COM server realizations: Instruction flow comparisson

HLL (Delphi, but no matter, mostly inline assembler) vs flat assembler

Nikiforov Prokhor Mikhailovich (aka ProMiNick)

23.06.2019

This document represent list of all procedures used in realizations of COM server.They combined in pairs – one maked in HLL and corresponding one maked in flat assembler.First page mostly placeholder to save even number of pages and realize table of contents.

= Errornous Fasm compiled COM Server = Page 2

Table of Contents

DllGetClassObject 3-4

DllCanUnloadNow 5-6

hlpr__RegCreateKeyWithValue 7-8

DllRegisterServer 9-10

DllUnregisterServer 11-12

hlpr_HeapFreeStack &hlpr_allocCOMobj & hlpr_allocCOMobjStr 13-14

QRClassFactory_QueryInterface 15-16

QRClassFactory_AddRef 17-18

QRClassFactory_Release 19-20

QRClassFactory_CreateInstance 21-22

QRClassFactory_LockServer 23-24

ContextMenu_QueryInterface 25-26

ContextMenu_AddRef 27-28

ContextMenu_Release 29-30

ContextMenu_QueryContextMenu 31-32

hlpr_RegisterCOMServer 33-34

hlpr_MessagingInvokeCommand 35-36

= Working HLL compiled COM Server = Page 3

```
; Exported entry   4. DllGetClassObject


; Attributes: bp-based frame

; HRESULT __stdcall DllGetClassObject(const IID *const rclsid, const IID *const riid, LPVOID *ppv)
public DllGetClassObject
DllGetClassObject proc near

rclsid= dword ptr  8
riid= dword ptr  0Ch
ppv= dword ptr  10h

push    ebp
mov     ebp, esp
push    ebx
mov     ebx, [ebp+ppv]
test    ebx, ebx
jz      short _ret_E_POINTER
```

```
xor     eax, eax
mov     [ebx], eax
push    esi
push    edi
mov     esi, [ebp+rclsid]
mov     edi, offset CLSID_QuickRegister
mov     ecx, 4
repe cmpsd
pop     edi
pop     esi
jnz     short _ret_CLASS_E_CLASSNOTAVAILABLE
```

```
push    ebx
mov     eax, [ebp+riid]
push    eax
mov     eax, offset QRClassFactory_COMobject
push    eax
mov     eax, [eax]
call    dword ptr [eax]
jmp     short _locret
```

```
_ret_CLASS_E_CLASSNOTAVAILABLE:
mov     eax, CLASS_E_CLASSNOTAVAILABLE
```

```
_ret_E_POINTER:
mov     eax, E_POINTER
jmp     short _locret
```

```
_locret:
pop     ebx
pop     ebp
retn    0Ch
DllGetClassObject endp
```

= Errornous Fasm compiled COM Server = Page 4

```
; Exported entry   1. DllGetClassObject


; Attributes: bp-based frame

; HRESULT __stdcall DllGetClassObject(const IID *const rclsid, const IID *const riid, LPVOID *ppv)
public DllGetClassObject
DllGetClassObject proc near

rclsid= dword ptr  8
riid= dword ptr  0Ch
ppv= dword ptr  10h

push    ebx
mov     ebx, [esp+4+ppv]
test    ebx, ebx
jz      short _ret_E_POINTER
```

```
xor     eax, eax
mov     [ebx], eax
push    esi
push    edi
cld
mov     esi, [esp+0Ch+rclsid]
mov     edi, offset CLSID_QuickRegister
mov     ecx, 4
repe cmpsd
pop     edi
pop     esi
jnz     short _ret_CLASS_E_CLASSNOTAVAILABLE
```

```
push    ebx
push    [esp+8+riid]
mov     eax, QRClassFactory_COMobject
push    eax
mov     eax, [eax]
call    dword ptr [eax]
jmp     short _locret
```

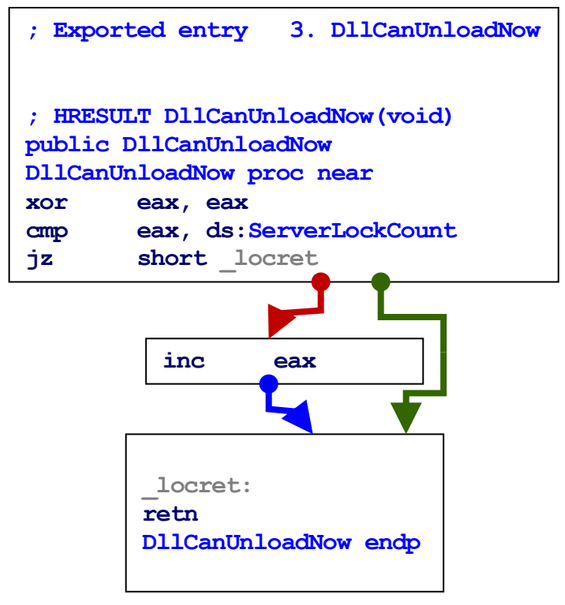
```
_ret_CLASS_E_CLASSNOTAVAILABLE:
mov     eax, CLASS_E_CLASSNOTAVAILABLE
```

```
_ret_E_POINTER:
mov     eax, E_POINTER
jmp     short _locret
```

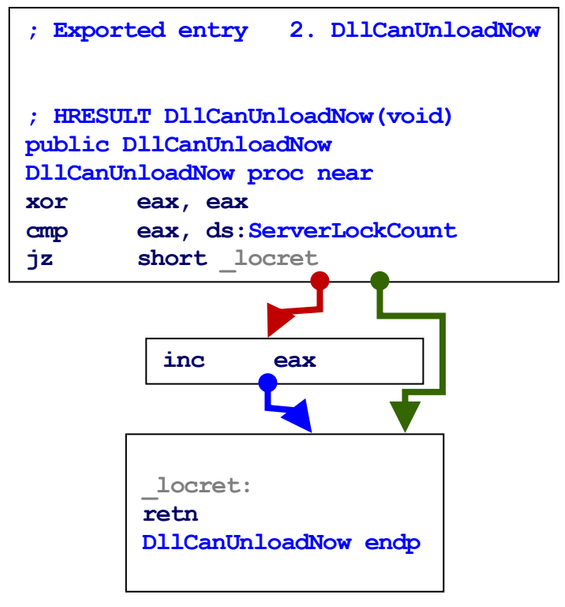
```
_locret:
pop     ebx
retn    0Ch
DllGetClassObject endp
```

