

Firebird Versão 1



Notas de Lançamento

Março de 2002

Índice

[Introdução](#)

[Compatibilidade com Versões Anteriores](#)

[Novas Funcionalidades](#)

[Extensões à Linguagem](#)

❑ [Novas Palavras Reservadas\("Reserved Words"\)](#)

[ISQL Funcionalidades](#)

[Funções Externas \(UDFs\)](#)

❑ [Livraria ib_udf](#)

❑ [Livraria BUDEF UDF \(usando Descriptors\)](#)

[Novos Parâmetros de Configuração](#)

[Melhoramentos no API](#)

[Notas de Instalação](#)

❑ [Windows 32-bit](#)

❑ [Linux/UNIX](#)

❑ [Solaris](#)

❑ [MacOS X](#)

❑ [FreeBSD](#)

❑ [Driver Pure Java JCA-JDBC XA-Capable](#)

❑ [Driver InterClient JDBC](#)

[Mais Informação](#)

[Documentação](#)

[Bugs Corrigidos](#)

[Problemas Conhecidos](#)

Introdução

A base de dados Firebird foi desenvolvida por um grupo independente de programadores voluntários a partir do código fonte do InterBase(tm), disponibilizado pela Borland ao abrigo da "InterBase Public License v.1.0" em 25 de Julho de 2000.

Esta versão foi compilada a partir de código que foi sujeito a uma extensa limpeza, eliminação de "bugs" e várias versões "beta", durante o ano que passou entre a disponibilidade do código fonte e esta versão. A "On-Disk Structure" é ainda a ODS 10. Algumas funcionalidades que obrigam a alterar a ODS foram adiadas para uma versão posterior.

Uma lista dos "bugs" que foram eliminados está disponível neste documento.

Foram adicionadas algumas extensões à linguagem: o seu sintaxe está patente no capítulo "Extensões à Linguagem" neste mesmo documento.

Os binários do Firebird podem ser carregados a partir do domínio FirebirdSQL.org -

<http://www.firebirdSQL.org>

Refira ao capítulo "Documentação" no final destas Notas de Lançamento para obter a localização da documentação recomendada.

Compatibilidade com Versões Anteriores

Se está a planear experimentar o Firebird utilizando uma base de dados InterBase existente, com a intenção de posteriormente reverter de novo para InterBase, tome todas as precauções fazendo um "backup" da sua versão actual e trabalhe com o "restore" da mesma para os seus testes.

Existiram alterações na livraria standard **ib_udf** descritas no capítulo "Extensões À Linguagem".

Qualquer base de dados InterBase será 100% "upwardly" compatível com este software desde que se sigam as regras normais de migração de servidores InterBase 4.x ou 5.x (ODS menor que versão 10):

- ❑ Criação de uma cópia de segurança "TRANSPORTABLE" do ficheiro InterBase utilizando a versão do utilitário gbak (gbak.exe em Windows) que é adequado à ODS da base de dados existente. Recomenda-se que este ficheiro tenha a extensão ".gbk". Para as versões 4 e 5 de Windows, o utilitário "ibmgr32.exe" possui um interface gráfico para o "gbak.exe" no menu em "Tasks | Backup". NÃO utilize este utilitário para operar sobre uma base de dados Firebird ou InterBase 6 (ODS 10).
- ❑ Se necessário, a cópia de segurança pode ser "zipada" e movida por intermédio de um qualquer meio amovível. Não esqueça que ficheiros gravados e posteriormente recolhidos de um CD-ROM terão de ter o atributo "Read-Only" removido.
- ❑ Utilizando a versão do "gbak" da instalação do Firebird, faça um "restore" da cópia de segurança para a localização de onde deseja correr a sua base de dados Firebird. Esta directoria deverá estar localizada na mesma máquina física que o programa "ibserver".
- ❑ A sua base de dados está convertida para correr como uma base de dados Firebird com Dialecto 1.
O "Operations Guide" da documentação contém o sintaxe para o utilitário "gbak".

Alguns utilitários, como o IBConsole, providenciam um interface gráfico que permite elaborar/repor cópias de segurança de uma base de dados com ODS 10. O utilitário recomendado é o IBBackup, um binário "freeware" de Martin Schmid <M.Schmid@EQUITANIA.de>, que pode ser descarregado a partir de <http://www.equitania.de/interbase/downloads/ibBackup.zip>.

Programas Clientes de Base de Dados ("Database Desktop Client Programs")

Nos meses que antecederam o lançamento da versão "open-source" independente Firebird, a comunidade Interbase Borland encerrou os esforços em relação ao desenvolvimento do IBConsole "open-source". Os responsáveis do IBConsole declinaram qualquer hipótese de aplicar "patches" para suportar todas as correcções e novas funcionalidades que o Firebird teve ao longo destes 19 meses de desenvolvimento. Por esta razão, não existe qualquer garantia que toda a funcionalidade do IBConsole funcione com o Firebird.

Existem algumas alternativas excelentes que estão indicadas na página "Contributed Downloads" em <http://www.ibphoenix.com>. Algumas são "open-source", outras são "freeware", e outras são produtos comerciais reconhecidos.

Dois programas Clientes são recomendados:

- ❑ O IB_SQL de Jason Wharton, que está disponível gratuitamente em <http://www.ibobjects.com>. Este software providencia uma vasta gama de utilitários para administração de base de dados, "scripting", "data-pumping", níveis de segurança, etc.
- ❑ O IBQuery de Michael Mutl, também disponível gratuitamente, é uma caixa de ferramentas SQL com um GUI elegante e um "small footprint". Pode ser obtido através do Torrys website em <http://www.torry.net/apps/utilities/database/mitecibquery.zip>.

Outros utilitários gratuitos:

- ❑ IBAccess, também ele gratuito, desenvolvido com componentes IBX, possui versões quer para Windows quer para Linux. Pode ser obtido em <http://www.ibaccess.org/>. Entre em contacto com o autor, Toni Martir (o seu endereço electrónico está no "web site") se encontrar algum problema de compatibilidade. Trata-se de um projecto "open-source" sem qualquer afinidade com a Borland, pelo que qualquer incompatibilidade que surja é provável de ser corrigida.
- ❑ O Marathon 2 está neste momento sobre "open-source" com a tutela de Tilo Muetze. Veja o website do projecto em <http://www.alanti.net/firebird/marathon/>

Componentes para Delphi/Borland C++Builder

Algumas dificuldades podem ser esperadas com a utilização dos componentes Borland InterBaseXpress (IBX) e DBExpress (DataSnap nas versões 6) para Delphi e C++Builder. É também muito pouco provável que a Borland repare ou melhore o seu driver BDE para InterBase 6 de forma a que este suporte a linguagem do Firebird e as suas extensões à API.

Os utilizadores de Delphi e C++Builder possuem duas alternativas poderosas à utilização do IBX ou BDE, ambas com um excelente suporte do programador:

- ❑ Os IB Objects de Jason Wharton em <http://www.ibobjects.com>
- ❑ O FIBPLus em <http://www.devrace.com>

"Version Strings" para as versões de Firebird

As "Version strings" são mantidas no Firebird 1.0 e retornadas da mesma forma de sempre, i.e.:

"LI-V6.2.0.nnn Firebird" (onde nnn indica o numero do "build").

Desta forma o caminho está aberto para o sistema de conexões múltiplas do futuro Firebird 2:

- ❑ O servidor Firebird em modo nativo irá escutar uma porta diferente (provavelmente a 3051) e retornar uma "version string" da forma "LI-V2.0.0.XXX Firebird".
- ❑ Irá possuir uma "opção de compatibilidade" onde irá escutar a porta 3050 e retornar uma "version string" compatível (possivelmente "LI-V6.2.0.368 Firebird"). Esta "string" de compatibilidade irá muito provavelmente ser configurável via definição no "Registry" ou num ficheiro de configuração.

Novos itens no parâmetro isc_database_info - isc_vendedor_info e isc_info_fb_version - irão permitir a uma aplicação cliente detectar se está conectada a um servidor Firebird ou a um servidor Borland Interbase, e saber a que versão do Firebird.

InterClient e Conectividade JDBC

A equipa do Firebird-Java está a implementar um driver "pure Java" para o Firebird. Este driver é baseado tanto no novo standard JCA para conexões a servidores de aplicações para "enterprise information systems" e o bem conhecido standard JDBC. Para detalhes, consulte o capítulo de instalação.

Versões Linux

Todas as versões Linux do Firebird estão compiladas com glibc 2.2. Antigas versões do Linux, como a RH 6.2, Debian etc., necessitarão de um "upgrade". Paul Reeves e Pavel Cisar estão a investigar as possibilidades de compilações com 32-bit I/O também funcionarem com glibc 2.1, para que sejam fornecidas como versões "adicionais" para estas plataformas Linux antigas.

Uma "client-library" para Linux **gdslib.so**, "thread-capable", está incluída nos últimos "builds".

Novas Funcionalidades

Melhor “Dependency Checking”

UDFs e Generators são neste momento verificados, de forma que não é possível a utilização do comando “drop” sobre os mesmos se estes estiverem a ser utilizados por uma “procedure” ou por uma “computed column”.

Suporte a “Large Database File”

(Sean Leyne, David Trudgett)

Uma base de dados lógica pode ser constituída por um ficheiro primário e inúmeros ficheiros secundários (O recorde conhecido para uma única base de dados FB/IB é de 980GB!). Porém, esta nova funcionalidade aplica-se apenas a base de dados de apenas um único ficheiro.

O tamanho máximo deste ficheiro envolveu três factores distintos:

- ❑ a implementação (em código) no FB/IB de funções I/O
- ❑ o suporte do sistema operativo para operações de ficheiros de 64-bit
- ❑ os limites do tamanho de ficheiro impostos pelo “filesystem”

Suporte para Operações de Ficheiro I/O de 64-bit

Todas as versões do Firebird/IB anteriores à versão 1.0 do FB (incluindo a versão 6.0 do IB) utilizavam tanto chamadas de 32-bit ao API do Sistema Operativo, assim como estruturas de ponteiros de ficheiros de 32-bit (inteiros). A capacidade do “engine” estava assim limitada a manusear “handles” de ficheiros com um tamanho máximo de 4Gb. No código Unix estas estruturas estavam definidas como inteiros (signed), limitando assim as implementações Unix e Linux a ficheiros de 2Gb.

MS Windows

“By design”, todas as versões Win32 (Win9x/ME/NT/2000) possuem suporte completo para “64-bit pointer file operations”, desde que a correcta estrutura de “file pointer” seja passada. Para adicionar o suporte de “large file” ao Firebird apenas foi necessário alterar a estrutura de “file pointer” para LARGE_INTEGER assim como pequenas modificações nas chamadas I/O.

Linux

Para Linux, operações de ficheiros de 64-bit são suportadas com uma versão de glibc 2.2 ou superior, utilizando o novo Kernel. As versões 2.1.3 do glibc contém um suporte limitado (e quebrado) para “large files”. O Red Hat Linux 7.1 é distribuído com glibc versão 2.2.2-10 e foi testado para funcionar com ficheiros maiores que 2Gb (utilizando o ext2 native Linux filesystem, que é o filesystem por defeito em todas as distribuições Linux actuais). Genericamente, qualquer distribuição recente do Linux que usa o kernel 2.4, ext2 filesystem, e o glibc 2.2 ou superior, irá suportar ficheiros maiores que o anterior limite máximo de 32-bit/2Gb.

Devido ao suporte de ficheiros 64-bit não serem inequívocos nas distribuições Linux e nas plataformas Unix comerciais, as versões Firebird para *nix são disponibilizadas em dois tipos: 32-bit e 64-bit I/O. Escolha a versão que saiba que a sua plataforma possa suportar.

