

Firebird® Version 1.5.2



Notes de version v.152_09
11 Décembre 2004

REGRESSIONS CORRIGÉES

PROBLEME	SF Bug # Corrigé par
<p>Les blobs contenant plus de 65535 segments n'étaient pas sauvegardés correctement par GBAK.</p> <p><u>Solution</u> Corrigé dans 1.5.2</p>	<p>v.1.0 Régression N. Samofatov</p>
<p>Quand une erreur SEGV (ou une autre exception asynchrone) est retournée par une UDF mal écrite, le serveur devrait enregistrer son nom. Cette fonctionnalité était cassée dans FB 1.5.</p> <p><u>Solution</u> Rétabli dans v.1.5.2</p>	<p>v.1.5.0 régression V. Horsun</p>
<p>Dans v.1.5.0, exit(3) était appelé pour les erreurs critiques sous Windows, empêchant l'utilisation du débogueur JIT pour analyser les problèmes dans les routines UDF. Une correction dans v.1.5.1 appelait le débogueur systématiquement, ce qui créait un problème avec le redémarrage automatique du serveur.</p> <p><u>Solution</u> V. 1.5.2 appelle maintenant le débogueur que si BugcheckAbort est configuré à 1 dans le fichier de configuration. Le but principal de la correction dans v1.5.2 est d'éviter l'apparition de la fenêtre Dr.Watson.</p>	<p>Régression V. Horsun, N. Samofatov</p>
<p>Le support des champs tableaux multi-dimensionnels était cassé</p> <p><u>Solution</u> Rétabli dans v.1.5.2</p>	<p>v1.5.1 régression C. Valderrama</p>

Les plans pour les procédures stockées sélectionnables contenant de multiples boucles FOR étaient rapportés dans un ordre inverse, comparé avec v.1.5.0.

v1.5.1
régression

Solution

Rétabli dans v.1.5.2

V. Horsun, N.
Samofatov

Une régression de l'optimiseur, présente depuis v.1.5.0, fait que dans certains cas l'optimisation était mal faite quand la syntaxe outer join était utilisée pour des inner joins.

v1.5.0
régression

Exemple 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 renvoyait ce plan:

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

RDB\$RELATION_FIELDS scrutait tous les enregistrements, alors qu'il aurait du scruter que ceux qui correspondaient à r.RDB\$RELATION_NAME = rf.RDB\$RELATION_NAME

Exemple 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 >= ' z z z ')
```

Les égalités étaient distribuées de manière éronnée.

Solution

Dans 1.5.2, l'exemple 1 retourne ce plan:

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

Le problème de l'exemple 2 est aussi corrigé.

Dans la documentation du Serveur Windows Embedded, README.user.embedded dans le sous répertoire /doc de l'installation Windows, la section 2.2 "Database Access" a été "corrigée" à tort dans v.1.5.1.

Régression

Solution

Cette section n'était pas claire dans v.1.5.0. Dans v.1.5.1 c'était faux. C'est maintenant correct.

B U G S C O R R I G E
SOLUTION**SF Bug #
Corrigé par**

Le Superserveur V. 1.5 avait un sévère bug potentiel qui créait un verrou mortel si gbak essayait de restaurer une base dans un répertoire dont l'accès était interdit par le paramètre DatabaseAccess de firebird.conf.

bug v.1.5.0

Solution

La correction de ce bug a été reportée depuis la branche Firebird2 HEAD vers la v.1.5.2.

A. Peshkov

Un client Windows XP SP2 avait une connection lente vers un serveur Linux.

1065511

Solution

Résolu.

N. Samofatov

Un prédicat EXISTS ou SINGULAR dans une fonction d'aggregat créait une mauvaise détection pour la requête d'aggregat.

1063254

Solution

Résolu.

A. Brinkman

ISQL avait quelques problèmes d'analyse syntaxique:

Non enregistré

1. Les points-virgules et les séquences '--' étaient interprétés comme des terminaisons et des commentaires, respectivement, dans des expressions multi-lignes littérales
2. Les tabulations dans les chaînes littérales étaient converties en espaces

Solution

Tous ces problèmes sont corrigés.

D. Sibiryakov,
C. Valderrama,
N. Samofatov

Le serveur pouvait connaitre un crash quand execute_immediate était utilisé pour liberer ou annuler (rollback) une transaction vers un savepoint non existant.

Non enregistré

Solution

Maintenant corrigé.

N. Samofatov

Un ancien bug hérité continuait à exister, quand un client avait des événements enregistrés et que sa connexion réseau se terminait anormalement (problème matériel, reset ou autre arrêt intempestif), alors le serveur utilisait 100% du temps CPU tant que le port "parent" (la connection cliente qui a appelé la routine API `isc_que_events()`) n'annonçait pas son arrêt.

1045970

Ce bug affectait toutes les versions de FB (plus ou moins, selon la configuration de l'option `DummyPacketInterval`) et que les connexions TCP/IP.

Solution

D.Yemanov

Le problème semble réglé dans la v.1.5.2.

Dans v.1.5.1, un bug de l'optimiseur créait des "fetches" non nécessaires dans les relations jointes quand "OR" était utilisé dans la relation de base, comme dans l'exemple suivant :

Non enregistré

