

PostgreSQL for Windows の概要

日本PostgreSQLユーザ会

さいとう ひろし

--- Agenda ---

- 1.Windows対応の歴史(Challenge for windows)
- 2.PostgreSQL 8.0 for Windows の特化した機能
- 3.PostgreSQL 8.0 for Windows の性能
- 4.コントロールセンターとしてのpgAdminIII
- 5.pginstaller
- 6.おもしろい話題



1.Windows対応の歴史(Challenge for windows)

- Myron Scott氏によるSolaris-Pthread対応(Version 7.0)
- Single-User限定のThread-Modelの実験(Version 7.1)
- Jan Wieck氏のProcess-Modelの実現(Version 7.2)
- PowerGresでのThread-Modelの実現(Version 7.3)
- 本家のPostgreSQL8.0リリース



▪ Myron Scott氏によるSolaris-Pthread対応(Version 7.0)

-Thread化によって排他的な領域操作を自分で行わなければならないため、
アクセスされるスレッド広域変数のすべてをスレッド個別に割当てて取り込みました。

```
Env* GetEnv(void) {  
    Env* env;  
    thr_getspecific(*envkey,(void*)&env);  
    return env;  
}
```

-Threadごとにメモリコンテキストの取得

```
context = (PortalHeapMemory) GetEnv()->CurrentMemoryContext;
```

※どの程度の効果があったのか?

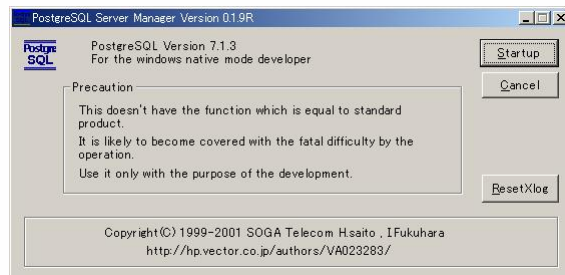
20 clients, 900 inserts per client, 1 insert per transaction, 4 different tables.

7.0.2 標準 10:52 完了平均

同スレッド対応 2:42 完了平均

7.1beta3 1:13 完了平均

・Single-User限定のThread-Modelの実験(Version 7.1)



CreateThread対応で作成を始める、しかし、改造が大量のため遅々として進まず。

```
ServLp_handle = CreateThread(NULL, 0, (unsigned long) ServerLoop_runct1, (void *) SS_port, 0, &ThreadId);
```

また、本体のほとんどのdll化を試みて効果を確認。

SQL Study用途として公開を始める。

・Jan Wieck氏のProcess-Modelの実現(Version 7.2)

Peer Direct社のライセンスを含んだが一般公開される。

```
/* Point to the DOS header in memory */
PIMAGE_DOS_HEADER pDosHdr = (PIMAGE_DOS_HEADER)hMod;
/* From the DOS header, find the NT (PE) header */
PIMAGE_NT_HEADERS pNtHdr = (PIMAGE_NT_HEADERS)(hMod + pDosHdr->e_lfanew);
PIMAGE_SECTION_HEADER pSection = IMAGE_FIRST_SECTION( pNtHdr );
/* RVA is offset from module load address */
DWORD rva = (DWORD)addr - hMod;
```

というスタックメモリを取り込むなど荒業ではあるが本体構造に合った方式が取れる。

ようするに、スタックを引継ぎ子供のフリをさせることで擬似的にfork()と同じ仕組みにする。

※本体はconversionを含まなかったため、対応して公開を打診したら、Jan氏がPeerDirect社を離れたため公開は頓挫してしまった。

PostgreSQL for Windows



・PowerGresでのThread-Modelの実現(Version 7.3)