A exigência básica é o glibc 2.2 (+ suporte em utilitários) e o kernel 2.4 kernel (para VFS). Estas condições estão presentes no RH 7.1, MDK 8.0 e SuSE 7.2 assim como novas versões. Mas tome cuidado, porque algumas distribuições dão a opção de instalar componentes antigos! Por exemplo, o Mandrake 8.x permite a instalação do Kernel 2.2 na instalação avançada.

MacOS X 10.0

O Firebird não suporta MacOS X 10.0 devido em parte ao bug existente nesta versão referente ao suporte para 64 bit I/O.

“Filesystems”

Embora o “engine” possa agora suportar operações de 64-bit, é importante lembrar que nem todos os sistemas de ficheiros suportam “large files”.

No caso do Win32, o tamanho máximo para um ficheiro é de: (não confundir com o tamanho máximo de um volume/partição):

FAT16 para Win9x/ME	tamanho máximo de um ficheiro = 2GB - 1byte
FAT16 para WinNT/2000	tamanho máximo de um ficheiro = 4GB - 1byte
FAT32 para Win9x/ME/2000	tamanho máximo de um ficheiro = 4GB - 1byte
NTFS para WinNT/2000	tamanho máximo de um ficheiro = 16,384GB - 1byte

Em Linux e Unix, existem diversos filesystems, pelo que aconselhamos a consulta deste URL para obter mais detalhes: http://www.suse.de/~aj/linux_ufs.html.

As boas notícias são que todas as distribuições Linux mais conhecidas utilizam o ext2 filesystem, que suporta “large files”.

Importa em:

1. Utilizar o kernel 2.4 ou seguintes
2. Utilizar glibc 2.2 ou seguintes
3. Utilizar ext2 filesystem, ou outro “filesystem” que suporta o interface Kernel LFS (“large file system”).

(David Trudgett)

4. Deverá funcionar também em ReiserFS e Ext3 (mas não foi verificado). (Pavel Cisar)

Adicionado o Suporte para “pages” de 16kb

Tanto as rotinas do “engine” como as do GBAK foram alteradas de forma a permitir um PAGE_SIZE maior que o prévio limite de 8Kb. Aumentando o limite de tamanho de uma página aumenta-se as opções para melhoria de performance em determinado hardware.

Win32 Forced Writes Default = ON

A partir da versão 6.0 do IB 6.0, o parâmetro FORCED WRITES para bases de dados novas, sobre Win32, foi alterado para por defeito ser OFF, para melhorar a performance da base de dados à custa da integridade/recuperação da base de dados. Este parâmetro foi de novo revertido para o valor “ON”.

Um problema que reportava o erro “bad connection string”, que impedia de alterar o “forced writes” em modo multi-utilizador foi também resolvido.

Aumentado o limite do número de Tabelas Externas abertas em simultâneo (NT)

O limite formal do número de tabelas externas que podem ser utilizadas em simultâneo no NT é o número de "fopens" por defeito permitidos menos alguns ficheiros, ou seja, cerca de 510.

Esta alteração faz o Firebird alterar o número de "fopens" do NT para o máximo permitido pelo Sistema Operativo, 2048. O limite actual do número de Tabelas Externas que podem ser abertas em simultâneo nunca foi testado, mas acredita-se que esteja em cerca de 2040, dependendo do número de Bases de Dados abertas, do número de ficheiros por Base de Dados, etc.

Os ficheiros externos são agora fechados quando a tabela externa é "dropped"

Uma pequena alteração ao módulo **Deferred Work Handler (dfw.e)** permite que o ficheiro externo seja encerrado (closed) quando a tabela externa que está anexada a este é "dropped". Isto liberta-nos do estranho problema que se via ao elaborar "drop" da tabela e não se conseguir apagar o ficheiro externo sem primeiro encerrar o servidor.

Especificar a Porta na "String" de conexão

Funcionalidade id 1468, SF ID 447400

A possibilidade de indicar a porta de conexão é uma nova funcionalidade do Firebird. Para conectar a uma porta que não a standard 3050, deverá indicar o número da porta na "string" de conexão. O sintaxe é o seguinte:

Win32:

```
Servidor/Porta:Disco:\Directoria\Database.gdb
```

Linux e outros servidores Unix:

```
Servidor/Porta:/device/Directoria/Database.gdb
```

Repare na utilização do "/" e não ":" entre o nome do servidor e da porta. Os ":" são necessários antes da "string" que define a "path" para a base de dados.

A Conexão é feita por defeito na Porta 3050

O Cliente e o Servidor utilizam a porta TCP 3050 se nenhuma entrada estiver explícita no ficheiro "Services". Isto pretende resolver um problema comum de instalação.

GBAK tem uma nova opção - COUNTER - quando em modo -V(erbose)

O modo -V (Verbose) do GBAK permite agora que um contador seja especificado, i.e.

```
GBAK ... -V 20000
```

Este valor irá dar mais informação para o utilizador, à medida que as linhas forem sendo guardadas ou restauradas. Sempre que o GBAK processe o número de linhas indicado (tanto guardadas, como restauradas e/ou indexadas), o GBAK irá imprimir uma mensagem indicando o número do contador de linhas.

Por defeito, o valor do contador é de 10,000.

“JOIN’s” ambíguos são rejeitados

O InterBase não o impede de submeter uma declaração como esta:

```
SELECT A.FIELD A, B.FIELD B  
FROM A JOIN B  
ON FIELDX = FIELDY  
WHERE FIELD A="99"  
ORDER BY FIELD A
```

Este tipo de declarações podem retornar resultados imprevisíveis. O Firebird, em dialecto 3, retorna um erro se existir algum identificador de coluna não perfeitamente identificado num JOIN. No dialecto 1, o Firebird retorna um “warning” mas a query será à mesma processada.

Extensões à Linguagem

CURRENT_USER e CURRENT_ROLE

Estas duas variáveis de contexto foram introduzidas para referenciar o utilizador ("USER") e, se implementado, o "ROLE" da conexão em curso.

```
CREATE GENERATOR GEN_USER_LOG;
CREATE DOMAIN INT_64 AS NUMERIC(18,0);
COMMIT;
CREATE TABLE USER_LOG(
  LOG_ID INT_64 PRIMARY KEY NOT NULL,
  OP_TIMESTAMP TIMESTAMP,
  LOG_TABLE VARCHAR(31),
  LOG_TABLE_ID INT_64,
  LOG_OP CHAR(1),
  LOG_USER VARCHAR(8),
  LOG_ROLE VARCHAR(31));

COMMIT;

CREATE TRIGGER ATABLE_AI FOR ATABLE
ACTIVE AFTER INSERT POSITION 0 AS
BEGIN
  INSERT INTO USER_LOG VALUES(
    GEN_ID(GEN_USER_LOG, 1),
    CURRENT_TIMESTAMP,
    'ATABLE',
    NEW.ID,
    'I',
    CURRENT_USER,
    CURRENT_ROLE);
END
```

CURRENT_USER é um sinónimo DSQL para USER que surge no SQL standard. São idênticos. Não existe nenhuma vantagem de utilizar CURRENT_USER em vez de USER.

¹ Se insistir em utilizar bases de dados InterBase v.4.x ou 5.1 com o Firebird, ROLE não é suportado, pelo que current_role será sempre NONE (como indicado pelo SQL standard na ausência de uma ROLE explícita) mesmo que o utilizador passe um nome para a role. Se utilizar IB 5.5, IB 6 ou o Firebird, então a ROLE que se passar é verificada. Se a ROLE não existir, é então definida como NONE sem retornar qualquer erro.

Isto indica que no FB nunca terá uma mensagem de "invalid ROLE" retornada por CURRENT_ROLE, porque esta será reinicializada para NONE. Esta forma contrasta com o IB, onde o valor errado é tratado internamente, embora não seja visível pelo SQL.

DROP GENERATOR

Permite que um Generator seja removido da base de dados. O espaço que ocupa será libertado após o próximo "restore". Disponível em SQL e DSQL.

```
DROP GENERATOR <generator name>;
```

GROUP BY UDF

É agora possível agregar um SELECT por agrupamento do output por uma UDF, i.e.:

```
select strlen(rtrim(rdb$relation_name)), count(*) from rdb$relations
group by strlen(rtrim(rdb$relation_name))
order by 2
```

Um efeito colateral do agrupamento por UDFs é que, onde antigamente não era permitido a utilização de funções internas do Firebird, agora tal é permitido pela utilização de uma “dummy” UDF:

```
select count(*)
from rdb$relations r
group by bin_or((select count(rdb$field_name) from rdb$relation_fields f
where f.rdb$relation_name = r.rdb$relation_name),1)
```

RECREATE PROCEDURE

Este novo comando DDL permite criar uma “Stored Procedure” com o mesmo nome de uma já existente, substituindo a anterior, sem necessitar de fazer “drop” primeiro à antiga. O sintaxe é idêntico a CREATE PROCEDURE.

Disponível em SQL e DSQL.

RECREATE TABLE

Este novo comando DDL permite criar uma nova estrutura para uma tabela existente, sem necessitar de fazer “drop” primeiro à tabela antiga. O sintaxe é idêntico a CREATE TABLE.

Note que RECREATE TABLE **não preserva** os dados da tabela antiga.

Disponível em SQL e DSQL.

SELECT [FIRST (<integer expr m>)] [SKIP (<integer expr n>)]

Retorna as primeiras m linhas do output seleccionado. O parâmetro opcional SKIP causa as primeiras n linhas serem ignoradas e retorna assim um output de m linhas começando em $n + 1$. Na sua forma mais simples, m e n são inteiros, mas qualquer expressão Firebird que possa ser avaliada como um inteiro é válida. Um identificador que possa ser avaliado como um inteiro pode também ser usado em GDML, mas não em SQL ou DSQL.

Os parênteses são obrigatórios se se utilizar expressões como argumento, caso contrário são opcionais.

Podem ainda ser utilizadas variáveis, isto é, uma expressão do tipo SKIP ? * FROM ATABLE retorna o dataset restante depois de ignorar as primeiras n linhas iniciais, em que n é passado através da variável “?”. SELECT FIRST ? COLUMNA, COLUMNB FROM ATABLE retorna as primeiras m linhas e ignora todas as restantes.

A clausula FIRST também é opcional, isto é, pode-se incluir SKIP numa declaração sem utilizar FIRST, para obter um “output set” que simplesmente ignora as linhas apontadas por SKIP.

Disponível em SQL e DSQL excepto quando indicação em contrário.

Exemplos:

```
SELECT SKIP (5+3*5) * FROM MYTABLE;
```

```
SELECT FIRST (4-2) SKIP ? * FROM MYTABLE;
```

```
SELECT FIRST 5 DISTINCT FIELD FROM MYTABLE;
```

Um "Gotcha" com SELECT FIRST

Esta declaração

```
delete from TAB1 where PK1 in (select first 10 PK1 from TAB1);
```

irá eliminar todas as linhas na tabela. Ouch! O "sub-select" avalia as primeiras 10 linhas candidatas para apagar, apaga-as, avalia as próximas 10, apaga-as, e assim sucessivamente até que não existam mais linhas disponíveis na tabela.

Cuidado!

SUBSTRING(<string expr> FROM <pos> [FOR <length>])

Função interna que implementa a função ANSI SQL SUBSTRING(). Irá retornar uma "string" que consiste no byte na posição <pos> e todos os bytes subsequentes até ao final da "string". Se a opção FOR <length> estiver especificada, irá retornar o menor destes dois valores: o tamanho <length> especificado, ou o número de bytes até ao final da "string".

