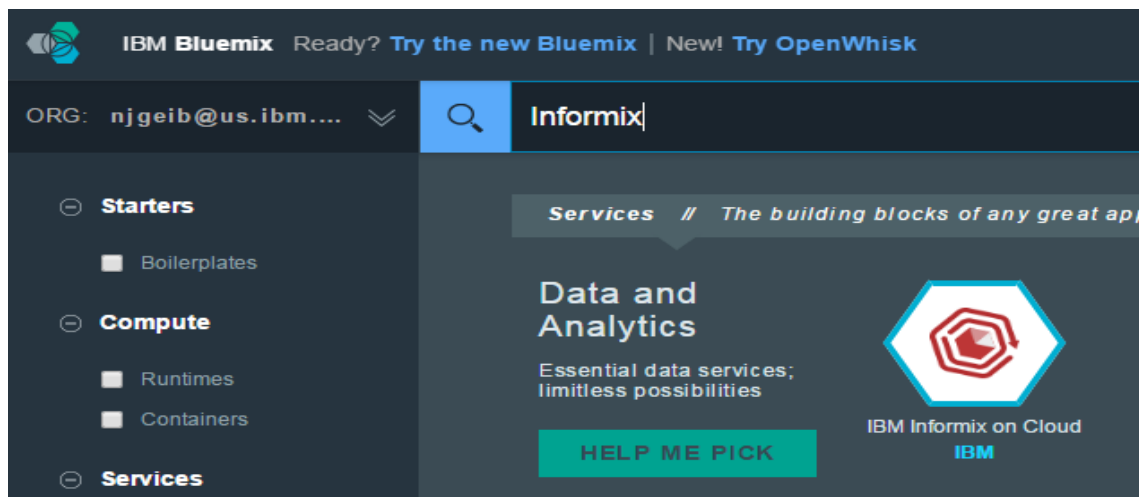
Im folgenden Beispiel werden somit die Daten überschrieben, und nach jeweils 100'000 Werte werden diese mittels Commit in einer Transaktion geschrieben:

```
execute procedure TSL_Commit(  
    @meters_data_irr_ts|meter_data@,5,100000  
);
```

## TechTipp: INFORMIX in der Cloud - „Informix hosted“

Informix steht als vorinstallierte und für OLTP Anwendungen optimierte Installation in der IBM Cloud bei „IBM SoftLayer“ zur Verfügung. Sie haben hierbei neben dieser vorinstallierten Instanz die Option, weitere Instanzen auf dieser Plattform zu erstellen, oder die Konfiguration der Instanz nach den individuellen Bedürfnissen anzupassen.

Das Angebot ist zu finden unter IBM Bluemix:



Es stehen vier Größen der Installation zur Auswahl:

Size & Informix Plan	Small	Medium	Large	XL
<b>Nodes</b>	Virtual Private	Virtual Private	Virtual Private	Bare Metal
<b>Cores</b>	2x2.0 GHz	4x2.0 GHz	8x2.0 GHz	12x2.4 GHz
<b>Memory</b>	8GB	16GB	32GB	128GB
<b>Storage</b>	100 GB 500 GB SAN 100 GB @ 500 IOPS	100 GB 1 TB SAN 100 GB @ 1200 IOPS	100 GB 2 TB SAN 100 GB @ 1600 IOPS	2 x 800GB SSD @RAID1 (~800GB) 8 x 1.2TB SSD @RAID 10 (~3.5TB)
<b>Network</b>	1 Gbps Public & Private Uplinks			10 Gbps Redundant Public & Private Uplinks
<b>OS</b>	CentOS			

Die Instanz der Grösse „S“ mit 2 Cores und 8 GB Memory reicht in vielen Fällen für die Entwicklung oder den ersten „Rollout“ einer Applikation aus.

Die Instanz der Grösse „XL“ mit 12 Cores und 128 GB Memory deckt den Bereich ab, in dem viele Instanzen, die als Produktionsserver bei Kunden die Hauptapplikation tragen, zu finden sind.

Auf die beliebten Kleidergrössen „XS“ (als Kleidung beliebt bei den Töchtern) und „XXL“ (... kein Kommentar ...) wurde bewusst verzichtet, da die Grösse „XS“ der DeveloperEdition entspricht und daher meist lokal zu Entwicklung und Test genutzt wird. Wer die Grösse „XXL“ benötigt, der sollte Kontakt zu IBM Sales aufnehmen, um hier die optimal passende Installation zu erarbeiten.