= Working HLL compiled COM Server =

Page 5



= Errornous Fasm compiled COM Server = Page 6



= Working HLL compiled COM Server =
Page 7

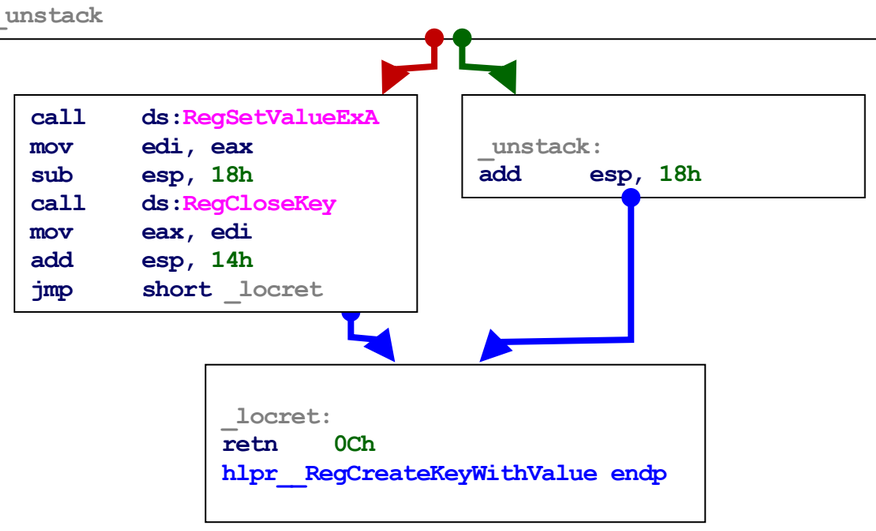
```
; Attributes: bp-based frame

; int __stdcall hlpr__RegCreateKeyWithValue(LPCSTR lpSubKey, LPCSTR lpValueName, BYTE *lpData)
hlpr__RegCreateKeyWithValue proc near

dwDisposition= dword ptr -8
phkResult= dword ptr -4
lpSubKey= dword ptr  8
lpValueName= dword ptr  0Ch
lpData= dword ptr  10h

push    ebp
mov     ebp, esp
add     esp, -8
push    ebx
push    esi
mov     esi, [ebp+lpData]
lea     eax, [ebp+dwDisposition]
push    eax             ; lpdwDisposition
lea     eax, [ebp+phkResult]
push    eax             ; phkResult
push    0               ; lpSecurityAttributes
push    KEY_READ or KEY_WRITE ; samDesired
push    0               ; dwOptions
push    offset Class    ; lpClass
push    0               ; Reserved
mov     eax, [ebp+lpSubKey]
push    eax             ; lpSubKey
mov     eax, ds:RootKey
push    eax             ; hKey
call    RegCreateKeyExA
mov     ebx, eax
test    ebx, ebx
jnz     short _locret
```

```
mov     eax, esi        ; char *
call    @SysutilsMini@StrLen$qqrpxc ; SysutilsMini::StrLen(char *)
inc     eax
push    eax             ; cbData
push    esi             ; lpData
push    REG_SZ          ; dwType
push    0               ; Reserved
mov     eax, [ebp+lpValueName]
push    eax             ; lpValueName
mov     eax, [ebp+phkResult]
push    eax             ; hKey
call    RegSetValueExA
mov     ebx, eax
mov     eax, [ebp+phkResult]
push    eax             ; hKey
call    RegCloseKey
```

```
; Attributes: library function

; _DWORD __fastcall SysutilsMini::StrLen(const char *)
@SysutilsMini@StrLen$qqrpxc proc near
mov     edx, edi
mov     edi, eax
mov     ecx, -1
xor     al, al
repne scasb
mov     eax, -2
sub     eax, ecx
mov     edi, edx
retn
@SysutilsMini@StrLen$qqrpxc endp
```

```
_locret:
mov     eax, ebx
pop     esi
pop     ebx
pop     ecx
pop     ecx
pop     ebp
retn    0Ch
hlpr__RegCreateKeyWithValue endp
```

```
; int __stdcall hlpr__RegCreateKeyWithValue(LPCSTR lpSubKey, LPCSTR lpValueName, BYTE *lpData)
hlpr__RegCreateKeyWithValue proc near

lpSubKey= dword ptr  4
lpValueName= dword ptr  8
lpData= dword ptr  0Ch

cld
mov     edi, [esp+lpData]
mov     ecx, -1
xor     eax, eax
repne scasb
not     ecx
sub     edi, ecx
push    ecx             ; cbData
push    edi             ; lpData
push    REG_SZ          ; dwType
push    eax             ; Reserved
push    [esp+10h+lpValueName] ; lpValueName
push    eax             ; hKey
mov     edi, esp
push    eax             ; lpdwDisposition
push    edi             ; phkResult
push    eax             ; lpSecurityAttributes
push    KEY_READ or KEY_WRITE ; samDesired
push    0               ; dwOptions
push    eax             ; lpClass
push    eax             ; Reserved
push    [esp+34h+lpSubKey] ; lpSubKey
push    RootKey         ; hKey
call    ds:RegCreateKeyExA
test    eax, eax
jnz     short _unstack
```