O primeiro argumento poderá ser uma expressão qualquer, constante ou identificador que possa ser avaliado para uma "string".

<pos> necessita de ser avaliado para um inteiro.

<pos> começa em 1, como outros comandos SQL.

Nem <pos> nem <length> podem ser parâmetros de Querys.

Dado que <pos> e <length> são posições de byte, o identificador pode ser um blob binário, ou um blob "sub_type 1 text blob" com um charset de um-byte-por-caracter. A função não admite de momento blobs de texto com character sets Chinese (2 byte/char máximo) ou Unicode (3 byte/char máximo). Para um argumento do tipo "string" (ao contrário de um blob), a função trabalha com qualquer charset.

Disponível em SQL e DSQL.

```
UPDATE ATABLE
SET COLUMNB = SUBSTRING(COLUMNB FROM 4 FOR 99)
WHERE ...
```

Refira ainda para a secção de "Funções Externas" (UDFs) seguinte, para detalhes de alterações e adições à livreria standard de UDF.

Adicionado o Collation Set Húngaro "Case Insensitive"

Adicionado o Collation Set Húngaro, "case insensitive", desenvolvido e testado por Sandor Szollosi (ssani@freemail.hu).

Adicionado o Charset ISO8859-2

O Firebird suporta o character set ISO-8859-2 (para suporte da Língua Checa).

Novo Marcador de Comentário

Para ser utilizado em "scripts", DSQL, "stored procedures" e "triggers".

```
-- Este é um comentário
```

Este novo marcador pode ser utilizado para comentar uma simples linha de código num "script", numa declaração DDL/DML, "stored procedure" ou "trigger".

A lógica é ignorar todos os caracteres que o sigam.

1. Saltar os caracteres "--" se este símbolo for o primeiro par de caracteres depois de uma marca fim-de-linha "EOL" (LF em Linux/Unix, CRLF em Windows)
2. Continuar a ignorar os caracteres até à próxima marca fim-de-linha (EOL)

Esta lógica não é feita para se misturar com um bloco de comentário lógico (/* um comentário */). Em outras palavras, não utilize os marcadores "--" dentro de um bloco de comentário, e não utilize um bloco de comentário numa linha que utiliza os marcadores "--".

SESSÕES ISQL INTERACTIVAS: Tenha atenção quando trabalhar numa sessão ISQL interactiva. O ISQL aceita partes de uma declaração em segmentos contínuos, mostrando a "prompt" CON> até que receba o símbolo de terminação (normalmente ;). Se utilizar "--" no início de uma linha de continuação, a lógica de ignorar irá continuar até encontrar a marcação de fim-de-linha (EOL) que está impressa no visor ou no seu ficheiro de "output" quando premir ENTER. Existe uma fonte potencial de erros se continuar a adicionar uma continuação, esperando que esta seja ignorada.

O problema com o isql surge devido aos seus comandos especiais que deverão ser objecto de um "parse" apenas pelo ISQL. Se não são reconhecidos devido a uma colocação inesperada de "--", então são enviados ao "engine". Obviamente, o "engine" não compreende os comandos de isql SET e SHOW e rejeita-os.

Alter Trigger deixou de incrementar o "Contador de Alterações" na Tabela

Quando o contador das alterações da "metadata" alcança o valor máximo de 255, a base de dados fica indisponível. Uma operação de "Backup/Restore" é necessária para elaborar um "reset" do contador e tornar a base de dados disponível de novo. A intenção é forçar a uma limpeza da base de dados (cleanup) quando a estrutura das tabelas já sofreram bastantes alterações, e não para inibir as capacidades do "engine".

Antigamente, cada vez que um "trigger" fosse colocado ACTIVE|INACTIVE por uma operação de ALTER TRIGGER, o Contador de Alterações da tabela associada era incrementado. Em consequência, o uso destas funções em operações regulares não era possível, pois o número atingia o seu máximo rapidamente.

Novas Palavras Reservadas (“Reserved Words”)

Os seguintes “**Firebird keywords**” devem ser adicionados à lista de “**reserved words**” publicada para o InterBase 6.0.1. Os que estão marcados com um asterisco (*) estão reservados para o futuro.

ABS *	BOTH *	BREAK
CASE *	CHAR_LENGTH *	CHARACTER_LENGTH *
COALESCE *	CURRENT_ROLE	CURRENT_USER
DESCRIPTOR	FIRST	IIF *
LEADING *	NULLIF *	OCTET_LENGTH *
RECREATE	SKIP	SUBSTRING
TRIM *	TRAILING *	

Os seguintes “**InterBase 6.5 keywords**” (não reservados no Firebird) devem também ser tratados como reservados, para compatibilidade:

PERCENT	ROWS	TIES
---------	------	------

Funcionalidades ISQL

Nova opção PLANONLY para declarações

Foi adicionado o suporte para declarações PLANONLY, permitindo que uma declaração ou "query" seja submetida ao "engine" e o seu plano ser apurado, sem execução dessa mesma declaração/query. Este é um dos comandos "SET" que apenas o isql entende.

```
SET PLANONLY; /* altera o estado */  
SET PLANONLY ON|OFF;
```

A Query é submetida ao engine e é preparada, mas não executada. É equivalente ao comando "Prepare" do IB_SQL e IBConsole que mostra o Plano escolhido pelo engine.

Uma aplicação do PLANONLY será, por exemplo, o teste de declarações com parâmetros sem existir necessidade de indicar os valores desses mesmos parâmetros:

```
SQL> SET PLANONLY ON;  
SQL> select first ? fld from tbl;
```

NOTA : Inicialmente, PLAN e PLANONLY estão no modo OFF. (O problema com o comando PLAN é que é necessário executar a declaração para conhecer o plano do "engine")

- ❑ SET PLANONLY (ou SET PLANONLY ON) irá activar tanto o PLANONLY como o PLAN
- ❑ SET PLANONLY (ou SET PLANONLY OFF) irá desactivar o PLANONLY mas não o PLAN. (PLAN mantém-se ON).

SET => que retorna uma lista de modos activos, inclui agora o modo PLANONLY.

Comando interactivo de linha ADD activo em isql (com limitações)

Utilizando o novo comando isql ADD é agora possível inserir interactivamente linhas para uma tabela, incluindo blobs carregados de um ficheiro:

```
SQL> ADD aTable ;
```

Este comando adiciona um novo registo a aTable e inicia o procedimento interactivo de pedir valores ao utilizador, campo a campo. Termina quando se responde a um pedido de valor premindo apenas a tecla "Enter".

Quando o campo é um blob, permite o "edit" (edição) ou o "pump" (carregamento) de um ficheiro:

- ❑ EDIT faz o comando abrir um ficheiro temporário e chama o editor que tenha definido para BLOBVIEW. Após gravar o ficheiro que criou com o editor, automaticamente os resultados são carregados no registo que está a ser inserido.
- ❑ Se escrever uma "string" que não seja o comando EDIT, o comando add_row() assume-o como o nome de um ficheiro e simplesmente abre esse ficheiro e carrega-o no campo blob do registo que está a ser inserido.

NOTA: a função addrow() está limitada em algumas situações:

- ❑ Depois de carregar no ENTER após uma entrada, é adicionado um “line-feed” à entrada. Assim, quando premir ENTER no ultimo campo do registo, deverá completar a entrada com um “Ctrl-Z” em adição ao ENTER final.
- ❑ Existem ainda algumas deficiências com a introdução de datas [Bug do Firebird nº 518343] não corrigindo o ano, o que permite a introdução de datas inválidas que são aceites pelo próprio “engine”. O comando addrow() utiliza a estrutura tm para campos data mas quem o implementou esqueceu-se que o ano é estruturado com base em 1900-, causando o ano guardado a retroceder 900 anos. Por exemplo,

```
SQL> create table t(d timestamp);
SQL> add t;
Enter data or NULL for each column. RETURN to end.
Enter D as M/D/Y>1/1/2002
Enter D as M/D/Y>
Bad date
SQL> select * from t;
D
=====
3902-01-01 00:00:00.0000
```

Utilização de aspas duplas em isql com as opções -x e -a foi “permitida”

Ao extrair nomes de objectos de um “script” com a opção -x ou -a, o isql seguia sempre a via mais simples e colocava aspas duplas entre identificadores. Este facto era desagradável para muitos utilizadores. Agora, o isql apenas coloca aspas duplas nos identificadores apenas se o nome não poder ser expresso sem as mesmas.

Um identificador não necessita de aspas duplas se contém:

- Apenas ASCII de A-Z (maiúsculas apenas)
- dígitos de 0..9
- sublinhado (“underscore”) (_) e \$ desde que seja na primeira posição, e que apenas a) seja encontrado.

O “engine” não distingue entre identificadores BLANK e identificadores “zero-length”. A razão é que as tabelas de sistema utilizam CHAR e não VARCHAR pelo que alguns espaços no final não são significativos. O DSQL não consegue distinguir entre um campo definido como BLANK e um identificador de output que é BLANK porque este provém de uma expressão que é semelhante a uma operação matemática. Para prevenir a confusão, os identificadores BLANK e “zero-length” estão proibidos. O Firebird pode apenas gerar esses nomes identificadores BLANK em expressões inline numa declaração SQL, porque não cria um identificador artificial para as colunas “computed on the fly”.

Normalmente, deverá utilizar a palavra chave “AS” e indicar implicitamente o nome do campo de output. Se se omitir, está-se a criar efectivamente um campo blank (ou zero) e deve ser representado por um par de aspas duplas.

Melhoramentos à API

Adições e Alterações a *isc_info_database*

Foram adicionados três itens:

isc_info_db_provider, cujos valores são:

```
enum info_db_provider
{
isc_info_db_code_rdb_eln,
isc_info_db_code_rdb_vms,
isc_info_db_code_interbase,
isc_info_db_code_firebird
};
```

isc_info_db_class, cujos valores são:

```
enum info_db_class
{
isc_info_db_class_access = 1,
....
isc_info_db_class_cache,
isc_info_db_class_classic_access,
isc_info_db_class_server_access
};
```

isc_info_firebird_version, os valores estão definidos em *inf.c*

Um item foi renomeado:

isc_info_isc_version

que era antigamente *isc_info_version*.

O nome antigo é "#defined" para o novo, de forma que tudo funcione como até então. No futuro poder-se-à alterar *isc_info_version* para apontar para *isc_info_firebird_version*.

Inf.c tem o código para manipular estes novos itens.

GDS_VERSION foi redefinida em *license.h* para reflectir o facto de o Firebird ser em simultâneo 6.2 e 1.0. Também foi definida *FB_VERSION* que pode ser eventualmente utilizada.

A lista de implementações em *util.c* foi cortada de *InterBase/xxx* para *Firebird/xxx*. Os números de implementação em *common.h* e *ibase.h* foram alterados para coincidirem com a lista de implementações em *util.c*

Melhorias ao item *isc_database_info*

Fundado by Jason Wharton, CPS (IB Objects)

Na API, 4 "request buffers" foram adicionados à estrutura *isc_database_info*, para chamadas que pretendam informações de estatísticas sem necessidade de inicializar e chamar as Services API.

Item

65 isc_info_oldest_transaction
66 isc_info_oldest_active
67 isc_info_oldest_snapshot
68 isc_info_next_transaction

Todos os itens retornam inteiros. Consulte o "API Guide" para obter exemplos de chamadas a `isc_database_info()`. O exemplo expresso no Guide é muito semelhante, pois retorna valores inteiros, e estes novos itens são inteiros também.

