



Firebird 1.5 クイック・スタート・ガイド

IBPhoenix Editors

Updates, additions: Firebird Project

2 December 2005 – Document version 2.2 ...between revisions...

Translation for Japanese: Tsutomu Hayashi

目次

クイック・スタート・ガイドについて	3
バイナリキットには何が含まれているのか?	3
Classicサーバー か Superサーバー か?	3
インストール先の既定値	5
Firebirdのインストール	7
Firebirdサーバーのインストール	7
インストール後のテスト	9
その他の必要条件	13
クライアントのみのインストール	15
データベース上での作業	16
サンプル・データベースへの接続	17
isqlを使用したデータベースの作成	19
Firebird SQL	20
セキュリティ・ガイドライン	24
サーバーのセキュリティオプション	24
Backup	24
データベースの壊し方（やってはいけないこと）	25
Helpの探し方	27
Firebirdプロジェクト	27
A. Document History	29
B. License notice	31
Alphabetical index	32

クイック・スタート・ガイドについて

このガイドは全くの初心者が、Firebirdのバイナリキットを利用してクイック・スタートするための少々のエッセンスをまとめたものです。サーバー上での実行に関しての詳細な設定やチューニングについては4-6節で紹介している Using Firebirdマニュアル ([IBPhoenix社監修のCD所収](#))を参照して下さい。

Firebird クイック・スタート・ガイドは、[IBPhoenix社](#)が販売するUsing Firebirdマニュアルの Chapter 1 として記述されたのがその始まりです。後になって分離されてインターネット上で公開されることになりました。2004年6月に、IBPhoenix社は本ガイドを Firebird Project に寄贈したのです。

重要項目

本書をお読みなる前に、お使いのFirebirdのバージョンが本ガイドと一致するかどうかお確かめ下さい。本ガイドはバージョン1.5又はそれ以上について記載されています。もしもお使いのFirebirdのバージョンが1.0, 1.0.2, 又は1.0.3である場合は、適切なバージョンのクイック・スタート・ガイドを次のURLから入手して下さい。

<http://www.firebirdsql.org/manual/> (HTML)

又は

<http://www.firebirdsql.org/pdfmanual/> (PDF).

バイナリキットには何が含まれているのか?

全てのバイナリキットにはFirebirdサーバーのインストールに必要なものは全て含まれています:

- ・ Firebirdサーバー
- ・ サーバマシン上に配置されるクライアントライブラリ
- ・ コマンドラインツール群
- ・ 標準ユーザ定義関数ライブラリ群
- ・ サンプルデータベース
- ・ C言語ヘッダファイル (初心者には必要無いものですが)
- ・ リリースノート - 必ずお読み下さい

Classicサーバー か Superサーバー か?

Firebirdは2種類があり、アーキテクチャから ClassicサーバーとSuperサーバーと呼ばれています。どちらをインストールすべきでしょうか？それは、読者のおかれている状況によっています。両者の重要な相違点について、以下に簡単な概要をしめします。

表 1. Firebird 1.5 Classicサーバー vs. Superサーバー

Classicサーバー	Superサーバー
<p>On Linux: 完全に成熟している</p> <p>On Windows: version 1.5.1 から完全に成熟している</p>	<p>Windows と Linuxの両方で完全に成熟している.</p>
<p>クライアントからの接続毎に独立したプロセスを生成し、各プロセスは独立したキャッシュを保持します。接続数が少なければリソースの消費は少なくなります。</p>	<p>全接続に対して単一のサーバープロセスが対応します、リクエストの処理はスレッドにて行われます。同時接続数が増加した場合により効果的です。</p>
<p>Linux上のローカル接続では、高速なダイレクトI/Oによるデータベースファイルへのアクセスが行えます。Windows上では、ローカル接続はネットワーク・スタイルで行わなくてはならず、localhostを使用して接続を行います。</p>	<p>Linux上では、ローカル接続はネットワークスタイルで行われ、localhostを（暗黙的にも）使用します。Windows上では、オプションとしてダイレクトなローカル接続を行うことが可能ですが、Linux上の「Classic」サーバーほど高速ではなく、また安全性も多少犠牲になります。</p>
<p>1.5: Services Managerは部分的に実装されていますので、バックアップ/リストアやデータベースのシャットダウン等をネットワーク越しに行うことが出来ます。その他の管理作業はFirebirdと一緒に提供されている各クライアントツール群(小さな実行ファイル)を利用して行います。</p> <p>1.5.1以上:完全版の Service Manager が実装されています。</p>	<p>完全版の Services Manager (on Windows and Linux) が提供されています。全ての管理作業（バックアップ/リストア、データベースのシャットダウン、ユーザー管理、統計処理、他）をプログラム上で行うことが出来ます。ネットワーク越しにService Managerに接続し、これらの処理をリモートで行うことが出来ます。</p>
<p>SMP (symmetrical multi-processor) サポート。依存関係のない複数のそれほど多くない接続が実行される場合にベターなパフォーマンスを発揮します。</p>	<p>SMP サポートは提供されません。マルチ・プロセッサのWindowsマシン上では、OSがプロセスをCPU間で移動するためにパフォーマンスが劇的に低下する可能性があります。これを防ぐためには、CpuAffinityMaskパラメーターをfirebird.conf設定ファイル中で設定して下さい。</p>

以上のように、どちらがより優れているとは言えない状態です。驚くほどのことではありませんが、もしいずれかのアーキテクチャーが全戦線に渡って勝利を収めているなら、両方

のアーキテクチャーを継続することは無いでしょう。

もし、どちらを選ぶべきか迷った場合（これらの技術的な話題に閉口しているとか）には、以下のルールを使用して下さい。

- ・ Windows では、Superサーバーを選んで下さい
- ・ Linux では、どちらを選んでもかまいません。ほとんどの環境で、パフォーマンスの違いに気づくことは無いでしょう。

注意：後になってから他のアーキテクチャーへ移行することはいつでも可能です。アプリケーションやデータベースには影響を与えません。（ClassicサーバーのService Managerがサポートしていない機能や不完全な機能を使用している場合を除いて）

Linux版Superサーバーは、FirebirdSSで始まるパッケージをダウンロードして下さい。ClassicサーバーはFirebirdCSで始まるパッケージです。Windows版では、インストールパッケージに両方が含まれていますので、インストール・プロセス中に選択することになります。

注意

まだ全てをお話ししたわけではありませんでした。Firebirdには3つ目の選択肢があります： Embedded サーバーですが、これは完全に違うもので、典型的なクライアント-サーバー環境向けではありません。詳細についてはリリースノートをご覧ください。

インストール先の既定値

以下の表ではWindowsとLinuxへのデフォルトのインストール先についての詳細を示します

v. 1.5から開始された大幅なコードベースの改訂によって、InterBase由来の古いハード・リンクは取り除かれ、多くの主要なコンポーネントの名前が変更されました。結果として、Firebird 1.5はInterBase®サーバーがインストールされている環境へもインストールすることが出来るようになりました。（両方のサーバーが同じTCP/IPのポートを開いていないことが前提ですが）この件については、Firebirdと一緒にインストールされるリリースノートの Configuring the port serviceをご覧ください。リリースノートは docにインストールされています。

表 2. Firebird 1.5 インストールされるコンポーネント

プラットフォーム	コンポーネント	ファイル名	既定のインストール先
32-bit and 64-bit Windows (Windows 95, 98, ME, NT, 2000, XP, ...)	インストール先ディレクトリ (以降は <InstallDir> と呼びます)		C:\Program Files\Firebird\Firebird_1_5
	Firebird server	fbserver.exe (SS) or fb_inet_server.exe (CS)	<InstallDir>\bin
	コマンドライン・ツール	gbak.exe, gfix.exe, gstat.exe, etc.	<InstallDir>\bin
	サンプル・データベース	employee.fdb	<InstallDir>\examples
	ユーザー定義関数 (UDF) ライブラリ	ib_udf.dll & fbudf.dll	<InstallDir>\UDF
	Firebirdクライアント	fbclient.dll (with an optional gds32.dll, to support legacy apps)	<InstallDir>\bin (with an optional copy in the Windows System dir - see note below table)

