



standard: Standard Input And Output Using Recio
Version 1

CML00030-01

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com
Copyright © 2014 by Code Magus Limited
All rights reserved



August 16, 2016

Contents

1	Introduction	2
2	Open String Specification	2
2.1	Open Specification Access Method Name	2
2.2	Open Specification Object Name	2
2.3	Open Specification Option Name-Value Pairs	2
2.4	Open String Specification Examples	2
3	standard access method definition file	3
3.1	standard input access method definition file	3
4	Environment	4

1 Introduction

The `standard` access method is a module which implements the `recio` [1] provider interface allowing the `recio` user interface to support reading from standard input and writing to standard output and error on byte-stream file systems. This includes under the operating systems VOS, UNIX, Windows/DOS and z/OS (both classic MVS and HFS), all using the Standard C I/O Functions of streams.

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

For the `standard` access method the access method name must be specified as `standard`.

2.2 Open Specification Object Name

The object name must be either “in” for the source standard in, “out” for the destination standard out and “error” for the destination standard error.

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 should be used to read records from standard in.

```
standard(in)
```

The following open string specification should be used to write records to standard out.

```
standard(out)
```

The following open string specification should be used to write records to standard error.

standard(error)

3 standard access method definition file

The access method definition file should be consulted for the description, if any, 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.

3.1 standard input access method definition file

```
access standard(mode="text",reclen=0);

-- File: STANDARD.amd
--
-- This file contains an access method definition which is used to read
-- records from standard in and write to standard out and standard error.
--
-- Author: Patrick Hayward [hayward@codemagus.com].
--
-- Copyright (c) 2008 Code Magus Limited. All rights reserved.

-- $Author: hayward $
-- $Date: 2015/03/11 11:16:49 $
-- $Id: STANDARD.amd,v 1.4 2015/03/11 11:16:49 hayward Exp $
-- $Name: $
-- $Revision: 1.4 $
-- $State: Exp $
--
-- $Log: STANDARD.amd,v $
-- Revision 1.4 2015/03/11 11:16:49 hayward
-- Allow 't' for text mode.
--
-- Revision 1.3 2009/03/17 21:34:57 hayward
-- Fix description typos.
--
-- Revision 1.2 2008/09/19 19:41:10 hayward
-- Allow for binary stdin/out/err. This uses a new option called mode which
-- is defaulted to text so that no changes are necessary for any current
-- users. With mode=binary a further option reclen is also optional. If
-- specified then access is in fixed record chunks and the input/output
-- data size must be a multiple of this value; if not used then data access
-- is done as RDW variable access.
-- This feature allows various Unix tools and other CML tools that use Recio
```

```

-- for input and output to use common Unix techniques of piping data between
-- the stdout of one process and the stdin of another.
--
-- Revision 1.1.1.1 2008/07/28 16:33:09 hayward
-- Import sources into CVS.
--
--

modes seq_input, seq_output;

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

describe mode as
    "The 'mode' option is used to switch between text and binary "
    "data transport. The default (text) always strips or adds a newline "
    "with every record, where as binary mode writes the data as is.";

describe reclen as
    "The 'reclen' option is only used for binary data that contains "
    "fixed length records; if it is not specified then each record is "
    "assumed to be variable length and prefixed with a Record Descriptor "
    "Word (RDW) that determines the length of each record.";

constrain mode as "^\(text\|t\|binary\|b\)\$";
constrain reclen as "[0-9][0-9]*\$";

path = ${CODEMAGUS_AMDLIBS} "%s";
module = "stdam" ${CODEMAGUS_AMDSUFDL};
entry = stdam_init;

end.

```

4 Environment

The location and format of the access method definition file is required to be specified by the environment variable CODEMAGUS_AMPATH. 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_AMPATH='DNCT00.SRDA1.AMDFILES(%s)'
```

On a Unix-based system, the value might be set in a shell profile file such as:

```
export CODEMAGUS_AMPATH=$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](#).