Substitua os itens declarados no exemplo com:

```
char db_items[] = {  
isc_info_oldest_transaction,  
isc_info_oldest_active,  
isc_info_oldest_snapshot,  
isc_info_next_transaction,  
isc_info_end};
```

O exemplo tem um loop com um "switch()". Dentro do "loop", altere o "case" para estes novos valores. Deverá existir 4 "cases" mais o "default" em vez de apenas 2 cases + default do exemplo.

Funções Externas (UDFs)

Na livreria "standard" ib_udf.dll, ib_udf.so

SUBSTR(<string expr>, <pos1>, <pos2>)

Retorna uma "string" que consiste nos caracteres de <pos1> a <pos2> inclusive. Se a <pos2> passar do final da "string", a função retorna todos os caracteres até ao final da "string". NOTE que esta forma é diferente da forma que SUBSTR trabalha em IB e em versões prévias do Firebird, que retornava NULL quando <pos2> passava do final da "string".

```
UPDATE ATABLE
SET COLUMNB = SUBSTR(COLUMNB, 4, 32765)
WHERE...
```

SUBSTRLEN(<string expr>, <pos>, <length>)

Retorna uma "string" do tamanho <length> começando na posição <pos>. O tamanho da "string" será o menor destes dois valores: o valor <length> ou o número de caracteres desde <pos> até ao final da string.

```
UPDATE ATABLE
SET COLUMNB = SUBSTRLEN(COLUMNB, 4, 99)
WHERE...
```

ascii_char()

Claudio Valderrama reparou um antigo "bug" na declaração de ascii_char em ib_udf.sql. Foi corrigida de forma que retorna um único carácter C string em vez de um tipo InterBase CHAR(1), o que era incorrecto:

```
DECLARE EXTERNAL FUNCTION ascii_char
  INTEGER
  RETURNS CSTRING(1) FREE_IT
  ENTRY_POINT 'IB_UDF_ascii_char' MODULE_NAME 'ib_udf';
```

Na nova livraria Firebird UDF, FBUDF.dll

Desenvolvida por Claudio Valderrama C., fundada por Craig L. Leonard

Distribuída com esta versão encontra a FBUDF.dll, a primeira livraria Firebird UDF que utiliza o sintaxe BY DESCRIPTOR para passar argumentos, o que permite uma maior versatilidade. Na declaração de funções em SQL, note as diferenças entre as múltiplas declarações que referenciam a mesma função. Por exemplo, repare que INULLIF() referencia tanto INULLIF() como I64NULLIF().

O autor, Claudio Valderrama, comenta que a livraria ainda está numa fase beta e adverte que podem existir alguns bugs. Até agora não foi compilada em mais nenhuma plataforma que não Windows. Relatórios de bugs e comentários são bemvidos.

O código fonte assim como o DDL para as declarações estão na árvore CVS do Firebird. Para o encontrar, vá a <http://sourceforge.net/projects/firebird>, clique em "Browse CVS Repository" e depois seleccione Developers | Latest sources | interbase | extlib | fbudf.*.

NVL() - funções para precisão exacta ("invl") e parâmetros de string ("snvl")

Estas funções tendem a copiar a função NVL do Oracle, para extrair um valor actual quando a coluna tem um valor NULL. Pegam em dois argumentos, o primeiro sendo a expressão que será testada para NULL, o segundo o valor de retorno se o primeiro for NULL. NVL retorna o primeiro argumento se este não for NULL e o Segundo argumento se o primeiro for NULL. Se ambos forem NULL, retorna NULL.

O para de parâmetros devem ser compatíveis, ou dois valores numéricos (smallint, int, int64) ou duas "strings" (char, varchar, cstring). O engine não verifica os tipos de parâmetros quando usa esta técnica no FBUDF, pelo que se misturar um parâmetro numérico e uma "string" como argumentos terá resultados errados.

NULLIF() para parâmetros string ("snullif"), inteiro ("inullif") e INT64 ("i64nullif")

NULLIF pega em dois argumentos, retornando NULL se forem equivalentes, ou o resultado da primeira expressão se não o forem. Devido a um percalço no "engine" que previne o retorno de valores NULL por uma UDF, cada uma destas três funções retorna um equivalente-zero. Este método não standard torna a função não utilizável para "casting" de alguns valores como NULL quando existem algumas funções agregadas que ignoram nulls.

NOTE que a função é chamada de igual forma quer para integer quer para int64 ("inullif").

Funções "Dia-Da-Semana" (Day-of-Week) - uma que retorna uma pequena "string" ("SDOW"), outra que retorna uma maior ("DOW"), para um input do tipo "timestamp". A "string" retornada pode ser localizada.

Várias funções que **segmentam "time" para um "timestamp"** - "addDay", "AddWeek", etc.

Função RIGHT() (como RString em BASIC) que retorna os "n" caracteres à direita de uma "string".

Função **GetExactTimestamp()** que retorna o "timestamp" com precisão de milisegundos.

Truncate() e **i64truncate()** truncam inteiros 32-bit e 64-bit respectivamente, aceitando como parâmetros um valor numérico de precisão exacta de qualquer escala (até 9 em dialecto 1 ou até 19 em dialecto 3) e retorna a parte inteira. Não funciona com tipos float ou double.

```
truncate(14.76) retorna 14
```

round(14.22) retorna 14

Round() e **i64Round()** aceitam parâmetros inteiros de 32-bit e 64-bit respectivamente, numéricos de precisão exacta de qualquer escala (até 9 em Dialecto 1 e até 19 em Dialecto 3) e retorna o valor inteiro mais próximo. Não se pode especificar o número de casas decimais.

round(14.76) retorna 15
round(14.22) retorna 14

Estas funções estão compiladas para devolver arredondamentos matemáticos, i.e., sempre para cima desde que o valor seja começado com o número 5. Existe aqui uma diferença em relação a algumas funções da Borland que arredondam para o número ABSOLUTE mais alto (são simétricas).

round(1.5) => 2
round(-1.5) => -1

String2blob() converte um tipo char ou varchar num tipo blob. É como a função existente na freeUdfLib, mas é muito mais simples internamente.

Novos Parâmetros de Configuração

Alterações que afectam os ficheiros de configuração ibconfig/iscconfig

Adicionado o "switch" -NONAGLE para as versões Linux

Desactivar o "TCP/IP Nagle Algorithm" melhora a velocidade em redes lentas.

O "TCP/IP Nagle Algorithm" foi concebido para resolver problemas com pacotes pequenos, chamados "tinygrams", em redes lentas. O algoritmo diz que se uma conexão TCP/IP só pode ter apenas um pequeno segmento em espera então ainda não foi considerada em "acknowledged". A definição de "pequena" varia, mas normalmente é definida como menor que o tamanho do segmento em que a Ethernet está, cerca de 1500 bytes.

Por defeito, a livreria de "socket" irá utilizar um algoritmo interno, conhecido como "Nagle's Algorithm" para elaborar um "buffer" de bytes em escrita antes de enviar esse mesmos dados, de forma a minimizar o número de escritas físicas.

A presença de um novo "switch" em Linux permite aos programadores determinarem, por eles mesmos, os prós e os contras de utilizar esta alternativa de "packet handling".

Por defeito o isc_config/ibconfig utiliza o algoritmo Nagle:

```
#TCP_NO_NAGLE          0
```

Para o desabilitar deverá fazer "uncomment" desta linha e alterá-la para:

```
TCP_NO_NAGLE          1
```

Parâmetro de Configuração CPU_AFFINITY

Com o Firebird SuperServer em Windows, existe um problema com o Windows em constante mudança de processos entre os vários processadores em máquinas SMP. Isto arruina a performance. Até agora, para colocar o ibserver num único CPU, era necessário colocar o servidor como uma aplicação e correr um utilitário (IB_Affinity.exe) sobre o servidor.

Este novo parâmetro de configuração pode ser adicionado ao ibconfig para remover a necessidade de correr um programa externo para endereçar o Firebird a um determinado CPU num sistema Windows SMP. Este parâmetro tem efeito quando o sistema arranca.

O parâmetro CPU_AFFINITY consiste num numero inteiro que a CPU "mask".

Exemplo:

```
CPU_AFFINITY 1
Corre apenas no primeiro CPU (CPU 0).
CPU_AFFINITY 2
Corre apenas no segundo CPU (CPU 1).
CPU_AFFINITY 3
Corre tanto no primeiro como no terceiro CPU.
```

NOTA: este parâmetro não tem efeito em W9x, dado que usa chamadas API do NT API. Os sistemas W9x não tiram proveito de máquinas com múltiplos CPUS.

Definir Expressamente a Localização de Ficheiros Externos

O novo parâmetro de configuração para ficheiros externos é específico para a plataforma. Está implementado de momento apenas em Windows. Funciona desta forma:

No ficheiro *ibconfig* construa uma entrada para cada directoria (existente) onde pretende que o Firebird procure os ficheiros externos. Repare que tem de colocar o nome das directorias entre aspas duplas:

```
EXTERNAL_FILE_DIRECTORY "H:\teste"  
EXTERNAL_FILE_DIRECTORY "H:\externas"
```

Não existe limite no numero de directorias que podem ser utilizadas. O Firebird irá utilizar estas séries de directorias numa "search list". Assim:

```
isql> create table foo1 external file 'foo1.txt' (afield(char1), crlf char(2));
```

Utilizando os nossos exemplos, o Firebird irá procurar tanto em h:\teste como em h:\externas para verificar se o ficheiro foo1.txt existe. Se existe, a tabela externa irá ser criada e irá apontar para o ficheiro na directoria correcta (embora o campo rdb\$external_file na tabela rdb\$relations guarde apenas foo1.txt, sem qualquer detalhe de nome de directoria).

Quando a tabela foo1 é acedida, o ficheiro é recolhido automaticamente da devida directoria.

Se o ficheiro não existe, será criado por defeito na directoria corrente (por defeito): onde se encontrar na altura em que correr o isql ou o utilitário que estiver a usar é irrelevante. Neste caso, o nome da directoria é extraído da definição de ficheiro externo em rdb\$relations porque já está definido em EXTERNAL_FILE_DIRECTORY. Parte-se do principio que, se não está a especificar o caminho, está-se a referir a um ficheiro externo numa directoria externa pré-definida. Não poderá assim, correr o isql a partir de uma determinada directoria onde queira que a base de dados o encontre!

Se não existir o parâmetro EXTERNAL_FILE_DIRECTORY no ficheiro de configuração, o ficheiro externo obtém a mesma path completa que ISC_expand_filename. Se fizer

```
isql> create table foo1 external file 'h:\files\foo1.txt' (  
    afield(char1),  
    crlf char(2));
```

i.e. especificando o nome da directoria do ficheiro externo. Se o ficheiro existir é definido como normal, se não existir, é criado como normal na directoria especificada.

Notas de Instalação

Instalação em Windows 32

Conteúdo

- Antes da Instalação
- Instalação num Sistema com InterBase®
- Pressupostos da Instalação
- Desinstalação
- Outras Notas

Antes da Instalação

IMPORTANTE!

O programa de instalação irá tentar detectar se existe uma versão instalada e/ou a correr do Firebird ou Interbase. Deverá PARAR o servidor actual e/ou remover a versão instalada antes de prosseguir.

Parando o Servidor

- Se está a correr como um Serviço, pare via “Painel de Controlo | Serviços”.
- Se é uma aplicação, fecha-a simplesmente.

Removendo um Servidor Existente

É recomendado que se desinstale uma versão prévia do Firebird ou InterBase, mas não é uma obrigação. Consulte a secção de “Desinstalação” mais abaixo para obter mais detalhes sobre a rotina de desinstalação do Firebird.