プラットフォーム	コンポーネント	ファイル名	既定のインストール先
Linux 及びその他の利用可能な UNIX システム	インストール先ディレクトリ (以降は <InstallDir> と呼びます)		/opt/firebird
	Firebirdサーバー	fbserver (SS) or fb_inet_server (CS)	<InstallDir>/bin
	コマンドライン・ツール	gbak, gfix, gstat, etc.	<InstallDir>/bin
	サンプル・データベース	employee.fdb	<InstallDir>/examples
	UDFライブラリ	ib_udf.so, fbudf.so	<InstallDir>/UDF
	Firebirdクライアント	libfbclient.so.1.5.n (binary); libfbclient.so.1, libfbclient.so (symlinks) Legacy libgds* symlinks are also installed.	/usr/lib (actually, the real stuff is in <InstallDir>/lib, but you should use the links in /usr/lib)

注意

Windowsのシステム・ディレクトリについて、正確なパスはWindowsのバージョンに依拠しています。 通常は以下のディレクトリとなります:

- ・ for Windows 95/98/ME: C:\Windows\System
- ・ for Windows NT/2000: C:\WINNT\System32
- ・ for Windows XP: C:\Windows\System32

Firebirdのインストール

Firebirdサーバーのインストール

ドライバのインストール

Firebirdサーバー、また作成するデータベースや接続するデータベースは、全て ホストマシンに物理的に接続されたハードディスク上に配置して下さい。サーバーのコンポーネントや データベースを共有ファイルシステムやネットワークファイルシステム上に配置することは出来ません。

注意

CD-ROM上に読み取り専用データベースを配置することは出来ませんが、FirebirdサーバーをCD-ROMから 実行することは出来ません。

スクリプトとプログラムのインストール

Firebirdをファイルのコピーによってインストールすることは可能です、例えば スナップショット・ビルドのファイルを「unrar」することや、WinZipで .zipファイルを解凍することですが、しかし、 初めてFirebirdをインストールする際には配布されているリリースキットを使用する ことを強くお勧めします。Windows用のインストーラー、Linux用のrpm (RPM Package Manager 元々は RedHat Package Manager) プログラム 、その他のPOSIXプラットフォーム用の公式 tar.gz ファイルはいくつかの必須のセットアップ処理を実行します。以下の手順に従って作業をすれば、 他にすることは特にありません。後は、ログインするだけです。

Windowsプラットフォーム

FirebirdのインストーラーではSuperサーバーとClassicサーバーのどちらを インストールするか選択しなくてはなりません。前述したように、両者の違いを 理解した上でClassicを選択するのでなければ、Superサーバーを選んで下さい。

Windows95/98/MEにFirebirdをインストールする場合には、コントロールパネル・アプレットの インストール・オプションのチェックを外して下さい。これらの プラットフォーム上では、コンパネ・アプレットは動作しません。本ガイド中で、この場合に 利用可能なアプレットについて後述します。

サーバー・プラットフォーム、Windows NT/2000/XP、においてはFirebirdサービスがインストール完了後 に実行されます。次の起動時には、自動的にサービスが開始されるようになっていきます。

Windowsの非サーバー・プラットフォーム、Windows95/98/ME、では、サービスがサポートされていません。そのため、インストーラーはFirebirdサーバーをアプリケーションモードで 実行します、その際サーバー・アプリケーションはガーディアンとして知られているアプリケーション によって保護されます。何らかの理由でサーバー・アプリケーションが異常終了した場合には、 ガーディアンがサーバー・アプリケーションを再起動します。

Posixプラットフォーム

全ての場合において、これからインストールしようと考えている当該のバージョンのFirebirdの リリースノートをお読み下さい。リリースを繰り返して多くのバージョンが存在しているかもしれ ません、特にオープンソース界ではこの傾向は顕著です。可能であれ

ば、各バージョンのFirebirdの開発者達は、知っておくべき事をドキュメントかしているはずです。

ティップ

もしもリリースノートがインストール・キット中に見あたらない場合には、Firebirdプロジェクトの webサイト<http://firebird.sourceforge.net>の[Downloads page](#)へ戻って、そこからリリースノートをダウンロードして下さい。

もしもrpmでのインストールをサポートしているLinuxディストリビューション を使用しているのであれば、RPM Package Managerを使用して各プラットフォーム に適切なドキュメントを入手して下さい。ほとんどのディストリビューションで、コマンドライン・シェルまたは GUIインターフェースによるインストールを選択出来ると思います。

rpmが動作しないLinuxディストリビューションや、各種にUNIXシステムでは .tar.gzパッケージを使用して下さい。詳細については リリースノートをご覧ください。

シェル・スクリプトが提供されていますが、希なケースでは、リリースノートにおいてスクリプトの修正を 何点か指示している場合があるかも知れません。

インストール後のテスト

手順通りに事が運べば、インストールの終了後にサーバー上でFirebirdサーバープロセスが実行されます。サーバーを再起動することなく、自動的にスタートするようになっています。

ここまでくれば、TCP/IPプロトコルを使用して（推奨）Firebirdのクライアント・サーバー・ネットワークを 利用できるようになりました。

注意

NetBEUIプロトコルを全てのWindows環境で利用するための情報については、Using Firebird manualの Chapter 6: Network Configurationをご覧ください。

警告

IPX/SPXネットワークはFirebirdではサポートされません。

サーバーへのPing

通常、インストールが完了した際には一番最初にPingをサーバーにうとうと思うものです。そうすることによって、クライアントマシンがサーバーへ接続することが本当に出来るかどうかを実際に確認出来るからです。例えば、クライアントからアクセス可能なサーバーのIPアドレスが 192.13.14.1だったとしましょう。コマンドシェルから 次のコマンドを実行して下さい

```
ping 192.13.14.1
```

実際のサーバーのIPアドレスとこの例のIPアドレスを入れ替えて下さい。

警告

タイムアウトが表示された場合、Using Firebird manual - Chapter 6: Network Configuration、Chapter 7: Troubleshooting Connectionsや、その他の事項を参照して下さい。

注意：ローカルマシンからサーバーへ接続しようとしている場合、もしサーバーと同一マシン上でクライアントが動作しているのであれば 仮想TCP/IPループバックサーバーを使ってPINGを打つことが出来ます

```
ping localhost -or- ping 127.0.0.1
```