SRAと新生開発(SKC)による共同開発により実現。

Thread方式での問題点をすべて網羅した。

Microsoft Windowsの基本に則ったVC6による開発。

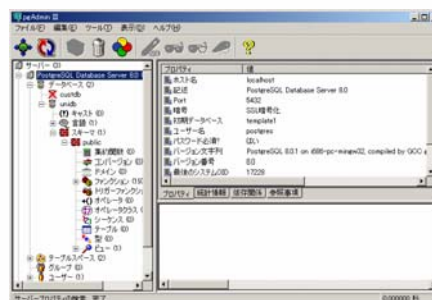
本家へのフィードバックを前提にBruce Momjian氏も参画。

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



・本家のPostgreSQL 8.0リリース



本家コミュニティによる実現であり、本体のライセンスで供給される。

これまでのChallenge for windows関係のフィードバックされた技術を網羅して最適化された対応がされている。

pgAdminIIIやpginstaller,クライアントアクセスAPI群、contribモジュール、PostGISなどそのままWindowsユーザに使える環境を提供することができた。

標準でOpenSSLを組み込んでセキュアなコネクションも可能になっている。

Copyright © 2005 Hiroshi Saito All rights reserved.

2. PostgreSQL 8.0 for Windows の特化した機能

- ・fork()システムコールの対応
- ・シグナルシステムコールの対応
- ・sync()システムコールの対応
- ・Windows イベントログの対応
- ・シンボリックリンク(リパースポイント)の対応
- ・Windows サービスの対応

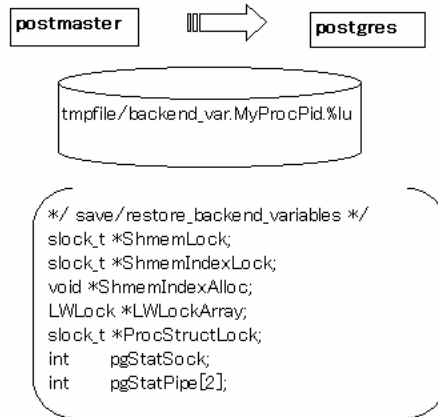
- ・fork()システムコールの対応

Unixシステムのfork()は親の状態を複製できるがWindowsのCreateProcess()は生成するプロセスが初期状態で起動される。

8.0では、この解決にテンポラリファイルを中継させることで実現させている。

8.0以前のPostgreSQLは内部処理関数でstaticな領域を多用していたため困難であったが現在はほとんどオート変数で書き換えられている。

CreateProcessするごとにグローバル変数領域を引き継ぐ



・シグナルシステムコールの対応

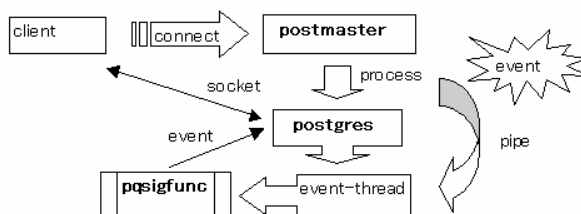
Windowsでは、Unixなみのシグナルを備えていない。

PostgreSQLは、さまざまなシグナルを使いその挙動をコントロールしている。

HUP INT QUIT ABRT TERM USR1 USR2など

この代替実装として、イベント、メッセージ、パイプを使い相当の対応をしなくてはならない。

シグナルの代替実装概念図



- 1) ioctlsocket(s, FIONBIO, &on) <=== Blocking socket functions
- 2) WaitForMultipleObjectsEx <==== socket & signal

WaitForMultipleObjectExはsocket,eventを捉えることができる。

Pipeメッセージからは、**WaitForSingleObject()**を待機させて捉えるなど処理は複雑。

・sync()システムコールの対応

以前のバージョンでは、checkpointのためのディスクフラッシュのためsync()システムコールを呼び出す必要がありました。これは、バックエンドライタープロセスの実現により基本的なディスク書き込み機能をサポートしています。