Instalação num Sistema com InterBase®

O Firebird 1.0 não pode ser executado em simultâneo com o InterBase® (qualquer versão). Deverá desinstalar o InterBase® antes de instalar o Firebird. Isto pode ser alterado com futuras versões do Firebird.

É recomendado que não instale o Firebird por cima de uma versão de Interbase®. Instale o Firebird noutra directoria.

Base de Dados de Segurança (utilizadores e passwords)

- Utilize o **gbak** para fazer uma cópia de segurança da sua antiga base de segurança **isc4.gdb**.
- Reponha a cópia com um novo nome.
- Pare o servidor Firebird.
- Renomeia-a para **isc4.gdb** e reinicie o servidor.

Se possui algumas configurações personalizadas no ficheiro **ibconfig** pode fazer “cortar e colar” deste para o novo **ibconfig**. Não esqueça que isto não fará efeito enquanto não reinicializar o servidor.

Pressupostos da Instalação

- São necessários Direitos de Administrador para instalar o Firebird como um serviço. Esta obrigação não é aplicada em Win95, Win98 ou Win ME.

- Se a instalação encontrar uma nova versão do ficheiro GDS32.DLL será questionado para a substituir. É recomendado responder "YES" se estiver a fazer uma instalação de servidor.
- Se for detectada uma qualquer versão do ficheiro MSVCRT.DLL esta não será actualizada. Este ficheiro será apenas instalado se não existir no sistema.
- Se alguns ficheiros de configuração forem detectados na directoria de instalação, estes serão conservados. Esses ficheiros são:
 - isc4.gdb
 - interbase.log
 - ibconfig

Desinstalação

A rotina de desinstalação do Firebird preserva os seguintes ficheiros:

```
isc4.gdb
interbase.log
ibconfig
```

Nenhuma tentativa é feita de remover ficheiros que não tenham feito parte da instalação original. Ficheiros partilhados como é o caso do gds32.dll serão removidos se o contador de partilha ("share count") indicar que nenhuma outra aplicação o está a utilizar. As chaves do Registro que foram criadas são removidas.

Outras Notas

Winsock2

O Firebird necessita do WinSock2. Todas as plataformas Win32 deverão já possuir o WinSock2, excepto o Win95. É feito um teste à presença das bibliotecas Winsock2 durante a instalação. Se não for detectado a instalação irá falhar. Para fazer upgrade ao seu sistema consulte este link:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;q177719>

Windows ME and XP

No Windows ME e XP (Edições Home e Professional) existe uma "Funcionalidade" chamada "System Restore", que faz uma cópia de segurança automática ("backup caching") de todos os ficheiros do sistema que possuem uma terminação ".gdb". O efeito retarda o acesso a bases de dados InterBase/Firebird porque é elaborado um "backup" destes ficheiros sempre que ocorre uma operação de I/O. (No XP não existe "System Restore" em Servidores .NET).

Um ficheiro da directoria Windows do ME, c:\windows\system\filelist.xml, contém os "protected file types", e os tipos ".gdb" são referidos aí. Charlie Caro de início sugeriu que se removesse a extensão GDB da secção "includes" deste ficheiro. Porém, desde que se execute esta operação, ficou demonstrado que o WinME pode restaurar esta lista. Em XP, não é possível editar o ficheiro filelist.xml.

Em ME, sugere-se uma alternativa que consiste em:

- Utilizar a extensão FDB (Firebird DB) para os ficheiros Base de Dados
- Mover a Base de Dados para C:\Meus Documentos, que é ignorada pelo "System Restore"
- Desactivar por completo a opção "System Restore" (consulte a documentação do Windows para instruções).

No Windows XP Edições Home e Professional poderá mover as suas bases de dados para uma partição em separado, e configurar o "System Restore" para excluir esse volume.

O Windows XP utiliza uma "smart copy", pelo que o "overhead" visto em Windows ME não deve ser tão notável em XP, pelo menos para ficheiros pequenos. Para ficheiros maiores (como qualquer Base de Dados Firebird tem tendência a ser!) não parece existir uma resolução se tiver os seus ficheiros ".gdb" localizados no filesystem geral.

Há a referir ainda a base de dados de segurança isc4.gdb, que é considerada com "escrita permitida" pelo código que simplesmente valida um "login" do utilizador, de forma que a "gheader" possa ser actualizada por essa transação. Assim, o WinME irá fazer muito provavelmente um "backup" de cada vez que um utilizador se valide no sistema.

Estamos a tentar obter uma descrição acurada do problema e uma solução para o mesmo será colocado aqui. Se pode ajudar com uma descrição do problema ou com uma solução alternativa, por favor envie uma mensagem para a lista ib-support ou para a lista firebird-devel em news://news.atkin.com

Instalação em UNIX / Linux

(Mark O'Donohue)

O servidor Firebird é fornecido em dois formatos, "Classic" que corre como um serviço, ou "SuperServer" que corre como um "background daemon". Embora o futuro tenha tendência a ser o SuperServer, aconselha-se o utilizador que está a iniciar com o Firebird a utilizar o "Classic", pois com certeza será uma melhor plataforma para experimentar o Firebird.

NOTAS:

- 1) Tem de possuir direitos "root" para instalar o Firebird.
- 2) Para uma instalação eficaz do SuperServer deve adicionar localhost ao seu ficheiro /etc/hosts.equiv.
- 3) Se necessitar de acesso remoto de outras máquinas à base de dados, deverá ainda adicionar o nome dessas máquinas ao ficheiro /etc/hosts.equiv.

A edição Super Server instala-se da mesma forma que a indicada em diante, com a particularidade dos ficheiros de instalação estarem marcados SS em vez de CS.

Para uma instalação rpm linux

```
$rpm -Uvh FirebirdCS-1.0.0-nnn.i386.rpm
```

Para uma instalação .tar.gz linux

```
$tar -xzf FirebirdCS-1.0.0-nnn.tar.gz
$cd install
$./install.sh
```

* OU FirebirdSS-1.0.0-nnn

O que faz a instalação Linux

A instalação Linux:

1. Tenta parar algum servidor que esteja em execução;
2. Se existir uma instalação prévia do Firebird, então esta e todos os ficheiros associados em /usr/lib /usr/include serão arquivados no ficheiro /opt/interbase_<datetimestamp>.tar.gz e de seguida serão eliminados.

3. Instala o software na directoria /opt/interbase e as livrarias em /usr/lib e as "headers" em /usr/include
4. Automaticamente adiciona gds_db para a porta 3050 em /etc/services
5. Automaticamente adiciona localhost.localdomain e HOSTNAME a /etc/host.equiv
6. A versão SuperServer ainda instala um script de arranque em /etc/rc.d/init.d/firebird. Um novo "link" rcfirebird é criado em /usr/bin para o script init.d. Este método é preferível a executar directamente o Firebird no script initd, pois /usr/bin está na "search path" de todos os utilizadores.
7. É criada a entrada Firebird em /etc/rc.config para SuSE (SuSE specific configuration for service startup management).

O Firebird deve arrancar automaticamente em runlevel 2, 3 e 5.

Todos os packages de instalação foram desenvolvidos e testados em RH 7.x, MDK 7.x e SuSE 7.x e versões posteriores.

Os packages .tar.gz não suportam a desinstalação.

A instalação "Classic" automaticamente configura a entrada xinetd se a directoria /etc/xinetd.d for encontrada, senão cria uma entrada inetd. Como algumas distribuições colocam o xinetd numa localização diferente da directoria /etc/xinetd.d, uma configuração manual é requerida nestas condições.

Testes da instalação Linux

Para testar o acesso local à sua instalação:

```
$cd /opt/interbase/bin
$isql -user sysdba -password <password*>

>connect /opt/interbase/examples/employee.gdb;

>select * from sales;
>select rdb$relation_name from rdb$relations;
>help;

>quit;
```

Para testar o acesso remoto:

```
$cd /opt/interbase/bin
$isql -user sysdba -password <password*>

>connect '<hostname>:/opt/interbase/examples/employee.gdb';

>select * from sales;
>select rdb$relation_name from rdb$relations;
>help;

>quit;
```

*Se tiver gerado uma password na sua instalação, obtenha-a no ficheiro /opt/interbase/SYSDBA.file.

Considerações em Ambientes Linux

Adicionalmente aos ficheiros standard de instalação, são fornecidos na directoria bin do Firebird os seguintes três scripts:-

(Substitua XX nos dois scripts por CS para o Firebird Classic ou por SS para o Firebird SuperServer.)

XXchangeRunUser.sh - Cria uma nova conta de utilizador unix firebird e altera o "owner" da Instalação Firebird e das tarefas em "background" para correr como este utilizador firebird.

XXrestoreRootRunUser.sh - Repõe o "owner" dos ficheiros Firebird files, e do "owner" utilizador das tarefas em "background" para o inicial da instalação por defeito "root"

É VIVAMENTE RECOMENDADO para uma instalação segura do Firebird que os processos do servidor não sejam corridos como root.

Se o fizer terá algumas restrições sobre quem poderá inicialmente criar bases de dados Firebird e onde estas poderão ser criadas.

changeDBAPassword.sh - Altera a password do utilizador Firebird SYSDBA e, se necessário, altera o "init script" /etc/rc.d/init.d/firebird com a nova password

//Fim das notas em Linux/Unix

Instalação do Firebird Classic & SuperServer em Solaris 2.7 Sparc

(Neil McCalden)

A versão lançada utiliza acessos de 32-bit como as prévias, isto é, aplica-se o limite de tamanho de ficheiros de 2Gb. Porém, existe uma versão de 64-bit Superserver na página de Downloads. Não está ainda na área de downloads devido a não terem sido elaborados ainda os testes suficientes.

Passos básicos para instalação (Classic e Super Server):

Como root, descomprima o ficheiro .tar na directoria de sua escolha.

Crie os "links" :-

```
ln -s /ExtractDirPath/interbase /usr/interbase
ln -s /usr/interbase /opt/interbase
```

```
cd /usr/interbase
./install
```

Esta operação ira criar "links" para ficheiros "header", livrarias e actualizar os ficheiros /etc/services e /etc/inetd.conf.

Passos extras para Super Server

Adicione localhost a /etc/hosts.equiv

Crie um utilizador e grupo interbas ou firebird

Crie um "script" em /etc/init.d|rc3.d para iniciar o servidor

- para um exemplo consulte /usr/interbase/bin/firebird

Se possui uma versão anterior do InterBase(tm) instalada, provavelmente em /opt/interbase, deverá remover o package ou renomear a directoria como apropriado.

Note que os ficheiros de exemplo não estão disponíveis nesta versão. Estão porém disponíveis para download em firebird.sourceforge.net.

Veja o ficheiro README ou consulte www.firebirdsql.org para mais informação sobre o Firebird.

//Fim das notas relativas a Solaris 2.7 Sparc

Instalação do Firebird Classic em MacOS X / Darwin

(John Bellardo)

Última Actualização 20 February 2002

O Firebird requiere o MacOS X 10.1 (Darwin kernel 5.2) ou superior. Faça "Download" da versão do Firebird para Darwin.

Esta é distribuída como um arquivo.tar.gz. Muitos browsers irão chamar automaticamente o Stuffit Expander para extrair o ficheiro automaticamente. Certifique-se que a versão nativa do Stuffit Expander para o OS X é utilizada. Se utilizar a versão Classic, terá alguns problemas de instalação.

Uma alternativa à utilização do Stuffit Expander para extrair o ficheiro é:

1. Abra o terminal.
2. digite "tar -zxf " e NÃO prima return.
3. Vá ao "finder". Arraste o ficheiro FirebirdCS-1.0.0-Beta2-Darwin.tar.gz que fez download e largue-o na janela terminal.
4. Volte à janela terminal e prima return.

