

## Willkommen zum „IBM Informix Newsletter“

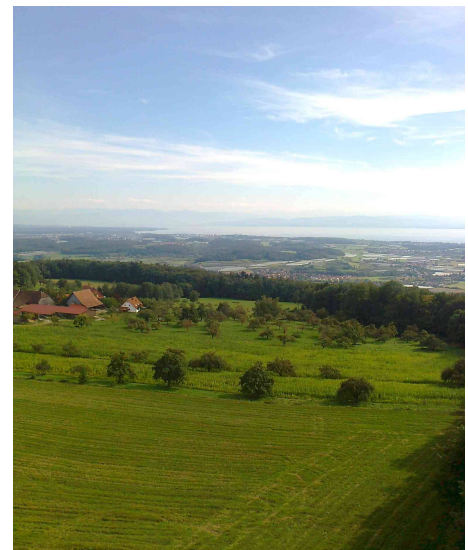
### Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: ONCONFIG – DBSPACETEMP (Teil 1: Parallelisierung).....	2
TechTipp: ONCONFIG – DBSPACETEMP (Teil 2: Größe der DBSpaces).....	3
TechTipp: ONCONFIG – DBSPACETEMP (Teil 3: Transaktionslog).....	4
TechTipp: Environment - FORCE_DDL_EXEC.....	5
TechTipp: SQL Funktion SIGN().....	5
TechTipp: Neue SQL Funktionen.....	6
TechTipp: Built-in Tasks and Sensors (Teil 1).....	8
Referenzen: National Institute for Testing and Evaluation.....	11
Termine: 58. IUG-Workshop in Hamburg.....	11
WebTipp: Analyse Tools im Bundle mit INFORMIX UWE.....	12
Versionsinfo: 11.10.xC3W2 ist verfügbar.....	12
Anmeldung / Abmeldung / Anmerkung.....	12
Die Autoren dieser Ausgabe.....	13

### Aktuelles

Liebe Leserinnen und Leser,

die Sommerferien sind nun auch im Süden vorbei und INFORMIX startet mit einem Workshop zum Thema Warehouse Accelerator in Berlin gleich wieder durch. In dieser Ausgabe haben wir uns auf die neuen SQL Features und das Thema Tasks und Sensors konzentriert. Tasks, die unbekanntes Helfer im Hintergrund, können den Administrator unterstützen und wiederkehrende Aufgaben übernehmen. Sensoren erfassen die Systemzustände und können Alarm schlagen, wenn z.B. der Platz ausgeht oder das regelmäßige Backup nicht gelaufen ist. Nutzen Sie diese Möglichkeiten und lassen Sie die Datenbank für sich arbeiten.



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

Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

## TechTipp: ONCONFIG – DBSPACETEMP (Teil 1: Parallelisierung)

In der ONCONFIG kann unter DBSPACETEMP eine Liste von DBSpaces angegeben werden, in denen temporäre Tabellen abgelegt werden sollen.

Ist dieser Parameter nicht gefüllt, so werden temporäre Tabellen in dem DBSpace aufgebaut, in dem die Datenbank mit dem Befehl „create database“ angelegt wurde. Temporäre Tabellen des Systems, wie sie z.B. beim Backup notwendig sind, werden dann im ROOTDBS abgelegt.

Vor einem Backup prüft das System, ob im DBSPACETEMP noch genug Platz frei ist. Andernfalls bricht der Backup-Aufruf mit einer Fehlermeldung ab.

Sind in der ONCONFIG als DBSPACETEMP oder unter der gleichnamigen Umgebungsvariablen mehrere DBSpaces für temporäre Daten eingetragen, so werden diese parallel genutzt (falls sie den Loggingkriterien entsprechen).

Wird stattdessen nur ein DBSpace mit mehreren Chunks angegeben, so füllen sich die Chunks sequentiell je nach Bedarf.

