Rexx/CSV Reference

Version 1.0

Table of Contents

TABLE OF CONTENTS....1

1. RexxCSV/Introduction [Modules]....2

2. RexxCSV/Constants [Modules]....2

2.1. Constants/MiscellaneousFlags [Definitions]....3

3. RexxCSV/Functions [Modules]....3

3.1. Functions/CSVfini [Functions]....4
3.2. Functions/CSVfree [Functions]....4
3.3. Functions/CSVget_delim [Functions]....5
3.4. Functions/CSVget_opts [Functions]....6
3.5. Functions/CSVget_quote [Functions]....6
3.6. Functions/CSVinit [Functions]....7
3.7. Functions/CSVparse [Functions]....8
3.8. Functions/CSVset_delim [Functions]....9
3.9. Functions/CSVset_opts [Functions]....10
3.10. Functions/CSVset_quote [Functions]....10
3.11. Functions/CSVwrite [Functions]....12

4. RexxCSV/PackageManagement [Modules]....12

4.1. PackageManagement/CSVLoadFuncs [Functions]....12
4.2. PackageManagement/CSVDropFuncs [Functions]....13
4.3. PackageManagement/CSVVariable [Functions]....14
4.4. PackageManagement/CSVQueryFunction [Functions]....title

1. RexxCSV/Introduction [Modules]

[Top] [Modules]

DESCRIPTION

Rexx/CSV is an external function package that ...

USAGE

To make the external functions available add the following lines before: the first call to one of the functions in the package:

```
Call RxFuncAdd 'CSVLoadFuncs', 'rexxcsv', 'CSVLoadFuncs'
Call CSVLoadFuncs
```

DERIVED FROM

TODO

BUGS

PORTABILITY

SEE ALSO

Rexx/CSV lives at http://rexxcsv.sf.net

COPYRIGHT

2. RexxCSV/Constants [Modules]

[Top] [Modules]

DESCRIPTION

The following "constants" are defined when Rexx/CSV starts. By default, all constants are stored in an array with the stem preset to !REXXCSV.! This can be changed by using the 'CONSTANTPREFIX' value of CSVvariable(). If you use "Procedure" on your labels, you MUST "EXPOSE !REXXCSV." or the stem you set with CSVvariable() will not be visible. To reference the constants defined below, you must prefix them. So the "constant" HAVE_REXXCALLBACK would be, by default, referenced in your code as !REXXCSV.!HAVE_REXXCALLBACK.

SEE ALSO

CSVvariable

2.1. Constants/MiscellaneousFlags [Definitions]

[Top] [Constants] [Definitions]

NAME

MiscellaneousFlags

DESCRIPTION

The following is a list of miscellaneous flags.

ATTRIBUTES

- HAVE_REXXCALLBACK - 1 if the interpreter supports RexxCallback() API
- other common constants for package

3. RexxCSV/Functions [Modules]

[Top] [Modules]

DESCRIPTION

These following functions implement the package functionality.

3.1. Functions/CSVfini [Functions]

[Top] [Functions] [Functions]

NAME

CSVfini

SYNOPSIS

rcode = **CSVfini**(p[, cb1[, cb2, data])

FUNCTION

??

ARGUMENTS

- p -
- cb1 -
- cb2 -
- data -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
rcode = CSVfini( p[, cb1[, cb2, data] )
```

3.2. Functions/CSVfree [Functions]

[Top] [Functions] [Functions]

NAME

CSVfree

SYNOPSIS

Call **CSVfree** p

FUNCTION

??

ARGUMENTS

- p -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
Call CSVfree p
```

3.3. Functions/CSVget_delim [Functions]

[Top] [Functions] [Functions]

NAME

CSVget_delim

SYNOPSIS

rcode = **CSVget_delim**(p)

FUNCTION

??

ARGUMENTS

- p -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
rcode = CSVget_delim( p )
```

3.4. Functions/CSVget_opts [Functions]

[Top] [Functions] [Functions]

NAME

CSVget_opts

SYNOPSIS

rcode = **CSVget_opts**(p)

FUNCTION

??

ARGUMENTS

- p -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
rcode = CSVget_opts( p )
```

3.5. Functions/CSVget_quote [Functions]

[Top] [Functions] [Functions]

NAME

CSVget_quote

SYNOPSIS

rcode = **CSVget_quote**(p)

FUNCTION

??

ARGUMENTS

- p -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
rcode = CSVget_quote( p )
```

3.6. Functions/CSVinit [Functions]

[Top] [Functions] [Functions]

NAME

CSVinit

SYNOPSIS

handle = **CSVinit**([options])

FUNCTION

Initialises the CSV function package

ARGUMENTS

- options - [optional] Settings for behaviour of CVS parser.

RESULT

The CSV handle used in all other Rexx/CSV functions.

SEE ALSO

CSVfini, CSVset_opts

SOURCE

```
...
handle = CSVinit( options )
```

3.7. Functions/CSVparse [Functions]

[Top] [Functions] [Functions]

NAME

CSVparse

SYNOPSIS

rcode = **CSVparse**(handle, str, callback, stem[, userdata])

FUNCTION

This function parses the contents of <str> and for each complete field, sets the array variable specified by <stem>. At the end of a complete record, the Rexx procedure specified by <callback> is called at which time the 0th entry for the <stem> is set.

ARGUMENTS

- handle - The CSV parser handle created by CSVinit
- str - The piece of the CSV data to be parsed
- callback - The Rexx procedure to be called at the end of a complete record
- stem - The name of the Rexx array in which the CSV record is placed.
- userdata - Optional user supplied data (any number of arguments)

RESULT

non-zero - error (see !REXXCSV.!CSVCODE and !REXXCSV.!CSVERRM) zero - success

SOURCE

