

PostgreSQL for IBM i 導入ガイド

目次

1. 事前準備
2. IBM i V5R4 及び V6R1 での導入方法
3. POSTGRES ユーザのプロファイルを作成する
4. PostgreSQL の導入
5. PostgreSQL の開始
6. PostgreSQL の確認
7. PostgreSQL の停止
8. IBM i V7R1 での導入方法
9. PHP のサンプルコード

全体の流れ

IBM i V5R4 及び V6R1 での導入と IBM i V7R1 では導入の方法が違います。

そのため IBM i V5R4 及び V6R1 での導入の場合は

1. 事前準備
2. IBM i V5R4 及び V6R1 での導入方法
3. POSTGRES ユーザのプロファイルを作成する
4. PostgreSQL の導入

を行います。

IBM i V7R1 では

8. IBM i V7R1 での導入方法

を行います。

5. PostgreSQL の開始
6. PostgreSQL の確認
7. PostgreSQL の停止
9. PHP のサンプルコード

は IBM i V5R4 及び V6R1 と IBM i V7R1 で共通です。

1. 事前準備

(ア) ソースコードのダウンロード

下記のサイトより必要なソースをダウンロードする。AIX Version のあるものは V4.3 を選択する。

Gcc: <http://bio.gsi.de/DOCS/AIX/aixpdslib.seas.ucla.edu/packages/gcc.html>

Gzip: <http://bio.gsi.de/DOCS/AIX/aixpdslib.seas.ucla.edu/packages/gzip.html>

Make: <ftp://ftp.sunfreeware.com/pub/freeware/SOURCES/make-3.81.tar.gz>

Postgresql: <http://www.postgresql.org/ftp/source/> (linux)

Readline : <http://tiswww.case.edu/php/chet/readline/rltop.html>

Zlib : <http://www.zlib.net/>

(イ) IBM i のユーザ(POSTGRES)を作る

WRKUSRPRF USRPRF (POSTGRES)

CHGJOB LANGID (ENU) CNTRYID (US) CCSID (5035)

(ウ) PASE 上にディレクトリを作る

/usr/local/pgsql :ここに Postgresql のデータベースが作られる

/usr/local/pgsql/src :ここはコンパイル等、構築のためのワークエリア

(エ) ダウンロードしたソースプログラムをアップロードする

gcc.4.1.0.tar.Z /usr/local/pgsql/src/gcc.4.1.0.tar.Z

gzip.1.2.4a.tar.Z /usr/local/pgsql/src/gzip.1.2.4a.tar.Z

make-3.82.tar.gz /usr/local/pgsql/src/make-3.82.tar.gz

postgresql-8.4.7.tar.gz /usr/local/pgsql/src/postgresql-8.4.7.tar.gz

readline-6.1.tar.gz /usr/local/pgsql/src/readline-6.1.tar.gz

zlib-1.2.5.tar.gz /usr/local/pgsql/src/zlib-1.2.5.tar.gz

(アップロードは FTP で行うが、今回は「FFFTP」のツールを利用した)

2. IBM i V5R4 及び V6R1 での導入方法

(1) gcc の導入(gcc.4.1.0)

```
cd /usr/local/pgsql/src
```

```
uncompress gcc.4.1.0.tar.Z
```

```
tar xvf gcc.4.1.0.tar
```

導入後の確認として以下のコマンドで確認する

```
コマンド: /usr/local/pgsql/src/usr/local/bin/gcc -v
```

表示結果:

```
Using built-in specs.
```

```
Target: powerpc-ibm-aix5.3.0.0
```

```
Configured with: ../gcc-4.1.0/configure --disable-nls
```

```
Thread model: aix
```

```
gcc version 4.1.0
```

(2) gzip の導入(gzip.1.2.4a)

```
cd /usr/local/pgsql/src
```

```
uncompress gzip.1.2.4a.tar.Z
```

```
tar -xvf gzip.1.2.4a.tar
```