Para instalar o software deverá ter permissões de administrador. Se não possuir privilégios de administrador poderá obtê-los desta forma:

1. Vá a System Preferences->Users.
2. Duplo-clique no seu nome de utilizador.
3. seleccione o "tab" da password.
4. Clique no icon lock no canto inferior esquerdo da janela e peça a alguém com direitos de administração para introduzir a password.
5. Clique na "checkbox" "Allow user to administer this computer".

Assim que tiver direitos de administração pode instalar o Firebird da seguinte forma:

1. Abra o terminal (pode utilizar uma janela de terminal existente)
2. Escreva "cd " e não prima "return".
3. Vá ao Finder. Seleccione a pasta "firebird_install" criada pelo download descomprimido. Arraste esta pasta e largue-a na janela do Terminal.
4. Vá à janela do terminal e prima return.
5. Escreva "chmod a+x install" e prima return.
6. Escreva "./install" e prima return.
7. Pode ver um erro "no such process...". Este erro pode ser ignorado.

Se tiver problemas na instalação ou na utilização do Firebird não sofra em silêncio. Existe um grande número de fóruns online que providenciam um excelente suporte técnico. Em particular, veja o grupo "ib-support" localizado em <http://groups.yahoo.com/>.

A instalação termina aqui, o Firebird está instalado e preparado para trabalhar.

Os utilitários standard de linha de comandos do Firebird estão instalados em /Library/Frameworks/Firebird.framework/Resources/bin. Deve considerar adicionar este caminho à sua "path". Não existe nenhum utilitário GUI de momento. Toda a documentação usual do Interbase/Firebird aplica-se a esta versão para MacOS X.

Para os utilizadores mais curiosos, o "script" de instalação elabora as seguintes alterações ao seu computador:

1. Instala a "framework" em /Library/Frameworks
2. Instala e carrega as extensões "system v semaphore kernel" que o CS necessita. A extensão é instalada em /Library/StartupItems o que assegura que é carregada de cada vez que o computador reinicia.
3. Adiciona um utilizador firebird ao seu computador. Sempre que possível os servidores firebird e os processos em "background" usam este utilizador para aumentar a segurança. Se instalar o SS então o utilizador firebird necessita de ter direitos de leitura/escrita em todas as bases de dados.
4. Adiciona a entrada gds_db no ficheiro /etc/services e em netinfo.
5. Instala o servidor de base de dados desta forma:
 - a. Na versão CS, cria uma entrada no ficheiro inetd.conf.
 - b. Na versão SS, adiciona o script firebird em /Library/StartupItems e inicia o servidor.

Remoção da Instalação

1. Não existe um script para remoção. Para remover o firebird faça o seguinte, como root:
2. Apague /Library/Framework/Firebird.framework, /Library/StartupItems/FirebirdStartupItem, e os /Library/StartupItems/SysV Semaphores.
3. Comente (ou apague) a entrada do firebird em /etc/inetd.conf e HUP inetd (pode fazer restart do computador para reler o HUP).
4. Apague o utilizador firebird em System Preferences->Users (opcional)
5. Remova a entrada gds_db do /etc/services e netinfo (opcional)
6. Reinicie o computador. Esta operação remove o semáforo SysV Semaphore kext. Não é possível de momento remover o semáforo sem reinicializar o computador.

John Bellardo
<bellardo@cs.ucsd.edu>

Build ou Instalar Firebird em FreeBSD

(Geoffrey Speicher, actualizado por Chris Knight)

As versões de FreeBSD anteriores à 4.4 terão de possuir as livrarias DES crypt libraries instaladas e definidas como as livrarias crypt por defeito. A forma mais fácil de fazer isto é:

```
# ls -l /usr/lib/libcrypt.so lrwxr-xr-x 1 root wheel 14 Apr 24 2001
/usr/lib/libcrypt.so@ ->libdescrypt.so
```

Se tiver libcrypt.so depois do ->, então possui umas "MD5 crypt libraries" por defeito. Deverá correr o *sysinstall* para adicionar a distribuição crypto.

Isto pode trazer problemas com outros programas, por isso teste numa "box" que não está em produção. Se não estiver certo, instalar o Firebird em FreeBSD 4.4 ou superior é recomendado, pois o interface da "crypt library" é mantido automaticamente.

A forma recomendada é "build" e instalar a versão (como root):

```
# cd /usr/ports/databases/firebird
# make install
```

Uma alternativa é instalar a versão do "package" (como root) com o auxílio de pkg_add:

```
# pkg_add http://prdownloads.sourceforge.net/firebird/firebird-1.0_xx.tgz
```

onde xx é a versão de FreeBSD que o package foi construído para.

Por exemplo, para FreeBSD 4.5, deverá correr:

```
# pkg_add http://prdownloads.sourceforge.net/firebird/firebird-1.0_45.tgz
```

Driver Puro Java JCA-JDBC XA Capable

A equipa do Firebird implementou um driver puro Java para o Firebird. Este driver baseia-se tanto no novo JCA standard para conexão a servidores de aplicações para sistemas de informação empresariais como no bem conhecido JDBC standard.

O standard JCA especifica uma arquitectura em que um servidor de aplicações pode cooperar com um driver de forma a que o servidor de aplicações gere transacções, segurança e alocação de recursos, e o driver apenas fornece a funcionalidade de conexão. Embora semelhante à ideia JDBC 2 XADataSource, a especificação JCA é consideravelmente mais clara na divisão de responsabilidades entre o servidor de aplicações e o driver.

Funcionalidades Implementadas

- ❑ Maior parte das funcionalidades JDBC mais úteis ("mais úteis " na opinião dos programadores).
- ❑ Suporte completo a jca spi: pode ser utilizado directamente em servidores de aplicações jca -como por exemplo o JBoss.
- ❑ Transacções XA com commit de duas fases reais, quando utilizado como um recurso adaptador jca num ambiente de gestão (com um TransactionManager e suporte jca deployment).
- ❑ Inclui uma opcional "internal connection pooling" para uso isolado e uso em ambientes não-jca como o Tomcat 4.
- ❑ Implementação ObjectFactory para uso em ambientes com JNDI mas sem TransactionManager como o Tomcat 4.
- ❑ Implementações de DataSource com e sem pooling.
- ❑ Implementação do Driver para utilização em "legacy applications".
- ❑ Acesso completo às definições de parâmetros do Firebird: "parameter block" e "transaction parameter block".
- ❑ "integrated logging" opcional através de log4j.
- ❑ JMX mbean para gestão da base de dados (por enquanto apenas CREATE e DROP).

Instalação

Em todos os casos as classes para os seguintes pacotes deverão estar disponíveis:

concurrent.jar
connector.jar
jaas.jar
jta-spec1_0_1.jar
log4j-core.jar

e ainda as classes jdbc 2 e 3.

Pode utilizar o tanto num agente jmx (MbeanServer) ou como uma classe individual. Por enquanto está testada apenas em jmxri, mas como é muito simples não deverá existir problemas em qualquer implementação jmx.

Se utilizar num ambiente de gestão como um "jca resource adapter", forneça o firebirdsql.rar de acordo com mecanismo de fornecimento existente. Por exemplo, se utilizar JBoss 3, coloque o ficheiro na directoria de fornecimento (deploy directory). Terá de configurar ConnectionFactories de acordo com os requisitos do servidor de aplicações. Um exemplo de configuração para JBoss 3 está no manual online do JBoss.

Para todos os ambientes que não suportam "jca deployment", torne as classes em firebirdsql.jar disponíveis para a sua aplicação.

Para utilização num ambiente com JNDI mas sem suporte para jca ou transaction manager, use a FBDataSourceObjectFactory para criar uma referência à DataSource no jndi. Tomcat 4 é um exemplo deste cenário. A implementação JNDI necessita de suportar a utilização de References/Referenceable. Isto não irá funcionar na implementação de JNDI que apenas suporta "binding serialized objects".

Para utilização numa aplicação isolada com múltiplas conexões que poderão beneficiar de "connection pooling", utilize uma instância de FBWrappingDataSource configurada para "pooling".

Par uma aplicação standalone que necessita apenas de uma conexão, utilize ou o FBWrappingDataSource ou o FBDriver.

David Jencks, d_jencks@users.sourceforge.net

O Suporte é fornecido no fórum Firebird-Java (espelhado por news://news.atkin.com)

Firebird-Java@yahoogroups.com

Driver InterClient JDBC

A equipa do Firebird corrigiu vários bugs que existiam no driver Interclient 2 original lançado pela Borland. O Firebird possui pacotes de instalação tanto para Windows como para Linux. Estão disponíveis nas áreas de download.

Encorajamos todos para utilizarem o driver pure-Java (veja em cima) e verificar se resolve as suas necessidades. O suporte é fornecido em Firebird-Java@yahoogroups.com e na lista de desenvolvimento. Quando colocar a sua mensagem, por favor refira claramente que está a colocar uma questão de InterClient.

Mais Informação

Poderá obter mais informação sobre o Firebird em:

<http://firebird.sourceforge.net>

ou nos sites afiliados:

<http://firebirdsql.org>

<http://www.ibphoenix.com>

<http://www.cvalde.com>

Se está interessado em participar no desenvolvimento do Firebird, ou se deseja colocar um possível bug para discussão, junte-se à nossa lista de discussão firebird-devel. Para subscrever, envie simplesmente um "e-mail" em branco para:

firebird-devel-request@lists.sourceforge.net

com a palavra "subscribe" no campo do Assunto.

Por favor não utilize a lista firebird-devel para perguntas de suporte.

Para suporte técnico, utilize a lista ib-support em

<http://www.yahoogroups.com/groups/ib-support>

Para InterClient e desenvolvimento Java, o suporte tem uma lista especializada:

<http://www.yahoogroups.com/groups/Firebird-Java>

A lista ib-support abarca questões técnicas sobre o Firebird e o InterBase(R). Por favor não traga as suas questões de Delphi ou outras linguagens de programação para este fórum.

A comunidade "open-source" trabalha com várias listas de discussão no desenvolvimento do Firebird. Por favor refira às Mail Lists e Newsgroups em [Firebird community site](#).

A lista de programadores Firebird e a lista geral da comunidade, assim como outras listas de interesse para programadores Firebird e InterBase, são espelhadas como "newsgroups" em

<news://news.atkin.com>

Suporte Pago ao Firebird pode ser obtido através do IBPhoenix (os endereços de contacto assim como os números de contactos estão disponíveis em <http://www.ibphoenix.com>). Alguns membros da equipa Firebird estão também disponíveis para suporte ou consultoria. Por favor contacte-os directamente.

Serviços de Recuperação de Bases de Dados em Firebird ou InterBase podem ser executados por [IBPhoenix](#).

Ofertas para patrocinar melhoramentos ao Firebird devem ser enviados directamente para a equipa do Firebird enviando um mail para firebird-contact@lists.sourceforge.net.

Discussão Geral sobre melhorias ao FB podem ser obtidas na lista Firebird-priorities:

<http://www.yahoogroups.com/community/Firebird-priorities>.

IB-Architect (<http://www.yahoogroups.com/community/ib-architect>) é apenas para **discussões técnicas de design APENAS**. Questões de Suporte/conversão não são bem vindas aqui.

