

Ruby trunk - Feature #7525

How to avoid memory leak when something gets wrong and throw exception when using win32api?

12/06/2012 04:52 PM - mghomn (Justin Peal)

Status:	Rejected
Priority:	Normal
Assignee:	usa (Usaku NAKAMURA)
Target version:	2.0.0
Description	
require 'win32api'	
module Crypto	
# Common API	
NULL = 0	
@GetLastError = Win32API.new('kernel32', 'GetLastError', '', 'I')	
@lstrlen = Win32API.new('kernel32', 'lstrlenW', 'L', 'I')	
# Memory API	
@RtlMoveMemory = Win32API.new('kernel32', 'RtlMoveMemory', 'PLL', 'I')	
@LocalFree = Win32API.new('kernel32', 'LocalFree', 'L', 'I')	
# Crypto API	
CRYPTPROTECT_UI_FORBIDDEN = 0x01	
@CryptProtectData = Win32API.new('crypt32', 'CryptProtectData', 'PPPPPLP', 'I')	
@CryptUnprotectData = Win32API.new('crypt32', 'CryptUnprotectData', 'PPPPPLP', 'I')	
def self.error func	
puts "#{func} Error = #{@GetLastError.call()}"	
end	
def self.encrypt str, entropy, desc	
pDataIn = [str.bytesize, str].pack('Lp')	
szDataDescr = (desc + "\0").encode(Encoding::UTF_16LE)	
pOptionalEntropy = [entropy.bytesize, entropy].pack('Lp')	
pvReserved = pPromptStruct = NULL	
dwFlags = CRYPTPROTECT_UI_FORBIDDEN	
pDataOut = [0, "].pack('Lp')	
return error('CryptProtectData') if @CryptProtectData.call(pDataIn, szDataDescr, pOptionalEntropy, pvReserved, pPromptStruct, dwFlags, pDataOut) == 0	
cbData, pbData = pDataOut.unpack('LL')	
ret = ''.encode(Encoding::BINARY) * cbData	
return error('RtlMoveMemory') if @RtlMoveMemory.call(ret, pbData, cbData) == 0	
return error('LocalFree') if @LocalFree.call(pbData) != NULL	
ret	
end	
def self.decrypt str, entropy, desc	
pDataIn = [str.bytesize, str].pack('Lp')	
ppszDataDescr = [NULL].pack('L')	
pOptionalEntropy = [entropy.bytesize, entropy].pack('Lp')	
pvReserved = pPromptStruct = NULL	
dwFlags = CRYPTPROTECT_UI_FORBIDDEN	
pDataOut = [0, "].pack('Lp')	
return error('CryptUnprotectData') if @CryptUnprotectData.call(pDataIn, ppszDataDescr, pOptionalEntropy, pvReserved, pPromptStruct, dwFlags, pDataOut) == 0	
pszDataDescr = ppszDataDescr.unpack('L').first	
szDataDescr = ''.encode(Encoding::UTF_16LE) * @lstrlen.call(pszDataDescr)	
return error('RtlMoveMemory') if @RtlMoveMemory.call(szDataDescr, pszDataDescr, szDataDescr.bytesize) == 0	
return error('LocalFree') if @LocalFree.call(pszDataDescr) != NULL	
szDataDescr.encode!(Encoding::UTF_8)	
cbData, pbData = pDataOut.unpack('LL')	
ret = ''.encode(Encoding::BINARY) * cbData	
return error('RtlMoveMemory') if @RtlMoveMemory.call(ret, pbData, cbData) == 0	

```
return error('LocalFree') if @LocalFree.call(pbData) != NULL
desc = " unless desc
return error('Unmatched description') unless desc == szDataDescr
ret.force_encoding(Encoding::UTF_8)
end
end

if $0 == FILE
def test plain, entropy, desc
puts "plain = #{plain}, entropy = #{entropy}, desc = #{desc}"
cipher = Crypto.encrypt(plain, entropy, desc)
puts "cipher = #{cipher.unpack('H*').first}"
recover = Crypto.decrypt(cipher, entropy, desc)
puts "recover = #{recover}"
end

begin
test('abcd', 'efgh', 'ijkl')
rescue
puts $!.to_s.force_encoding(Encoding::UTF_8), $!.backtrace.join($/)
end
end
```

History

#1 - 12/09/2012 09:11 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned
- Assignee set to usa (Usaku NAKAMURA)
- Target version set to 2.0.0

Is this really a feature request? Or bug report?

Assigning to usa.

Usa-san, if this is a feature, please set the target to next minor.

If this is a bug report, please consider how to fix.

--

Yusuke Endoh mame@tsq.ne.jp

#2 - 12/21/2012 09:56 PM - usa (Usaku NAKAMURA)

- Status changed from Assigned to Rejected

This is not a feature request nor a bug report. Only question.

Generically, you can use begin-ensure-end syntax to release resources.

For example:

```
begin
foo = get_some_resource
do_something_and_raises_exception
ensure
release_the_resource(foo) if foo
end
```

#3 - 01/30/2013 09:53 AM - mghomn (Justin Peal)

Thank you for the resource release schema!