Das Angebot mit Preisen ist zu finden unter:

<https://console.eu-gb Bluemix.net/catalog/services/informix-on-cloud/>

Als Betriebssystem ist CentOS 7 (derzeit 7.2.1511) installiert. Bei der Informix Instanz handelt es sich um eine Advanced Enterprise Edition der Version 12.10.FC7W2AEE.

Als User sind im Default die User „root“ und „informix“ vorhanden. Bei beiden wird eine sofortige Änderung des Passwortes (auf einen sicheren Standard) verlangt, so dass dies nicht mehr dem Default entspricht.

## TechTipp: Extendable Chunks – aus Fehlern lernen

Seit Version 11 gibt es die Option, dass Chunks erweiterbar sind. Der Task, mit dem dies eingerichtet wird, und wie ggf. eine manuelle Erweiterung und Spezifizierung der Limits vorgenommen werden kann, wurde im Informix Newsletter Januar 2016 ausführlich vorgestellt. Was mit Version 12 neu hinzu kam ist, dass die Datenbankinstanz aus Fehlern lernt und nur einmal „gegen die Wand fährt“, dies daraufhin aber nicht wieder versucht. Gemeint ist, dass ein Chunk, der als extendable markiert wurde, diese Markierung verliert, falls beim Erweitern des Chunks das Problem auftrat, dass nicht genügend Plattenplatz vorhanden war. Dies verhindert, dass bei jedem neuen Statement, das Daten einfügen würde, die Instanz wieder auf das selbe Problem stösst

Die Fehlermeldung hierzu hat folgende Form:

**Extend chunk 42 failed. Its extendable flag will be removed.**

Wurde neuer Plattenplatz zur Verfügung gestellt, so muss mittels Aufruf eines Tasks das Flag erneut gesetzt werden, dass sich der Chunk vergrössern darf.

## TechTipp: SPL – Datentyp „List“ als Parameter einer Funktion (\*G1\*)

An dieser Stellen wollen wir Ihnen den Gastbeitrag eines Lesers präsentieren. Die hier vorgestellte Lösung der Parameterübergabe bei Stored Procedures ist sehr trickreich und ermöglicht es eine Prozedur zu erstellen, obwohl die Anzahl der Parameter unbekannt ist. Hierbei wird als Argument eine Liste unbestimmter Länge übergeben.

```
CREATE PROCEDURE sp_list_parameter2(p_param1 LIST( ROW(partkey
CHAR(20), partvalue INT) NOT NULL ))
    RETURNING VARCHAR(80) as value;

DEFINE p_info VARCHAR(80);
DEFINE p_partkey VARCHAR(20);
DEFINE p_partvalue INT;

set debug file to "tmp.out";
trace on;

    FOREACH SELECT *
        INTO p_partkey, p_partvalue
        FROM TABLE(p_param1)
        LET p_info = TRIM( p_partkey )||" = "||p_partvalue;
        RETURN p_info WITH RESUME;
    END FOREACH

END PROCEDURE
DOCUMENT "Ein Beispiel für LIST als Aufrufparameter"
WITH LISTING IN "./sp_list_parameter2.log";
```

Aufruf:

```
CALL sp_list_parameter2(LIST{ROW(@breite@,123),ROW(@hoehe @,321),
ROW(@laenge@,142)});
```

Ergebnis:

```
value breite = 123
value hoehe = 321
value laenge = 142
```

Der Autor, Fedor Schönhals, kann auf einen fundierten Erfahrungsschatz zurückgreifen. Er arbeitet seit langer Zeit mit Informix und ESQL/C und 4GL. Wer ihn persönlich sprechen will, der sollte eine der Informix Veranstaltungen (IUG, Bootcamp, ...) besuchen, auf denen er immer ein gerngesehener Gast ist.