ノード単位にコントロールしてfsyncの対応となりました。

Windowsでは、同等の機能を_commitにより実現しています。

これらは、wal_sync_methodにより、強制的にWALをディスクに更新させるために使用される方法を設定できます。

取り得る値は、下記があります。

fsync (コミットの度にfsync)を呼び出します。)

fdatasync (コミットの度に fdatasync) を呼び出します。)

open_sync (O_SYNCオプション付きのopen() でWALファイルを書き出します。)

open_datasync (O_DSYNCオプション付きのopen() でWALファイルを書き出します。)

しかし、すべてがそのプラットフォームで使える方法ではありません。



```

Simple write timing:
write 1.582000
Compare fsync times on write() and non-write() descriptor:
(If the times are similar, fsync() can sync data written
on a different descriptor.)
write, fsync, close 61.979000
write, close, fsync 58.104000
Compare one o_sync write to two:
one 16k o_sync write 0.490000
two 8k o_sync writes 0.471000
Compare file sync methods with one 8k write:
(o_sync unavailable)
open o_sync, write 0.240000
(fdatsync unavailable)
write, fsync, 41.330000
Compare file sync methods with 2 8k writes:
(o_sync unavailable)
open o_sync, write 0.470000
(fdatsync unavailable)
write, fsync, 42.351000
    
```

src/tools/fsyncにfsynctestプログラムが追加されました。

これはwal_sync_methodの設定を助けます。

左記は、Windows-XPでの結果です。

(open_datasync対応は、8.0.2以降で反映されています。)

```

+#ifdef WIN32
+#define fsync(a) _commit(a)
+#define O_DSYNC 0x0080
    
```

※open_datasync対応によって極端な差を見ることが出来ます。

※左記票は、O_SYNCをO_DSYNCで代用試験した結果です。(8.0.1ベース)

O_NOINHERIT 0x0080 と等価
(共有ファイル記述子の作成を禁止)



・Windowsイベントログの対応

新たに、標準でpgeventが加わりました。

アプリケーションログ用のソースとして登録します。

```

SYSTEM%%CurrentControlSet%%Services%%EventLog%%Application%%PostgreSQL
    
```

EventMessageFile = pgevent.dll <== 実際はインストールパス

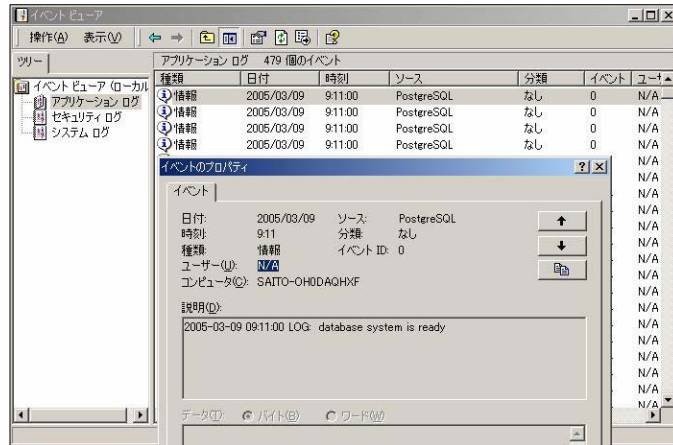
```

TypesSupported = EVENTLOG_ERROR_TYPE | EVENTLOG_WARNING_TYPE |
EVENTLOG_INFORMATION_TYPE
    
```

postgresql.confの出力先を切り替えます。

```

log_destination = 'eventlog'
    
```



・シンボリックリンク(リパースポイント)の対応

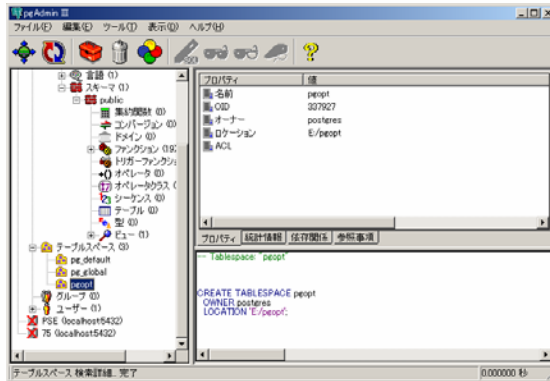
8.0は、新しくtablespace機能を実現しています。

これは、シンボリックリンクによりファイルシステムを分けてアクセス効率を上げる手段となります。

Windowsでは、この機能をNTFSのリパースポイントという機能で実現しています。

しかし、それが使えるのは、NTFS5以上である必要があります。(古いNTFS4やFATでは利用できません)

