****

**Linux PERL安装以及环境部署手册**

第一部分 SUSE Linux上PERL程序安装

## 前提条件

首先需要安装Perl的PC上需要首先安装Teradata的TTU13.10，再者准备好Perl的DBD以及DBI软件。

A)确认Linux上已经安装了gcc，SUSE Linux默认的情况下gcc是安装好的，检查gcc的安装情况请在命令行中输入如下命令：

# gcc -v

Using built-in specs.

Target: x86\_64-suse-linux

Configured with: ../configure --prefix=/usr --infodir=/usr/share/info --mandir=/usr/share/man --libdir=/usr/lib64 --libexecdir=/usr/lib64 --enable-languages=c,c++,objc,fortran,obj-c++,java,ada --enable-checking=release --with-gxx-include-dir=/usr/include/c++/4.3 --enable-ssp --disable-libssp --with-bugurl=http://bugs.opensuse.org/ --with-pkgversion='SUSE Linux' --disable-libgcj --disable-libmudflap --with-slibdir=/lib64 --with-system-zlib --enable-\_\_cxa\_atexit --enable-libstdcxx-allocator=new --disable-libstdcxx-pch --enable-version-specific-runtime-libs --program-suffix=-4.3 --enable-linux-futex --without-system-libunwind --with-cpu=generic --build=x86\_64-suse-linux

Thread model: posix

**gcc version 4.3.4 [gcc-4\_3-branch revision 152973]** (SUSE Linux)

标记为红色的即为该PC上的gcc版本。

B)确认PC上已经安装Perl，默认的情况下Suse Linux上已经安装了Perl，查看当前的Perl版本方法如下：

# perl -v

This is **perl, v5.10.0 built for x86\_64-linux-thread-multi**

Copyright 1987-2007, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the

GNU General Public License, which may be found in the Perl 5 source kit.

Complete documentation for Perl, including FAQ lists, should be found on

this system using "man perl" or "perldoc perl". If you have access to the

Internet, point your browser at http://www.perl.org/, the Perl Home Page.

标记为红的即为该PC上的perl版本（5.8以上版本即可）。

C)安装Teradata ODBC，当Teradata的TTU13.10都安装好后，可在目录“系统安装目录在/opt/teradata/client/13.10”下查找到名称为odbc.ini文件。可通过下面的命令进行查找：

