

Firebird® Version 1.5.2



Informationen zum Release v.152_09 11. Dezember 2004

B E S E I T I G T E R E G R E S S I O N E N

PROBLEM	SF Bug # Beseitigt von
<p>Blobs mit mehr als 65535 Segmenten wurden von GBAK nicht richtig gesichert.</p> <p><u>Lösung</u> Beseitigt in 1.5.2</p>	<p>v.1.0 Regression</p> <p>N. Samofatov</p>
<p>Falls ein SEGV Fehler (oder eine andere asynchrone Exception) von einer schlecht programmierten UDF geworfen wird, dann sollte der Server dessen Namen mitprotokollieren. Dieses Feature war in FB 1.5 nicht funktionsfähig.</p> <p><u>Lösung</u> Wiederhergestellt in v.1.5.2</p>	<p>v.1.5.0 Regression</p> <p>V. Horsun</p>
<p>In v.1.5.0: exit(3) wurde bei einem kritischen Fehler unter Windows aufgerufen, das die Verwendung des JIT Debuggers verhindert, um das Problem in UDF Routinen zu analysieren. Ein Fix in v.1.5.1 hatte zur Folge, dass der Debugger immer aufgerufen wurde. Dabei entstand jedoch ein neues Problem, nämlich dass der automatische Restart des Servers nicht mehr einwandfrei funktionierte.</p> <p><u>Lösung</u> V. 1.5.2 ruft nur dann den Debugger auf, wenn der BugcheckAbout Konfigurationsdateiparameter auf 1 gesetzt ist. Das Hauptziel dieses Fixes in v1.5.2 ist, zu vermeiden, dass Dr.Watson den Schliessen-Oder-Debug Dialog anzeigt.</p>	<p>Regression</p> <p>V. Horsun, N. Samofatov</p>
<p>Unterstützung für mehrdimensionale Arrayfelder war fehlerhaft.</p> <p><u>Lösung</u> Wiederhergestellt in v.1.5.2</p>	<p>v1.5.1 Regression</p> <p>C. Valderrama</p>

Zugriffspläne für Selectable Stored Procedures, die mehrere FOR Schleifen beinhalten, wurden im Vergleich zu v.1.5.0 in umgekehrter Reihenfolge ausgegeben.

v1.5.1
Regression

Lösung

Wiederhergestellt in v.1.5.2

V. Horsun, N.
Samofatov

Im Optimizer wurde ein Bug beseitigt, der seit v.1.5.0 vorhanden ist. Es handelt sich hierbei um bestimmte Fälle, wo kein optimierter Zugriff durchgeführt werden konnte, wenn eine Outer Join Syntax dazu verwendet wurde, um Inner Joins auszuführen.

v1.5.0
Regression

Beispiel 1

```
SELECT *
FROM RDB$RELATIONS r
  LEFT JOIN RDB$RELATION_FIELDS rf
    ON (1 = 1)
WHERE
  r.RDB$RELATION_NAME = rf.RDB$RELATION_NAME
```

FB1.5 gab folgenden Plan zurück:

```
PLAN JOIN (R NATURAL,RF NATURAL)
```

Es werden alle Datensätze von RDB\$RELATION_FIELDS abgerufen, wobei allerdings nur die Datensätze abgerufen werden müssten, die mit dem Kriterium r.RDB\$RELATION_NAME = rf.RDB\$RELATION_NAME übereinstimmen.

Beispiel 2

```
SELECT
  R.RDB$RELATION_NAME ,
  RF.RDB$RELATION_NAME
FROM
  RDB$RELATIONS R
  LEFT JOIN RDB$RELATION_FIELDS RF ON
    (RF.RDB$RELATION_NAME = R.RDB$RELATION_NAME
     AND RF.RDB$RELATION_NAME >= 'zzz')
```

Die Gleichheit wird fehlerhaft verteilt. Dies hat zur Folge, dass diese dem Outer Join zugeteilt werden, anstatt dem inneren Datenstrom.