PostgreSQL for Windows



C:\Program Files\PostgreSQL\8.0\data\pg_tblspc>dir

```
2005/03/10 14:09 <DIR> .
2005/03/10 14:09 <DIR> ..
2005/03/10 14:09 <JUNCTION> 337927
```

E:\pgopt>dir

```
2005/03/10 14:10 <DIR> .
2005/03/10 14:10 <DIR> ..
2005/03/10 14:10 <DIR> 337928
2005/03/10 14:09 4 PG_VERSION
```

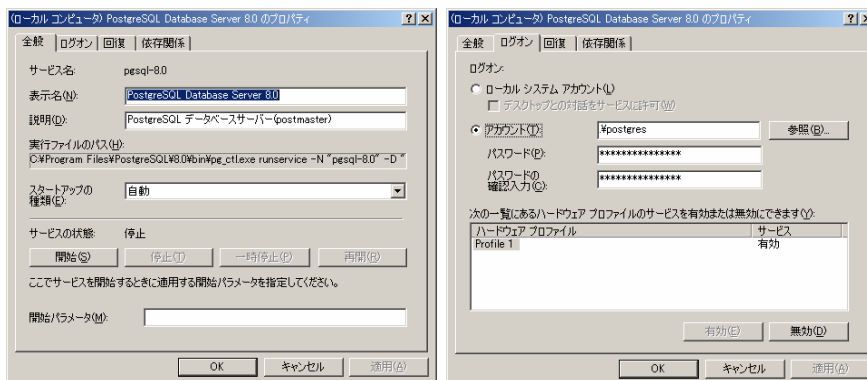
リバースポイントは、<JUNCTION>として表現されます。
しかし、エクスプローラで認識することが困難ですので注意が必要です。

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



・Windowsサービスの対応



PostgreSQLはセキュリティ上、Admin権限を持ったユーザでは起動できないように制限しています。

Copyright © 2005 Hiroshi Saito All rights reserved.



3.PostgreSQL 8.0 for Windows の性能

- ・パフォーマンス測定
- ・障害時の信頼性



・パフォーマンス測定

AMD Athlon(tm) 600MHz CPU Memory 256	PentiumIII 866MHz CPU Memory 512
FreeBSD 5.1-RELEASE	Windows 2000 SP4
WDC WD400BB-00DEA0 7200RPM	WDC WD400BB-00DEA0 7200RPM

パラメータ	c=1,t=10	c=5,t=10	c=10,t=10	c=15,t=10	c=20,t=10	c=25,t=10	c=30,t=10	c=35,t=10	c=40,t=10
BSD(8.0)	95.52102	91.94083	79.53526	74.45595	72.91598	71.11498	64.48798	66.8374	65.87651
W2K(7.4)	33.33911	34.85056	39.67837	40.50006	39.13064	40.70015	38.83524	35.82964	35.44608
W2K(7.5)	26.31344	32.31617	29.23873	32.75396	29.41947	29.84424	29.44322	30.33879	29.80474
W2K(8.0)	20.67654	26.41111	27.03335	24.4319	26.37327	24.10967	24.23388	23.16796	23.71169
W2K(8.0)※	60.19238	62.30545	63.60862	63.78573	61.96326	60.29164	60.00359	60.28258	59.85023