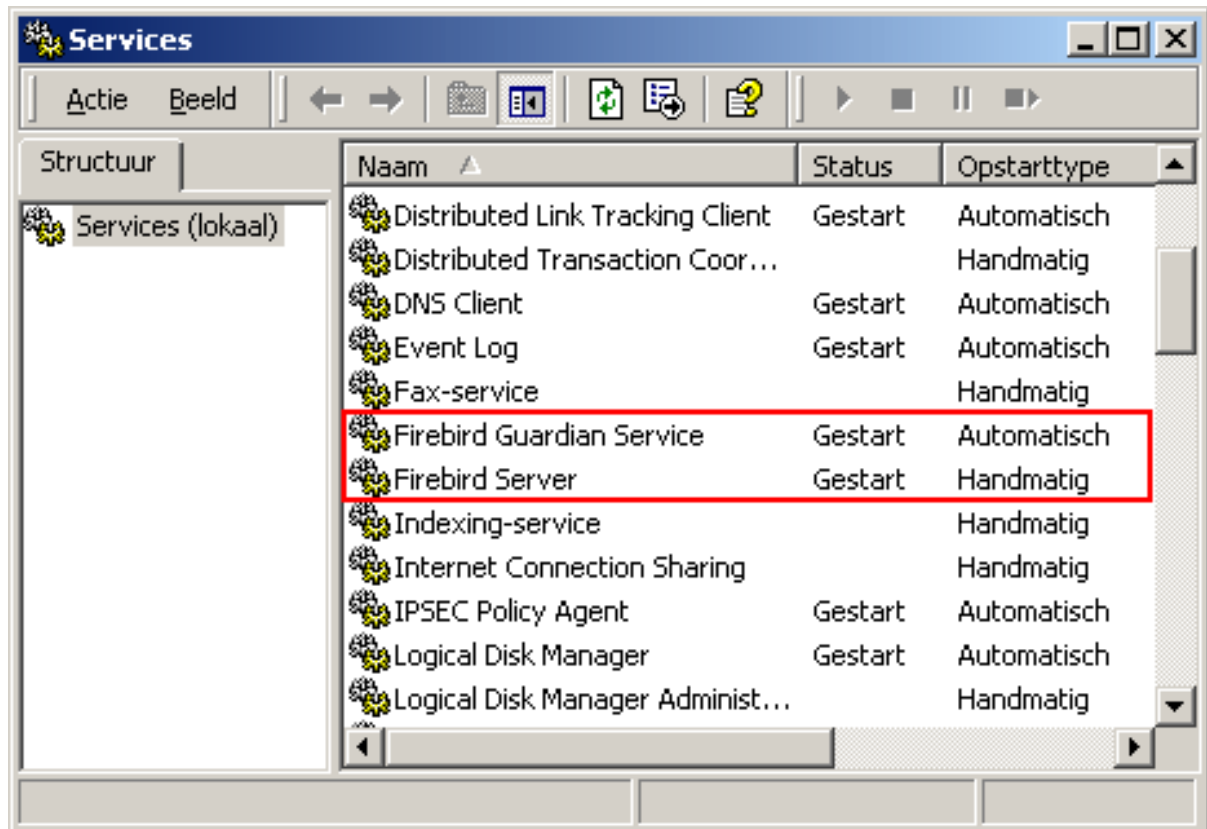
Firebirdサーバーが起動しているかどうかのチェック

Windows NT4、2000、XP上では、Firebirdサーバーはインストール後に、 サービスとして起動するようになっています

Windows NT4, 2000 and XP

次のコントロールパネルを開きます -> Services (NT) or Control Panel -> Administrative Tools -> Services (2000, XP).

この図ではWindows2000上のサービス・アプレットが表示されています。 Windowsサーバーや他のバージョンのものとは見え方が少し違うかもしれません



もし、ガーディアンが起動していたら（スクリーンショットの上部を見て下さい）、バー

ジョンの違いによって名前が違っているかも知れません。

注意

Windows2000、XP上では、ガーディアンは便利ではありますが必須ではありません。なぜならこの2つのOSはサービスを監視して必要な時に再起動する機能を持っているからです。他のプラットフォームではガーディアンを起動しておくことが推奨されます。何かの理由でサーバーが停止するようなことがあった場合に、SYSDBAが手動で再起動できないような状況では特にそうです。

Windows 9x or ME

Windows9xやMEでは、Firebirdサーバーはアプリケーションとして起動します。サーバープロセスはガーディアンによって監視されています。ガーディアンのアイコンがタスクトレイに緑色のアイコンで表示されているはずです。もし、このアイコンが点滅していたり赤色で表示されているとすれば、ガーディアンがサーバーを起動中であるか又は起動に失敗したことを示しています

インストールキットを利用してインストールを行った場合でも、自動的にガーディアンとFirebirdサーバーが起動しない時には、以下の手順に従って手動で設定をして下さい

1. ガーディアン・プログラム(fbguard.exe)の実行ファイルを配置して、ショートカットを作成します。それをスタートメニューのスタートアップフォルダに入れて下さい。
2. ショートカットのプロパティ・ダイアログを開いて、コマンドライン・フィールドへ移ります
3. コマンドラインを以下のように変更して下さい。

`fbguard.exe -a (for スーパーサーバー)`

`fbguard.exe -c (for Classicサーバ)`

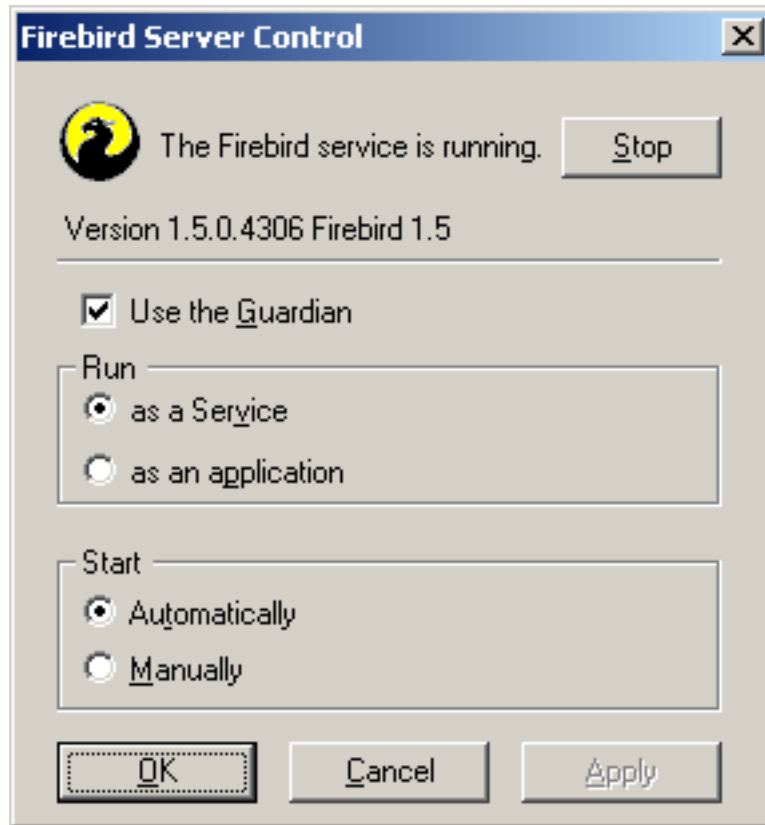
4. プロパティ・ダイアログを閉じて編集を保存します
5. ショートカットをダブルクリックしてガーディアンを開始します。ガーディアンはfbserver.exe又はfb_inet_server.exeを自動的に開始します

ガーディアンは次回Windows 9x/MEの起動時に、自動的に開始されます

別の方法として、コントロールパネルのアプレットを使用して、Firebirdサーバーを 起動/停止させることも出来ます

Windows コントロールパネル・アプレット

バージョン1.0.3から、Firebirdの配布パッケージにコントロールパネル・アプレットが含まれるようになっていました。アプレットは必須の構成要素ではありませんが、サーバーの起動/停止用途では便利に使えるので提供されています。



残念ながら、バンドルされているアプレットはWindows NT/2000/XPのみでしか動作しません。Windows 9x/MEをお使いの場合、便利なアプレット・プログラムが以下のwebサイトで配布されています

<http://www.achim-kalwa.de/fbcc.phtml>

Firebird Control Center [fbcc-0.2.6.exe](#)をダウンロードして下さい

このアプレットは上で示したスクリーンショットとは多少違っていますが、機能的には同等のものです

Posix サーバー

コマンドシェルで `top` コマンドを使うと、プロセスの起動状態をインタラクティブに確認することが出来ます。Firebirdのスーパーサーバーが起動中であれば、次のプロセスを確認することが出来るでしょう `fbguard`。これは ガーディアン・プロセスです。さらに、`fbserver`の 一つのメイン・プロセスとゼロ以上の子プロセスが起動しているでしょう

For Classic サーバー 版では、プロセス名は`fb_inet_server` となっています。ネットワークコネクション毎にこのプロセスが一つ起動しています。一つもアクティブなコネクションが無い場合、又はダイレクト・ローカル・コネクションしかない場合、`fb_inet_server` はプロセス・リスト上に現れません

