CODE MAGUS

repeat: Recio Repeat Access Method Version 1

CML00068-01

Business Partner IBM

August 16, 2016

Contents

1 Introduction

The `repeat` access method is a module which implements the `recio` [1] provider interface allowing the `recio` user interface to support reading any other access method in a repeated fashion. This access method is well suited in situations where a never ending stream of records are required as in a system under test where the end of a stream of data could cause a premature end to the test.

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 `repeat`.

2.2 Open Specification Object Name

The object name should be a suitable `recio` open specification string that refers to another access method. This underlying access method is the one from which the data will actually be read.

2.3 Open Specification Option Name-Value Pairs

Consult the access method definition file for the option name-value pairs supported by the `repeat` 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 repeatedly read a text file starting back at the beginning each time an end of file condition is encountered. Note that to fully specify another `recio` open specification as the object of the `repeat` access method it should be enclosed in square brackets.

```
repeat([text(/tmp/in.txt,mode=r)])
```

The following open string specification would perform the same function as the previous example, but would only read the file three times before returning an end of file condition.

```
repeat([text(/tmp/in.txt,mode=r)],iterations=3)
```

The following open string specification performs the same functionality as the first example, but on a binary input file.

```
repeat([binary(/tmp/in.bin,mode=rb,recfm=v)])
```

3 `repeat` 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 repeat(iterations=0);
  --
  -- File: repeat.amd
  --
  -- This file contains an access method definition which is used to read
  -- any other access method repeatedly.
  --
  -- Author: Code magus Limited [www.@codemagus.com].
  --
  -- Copyright (c) 2010 Code Magus Limited. All rights reserved.
  --
  -- $Author: hayward $
  -- $Date: 2010/07/08 16:16:26 $
  -- $Id: REPEAT.amd,v 1.1.1.1 2010/07/08 16:16:26 hayward Exp $
  -- $Name:  $
  -- $Revision: 1.1.1.1 $
  -- $State: Exp $
  --
  -- $Log: REPEAT.amd,v $
  -- Revision 1.1.1.1  2010/07/08 16:16:26  hayward
  -- Add repeatam code and documentation to CVS
  --

  modes seq_input;

  implements open;
  implements close;
  implements read;
```

```
describe iterations as
   "The amount of times to repeat set of read records. "
   "A value of zero means forever and the caller will never "
   "receive an end of file condition.";

-- iterations must be an integer.
constrain iterations as "^[0-9][0-9]*$";

path = ${CODEMAGUS_AMDLIBS} "%s";
module = "repeatam" ${CODEMAGUS_AMDSUFDL};
entry = repeatam_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.