



edit: Recio Edit Access Method Version 1

CML00047-01

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	Access Method Definition File	3
4	Environment	5

1 Introduction

The `edit` access method is a module which implements the `recio` [1] provider interface allowing the `recio` [1] user interface to support editing of input or output records after reading or before writing them using an underlying access method.

An edit process is often used to expand compressed records on reading or to contract records prior to writing them. For example, self defining compressed records are easily processed in this way allowing the application the ability to only process complete uncompressed records.

2 Open String Specification

As with all `recio` [1] 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 `edit`.

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 `edit` 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 will read the file `testfile.dat` using the underlying `binary` [2] access method with the specified attributes and then process it through the edit process defined by the buffer edit configuration file `testexp.bed`.

```
"edit(testfile.dat,bedfile=testexp.bed,using=binary,
with=[mode=rb,recfm=f,reclen=10])"
```

Normally there is a separate buffer edit configuration file for reading and writing; therefore the following open string specification is the opposite to the previous one in that it will process the output record as defined by the buffer edit configuration file `testcon.bed` and then write the record using the underlying binary [2] access method and its specified attributes to the file `testfile.dat`.

```
"edit(testfile.dat,bedfile=testcon.bed,using=binary,
with=[mode=wb,recfm=f,reclen=10])"
```

3 Access Method Definition File

The access method definition file should be consulted for the description of the options, their default values and for the processing modes supported by the access method.

Refer to the `recio` [1] library documentation for interpreting the contents of the access method definition file.

```
access edit(config,using="binary",with="recfm=v",on_error="environ");

-- File: EDIT.amd
--
-- This file contains an access method definition which is used to expand and
-- contract records while reading and writing files respectively. An
-- underlying Access method named in the "using" parameter with the options
-- specified in the "with" parameter is used to actually read and write the
-- physical records. The physical records are expanded and the logical records
-- contracted under the control of a buffer edit (bed) configuration file.
--
-- Author: Code Magus Limited [www.codemagus.com].
--
-- Copyright (c) 2009 Code Magus Limited. All rights reserved.

-- $Author: hayward $
-- $Date: 2014/03/07 09:55:17 $
-- $Id: EDIT.amd,v 1.5 2014/03/07 09:55:17 hayward Exp $
-- $Name: $
-- $Revision: 1.5 $
-- $State: Exp $
--
-- $Log: EDIT.amd,v $
-- Revision 1.5 2014/03/07 09:55:17 hayward
-- Update description for on_error to
-- include information about the error
-- log file and how to specify its name.
--
-- Revision 1.4 2013/04/11 12:13:05 hayward
-- Allow a value for on_error to come
-- from the environment variable. This
-- is now the default and if not set then
```

3 ACCESS METHOD DEFINITION FILE

```
-- the old behaviour of returning an error
-- will be in used.
--
-- Revision 1.3  2013/03/07 22:11:21  hayward
-- Add AM option on_error that allows the user of
-- this AM to choose what action the buffer edit
-- should take when an edit process fails to edit
-- a buffer correctly. Instead of returning an error
-- the user may now return a zero length record
-- (and optionally log the faulty record), return
-- the original record or return an error.
--
-- Revision 1.2  2009/06/22 11:53:09  hayward
-- Rename bufeditam to editam.
--
-- Revision 1.1.1.1  2009/06/18 21:47:19  hayward
-- Import sources to CVS.
--

modes seq_input, seq_output, skip_input;

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

describe config as
    "The 'config' option names the Buffer Edit configuration file that "
    "will control the expansion or contraction of records from or to the "
    "physical record";

describe using as
    "The 'using' option defines the underlying access method to be used for "
    "physical data access.";

describe with as
    "The 'with' option is used to specify all the access method options "
    "(as comma separated keyword=value pairs) of the underlying access "
    "method specified in the 'using' option. Refer to the access method "
    "definition file of that access method for information about the "
    "options that are valid.";

describe on_error as
    "The 'on_error' option identifies what should be done when the edit "
    "process fails to edit a record, usually due to a data error. If "
    "set to (the default) 'environ' the value is taken from the "
    "environment variable CODEMAGUS_EDIT_ON_ERROR. The other options include "
    "returning an error (return_error), returning a zero length record "
    "(return_zero_record), returning a zero length record and writing the "
    "record that failed to a log file (return_zero_record_and_log) or "
```

```

"returning the un-edited original record (return_original). The same "
"keywords should be used for the value of the environment variable "
"if that is to be used instead. If the original record is to be logged "
"then the file name must be defined in the environment variable "
"CODEMAGUS_EDIT_ERROR_LOG."
;

constrain config as ".*\.bed$";
constrain using as ".*";
constrain with as ".*";
constrain on_error as "^\(environ\)\\|\(return_"
    "\ \(error\|zero_record\|zero_record_and_log\|original_record\)\\)$";

path = ${CODEMAGUS_AMDLIBS} "%s";
module = "editam" ${CODEMAGUS_AMDSUFDL};
entry = editam_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](#).
- [2] binary: Fixed and Variable Length Record Stream Access Method Version 1. CML Document CML00005-01, Code Magus Limited, July 2008. [PDF](#).