次のスクリーンショットは、`top` コマンドの出力です。`grep` で、`fb`で始まるプロセスだけに限定してあります。

```
frodo:/inkomend/firebird # top -b -nl | grep fb
2587 firebird 24 0 1232 1232 1028 S 0.0 0.3 0:00.00 fbguard
2588 firebird 15 0 4124 4120 2092 S 0.0 0.9 0:00.04 fbserver
```

```
2589 firebird 15 0 4124 4120 2092 S 0.0 0.9 0:00.00 fbserver
2604 firebird 15 0 4124 4120 2092 S 0.0 0.9 0:00.00 fbserver
2605 firebird 15 0 4124 4120 2092 S 0.0 0.9 0:00.02 fbserver
2606 firebird 15 0 4124 4120 2092 S 0.0 0.9 0:00.00 fbserver
2607 firebird 15 0 4124 4120 2092 S 0.0 0.9 0:00.00 fbserver
```

topコマンドの代わりに、ps -axコマンドや ps -auxコマンドを利用することも出来ます。もちろん、grep をパイプで繋いでも結構です

その他の必要条件

サーバーのネットワーク・アドレス

- ・ 管理されているネットワークであれば、サーバー用のIPアドレスをシステム管理者から取得して下さい
- ・ クロスケーブルで接続した単純なネットワークに接続した2台のマシンであれば、どのIPアドレスを利用しても大丈夫です。(ただし127.0.0.1を除く。このIPアドレスはローカルループバック 用に予約されています) もちろん、クライアントマシンで利用しているIPアドレスも使えません。ネットワークカードの「native」なIPアドレスを知っている場合、それらが違うものであれば、そのまま使っても構いません
- ・ 一台のマシン上にクライアントとサーバーの両方をインストールしたのであれば、ローカルループバック・アドレスを使って下さい。localhost 又は 127.0.0.1のIPアドレスになります。

注意

Windows上のスーパーサーバーには、TCP/IPのループバックを使用せずに、ダイレクト接続することが可能です。これはTCP/IP接続ではなく、スレッド・セーフではありません。コマンドラインツール (gsec, gbak etc.) の単体での使用は問題なく動作します。対照的に、ダイレクトなデータベース接続は、Linux版のClassicサーバーでは、複数同時接続でも全く問題なく安全に動作します。

デフォルト・ユーザー名とパスワード

SYSDBAユーザーはサーバー上の全権限を保有しています。これはバージョン、OS、アーキテクチャー、インストール・プログラムによって、以下のいずれかの方法でインストールされます

- ・ SYSDBAユーザーがmasterkeyパスワード付でインストールされる (実際には、masterke: 8文字目より後の文字は無視されるので)
- ・ インストール中にパスワードの入力を促される
- ・ SYSDBA用にランダムなパスワードが生成され、/opt/firebird/SYSDBA.password に格納される

あなたのサーバーがインターネットに完全に曝されていて、パスワードが masterkeyのまま

だったら、即座にコマンドライン・ユーティリティのgsec を使ってパスワードを変更しなくてはなりません

SYSDBAユーザーのパスワード変更方法

Firebird には、gsecと呼ばれるコマンドライン・ツールが付属しています。 gsecを使ってユーザーアカウントの操作を行います

重要項目

幾つかのFirebirdのインストールパッケージでは、システム上の管理者権限 (root on Linux)がないとgsec を実行出来ない場合があります、あるいは、Firebirdサーバーのプロセスを実行しているユーザー 出ない場合も同様です。Windowsサーバープラットフォームでは、通常の場合Power Userグループかそれ以上の 権限でgsec を実行すれば大丈夫です

もし、十分な権限があるにもかかわらずgsec の実行に失敗し、「unavailable database - unable to open database」というメッセージが表示された場合

- ・ Windows版のClassicサーバーを実行しているか、-databaseの引数を 間違えている事が考えられます。あるいは、
- ・ サーバーが全く起動していないことも考えられます。そうであった場合、 [Testing your installation](#)へ戻って、問題を修正して下さい

SYSDBAのパスワードをicurry4meに変更することにしたとしましょう

1. サーバー上のコマンドシェルから、コマンドライン・ユーティリティのインストール位置へ ディレクトリを移動して下さい。 [Firebird installation components table](#)を参考 にしてその位置を探して下さい。
2. 次のコマンドを入力して下さい、Windows上のClassicサーバー を実行している場合は例外となります

```
gsec -user sysdba -password masterkey
```

Windows版のClassicサーバーの場合、必ずセキュリティ・データベース の位置を完全なネットワーク上の位置として指定して下さい

```
gsec -user sysdba -password masterkey -database "localhost:C:\Program Files\Firebird\Firebird_1_5\security.fdb" (パスは必要に応じて書き換えて下さい)
```

注意

- ・ Linux上では、gsecの代わりに./gsecとして下さい。 そうしないと「予期せぬ」gsec が起動されてしまう可能性 がある上、コマンドが見つからない可能性もあるからです
- ・ パスとファイル名は、Windows以外の全てのプラットフォームでは大文字小文字が区別されます。 パスワードは常に大文字小文字が区別されます

いずれのケースでも、gsecユーティリティのシェル・プロンプト が表示されているはず

GSEC>

3. 次のコマンドを実行して下さい

```
modify sysdba -pw icuryy4me
```

4. Enterキーを押下して下さい。新しいパスワード icuryy4me が暗号化されて保存されました。 masterkey はもう使用できなくなっています
5. gsec のシェルを終了します

```
quit
```

注意

Firebirdはパスワードの8文字を越えた部分を無視するので、icuryy4mも icuryy4monkeys もパスワードとして通用します

管理ツール

Firebirdインストールパッケージには、GUIの管理ツールは含まれていません。コマンドライン・ツールは 付属していますが、それらはFirebirdのインストール・ディレクトリのbinサブディレクトリに配置されています

Windowsのクライアントマシンで利用可能な、様々な素晴らしいGUIツールは、たくさんありすぎてここでご紹介することは出来ません。幾つかのGUIツールはBorlandの Kylixで書かれていて、Linuxクライアントマシンで利用可能ですが、完成度はまちまちです

以下のリンクを辿って下さい <http://www.ibphoenix.com>上の[Downloads > Contributed > Admin Tools page](#)。これらは全てオプションとなっています

注意

WindowsクライアントからLinuxサーバーに接続することも出来るし、その逆もまた可能です

クライアントのみのインストール

リモートクライアントマシンはそれぞれクライアント・ライブラリを必要としています。Posixクライアントでは libfbclient.so 、Windowsクライアント では fbclient.dll となります。それぞれのクライアント・ライブラリ はFirebirdサーバーのバージョンと一致するようにします

Firebirdのバージョン1.5以降では、バージョン1.0のライブラリ(「古い」 InterBaseからの名称で) と同名で、クライアント・ライブラリの コピー又はシンボリックリンクを古いクライアント・ライブラリに依存したサードパーティの製品との 交換性を保つためにインストールするようになっています

その他、幾つかのファイルがクライアントのみのインストールで必要となっています

Windows

現状では、Windowsクライアントへのクライアント・ライブラリのみをインストールする コンパクト・インストーラーは用意されていません。もし、Linuxやその他のPosixサーバー（あるいは 他のWindowsサーバー）に接続するWindowsクライアントを用意することがあれば、サーバーマシンの Firebirdサーバーのバージョンに対応したWindows用のフル・インストーラーをダウンロードしなくては なりません。

幸いにも、いったんインストーラーを入手すれば、Windowsのクライアント専用機へのインストール は大変簡単です。インストーラーを起動して、サーバーをインストールしたのと同じようにして、CLIENT ONLY オプションをメニューから選択すればOKです。