= Working HLL compiled COM Server =
Page 9

```
; Exported entry   2. DllRegisterServer


; Attributes: bp-based frame

; HRESULT __stdcall DllRegisterServer()
public DllRegisterServer
DllRegisterServer proc near
push    ebx
mov     ds:RootKey, HKEY_CLASSES_ROOT
mov     ebx, S_FALSE
push    offset CLSIDDescription ; "Quick Register Context Menu Shell Exten"...
push    offset NullStr  ; ""
push    offset CLSIDStr ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    261             ; nSize
push    offset REGFileName ; lpFilename
mov     eax, ds:hInstance
push    eax             ; hModule
call    GetModuleFileNameA
test    eax, eax
jbe     _unregister
```

```
push    offset REGFileName ; lpData
push    offset NullStr  ; ""
push    offset CLSIDInprocServer ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset ApartmentStr ; "Apartment"
push    offset ThreadingModelStr ; "ThreadingModel"
push    offset CLSIDInprocServer ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset NullStr  ; ""
push    offset NullStr  ; ""
push    offset dllfile_shellex_Str ; "dllfile\\shellex"
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset NullStr  ; ""
push    offset NullStr  ; ""
push    offset dllfile_shellex_CMH_Str ; "dllfile\\shellex\\ContextMenuHandlers"
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset CLSIDString_QuickRegister ; "{40E69241-5D1A-11D1-81CB-0020AF3E97A9}"
push    offset NullStr  ; ""
push    offset dllfile_shellex_CMH_QR_Str ; "dllfile\\shellex\\ContextMenuHandlers\\"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
_unregister:
call    DllUnregisterServer
mov     ebx, SELFREG_E_CLASS
push    MB_ICONERROR or MB_OK ; uType
push    offset REGFileName ; lpCaption
mov     eax, ds:lpErrUpdReg  ; offset ErrUpdRegistry
push    eax             ; lpText
push    0               ; hWnd
call    MessageBoxA
```

```
mov     ds:RootKey, HKEY_LOCAL_MACHINE
push    offset NullStr  ; ""
push    offset NullStr  ; ""
push    offset cur_vers_shellex_Str ; "SOFTWARE\\Microsoft\\Windows\\CurrentVe"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _locret
```

```
push    offset CLSIDDescription ; "Quick Register Context Menu Shell Exten"...
push    offset CLSIDString_QuickRegister ; "{40E69241-5D1A-11D1-81CB-0020AF3E97A9}"
push    offset cur_vers_shellex_approv_Str ; "SOFTWARE\\Microsoft\\Windows\\CurrentVe"...
call    hlpr__RegCreateKeyWithValue
mov     ebx, eax
jmp     _locret
```

```
_locret:
mov     eax, ebx
pop     ebx
retn
DllRegisterServer endp
```

= Errornous Fasm compiled COM Server =
Page 10

```
; Exported entry   3. DllRegisterServer


; HRESULT __stdcall DllRegisterServer()
public DllRegisterServer
DllRegisterServer proc near
push    edi
mov     ds:RootKey, HKEY_CLASSES_ROOT
push    offset CLSIDDescription ; "Quick Register Context Menu Shell Exten"...
push    offset NullStr  ; ""
push    offset CLSIDStr ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    261             ; nSize
push    offset REGFileName ; lpFilename
mov     eax, ds:hInstance
push    eax             ; hModule
call    ds:GetModuleFileNameA
test    eax, eax
jbe     _unregister
```

```
push    offset REGFileName ; lpData
push    offset NullStr  ; ""
push    offset CLSIDInprocServer ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset ApartmentStr ; "Apartment"
push    offset ThreadingModelStr ; "ThreadingModel"
push    offset CLSIDInprocServer ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset NullStr  ; ""
push    offset NullStr  ; ""
push    offset dllfile_shellex_Str ; "dllfile\\shellex"
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset NullStr  ; ""
push    offset NullStr  ; ""
push    offset dllfile_shellex_CMH_Str ; "dllfile\\shellex\\ContextMenuHandlers"
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
push    offset CLSIDString_QuickRegister ; "{40E69241-5D1A-11D1-81CB-0020AF3E97A9}"
push    offset NullStr  ; ""
push    offset dllfile_shellex_CMH_QR_Str ; "dllfile\\shellex\\ContextMenuHandlers\\"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _unregister
```

```
_unregister:
call    DllUnregisterServer
mov     ebx, SELFREG_E_CLASS
push    MB_ICONERROR or MB_OK ; uType
push    offset REGFileName ; lpCaption
mov     eax, offset ErrUpdRegistry ; "Error updating registry"
push    eax             ; lpText
push    0               ; hWnd
call    ds:MessageBoxA
```

```
mov     ds:RootKey, HKEY_LOCAL_MACHINE
push    offset NullStr  ; ""
push    offset NullStr  ; ""
push    offset cur_vers_shellex_Str ; "SOFTWARE\\Microsoft\\Windows\\CurrentVe"...
call    hlpr__RegCreateKeyWithValue
test    eax, eax
jnz     _ret_S_FALSE
```

```
_ret_S_FALSE:
xor     eax, eax
inc     eax
```

```
push    offset CLSIDDescription ; "Quick Register Context Menu Shell Exten"...
push    offset CLSIDString_QuickRegister ; "{40E69241-5D1A-11D1-81CB-0020AF3E97A9}"
push    offset cur_vers_shellex_approv_Str ; "SOFTWARE\\Microsoft\\Windows\\CurrentVe"...
call    hlpr__RegCreateKeyWithValue
jmp     _locret
```

```
_locret:
pop     edi
retn
DllRegisterServer endp
```