Lösung

In 1.5.2, Beispiel 1 gibt diesen Plan zurück:

```
PLAN JOIN (R NATURAL,RF INDEX (RDB$INDEX_4))
```

Das Problem in Beispiel 2 ist ebenfalls durch diesen Fix beseitigt.

A. Brinkman

Die Dokumentation für den Windows Embedded Server, nämlich README.user.embedded im /doc Unterverzeichnis der Windows-Distribution - Abschnitt 2.2 "Database Access" - wurde in v.1.5.1 fehlerhaft "berichtigt".

Regression

Lösung

H. Borrie

Dieser Abschnitt war in v.1.5.0 unklar, in v.1.5.1 einfach falsch und ist nun korrekt.

B E S E I T I G T E F E H L E R

PROBLEM	SF Bug # Beseitigt von
<p>V. 1.5 Superserver hatte einen potentiellen, schwerwiegenden Fehler, der einen Deadlock verursachte, wenn gbak versuchte eine Datenbank in ein Verzeichnis zurück zu sichern, auf dessen der Zugriff über den Konfigurationsparameter DatabaseAccess in firebird.conf untersagt war.</p> <p><u>Lösung</u> Der Fix für diesen Fehler ist von Firebird2 HEAD nach v.1.5.2 portiert worden.</p>	<p>v.1.5.0 Fehler A. Peshkov</p>
<p>Die Datenbankverbindung von einem Windows XP SP2 Client zu einem Linux Server war langsam.</p> <p><u>Lösung</u> Gelöst.</p>	<p>1065511 N. Samofatov</p>
<p>Ein EXISTS oder SINGULAR Prädikat innerhalb einer Aggregatsfunktion hatte eine Fehlerkennung bzw. -abbildung bei aggregierenden Abfragen zur Folge.</p> <p><u>Lösung</u> Gelöst.</p>	<p>1063254 A. Brinkman</p>
<p>ISQL hatte ein paar Parsing-Probleme:</p> <ol style="list-style-type: none"> 1. Strichpunkte und '--' Sequenzen wurden in mehrzeiligen Literalen als Terminierungssymbole und Kommentare interpretiert 2. Tabulatoren wurden in Zeichenketten in Leerzeichen übersetzt <p><u>Lösung</u> Beseitigt</p>	<p>Nicht protokolliert D. Sibiryakov, C. Valderrama, N. Samofatov</p>
<p>Der Server konnte abstürzen, falls execute_immediate verwendete wurde, um eine Transaktion eines nicht vorhandenen Savepoints freizugeben (release) oder zurückzusetzen (rollback).</p> <p><u>Lösung</u> Beseitigt</p>	<p>Nicht protokolliert N. Samofatov</p>

Ein altes Problem, das uns seit geraumer Zeit begleitet, sollte beseitigt sein: 1045970
 Wenn ein Client Events registriert hat und dessen Netzwerkverbindung aus unerklärlichen Gründen beendet wird (Hardwareproblem, Resettaste, Task Manager), begann der Server mit dem Verbrauch von 100% CPU-Zeit, bis der "Ursprungsort" (Clientverbindung, die die API Funktion isc_que_events() aufgerufen hat) dessen Fehlschlag berichtet hat.

Dieser Bug betraf alle FB Versionen (mehr oder weniger, jedoch abhängig vom DummyPacketInterval Konfigurationsparameter) und nur TCP/IP Verbindungen.