TPC-B tps (including connections establishing)

c=client, t=transaction

W2K(7.4)は Microsoft VC6+ により作成で libpostgres.dll化を図り Thread対応のモデル

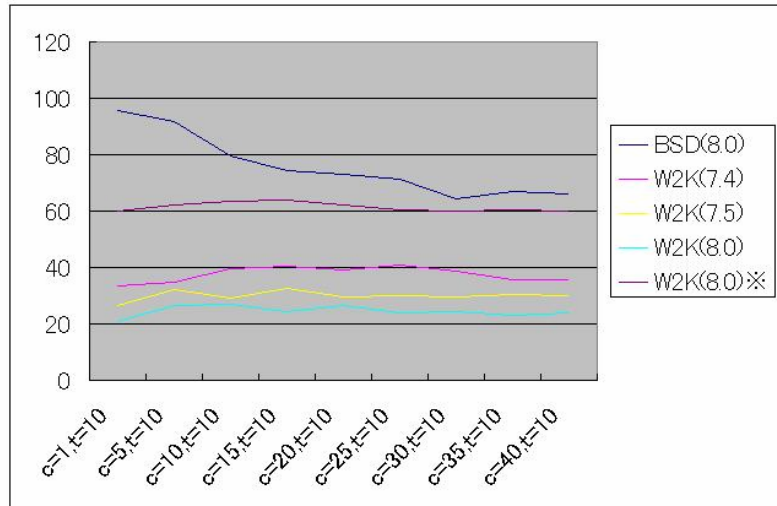
W2K(7.5)は Microsoft VC6+ により作成で libpostgres.dll化を図り Process対応のモデル

W2K(8.0)は MinGW-gcc で postmaster,postgresそれぞれ exe単体となる作り

※7.4,7.5は独自実装です。(ja Team版)

8.0 は、正規のリリース8.0.1版です。

8.0※ は、wal_sync_method=open_datasync CVS-Current対応版です。



Copyright © 2005 Hiroshi Saito All rights reserved.

・障害時の信頼性

PITR(Point In Time Recovery)の機能

Windowsでの指定方法はpostgresql.confの
archive_command = 'copy "%p" "F:/bkup/%f"'
のように設定します。

ベースバックアップの作成方法

```
template1=# SELECT pg_start_backup('SAITO');
pg_start_backup
```

```
-----
0/B46918
(1 row)
```

※→PGDATAのバックアップ

バックアップ履歴の作成方法

```
template1=# SELECT pg_stop_backup();
pg_stop_backup
```

```
-----
0/B46958
(1 row)
```

※WALより任意の時点から回復が可能になった。(しかし、WALの信頼性は・・・)

※使い方は基本的にUnixベースと同じ方法となります。

Copyright © 2005 Hiroshi Saito All rights reserved.

WAL(Write-Ahead Logging)の信頼性は?

たとえば電源遮断時の保証はあるのか?

HDD Write Back Cache はその問題をより複雑にする。

WindowsのジャーナルファイルシステムはWrite Back Cache の影響を受ける。

この試験のために、Dave Page氏がpowerfail.cプログラムを提供

```
CREATE TABLE pgtest(id serial, v1 int4, v2 int4, v3 int4, v4 int4, v5 int4, total int4, average int4);
```

```
-- テスト用レコードを10000 INSERT
```

```
* v1 ~ v5 = 1 ~ 10000 の範囲の乱数
```

```
* total = v1 + v2 + v3 + v4 + v5
```

```
* average = total / 5
```

```
-- RUN(トランザクション更新)
```

A. 1 ~ 1000 の範囲の乱数により一回のトランザクション更新回数を決定する。

B. トランザクションを開始し、A で決定した回数だけ乱数により選択したレコードを更新する。

C. トランザクションを COMMIT。

```
-- CHECK(整合性)
```

```
/* SELECT COUNT(DISTINCT(id)) FROM pgtest SB 10000 */
```

```
/* SELECT MAX(id) SB 10000 */
```

```
/* SELECT MIN(id) SB 1 */
```

```
/* SELECT COUNT(*) WHERE v1+v2+v3+v4+v5 != total SB 0 */
```

```
/* SELECT COUNT(*) WHERE (v1+v2+v3+v4+v5) / 5 != average SB 0 */
```