# find /opt/teradata/client/13.10 -name odbc.ini

/opt/teradata/client/13.10/odbc\_64/odbc.ini

odbc\_64两个目录下的odbc.ini文件分别表示已经安装了64位的Teradata ODBC。如果没有找到文件，则通过下面的步骤进行安装：

准备TTU13.10的安装介质TTU131001\_TUP\_V2.zip，上传到SUSE Linux环境上。

使root用户登陆，进入TTU13.10的安装介质目录，用下面的命令进行解压：

#unzip TTU131001\_TUP\_V2.zip

解压成功后，进入解压后的unzip TTU131001\_TUP\_V2目录，对目录下.sh后缀的文件添加执行权限：

#chmod u+x ./\*.sh

#chmod u+x ./.\*.sh

#ls –l

total 284

-rw-r--r-- 1 root root 405 Jun 30 2002 #.bat

-rw-r--r-- 1 root root 28000 Jul 16 2010 .remove\_teradata\_client\_packages.sh

-rw**x**-r--r-- 1 root root 163530 Jul 31 2010 .setup.sh

drwxr-xr-x 9 root root 4096 Dec 5 12:53 HP-UX

drwxr-xr-x 9 root root 4096 Dec 5 12:53 Linux

-rw-r--r-- 1 root root 32 Jul 15 2010 MEDIALABEL

drwxr-xr-x 2 root root 4096 Dec 5 12:53 TeraJDBC

-rw-r--r-- 1 root root 52 Jun 30 2002 autorun.inf

-rw**x**r--r-- 1 root root 22631 Jul 16 2010 package\_size.sh

-rw-r--r-- 1 root root 299 Apr 14 2009 setup.bat

-rw-r--r-- 1 root root 21694 Aug 3 2010 tar\_teradata\_client\_packages.bat

-rw**x**r--r-- 1 root root 14105 Aug 3 2010 tar\_teradata\_client\_packages.sh

执行.setup.sh脚本：

#sh ./.setup.sh

#############

###

###

### ###### ####### ## ###### ## ######### ##

### ## ## ## #### ## ## #### ## ####

### ## ## ## ## ## ## ## ## ## ## ## ##

### ## ## ## ## ## ## ## ## ## ## ## ##

### ##### ## ## ## ## ## ## ## ## ## ## ##

### ## ## ## ###### ## ## ## ###### ## ## ###### ##

### ## ## ## ## ## ## ## ## ## ## ## ##

### ## ## ## ## ## ## ## ## ## ## ## ##

### ######## #### ## ##### ## ## # ## ##

##

Raising Intelligence

Welcome to the Teradata Tools and Utilities 13.10 installation v.13.10.01.02

Copyright 2002-2010. Teradata Corporation. All Rights Reserved.

Platform: Linux 64bit

Media: Teradata Utility Pak - Volume 2

Which software do you want to install?

1. TeraGSS - Teradata GSS client package

2. tdicu - Teradata Shared Component for Internationalization

3. tdodbc - Teradata ODBC Driver

4. cliv2 - Teradata CLIv2

5. piom - Teradata Data Connector API

6. npaxsmod - Teradata Named Pipes Access Module

7. bteq - Teradata BTEQ Application

a. Install all of the above software

u. Remove previously installed Teradata Tools and Utilities software (pre 1310)

s. Display installed package sizes

q. Quit the installation

Enter one or more selections (separated by space):

输入‘3’安装Teradata ODBC Driver。

系统提示需要在安装ODBC前安装Teradata GSS client package，询问是否安装，回车键确定。

Enter one or more selections (separated by space): 3

Warning: Teradata GSS client package is a prerequisite for the

<cliv2> and <tdodbc> software.

The Teradata GSS client package software will be installed.

Do you want to continue with the installation? [y/n (default: y)]:

安装完毕后，输入‘s’显示已安装的程序包

Which software do you want to install?

1. TeraGSS - Teradata GSS client package

2. tdicu - Teradata Shared Component for Internationalization

3. tdodbc - Teradata ODBC Driver

4. cliv2 - Teradata CLIv2

5. piom - Teradata Data Connector API

6. npaxsmod - Teradata Named Pipes Access Module

7. bteq - Teradata BTEQ Application

a. Install all of the above software

u. Remove previously installed Teradata Tools and Utilities software (pre 1310)

s. Display installed package sizes

q. Quit the installation

Enter one or more selections (separated by space): s

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Teradata Utility Pak - Volume 2 v13.10.01.02

PATH=/usr/infa/software/TTU13.10/TTU131001\_TUP\_V2

Platform: Linux

Package: TeraGSS\_redhatlinux-i386 Size: 32,587 KB

Package: TeraGSS\_suselinux-x8664 Size: 50,341 KB

Package: bteq-i386 Size: 370 KB

Package: cliv2-noarch Size: 1,260 KB

Package: npaxsmod-i386 Size: 136 KB

Package: piom-noarch Size: 199 KB

Package: tdicu-noarch Size: 20,995 KB

Package: tdodbc-noarch Size: 42,226 KB

Package: Total i386/noarch Size: 97,773 KB

Package: Total x86\_64 Size: 50,341 KB

红色部分显示tdodbc（Teradata ODBC 驱动）已经安装完毕。

D）odbc.ini文件的配置

将odbc.ini文件拷贝到使用ODBC用户的目录下，（如etl用户拷贝到/home/etl）。

#cp /opt/teradata/client/13.10/odbc\_64/odbc.ini /home/etl/.odbc.ini

编辑/home/etl/.odbc.ini（当前PICC项目使用的是/usr/infa/.odbc.ini），修改ODBC配置

[ODBC]

