CODE MAGUS

MVS: MVS Record Stream Access Method Version 1

CML00016-01

Business Partner IBM

August 16, 2016

Contents

1 Introduction

The `MVS` access method is a module which implements the `recio` [1] provider interface allowing the `recio` user interface to support fixed and variable length record files implemented on MVS systems.

2 Open String Specification

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

2.1 Open Specification Access Method Name

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

2.2 Open Specification Object Name

The object name defines a file on an `MVS` system using a standard `JCL DD` statement without the `DD` name portion. The access method will determine the `DD` name, allocate the file and call the underlying access method.

2.3 Open Specification Option Name-Value Pairs

Consult the access method definition file for the option name-value pairs supported by the `MVS` 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

The commands below may be split over several lines; this is only for readability within this document. Normally all open string specifications are one continuous string.

Note also that in some instances the 'with' option's value is enclosed in brackets. This enables a value to have special characters (for example the delimiter comma or another equal sign), but still be recognised as a single token. For more information see the `Recio User Guide: recio.pdf`.

The following open string specification could be used to open a file as a text record delimited file.

```
"mvs([DSN=CODEMAGUS.A1.SOURCE(PROGRAMA),DISP=SHR],
    using=text,with=[mode=r])"
```

The following open string specification could be used to open a file as a fixed length record file.

```
"mvs([DSN=CODEMAGUS.A1.DATAFILE,DISP=SHR],
    using=binary,
    with=[mode=[rb,type=record],recfm=f,reclen=256,type=record])"
```

The following open string specification could be used to open a file as a variable length record file.

```
"mvs([DSN=CODEMAGUS.A2.DATAFILE,DISP=SHR],
    using=binary,
    with=[mode=[rb,type=record],recfm=v,type=record])"
```

The following open string specification could be used to open a file as a variable length record file for output. At the same time the file will be allocated.

```
"mvs([DSN=CODEMAGUS.RECIOTST.BINARY.FIXED,DISP=(NEW,CATLG,DELETE),
    SPACE=(TRK,(5,1)),RECFM=FB,LRECL=52,BLKSIZE=5200,DSORG=PS,
    VOL=SER=USER01],
    using=binary,
    with=[mode=[wb,type=record],recfm=fb,reclen=52,type=record])"
```

3   **MVS** 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 constraining values 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 mvs(using="binary",with);

  -- File: MVS.amd
  --
  -- This file contains an access method definition which is used to read and
  -- write MVS files using standard MVS JCL options to allocate the file. The
  -- MVS access method uses the standard JCL DD statement (without the DD name
  -- and the keyword DD) to allocate a file on an MVS system. Once allocated the
  -- underlying access method identified in the 'using' option is invoked with
  -- any options required supplied in the 'with' option as comma separated
  -- keyword=values pairs.
  -- All valid DD statement parameters may be used; this allows a user to have
  -- fine control over the dataset allocation, including allocating new files,
  -- deleting files etc.
  --
  -- Author: Patrick Hayward [hayward@codemagus.com].
```

```
--
-- Copyright (c) 2008 Code Magus Limited. All rights reserved.

-- $Author: hayward $
-- $Date: 2008/07/15 11:41:42 $
-- $Id: MVS.amd,v 1.4 2008/07/15 11:41:42 hayward Exp $
-- $Name:  $
-- $Revision: 1.4 $
-- $State: Exp $
--
-- $Log: MVS.amd,v $
-- Revision 1.4  2008/07/15 11:41:42  hayward
-- Update documentation in support of the new finalised options.
--
-- Revision 1.3  2008/06/04 13:54:32  hayward
-- Changes required for new options "using" and "with".
-- Removed option "mode".
--
-- Revision 1.2  2008/05/14 19:13:52  hayward
-- Remove recfm and reclen.
-- They are generated programmatically if needed.
--
-- Revision 1.1.1.1  2008/05/07 11:00:09  hayward
-- Import MVS access method files.
--
--

 modes seq_input, seq_output, skip_input;

 implements open;
 implements close;
 implements read;
 implements write;
 implements point;
 implements tell;

 describe using as
    "The 'using' option defines the underlying access method to be used for "
    "data access. After the MVS access method allocates the file all "
    "requests are passed on to this access method.";

 describe with as
    "The 'with' option is used to specify all the access method options "
    "(as comma separated keyword=value pairs) of the underlying access "
    "method specified in the 'using' option. Refer to the access method "
    "definition file of that access method for information about the "
    "options that are valid for that access method.";

 constrain using as ".*";
 constrain with as ".*";

 path = ${CODEMAGUS_AMDLIBS} "%s";
```

```
   module = "mvsam" ${CODEMAGUS_AMDSUFDL};
   entry = mvsam_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.