Linux とその他の Posixクライアント

同様に、Linuxクライアント用の小さな専用インストーラーは用意されていません。加えて、Posixの流儀のために、Linuxであったとしても、幾つかの固有の条件がファイルシステム上にあるため、全てのディストリビューション(*x)に対してFirebirdはクライアントのみのインストールオプションを提供 していません。

ほとんどのLinuxの流儀のために、以下のプロシージャがFirebirdのクライアントのみのインストールのために提供されています。rootとして ログインして実行して下さい。

1. Look for libfbclient.so.1.m.n (m.n being the minor plus patch version number) in /opt/firebird/lib on the server where Firebird server is installed. Copy it to /usr/lib on the client.
2. Create chained symlinks using the following commands:

```
ln -s /usr/lib/libfbclient.so.1.m.n /usr/lib/libfbclient.so.1
```

```
ln -s /usr/lib/libfbclient.so.1 /usr/lib/libfbclient.so
```

replacing 1.m.n with your version number, e.g. 1.5.0 or 1.6.1

If you're running applications that expect the legacy libraries to be present, also create the following symlinks:

```
ln -s /usr/lib/libfbclient.so /usr/lib/libgds.so.0
```

```
ln -s /usr/lib/libfbclient.so /usr/lib/libgds.so
```

3. Copy the firebird.msg file to /opt/firebird
4. In the system-wide default shell profile, or using setenv() from a shell, create the FIREBIRD environment variable and point it to /opt/firebird, to enable the API routines to locate the messages.

データベース上での作業

サンプル・データベースへの接続

Firebirdのインストールディレクトリ中のexamples に、employee.fdbというサンプル・データベースがあります。このデータベースを使って「腕試し」をしてみてください。

サーバー名とパス名

サンプル・データベースを移動する場合には、サーバーマシンに物理的に接続されたハードディスク に配置されていることを確認して下さい。共有ディスクやネットワークドライブ、(Unixでは) SMB(samba) ファイルシステムでマウントされている場合動作しません。同じ事は、これからあなたが作成するデータベース についても言えます。

TCP/IP接続に関して、2つの要素があります：サーバー名とディスク/ファイルシステムのパス名です これらは以下のように記述して下さい：

- ・ Linux サーバー：

サーバー名:/ファイルシステムのパス名/データベースファイル名

Linux とその他の Posix サーバーでの例：サーバー名serverxyz：

serverxyz:/opt/interbase/examples/employee.fdb

- ・ Windows サーバー：

サーバー名：ドライブレター：¥ファイルシステムのパス名¥データベースファイル名

Windows での例：

serverxyz:C:¥Program Files¥Firebird¥examples¥employee.fdb

CONNECT ステートメント

Firebirdのデータベースに接続するためには、ユーザー名と正しいパスワードの組み合わせによる ユーザー 認証が必要となります。SYSDBA、 root(Posixシステム上)又は Administrator (Windowsシステム上で、Firebirdがこのユーザー権限で動作している場合) 以外のユーザーは、データベース内の 各オブジェクトに対するパーミッションも持っていないといけません。問題を簡単にするために、ここでは SYSDBAユーザーをパスワード masterkeyで認証している ということにしましょう。

isqlの使い方

isqlにはデータベースへの接続方法がいくつかあります。一つは、 isqlをシェルで起動する方法です。まずFirebirdのインストール先ディレクトリ のbinサブディレクトリへ移動して、プロンプトからisql コマンドをタイプして下さい (Linuxでは、./isql) [↵は「 エンターキーを押下してください」]：

```
C:¥Program Files¥Firebird¥Firebird_1_5¥bin>isql↵
Use CONNECT or CREATE DATABASE to specify a database
SQL>CONNECT "C:¥Program Files¥Firebird¥Firebird_1_5¥examples¥employee.fdb"↵
```

```
CON>user 'SYSDBA' password 'masterkey';↵
```

重要項目

- ・ isqlでは、全てのSQL文はセミコロンで区切ります。Enterキーを押下しても、セミコロンがついていないと、isqlは文が 次の行に続いていると判断して、プロンプトをSQL>から CON>に変更して表示します。この機能によって、長い文を複数の行に またがって記述することが出来るようになっていきます。Enterを押下した 時に、もしセミコロンを忘れていたら、CON>プロンプトの直後に セミコロンを打って、再度Enterキーを押下して下さい。
- ・ もしLinux上でClassicサーバーを実行している場合、データベースのパスがホスト名で開始されていない場合、高速なダイレクト・ローカル接続が試みられます。この接続法は、 ユーザーのLinuxへのログインアカウントにデータベース・ファイルへの十分なアクセス権限が ない場合失敗します。この場合、localhost: /<path>で接続して下さい。サーバー・プロセス がファイルをオープンします (Firebird 1.5 は通常firebirdユーザーとして実行されています)。逆に、ネットワーク・スタイルの接続では、 ユーザーがClassicサーバーのローカル・モードでデータベースを作成したり、サーバーにファイル への十分なアクセス権限がない場合には接続に失敗することになります。
- ・ もし、ClassicサーバーをWindows上で実行している場合、必ず ホスト名を指定しなくてはなりません(localhost も含まれます)。データベースへのフルパスにホスト名を付加しない場合、接続は失敗します。

注意

一重引用符はFirebirdの「標準」区切り文字ですが、二重引用符 が上記の例で、データベースのパスを指定する所で使用されています。このように、 コマンドライン・ユーティリティの一部で空白を含むパスを指定する際に必要と なる場合があります。一重引用符も空白を含まないパスに対してはきちんと動作 します。

「SYSDBA」と「masterkey」の両側の引用符はオプション となります。空白を含まないデータベースのパスも引用符は必要ではありません。

この点は、isqlが接続した時点で表示します。

```
DATABASE "C:\Program Files\Firebird\Firebird_1_5\examples\employee.fdb",
User: sysdba
SQL>
```

employee.fdbを利用して遊んでもらってもいいでしょう。 isqlの由来はinteractive SQLから来ています。isqlを使って、データを検索したり、メタデータの情報を表示したり、データベース・オブジェクトを作成したり、データ定義言語を実行するなど、様々なことが出来ます

コマンドプロンプトに戻るには、次のようにタイプして下さい

```
SQL>QUIT;↵
```

isqlについての詳細な情報は、Using FirebirdのChapter 10:Interactive SQL Utility (isql) をご覧下さい。

GUIクライアントの利用方法

GUIクライアントは通常、いったん入力されたCONNECT 文字列の生成や、サーバー、パス、ユーザー名、パスワードの情報等の構成を管理しています。各要素には前節でご説明した内容を使用して下さい。

注意

- ・ これらのツールでは、たいていの場合、サーバー名 + パス名 を一つながりの文字列として扱っています。
- ・ Linux/Posixシステムではファイル名やコマンド名は大文字小文字を区別 するということに注意して下さい。

isqlを使用したデータベースの作成

isqlを使用したデータベースの作成方法は一つだけ ではありません。ここでは、データベースをインタラクティブに作成する単純な方法を見 ていく こととします、実際のデータベース作成業務においてはこうした方法ではなく、 データ定義言語 (DDL) を利用してデータベースの作成とメンテナンスを行うべきです。 Using Firebirdマニュアルの該当部分をご覧ください。

isqlの開始

isqlを利用してデータベースをインタラクティブに作成 するためには、Firebirdのインストールディレクトリ下の binサブディレクトリで、isql (Windows) 又は ./isql (Linux) を実行して下さい。

```
C:\Program Files\Firebird\Firebird_1_5\bin>isql↵
Use CONNECT or CREATE DATABASE to specify a database
```

CREATE DATABASE 文

さて、新しいデータベースをインタラクティブに作成する準備が出来ました。 それでは、test.fdbという名前のデータベースを作成してみま しょう。作成場所はdataディレクトリを Dドライブに作成して、そこへ配置 することにします。

```
SQL>CREATE DATABASE 'D:\data\test.fdb' page_size 8192↵
CON>user 'SYSDBA' password 'masterkey';↵
```