PostgreSQL for Windows



```
bsd2% powerfail run W2K custdb postgres postgres
Checking data consistency...
Starting run of 525 updates.
Checking data consistency...
Starting run of 190 updates.
Checking data consistency...
Starting run of 487 updates.
Checking data consistency...
Starting run of 600 updates.
=== Power off
Checking data consistency...
Starting run of 964 updates.
could not receive data from server: Operation timed out
=== Power on
bsd2% powerfail check W2K custdb postgres
Checking data consistency...
```

This was repeated.

```
=====
01) None
02) None
03) None
04) None
05) None
06) None
07) None
08) None
09) None
10) None
```

I check the reliability of 8.0.1 with the windows.
<http://archives.postgresql.org/pgsql-hackers/2003-01/msg01315.php>

I did check operation.

```
-----
PentiumIII 866MHz 512MB
WDC WD400BB-00DE00
Write back cache on IDE disk disabled.
```

コンセントを抜くと・・・

マシンが壊れるか、Windowsが
壊れるか、PostgreSQLが壊れる
か？

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



4.コントロールセンターとしてのpgAdminIII

クロスプラットフォームのwxWidgets(旧wxWindows)を
利用し、Linux,FreeBSD,Windowsで動作する。



やっと
入れた

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



The screenshot displays the pgAdmin III interface for a PostgreSQL Database Server 8.0 (localhost:5432). The left pane shows a tree view of the database structure, including a table named 'branches'. The central pane shows a query editor with the SQL command 'select count(*) from branches;' and a diagram illustrating a relationship between 'branches' and 'Assesrate'. The right pane shows the 'Properties' window for the 'postgres' database, listing various parameters such as '名前' (name), 'OID', '所有者' (owner), 'ACL', 'テーブル空間' (tablespace), 'エンコーディング' (encoding), and '初期スキーマ' (initial schema). Below the properties, there is a section for 'プロパティ' (properties) with a table of values and a section for 'SQL' commands, including 'CREATE DATABASE postgres WITH OWNER = postgres ENCODING = 'EUC_JP' TABLESPACE = pg_default;'.

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



5.pginstaller

1. WiX toolkit (<http://wix.sourceforge.net>)
2. Perl (tested with ActiveState perl 5.8.3)
3. MinGW*Msys
4. Installed tree of postgresql
5. PostgreSQL source tree
6. pgAdmin Support module
7. Npgsql (from <http://gborg.postgresql.org/project/npgsql>)
8. psqLODBC (from <http://gborg.postgresql.org/project/psqlodbc>)
9. PostgreSQL JDBC (from <http://jdbc.postgresql.org>)
10. PgOleDb (from <http://gborg.postgresql.org/projects/oledb>)
- 11a. PostGIS (from <http://postgis.refractor.net>)
- 11b. libgeos (from <http://geos.refractor.net>)
- 11c. proj4 (from <http://www.remotesensing.org/proj>)

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



米Microsoftが「Windows Installer Xml」(WiX)と呼ばれるWindows製品インストーラーをオープンソースソフトとして公開した。オープンソースプロジェクトのWebサイト、SourceForge.netに掲載した。同社がオープンソースとしてプログラムを提供するのは初めて。

「WiX」は、XMLソースコードからWindows製品をインストールするパッケージを構築するツールセット。コンパイラ、リンカー、ライブラリツール、デコンパイラで構成され、これを利用するとMSIファイルやMSMファイルを生成できる。

WiXはもともとMicrosoft社内で開発されてきたもので、同社のOffice、SQL Server、BizTalkなどの製品開発チームは実際に同ツールを利用してインストーラーを構築しているという。ライセンス形態はCommon Public License(CPL)で提供する。CPLは、Open Source Initiative(OSI)が認めているライセンスの一つで、コードの改変と商利用が可能。