Beispiel:

Es wurden drei DBSpaces (tmp1dbs,tmp2dbs,tmp3dbs) für temporäre Daten ohne Logging angelegt:

```
IBM Informix Dynamic Server Version 11.70.UC3 -- On-Line -- Up 00:08:32 ...
```

### Dbspaces

address	number	flags	fchunk	nchunks	pgsize	flags	owner	name
571ed818	1	0x60001	1	1	2048	N BA	informix	rootdbs
5a285dd8	2	0x60001	2	1	2048	N BA	informix	logdbs
5a029d48	3	0x60001	3	1	2048	N BA	informix	datadbs
5a03d810	4	0x42001	4	1	2048	N TBA	informix	tmp1dbs
5a029978	5	0x42001	5	1	2048	N TBA	informix	tmp2dbs
5a155088	6	0x42001	6	1	2048	N TBA	informix	tmp3dbs

### Chunks

address	chunk/dbs	offset	size	free	bpages	flags	pathname
571ed988	1	0	100000	56103		PO-B-D	./rootdbs
59c5b2f8	2	0	50000	4947		PO-B-D	./logdbs
59c5b4e0	3	0	50000	41871		PO-B-D	./datadbs
59c5b6c8	4	0	1000	563		PO-B--	./tmp1dbs
59c5b8b0	5	0	2500	1679		PO-B--	./tmp2dbs
59c5ba98	6	0	10000	9563		PO-B--	./tmp3dbs

Wird eine temporäre Tabelle ohne Logging erstellt, so wird sie als eine fragmentierte temporäre Tabelle mit Fragmenten in allen drei DBSpaces erstellt:

```
select t.tabname
from systables t, syscolumns c
into temp xx with no log
```

Wird „with no log“ nicht explizit angegeben und ist auch der ONCONFIG-Parameter „TEMPTAB\_NOLOG“ nicht gesetzt, so werden alle temporären Tabellen mit Logging erstellt. Diese können dann auch nicht in Temporären DBSpaces aufgebaut werden.

Mit einer Abfrage in der Datenbank sysmaster können die temporären Tabellen analysiert werden:

```
select dbsname[1,12] as database, tabname[1,18], n.owner[1,8],
name[1,12] as dbspace, ti_nrows::char(6) as rows ,ti_rowsize::char(6) as rowsz
from sysdbspaces d, systabinfo i, systabnames n
where d.dbsnum = ( trunc ( i.ti_partnum/1048576 ) )
      and i.ti_partnum=n.partnum
      and (bitval(ti_flags,'0x0020')=1 or bitval ( ti_flags, '0x0040' ) = 1 )
```

Ergebnis:

database	tabname	owner	dbspace	rows	rowsz
stores	xx	informix	tmp1dbs	26532	129
stores	xx	informix	tmp2dbs	26532	129
stores	xx	informix	tmp3dbs	26532	129

So lange genug Platz in den TempDbspaces ist, werden diese nahezu gleichmässig gefüllt.

Diese Parallelisierung bedeutet eine bessere Nutzung der Systemressourcen und daher bessere Antwortzeiten (abhängig von der Anzahl der Cores im System). Liegen die TempDBSpaces zudem auf unterschiedlichen Devices, kann auch hier der Durchsatz gegenüber der Nutzung eines DBSpaces mit mehreren Chunks deutlich erhöht werden.

## TechTipp: ONCONFIG – DBSPACETEMP (Teil 2: Größe der DBSpaces)

In früheren Versionen (bis 9.40) sollten möglichst alle DBSpaces, die unter DBSPACETEMP angegeben waren, gleich groß erstellt werden damit es nicht zu Fehlermeldungen kommt.

Sobald der erste der angegebenen DBSpaces voll lief, brach die Verarbeitung mit dem Fehler „dbspace full“ ab.

In aktuellen Versionen werden die DBSpaces so weit als möglich gleichmässig genutzt. Ist ein DBSpace voll, so werden die weiteren Daten auf die verbleibenden DBSpaces verteilt.

