

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: Warehouse Accelerator – Views im Data Mart.....	2
TechTipp: dbimport – no constraint check.....	3
TechTipp: Verschieben und Zurücksetzen der Datenbank sysadmin	3
TechTipp: What's new in 12.10.xC4.....	4
TechTipp: Spatial Datablade - Einführung (Teil 1).....	5
TechTipp: ONCONFIG - SESSION_LIMIT_LOCKS.....	9
TechTipp: Environments - IFX_SESSION_LIMIT_LOCKS.....	9
TechTipp: infoshp – Anzeigen der Informationen eines Shapefiles.....	10
TechTipp: DBSpaces im Wachstum begrenzen – max_size in sysmaster.....	14
TechTipp: Tasks - reset config	14
Anmeldung / Abmeldung / Anmerkung.....	15
Die Autoren dieser Ausgabe.....	15

Aktuelles

Liebe Leserinnen und Leser,

zum Start der Ferien in einigen Bundesländern wollten wir den Newsletter kompakter halten, damit dieser schnell im Urlaub überflogen werden kann. Die Fülle an Informationen rund um Informix hat aber doch wieder zu einer umfangreichen Ausgabe geführt. Da die Redaktion im August selbst durch umfangreiche Recherchen zu Sonne, Sand und Meer gebunden ist, planen wir aktuell eine kürzere Version für August, bevor wir dann wieder mit neuer Energie in den Herbst starten.



Wie immer haben wir für Sie eine Reihe an Tipps und Tricks zusammengestellt. Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

TechTipp: Warehouse Accelerator – Views im Data Mart

Zusätzlich zur Unterstützung von Synonymen auf Tabellen (siehe auch Ausgabe Juni 2014 des Newsletters) erlaubt die neue Version 12.10.xC4 auch die Verwendung von Views in einem Data Mart. So wie für Synonyme sind auch für Views einige Besonderheiten zu beachten.

- Enthält ein Data Mart Views, dann müssen die Abfragen, für die der Data Mart erstellt wurde, diese Views auch benutzen (und nicht die darunterliegenden Tabellen, auf denen die Views basieren). Nur so kann der Algorithmus des "Query Matching" auch zuverlässig den zu benutzenden Data Mart bestimmen. Benutzen Abfragen statt den Views die darunterliegenden Tabellen, dann kann es sein, dass sie nicht beschleunigt werden, weil der passende Data Mart nicht als solcher identifiziert werden kann.
- Der Systemkatalog enthält keine Information über die Anzahl von Datensätzen, die ein View produziert. Daher können die Algorithmen des "Query Matching" und "Query Probing" für Views diese Information auch nicht benutzen, um die Faktentabelle einer Abfrage zu bestimmen. Abfragen, die Views benutzen, müssen daher so formuliert sein, dass es anhand der Syntax der Abfrage möglich ist, die Faktentabelle zu bestimmen. Es kann somit notwendig sein, die Abfrage mit einer Optimizer-Direktive für die Faktentabelle zu ergänzen. Solche Optimizer-Direktiven haben die Form {+FACT *tabellen- oder viewname*} .
- Da es nicht möglich ist, für einen View einen Index zu erzeugen, kann beim Erstellen eines Data Marts für einen View, der in einer Referenz die Elterntabelle ist, das Vorhandensein eines Indexes auch nicht geprüft werden. Dies würde bedeuten, dass in einem Data Mart eine Referenz nicht die bevorzugte 1:n Kardinalität haben kann, wenn die Elterntabelle ein View ist.

Damit eine Referenz die 1:n Kardinalität haben kann, obwohl die Elterntabelle ein View ist, kann die Überprüfung der Eindeutigkeit abgeschaltet werden. Dies geschieht mit folgendem SQL-Befehl, bevor ein Data Mart erzeugt wird:

```
set environment use_dwa 'uniquecheck off';
```

Der Befehl schaltet die Eindeutigkeitsprüfung für alle 1:n Referenzen im Data Mart ab, nicht nur für solche, deren Elterntabelle ein View ist. **Es liegt in der Verantwortung des Datenbankadministrators sicherzustellen, dass die Schlüssel eines Views, der eine Elterntabelle in einer Referenz ist, trotzdem eindeutig sind. Wird eine 1:n Referenz benutzt, ohne dass diese Bedingung zutrifft, dann liefert der Data Mart falsche Ergebnisse!**

- Die Funktionalitäten von Data Mart Refresh und Trickle Feed sind für Data Marts mit Views nicht unterstützt.