```
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
```

Solution

A. Brinkman

Corrigé

UDF avec paramètre NULL en entrée. [Expliqué ici](#).

544132

Solution

C. Valderrama

Corrigé

Une jointure gauche troublait l' UDF en dénaturant un descripteur null.

728839

[Expliqué ici](#).Solution

C. Valderrama

Corrigé

Erreur sur tentative de suppression d'enregistrement depuis une vue naturellement modifiable qui contient une expression calculée. Si vous aviez une vue comme :

Non enregistré

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

alors le serveur retournait l'erreur "attempted update of read-only column" quand vous vouliez faire une opération DELETE.

Solution

D. Yemanov

Corrigé dans FB 1.5.2 et suivantes.

Les types de données numériques représentés par des variables en virgule flottante étaient mal interprétés dans un "EXECUTE STATEMENT" avec une base en dialect 1 --- les nombres étaient mal évalués:

Non enregistré

```
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;
```

renvoyait 12.346 au lieu de 12345.678.

Solution

A. Peshkoff

Corrigé

Si DISTINCT était utilisé dans une fonction d'agrégat et que l'enregistrement concerné (agrégé) est vide, alors nous avons une fuite de mémoire. Cette mémoire n'était libérée qu'à la déconnexion.

Non enregistré

Cette routine pouvait consommer 120MB avec FB 1.5.1 et précédentes:

```
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
```

Solution

V. Horsun

Corrigé dans v.1.5.2

Le serveur avait une fuite de ressources quand une exception était levée via une procédure stockée sélectionnable. Le requête de la procédure n'était pas correctement libérée et causait des erreurs du type "too many concurrent executions of the same request" après 750-1000 iterations. Not logged

```
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
```

Les ressources étaient libérées après la déconnexion.

Solution
Corrigé

D. Yemanov

Il y avait un problème avec la détection de verrous mortels quand les verrous pessimistes (WITH LOCK) étaient utilisés. Non enregistré

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

Transaction 1 (READ COMMITTED, WAIT):

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

Transaction 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;
```

Cet ensemble de conditions créerait un verrou mortel permanent.

Solution

N. Samofatov

Corrigé. V.1.5.2 detecte et rapporte de tel verrous mortels comme une erreur.

Le serveur crash quand NULL était passé à EXECUTE STATEMENT ... INTO. Par exemple,

Non enregistré

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

...
fait tomber le serveur.

Solution
Corrigé

A. Peshkoff

Les log du serveur étaient pollués par les erreurs SIGPIPE quand on utilisait SuperServer UNIX. Le code hérité d'InterBase enregistrait les erreurs sigpipe pour SS tournant sous *nix. Malheureusement les erreurs sigpipe peuvent arriver par milliers (quand elles arrivent) ce qui faisait grossir le fichier log très vite. Dans des cas extrêmes on pouvait remplir une partition entière.

Non enregistré

Solution
L'enregistrement des erreurs SIGPIPE a été désactivé.

P. Reeves

Dans certains rares cas le thread cache_writer pouvait utiliser 100% CPU (rapporté par Adrianos dos Santos Fernandes). Pour le reproduire, ouvrir deux lignes de commandes.

Non enregistré

prompt1:

```
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
```

prompt2:

```
isql test.fdb
```

prompt1:

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

Le serveur va consommer 99% du CPU tant que isql t prompt2 ne sera pas déconnecté. Ce bug n'existe pas avec l'option -GARBAGE_COLLECT dans la dernière commande.

Solution
Corrigé

V. Horsun

Une cause possible de crash serveur a été découverte avec op_connect. Quand un paquet TCP/IP sans information utilisateur était reçu sur le port du serveur, le serveur pouvait tomber. Parce que c'était le premier paquet (op_connect) dans le protocole client-server, cela exposait le serveur a une attaque DoS. N'importe qui pouvait faire tomber le serveur avec un seul paquet TCP.

Non enregistré

Solution
Corrigé

A. Karyakin, D. Yemanov

Le serveur pouvait tomber avec des requêtes complexes où de nombreux flux étaient utilisés dans un tri/fusion. Une union complexe avec de nombreux agrégats et des jointures pouvait faire tomber le serveur du fait de buffer overflow. Même si la limite actuelle est de 255 flux par requête, le tampon temporaire ne pouvait gérer que 128 items.

Non enregistré

Solution
Corrigé

D. Yemanov

Le server bloquait les événements (events) quand ils étaient utilisés avec des gateways NAT (Network Address Translation). Les connexions auxiliaires (pour les événements) étaient établies par la bibliothèque cliente en utilisant l'adresse TCP/IP rapportée par le serveur. Mais l'adresse retournée par le serveur était incorrecte s'il était derrière une boîte NAT.

Non enregistré

Solution

La correction était d'utiliser l'adresse qui était utilisée pour le socket principal, pas l'adresse rapportée par le serveur.

C. Waters, D. Yemanov

Le sweeper ne relachait pas son verrou quand un shutdown de la base était exécuté. Un crash du serveur pouvait arriver quand un shutdown de la base était initié pendant qu'un sweep était en cours.