Im Beispiel wurde die Rowsize der temporären Tabelle vergrößert. Der DBSpace tmp1dbs ist nun zu klein für 1/3 der temporären Tabelle und läuft während des SQL Statements voll. Die anderen beiden DBSpaces werden weiter gefüllt:

database	tabname	owner	dbspace	rows	rowsz
stores	xx	informix	tmp1dbs	13867	500
stores	xx	informix	tmp2dbs	32865	500
stores	xx	informix	tmp3dbs	32864	500

Statt einer Fehlermeldung „no space left“ wird der volle TempDBSpace bei der Speicherung weiterer Daten nur nicht mehr berücksichtigt. Erst wenn keiner der TempDBSpaces mehr Platz hätte, käme die Meldung „no space left“ und das Statement würde abbrechen (ohne Autoexpand und Autoextend).

## TechTipp: ONCONFIG – DBSPACETEMP (Teil 3: Transaktionslog)

Als DBSPACETEMP können sowohl temporäre DBSpaces (ohne Logging und Backup), als auch DBSpaces mit Logging angegeben werden. Je nachdem, ob ein Statement eine temporäre Tabelle mit oder ohne Logging erstellt, wird diese dann in den entsprechenden DBSpaces, die als DBSPACETEMP angegeben sind, erstellt.

Beispiel:

DBSpace tmp2dbs wurde mit Logging (ohne das Flag -t beim Erstellen) aufgebaut.

```
select t.tabname
from systables t, syscolumns c
into temp xx with no log;

select t.tabname
from systables t, syscolumns c
into temp yy ;
```

Die temporäre Tabelle yy wird nur im DBSpace tmp2dbs, also mit Logging aufgebaut:

database	tablename	owner	dbspace	rows	rowsz
stores	xx	informix	tmp1dbs	39798	129
stores	yy	informix	tmp2dbs	79596	129
stores	xx	informix	tmp3dbs	39798	129

Ist kein temporärer DBSpace mit Logging verfügbar (tmp2dbs wurde für den nächsten Test gelöscht und neu ohne Logging, also mit Flag -t erstellt) und es soll eine temporäre Tabelle mit Logging aufgebaut werden, so wird dafür der DBSpace genutzt, in dem die Datenbank erstellt wurde:

database	tablename	owner	dbspace	rows	rowsz
stores	yy	informix	datadbs	79596	500
stores	xx	informix	tmp1dbs	13867	500
stores	xx	informix	tmp2dbs	32864	500
stores	xx	informix	tmp3dbs	32865	500

Fazit:

Es ist empfehlenswert mehrere TempDBSpaces zu erstellen und diese in der ONCONFIG als DBSPACETEMP einzutragen. Bei temporären Tabellen ohne Logging verbessert dies die Performance durch Parallelisierung.

Zudem sollte auch immer zumindest ein TempDBSpace eingetragen werden, der mit Logging arbeitet um für das Backup und temporäre Tabellen mit Logging zur Verfügung zu stehen.

TEMPDBSpaces mit mehreren Chunks machen kaum Sinn und sollten daher vermieden werden.

## TechTipp: Environment - FORCE\_DDL\_EXEC

Administratoren haben immer wieder das Problem, dass Änderungen an der Datenbank nicht vorgenommen werden können, da solche Aktionen von laufenden Applikationen verhindert werden und dann den Fehler „table is locked“ verursachen.

Mit der Session-Umgebungsvariablen **FORCE\_DDL\_EXEC** können Transaktionen beendet werden, die Sperren auf der Tabelle halten, und dadurch eine DDL Anweisung vom Typ **ALTER FRAGMENT ON TABLE** blockieren würden.

**FORCE\_DDL\_EXEC** kann dabei mit dem Befehl „SET ENVIRONMENT ...“ auf folgende Werte gesetzt werden:

- „OFF“, „off“, „0“                    Das Feature ist ausgeschaltet (Default).
- „ON“, „on“, „1“                      Das Feature ist aktiviert und beendet sperrende Sessions bzw. Applikationen.
- n (integer > 0)                      Maximale Zeit in Sekunden, die versucht wird die sperrende Transaktion zu beenden.  
Um unnötige Fehler in dieser Zeit zu vermeiden, sollte zudem „lock mode to wait n“ gesetzt werden.

