text: File Access Method Using POSIX Streams Version 1

CML00031-01

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

August 16, 2016

Contents

1 Introduction

The `text` access method is a module which implements the `recio` [1] provider interface allowing the `recio` user interface to support delimited record text files implemented on byte-stream file systems and on MVS for record based files. This includes VOS stream files, UNIX text files, Windows/DOS text files and z/OS classic MVS and HFS files. With the exception of MVS classic files (which are record based files) all the other types (i.e. stream based files) may be read or written on any of the supported platforms.

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 `text` access method the access method name is `text`.

2.1 Open Specification Access Method Name

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

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 systems `fopen` file name parameter.

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

The following open string specification could be used to open an input file, as a locally formatted text file, for reading.

```
text(/tmp/in,mode=r)
```

The following open string specification could be used to open an output file, as a locally formatted text file, for reading.

```
text(/tmp/out,mode=w)
```

The following open string specification could be used to open an input file, as a UNIX formatted text file, for reading.

```
text(/tmp/out,mode=r,texttype=UNIX)
```

The following open string specification could be used to open an output file, as an MVS formatted text file (using record IO), for writing. Note that this format only makes sense when the `text` access method actually runs on MVS, as there are no stream record delimiters.

```
text(/tmp/out,mode=[w,type=record],texttype=MVS)
```

3 `text` 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 text(mode,texttype="LOCAL",delimiter="NA");

  -- File: TEXT.amd
  --
  -- This file contains an access method definition which is used to read
  -- and write local files of as a text stream. Support is provided for
  -- various different text types including DOS, UNIX and MVS (or USS).
  -- Any text type file can be read or written on any of the platforms
  -- and if not specified it is defaulted to LOCAL (ie the platform the
  -- Access Method is running on).
  --
  --
  -- Author: Stephen R. Donaldson [stephen@codemagus.com].
  --
  -- Copyright (c) 2008 Code Magus Limited. All rights reserved.

  -- $Author: hayward $
  -- $Date: 2014/09/24 09:06:25 $
  -- $Id: TEXT.amd,v 1.11 2014/09/24 09:06:25 hayward Exp $
  -- $Name:  $
  -- $Revision: 1.11 $
  -- $State: Exp $
```

```
--
-- $Log: TEXT.amd,v $
-- Revision 1.11  2014/09/24 09:06:25  hayward
-- Add the ability to handle any delimiter
-- specified as a multiple of two hex digits.
--
-- Revision 1.10  2012/09/11 11:36:46  hayward
-- Make mode 'b' independant of ',type=record'
--
-- Revision 1.9  2012/08/28 18:00:10  hayward
-- Add the ability to add 'b' to open mode
-- when using 'type=record' as this is required
-- on record based systems like z390 in order to
-- write non local platform line endings.
--
-- Revision 1.8  2009/11/10 11:15:56  hayward
-- Allow text to use SKIP_INPUT and point method.
--
-- Revision 1.7  2009/05/27 08:52:39  hayward
-- Correct documentation.
--
-- Revision 1.6  2009/03/12 11:50:55  hayward
-- After testing on MVS and USS and reading the C runtime manual for
-- fwrite() and fread() it transpires that MVS and USS can be handled
-- in the same way. This requires using 0x15 as the line delimiter and
-- opening the file without "b,type=record". On USS the 0x15 is written
-- to the file (in the same manner as 0x0a on Unix) and on MVS it is
-- stripped at write time and a record boundary is inserted; whereas
-- on read a record boundary is replaced with a trailing 0x15.
-- This is the best outcome because now we can generate any platform
-- based file on any platform (not including ASCII/EBCDIC issues though).
--
-- Revision 1.5  2009/03/10 08:42:36  hayward
-- Major change to add texttype parameter to the AMD and how we read/write
-- text files.
--
-- Revision 1.4  2008/04/22 17:32:15  hayward
-- Add ability for callers to append to files by
-- adding 'a' to the mode constraint.
--
-- Revision 1.3  2008/04/09 13:57:00  hayward
-- Fix path statement.
--
-- Revision 1.2  2008/03/31 22:30:45  stephen
-- Add usage of environment variables to AMD file
--
-- Revision 1.1  2008/03/20 14:56:25  stephen
-- Add new contents of recio text access method
--

 modes seq_input, seq_output, skip_input;
```

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

   describe mode as
      "The mode is the open mode string which will be passed to the C Standard "
      "I/O Library.";

   describe texttype as
      "The texttype parameter determines the line end delimiter for the text "
      "being written. Either a common line end delimiter may be chosen, or "
      "if required a custom line end can be specified.";

   describe delimiter as
      "The end of line delimiter is only specified when texttype=custom is used. "
      "It must be a valid hex representation (even length of only [0-9a-fA-F]) "
      "of the text line delimiter in the data being read or written.";

   constrain mode as "^[rwa]b\?\(,type=record\)\?$";
   constrain texttype as "^\(LOCAL\|DOS\|UNIX\|MVS\|CUSTOM\)$";
   constrain delimiter as "^\(NA\)\|\(\([0-9a-fA-F]\{2\}\)\+\)$";

   path = ${CODEMAGUS_AMDLIBS} "%s";
   module = "textam" ${CODEMAGUS_AMDSUFDL};
   entry = textam_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.