= Working HLL compiled COM Server =
Page 11

```
; Exported entry   1. DllUnregisterServer


; HRESULT __stdcall DllUnregisterServer()
public DllUnregisterServer
DllUnregisterServer proc near
push    ebx
mov     ebx, offset RootKey
mov     dword ptr [ebx], HKEY_CLASSES_ROOT
push    offset dllfile_shellex_CMH_QR_Str ; "dllfile\\shellex\\ContextMenuHandlers\\"...
mov     eax, [ebx]
push    eax             ; hKey
call    RegDeleteKeyA
push    offset dllfile_shellex_CMH_Str ; "dllfile\\shellex\\ContextMenuHandlers"
mov     eax, [ebx]
push    eax             ; hKey
call    RegDeleteKeyA
push    offset dllfile_shellex_Str ; "dllfile\\shellex"
mov     eax, [ebx]
push    eax             ; hKey
call    RegDeleteKeyA
push    offset CLSIDInprocServer ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
mov     eax, [ebx]
push    eax             ; hKey
call    RegDeleteKeyA
push    offset CLSIDStr ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
mov     eax, [ebx]
push    eax             ; hKey
call    RegDeleteKeyA
push    eax             ; hKey
call    RegDeleteKeyA
xor     eax, eax
pop     ebx
retn
DllUnregisterServer endp
```

= Errornous Fasm compiled COM Server =
Page 12

```
; Exported entry   4. DllUnregisterServer


; HRESULT __stdcall DllUnregisterServer()
public DllUnregisterServer
DllUnregisterServer proc near
mov     RootKey, HKEY_CLASSES_ROOT
push    offset dllfile_shellex_CMH_QR_Str ; "dllfile\\shellex\\ContextMenuHandlers\\"...
push    RootKey         ; hKey
call    ds:RegDeleteKeyA
push    offset dllfile_shellex_CMH_Str ; "dllfile\\shellex\\ContextMenuHandlers"
push    RootKey         ; hKey
call    ds:RegDeleteKeyA
push    offset dllfile_shellex_Str ; "dllfile\\shellex"
push    RootKey         ; hKey
call    ds:RegDeleteKeyA
push    offset CLSIDInprocServer ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
push    RootKey         ; hKey
call    ds:RegDeleteKeyA
push    offset CLSIDStr ; "CLSID\\{40E69241-5D1A-11D1-81CB-0020AF3"...
push    RootKey         ; hKey
call    ds:RegDeleteKeyA
retn
DllUnregisterServer endp
```

= Working HLL compiled COM Server = Page 13

```
; BOOL __stdcall hlpr_HeapFreeStack(LPVOID lpMem)
hlpr_HeapFreeStack proc near

lpMem= dword ptr  4

pop     ebx
push    0                   ; dwFlags
push    ds:pHeap            ; hHeap
push    ebx
jmp     HeapFree
hlpr_HeapFreeStack endp
```

```
; LPVOID hlpr_allocCOMobj
hlpr_allocCOMobj proc near
push    10h         ; SIZE_T
jmp     _alloc
```

```
; LPVOID hlpr_allocCOMobjStr
hlpr_allocCOMobj proc near
push    261         ; SIZE_T
```

```
_alloc:
push    0           ; dwFlags
push    ds:pHeap    ; hHeap
call    HeapAlloc
test    eax, eax
retn
```

```
hlpr_allocCOMobj endp
```

```
hlpr_allocCOMobjStr endp
```

= Errornous Fasm compiled COM Server =

Page 14

```
; BOOL __stdcall hlpr_HeapFreeStack(LPVOID lpMem)
hlpr_HeapFreeStack proc near

lpMem= dword ptr  4

pop     ebx
push    0               ; dwFlags
push    ds:pHeap        ; hHeap
push    ebx
jmp     ds:HeapFree
hlpr_HeapFreeStack endp
```

```
; LPVOID hlpr_allocCOMobj
hlpr_allocCOMobj proc near
push    10h         ; SIZE_T
jmp     _alloc
```

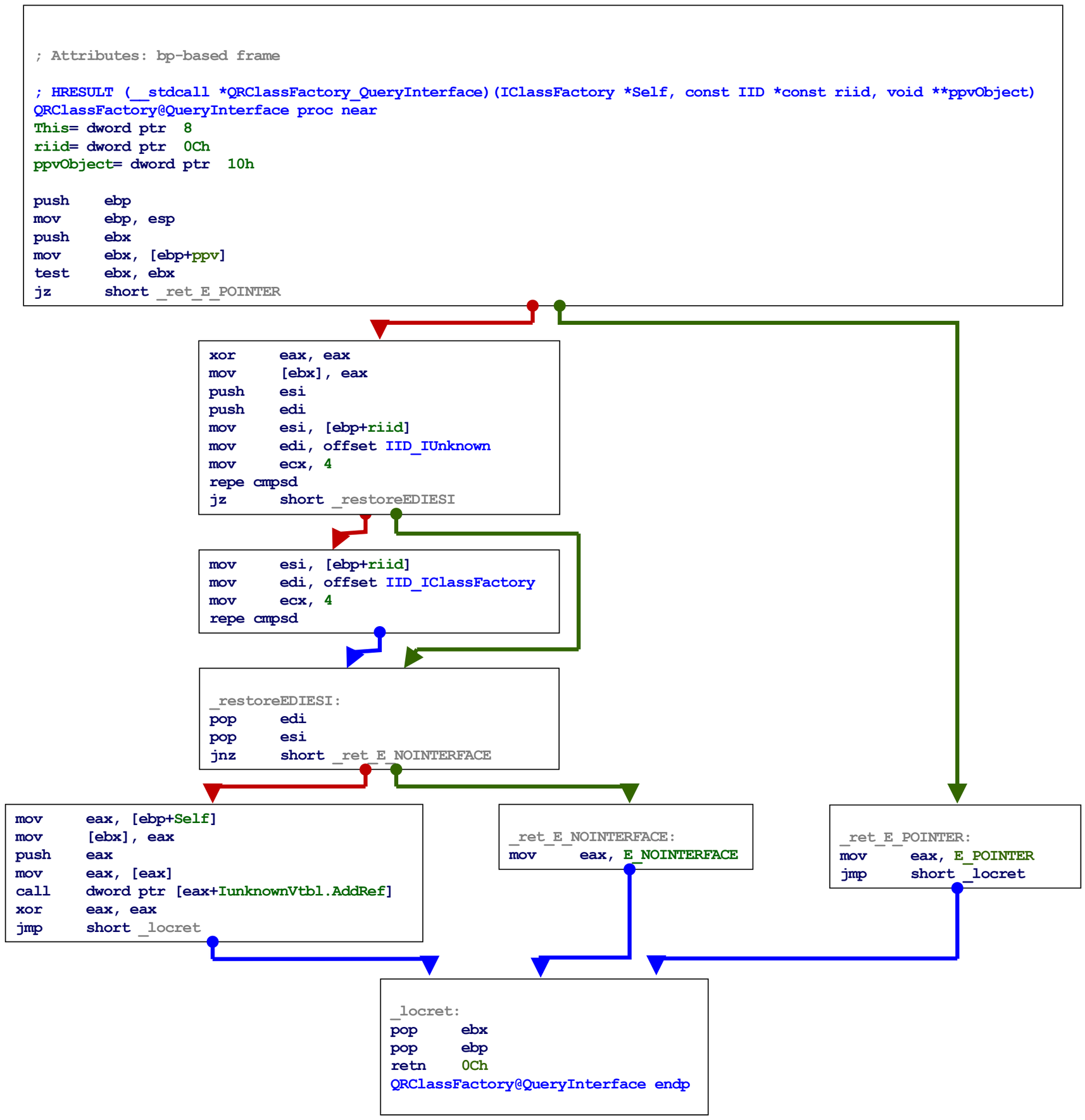
```
; LPVOID hlpr_allocCOMobjStr
hlpr_allocCOMobj proc near
push    261         ; SIZE_T
```

```
_alloc:
push    0           ; dwFlags
push    ds:pHeap    ; hHeap
call    ds:HeapAlloc
test    eax, eax
retn
```