Beispiel: „SET ENVIRONMENT FORCE\_DDL\_EXEC '1';“.

Das Feature kann nur von einem User mit DBA-Rechten eingesetzt werden. Zudem muss die Datenbank mit Logging betrieben werden.

## TechTipp: SQL Funktion SIGN()

Immer wieder erreichen unsere Redaktion Anfragen zur Verfügbarkeit und zum Aufruf von Funktionen im SQL. Obwohl der Umfang der mitgelieferten Funktionen bereits sehr groß ist, und mit fast jeder Version neue Funktionen hinzu kommen, gibt es doch immer wieder die Notwendigkeit, eine Funktionalität selbst mit Hilfe einer Stored Procedure zu erstellen. Anbei ein Beispiel für die Funktion SIGN(), die das Vorzeichen eines Wertes zurück gibt:

```
create function if not exists sign(x_in dec)
returning int
if x_in >= 0 then return 1;
else return -1;
end if
end function;
```

Die Funktion kann entweder explizit mittels „execute function“ aufgerufen werden:

```
execute function sign(42);
execute function sign(0);
execute function sign(-1.4223);
```

oder in der SELECT- oder WHERE-Clause eines SQL-Statements eingebunden werden:

```
select order_num, sign(ship_weight) as sign from orders;
```

## TechTipp: Neue SQL Funktionen

Wie bereits im Newsletter August angekündigt, wollen wir hier die neuen SQL-Funktionen der Version 11.70.xC3 anhand von praktischen Beispielen vorstellen.

Die Funktionen jeweils mit einem Beispiel:

➤ **CHARINDEX(Suchstring,Text,Startposition)**

Die Startposition eines Suchstrings ermitteln (mit optionaler Angabe für Startpunkt)

```
select charindex("Test","IFX Test Test Test")
      from systables where tabid = 1;
Ergebnis: 5
```

```
select charindex("Test","IFX Test Test Test Test",7)
      from systables where tabid = 1;
Ergebnis: 10
```

Hinweis: In der Dokumentation wurde ein falsches Beispiel angegeben, bei dem die Reihenfolge der Argumente vertauscht ist:

```
select charindex("www.ibm.com","com") from systables where tabid = 1
```

➤ **INSTR(Text,Suchstring,Startposition,Anzahl)**

Die Startposition des n-ten Vorkommens eines Suchstrings ermitteln.

```
select instr("IFX Test Test Test","Test")
      from systables where tabid = 1;
Ergebnis: 5
```

```
select instr("IFX Test Test Test","Test",1,2)
      from systables where tabid = 1;
Ergebnis: 10
```

➤ **LEFT( )**

Die ersten x Zeichen eines Strings ausgeben

```
select left("IBM INFORMIX Newsletter",3)
      from systables where tabid = 1;
Ergebnis: IBM
```

➤ **RIGHT( )**

Die letzten x Zeichen eines Strings ausgeben

```
select right("IBM INFORMIX Newsletter",10)
      from systables where tabid = 1 ;
Ergebnis: Newsletter
```

➤ **LEN(String)**

Alias für für die bereits existierende Funktion Length()

```
select len("IBM INFORMIX Newsletter")
      from systables where tabid = 1;
Ergebnis: 23
```

- **REVERSE(String)**  
Umkehrung der Schriftrichtung

```
select reverse("IBM INFORMIX Newsletter")
      from systables where tabid = 1;
Ergebnis: rettelsweN XIMROFNI MBI
```

- **SPACE(anzahl)**  
String bestimmter Länge aus Leerzeichen (blancs) erzeugen

```
select "_"||space(13)||"_" from systables where tabid = 1;
Ergebnis: "_                _"
```