Obwohl die obigen Einschränkungen hinderlich erscheinen, sind sie doch ein kleiner Preis für all die neuen Möglichkeiten, die durch die Benutzung von Views in Data Marts eröffnet werden. Zum Beispiel kann ein einfacher View benutzt werden, um einen Datentyp oder eine Funktion, die IWA selbst nicht unterstützt, auf etwas abzubilden, das mit IWA funktioniert. Oder ein View kann dazu benutzt werden, nur eine Teilmenge einer sehr grossen Tabelle in einem Data Mart zur Verfügung zu stellen. Dies kann sehr nützlich sein, wenn eine Tabelle grosse Mengen historischer Daten enthält, jedoch nur ein Teil davon für beschleunigte Analyse-Abfragen von Interesse ist.

Andererseits ist es nicht sehr effektiv, einen View mit komplexen Verknüpfungen mehrerer Tabellen zu definieren und diesen dann in einem Data Mart zu benutzen, denn die Verarbeitung solcher Verknüpfungen ist eine Domäne von IWA selbst. Auf jeden Fall würde so ein View sehr lange Ladezeiten für den Data Mart verursachen, denn die Verarbeitung des komplexen Views fällt beim Ladevorgang zu Lasten des Informix Servers.

TechTipp: dbimport – no constraint check

Für den Aufruf von „dbimport“ wurde neu die Option „-nv“ eingeführt, die beim Import einer Datenbank die Prüfung von Fremdschlüsseln (foreign keys) vermeidet.

Sind die Daten aus einer konsistenten Quelle exportiert, so kann durch Weglassen dieser dann überflüssigen Prüfung die Performance erheblich gesteigert werden.

TechTipp: Verschieben und Zurücksetzen der Datenbank sysadmin

Die Datenbank sysadmin liegt per Default im Rootdbspace. Da diese Datenbank mit Nutzung des Tasks „Save SQL Trace“ durchaus stark anwachsen kann, macht es Sinn, diese Datenbank in einen eigenen DBSpace zu verschieben.

Das Verschieben baut die Datenbank sysadmin neu auf, so dass hierbei gleichzeitig eine Bereinigung der angesammelten Werte stattfindet. Dabei werden auch alle Informationen über die Command History und die Laufzeiten der Tasks gelöscht.

Der Befehl um die Datenbank sysadmin neu zu initialisieren lautet:

```
EXECUTE FUNCTION task (reset sysadmin');
```

Soll dabei die Datenbank in einen eigenen DBSpace verschoben werden, so wird als weiteres Argument der Name des DBSpaces mitgegeben:

```
EXECUTE FUNCTION task('reset sysadmin', 'admindbs');
```

Obwohl beide Befehle die Datenbank sysadmin neu aufbauen, werden sie aus der Datenbank sysadmin heraus aufgerufen wie alle Tasks.

TechTipp: What's new in 12.10.xC4

Wie in allen Releases sind neben Fehlerbehebungen auch neue Features erstellt, bzw. Anpassungen vorgenommen worden.

Anbei eine kurze Auflistung der Änderungen (ohne Anspruch auf Vollständigkeit):

Migration

- Neu in der ONCONFIG: CDR_MEM, SESSION_LIMIT_LOCKS
- Neue Umgebungsvariable: IFX_PUA_DISPLAY_MAPPING
- Neue Session Umgebungsvariable: IFX_SESSION_LIMIT_LOCKS
- Neue Datenbanktabelle: tenant
- Neue Spalten in der Tabelle „sysindices“: indexattr und jparam
- Vereinfachter Import von Tabellen mit „large rows“

Installation

- Einfachere Installation der 32-bit Version auf Windows 64-bit
- Neu: Uninstall von Informix auf Windows aus dem Control Panel