InstallDir=/opt/teradata/client/13.10/odbc\_64

Trace=0

TraceDll=/opt/teradata/client/13.10/odbc\_64/lib/odbctrac.so

TraceFile=/usr/joe/odbcusr/trace.log

TraceAutoStop=0

[ODBC Data Sources]

dbc=tdata.so

[dbc]

Driver=/opt/teradata/client/13.10/odbc\_64/lib/tdata.so

Description=Teradata running Teradata V1R5.2

DBCName=picctdcop1

LastUser=

Username=

Password=

Database=

DefaultDatabase=

上面的配置文件中，红色部分为修改的地方。定义了一个名为dbc的ODBC Data Source，并且配置DBCName为TD数据库的host别名picctdcop1。DBCName也可以填写TD数据库的IP地址。

下面设置环境变量ODBCINI指向.odbc.ini文件的所在位置。编辑用户主目录下的.bash\_profile文件(PICC ETL Server环境下的/usr/infa下)

#vi /usr/infa/.bash\_profile

在文件结束添加一行：

export ODBCINI=/home/etl/.odbc.ini

E）测试ODBC是否配置正确

运行ODBC自带的例子：

#/opt/teradata/client/13.10/odbc\_64/samples/C/adhoc

linux-appserver152:/opt/teradata/client/13.10/odbc\_64/samples/C # /opt/teradata/client/13.10/odbc\_64/samples/C/adhoc

Enter Data Source Name: **dbc**

Enter UserID: **dbc**

Enter Password: **XXX**

Connecting with SQLConnect(DSN=dbc,UID=dbc,PWD=\*)...

.....ODBC connection successful.

ODBC version = -03.52.0000-

DBMS name = -Teradata-

DBMS version = -13.10.0209 13.10.02.09-

Driver name = -tdata.so-

Driver version = -13.10.00.02-

Driver ODBC version = -03.51-

(type quit to terminate adhoc)

Enter SQL string : sel date;

Executing SQLExecDirect("sel date;")...

SQL Statement [1]: 1 rows affected.

Date

2012-02-22

Enter SQL string : quit

'quit' command detected

ODBC connection closed.

输入数据源、用户名、密码后回车，程序显示了ODBC和Teradata数据库的版本信息，并且可以输入SQL语句对数据库进行操作。

## 安装DBI

用root用户进入DBI安装介质目录，解压安装包，进入解压目录，进行安装

#tar -xvf DBI-1.608.tar.gz

#cd DBI-1.608

# perl Makefile.PL

#make

#make install

如果不报错误，表示安装成功。

## 安装DBD

用root用户进入DBI安装介质目录，解压安装包，进入解压目录，进行安装。安装时需要指定teradata odbc的安装目录（-o/opt/teradata/client/13.10/odbc\_64）。

#tar -xvf DBD-ODBC-1.25.tar.gz

#cd DBD-ODBC-1.25

# perl Makefile.PL –o /opt/teradata/client/13.10/odbc\_64

#make

#make install

如果不报错误，表示安装成功。

## 查看DBI安装情况

用下面的命令可以查看DBI安装的信息：

# perl -MDBI -e "DBI->installed\_versions"

Perl : 5.010000 (x86\_64-linux-thread-multi)

OS : linux (2.6.32)

DBI : 1.608

DBD::Sponge : 12.010002

DBD::SQLite : 1.14

DBD::Proxy : 0.2004

DBD::ODBC : 1.25

DBD::Gofer : 0.011565

DBD::File : 0.36

DBD::ExampleP : 12.010007

DBD::DBM : 0.03

## PERL连接测试

最后为了验证整个安装是否成功，还需要编写一个ERPL测试脚本进行测试。下面是一个测试例子，testodbc.pl，这个脚本连接数据库并且从dbc.tables中读取数据打印输出到屏幕上。执行方法（案例脚本在/usr/infa/software/DBD\_PERL/test目录下）：

#perl testodbc.pl <数据源> <用户名> <密码>

连接成功输出“Connect Teradata database ok”，表示安装正常，整个可以使用了。

下面是testodbc.pl的内容

use strict;

use DBI;

sub connect\_db

{

my $dbh = DBI->connect("dbi:ODBC:$ARGV[0]",

"$ARGV[1]",

"$ARGV[2]",

{ RaiseError=>1,

AutoCommit =>0

}

);

unless (defined($dbh)) {

die "cann't connect database !";