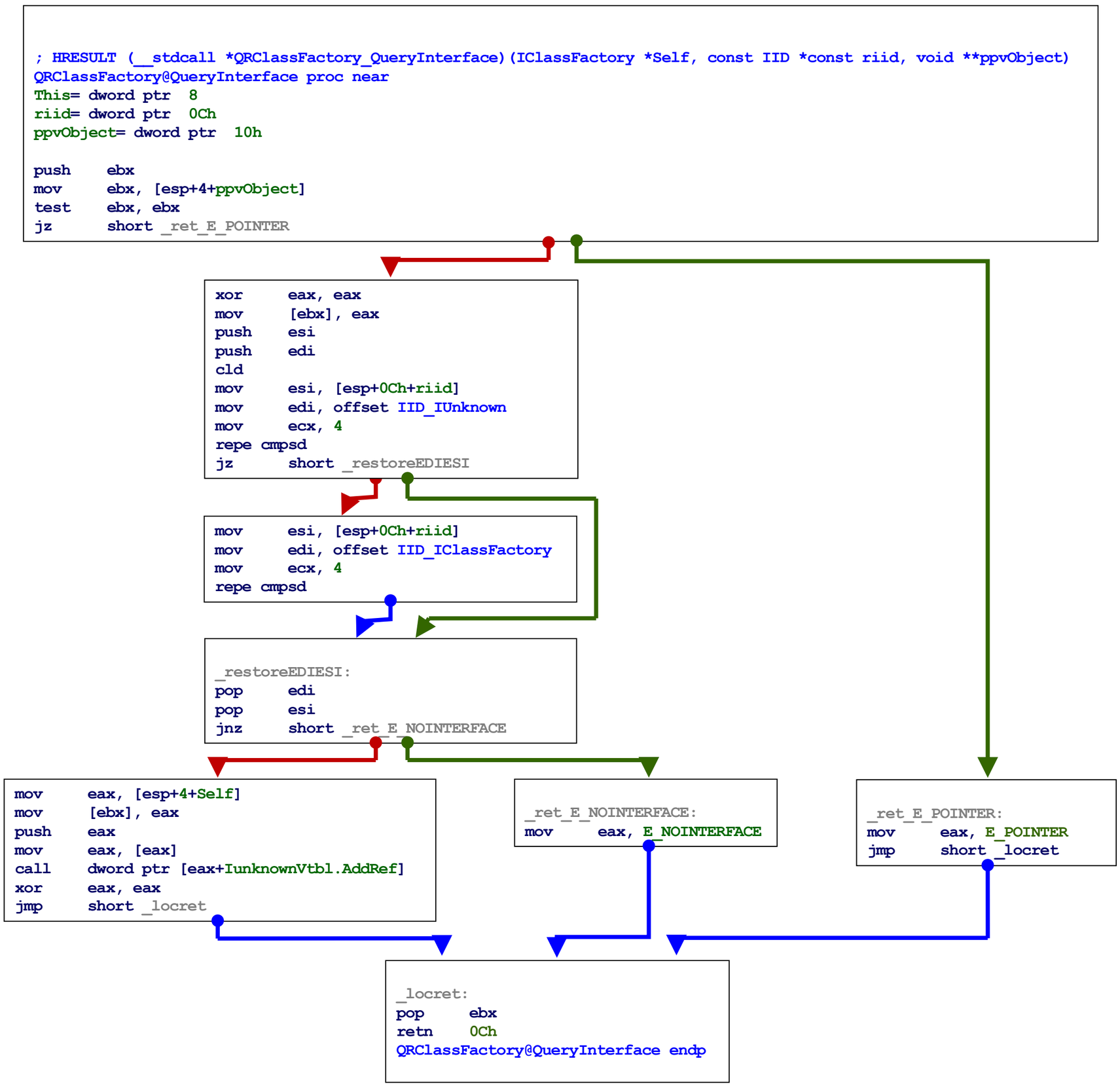
```
hlpr_allocCOMobj endp
```

```
hlpr_allocCOMobjStr endp
```