Administration

- Multitenancy in Informix (ein TechTipp hierzu erscheint im Newsletter August)
- Erweiterungen des Open Admin Tools (OAT)
- Einfachere Bereinigung von „Pending alter Operationen“
- Begrenzung der Grösse bei „extendable storage spaces“
- Begrenzung der Anzahl der Sperren einer Session (siehe Tipp weiter unten)
- Neuer Default Mode für „VP_MEMORY_CACHE_VP“ in der ONCONFIG

Replikation

- Erweiterung der Enterprise Replication zur Nutzung des Memory Pools
- Replikation von Hertz und komprimierten Daten der TimeSeries
- Neuer Event Alarm für blockierte Transaktionen der Replikation

Performance

- Schnellere „Storage Optimization“ (Compression)
- Schnellere Joins bei Nutzung von „UNION ALL“ mit Views

Applikationsentwicklung

- Anpassung der Anzeigebreite von „Unicode private-use characters“
- SQL Kompatibilität: LIMIT-clause am Ende der Abfrage erlaubt (analog DB2)

JSON Kompatibilität

- Erweiterte JSON Kompatibilität
- Zugriff auf Informix von REST API Clients
- Erstellen einer TimeSeries mit REST API oder MongoDB API
- Basic Text Searching (BTS) Support für JSON und BSON Daten
- Zugriff auf BSON Daten über JDBC Applikationen
- Schneller Export von relationalen Tabellen in BSON oder JSON Dokumente

TimeSeries

- Aufnahme von JSON Dokumenten in TimeSeries
- Erweiterungen der TimeSeries Java Class Bibliothek
- Hertz als Granularität der TimeSeries (Verfügbar bereits ab xC3)

Sicherheit

- PAM Passwort Authentifizierung für DRDA Verbindungen

TechTipp: Spatial Datablade - Einführung (Teil 1)

In diesem Newsletter wollen wir Ihnen die Nutzung von Spatial Daten innerhalb Ihrer Informix Instanz vorstellen. Das Spatial Datablade wird seit langer Zeit mit dem Datenbank Server ausgeliefert und ist (wenn es nicht bei der Installation explizit abgewählt wird) im Unterverzeichnis \$INFORMIXDIR/extend/ als Verzeichnis „spatial.8.21.FCx“ zu finden.

Vor der Nutzung der Spatial Daten sollte das Datablade in der Datenbank registriert werden (wir empfehlen diese explizite Registrierung, da diese gegenüber der automatischen Registrierung bei Nutzung ein Reihe an Meldungen vermeidet).

Vorarbeit: (Registrierung mittels „Blade Manager“)

Im „Blade Manager zeigt der Befehl „show modules“ die Module, die zur Verfügung stehen:

blademgr

```
ifxibm>show modules
```

```
11 DataBlade modules installed on server ifxibm:
```

ifxbuiltins.1.1	TimeSeries.6.00.FC3
ifxrmtree.2.00	LLD.1.20.FC2
mqlblade.2.0	wfs.1.00.FC1
spatial.8.21.FC6	bts.3.00
binaryudt.1.0	Node.2.0
excompat.1.0	

Mit dem Befehl „register <datablade_name> <database_name>“ kann nunmehr das Datablade in der Datenbank registriert werden (Wir nutzen die Testdatenbank „stores“):

```
ifxibm>register spatial.8.21.FC6 stores
```

```
Register module spatial.8.21.FC6 into database stores? [Y/n]Y
```

```
Registering DataBlade module... (may take a while).
```

```
Module spatial.8.21.FC6 needs interfaces not registered in database stores.
```

```
The required interface is provided by the modules:
```

```
1 - ifxrmtree.2.00
```

```
Select the number of a module above to register, or N :- 1
```

```
Registering DataBlade module... (may take a while).
```

```
DataBlade ifxrmtree.2.00 was successfully registered in database stores.
```

```
Registering DataBlade module... (may take a while).
```

```
DataBlade spatial.8.21.FC6 was successfully registered in database stores.
```

Durch diese Registrierung werden die zusätzlichen Datenstrukturen und Funktionen des Datablade in der Datenbank initialisiert.