- **SUBSTRING\_INDEX( )**  
Teilstring bis zu einem beliebigen Delimiter ausgeben

```
select substring_index("www.informix.com",".",1)
      from systables where tabid = 1;
Ergebnis: www
```

```
select substring_index("www.informix.com",".",2)
      from systables where tabid = 1;
Ergebnis: www.informix
```

```
select substring_index("www.informix.com",".",3)
      from systables where tabid = 1;
Ergebnis: www.informix.com
```

```
select substring_index(substring_index("www.informix.com",".",2),"",-1)
      from systables where tabid = 1;
Ergebnis: informix
```

- **DEGREES(Kreissegment)**  
Gradzahl zu einem Kreissegment errechnen ( $\text{Kreisbogen} \cdot 180/\text{Pi}$ )

```
select degrees(3.14159) from systables where tabid = 1;
Ergebnis: ~180°
```

- **RADIANS (Grad)**  
Kreissegment zu einer Gradzahl errechnen ( $\text{Grad} \cdot \text{Pi}/180$ )

```
select radians(180) from systables where tabid = 1;
Ergebnis: Die Zahl Pi
```

## TechTipp: Built-in Tasks and Sensors (Teil 1)

Eine Reihe von Task und Sensoren werden bei der Installation in der Datenbank sysadmin bereits als „Built-in“ mit installiert. Teilweise sind diese Tasks bereits aktiv, teilweise müssen diese bei Bedarf erst aktiviert werden.

Mit diesem Artikel wollen wir eine Übersicht schaffen, was in der Datenbank im Hintergrund läuft, und was sie eventuell zusätzlich nutzen können.

Da sowohl die Tasks, als auch die Sensors in der selben Tabelle verwaltet werden, und auch nach dem selben Prinzip aktiviert werden, wird in diesem Artikel nicht zwischen diesen beiden Arten von Jobs unterschieden.

Task Name	Default	Interval	Beschreibung
Add storage	enabled	bei Bedarf	Erweiterung von Chunks/DBSpaces bei autoextend/autoexpand
Alert Cleanup	enabled	täglich	Löschen alter Alarme
auto_crtd	disabled	täglich	Compress/Shrink/Repack
auto_tune_cpu_vps	disabled		CPU Vps beim Systemstart hinzufügen
autoreg exe	enabled	bei Bedarf	Registrierung von Datablades bei Nutzung
autoreg migrate-console	enabled	bei Start	Erstellt Aufrufe für „autoreg migrate <db>“
Autoreg migrate <db>	enabled	Bei Start	Migriert Datablades je Datenbank falls notwendig
autoreg vp	enabled	bei Bedarf	Erstellung von VPs für Datablades bei Nutzung
Auto Update Statistics Evaluation	enabled	täglich	Ermittelt ob update statistics notwendig ist und erstellt die notwendigen Statements anhand der Vorgaben aus ph_threshold ( AUS_AGE, AUS_CHANGE, AUS_AUTO_RULRES, AUS_SMALL_TABLES)
Auto Update Statistics Refresh	enabled	(Sa/So 01:00-05:00)	Führt vom „Auto Update Statistics Evaluate“ erstellte Statements aus.
bad_index_alert	disabled	täglich	Alarmiert bei defekten Indices
bar_act_log_rotate	disabled	30Tage 03:00	Rotation der bar_act_log files
bar_debug_log_rotate	disabled	30Tage 03:00	Rotation der bar_act_log files
check backup	enabled	täglich	Prüft ob ein aktuelles Backup vorliegt
check_for_ipa	disabled	wöchentlich	Prüft auf pending alter table
idle_user_timeout	disabled	2 stündlich	Prüft auf inaktive Sessions (>60 min idle)
ifx_ha_monitor_log_replay_task	enabled	bei Bedarf	Protokolliert die HA Replaypositionen
Job runner	enabled	bei Bedarf	Tasks für das OAT
Job results cleanup	enabled	bei Bedarf	Räumt nach 30 Tagen die Infos des Job runners auf
Low Memory Manager	disabled		Systemanpassung falls der verfügbare Speicher knapp wird
Low Memory Reconfig	enabled	stündlich	Anpassung des Systems an vorhandenen Speicher