= Working HLL compiled COM Server =
Page 15



= Errornous Fasm compiled COM Server =
Page 16



= Working HLL compiled COM Server =
Page 17

```
; Attributes: bp-based frame

; ULONG (__stdcall *QRClassFactory_AddRef)(IClassFactory *Self)
QRClassFactory@AddRef proc near
push    ebp
mov     ebp, esp
push    offset ServerLockCount ; lpAddend
call    InterlockedIncrement
mov     eax, 2
pop     ebp
retn    4
QRClassFactory@AddRef endp
```

```
; ULONG (__stdcall *QRClassFactory_AddRef)(IClassFactory *Self)
QRClassFactory@AddRef proc near
push    offset ServerLockCount ; lpAddend
call    ds:InterlockedIncrement
mov     eax, 2
retn    4
QRClassFactory@AddRef endp
```

= Working HLL compiled COM Server =
Page 19

```
; Attributes: bp-based frame

; ULONG (__stdcall *QRClassFactory_Release)(IClassFactory *Self)
QRClassFactory@Release proc near
push    ebp
mov     ebp, esp
push    offset ServerLockCount ; lpAddend
call    InterlockedDecrement
mov     eax, 1
pop     ebp
retn    4
QRClassFactory@Release endp
```

```
; ULONG (__stdcall *QRClassFactory_Release)(IClassFactory *Self)
QRClassFactory@Release proc near
push    offset ServerLockCount ; lpAddend
call    ds:InterlockedDecrement
xor     eax, eax
inc     eax
retn    4
QRClassFactory@Release endp
```

= Working HLL compiled COM Server =
Page 21

```
; Attributes: bp-based frame

; HRESULT (__stdcall *QRClassFactory_CreateInstance)(IClassFactory *Self, IUnknown *pUnkOuter, const IID *const
riid, void **ppvObject)
QRClassFactory@CreateInstance proc near

pUnkOuter= dword ptr  0Ch
riid= dword ptr  10h
ppvObject= dword ptr  14h

push    ebp
mov     ebp, esp
push    ebx
mov     ebx, [ebp+ppvObject]
test    ebx, ebx
jz      short _ret_E_POINTER
```

```
xor     eax, eax
mov     [ebx], eax
cmp     [ebp+pUnkOuter], eax
jnz     short _ret_CLASS_E_NOAGGREGATION
```

```
call    hlpr_allocCOMobj
jz      short _ret_E_OUTOFMEMORY
```

```
push    eax
call    hlpr_allocCOMobjStr
jz      short _free_ret_E_OUTOFMEMORY
```

```
pop     edx
mov     byte ptr [eax], 0
xchg    esp, edx
add     esp, 10h
push    eax
push    0
push    offset ShellExtInit
push    offset ContextMenu
xchg    esp, edx
push    edx
push    eax
push    ebx
push    [ebp+riid]
push    edx
mov     eax, [edx]
call    dword ptr [eax+IunknownVtbl.QueryInterface]
cmp     eax, 0
jl      short _ret_E_FAIL
```

```
_free_ret_E_OUTOFMEMORY:
call    hlpr_HeapFreeStack
```

```
pop     ebx
pop     ebx
push    offset ServerLockCount ; lpAddend
call    InterlockedIncrement
xor     eax, eax
jmp     short _locret
```

```
_ret_E_FAIL:
call    hlpr_HeapFreeStack
call    hlpr_HeapFreeStack
mov     eax, E_FAIL
```

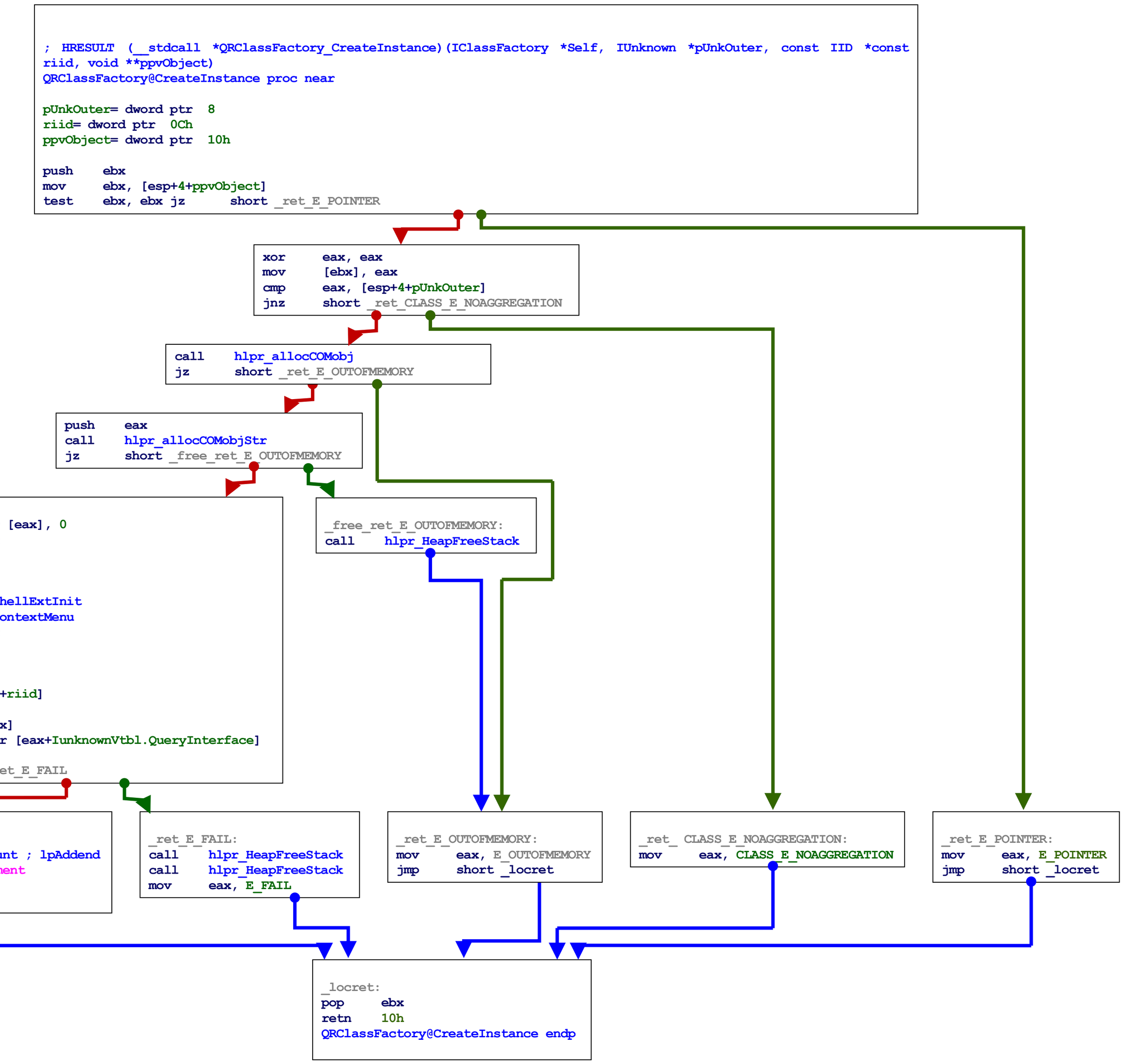
```
_ret_E_OUTOFMEMORY:
mov     eax, E_OUTOFMEMORY
jmp     short _locret
```

```
_ret_ CLASS_E_NOAGGREGATION:
mov     eax, CLASS_E_NOAGGREGATION
```