Lösung D.Yemanov
 Dieses Problem sollte in v.1.5.2 beseitigt sein.

v.1.5.1 besitzt einen Bug im Optimizer, der ein unnötiges Abrufen von Datensätzen bei verbundenen Relationen zur Folge hat, falls "OR" auf einer Basisrelation angewendet wird, wie folgendes Beispiel veranschaulicht: Nicht protokolliert

```
SELECT * FROM RDB$RELATIONS r
LEFT JOIN RDB$RELATION_FIELDS rf
ON (r.RDB$RELATION_NAME = rf.RDB$RELATION_NAME)
WHERE
  r.RDB$RELATION_ID = 0 OR 1 = 0
```

Lösung A. Brinkman
 Beseitigt

UDF mit NULL Eingabeparameter. [Hier beschrieben](#). 544132

Lösung C. Valderrama
 Beseitigt

Problem mit UDF und NULL Deskriptoren bei Left Join. [Hier beschrieben](#). 728839

Lösung C. Valderrama
 Beseitigt

Fehler beim Versuch, Daten aus einer natürlich änderbaren View zu löschen, falls diese berechnete Ausdrücke beinhaltet. Wenn Sie eine View wie folgt hatten: Nicht protokolliert

```
create view v_test (f, s)
as
  select f1, f2 + f3
from t
```

dann gab der Server einen "attempted update of read-only column" Fehler zurück, wenn man versuchte, eine Löschoperation (DELETE) auszuführen.

Lösung D. Yemanov
 Dieses Fehlverhalten ist in FB 1.5.2 (und später) beseitigt.

Numerische Datentypen, die als Fließkommavariablen dargestellt sind, werden in einem EXECUTE STATEMENT in einer Dialekt 1 Datenbank inkorrekt verarbeitet. Das Ergebnis ist eine fehlerhafte Skalierung der Zahlen. Nicht protokolliert

```
create table a (b numeric(18,3));
commit;
insert into a values(12345.678);
commit;
set term ^;
create procedure c
  returns(d numeric(18,3))
as
begin
  for execute statement 'select b from a'
  into :d
  do suspend;
end
^
commit^
set term ;^
select * from c;
```

gibt 12.346 anstatt von 12345.678 zurück.

Lösung
Beseitigt

A. Peshkoff

Falls DISTINCT in einer Aggregierungsfunktion verwendet wird und die verarbeitete Ergebnismenge leer ist, dann hatte dies ein kleines Speicherleck zur Folge. Der Speicher würde erst mit einem Disconnect freigegeben werden. Nicht protokolliert

Diese Routine würde 120MB unter FB 1.5.1 (und früher) verbrauchen:

```
CREATE PROCEDURE MEM_LEAK
AS
  DECLARE I INT = 1000;
  DECLARE T INT;
  DECLARE C INT;
BEGIN
  WHILE (I > 0) DO
  BEGIN
    SELECT RDB$INDEX_TYPE,
    COUNT(DISTINCT RDB$RELATION_NAME)
    FROM RDB$INDICES
    WHERE 0=1
    GROUP BY 1
    INTO :T, :C;

    I = I - 1;
  END
END
```

Lösung
Beseitigt in v.1.5.2

V. Horsun

Ein weiteres Speicherleck wurde beseitigt, nämlich wenn eine Exception in einer Selectable Stored Procedure geworfen wird. Die prozedurale Anfrage wurde nicht freigegeben und hatte nach 750-1000 Iterationen Fehler wie "too many concurrent executions of the same request" zur Folge.

Nicht protokolliert

```
CREATE PROCEDURE P (INP INTEGER)
RETURNS (OUTP INTEGER)
AS
BEGIN
    OUTP = INP / 0;
    SUSPEND;
END
```

```
UPDATE T SET ID = 1
WHERE (SELECT OUTP FROM P(1)) = 1
```

Die Speicherblöcke wurden erst bei einem Disconnect freigegeben.

Lösung
Beseitigt

D. Yemanov

Es gab ein Problem mit der Deadlock-Erkennung, wenn das pessimistische Sperrverhalten (WITH LOCK Syntax) verwendet wurde.

Nicht protokolliert

```
create table test (id integer);
insert into test values(1);
insert into test values(2);
Commit;
```

Transaktion 1 (READ COMMITTED, WAIT):

```
select * from test
where id = 1 with lock;
```

Transaktion 2 (READ COMMITTED, WAIT):

```
select * from test
where id = 2 with lock;
select * from test
where id = 1 with lock;
```

Transaction 1:

```
select * from test
where id = 2 with lock;
```

Dieser Zustand hatte einen permanenten Deadlock zur Folge.