}

return $dbh;

}

sub sampleSQL

{

my ($dbh) = @\_;

unless (defined($dbh)) {

return -1;

}

my $sqlStr = "SELECT DatabaseName,TableName from dbc.tables;";

my $sth = $dbh->prepare($sqlStr);

unless ($sth) {

print "Failed to prepare statement for $sqlStr.\n";

return -1;

}

my $ret = $sth->execute();

unless ($ret) {

print "Failed to execute statement for $sqlStr.\n";

return -1;

}

my( $databasename, $tablename);

$sth->bind\_columns( undef, \$databasename, \$tablename);

while( $sth->fetch() ) {

print "$databasename, $tablename\n";

}

$sth->finish();

return 0;

}

sub main

{

my $dbh = connect\_db();

unless ($dbh) {

print "Failed to invoke function connect\_db().\n";

return 1;

}

my $ret = sampleSQL($dbh);

if ($ret < 0) {

print "Failed to invoke function sampleSQL().\n";

return -1;

}

print "Connect Teradata database ok.\n";

$dbh->disconnect();

return 0;

}

if ( $#ARGV != 2 ) {

print "Usage: perl testodbc.pl <datasource> <userID> <passwd>.\n";

exit(-1);

}

my $ret = main();

exit($ret);

第二部分 PICC ETL Perl环境说明

## ETL目录架构

TD沿用的ETL目录架构如下：

ETL

APP

--SAMPLE （连接测试程序）

--EDW （ods=>pdata迁移程序）

--orcl （ods=>gls程序）

bin --td ETL配置和程序

etc --连接数据库密码文件

主目录下有bin,etc和APP目录，bin目录存放比较常用的工具和脚本，etc下目前存放的bteq使用的登陆串文件，perl脚本都存放在bin目录下。

在调用perl脚本时，一般都采用全路径+文件名的方式进行调用：

|  |
| --- |
| perl C:\PROJECTS\PICC\ETL\APP\SYSTEM1\JOB11\bin\job1.pl |

说明：Etl\_pub.pm包会解析文件路径，将文件名所在目录往上2层的目录名称（JOB11）获取并存放在全局变量中$JOB ，将文件名所在目录往上3层的目录名称（SYSTEM1）获取并存放在全局变量$SYS中。

## 2、密码加密和解密

为了保护bteq登陆数据库的密码，ETL环境下使用IceCode加密程序，对数据库的密码进行加密。IceCode位于$AUTO\_HOME/bin目录下，有windows和Linux两个版本，在使用的时候需要进行区分。

### 2.1. IceCode加密密码

在命令行下输入

|  |
| --- |
| #IceCode.exe -d 密码 用户名 |

输出得到加密后的密码

[(G)$JI~L0RXCA}KR)OSWZW

### 2.2 IceCode解密密码

在命令行下输入

|  |
| --- |
| #IceCode.exe -e 加密后密码 用户名 |

输出得到解密后的密码

Z1E1FWO1GW8ZIIF`DAQJ613

### 2.3 使用EncodeLogon.pl生成BTEQ登陆串文件

在使用BTEQ时，为了管理方便，登陆串存放在$AUTO\_HOME/etc目录下。登陆串文件的命名为变量“系统”($SYS)+“\_LOGON” ($SYS\_LOGON)，如果“$SYS” 变量没有取到值，则使用默认的登陆串文件（目前配置的文件名为ETL\_LOGON）。登陆串的格式如下：

.LOGON <用户名>,<使用IceCode.exe加密后的密码>

注意：这里没有指定ip地址/主机名名，bteq使用的是默认的dbc主机，要在hosts文件中配置，windows的hosts文件在<WINDOWS安装目录>\system32\drivers\etc下，linux的hosts文件为/etc/hosts。用文本编辑器打开hosts文件后，在文件最后增加一行