重要項目

- ・ CREATE DATABASE文中で、パス名・ユーザー名・パスワードには引用符がありますが、これは必須です。この点では、CONNECT文とは違いがあります。
- ・ もし、Linux上でClassicサーバーを実行している場合、データベースのパス名をホスト名から始めてはいけません。その場合、データベースファイルはあなたのLinuxログインユーザーをオーナーとして作成されることになります。それでいいかどうかはあなた次第ですが(アクセス権限を考慮して下さい、他のユーザーがアクセス出来るかどうかについてです)。localhost:をパスに付けた場合、サーバープロセス(Firebird 1.5では通常firebirdが使われています)がファイルを生成して所有します。
- ・ Windows上でClassicサーバーを実行している場合、ホスト名は必ず特定しなくてはなりません(localhostかもしれませんが)。パス名にホスト名が付加されていないとデータベースの作成は失敗します。

さあ、データベースが作成されました。ちょっとだけ待って、SQLプロンプトが現れましたね。この状態で、新しいデータベースに接続していますので、新しいテストオブジェクトを作成することが出来るようになっています。

データベースが本当に作成されているのかを確認する為に、次のようにタイプして下さい

```
SQL>SELECT * FROM RDB$RELATIONS;↵
```

スクリーンに沢山のデータが表示されましたね! このクエリーはFirebirdがメタデータを格納しているシステムテーブル中の全ての行を表示させるものです。「空の」データベースは全くの空というわけではありません、データベースにはあなたが作成するオブジェクトを格納するためのメタデータが既に存在しているのです。

コマンドプロンプトに戻るには次の様にタイプして下さい

```
SQL>QUIT;↵
```

isqlに関する詳細は、Using Firebird の Chapter 10:Interactive SQL Utility (isql)を参照して下さい。

Firebird SQL

データベースマネジメントシステムには通常、それぞれ固有の方法でSQLが実装されています。Firebirdは従兄弟のInterBase®を除く他のRDBMSと比較して、より厳格にSQL標準に準拠しています。そのため、SQL標準に準拠していない他製品から移行してきた開発者は、Firebirdはクセが強いとよく言いますが、クセと言われているものの大半は実は全くクセではなく標準なのです。

整数と整数の除算

FirebirdはSQL標準に準拠しているため、整数÷整数の計算の結果(商)は切り捨てられ、次に小さい整数となります。このことは、ヘンテコな結果になるので十分注意して下さい。

例えば次の計算はSQLにおいては正解です:

1 / 3 = 0

RDBMSからの、整数÷整数の結果を浮動小数点型に改善したい場合、その除数と被除数の 両方又はどちらかについてそのデータ型について、浮動小数点型か高精度数値型を使用する必要があります。

例えば、上の計算式で非ゼロ値を結果として得たい場合には次のような 変更をします：

1.000 / 3 = 0.333

文字列に関する情報

文字列の区切記号

Firebirdで文字列を扱う場合には一重引用符(アポストロフィ) が必要です - 'I am a string' - (ASCIIコード39で、96ではない)。InterBase® に対して初期のバージョンの Firebirdを使う場合、文字列の区切記号として一重引用符と二重引用符 は交換可能であったことを想起できるでしょう。二重引用符はFirebirdのSQL文において文字列の 区切記号としては使うことが出来ません。

文字列中のアポストロフィ

Firebirdの文字列中でアポストロフィを使用したい場合には、 アポストロフィをもう一つ前に付けて「エスケープ」して下さい。

例えば、次の文字列はエラーになります：

'Joe's Emporium'

パーサーはアポストロフィに出会ったところで、'Joe' と解釈して、その後は不明なキーワードとなってしまいます。

規約に適合する文字列にするためには、アポストロフィを二重にします：

'Joe''s Emporium'

これは二つの一重引用符で二重引用符ではないことに注意して下さい。

C文字列の結合

SQLにおける結合記号は二つの「パイプ」記号(ASCIIコード 124、スペースを入れず 二つ並べる)です。SQL中で、「+」記号は算術演算子であって、文字列の結合に 使用した場合にはエラーとなります。次の表現では文字型のカラムに「Reported by: 」を前に付加しています。

'Reported by: ' || LastName

結合は慎重に行って下さい。もしも、2つ又は3つのCHAR型又はVARCHAR型のカラムを結合した 場合に、CHAR型又はVARCHAR型の最大長である32KBの制限を超えてしまうとエラーになるので、 注意して下さい。

以下の点も参照して下さい、 [Expressions involving NULL-ja](#) NULLを含んだ結合表現について
concatenating in expressions involving

二重引用符識別子

SQL-92標準以前には、データベース中で大文字小文字の別や空白を含んだキーワード が二重化する場合のオブジェクト名(識別子)に関しての規約はありませんでした。SQL-92 が一つの新しい規格を取り入れてその規約にしたため、識別子は二つの二重引用符(ASCIIコード 34)で囲まれるものと定義し、以降は常に二重引用符で囲まれる事になりました。

この「贈り物」の目的というのは、非標準のRDBMSから標準準拠のRDBMS への以降を容易にするためのものでした。この事で悪い面があるとすれば、二重引用符 で識別子を定義した場合、それらは大文字小文字が区別され、その後も二重引用符を 使用することが強制されると言うことです。

Firebirdは大変制限された環境下でも、いくらかは気を抜けるようになっています。 識別子が二重引用符で定義されていたとします:

1. その場合大文字で定義されたものと見なします
2. キーワードではなく
3. スペースを含みません

また、引用符を使わずに大文字小文字を区別しないという使い方も出来ます (しかしすぐにも二重引用符を付けた方が良いでしょう、大文字小文字を合わせる べきです)。

警告

あまりスマートではありませんが、例として、以下のテーブルがあるとしましょう
"TESTTABLE" and "TestTable"、 両方が二重引用符を付けて定義されていて、次のコマンドを実行します:

```
SQL>select * from TestTable;
```

あなたがレコードを得るのは、"TESTTABLE"からで "TestTable"ではありません!

引用符による識別子がどうしても必要な理由がある場合でなければ、通常は使用を 避けた方が良いでしょう。Firebirdは幸運なことに引用符付と引用符無しの識別子の 混在を認めています - 必要であれば、古いデータベースから引き継いだキーワードを含め ることにはなんの問題もありません。

警告

データベース管理ツールの中には、デフォルトで全ての 識別子に二重引用符を強制するものがあります。二重引用符オプションを設定可能か 試してみてください。

NULLを含んだ表現式

SQLでは、NULLは値ではありません。NULLはデータの条件又は 状態であって、値としては不明なものを意味します。NULL を含んだ演算や、NULLと値を含んだ他の表現式 は、ほとんどの場合常にNULLを結果として返します。NULL はゼロや空白や「空文字」ではありませんし

、それらの様に振る舞う こともありません。

試してみた場合にびっくりするかも知れませんが、NULLによる 比較や演算を実行してみた場合の例をいくつか示しましょう：

- ・ `1 + 2 + 3 + NULL = NULL`
- ・ `not (NULL) = NULL`
- ・ `'Home' || 'sweet' || NULL = NULL`
- ・

```
if (a = b) then
  MyVariable = 'Equal';
else
  MyVariable = 'Not equal';
```

これらのコードを実行してみた後で、aとb の両方をNULLにした場合、MyVariableには 'Not equal' が入ります。理由は、'a = b' の結果は どちらか一方がNULLであった場合、NULLになる からです。テスト例にある「if」文はNULL の場合にはfalseとして振る舞うため、'then' ブロックはスキップされて、'else' ブロックが実行されます

警告

表現式がfalseとして振る舞うにもかかわらず、 この場合には、まだこれはNULLのままです。つまり、not() を使って結果を反転させようと思っても、結局これもNULL を返すことになり、trueは返ってきません。