Lösung

N. Samofatov

Beseitigt. V.1.5.2 erkennt und berichtet solch einen Deadlock als Fehler.

Der Server wird beendet, falls NULL an EXECUTE STATEMENT ... INTO übergeben wird. Zum Beispiel,

Nicht protokolliert

```
...
VAR = NULL;
EXECUTE STATEMENT :VAR;
...
```

hatte zur Folge, dass der Server abstürzt.

Lösung
Beseitigt

A. Peshkoff

Das Server-Log wurde durch SIGPIPE Fehlermeldungen überschwemmt, wenn man SuperServer unter UNIX verwendete. Der "alte" InterBase Code protokollierte Sigmoid Fehler für SS unter *nix. Unglücklicherweise ist es jedoch so, dass beim Auftreten von Sigmoid Fehlern diese gleich zu Tausenden passieren (wenn diese auftreten). Dies hatte zur Folge, dass die Log-Datei ziemlich rasch anwuchs und in Extremfällen die komplette Partition mit Logdaten aufgefüllt wurde.

Nicht protokolliert

Lösung
Protokollierung von SIGPIPE Fehlermeldungen wurde deaktiviert.

P. Reeves

100% CPU-Auslastung wurde in bestimmten Situationen durch den cache_writer Thread hervorgerufen (berichtet von Adrianos dos Santos Fernandes). Um dies zu reproduzieren, öffnen Sie zwei Eingabeaufforderungen.

Nicht protokolliert

Eingabeaufforderung 1:

```
isql
CREATE DATABASE 'test.fdb';
CREATE TABLE T (N INTEGER);
EXIT;

gbak -B test.fdb test.fbk
del test.fdb
gbak -C test.fbk test.fdb
```

Eingabeaufforderung 2: isql test.fdb Eingabeaufforderung 1:

```
gbak -B test.fdb test.fbk
```

Der Server besitzt nun eine CPU-Auslastung von 99%, bis die Verbindung in Eingabeaufforderung 2 beendet wird. Dieser Bug trat nicht auf, wenn man bei der letzten Anweisung -GARBAGE_COLLECT verwendete.

