



db2query: Recio DB2 Query Access Method
Version 1

CML00050-01

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	2
1.1	Points to note	2
2	Open String Specification	2
2.1	Open Specification Access Method Name	2
2.2	Open Specification Object Name	2
2.3	Open Specification Option Name-Value Pairs	3
2.4	Open String Specification Examples	3
3	db2query access method definition file	3
4	Environment	5
5	Installation on z/OS	6
6	Installation on Linux, Unix and Windows	6

1 Introduction

The `db2query` access method is a module which implements the `recio` [1] provider interface allowing the `recio` user interface to support DB2 SQL queries. This includes UDB on Linux, UNIX and Windows, and DB2 for z/OS.

The `db2query` access method allows any select statement that does not include parameter markers to be executed by the named DB2 instance or subsystem.

If there is a requirement to map the returned data then this access method can be used in conjunction with the DB2 `DCLGEN` access method [?].

1.1 Points to note

- If a column in the selected result set is defined as a `VARCHAR` type then the column data will occupy the maximum length of the column and will be preceded by a short integer (2 bytes) that holds the length of the data in the column.
- Columns that are nullable will also have a single byte indicator either preceding or following the column data in the record. If the column is `NULL` then this indicator is set and the actual value of the column is undefined. The indicator value and its position can be set via options; see section 3 on page 3.
- All records returned from the result set will have the same length. That is, the record length is calculated by summing the maximum length of all columns and adding one for each column that is nullable.

2 Open String Specification

As with all `recio` 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 `db2query`.

2.2 Open Specification Object Name

The object name must be a DB2 select statement without any parameter markers.

2.3 Open Specification Option Name-Value Pairs

Consult the access method definition file for the option name-value pairs supported by the `db2query` access method. The access method definition file also supplies details of the default values (if any) of the options; see section 3 on page 3.

2.4 Open String Specification Examples

The following open string specification could be used to list the names of all tables on a DB2 instance on Linux.

```
db2query([select name from sysibm.syscolumns],subsys=sample,
        user=another,password=anpwd)
```

Similarly, The following open string specification could be used to list the names of all tables on a DB2 subsystem on z/OS. Note that in this case only the plan name is required for authentication as access to the database is controlled through the security server and access must be granted via the database to the plan.

```
db2query([select name from sysibm.syscolumns],subsys=DSN1,
        plan=APLAN)
```

3 db2query access method definition file

The access method definition file should be consulted for the description of the options and their default values. 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 db2query(subsys,user="NA",password="NA",plan="DB2QUERY",
        nulls="after",null_indicator="?",init_image=" ");

-- File: DB2QUERY.amd
--
-- This file contains an access method definition which is used to prepare
-- a select statement for IBM's DB2 database and read the resultant rows as
-- records.
--
-- Author: Stephen R. Donaldson [www.codemagus.com].
--
-- Copyright (c) 2009 Code Magus Limited. All rights reserved.

-- $Author: hayward $
-- $Date: 2009/11/25 06:44:22 $
```

3 DB2QUERY ACCESS METHOD DEFINITION FILE

```
-- $Id: DB2QUERY.amd,v 1.5 2009/11/25 06:44:22 hayward Exp $
-- $Name: $
-- $Revision: 1.5 $
-- $State: Exp $
--
-- $Log: DB2QUERY.amd,v $
-- Revision 1.5 2009/11/25 06:44:22 hayward
-- Make default plan more specific.
--
-- Revision 1.4 2009/11/23 22:36:08 hayward
-- Changes required for port to MVS.
--
-- Revision 1.3 2009/10/21 20:14:39 hayward
-- Alloc user to set the initial byte image for the buffer
-- into which the DB2 row is read. This allows a VARCHAR
-- to have this value as the trailing byte image.
--
-- Revision 1.2 2009/09/30 17:33:41 hayward
-- Add new options to the access method definition
-- file to handle where to situate the NULL indicator and
-- what value it should have when a column is NULL.
--
-- Revision 1.1.1.1 2009/09/28 15:27:30 hayward
-- Initial take on of the DB2 recio query access method.
--
--
modes seq_input;

implements open;
implements close;
implements read;

describe subsys as
    "The subsys defines which database instance to connect to for executing "
    "the select statement.";

describe user as
    "The user is the DB2 user under wich the SQL will be executed. It is "
    "optional as some DB2 instances use the logged on user as the "
    "authorization ID.";

describe password as
    "The password is the DB2 password under wich the SQL will be executed. It is "
    "optional as some DB2 instances use the logged on user as the "
    "authorization ID.";

describe plan as
    "The plan is the MVS (z/OS) DB2 plan name under which the request "
    "should run. It is only applicabile to running a query on MVS.";

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 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 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";

constrain subsys as "^.*$";
constrain user as "^.*$";
constrain password as "^.*$";
constrain plan as "^.*$";
constrain nulls as "^\(before\|after\) $" ;
constrain null_indicator as "^[A-Za-z!$%&*+#?]" ;
constrain init_image as "^\. \(|0[xX] [0-9] [0-9] \)" ;

path = ${CODEMAGUS_AMDLIBS} "%s";
module = "db2queryam" ${CODEMAGUS_AMDSUFDL};
entry = DB2QINIT;

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 z/OS 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
```

5 Installation on z/OS

Once the package has been unpacked, in order to use the `db2query` access method on z/OS the following steps must be performed:

1. Upload the DBRM to z/OS.

Copy the DB2 for z/OS DBRM `lib/db2queryam.mvs.dbrm` to a PDS. The PDS is usually defined as fixed block with a record length of 80 and block size of 3120. It can be a PDSE.

2. Bind the DBRM.

Customise and submit the JCL `bin/db2queryam.mvs.bind` in order to bind the plan on the DB2 subsystem.

3. Grant the user that will execute the query access to the plan. For example the following DB2 command grants execute permission on the plan to the user CMLUSER:

```
grant execute on plan DB2QUERY to CMLUSER;
```

4. In order to access DB2 via the DB2 RRSF attachment facility in an OMVS controlled environment the security setting for the user under which the server runs needs to be changed. Read access must be granted to the facility `BPX.DAEMON.HFSCTL` for the user under which the server runs; This allows a controlled program to call a (non controlled) program that resides in a PDS or PDSE. The following RACF commands will allow the the server running under the user `STARTED` to call DB2:

```
RDEFINE FACILITY BPX.DAEMON.HFSCTL UACC(NONE)
```

```
PERMIT BPX.DAEMON.HFSCTL CLASS(FACILITY) ID(STARTED) ACCESS(READ)  
SETROPTS RACLIST(FACILITY) REFRESH
```

5. The user executing the query must be granted access to the DB2 sub system. Only read access is required. The following command shows granting the user CMLUSER access to the sub system `DSN1`.

```
PERMIT DSN1.RRSF ACCESS(READ) CLASS(DSNR) ID(CMLUSER)  
SETROPTS RACLIST(DSNR) REFRESH
```

6 Installation on Linux, Unix and Windows

Once the package has been unpacked, in order to use the `db2query` access method on Linux, Unix and Windows the following steps must be performed:

1. Bind the DBRM.

Using the `db2` command bind the plan with the following command:

```
connect to mysubsystem
```

```
bind lib/db2queryam.bnd
```

The name `mysubsystem` should be changed to the name of the subsystem on which to bind the plan; A user ID and password may also be required. See the DB2 documentation of the `bind` command for other parameters that may be relevant to the DB2 instance.

References

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