mon_checkpoints	enabled	stündlich	Informationen über Checkpoints archivieren in sysadmin:mon_chckpoint
mon_command_history	enabled	täglich	Command history auf 30 kürzen
mon_compression_estimates	enabled	wöchentlich	Ermitteln der möglichen Compression
mon_config	enabled	täglich	Aktuelle Werte der onconfig protokollieren
mon_config_start	enabled	bei Start	Aktuelle Werte der onconfig beim start
mon_low_storage	enabled	stündlich	ggf. Erweitern von Chunks bei Unterschreitung Threshold
mon_memory_system	enabled	stündlich	Memory der Instanz
mon_page_usage	enabled	täglich	Page Usage ermitteln
mon_profile	enabled	4-stündlich	Profile der Instanz
mon_sysenv	enabled	beim Start	Environments beim Start
mon_table_names	enabled	täglich	Liste der Tabellen mit Erstellungsdatum in der Tabelle sysadmin:mon_table_names
mon_table_profile	enabled	täglich	Profile der Tabellen (insert/update/deletes)
mon_users	enabled	4-stündlich	Profile der User
mon_vps	enabled	4-stündlich	Profile der VPs
online_log_rotate	disabled	30 Tage 03:00	Rotation des online log files
post_alarm_message	enabled	stündlich	Verbreitet Alarme
refresh_table_stats	disabled		Tabellen Statistiken aktualisieren
Save SQL Trace	disabled	15min	Archivieren der SQL History in der Tabelle sysmaster:syssqltrace_info

Je nach eingesetzter Version und Aufgabenstellung kann es sein, dass ein Teil dieser Tasks unnötig läuft und Ressourcen frisst. So ist z.B. der mon\_compression\_estimates wöchentlich aktiv, auch wenn man dieses Feature nicht lizenziert hat und es daher nicht einsetzen darf. Der Task „online\_log\_rotate“ hingegen, der sehr nützlich sein kann um das online.log übersichtlich zu halten, ist deaktiviert.

Der „auto\_crtd“ ist per Default inaktiv. Da dieser jedoch nicht nur die Compression beinhaltet, sondern auch den Repack und Shrink, die in allen Versionen eingesetzt werden dürfen, sollte dieser Task aktiviert werden um die automatische Reorganisation von Tabellen mit vielen Extents zu ermöglichen.

Bei Systemen, bei denen immer wieder Änderungen vorgenommen werden, ist eine Historie manchmal viel Wert. Wird z.B. eine Tabelle gelöscht und neu aufgebaut, können die Erstellungsdaten der Tabelle rückverfolgt werden.

Die Tasks können bequem über das OpenAdminTool aktiviert und deaktiviert werden. Dort können auch die Einstellungen vorgenommen werden, an welchen Tagen und in welchem Zeitraum die Tasks aktiv werden dürfen.

Task Übersicht im OAT:

Name	Start Time	Task Schedule							Enabled			
		Stop Time	Run Frequency	M	T	W	T	F		S	S	
check_backup	05:00:00	NEVER	1 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
mon_vps		NEVER	0 04:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
add_storage		NEVER		✓	✓	✓	✓	✓	✓	✓	✓	✓
mon_low_storage		NEVER	0 01:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
mon_memory_system		NEVER	0 02:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
autoreg migrate-console		NEVER		✓	✓	✓	✓	✓	✓	✓	✓	✓
Auto Update Statistics Evaluation	01:00:00	01:10:00	1 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
Auto Update Statistics Refresh	01:11:00	05:00:00	1 00:00:00	✗	✗	✗	✗	✗	✓	✓	✓	✓
mon_profile		NEVER	0 04:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
mon_users		NEVER	0 04:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
Alert Cleanup	02:00:00	NEVER	1 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
Job Runner		NEVER		✓	✓	✓	✓	✓	✓	✓	✓	✗
Low Memory Reconfig		NEVER	0 01:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
auto_crsd	03:00:00	NEVER	7 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✗
auto_tune_cpu_vps		NEVER		✓	✓	✓	✓	✓	✓	✓	✓	✗
autoreg exe		NEVER		✓	✓	✓	✓	✓	✓	✓	✓	✗
autoreg vp		NEVER		✓	✓	✓	✓	✓	✓	✓	✓	✗
bad_index_alert	04:00:00	NEVER	1 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✗
bar_act_log_rotate	03:00:00	03:01:00	30 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✗
bar_debug_log_rotate	03:00:00	03:01:00	30 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✗
check_for_ipa	04:00:00	NEVER	7 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✗
idle_user_timeout		NEVER	0 02:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✗
ifx_ha_monitor_log_replay_task		NEVER		✓	✓	✓	✓	✓	✓	✓	✓	✓
mon_checkpoint		NEVER	0 01:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓
mon_config		NEVER	1 00:00:00	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tasks bzw. Sensors, die Ergebnistabellen erstellen, schaffen die Grundlage zur nachfolgenden Auswertung mittels SQL. Dies ist z.B. der Fall bei der Protokollierung der neuen Tabellen im Task „mon\_tabnames“, der die Tabelle „mon\_tabnames“ in der Datenbank „sysadmin“ erstellt.

Beispiel zur Auswertung der erfassten Werte des Tasks „mon\_tabnames“:

```
select dbsname[1,12], tabname[1,28], owner[1,12],
dbinfo('utc_to_datetime',created) as cre_dat
from mon_table_names
order by 1,2,3
```

Ergebnis:

dbsname	tabname	owner	cre_dat
stores	customer	informix	2011-05-24 21:10:47
stores	bogy	carmen	2010-12-23 02:12:42
stores	kalu	kalu	2010-12-13 08:42:13
stores	lindau	marion	2011-08-24 13:42:23
stores	orders	informix	2011-05-24 21:10:47
...			

Die Liste der eingebauten Tasks (\*) ist zu finden unter:

[http://publib.boulder.ibm.com/infocenter/idshelp/v117/index.jsp?topic=/com.ibm.admin.doc/ids\\_admin\\_1328.htm](http://publib.boulder.ibm.com/infocenter/idshelp/v117/index.jsp?topic=/com.ibm.admin.doc/ids_admin_1328.htm)

(\*) Die Liste diene als Vorlage für die obige Tabelle. Es wurden die Tasks hinzugefügt, die auf einem frisch installierten System in der Version 11.70.xC3 in der Tabelle ph\_task eingetragen sind. Je Version unterscheiden sich die automatisch installierten Tasks, bzw. deren Defaulteinstellung (bei Migration InPlace bleiben frühere Tasks erhalten).

Teil 2 zu „Tasks und Sensors“, geplant für die Ausgabe November 2011, wird auf die Steuerung von Tasks über die Parameter der Tabelle ph\_threshold eingehen und an einem Beispiel zeigen, wie einfach ein neuer Task für eigene Zwecke erstellt werden kann.

## Referenzen: National Institute for Testing and Evaluation

Das „National Institute for Testing and Evaluation (NITE)“ hat sich für INFORMIX entschieden, da diese Lösung in allen Bereichen den größten Nutzen bietet. Sowohl bei der Hochverfügbarkeit, der Performance als auch bei den Hardware- und Softwarekosten wurde INFORMIX als beste Lösung gewählt. Zudem bietet INFORMIX durch den geringen Administrationsaufwand den Vorteil, dass die Datenbankadministratoren Zeit haben, sich um neue Projekte und die Ausweitung der Serviceangebote zu kümmern.