Documentação

A documentação do InterBase v 6.0 também se aplica a esta versão do Firebird. Uma versão beta dos manuais em formato Adobe Acrobat pode ser obtida em

<http://www.ibphoenix.com/downloads/60All.zip>

Um índice da Documentação é mantido no site da comunidade Firebird em

<http://firebird.sourceforge.net/index.php?op=doc>

Isto é "work-in-progress" e todas as contribuições são bem vindas - envie uma mensagem para firebird-docs@lists.sourceforge.net

Alguns guias de instalação e alguns "HowTos" podem ser obtidos na área de documentação em:

<http://www.firebirdsql.org>

Ou mais directamente em

<http://sourceforge.net/projects/firebird>

O principal local para conselhos de utilização e técnicos no site da IBPhoenix -

<http://www.ibphoenix.com>

E alguma documentação adicional pode ser obtida visitando a área técnica da Borland:

<http://www.borland.com/techpubs/interbase/>

Bugs Corrigidos desde o Lançamento do Código Fonte

SFID 492181.- Problemas de "performance" em campos Blob utilizando JDBC

O acesso JDBC era extremamente lento (cerca de 50Kb/seg). Foi observado que o problema poderia ser resolvido aumentando o tamanho do buffer do BLOB (org.firebirdsql.jdbc.FBBlob.bufferlength). Por exemplo, aumentando o buffer de 2Kb para 20Kb aumentava-se a velocidade por um factor de 9. Corrigido.

SFID 527669.- Cliente Linux não suporta conexões em "thread"

Foi reportado que a library cliente do Linux (gdslib.so) não suportava a possibilidade de abrir/usar conexões em "threads" individuais na aplicação. Corrigido na versão 1. NOTE que a gdslib.so não é ainda "thread-safe".

SFID 217138.- JOIN com uma view complexa pára o servidor

"Joining" views complexas cujas tabelas inerentes possuem índices nos campos utilizados pelo join deitam abaixo o servidor imediatamente. Corrigido.

SFID 216733.- Muitos Generators podem corromper a base de dados

O número de generators que pode utilizar no IB está dependente da relação (tamanho da página-overhead desconhecido) / tamanho do generator. Pode criar generators que ultrapassem este limite sem qualquer retorno da base de dados, mas estes generators irão retornar informação e corromper a base de dados. Corrigido.

SFID 225283.- ORDER BY numa view faz surgir valores NULL nos campos.

O problema foi resolvido na versão 1.0 por Claudio Valderrama.

SFID 228135.- NULL é retornado como zero num left join numa view simples.

O problema foi resolvido na versão 1.0 por Claudio Valderrama.

SFID 518279.- Left join causa o erro "string truncation".

O problema foi resolvido na versão 1.0 por Claudio Valderrama.

SFID 514973.- Ficheiros Temporários de Ordenação > 2GB Falham

O problema foi corrigido na Versão 1.0 por Mike Grover.

SFID unknown.- Em Linux, a library gdslib.so não é "thread-safe".

O problema foi resolvido na versão 1.0.

SFID 518273.- Divisão de Números negativos em Dialecto 3 altera os dados.

"O Bug Klapperich" - descoberto em Fevereiro de 2002 por Guido Klapperich - estava inserido no source do IB 6. Confirmado, reportado e corrigido no Firebird 1 por Claudio Valderrama.

SFID 213462.- O "Path String Bug" no Windows foi corrigido.

O InterBase tinha um bug horrível que foi identificado na versão 5 do IB e ignorado na versão 6 beta. Corrompia os dados de uma forma particularmente horrível.

O caminho correcto numa "string" de conexão a um servidor Windows é:

```
C:\caminho\BaseDeDados.gdb
```

Porém, o Windows permite

```
C:caminho\BaseDeDados.gdb
```

Se dois utilizadores conectarem, um utilizando o primeiro sintaxe e o outro o segundo, o servidor recolhe mensagens erradas do Windows e julga que os utilizadores estão a conectar-se a diferentes bases de dados. As transacções de cada utilizador não estão a par umas das outras, e o mais provável é a base de dados ficar corrompida.

Com esta correcção, a primeira conexão de um utilizador estabelece um "exclusive access lock" sobre a base de dados, com base na "string" de conexão utilizada. Outro utilizador que tente conectar-se com uma "string" alternativa irá obter o seguinte erro:

```
I/O error for file "C:caminho\BaseDeDados.gdb"
Error while trying to open file
The process cannot access the file because it is being
used by another process.
```

Uma anomalia da RC1, envolvendo esta correcção e a possibilidade de definir "Forced Writes" numa base de dados em utilização, foi também corrigida.

SFID 448062.- **CREATE DOMAIN ... CHECK (condição)** leva a rdb\$fields.rdb\$validation_source a assumir CHECK (condição) mas ALTER DOMAIN ... ADD CONSTRAINT CHECK (condição) leva a que rdb\$fields.rdb\$validation_source seja CONSTRAINT CHECK (condição). Este facto é inconsistente. Agora, o primeiro formato é sempre guardado; o CONSTRAINT word extra é sempre omitido.
Correcção financiada por Jason Wharton, CPS (IB Objects)

SFID 227760.- Os identificadores de tamanho zero são agora proibidos

A estrutura XSQLVAR utilizada para transmitir informação para a aplicação cliente não traz informação que permita distinguir entre "no ident (NULL)" e "blank ident". Assim, os identificadores em branco (nomes de tamanho zero) são proibidos. O servidor internamente executa um "trim" dos brancos consequentes, pelo que um nome do tipo " " (apenas espaços) torna-se um identificador de tamanho zero e também este não é permitido. A mesma correcção foi aplicada tanto ao sintaxe CREATE como ao ALTER e para todos os tipos de objectos. Mais ainda: vários locais onde o "engine" não podia trabalhar com "embedded" ou espaços em branco nos identificadores de dialecto 3 (uso legítimo) foram corrigidos.

Correcção financiada por Jason Wharton, CPS (IB Objects)

SFID 428889.- Posição da Coluna era tratada como base 0, mas deveria ser base 1

O sintaxe

```
alter table...alter column...position <n>;
```

é de base-1 no Firebird. Num nível lógico, é tratada da mesma forma que ORDER BY <n>, ignorando o facto que, internamente, o "engine" utiliza a convenção do C e inicia a zero.

Assim, para colocar alguns campos na primeira posição, o comando é:

```
alter table tbl
alter column cln position 1;
```

Existe agora uma incompatibilidade entre o Firebird e o Interbase, pois o Firebird foi corrigido para ir de encontro tanto ao standard como à documentação do IB6. No EmbedSQL.pdf, capítulo 5:

...

O comando ALTER TABLE ALTER permite que altere a posição da coluna assim como o seu nome.

Por exemplo, a declaração seguinte move uma coluna, EMP_NO, da terceira para a Segunda posição na tabela EMPLOYEE:

```
ALTER TABLE EMPLOYEE ALTER EMP_NO POSITION 2;
```

...

O exemplo indicado, tirado do manual, funciona em Firebird. Em IB não faz nada, pois 2 é a terceira posição em InterBase 6, cujo sintaxe ainda possui o bug de tratar as posições de colunas como se fossem baseadas em zero.

SFID 228526.- **Declarações JOIN Ambíguas são agora rejeitadas**

O InterBase permite declarações como esta:

```
SELECT A.FIELD1, B.FIELD1  
FROM A JOIN B  
ON FIELDX = FIELDY  
WHERE FIELD1="99"  
ORDER BY FIELD1
```

Estas declarações retornam dados não previsíveis. Agora o Firebird, em Dialecto 3, retorna um erro se existir algum campo não qualificado em declarações JOIN. No Dialecto 1, irá retornar um "warning" mas a query ambígua é autorizada a proceder.

SFID 223133.- **"self join" ambíguo produz resultados bizarros**

Também corrigido.

SFID 460261.- **"Blob API" com problemas com nomes com espaços embebidos**

As seguintes chamadas de API falham se um nome dialecto 3 contém espaços embebidos:

```
isc_blob_default_desc  
isc_blob_lookup_desc  
isc_blob_set_desc
```

A causa, a função get_name() em blob.e, foi corrigida.

SFID 436462.- **Valores incorrectos de "Rows afectadas" num trigger BEFORE UPDATE em views**

Os valores de "Rows Affected" (linhas que foram afectadas) agora indicam operações virtuais, não operações físicas. Tenha em atenção que esta forma de actuação em views não-actualizáveis com triggers irão sempre incrementar em uma unidade cada operação física.

SFID 444463.- **Triggers BEFORE eram disparados depois das verificações**

Triggers do tipo Before (insert e update) eram disparados depois de verificações CHECK, o que permitia a alteração de valores que falhavam as verificações da tabela (check constraints). Esta operação era errada e poderia produzir dados que não poderiam ser repostos após uma cópia de segurança.

SFID 229009.- **CREATE VIEW não retornava “erro de sintaxe”**

O IB6 permitia o comando CREATE VIEW em que o numero das colunas no view não correspondiam ao número de colunas da declaração select. De acordo com as regras do SQL92, estas declarações deveriam retornar um erro de sintaxe. É o que faz o Firebird.

SFID 458888.- **Privilégios de REFERENCES implícitos podiam causar “crashes”**

O bug reportado indicava que uma tabela dependente necessitava que os privilégios REFERENCES fossem declarados explicitamente para o seu “foreign key trigger” tivesse permissões na tabela que referenciava. Isto foi corrigido de forma que uma tabela que possua um trigger interno adquira estes privilégios na outra tabela. O código que verifica as REFERENCES em “run-time” (no tempo DML) foi colocado com comentário e a resposta ao requisitado pela trigger foi suavizado de forma que as condições para o bug não possam surgir.

A análise deste bug descobriu mais dois bugs, que também foram corrigidos:

- Verificação de REFERENCES em “design-time” (DDL) era susceptível de ser ultrapassada, pelo que foi adicionado código para forçá-la, de forma que não possa ocorrer um “bypass”. Agora é necessário para um criador de uma “foreign key” possuir privilégios de REFERENCES na tabela mestre ou ser o “owner” da tabela mestre.

- A Correção do bug também corrigiu um bug que foi discutido mas não documentado. Este bug de REFERENCES era também indirectamente responsável por uma situação que foi reportada, onde elaborar o “backup” e o “restore” de uma base de dados complexa poderia causar o crash do servidor depois de seleccionar dados de qualquer tabela. A análise do bug, que não foi simples, descobriu um “deadly circle” de verificação reentrante de REFERENCES.

SFID 446237.- **Erro “Column not found” ocorrido onde não deveria ocorrer**

Eliminado. Surgia quando uma view era baseada numa tabela cujos campos eram campos dialecto 3 com espaços embebidos. Mesmo que este seja o único bug documentado, era um dos mais graves bugs com espaços embebidos que foram corrigidos.

SFID 229860.- **Mensagem de Erro Errada**

A mensagem “ DATE data type is now called TIMESTAMP” estava a ser retornada em erros não relacionados, tais como “typos”. Corrigido.

SFID 460624.- **Extracção incorrecta de Parâmetros de Procedimentos em isql - Dialecto 3**

Resolvido um “bug” no isql onde este não extraía correctamente os nomes dos parâmetros no dialecto 3, se os nomes exigissem-se “double quotes” e estas não fossem fornecidas. Uma declaração de “parâmetro incorrecto” na rotina que elabora as “double quotes” também foi resolvida.

SFID 451798.- **FIRST é aplicado antes da agregação**

SFID 451810.- **SKIP está errado em 1**

Ambos resolvidos.

SFID 412417.- **Erro ao alterar de CHAR para VARCHAR**

A alteração de uma coluna de CHAR para VARCHAR estava a adicionar 2 bytes ao tamanho do campo. Mais: era possível alterar de CHAR para VARCHAR de tamanho inferior, o que causava mensagens de erros de "string truncation" quando se seleccionava dados dessa tabela danificada. Resolvido.