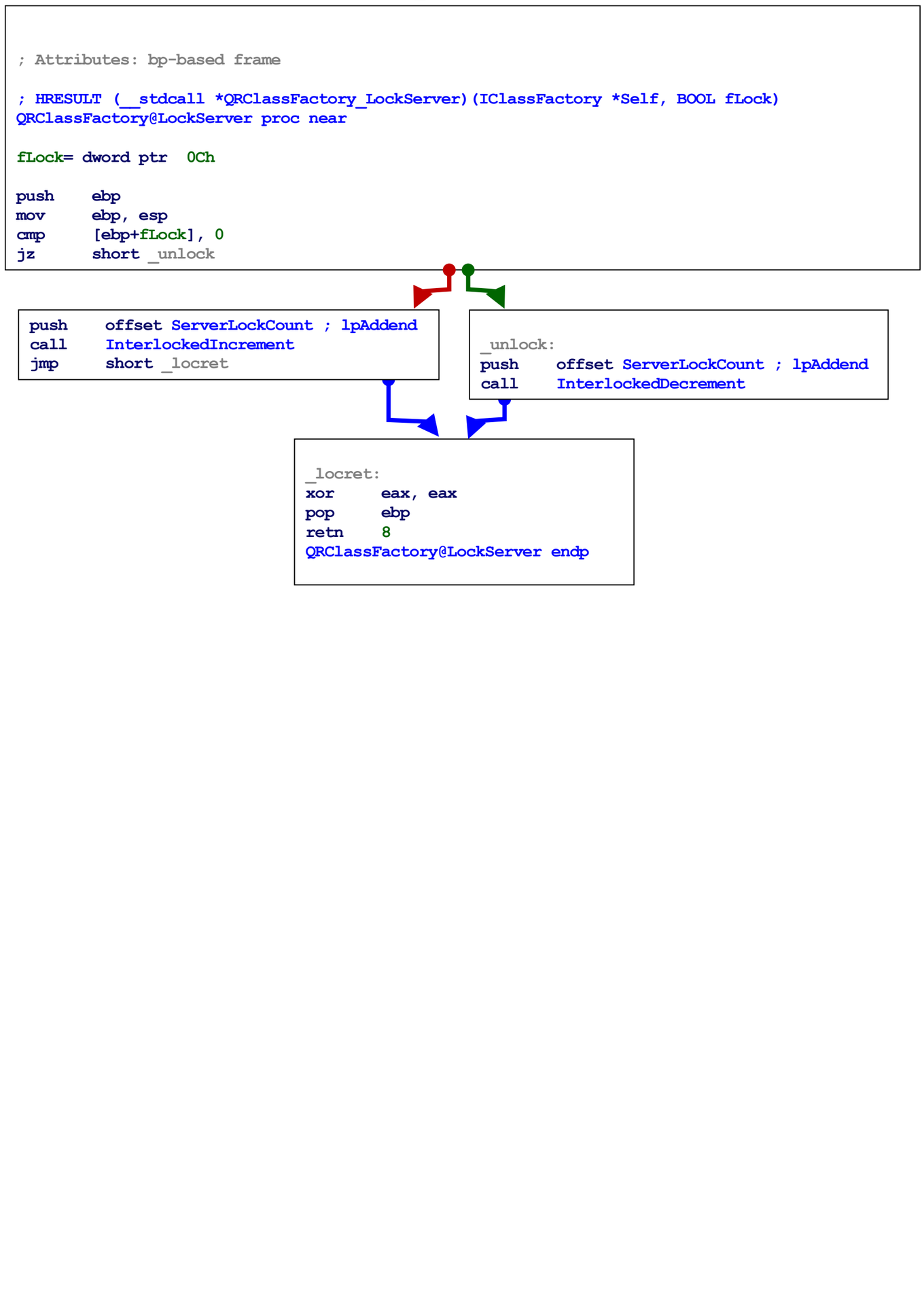
```
_ret_E_POINTER:
mov     eax, E_POINTER
jmp     short _locret
```

```
_locret:
pop     ebx
pop     ebp
retn    10h
QRClassFactory@CreateInstance endp
```