- ・

```
if (a <> b) then
  MyVariable = 'Not equal';
else
  MyVariable = 'Equal';
```

ここでMyVariable は'Equal'になりますが、 それはaがNULLでbが NULL出はない時、あるいはその逆の時です。説明としては 左記の例と同様になります。

- ・ `FirstName || ' ' || LastName`

この式はFirstNameとLastNameのどちらかがNULLの時にはNULLを返します。

ティップ

NULLをUNKNOWNとして考えてみれば 上記のヘンテコな結果も、なるほどと頷けるでしょう。もしNumber が不明だったとしたら、'1 + 2 + 3 + Number' もまた、不明（つまりNULL）になることは自明です。MyStringが 不明であれば、'MyString || YourString' も不明となります(YourStringが非NULLであっても)。その他も同様ですね。

NULLに関する追加情報

NULLに関する詳細な情報は、以下から入手出来る Firebird Null Guideをご覧ください：

- ・ <http://www.firebirdsql.org/manual/nullguide.html> (HTML)
- ・ <http://www.firebirdsql.org/pdfmanual/Firebird-Null-Guide.pdf> (PDF)

セキュリティ・ガイドライン

サーバーのセキュリティオプション

Firebird 1.5は新しいセキュリティに関する既定をいくつか持っています。これらを探索してみてください。多くの設定可能な機能のデフォルト値は古いままで、「セキュアで無い」振る舞いをしますが、これは既存のアプリケーションの動作を保証するためです、しかし可能な限り保護レベルを向上させようとすれば、システムのセキュリティをさらに高めることが出来るようになっていきます。

新規又は変更されたセキュリティ機能は以下の通りです：

- ・ Posixシステムでは、Firebirdはデフォルトでfirebird として実行されるようになります。（以前は rootでした）。
- ・ Windowsサーバープラットフォームでは、Firebirdサービスを指定されたユーザーアカウント（例：Firebird）で実行することも可能です。サービスを実行する既定のユーザーはLocalSystemですが、インターネット接続環境下ではシステムが危険にさらされる可能性があります。docにあるREADME.instsvcを読んで、この設定について学んで下さい。
- ・ Database Aliases は、クライアントから物理的なデータベースの位置を隠蔽します。エイリアスを利用することで、クライアントは例えば次のようにしてデータベースに接続します「frodo:zappa」、つまりfrodo:/var/firebird/music/underground/mothers_of_invention.fdbという物理配置を公開せずに済みます。エイリアスを利用することで、また、データベースの配置を変更しても、クライアントの接続文字列を変更しなくても済むようになります。
- ・ DatabaseAccess パラメータによって、Restrictの場合、制限されたアクセスを設定することが可能です。この場合、明示的に記述されたファイルシステム上の位置に限定することが出来ます。また、Noneを設定すると、エイリアスで指定されたデータベースのみアクセス可能となります。既定値はFullとなっていて、無制限のアクセスを認めています。（訳注：原文ではAllだが、これはFullのはずです）
- ・ ExternalFileAccess パラメータは、外部テーブルへのアクセスを制御しています
- ・ UdfAccess パラメータは、ユーザー定義関数の配置場所についての制御をしています。

データベース・エイリアスはaliases.confで指定されています。また、設定パラメータはfirebird.confで指定されています。インストールした Firebirdのリリースノートをよく読んで、適切な使用方法を理解するようにして下さい（また、幾ばくか 有用なアドバイスが記述されているはずです）。

Backup

Firebirdには、データベースをバックアップ・リストアするためのツールが付属しています。それはgbakという名前で、Firebirdのインストール先ディレクトリの binサブディレクトリに配置されています。Firebirdの データベースはユーザーが接続し通常の動作を行っ

ている最中でもバックアップすることが可能です。バックアップはバックアップ開始時点でのデータベースの状態のスナップショットとして作成されます。

定期的なバックアップと時々リストアを行うために、gbakを使用することは、データベース管理業務として定期的に行うべきです。

警告

データベースが動作中に、外部のプロプライエタリなユーティリティやファイルコピーツール類 (WinZip, tar, copy, xcopy, 等々) を使用しないで下さい。バックアップが正確に行われないばかりでなく、これらのツールの使用によって引き起こされるディスクレベルでの排他処理が動作中のデータベースを壊してしまう可能性があります。

重要項目

リストア中のデータベースの振る舞いについて、次のセクションの警告から学んで下さい

データベースの壊し方（やってはいけないこと）

以下のリストに挙げたのはやってはいけないことです。こういったことを行くとFirebirdのデータベースは壊れてしまう可能性があります。

メタデータテーブルを変更する

Firebirdは自身のオブジェクトとユーザーが作成した全てのオブジェクトのメタデータをデータベース中に格納して管理しています。もっとはっきり言えば、それらのオブジェクトはデータベース自身の関連テーブルに正しく格納されています。システムテーブルとそれらのカラムやその他の幾つかのシステムオブジェクトの識別子は RDB\$で開始されています。

これらは通常のデータベースオブジェクトであるため、ユーザー定義のオブジェクトと同様にクエリーをかけたたり操作したりすることが出来ます。しかしながら、これらの操作をすべきではありません。Firebirdのデータベースエンジンはメタデータオブジェクトを定義し操作するために、高いレベルでSQL (DDL) のサブセットを実装しています。それらは、CREATEやALTER、またDROP 文として実行することが可能です。

システムテーブルの直接的な操作をSQLでやることはもちろん、メタデータを変更したり削除したりする時に、DDLを使うことはあまり強くお勧め出来ません。SQLの技術とFirebirdエンジンに関する知識が十分なレベルに到達するまで、その「衝動」を先延ばしした方がよいでしょう。大破したデータベースはちょっとした注意や簡単な修理で何とかなるものではないからです。

Windowsサーバー上でForeced WritesをOFFにする

Firebirdはデフォルトでforced writes (同期書込モード) をONにしてインストールされています。データの変更や新規の挿入はディスクへ即座に書き込まれています。

データベースを非同期書込モードに変更することは可能ですが、それによってデータの変更

や 新規のデータはメモリキャッシュ中に保持され、オペレーティングシステムのI/Oサブシステムが定期的なその内容をディスクへ書き込むことになります。こうした変更は、forced writes off (又はdisabled)の設定をすることで可能です。この設定は 大規模なバッチオペレーション等でパフォーマンスの向上を図るためになされることがあります。

しかし重大な警告をしなくてはなりません。Windowsサーバー上では、forced writesをdisableに絶対にしてはいけません。Windowsサーバーでは、Firebirdサービスをシャットダウンするまで書きキャッシュがフラッシュされなかったという事が報告されています。電源以上による中断は別として、Windowsサーバーの行いはかなり酷いものです。もし、OSがハングして I/Oシステムにまでそれが及んだ場合、ユーザーの行った操作はリブートプロセスによって全て失われて しまいます。

注意

Windows 9x / ME は遅延書込をサポートしていません。

Linuxサーバー上でForced WritesをOFFにする

Linuxサーバーは一時的にforced writesをOFFにして運用しても安全です。とはいえ、大規模バッチジョブを終了したら元に戻すことを忘れないことです。もしも 大変安全な予備電源システムを使用しているなら別ですが。

運用中のデータベースにバックアップからリストアする

gbakのリストアのオプションの一つに(gbak -r[eplace]) があり、gbakが既存のデータベースを上書きすることを認めることが出来ます。この時、ユーザーがデータベースに ログインしていても警告すること成しにリストアが実行されてしまいます。結果として、データベースの破壊が たいていの場合は確実に起きてしまいます。

警告

管理ツールや管理用のプロシージャをデザインする際には、どのユーザー(SYSDBAも 含みます)であっても、いずれかのユーザーがログインしているデータベースに対してリストアを行わないよう に注意して下さい。

注意

gbakの使い方については、Using Firebirdの Chapter21 Database Backup and Restoreを参照して下さい。

ユーザーに対して排他的アクセスを行う方法については、Using Firebirdの Chapter14 Getting exclusive access to a databaseを参照して下さい。