Non enregistré

Solution
Corrigé

V. Horsun

Les bits les moins significants d'une valeur en virgule flottante étaient perdus quand on arrondissait la valeur vers un integer ou un int64.

Non enregistré

Base en dialect 3:

```
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$DATABASE
UNION ALL
SELECT CAST(CAST( 1.035E0 AS NUMERIC(15,2))
  AS VARCHAR(30)) FROM RDB$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 retourne

```

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

```

Solution

V. Horsun

Corrigé. FB 1.5.2 retourne

```

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

```

Quelques problèmes d'accès mémoire ont été détectés en testant la branche HEAD avec Valgrind.

Non enregistré

Solution

N. Samofatov

Les corrections de la branche HEAD ont été rapportées à v.1.5.2

La construction de SuperServer 64-bit sur des plateformes Linux/AMD64/NPTL, qui utilisent l'ordre haut des ID des threads 64-bit (high-order bits of a 64-bit thread ID), rapportait des erreurs run-time.

Non enregistré

<u>Solution</u> Corrigé	N. Samofatov
----------------------------	--------------

CURRENT_TIMESTAMP donnait des résultats imprévisibles sur les plateformes 64-bit.	Non enregistré
-----------------------------------------------------------------------------------	----------------

<u>Solution</u> Corrigé	N. Samofatov
----------------------------	--------------

AMELIORATIONS MINEURES

PROBLEME	SF Bug # Corrigé par
-----------------	---------------------------------

Amélioration des performances pour la vérification des permissions	n/a
--------------------------------------------------------------------	-----

<u>Solution</u> La liste des ressources est maintenant calculée à la volée quand nécessaire. Pour des schémas complexes, cela réduit significativement la consommation mémoire et temps CPU.	N. Samofatov, D. Urban
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------

Changement des constructions POSIX et empaquetage	n/a
---------------------------------------------------	-----

<u>Solution</u> 1. Correction des bugs avec GCC 3.3.2 et 3.3.3 2. Construction possible avec GCC 3.4 3. Limite des exports des bibliothèques de Firebird utilisant un script de version 4. Lien des blibliothèques clientes et UDF avec les threads POSIX. Cela corrige les problèmes avec les hotes mono-thread comme PHP liés avec libfbclient.so des paquets CS	N. Samofatov
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------

Encore des changements dans la construction POSIX et l'empaquetage	1027636
--------------------------------------------------------------------	---------

<u>Solution</u> ● Pour éviter que le statut de démarrage soit surchargé par le message suivant , le script /etc/init.d/firebird a besoin d'une ligne constituée uniquement de "echo" après RETVAL=\$?. ● Erik LaBianca a étendu le système de construction de Firebird pour générer des bundles source de manière générique et sans dépendance avec autoconf. Il a utilisé cette facilité avec ses paquets Fedora Core.	D. Mullins E. S. LaBianca
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------

Changements dans le script de déclaration de la bibliothèque standard ib_udf	n/a
------------------------------------------------------------------------------	-----

<u>Solution</u> Les declarations des routines de manipulations de chaînes dans ib_udf.sql on été changés pour accepter des chaînes de plus de 255 caractères	N. Samofatov
-----------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------

VERSIONS 64-BIT

par Nickolay Samofatov

Un nombre réduit de changements on été nécessaire pour construire des versions 64-bit entièrement fonctionnelles. Ces changements on été porté depuis la branche Firebird 2 vers Firebird 1.5.2. Les binaires produits sont rapides et utilisables en production. La compatibilité de protocole est complète: des clients 32-bit peuvent converser avec un serveur 64-bit servers et vice versa.

Toutefois, ces recommandations générales sont à considérer avant de déployer ces binaires.

Problèmes connus

1. Les handles de l'API publique vont changer dans la série 2.0 passant de pointeurs 64-bit à des integers 32-bit. Cela signifie qu'une application cliente 64-bit devra être recompilée pour fonctionner avec une bibliothèque cliente Firebird 2.0 .
2. Parce que les binaires 64-bit doivent utiliser un aligement de données 64-bit, la On-Disk Structure (ODS10) n'est pas la même pour les binaires 64-bit et 32-bit.

Cette situation devrait être résolue dans Firebird 2.0 qui devrait avoir la même 64-bit On-Disk Structure (ODS11) pour les versions 32-bit et 64-bit du moteur.

3. Quelques problèmes inconsistants subsistent dans le monde Linux x64 (aka AMD64). Plusieurs noyaux 2.6, y compris le dernier testé quand ce document a été écrit (2.6.9) ont des fonctionnalités 64-bit Interprocess Communications (IPC) pas vraiment fonctionnelles. Ceci est particulièrement vrai si vous avez une machine SMP ou NUMA.
4. Avant de rapporter un problème de blocage serveur essayez
 - de revenir à un noyau version 2.4 (les noyaux SMP sont connus pour bien fonctionner)
 - ou
 - testez avec un noyau 2.6 non-preemptible 2.6, avec le support SMP/NUMA désactivé.

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