Die zusätzlichen zur Verfügung stehenden Datentypen sind:

- **ST_Point**
- **ST_LineString**
- **ST_Polygon**
- **ST_MultiPoint**
- **ST_MultiLineString**
- **ST_MultiPolygon**

Als Default stehen eine Vielzahl von Referenzsystemen zur Verfügung, die in der Tabelle „spatial_references“ eingetragen sind:

```
EPSG 2000, version 8.1.1  
EPSG 2001, version 8.1.1  
...  
EPSG 4759, version 8.1.1  
EPSG 4760, version 8.1.1  
ESRI 54001, version 10.2.0  
ESRI 54002, version 10.2.0  
ESRI 54053, version 10.2.0  
NAD 1927 datum for use with unprojected lat/lon coords  
NAD 1983 datum for use with unprojected lat/lon coords  
Old Hawaiian datum for use with unprojected lat/lon coords  
Unspecified datum for use with unprojected lat/lon coords  
WGS 84 datum for use with unprojected lat/lon coords
```

Es können jedoch beliebige andere Referenzsysteme eingetragen werden, die dann als Basis der geometrischen Informationen dienen.

Die Erstellung eines neuen Referenzsystems für die flache Erde mit Longitude und Latitude könnte mit dem Aufruf „SE_CreateSrid()“ erfolgen:

```
EXECUTE FUNCTION SE_CreateSrid  
(-180, -180, 180, 180, "World: lat/lon coords");
```

Der Rückgabewert der Funktion entspricht der eindeutigen SRID, die für die Erfassung und Abfrage der Geodaten zu diesem Referenzsystem genutzt werden kann.

In unseren Beispielen nutzen wir das bereits vorhandene Referenzsystem mit der **SRID 4326** „EPSG 4326, version 8.1.1“.

Nun können wir eine Tabelle erstellen, in der der Datentyp „ST_POINT“ verwendet wird. In unserem Beispiel haben wir zusätzlich die Spalten „latitude“ und „longitude“ eingefügt, um bei einem INSERT nicht sofort die Syntax für ST_POINT verwenden zu müssen:

```
create table "informix".restaurant (  
    id char(18),  
    ...  
    tel char(18),  
    latitude decimal(12,9),  
    longitude decimal(12,9),  
    location "informix".st_point,  
    primary key (id)  
);  
  
insert into restaurant values (  
    "VAP_MUC1",  
    ...  
    48.1406128,  
    11.5750208,  
    null  
);
```

... dies simuliert im Beispiel die Stammdaten, die ja meist bereits vorhanden sind. (Weitere Testdaten, falls sie dies Beispiel nachstellen wollen: „48.1424192 / 11.5493539“, „48.148428900,11.6174928009“, „48.1494479, 11.4603127“).

Mit diesen Daten füllen wir nun den Datentyp „ST_Point“:

```
update restaurant  
set location = ST_Point(latitude,longitude,4326)  
where latitude is not null and longitude is not null  
and location is null;
```

Um eine Umgebungssuche zu starten, muss auf den Geodaten ein Index erstellt werden:

```
CREATE INDEX i1_restaurant ON restaurant  
    (location ST_Geometry_ops) USING RTREE;
```

Nun kann die Umgebungssuche beginnen. Wichtig hierbei ist der Bezugspunkt, von dem aus Sie suchen wollen. Haben Sie selbst eine Adresse in der Datenbank, dann können Sie diese als Zentrum der Suche nutzen. Falls Sie unterwegs sind, dann hilft meist die GPS-Funktion des Smartphones, indem Sie dort die Koordinaten ablesen können.

Angenommen Sie befinden sich in der Nähe der TU München (Kompass am Smartphone zeigt **48.1497659, 11.564514**) und suchen die naheliegenden Filialen eines bestimmten Restaurants. Hierzu gibt es die Funktion `SE_Nearest()`:

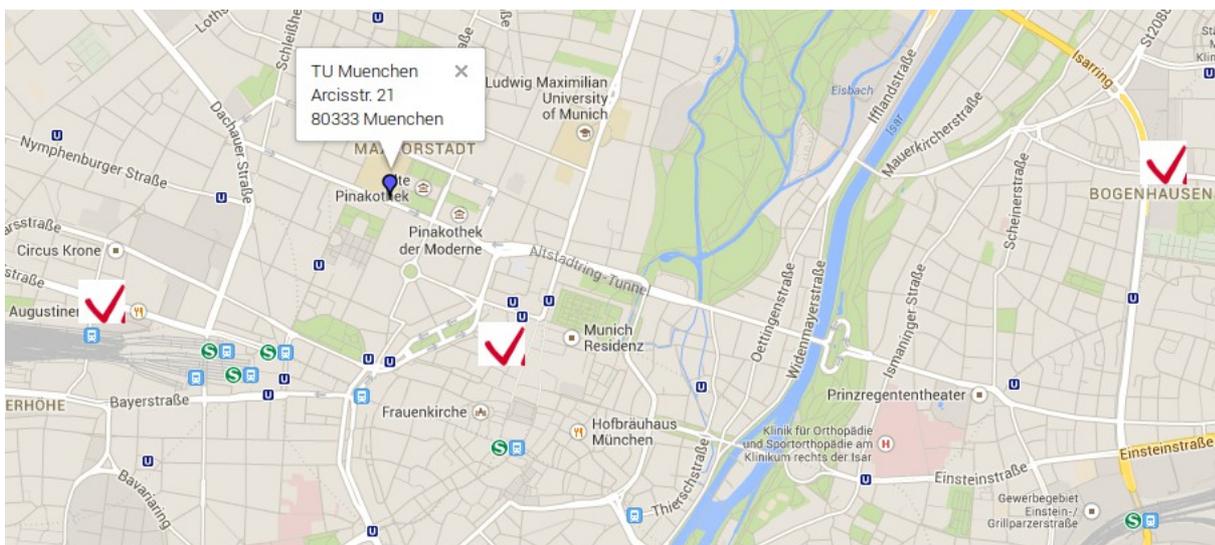
```
select first 2
*, ROUND(ST_Distance(restaurant.location,
    ST_point(48.1497659, 11.564514, 4326), 'kilometer'
),2) as km
from restaurant
where SE_Nearest(restaurant.location,
    ST_point(48.1497659, 11.564514, 4326)
)
```

Das Ergebnis:

```
id          VAP_MUC1
str         Theatinerstraße 15
...
location    4326 POINT (48.1406128 11.5750208)
km          1.39

id          VAP_MUC2
str         Hackerbrücke 4
...
location    4326 POINT (48.1424192 11.5493539)
km          1.68
```

Testen sie selbst am Beispiel von IBM München (48.13518, 11.67067)



Weitere Funktionen und Anwendungsbeispiele stellen wir in der kommenden Ausgabe des Informix Newsletters vor.

TechTipp: ONCONFIG - SESSION_LIMIT_LOCKS

Dieser Parameter ermöglicht es, die Anzahl der Sperren je Session zu begrenzen. Der Parameter ist in der „onconfig.std“ nicht vorhanden.

Der Defaultwert, wenn der Parameter nicht gesetzt ist, ist MAXINT (2147483647).

Als unterstes Limit kann 500 gewählt werden. Wird ein tieferer Wert gesetzt, so wird der Wert auf das erlaubte Limit von 500 geändert. Im online.log steht dann der Hinweis:

```
Parameter's user-configured value adjusted (SESSION_LIMIT_LOCKS)
```

Der Wert in der ONCONFIG kann mittels „onmode -wf / -wm“ dynamisch geändert werden.

TechTipp: Environments - IFX_SESSION_LIMIT_LOCKS

Diese Umgebungsvariable ermöglicht es, den in der ONCONFIG gesetzten Wert für SESSION_LIMIT_LOCKS für die aktuelle Session anzupassen.

Soll z.B. die Anzahl der Sperren für eine Session auf 42000 begrenzt werden, so kann dies in der Umgebung gesetzt werden:

```
set environment IFX_SESSION_LIMIT_LOCKS '42000'
```

Wird das Limit erreicht, so wird in der Session folgender Fehler zurückgegeben:

```
346: Could not update a row in the table.
```

```
134: ISAM error: no more locks
```

Zudem erfolgt im online.log ein Eintrag, welcher User wann das das Limit überschreiten wollte:

```
07/06/14 20:57:38 Session SID=113 User UID=1000 NAME=kalu  
PID=28307 has exceeded the session limit of 42000 locks.
```

TechTipp: infoshp – Anzeigen der Informationen eines Shapefiles

ESRI Shapefile (kurz: Shapefile) ist ein gängiges Datenformat zur Beschreibung von Vektordaten wie sie in Geographischen Informationssystemen (GIS) häufig genutzt werden.

