CodeMagus

---

circular: Circular Log Access Method Version 1

CML00089-01

CodeMagus

December 15, 2020

Contents

1 Introduction

The `circular` access method is a module which implements the `recio` [1] provider interface allowing the `recio` user interface to support writing to and reading from a circular log file. A circular log file is a normal file system file with a size limited to a specified number of bytes, used for recording the latest written data. This is done by wrapping the data once the physical end of file has been reached. In other words once the size limit amount of data has been written the access method will start to write data from the physical start of the file again, overwriting the oldest data first. This type of circular log means that the logical start and end of the data actually appear at any point in the file defined by the position in the file where the next record would be written.

Results are undefined if a reader of a circular log reads a log while a writer of the same log has it open, especially if the writer writes records during this time.

The mode, write or read, is determined by the recio mode specified by the calling software.

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 `circular` access method the access method name is `circular`.

2.1 Open Specification Access Method Name

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

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 `circular` 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 a circular log in order to write records to it (the size is set to a defaulted value by the CIRCULAR.amd):

```
circular(/tmp/log.ccl)
```

The following open string specification could be used to open a circular log in order to write records to it and limiting it to 4MB:

```
circular(/tmp/log.ccl,size=4000000)
```

The first example above can also be used to read from a circular log if the calling software opens the log with a recio read mode. For example `cmlcopy` will open the log for read in the following example (and write the data to standard out):

```
cmlcopy -i "circular(/tmp/log.ccl)" -o "standard(out)"
```

In the following example `cmlcopy` will write the contents of the file `input.txt` to a circular log.

```
cmlcopy -i "text(input.txt,mode=r)" -o "circular(/tmp/log.ccl,size=1000
```

3   `circular` 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 circular(size=10240,overwrite="N");

  -- File: CIRCULAR.amd
  --
  -- This file contains an access method definition which enables recio write
  -- and read access to a circular log. This log is limited in size and records
  -- are written to it in a circular fashion. In other words it will hold only
  -- the last n number of records that fit into the size and as new records are
  -- written so the oldest will fall off the log.
  --
  -- Author: Patrick Hayward [hayward@codemagus.com].
  --
  -- Copyright (c) 2012 Code Magus Limited. All rights reserved.

  -- $Author: francois $
  -- $Date: 2018/06/20 07:51:23 $
```

```
-- $Id: CIRCULAR.amd,v 1.4 2018/06/20 07:51:23 francois Exp $
-- $Name:  $
-- $Revision: 1.4 $
-- $State: Exp $
--
-- $Log: CIRCULAR.amd,v $
-- Revision 1.4  2018/06/20 07:51:23  francois
-- Hard paths removed
--
-- Revision 1.1  2018/06/15 10:10:42  Francois
-- *** empty log message ***
--
-- Revision 1.3  2014/10/23 11:17:01  hayward
-- Improve description of overwrite.
--
-- Revision 1.2  2013/06/25 21:39:44  hayward
-- Improve descriptions.
--
-- Revision 1.1.1.1  2013/06/20 15:02:08  hayward
-- Rename blackboxam to circularam
-- and import new sources in to CVS.
--
-- Revision 1.2  2012/12/04 15:33:14  hayward
-- Add overwrite parameter and
-- fix open error causing caller
-- to think stream is open when
-- it has not been.
--
-- Revision 1.1.1.1  2012/12/03 13:27:33  hayward
-- Import sources to CVS
--

 modes seq_output, seq_input;

 implements open;
 implements close;
 implements read;
 implements write;

 describe size as
    "The size parameter specifies the maximum size, in bytes, that the "
    "circular log may grow to when writing. It is ignored for reading."
    ;
 describe overwrite as
    "The overwrite parameter specifies whether to overwrite an output "
    "log or not. If not set when opening an existing log for write, then "
    "subsequent records are written at the logical end of the log, "
    "otherwise the log is started from the physical start again."
    ;

 -- Size must be an integer number.
 constrain size as "^[0-9][0-9]*$";
```

```
    constrain overwrite as "^Y\|N$";

    path = ${CODEMAGUS_AMDLIBS} "%s";
    module = "circularam" ${CODEMAGUS_AMDSUFDL};
    entry = circularam_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.