CODE MAGUS

niplogam: Network Log Reader Access Method
Version 1

CML00074-01

Business Partner IBM

August 16, 2016

Contents

1   Introduction

The `niplog` access method is a module which implements the `recio` [1] provider interface allowing the `recio` user interface to support reading captured network logs. This access method is primarily used by the Network Injection Portal Graphical User Interface (NIP GUI) [2] [4] and Eresia [3] [5] for reading captured logs to enable message manipulation and diagnosis.

This access method is essentially a record broker for the older style of access method so that the tools and systems can use the newer style format specification supported by Recio.

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 `niplog`.

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 `niplog` 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 read a network log file captured previously by another tool.

```
niplog(network.log,using='C:\CodeMagus\CodeMagus\bin\twblnipr.dll')
```

... OR ...

```
niplog(network.log,using='/home/user1/lib/twblnipr.so')
```

3 `niplog` 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 niplog(using,with="a=a");

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

  -- $Author: hayward $
  -- $Date: 2008/10/20 08:28:52 $
  -- $Id: NIPLOG.amd,v 1.2 2008/10/20 08:28:52 hayward Exp $
  -- $Name:  $
  -- $Revision: 1.2 $
  -- $State: Exp $
  --
  -- $Log: NIPLOG.amd,v $
  -- Revision 1.2  2008/10/20 08:28:52  hayward
  -- As the "with" option is not used in the program, but is retained
  -- for compatability to similar AM's (see FILE:fileram) it is given
  -- a default dummy value, so that a dummy value does not have to be
  -- specified at run time.
  --
  -- Revision 1.1.1.1  2008/09/22 12:37:42  hayward
  -- Add niplog Recio AM wrapper for oldstyle Eresia logreader DLL's.
  --
  --

  modes seq_input, skip_input;

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

```
implements tell;

describe using as
   "The 'using' option defines the underlying original file access method "
   "DLL to be used for log data access. All requests to read or write log "
   "data are passed on to this DLL.";

describe with as
   "The 'with' option is used to specify all the options, as comma "
   "separated keyword=value pairs, of the underlying original access "
   "method DLL specified in the 'using' option.";

constrain using as "^.*" ${CODEMAGUS_AMDSUFDL} "$";
constrain with as
"^\([^,=][^,=]*=[^,=][^,=]*\)\(,[^,=][^,=]*=[^,=][^,=]*\)*$";

path = ${CODEMAGUS_AMDLIBS} "%s";
module = "niplogam" ${CODEMAGUS_AMDSUFDL};
entry = niplogam_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.

[2] Eresia Network Injection Portal (NIP) Version 2. CML Document CML00038-02,
    Code Magus Limited, January 2009. PDF.

[3] Code Magus Eresia User Guide. CML Document CML00040-01, Code Magus
    Limited, January 2009. PDF.

[4] Eresia Network Injection Portal (NIP) Version 2.1. CML Document CML00038-21,
    Code Magus Limited, January 2009. PDF.

[5] Code Magus Eresia User Guide Version 2.1. CML Document CML00040-21, Code
    Magus Limited, January 2009. PDF.