Lösung
Beseitigt

V. Horsun

<p>Eine mögliche Ursache für einen Serverabsturz wurde im op_connect Handler gefunden. Wenn ein TCP/IP Paket (mit fehlenden Benutzerinformationen) am Serverport empfangen wurde, dann konnte der Server zum Absturz gebracht werden. Da es das erste Packet (op_connect) im Client-Server-Protokoll war, konnte es den Server einer beliebigen DoS Attacke aussetzen. Theoretisch konnte somit jeder den Server mit nur einem TCP Paket zum Absturz bringen.</p>	<p>Nicht protokolliert</p>
<p><u>Lösung</u> Beseitigt</p>	<p>A. Karyakin, D. Yemanov</p>
<p>Der Server konnte mit einer komplexen Abfrage, in der eine Vielzahl von Streams in einem Sort/Merge verwendet wurden, zum Absturz gebracht werden. Ein komplexes Union mit vielen Aggregaten und Merge Joins konnte den Server aufgrund eines Stream-Buffer-Overflows zum Absturz bringen. Obwohl das aktuelle Limit bei 255 Streams pro Anfrage liegt, konnte der temporäre Puffer nur 128 Einheiten aufnehmen.</p>	<p>Nicht protokolliert</p>
<p><u>Lösung</u> Beseitigt</p>	<p>D. Yemanov</p>
<p>Der Server blockierte, falls Events mit Network Access Translation (NAT)-Gateways verwendet wurden. Hilfsverbindungen (für Events) wurden durch die Clientbibliothek unter Verwendung der TCP/IP Adresse des Servers initiiert. Die Serveradresse konnte jedoch eine andere sein, falls sich der Server hinter einer NAT-Box befindet.</p>	<p>Nicht protokolliert</p>
<p><u>Lösung</u> Dieses Problem wurde dadurch beseitigt, dass nun die Adresse verwendet wird, die für die Hauptverbindung verwendet wurde und nicht die Adresse, die vom Server zurückgegeben wird.</p>	<p>C. Waters, D. Yemanov</p>
<p>Der Sweep-Prozess gibt seine Sperren nicht frei, falls ein Database Shutdown ausgeführt wird. Ein Serverabsturz konnte auftreten, wenn ein Database Shutdown initiiert wurde, während ein Sweep lief.</p>	<p>Nicht protokolliert</p>
<p><u>Lösung</u> Beseitigt</p>	<p>V. Horsun</p>
<p>Die Bits mit dem niedrigsten Stellenwert eines Fließkommawertes gingen verloren, falls eine Rundung zu einem Integer oder Int64 durchgeführt wurde.</p>	<p>Nicht protokolliert</p>
<p>Dialekt 3 Datenbank:</p> <pre> SELECT CAST(CAST(1.005E0 AS NUMERIC(15,2)) AS VARCHAR(30)) FROM RDB\$DATABASE UNION ALL SELECT CAST(CAST(1.015E0 AS NUMERIC(15,2)) AS VARCHAR(30)) FROM RDB\$DATABASE UNION ALL SELECT CAST(CAST(1.025E0 AS NUMERIC(15,2)) AS VARCHAR(30)) FROM RDB\$DATABASE UNION ALL </pre>	

```
SELECT CAST(CAST( 1.035E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$DATABASE
UNION ALL
SELECT CAST(CAST( 1.045E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$DATABASE
UNION ALL
SELECT CAST(CAST( 1.055E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$DATABASE
UNION ALL
SELECT CAST(CAST( 1.065E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$DATABASE
UNION ALL
SELECT CAST(CAST( 1.075E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$DATABASE
UNION ALL
SELECT CAST(CAST( 1.085E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$DATABASE
UNION ALL
SELECT CAST(CAST( 1.095E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$DATABASE
```

FB 1.5.1 gibt folgendes zurück:

```
F_1
----
1.00
1.01
1.03
1.04
1.05
1.05
1.06
1.08
1.09
1.10
```

Lösung

V. Horsun

Beseitigt. FB 1.5.2 gibt folgendes zurück:

```
F_1
----
1.01
1.02
1.03
1.04
1.05
1.06
1.07
1.08
1.09
1.10
```

Ein paar Speicherzugriffsprobleme wurden entdeckt, während HEAD unter Valgrind getestet wurde.

Nicht protokolliert
N. Samofatov

Lösung

HEAD Fixes wurden in v.1.5.2 eingearbeitet.

64-bit SuperServer Builds auf Plattformen wie Linux/AMD64/NPTL, die höherwertige Bits einer 64-bit Thread ID verwenden, wurden mit Laufzeitfehlern konfrontiert.	Nicht protokolliert
<u>Lösung</u> Beseitigt	N. Samofatov
CURRENT_TIMESTAMP hatte unvorhersagbare Ergebnisse auf 64-bit Plattformen zur Folge.	Nicht protokolliert
<u>Lösung</u> Beseitigt	N. Samofatov