Zur Analyse von Shapefiles steht bei INFORMIX das Programm „infoshp“ zur Verfügung. Damit können Dateien des Typs „.shp“, „.shx“ und „.dbf“ analysiert werden. Zugehörige Projektfiles „.prj“ werden automatisch zur Analyse hinzugezogen.

Am Beispiel eines im Internet frei verfügbaren Shapefiles (Quelle: diva-gis.org) wollen wir die Nutzung dieses Programms veranschaulichen:

1. Ausgabe der ESRI-Informationen eines Shapefiles:

```
infoshp -o info -f /home/kalu/IFX/SHAPEFILES/world.shp
```

```
Header info from /home/kalu/IFX/SHAPEFILES/world.dbf ...
```

```
File code           = 3
Year                = 95
Month               = 7
Day                 = 26
Number of records   = 250
Number of bytes in header = 225
Number of bytes in record = 75
Number of fields    = 6
```

Field	fnam	ftyp	flen	fdec
1	CODE	C	2	0
2	CNTRY_NAME	C	40	0
3	POP_CNTRY	N	10	0
4	CURR_TYPE	C	16	0
5	CURR_CODE	C	4	0
6	FIPS	C	2	0

```
Header info from /home/kalu/IFX/SHAPEFILES/world.shp ...
```

```
File code           = 9994
File length (16-bit words) = 205770
Version             = 1000
Shape type          = 5
Bounding box xmin   = -180.000000
Bounding box ymin   = -58.498611
Bounding box xmax   = 180.000000
Bounding box ymax   = 83.602203
Bounding box zmin   = 0.000000
```

```
Bounding box zmax          = 0.000000
Bounding box mmin         = 0.000000
Bounding box mmax         = 0.000000
```

Header info from /home/kalu/IFX/SHAPEFILES/world.shx ...

```
File code                  = 9994
File length (16-bit words) = 1050
Version                   = 1000
Shape type                 = 5
```

```
Bounding box xmin         = -180.000000
Bounding box ymin         = -58.498611
Bounding box xmax         = 180.000000
Bounding box ymax         = 83.602203
Bounding box zmin         = 0.000000
Bounding box zmax         = 0.000000
Bounding box mmin         = 0.000000
Bounding box mmax         = 0.000000
```

Coordinate system info from
/home/kalu/IFX/SHAPEFILES/world.prj ...

```
GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",
6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0
.017453292519943295]]
```

Nun interessiert uns, ob für diese Koordinaten ein Referenzsystem in der Tabelle „spatial_references“ existiert. Dies kann abgefragt werden mittels:

infoshp -o check -D ibm -f /home/kalu/IFX/SHAPEFILES/world.shp

... der erste Teil der Ausgabe ist identisch mit der obigen Ausgabe ...

Checking for qualified spatial references ...

```
SELECT
    srid,
    description,
    auth_name,
    auth_srid,
    falsex,
    falsey,
    xyunits,
    falsez,
    zunits,
    falsem,
    munits,
```

```

        srttext
FROM
        spatial_references

WHERE
        falsex <= -180.000000 AND
        falsey <= -58.498611 AND
        falsez <= 0.000000 AND
        falsem <= 0.000000 AND
        (180.000000 - falsex) * xyunits <= 9007199254740991
AND (83.602203 - falsey) * xyunits <= 9007199254740991
AND (0.000000 - falsez) * zunits <= 9007199254740991
AND (0.000000 - falsem) * munits <= 9007199254740991
AND srttext =
'GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984"
,6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",
0.017453292519943295]]'
ORDER BY srid;

```

No qualified spatial references found ... exiting.