實際上どうすればよいのかということでは、スペアのディスクスペースにgbak -c[reate] オプションを使用してリストアを行い、isql等のツールを使用して十分にテストを行うことを推奨します。リストアされたデータベースが問題なければ、サーバーをシャットダウンして、古いデータベース のコピーをファイルシステム上で行い、リストアしたデータベースで既存のデータベースを上書きして下さい。

リストア中にユーザーログインを認めてしまう

gbak -r[eplace]によってリストア中のデータベースにユーザーの ログインを認めてしまった場合、ユーザーのデータ操作によって、データベースが壊れてしまうでしょう。

Helpの探し方

Firebirdの廻りには多くの自発的なヘルパーのコミュニティが長きにわたって形作られてきました、InterBase® 6のソースコードが公開され、Firebirdプロジェクトが始まるもっと前からそれは連綿と続いて きました。見渡してみれば、Firebirdのコミュニティはあらゆる答えをすでに持っているのです。ボストンの バスルームで画板の上にそのデザインがなされる前から、Firebirdに関わってきた何名かの人々まで含まれて いるのです。(訳注：AnnやJimのことですね)

- ・ Firebird Projectの公式サイトは <http://firebird.sourceforge.net> こちらです。そして、サポートリストに参加して下さい。
- ・ Firebird knowledge site はこちらです。 <http://www.ibphoenix.com> Firebirdの開発と利用上の情報については、大変な量の情報を見いだすことが出来るでしょう。IBPhoenixはまたDeveloper CDを販売していますが、これにはFirebirdのバイナリとたくさん のドキュメント が含まれています。(訳注：Firebird日本ユーザー会でも取り扱っています)
- ・ Firebirdプロジェクト自体がプロデュースしている成長するドキュメントはこちらをご覧ください。 <http://firebird.sourceforge.net/manual/>.
- ・ 公式 Firebird Book は こちら から 発 注 で き ます http://www.ibphoenix.com/main.nfs?a=ibphoenix&s=1093098777:149734&page=ibp_firebird_book, 1100ページを越えるFirebird情報が満載です。(訳注：Amazonでも注文できます)

注意

IBPhoenix の 出版 する Using Firebird と The Firebird Reference Guide, は、Developer CDにまだ入っていますが、あまりメンテナンスされていません。しかし、これらに含まれていたほとんどの内容はアップデートされて公式ドキュメントに加えられて来ています。

Firebirdプロジェクト

Firebirdと幾つかのドライバの開発者、設計者、テスターはSourceForge上のFirebirdプロジェクトの メンバーとなっています。SourceForgeは幾千ものオープンソースプロジェクトのホームグラウンドで 驚異的な仮想コミュニティとなっています。Firebirdプロジェクトのアドレスは <http://sourceforge.net/projects/firebird>です。このサイトには、ソースコードツリーとバグとラッカー、その他いくつもの技術的ファイルがあり、コードベースの開発とテストのための種々の目的のためにダウンロードすることが出来ます。

Firebirdプロジェクトの開発者とテスターは以下のメーリングリストを利用しています。
firebird-devel@lists.sourceforge.net これは、「仮想研究室」となっていて、Firebird
の強化やバグフィックス、新しいバージョンのリリース等のためのコミュニケーションの
手段となっています。

このMLには興味を持っている人であれば誰でも参加ですが、しかし、ユーザーサポー
ト的な質問は妨げになるので歓迎されません。そういった質問は、 firebird-
support@yahoogroups.com へお願い致します。

Document History

The exact file history is recorded in the manual module in our CVS tree; see http://sourceforge.net/cvs/?group_id=9028

改訂履歴

0.0	2002	IBP	Published as Chapter One of Using Firebird.
1.0	2003	IBP	Published separately as a free Quick Start Guide.
1. x	June 2004	IBP	Donated to Firebird Project by IBPhoenix.
2.0	27 Aug 2004	PV	Upgraded to Firebird 1.5 Added Classic vs. Superserver section. Reorganised and corrected Disk Locations Table. Added (new) screenshots. Added section on security. Updated and completed information on Control Panel applets. Added more examples to 「Expressions involving NULL」 . Various other corrections and additions.
2.1	20 Feb 2005	PV	Enhanced GSEC section. Added more info to CONNECT and CREATE DATABASE sections. Added version number and document history.
2.1.1	1 March 2005	PV	Changed gbak r[estore] to r[eplace] in two places.
2.1.2	8 April 2005	PV	Reordered Firebird SQL subsections. Added links to Firebird Null Guide.
2.2	2 Dec 05	PV	Removed "Using the books by IBPhoenix" as it doesn't make sense in the QSG. Promoted "How to get help" to 1st-level section and removed "Where to next" shell. Removed link to UFB and RefGuide; added a note instead explaining their current status. Updated/corrected classic-super comparison table. Moved a number of sections on installing, working with databases, and (un)safety into newly created top-level sections.
–	Between revisions	PV	Corrected statement on SS thread usage in Classic-vs-Superserver table.

改訂履歴

License notice

The contents of this Documentation are subject to the Public Documentation License Version 1.0 (the 「License」); you may only use this Documentation if you comply with the terms of this License. Copies of the License are available at <http://www.firebirdsql.org/pdfmanual/pdl.pdf> (PDF) and <http://www.firebirdsql.org/manual/pdl.html> (HTML).

The Original Documentation is titled Firebird Quick Start Guide.

The Initial Writer of the Original Documentation is: IBPhoenix Editors.

Copyright (C) 2002-2004. All Rights Reserved. Initial Writer contact: hborrie at ibphoenix dot com.

Contributor: Paul Vinkenoog - see [document history](#).

Portions created by Paul Vinkenoog are Copyright (C) 2004-2005. All Rights Reserved. Contributor contact: paul at vinkenoog dot nl.

Alphabetical index

シンボル

インストール, 7
 Classicサーバー と Superサーバー, 3
 クライアントのみ, 15
 サーバー, 7
 スクリプトとプログラム, 8
 ドライバ, 7
 バイナリキットの内容, 3
インストール先, 5
ガーディアン, 8, 10, 12
コントロールパネル・アプレット, 11
サンプル・データベース, 16, 16
サーバーのチェック, 10
サーバー名とパス名, 17
サービス (Windows), 10
セキュリティ, 24
テスト, 9
データベース
 isqlでの作成, 19
 サンプル・データベース, 16
 セキュリティ, 24
 接続, 16
 GUIクライアントによる, 18
 isql, 17
ネットワーク・アドレス, 13
パスワード
 変更, 14
 既定の, 13
ユーザー名
 既定の, 13
二重引用符識別子, 22
接続, 16
整数の除算, 20
文字列
 区切記号, 21
 文字列中のアポストロフィ, 21
 結合, 21
文字列中のアポストロフィ, 21
管理ツール, 15

A

Aliases, 24

B

Backup, 24
Books

The Firebird Book, 27

C

Classicサーバー, 3
CONNECT ステートメント, 17
CREATE DATABASE 文, 19

D

Database
 Aliases, 24
Databases
 metadata, 25
 system tables, 25
 バックアップとリストア, 24, 26, 26
 破壊, 25
Document history, 29
Documentation, 27

F

Firebird Book, 27
Firebird project, 27
Firebird SQL, 20
Forced writes, 25

G

gsec, 14
Guardian, 11

H

Help, 27

I

isql
 データベースの作成, 19
 データベースへの接続, 17

L

License notice, 31

N

NULL, 22

P

Ping, 9
Project, 27

R

Restore, 24
 to a running database, 26
 user logins during restore, 26

S

SQL, 20

 CONNECT ステートメント, 17

 CREATE DATABASE 文, 19

Superサーバー, 3

SYSDBA, 13

System tables, 25

T

top コマンド (Linux), 12