KLEINERE ERWEITERUNGEN

ERWEITERUNG	SF Bug # Beseitigt von
Performanceverbesserungen beim Überprüfen von Zugriffsberechtigungen	n/a
<u>Lösung</u> Die Ressourcenliste zum Überprüfen der Zugriffsberechtigungen wird nun On-The-Fly (bei Bedarf) berechnet. Für komplexere Schemata wurde hiermit der Speicher- und CPU-Bedarf erheblich verringert.	N. Samofatov, D. Urban
Änderungen bei POSIX Build und Paketerstellung	n/a
<u>Lösung</u>	N. Samofatov
<ol style="list-style-type: none"> 1. Workarounds für Bugs in GCC 3.3.2 und 3.3.3 2. Support für GCC 3.4 Build 3. Limitierung der Exports der Firebird-Bibliotheken unter Verwendung eines Versionskripts 4. Linken der Client- und UDF-Bibliotheken mit POSIX Threads. Dies beseitigt Probleme mit Single-Threaded Hostanwendungen wie PHP mit Verlinkung zu libfbclient.so der CS Distribution. 	
Weitere Änderungen bei POSIX Build und Paketerstellung	1027636
<u>Lösung</u>	D. Mullins
<ul style="list-style-type: none"> ● Um zu verhindern, dass der Startup-Status von der nächsten Statusmeldung überschrieben wird, muss das /etc/init.d/firebird Skript eine Zeile aufweisen, die nur "echo" nach RETVAL=\$? beinhaltet. ● Erik LaBianca erweiterte das Firebird Build-System dahingehend, dass die Quellcodepakete in generischer Form und ohne Abhängigkeit zu autoconf generiert werden können. Er verwendet diese Vorgehensweise für seine Fedora Core Pakete. 	E. S. LaBianca
Änderungen am Deklarationsskript der Standard ib_udf Bibliothek	n/a

Die Default-Deklarationen der String-Funktionen in `ib_udf.sql` akzeptieren nun Strings mit bis zu 255 Zeichen

64 - BIT RELEASES

von Nickolay Samofatov

Eine minimale Anzahl an Änderungen, die notwendig sind, um voll-funktionstüchtige 64-bit Builds zu erzeugen, wurden in die Firebird 1.5.2 Serie aus dem Firebird 2 Entwicklungsbaum eingearbeitet. Die resultierenden Builds sind schnell und können in Produktionsumgebungen eingesetzt werden. Protokollkompatibilität ist vollständig gewährleistet: 32-bit Clients können mit einem 64-bit Server kommunizieren und umgekehrt.

Die allgemeine Empfehlung ist jedoch, mit Vorsicht vorzugehen, wenn man diese Builds einsetzt.

Bekannte Probleme

1. Die Typen der Public API Handles werden in der 2.0 Serie vermutlich von 64-bit Pointer auf 32-bit Integer geändert. Dies bedeutet, dass 64-bit Clientanwendungen neu übersetzt werden müssen, um mit Firebird 2.0 Clientbibliotheken zusammenarbeiten zu können.
2. Da 64-bit Builds eine 64-bit Datenausrichtung verwenden müssen, ist die On-Disk Structure (ODS10) derzeit nicht die selbe für 64-bit und 32-bit Builds.

Es wird erwartet, dass diese Situation in Firebird 2.0 gelöst sein wird, welche die selbe 64-bit On-Disk Structure (ODS11) für sowohl 32-bit als auch 64-bit Builds der Engine verwenden werden.

3. Einige Dinge in der Linux x64 (aka AMD64) Welt sind noch immer ziemlich unausgereift (flaky). Viele 2.6 Kernels, inklusive des Letzten zum Zeitpunkt des Verfassens dieses Dokumentes (2.6.9), haben eine 64-bit Interprocess Communications (IPC) Einrichtung, die in die eine oder andere Richtung fehlerhaft arbeitet. Dies ist vor allem dann der Fall, wenn Sie eine SMP oder NUMA Maschine verwenden.
 4. Bevor Sie ein Problem berichten, das ähnlich einem Serverstillstand oder vorübergehenden Blockierungen ist, versuchen Sie:
 - ein Downgrade des Kernels auf 2.4 (2.4 SMP Kernels sind bekannt dafür, dass diese wie erwartet funktionieren)
 - oder
 - verwenden Sie den Non-Preemptible 2.6 Kernel, mit deaktiviertem SMP/NUMA Support.
-

Copyright © 2000-2004, Firebird Project.
Firebird® is a registered trademark of FirebirdSQL Foundation