**<Teradata 服务器IP地址> dbccop1**

使用EncodeLogon.pl生成BTEQ登陆串文件的过程如下，提示要求输入文件名，用户名和密码，最后系统提示”Logon file was generated”，表示生成成功了。



### 2.4 IceCode解密连接串

命令行下执行

|  |
| --- |
| #IceCode.exe <连接串> |

注意：IceCode.exe后面接的连接串的格式必须为不带ip地址或者主机名的格式：.LOGON <用户名>,<使用IceCode.exe加密后的密码>，否则解密结果不正确。

VMZT~KCE`PN]Y@OSAYBHJ%3

## 3、etl\_pub.pm公用包

为了对Perl编程时使用的公用参数和函数进行统一管理，将一些系统公用的参数和函数放在$AUTO\_HOME/etc/etl\_pub.pm文件中。包中的内容目前包括2部分：

1. 公用参数的定义和赋值

以下面的ETL\_DSN变量为例：

|  |
| --- |
| my $ETL\_DSN;  $ETL::ETL\_DSN = $ENV{"AUTO\_DSN"};  if ( !defined($ETL::ETL\_DSN) ) {  $ETL::ETL\_DSN = "dbc";  } |

先用$ENV()函数获取名为AUTO\_DSN的环境变量值，如果AUTO\_DSN没有定义，则使用”dbc”默认值。

常用的变量包括：

|  |  |
| --- | --- |
| $ETLHOME | ETL系统的所在的主目录 |
| $LOGON\_FILE | 登录串文件名 |
| $ETL\_DSN | ETL系统的ODBC使用的DSN名称 |

如果需要用新的变量，还可以在文件中添加。

包中调用的方式为$ETL::<变量名>，在调用前需要在perl脚本中把etl\_pub.pm引用进来。

|  |
| --- |
| my $HOME = $ENV{"AUTO\_HOME"};  unshift(@INC, "$HOME/bin");  require etl\_pub; #通用处理模块 |

1. 公用函数

包中常用的

|  |  |
| --- | --- |
| connectETL | 连接ETL系统对应的数据库，对应的登陆文件为ETL\_LOGON |
| disconnectETL | 断开连接 |
| ExecuteBTEQ | 调用BTEQ执行SQL语句，入参1为SQL语句 |
| getCurrentDate | 获取当前日期 |
| getCurrentDateTime | 获取当前日期+时间 |
| getDBTimeStamp | 获取数据库服务器的日期+时间 |
| formatTXDate | 格式化TX\_Date，将格式化后的TX\_date字符串返回 |
| GetCOLS | 获取表的字段名称列表 |

调用的方式为$ETL::<函数名>，在调用前需要在perl脚本中把etl\_pub.pm引用进来。使用的例子如下：

|  |
| --- |
| my $HOME = $ENV{"AUTO\_HOME"};  unshift(@INC, "$HOME/bin");  require etl\_pub; #通用处理模块  my $timestamp= ETL::getDBTimeStamp();  print "TimeStamp from DB:$timestamp\n" ; |

## 4、Sample代码说明

说明：

涉及ODBC编程的例子，需要修改数据库的用户名、密码，和DSN数据源。

连接bteq的例子，可能需要修改连接串的IP或者主机名。

如果使用了加密功能的，需要使用EncodeLogon来配置ETL\_LOGON文件的用户名和密码，并且检查hosts下面是否已经配置了dbccop1（配置方法见本文档前面2.3的内容）。

4.1 Perl ODBC编程的例子：

connectdb.perl 连接数据库



select.pl 从数据库表中调用select语句的例子



Bind\_param.pl 使用绑定变量调用SQL的例子



Transaction.pl 实现对数据库事务的控制的例子

4.2 Etl\_pub.pm调用例子

下面是调用etl\_pub.pm包函数的例子



4.3 Perl+BTEQ例子

Bteq.pl 在Perl中调用简单的bteq命令



在Perl + bteq中使用goto语句的例子



一个比较完整的Perl + bteq例子



### 4.4 配置

Windows版本的IceCode.exe