Es wurden also keine Einträge in der Tabelle „spatial_refereces“ gefunden, die zu dieser Darstellung passen. Die fehlenden Einträge können mittels „create“ erstellt werden:

infoshp -o create -D ibm -f /home/kalu/IFX/SHAPEFILES/world.shp

```

INSERT statement to create new spatial reference ...
INSERT INTO spatial_references
(
        srid,
        description,
        auth_name,
        auth_srid,
        falsex,
        falsey,
        xyunits,
        falsez,
        zunits,
        falsem,
        munits,
        srttext
)
VALUES
(
        54054,

```

```

NULL,
NULL,
NULL,
-216.000000,
-72.708693,
22688159331841.289062,
0.000000,
9007199254740991.000000,
0.000000,
9007199254740991.000000,
'GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",
,6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",
0.017453292519943295]]' );

```

An dieser Stelle kommt eine Sicherheitsabfrage:

```
Do you want to execute the INSERT statement? (y/n) y
```

```
-----
SRID 54054 inserted ... exiting.
-----
```

Wird der Check erneut aufgerufen, so wird der eben eingefügte Wert gefunden:

```
-----
Qualified spatial reference ...
srid          = 54054
description   =
auth_name     =
auth_srid     =
falsex        = -216.000000
falsey        = -72.708693
xyunits       = 22688159331841.289062
falsez        = 0.000000
zunits        = 9007199254740991.000000
falsem        = 0.000000
munits        = 9007199254740991.000000
srtext        =
GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",
6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0
.017453292519943295]]

```

```
-----
One qualified spatial reference found ... exiting.
-----
```

Die Nutzung von Shapefiles im Zusammenspiel mit INFORMIX werden wir in der Ausgabe September des INFORMIX Newsletters an einem Beispiel veranschaulichen.

TechTipp: DBSpaces im Wachstum begrenzen – max_size in sysmaster

Um das Wachstum von DBSpaces, die extendable gesetzt wurden, zu begrenzen, wurde ein neues Argument im Task eingeführt, das die maximal erlaubte Grösse angibt. Diese Grösse wird dabei in der Tabelle „sysdbspaces“ der Datenbank „sysmaster“ eingetragen.

Beispiel: Der DBSpace „datadbs“ soll sich um jeweils 1 GB erweitern, bis zur maximalen Grösse von ca. 42 GB.

EXECUTE FUNCTION

```
task("modify space sp_sizes", "datadbs", 0, 1024000, 42000000);
```

Die Argumente dabei sind:

- Der Name des DBSpaces (hier „datadbs“)
- Die minimale Grösse (hier 0)
- Die Extend Size (in kB)
- Die maximale Grösse (in kB)

Wird die Maximale Grösse nicht angegeben, oder auf 0 gesetzt, so kann der DBSpace so lange wachsen, bis die Grenzen des Filesystems erreicht sind.

TechTipp: Tasks - reset config

Eine Reihe an Parametern der ONCONFIG können mittels „onmode -wm“ zur Laufzeit dynamisch geändert werden. Sollen diese Anpassungen rückgängig gemacht werden, dass wieder die Anfangswerte gesetzt sind, die beim Start der Instanz aktiv waren, so kann diese mittels eines Funktionsaufrufs des Tasks „reset config“ erfolgen.

Beispiel:

```
execute function task ('reset config', 'STMT_CACHE_HITS');
```

Setzt den Wert für STMT_CACHE_HITS wieder auf den Wert zurück, wie er beim Start der Instanz in der ONCONFIG stand.

Sollen alle Werte, die dynamisch verändert wurden, auf die Startwerte gesetzt werden, so kann dies erfolgen mittels:

```
execute function task ('reset config all');
```

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.iiug.org/intl/deu>

http://www.iug.de/index.php?option=com_content&task=view&id=95&Itemid=149

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

<http://www.drap.de/link/informix>

<http://www.nsi.de/informix/newsletter>

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

<http://www.listec.de/Newsletter/IBM-Informix-Newsletter/View-category.html>

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

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

Die Autoren dieser Ausgabe

Gerd Kaluzinski IT-Specialist Informix 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

Markus Holzbauer IBM Informix Advanced Support
IBM Software Group, Information Management Support
holzbauer@de.ibm.com

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Gerd Kaluzinski

(Brotzeit in Bregenz)