(3) GNU の make の導入(make-3.82)

```
cd /usr/local/pgsql/src  
/usr/local/pgsql/src/usr/local/bin/gunzip make-3.82.tar.gz  
tar -xvf make-3.82.tar  
cd make-3.82  
export PATH=$PATH:/usr/local/pgsql/src/usr/local/bin  
./configure --prefix=/usr/local/pgsql/src/usr/local  
make all  
make install
```

(4) Readline の導入(readline-6.1)

```
cd /usr/local/pgsql/src  
/usr/local/pgsql/src/usr/local/bin/gunzip readline-6.1.tar.gz  
tar -xvf readline-6.1.tar  
cd readline-6.1  
./configure --prefix=/usr/local/pgsql/src/usr/local  
make all  
make install
```

(5) zlib の導入(zlib-1.2.5)

```
cd /usr/local/pgsql/src  
/usr/local/pgsql/src/usr/local/bin/gunzip zlib-1.2.5.tar.gz  
tar -xvf zlib-1.2.5.tar  
cd zlib-1.2.5  
./configure --prefix=/usr/local/pgsql/src/usr/local  
make  
make install
```

(6) 最後の出来上がったオブジェクトをコピーする

```
cp -R /usr/local/pgsql/src/usr/local/* /usr/local
```

3. POSTGRES ユーザのプロファイルを作成する

(1) /home/POSTGRES/.profile に下記の内容を記述する

```
export PATH=$PATH:/usr/local/bin:/usr/local/pgsql/bin  
export LIBPATH=$LIBPATH:/usr/lib:/usr/local/lib:/usr/local/pgsql/lib  
export POSTGRES_HOME=/usr/local/pgsql  
export PGLIB=$POSTGRES_HOME/lib  
export PGDATA=$POSTGRES_HOME/data  
export MANPATH="$MANPATH":$POSTGRES_HOME/man  
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH":$PGLIB
```

4. PostgreSQL の導入(postgresql-8.4.7)

(1) ログインを「POSTGRES」で行う

(2) Postgresql のソースを解凍し、コンパイルを行う

```
cd /usr/local/pgsql/src
/usr/local/pgsql/src/usr/local/bin/gunzip postgresql-8.4.7.tar.gz
tar -xvf postgresql-8.4.7.tar
cd /usr/local/pgsql/src/postgresql-8.4.7
./configure --prefix=/usr/local/pgsql ¥
    --bindir=/usr/local/pgsql/bin --datadir=/usr/local/pgsql/share ¥
    --sysconfdir=/usr/local/pgsql/etc --libdir=/usr/local/pgsql/lib ¥
    --includedir=/usr/local/pgsql/include --with-template="aix" ¥
    --with-includes=/usr/local/include --with-libraries=/usr/local/lib
```

```
/usr/local/bin/make
```

(make の結果として「All of PostgreSQL successfully made. Ready to install.」が表示される)

```
/usr/local/bin/make install
```

(install の結果として「PostgreSQL installation complete.」が表示される)

(3) PostgreSQL のデータベースの初期化

```
initdb -D /usr/local/pgsql/data --no-locale -E UTF8
```

このとき「WARNING: enabling "trust" authentication for local connections

You can change this by editing pg_hba.conf or using the -A option the next time you run initdb.

Success. You can now start the database server using:」

postgres -D /usr/local/pgsql/data or pg_ctl -D /usr/local/pgsql/data -l logfile start」の表示がされる

(4) /usr/local/pgsql/data/postgresql.conf を修正

(WRKLNK OBJ ('/usr/local/pgsql/data/postgresql.conf'))

```
#wal_sync_method = fsync          # the default is the first option の行を下記のように修正する
wal_sync_method = fsync          # the default is the first option
```

5. PostgreSQL の開始

(1) ログインを「POSTGRES」で行う

(2) pg_ctl -D /usr/local/pgsql/data -l logfile start コマンドで開始

6. PostgreSQL の確認

(1) ログインを「POSTGRES」で行う

(2) psql での確認

```
> psql
psql (8.4.7)
Type "help" for help.
postgres=#
```

> create table test (shainno int,shimei text): (テーブル作成)

create table test (shainno int,shimei text):

