SMF: Recio SMF Access Method Version 1

CML00111-01

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com

December 15, 2020

Contents

1 Introduction

The SMF access method is a module which implements the recio [1] provider interface allowing the recio user interface to support reading SMF formatted binary records and selecting various record types and sub-types.

The default is to read all records using the underlying access method. The underlying access method is usually binary. On MVS the records are variable length spanned records and on Linux variable length RDW records.

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.

For the SMF access method the access method name is SMF.

2.1 Open Specification Access Method Name

The access method name should be specified as SMF.

2.2 Open Specification Object Name

The object name should be a suitable file stream name on the local system. This is a stream file name appropriate to the local system's fopen file name parameter. If the file is on a non-MVS system, it should have been downloaded with ftp using binary mode and the setting locsite rdw.

2.3 Open Specification Option Name-Value Pairs

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

In the following examples the lines have been split with a '\' in order to fit the width of the page, but should be read as one continuous string with no spaces.

The following open string specification could be used to copy an SMF file downloaded from MVS to Linux.

```
cmlcopy -i "smf(MVS.DAILY.SMF.sqfile,with=[mode=rb,recfm=v])" \
   -o "binary(MVS.DAILY.SMF.rdw,mode=wb,recfm=v)"
```

The following open string specification could be used to copy an SMF file downloaded from MVS to Linux, but to only select the SMF type 30 records.

```
cmlcopy -i "smf(MVS.DAILY.SMF.sqfile,with=[mode=rb,recfm=v],types=30)"
   -o "binary(MVS.DAILY.SMF.rdw,mode=wb,recfm=v)"
```

The following open string specification could be used to copy an SMF file downloaded from MVS to Linux, but to only select the SMF type 30 records, but only the subtypes 1,2,3 and 5 as well as type 70 records.

```
cmlcopy -i "smf(MVS.DAILY.SMF.sqfile,with=[mode=rb,recfm=v],\
                types=[30:1,2,3,5;70])" \
   -o "binary(MVS.DAILY.SMF.rdw,mode=wb,recfm=v)"
```

The following example demonstrates how to read a remote file from MVS and write the output to the local file system on Linux.

```
cmlcopy \
 -i "remote([smf(SMF.DATASET,using=binary,\
     with=[mode=[rb,recfm=VBS,type=record],recfm=v,type=record])],\
     host=MVS,user=uname,password=pwd)" \
 -o "binary(SMF_DATASET.rdw,mode=wb,recfm=v)"
```

3 SMF 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 smf(types="0",using="binary",bias="0",
           with="mode=[rb,recfm=VBS,type=record],recfm=v,type=record");

  -- File: SMF.amd
  --
  -- This file contains an access method definition which is used to read
  -- SMF data (using the underlying access method specified in using=) and only
  -- return the requested SMF tpyes and sub-types. NOTE that this access method
  -- returns only the data read and not the RDW. If you are using standard
  -- mapping meta data, you may have to offset the data by the length of the RDW
  -- defined in smfxxhdr (the two fields smfxxlen and smfxxseg).
  --
  -- Author: Code magus Limited [www.codemagus.com].
  --
  -- Copyright (c) 2012 Code Magus Limited. All rights reserved.

  -- $Author: hayward $
  -- $Date: 2016/08/17 15:30:45 $
  -- $Id: SMF.amd,v 1.2 2016/08/17 15:30:45 hayward Exp $
  -- $Name:  $
  -- $Revision: 1.2 $
  -- $State: Exp $
  --
  -- $Log: SMF.amd,v $
  -- Revision 1.2  2016/08/17 15:30:45  hayward
  -- Fix bugs related to checking the SMF subtypes.
  -- Add a bias to help map the record when it is
  -- right shifted or prepended with other data.
  --
  -- Revision 1.1.1.1  2012/08/02 11:30:56  hayward
  -- Import sources to CVS.
  --

   modes seq_input;

   implements open;
   implements close;
   implements read;

   describe types as
      "The option ’types’ specifies the types and sub types of the records "
      "to return to the caller. Specify types as a semi-colon separated list. "
      "Sub-types may be specified as a comma separated list "
      "immediately following the type and a colon. For example "
      "types=[30:1,2,3,5;70] specifies type 30 subtypes 1,2,3,5 only and "
      "all type 70 records. A single type of 0 (the default) implies "
      "selection of all types."
      ;

   describe bias as
      "The option ’bias’ is used if the actual start of the SMF record "
      "is not at offset 0. Set the ’bias’ to the offset of where the data "
      "actually starts in the record. This is usually at the same offset "
      "as the field smfxxflg in a standard SMF meta data structure. "
      "Note that this access method never returns the RDW field made up of "
      "smfxxlen and smfxxseg."
      ;

   describe using as
      "The option ’using’ specifies the underlying access method that will "
      "be used to read the data. If specified then the object list of the "
```

```
      "Recio open string must only contain object names, otherwise the list "
      "must contain complete Recio open specifications"
      ;

   describe with as
      "The option 'with' is used only with the option 'using' and specifies "
      "the Recio options for the access method named in the 'using' option. "
      "If no options are to be used specify 'NA'."
      ;

   constrain types as "^\([0-9]\+\(:[0-9]\(,[0-9]\+\)*\)\?"
                      "\(;[0-9]\+\(:[0-9]\(,[0-9]\+\)*\)\?\)*\)$";
   constrain bias as "^[0-9]*$";
   constrain using as "^.*$";
   constrain with as "^\(NA\)\|\(.*\)$";

   path = ${CODEMAGUS_AMDLIBS} "%s";
   module = "smfam" ${CODEMAGUS_AMDSUFDL};
   entry = smfam_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.