「MySQL Database Server 4.1」でも、インストーラ(WiX)が採用される。

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



The screenshot shows the 'サービスの構成' (Service Configuration) window of the PostgreSQL Windows installer. It contains the following fields and options:

- サービスのインストール
- サービス名: PostgreSQL Database Server 8.0.0-beta4-dev1
- アカウント名: postgres
- ドメイン名: SAITO-OH0DAQHXF
- パスワード: *****
- パスワードの確認: *****

Below the fields, there is a warning message: 「サービスアカウントは、PostgreSQLデータベース・サーバーを運営するアカウントです、それはローカルなアドミニストレータグループのメンバーではないけません。もし、まだアカウントを作成していなかったら、インストーラは、あなたのために作成することができます。アカウント名、およびパスワードを入力して、あるいはパスワードを空白のまま自動生成されたものを使ってください。」

At the bottom, there are three buttons: 「戻る(B)」, 「次へ(N)」, and 「キャンセル」.

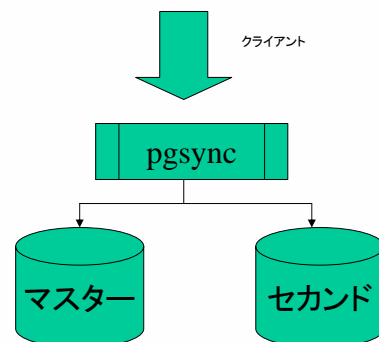
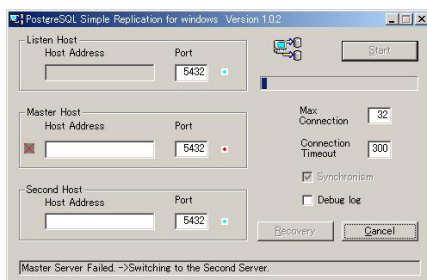
Copyright © 2005 Hiroshi Saito All rights reserved.

6. おもしろい話題

- ・pgsync on Windows
- ・libpgyb.dll (VBA,VB interface)
- ・PostgreSQL徹底活用ガイド

・pgsync on Windows

※PostgreSQL8.0で動作。



FailOverの動作機能を保持。

現在は同期方式のみであるが非同期対応を計画中。

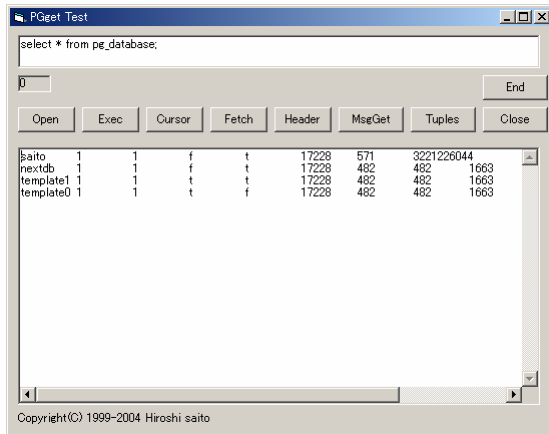
手軽さを第一優先で開発。

※スクリプトパーサを備えていないため検索のみもスレーブ動作させるためオーバヘッドがまだある・・・

PostgreSQL for Windows



・libpqyb.dll (VBA,VB interface)



- ・手軽なアクセスAPIを装備
- ・ODBCを使わない
- ・OpenSSL対応(libpq準拠)
- ・Excel-VBAでも手軽に作れる

Copyright © 2005 Hiroshi Saito All rights reserved.

PostgreSQL for Windows



・PostgreSQL徹底活用ガイド

- 書名: PostgreSQL徹底活用ガイド for Windows
- ISBN番号: 4-8443-2099-8
- 定価: ¥3,129(本体 ¥2,980+税)
- 発売日: 3月31日



Copyright © 2005 Hiroshi Saito All rights reserved.

ありがとうございました。



PostgreSQL

<http://www.postgresql.jp>

==== 終わり ====