```
...
handle = CSVInit()
...
rcode = CSVparse( handle, chunk, 'ProcessRecord', 'record.', mydata )
...
ProcessRecord: Expose !REXXCSV. record.
Parse Arg eor, stem, mydata
Say 'CSV record terminated by ASCII character:' eor
Say 'CSV record contained in stem:' stem
Say 'My userdata:' mydata
Do i = 1 To record.0
  Say 'Field:' i 'contains' record.i
End
Return
```

3.8. Functions/CSVset_delim [Functions]

[Top] [Functions] [Functions]

NAME

CSVset_delim

SYNOPSIS

Call **CSVset_delim** p, c

FUNCTION

??

ARGUMENTS

- p -
- c -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
Call CSVset_delim p, c
```

3.9. Functions/CSVset_opts [Functions]

[Top] [Functions] [Functions]

NAME

CSVset_opts

SYNOPSIS

rcode = **CSVset_opts**(p, options)

FUNCTION

??

ARGUMENTS

- p -
- options -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
rcode = CSVset_opts( p, options )
```

3.10. Functions/CSVset_quote [Functions]

[Top] [Functions] [Functions]

NAME

CSVset_quote

SYNOPSIS

Call **CSVset_quote** p, c

FUNCTION

??

ARGUMENTS

- p -
- c -

RESULT

??

SEE ALSO

??

NOTES

SOURCE

```
...
Call CSVset_quote p, c
```

3.11. Functions/CSVwrite [Functions]

[Top] [Functions] [Functions]

NAME

CSVwrite

SYNOPSIS

str = **CSVwrite**(src[, quote])

FUNCTION

This function accepts a field value and optional quote character and returns the field value modified with appropriate quotes suitable for writing out to a CSV file.

ARGUMENTS

- str - The CSV field data to be written out
- quote - Optional quote character to use for quoting the field

RESULT

The suitably quoted field data.

SEE ALSO

CSVset_quote, CSVget_quote

SOURCE

```
...
fn = 'my.csv'
field.1 = "3"
field.2 = "It's a field!"
field.0 = 2
Do i = 1 To field.0
   Call Charout fn, CSVwrite( field.i, "'" )
   If i \= field.0 Then Call Charout fn, ','
End
Say
```

4. RexxCSV/PackageManagement [Modules]

[Top] [Modules]

DESCRIPTION

These functions are common to most Rexx external function packages. They descibe how the behaviour of the package can be changed.

4.1. PackageManagement/CSVLoadFuncs [Functions]

[Top] [PackageManagement] [Functions]

NAME

CSVLoadFuncs

SYNOPSIS

rcode = **CSVLoadFuncs**()

FUNCTION

Loads all other RexxCSV external functions

ARGUMENTS

None

RESULT

0 in all cases

SEE ALSO

CSVDropFuncs

NOTES

4.2. PackageManagement/CSVDropFuncs [Functions]

[Top] [PackageManagement] [Functions]

NAME

CSVDropFuncs

SYNOPSIS

rcode = **CSVDropFuncs**(["UNLOAD"])

FUNCTION

Cleans up RexxCSV environment and optionally will drop the external functions.

ARGUMENTS

- UNLOAD - causes the external functions to be dropped.

RESULT

0 in all cases

SEE ALSO

CSVLoadFuncs

NOTES

4.3. PackageManagement/CSVVariable [Functions]

[Top] [PackageManagement] [Functions]

NAME

CSVVariable

SYNOPSIS

rcode = **CSVVariable**(Variable [,NewValue])

FUNCTION

Get or set an internal RexxCSV variable.

ARGUMENTS

- Variable - name of the variable to get or set. See NOTES for
- NewValue - the new value of "Variable", if the variable is settable

RESULT

When setting a variable, then 0 if success, any other value is an error When getting a variable, then the value of the variable is returned.

NOTES

The "Variable" argument can be one of:

```
DEBUG (settable)
   0 - no debugging
   1 - all Rexx variables set by RexxCSV are displayed as they are set
   2 - all RexxCSV functions are traced on entry with argument values and
       on exit with the return value
   4 - all internal RexxCSV functions are traced with their arguments
       (really only useful for developers)
   The values can be added together for a combination of the above details.
DEBUGFILE (settable)
   Where any debugging output is written. By default this goes to
   the system's error stream; usually 'stderr'.
CONSTANTPREFIX (settable)
   The variable name prefix for all RexxCSV constants. By default this is
   '!REXXCSV.!'. If you change this, it is useful to make the prefix result
   in stemmed variables; this makes it far easier to EXPOSE these constants.
VERSION (readonly)
   The full version details of RexxCSV in the format:
   package version version_date
   Where:
      package      - the string 'rexxcsv'
      version      - package version in n.n format; eg 1.0
      version_date - date package was released in DATE('N') format
```

SOURCE

```
...
Say 'We are running at debug level:' CSVVariable( 'DEBUG' )
```

4.4. PackageManagement/CSVQueryFunction [Functions]

[Top] [PackageManagement] [Functions]

NAME

CSVQueryFunction

SYNOPSIS

rcode = **CSVQueryFunction**(FunctionName|ResultArray[, Option])

FUNCTION

Populates an array of all functions supplied by this package depending on Option

ARGUMENTS

- FunctionName - the name of a function to query (no trailing period)
- ResultArray - the stem (trailing period) in which the list of functions is returned
- Option - one of 'R' (the default) for "registered" functions or 'A' for "available" functions

RESULT

0 if successful or 1 if the FunctionName is not found

NOTES

To determine if a FunctionName can be executed in your code, pass the function name as the first argument, and 'R' as the second. If the function can be called the function returns 0, otherwise it returns 1