CODE MAGUS

odbc: Recio ODBC Access Method Version 1

CML00063-01

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
`www.codemagus.com`
Copyright © 2014 by Code Magus Limited
All rights reserved

August 16, 2016

Contents

1   Introduction

The `odbc` access method is a module which implements the `recio` [1] provider interface allowing the `recio` [1] user interface to support reading data from an `ODBC` data source. This access method extends the scope of accessing data via `recio` [1] to any data source that can be defined by an `ODBC` data source name and should be used in conjunction with the `odbcdcl` access method [2] which generates meta data describing the result set that is returned.

2   Open String Specification

As with all `recio` [1] library open specification strings, three components comprise the open string: access method, object, and options name-value pairs.

2.1   Open Specification Access Method Name

The access method name should be specified as `odbc`.

2.2   Open Specification Object Name

The object name is an SQL query statement that resolves to a result set. This is usually a select statement.

2.3   Open Specification Option Name-Value Pairs

Consult the access method definition file for the option name-value pairs supported by the `text` access method. The access method definition file also supplies details of the default values (if any) of the options.

2.4   Open String Specification Examples

2.4.1   Introduction

The following examples and `ODBC` configuration files were run against and adhere to these versions of the respective databases:

- MySQL.

```
hayward@giraffe:odbcam>mysql -V
mysql  Ver 14.14 Distrib 5.1.42, for suse-linux-gnu (i686) using readline 6.0
```

- UDB DB2.

```
hayward@giraffe:odbcam>db2
 (c) Copyright IBM Corporation 1993,2007
Command Line Processor for DB2 Client 9.7.0
```

- Oracle.

```
hayward@giraffe:odbcam>sqlplus -v

SQL*Plus: Release 10.2.0.1.0 - Production
```

2.4.2 MySQL example

The following open string specification could be used to read the list of tables defined in a MySQL database using the data source name of `MySQL`

```
"odbc([select table_schema,table_name from information_schema.tables],
     source=MySQL,user=myuser,password=somepassword)"
```

as in the following cmlcopy (where the user and password are defaulted to the logged on user):

```
hayward@giraffe:odbcam>cmlcopy -i "odbc([select table_schema,table_name
     from information_schema.tables],source=MySQL)"
     -o "binary(sqlout.dat,mode=wb,recfm=v)"
Code Magus Limited Filetools V2.0: build 2010-03-17-16.36.58
[cmlcopy] $Id: odbcam.tex,v 1.8 2013/01/23 08:12:02 hayward Exp $
Copyright (c) 2001, 2002 by Stephen Donaldson. All rights reserved.
Copyright (c) 2003--2009 by Code Magus Limited. All rights reserved.
[Contact: stephen@codemagus.com].
odbc([select table_schema,tab ... : Input Records = 29.
binary(sqlout.dat,mode=wb,recfm=v): Output Records = 29.
```

On Unix an example of the user `ODBC` configuration file and the system `ODBC` installation configuration file that are required to support data access from MySQL using the data source name (DSN) `MySQL` is as follows:

˜/.odbc.ini

```
[MySQL]
Description = MySQLTest
Driver = MySQL
Server = localhost
Database = sample
Port =
Socket =
Option =
Stmt =
```

/etc/unixODBC/odbcinst.ini

```
[MySQL]
Description     = MySQL driver using unixODBC MySQL setup library
Driver          = //usr/lib/unixODBC/libmyodbc3.so
Driver64        = /usr/lib
Setup           = /usr/lib/unixODBC/libodbcmyS.so
Setup64         = /usr/lib
UsageCount      = 1
CPTimeout       =
```

2.4.3   UDB DB2 example

Similarly the following open string specification will read the list of tables defined in a UDB DB2 database

```
"odbc([select creator,name from sysibm.systables],source=SAMPLE)"
```

as in the following cmlcopy:

```
hayward@giraffe:odbcam>cmlcopy -i "odbc([select creator,name from
      sysibm.systables],source=SAMPLE)"
      -o "binary(sqlout.dat,mode=wb,recfm=v)"
Code Magus Limited Filetools V2.0: build 2010-03-17-16.36.58
[cmlcopy] $Id: odbcam.tex,v 1.8 2013/01/23 08:12:02 hayward Exp $
Copyright (c) 2001, 2002 by Stephen Donaldson. All rights reserved.
Copyright (c) 2003--2009 by Code Magus Limited. All rights reserved.
[Contact: stephen@codemagus.com].
odbc([select creator,name from... : Input Records = 434.
binary(sqlout.dat,mode=wb,recfm=v): Output Records = 434.
```