Die ausführliche Beschreibung finden Sie unter:

[http://www.ibm.com/software/success/cssdb.nsf/cs/JHAL-8J2UAP?OpenDocument&Site=corp&cty=en\\_us](http://www.ibm.com/software/success/cssdb.nsf/cs/JHAL-8J2UAP?OpenDocument&Site=corp&cty=en_us)

## Termine: 58. IUG-Workshop in Hamburg

Der 58. Workshop der Informix User Group findet am 18.10.2011 in Hamburg statt. Das Thema des Workshops ist diesmal:

**„Applikationsentwicklung, Informix als Backend von Business-Lösungen“**

Wie gewohnt findet am Vorabend ein Stammtisch statt, an dem sich die Gelegenheit bietet in gemütlicher Atmosphäre Kontakt zu knüpfen und Erfahrungen auszutauschen. Sollten sie noch kein Mitglied der IUG sein, dann wenden Sie sich an [info@iug.de](mailto:info@iug.de). Dort erfahren Sie mehr über die Vorteile, zu dieser Gemeinschaft zu gehören.

Der Stammtisch findet im  
ARCOTEL Rubin Hamburg  
Steindamm 63  
20099 Hamburg  
statt. Beginn ist 19:00 Uhr.

Der Workshop ist dann im Gebäude der  
IBM Deutschland GmbH  
Geschäftstelle Hamburg  
Beim Strohhause 17  
20097 Hamburg

Pünktlicher Beginn um 08:59 Uhr ist durch Dr. Aspiazu garantiert.

Weitere Informationen zum Workshop finden Sie im Web unter:

[http://www.iug.de/index.php?option=com\\_content&task=view&id=207&Itemid=279](http://www.iug.de/index.php?option=com_content&task=view&id=207&Itemid=279)

## WebTipp: Analyse Tools im Bundle mit INFORMIX UWE

„PreferredPartner“, ein IBM Premium Partner bietet im Datawarehouse Umfeld eine Komplettlösung aus Hardware, Software und INFORMIX UWE an. Der Einsatz des INFORMIX Ultimate Warehouse Accelerators bietet unschlagbare Performance, so dass auch komplexe Abfragen in wenigen Minuten ausgeführt werden können.

Mehr zur Lösung und den Vorteilen lesen Sie unter:

<http://archive.constantcontact.com/fs087/1105585926437/archive/1107046062481.html>

## Versionsinfo: 11.10.xC3W2 ist verfügbar

Seit einigen Tagen ist die Version 11.10.xC3W2 für alle unterstützten Plattformen und Editionen verfügbar. Da es von Version 11.10 zu 11.50 und 11.70 erhebliche Verbesserungen gibt, sollte bei einem Umstieg gleich eine höhere Version mit in Betracht gezogen 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](mailto:ifmxnews@de.ibm.com) senden.

Im Falle einer Abmeldung senden Sie „**ABMELDUNG**“ an diese Adresse.

## Neu hinzugekommen ist ein Archiv der INFORMIX Newsletters bei der International Informix User Group.

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.bytec.de/de/software/ibm\\_software/newsletter/](http://www.bytec.de/de/software/ibm_software/newsletter/)

<http://www.cursor-distribution.de/index.php/aktuelles/informix-newsletter>

[http://www.listec.de/Informix\\_Newsletter/](http://www.listec.de/Informix_Newsletter/)

<http://www.bereos.eu/software/informix/newsletter/>

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit. Da uns weder Tippfehler noch Irrtümer fremd sind, bitten wir hier um Nachsicht falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

## Die Autoren dieser Ausgabe

Gerd Kaluzinski                      IT-Specialist Informix Dynamic Server und DB2 UDB  
                                         IBM Software Group, Information Management  
[gerd.kaluzinski@de.ibm.com](mailto:gerd.kaluzinski@de.ibm.com)                      +49-175-228-1983

Martin Fuerderer                      IBM Informix Entwicklung, München  
                                         IBM Software Group, Information Management  
[martinfu@de.ibm.com](mailto:martinfu@de.ibm.com)

Sowie unterstützende Teams im Hintergrund.

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

Fotonachweis:                      Gerd Kaluzinski (Bodensee - Blick vom Höchsten/Deggenhausertal)