## TechTipp: Ermittlung des Platzbedarfs für Tabellen und Indices (\*G2\*)

Die Frage danach, wie viel Platz die Daten und die zugehörigen Indices in den DBSpaces belegen, hat sich wohl jedem Administrator schon einmal gestellt. Hierzu gab es bereits einige Annäherungen, allerdings nie eine exakte Betrachtung. Die folgende Abfrage, die wir als Gastbeitrag erhalten haben, berücksichtigt dass Systemtabellen nicht mitgezählt werden dürfen und bietet zudem eine sehr übersichtliche Ausgabe der Grössen an.

```
Database sysmaster;
SELECT
    name as database,
    FORMAT_UNITS(size_data,©b©) as datasize,
    FORMAT_UNITS(size_idx,©b©) as indexsize,
    FORMAT_UNITS(size_total,©b©) as totalsize
FROM (
    SELECT c.name[1,30],
        (SELECT SUM(b.npused*b.pagesize) size_data
        FROM informix.systabnames a, informix.sysactptnhdr b
        WHERE a.partnum = b.partnum
        AND a.dbsname = c.name
        AND nrows > 0) size_data,

        (SELECT SUM(b.npused*b.pagesize) size_idx
        FROM informix.systabnames a, informix.sysactptnhdr b
        WHERE a.partnum = b.partnum
        AND a.dbsname = c.name
        AND nrows = 0) size_idx,

        (SELECT SUM(n.pe_size*b.pagesize) size_total
        FROM informix.systabnames a, informix.sysptnhdr b,
            informix.sysptnext n
        WHERE a.partnum = b.partnum
        AND a.partnum = n.pe_partnum
        AND a.dbsname = c.name
        AND a.dbsname != DBINFO(©dbspace©,a.partnum)
        AND a.dbsname != "system" ) size_total

    FROM informix.sysdatabases c,
        OUTER (informix.systabnames d,
            OUTER informix.sysactptnhdr e)
        WHERE c.name = d.dbsname AND d.partnum = e.partnum AND
        bitval(e.pn_flags, "0x4") = 0 GROUP BY 1,2,3,4
)
ORDER BY size_total DESC;
```

Die Ausgaben erfolgen in einer sinnvollen Einheit je Datenbank, wobei einzeln die Werte für Daten, Indices und die gesamte Datenbank ausgegeben wird. Die Abfrage wurde im Hinblick auf die Performance optimiert, so dass diese auch auf Systemen mit sehr vielen Tabellen und Indices in relativ kurzer Zeit das Ergebnis anzeigt ohne grosse Zwischentabellen aufbauen zu müssen.

Auf unserem Testsystem zeigt die Abfrage z.B. folgende Werte:

```
database    kalu_timeseries_test
datasize    1.39 TB
indexsize   485.80 GB
totalsize    1.89 TB
```

```
database    sysadmin
datasize    15.2 MB
indexsize   6.41 MB
totalsize    24.9 MB
```

```
database    sysutils
datasize    3.40 MB
indexsize   1.39 MB
totalsize    6.25 MB
```

```
database    sysmaster
datasize    2.75 MB
indexsize   328 KB
totalsize    4.68 MB
```

Der Autor, Markus Holzbauer, hat jahrelange Erfahrung als Informix Administrator. Einige von Ihnen werden ihn auch aus der Zeit im Informix TechSupport kennen, in der er viele Fehlersituationen meistern musste, die (hoffentlich) in seiner Tätigkeit als Administrator nie vorgekommen sind und auch nicht auftauchen werden. Weiters ist er als (Co-)Autor von Beiträgen zu Informix in IBM Developerworks in Erscheinung getreten und war in seiner Zeit bei IBM in der Redaktion des Informix Newsletters tätig, bis es ihn zum Fernsehen zog. Wer mehr von Markus lesen will, der findet Anregungen in seinem Blog:

[https://www.ibm.com/developerworks/community/blogs/informix\\_admins\\_blog?lang=en](https://www.ibm.com/developerworks/community/blogs/informix_admins_blog?lang=en)

## 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.iiug.org/intl/deu>

[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.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, DB2, InfoSphere CDC, DataStage  
IBM Analytics  
[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)

(\*G1\*) Gastbeitrag von Fedor Schönhals  
IT Beratung | Softwareentwicklung  
Fliederstr. 30, 74889 Sinsheim  
eMails: [fedor@it-schoenhals.de](mailto:fedor@it-schoenhals.de)

(\*G2\*) Gastbeitrag von Markus Holzbauer  
Solution Architect · IT Infrastructure Solutions  
ProSiebenSat.1 Produktion GmbH  
eMails: [Markus.Holzbauer@P7S1Produktion.de](mailto:Markus.Holzbauer@P7S1Produktion.de)

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Gerd Kaluzinski

(Weihnachtsmarkt Bregenz)