On Unix an example of the user ODBC configuration file and the system ODBC installation configuration file that are required to support data access from UDB DB2 using the data source name (DSN) SAMPLE is as follows:

˜/.odbc.ini

```
[SAMPLE]
Description = UDB SAMPLE database
Driver = DB2
fileusage = 1
dontdlclose = 1
```

/etc/unixODBC/odbcinst.ini

```
[DB2]
Description     = DB2 ODBC driver
Driver          = /opt/ibm/db2/V9.7/lib32/libdb2.so
Driver64        = /usr/lib
Setup           = /usr/lib
Setup64         = /usr/lib
UsageCount      = 1
CPTimeout       =
```

```
CPReuse         =
```

2.4.4   Oracle example

And Lastly the following open string specification will read the list of tables defined in an Oracle database

```
"odbc([select owner,table_name from all_all_tables],
      source=OracleODBC10g,user=system,password=sys)"
```

as in the following cmlcopy:

```
hayward@giraffe:odbcam>cmlcopy -i "odbc([select owner,table_name
      from all_all_tables],source=OracleODBC10g,user=system,
      password=sys)" -o "binary(sqlout.dat,mode=wb,recfm=v)"
Code Magus Limited Filetools V2.0: build 2010-03-17-16.36.58
[cmlcopy] $Id: odbcam.tex,v 1.8 2013/01/23 08:12:02 hayward Exp $
Copyright (c) 2001, 2002 by Stephen Donaldson. All rights reserved.
Copyright (c) 2003--2009 by Code Magus Limited. All rights reserved.
[Contact: stephen@codemagus.com].
odbc([select owner,table_name ... : Input Records = 1125.
binary(sqlout.dat,mode=wb,recfm=v): Output Records = 1125.
```

On Unix an example of the user `ODBC` configuration file and the system `ODBC` installation configuration file that are required to support data access from Oracle using the data source name (DSN) `OracleODBC10g` is as follows:

˜/.odbc.ini

```
[OracleODBC10g]
Application Attributes = T
Attributes = W
BatchAutocommitMode = IfAllSuccessful
CloseCursor = F
DisableDPM = F
DisableMTS = T
Driver = Oracle 10g ODBC driver
DSN = OracleODBC-10g
EXECSchemaOpt =
EXECSyntax = T
Failover = T
FailoverDelay = 10
FailoverRetryCount = 10
FetchBufferSize = 64000
ForceWCHAR = F
Lobs = T
Longs = T
MetadataIdDefault = F
QueryTimeout = T
ResultSets = T
ServerName =
SQLGetData extensions = F
```

```
Translation DLL =
Translation Option = 0
UserID =
```

/etc/unixODBC/odbcinst.ini

```
[Oracle 10g ODBC driver]
Description     = Oracle ODBC driver for Oracle 10g
Driver          = /usr/lib/oracle/xe/app/oracle/product/10.2.0/
                                  server/lib/libsqora.so.10.1
Setup           =
FileUsage       =
CPTimeout       =
CPReuse         =
```

3   `odbc` access method definition file

The access method definition file should be consulted for the description of the options and their default values. This includes the description of the options. The access method definition file should also be consulted for the processing modes supported by the access method.