CREATE TABLE

postgres=#

> ¥d (テーブル一覧)

¥d

List of relations

Schema	Name	Type	Owner
--------	------	------	-------

-----+-----+-----+-----

public	test	table	postgres
--------	------	-------	----------

(1 row)

postgres=#

> ¥d test (フィールド一覧)

¥d test

Table "public.test"

Column	Type	Modifiers
--------	------	-----------

-----+-----+-----

shainno	integer	
---------	---------	--

shimei	text	
--------	------	--

postgres=#

> insert into test (shainno,shimei) values (1,'munetika'); (インサート)

insert into test (shainno,shimei) values (1,'munetika');

INSERT 0 1

postgres=#

> select * from test: (セレクト)

select * from test:

shainno	shimei
---------	--------

-----+-----

1	munetika
---	----------

(1 row)

postgres=#

> SELECT version (); (バージョンチェック)

SELECT version ();

version

PostgreSQL 8.4.7 on powerpc-ibm-os400, compiled by GCC gcc (GCC) 4.1.0, 32-bit

(1 row)

postgres=#

> ¥q (psqlの終了)

¥q

(3) ps ax での確認

```

297278      - A      0:00 postgres: writer process
297279      - A      0:00 postgres: wal writer process
297280      - A      0:00 postgres: autovacuum launcher process
297281      - A      0:00 postgres: stats collector process
297410      - A      0:00 /QOpenSys/usr/bin/-sh -i
297411      - A      0:00 ps ax

```

7. PostgreSQL の停止

- (1) ログインを「POSTGRES」で行う
- (2) 停止コマンド: `pg_ctl -D /usr/local/pgsql/data -l logfile stop`

8. IBM i V7R1 での導入方法

- (1) IBM i V6R1 または IBM i V5R4 で作られた PostgreSQL を圧縮する


```

cd /usr/local
tar cvf postgresql.tar ./pgsql
compress -f postgresql.tar (postgresql.tar.Z が作られる)

```
- (2) IBM i V7R1 の POSTGRES ユーザを作成する(前記と同様)
- (3) IBM i V7R1 の POSTGRES ユーザのプロファイルを作成す(前記と同様)
- (4) postgresql.tar.Z (59MB)を FTP で /usr/local にコピーする
- (5) `cd /usr/local`
- (6) `uncompress postgresql.tar.Z`
- (7) `tar xvf postgresql.tar`
- (8) POSTGRES でログインする
- (9) データベースの初期化を行う


```

initdb -D /usr/local/pgsql/data --no-locale -E UTF8

```
- (10) ファイルの修正


```

/usr/local/pgsql/data/postgresql.conf を修正(前記と同様)

```

9. PHP のサンプルコード

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>postgresql-test</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf8">
  </head>
  <body>
    <p>PostgreSQL テスト</p>
  </body>
</html>
<?php
  $conn = pg_connect ("host=localhost dbname=test user=xxxxxx password=xxxxxx");
  if (!$conn) {

```

```

        echo "コネクトでエラーが発生しました<BR>¥n";
        exit;
    }

    $sql = "select name,telno,email from address";
    $result = pg_query ($sql);
    if (!$result) {
        echo "照会でエラーが発生しました<BR>¥n";
        pg_close ($conn);
        exit;
    }
    $num = pg_num_rows ($result);
    echo "<table border=¥"1¥">¥n";
    echo "<caption>簡易アドレス帳 (PostgreSQL-PHP) </caption>¥n";
    echo "<tr><th>名前</th><th>電話番号</th><th>メールアドレス</th></tr>¥n";
    for ($i = 0; $i < $num; $i++) {
        $row = pg_fetch_row ($result, $i);
        echo "<tr>¥n";
        echo "<td>" . $row [0] . "</td>¥n";
        echo "<td>" . $row [1] . "</td>¥n";
        echo "<td>" . $row [2] . "</td>¥n";
        echo "</tr>¥n";
    }
    echo "</table>¥n";
    pg_close ($conn);
echo '</body></html>';
?>

```

以上