SFID 231998.- Espaço antes do CAST de uma expressão numérica em dialecto 1

Eliminado.

SFID 212177.- erro com colunas com "default's" com caracteres internacionais

O GBAK indicava um erro "transliteration error" durante a reposição de cópias de segurança se os valores por defeito guardados numa base de dados possuíam caracteres internacionais.

SFID 221589.- Campos numéricos e operações matemáticas

```
select field1 * field2 from mytable
OU
select field1 * (1+field2/100) from mytable
```

onde ambos os campos eram numeric (9,2) produziam resultados incorrectos.

SFID 450301.- SUBSTRING não funcionava

Esta nova função não funcionava quando incluída numa clausula "where" (isto é, com in ou =) ou ainda em concatenações de strings. Resolvido.

SFID 223059.- Actualização de VARCHAR não limpava dados antigos

Quando o IB 6 actualiza uma "string" guardada em VARCHAR, não coloca a zero o resto da "string" mas concatena o valor antigo ao valor actualizado. Como os campos VARCHARs contém o tamanho da "string", a aplicação cliente nunca iria reparar no problema (i.e. iria sempre obter o resultado correcto), mas o ficheiro gdb iria crescer mais depressa que o esperado (porque estes dados adicionais não seriam comprimidos), e a base de dados seria mais lenta (porque menos dados úteis iriam caber numa única página).

Resolvido no Firebird.

SFID 223512.- DROP VIEW causava um DROP à tabela com o mesmo nome

Significa que as tabelas e as views partilham o mesmo "namespace" mas DROP VIEW só devia actuar se existisse uma VIEW com esse nome. Se existir uma tabela com o mesmo nome, o comando deveria proceder como se esta não existisse. Corrigido.

SFID 226456.- SELECT/PLAN não compreendia nomes de índices SQL delimitados

Corrigido.

SFID 419964.- "buffer overflow" em remote/interface.c li

Era causado pelo buffer da "version string" ser apenas de 64 bytes. Foi aumentado.

SFID 453686.- Impossível de criar bases de dados dialecto 3 com Firebird 1 betas

Este problema ocorria com clientes (por exemplo IBConsole e IBExpert) que necessitavam de se conectar a uma versão superior a 5. Foi corrigido no Firebird RC 1.

SFID 233124.- Perca de conexão durante a execução de código errado

Eliminado.

SFID 425799.- Domain renomeado quebrava as suas dimensões

Com um domain definido como um array, mudar o nome do domain (rename) causava o rdb\$field_dimensions a permanecer inalterado: a relação entre um domain e a sua dimensão estava quebrada. Corrigido.

SFID 450405.- Role com "truque" poderia comprometer a segurança SQL básica

Foi dado um exemplo estranho: mas o "loophole" foi agora encerrado.

SFID 462800.- Par não-único em RDB\$FORMATS

Corrigido.

SFID 447377.- GDS erro ..."can't find TIP"

Existe um bug no InterBase 5.6, 6.01, e em algumas versões beta do Firebird que faz falhar o "lookup" de uma "transaction inventory page" se existirem mais de 32767 "transaction pages". Isto faz que os valores máximos de um ID de uma transação com:

```
1024 byte pages sejam de 131,596,287.  
2048 byte pages sejam de 265,814,016.  
4096 byte pages sejam de 534,249,472.  
8192 byte pages sejam de 1,071,120,384.
```

Embora estes números sejam de uma grande escala, deu-se o caso de uma base de dados que excedia mais de 131 milhões de transacções em seis meses. A tentativa de conectar à base de dados retornava o erro "gds internal consistency check, can't find TIP".

Sugestões:

- 1) não utilize uma page size de 1024 byte.
- 2) Verifique a sua "next transaction number" de tempos a tempos.
- 3) Se verificar que o "next transaction number" está a aproximar-se do limite, faça um backup/restore da base de dados.

SFID 229231.- REVOKE é sensível a Maiúsculas/Minúsculas dos nomes de utilizadores

Por exemplo:

```
grant all on config to "admin";  
revoke all on config from "admin";
```

À primeira vista este comando deveria resultar no utilizador "admin" não ter qualquer privilégio em "config" mas um select na tabela RDB\$USER_PRIVILEGE mostra que "admin" ainda tem acesso total! Para que se remova esses privilégios terá de executar:

```
revoke all on config from "ADMIN"
```

Adicionalmente, foi removida uma ambiguidade que existe no IB desde a versão 4:

```
grant update on tbl to <unknown>
```

O que é "unknown", uma "role" ou um "user"? Se uma "role" existir, é atribuída os direitos à mesma. Caso contrário, é atribuído os direitos ao "user". Mesmo que especifique a palavra chave USER, o IB não a contempla. Porém, no Firebird, a declaração anterior ainda é ambígua porque o Firebird permite que especifique se os direitos são atribuídos ao "user" ou à "role".

- ❑ Para um USER desconhecido o FB irá procurar apenas um USER, mesmo que exista uma ROLE.
- ❑ Para uma ROLE desconhecida o FB irá procurar apenas a ROLE. Erro se a ROLE não existe.

Ainda existe um bug em aberto (SFID 223128) que permite ao SYSDBA atribuir direitos a roles não existentes.

Um caso referido por Ivan Prenosil, onde o isql não conseguia copiar com o sintaxe "grant ... TO GROUP UNIX_group" também foi resolvido.

SFID 421260.- Comprimento do Caracter não preenchido por UDFs

Em ODS 10, foi escrito apenas a "stub". Não completado.

SFID 227375.- Agrupar em campos derivados processando valores NULL crasha o InterBase

Este bug ocorre quando uma declaração SELECT numa view que calcula um valor de um campo originado pela diferença de dois campos e se tenta elaborar um GROUP BY nesta coluna. O funcionamento em Firebird é correcto.

SFID 425949.- Engine CRASH Error

Esta declaração

```
select count(*),adresy.rdb$db_key from adresy
```

pode fazer "crashar" o InterBase. adresy pode ser uma tabela qualquer. Correção copiada para a árvore do Firebird a partir da árvore da Borland.

SFID 228467.- Bug de Segurança com um utilizador "hardcoded" com direitos totais a isc4.gdb

Esta foi a vulnerabilidade de segurança descoberta que afecta todas as versões do Interbase desde a Versão 4.x em diante. Foi corrigido no Firebird e também posteriormente pela Borland no InterBase 6, tanto no código como nos binários depois de Janeiro de 2001.

SFID 229121.- Directoria TEMP sendo preenchida

Corrigido.

SFID 213708 -Erro 502 "Declared cursor already exists"

Ocorre em programas Microfocus que se conectam localmente ao IB6 Classic no AIX e remotamente ao IB6 Superserver em WINNT. Corrigido.

SFID 214298.- Anomalia na expressão Select count(*) quando a tabela está vazia

select count(*) + 1 retornava zero quando a tabela não tinha quaisquer linhas mas retornava o valor correcto se count(*) fosse maior que zero. Corrigido.

SFID 216579.- Generators nas colunas COMPUTED BY retornam valores errados

A utilização de generators em colunas COMPUTED BY devolviam valores errados e tornavam a base de dados inutilizável. Corrigido.

SFID 222476.- **AVG e SUM retornam nomes de campos vazios no dialecto 3**

Corrigido. O mesmo problema foi resolvido para CAST e UPPER.

SFID 227717.- **Programas em COBOL retornam aleatoriamente um erro -901 request sync.**

Os programas em COBOL aleatoriamente retornam um erro -901 request synchronization. Título alternativo: Programas em COBOL perdem valores SQLCODE durante um UPDATE:

```
UPDATE SET ... WHERE x=..
```

Códigos de erro (“Mass Update”) retornados pelo update não eram retornados ao programa, e o programa recebia um erro “-901 request synchronization error”.

O erro era causado por código mal gerado pelo GPRE. Corrigido.

SFID 212328.- **IB Guardian deixava par trás alguns “handles”**

Corrigido. Porém, continua a criar mais “threads” cada vez que o FB crasha, até ser feito um shutdown ou ser duplo-clicado o icon no tray. Este bug apenas ocorre com o Guardian a correr como uma aplicação e não como um serviço, pois desta ultima forma não coloca um icon no tray.

SFID 212263.- **linha de comandos isql ignora -user / -password com opções -a ou -x**

Corrigido.

SFID 421262.- **ISQL reporta o parâmetro UDF BLOB por valor (“BY VALUE”)**

Na distribuição Firebird, foram corrigidos os BLOB UDFs em ibudf.

SFID 222563.- **isql extrai parâmetros incorrectos de stored proc com UNICODE**

O ISQL estava a ler rdb\$field_length em vez de rdb\$character_length em parâmetros de procedimentos. No caso dos campos de tabelas a informação correcta era lida e apresentada. O “engine” por si mesmo estava a actuar correctamente. Corrigido.

SFID 223516.- **Tipos desconhecidos em rdb\$types**

Vários, mas apenas os tipos não documentados foram adicionados.

SFID 451944.- **Trigger Activate/Deactivate aumentava o contador da metadata**

Corrigido.

Problemas Conhecidos

Pendentes

Permitir a desactivação de índices FK

Um aperfeiçoamento que permita desactivar os índices FOREIGN KEY, utilizando ALTER INDEX, para resolver problemas de performance relacionados com a manutenção de índices com uma baixa selectividade, foi adiado de forma a que se possa analisar os efeitos relacionados.

SFID 446227.- **Em Linux, a livraria cliente gdslib.so não é "thread-safe".**

Uma "threadable gdslib.so" foi incluída na release 1.mas nem todos os problemas relacionados com "threads" foram resolvidos.

Core Engine

SFID

479483 Mau tratamento de FIRST/SKIP num subselect
224810 DISTINCT propaga-se fora de uma VIEW
442140 Grant em Roles de Views não funcionam
222376 PLAN horrível com muitas condições OR
223514 IB crasha com duas procedures intermixed.
217042 IB não valida "weird constructions"
419065 Join em diferentes datatypes
223058 Multi-hop server ability broken Bug Confirmado
219525 "No current record" para uma operação de fetch
221921 ORDER BY não tem qualquer efeito
213460 Registos de Eventos com algumas configurações fazem o servidor "crashar"
233025 O Servidor pára quando executa uma Stored Proc mais que uma vez
221649 "Unique index" permitido em campos NULLABLE
211781 Win32: O Servidor não fecha "thread handles"
233644 não se pode especificar PLAN numa declaração UPDATE

DSQL

GBAK

228431 O gbak não pode elaborar um "restore" de um "backup" feito por IB v5

GPRE

416228 gpre gera chamadas isc_vtov invalidas
223513 Ambiguidade entre tabelas e views em comandos SHOW de isql.
223126 Collation mal colocada quando se extrai metadata com isql
225219 isql -a: ordem errada com domains baseados em campos de tabelas
450404 isql torna a role em maiúsculas na linha de comandos

InterClient

227414 Algumas vezes o Interbase/Interserver não permite conexões

Security Issues

229237 Passwords em branco são fracamente suportadas

229894 O cliente pode autenticar-se como qualquer utilizador.

222375 Grants escreve por cima de entradas prévias em rdb\$security_classes

Notas do Tradutor

Mantive na língua original alguns termos ingleses: expressões como 'bug', 'Backup', 'string', etc, são utilizadas na língua portuguesa de uma forma tão comum, que qualquer tentativa de tradução só iria produzir um estranho contexto.

Qualquer comentário ou sugestão pode ser dirigido para mim em artur@arsoft.pt.