Refer to the `recio` library documentation for interpreting the contents of the access method definition file.

```
access odbc(source,user="NA",password="NA",null_indicator="?",nulls="after",
        init_image=" ",meta_data="NO");

  -- File: ODBC.amd
  --
  -- This file contains an access method definition which is used to read
  -- records from a data source via ODBC.
  --
  -- Author: Code Magus Ltd [codemagus.codemagus.com].
  --
  -- Copyright (c) 2010 Code Magus Limited. All rights reserved.

  -- $Author: hayward $
  -- $Date: 2015/09/24 20:38:14 $
  -- $Id: ODBC.amd,v 1.6 2015/09/24 20:38:14 hayward Exp $
  -- $Name:  $
  -- $Revision: 1.6 $
  -- $State: Exp $
  --
  -- $Log: ODBC.amd,v $
  -- Revision 1.6  2015/09/24 20:38:14  hayward
  -- Allow the access method to use
  -- SQLDriverConnect() to connect.
  -- To do this the user codes the source name
```

```
-- as "dc:<driver parms>". This enables the use
-- of DSNless connections where the Data Source
-- does not need to be defined in odbc.ini.
-- For example for SQLite, instead of defining
-- a data source for every database file on a
-- system, use a source that looks like this:
-- "dc:Database=/path/to/database.sqlite3;Driver=SQLite"
-- where only the driver has to be pre-registered.
--
-- Revision 1.5  2013/01/22 11:29:38  hayward
-- Add \ to set of valid characters for the user
-- as it is required when connecting to SQLServer
-- using a domain user of the form domain\user.
--
-- Revision 1.4  2010/03/12 13:07:21  hayward
-- Fix typo: missing semi-colon.
--
-- Revision 1.3  2010/03/10 11:56:20  hayward
-- Separate the data and meta-data access into
-- two distinct access methods. They are "odbc"
-- for data access and "odbcdcl" for meta-data
-- access.
--
-- Revision 1.2  2010/03/02 15:45:41  hayward
-- Correct descriptions.
--
-- Revision 1.1.1.1  2010/03/02 14:11:33  hayward
-- Initial takeon of ODBCAM.
--

 modes seq_input;

 implements open;
 implements close;
 implements read;

 describe source as
    "source is the ODBC data source name to connect to as defined in "
    "the ODBC configuration ODBC.ini. If the source name starts with "
    "DC: (lower or upper case) then SQLDriverConnect() is used and "
    "source holds the connection parameters. This may cause the source "
    "DSN to be bypassed. Consult the documentation of the driver being "
    "used for information on the parameters and values that may be set."
    ;

 describe user as
    "user is the the user ID with which to connect to the ODBC data source."
    ;

 describe password as
    "password is the the authentication string for specified user."
    ;
```

```
describe null_indicator as
    "null_indicator refers to the value the NULL indicator is set to when a "
    "column in a result set is NULL.";

describe nulls as
    "nulls refers to a single byte indicator for each DB2 NULLABLE "
    "column that is embedded in the result row either before or after "
    "the data for that column.";

describe init_image as
    "init_image refers to the value the bytes of each new row are "
    "initialised to before being fetched. This mostly affects VARCHAR "
    "type columns where trailing bytes of the column will be set to this "
    "value";

describe meta_data as
    "meta_data is set to the value NO, which causes the access method to "
    "only return the result set associated with the SQL query given."
    ;

constrain source as "^\(\(([a-zA-Z0-9_]\+\)\)\|\([dD][cC]:.*\)\)$";
constrain user as "^[a-zA-Z0-9_\\]\+$";
constrain password as "^[a-zA-Z0-9_]\+$";
constrain null_indicator as "^[A-Za-z!$%&*+#?]$";
constrain nulls as "^\(before\|after\)$";
constrain init_image as "^\(.\|0[xX][0-9a-fA-F][0-9a-fA-F]\)$";
constrain meta_data as "^\([Nn][Oo]\)$";

path = ${CODEMAGUS_AMDLIBS} "%s";
module = "odbcam" ${CODEMAGUS_AMDSUFDL};
entry = odbcam_init;

end.
```

4   Environment

The location and format of the access method definition file is required to be specified by the environment variable CODEMAGUS_AMDPATH. This environment variable supplies a pattern to the full path of where access method definition (or amd) files are located. The format of the environment variable is that of a path with a %s appearing in the position in which the access method member name should appear. For example, on MVS systems this might have the form:

```
CODEMAGUS_AMDPATH='DNCT00.SRDA1.AMDFILES(%s)'
```

On a Unix-based system, the value might be set in a shell profile file such as:

```
export CODEMAGUS_AMDPATH=$HOME/bin/%s.amd
```

On Windows systems, the value might be supplied from the environment variables and look something like:

```
C:\CodeMagus\bin\%s.amd
```

References

[1] recio: Record Stream I/O Library Version 1. CML Document CML00001-01, Code Magus Limited, July 2008. PDF.

[2] odbcdcl: Recio ODBC DCL Generator Access Method Version 1. CML Document CML00064-01, Code Magus Limited, February 2010. PDF.