= Errornous Fasm compiled COM Server =
Page 22



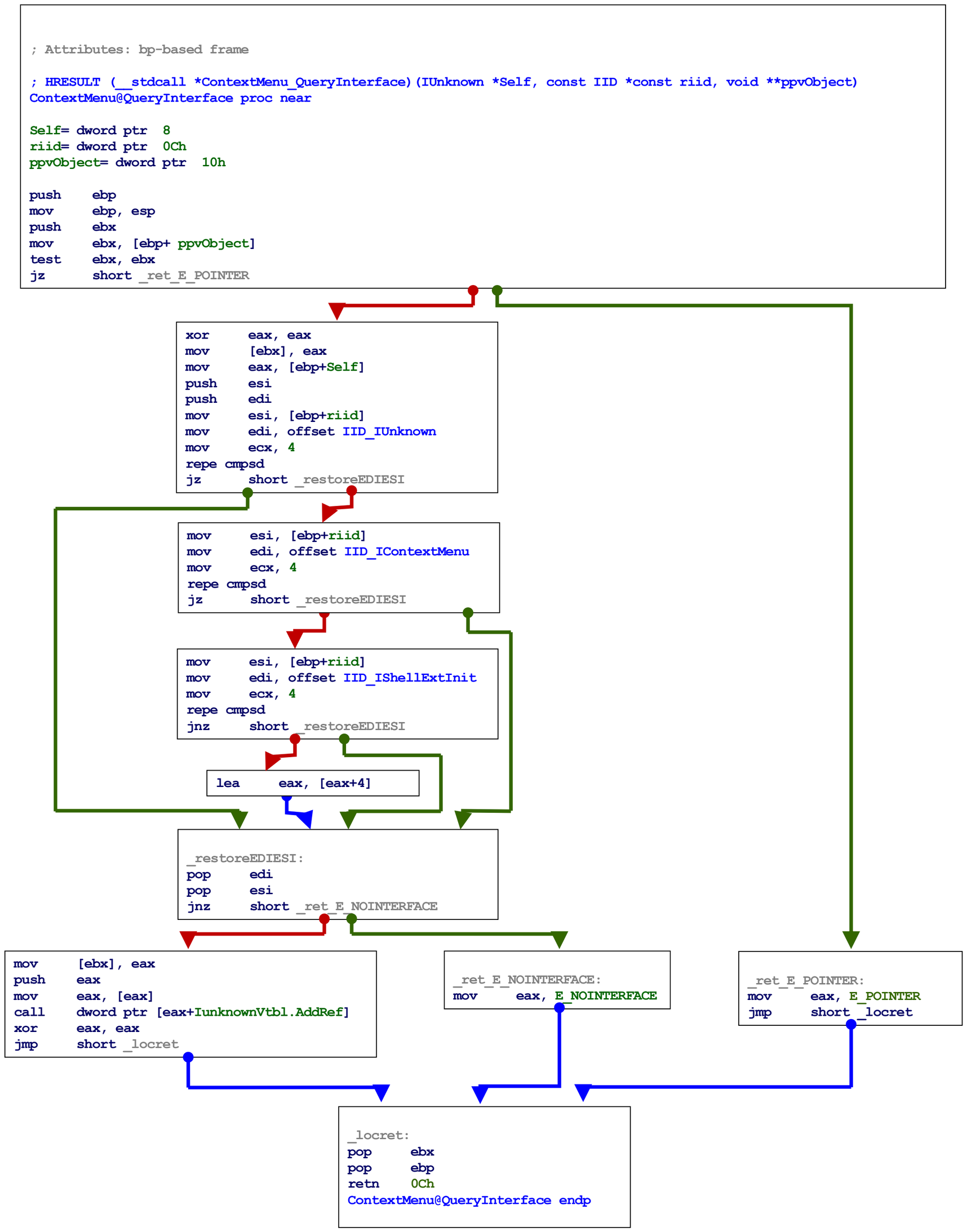
= Working HLL compiled COM Server =
Page 23



```
; HRESULT (__stdcall *QRClassFactory_LockServer)(IClassFactory *Self, BOOL fLock)
QRClassFactory@LockServer proc near

fLock= dword ptr  8

mov     eax, [esp+fLock]
lea     eax, _Interlockeds[eax*4]
push    offset ServerLockCount
call    dword ptr [eax]
xor     eax, eax
retn    8
QRClassFactory@LockServer endp
```

```
dd offset InterlockedIncrement
_Interlockeds:
dd offset InterlockedDecrement
dd offset InterlockedIncrement
```

= Working HLL compiled COM Server =
Page 25



= Errornous Fasm compiled COM Server =
Page 26

```
; HRESULT (__stdcall *ContextMenu_QueryInterface)(IUnknown *Self, const IID *const riid, void **ppvObject)
ContextMenu@QueryInterface proc near

Self= dword ptr  4
riid= dword ptr  8
ppvObject= dword ptr  0Ch

push    ebx
mov     ebx, [esp+4+ppvObject]
test    ebx, ebx
jz      short _ret_E_POINTER
```

```
xor     eax, eax
mov     [ebx], eax
mov     eax, [esp+4+Self]
push    esi
push    edi
cld
mov     esi, [esp+0Ch+riid]
mov     edi, offset IID_IUnknown
mov     ecx, 4
repe cmpsd
jz      short _restoreEDIESI
```

```
mov     esi, [esp+0Ch+riid]
mov     edi, offset IID_IContextMenu
mov     ecx, 4
repe cmpsd
jz      short _restoreEDIESI
```

```
mov     esi, [esp+0Ch+riid]
mov     edi, offset IID_IShellExtInit
mov     ecx, 4
repe cmpsd
jnz     short _restoreEDIESI
```

```
lea     eax, [eax+4]
```

```
_restoreEDIESI:
pop     edi
pop     esi
jnz     short _ret_E_NOINTERFACE
```

```
mov     [ebx], eax
push    eax
mov     eax, [eax]
call    dword ptr [eax+IunknownVtbl.AddRef]
xor     eax, eax
jmp     short _locret
```

```
_ret_E_NOINTERFACE:
mov     eax, E_NOINTERFACE
```

```
_ret_E_POINTER:
mov     eax, E_POINTER
jmp     short _locret
```

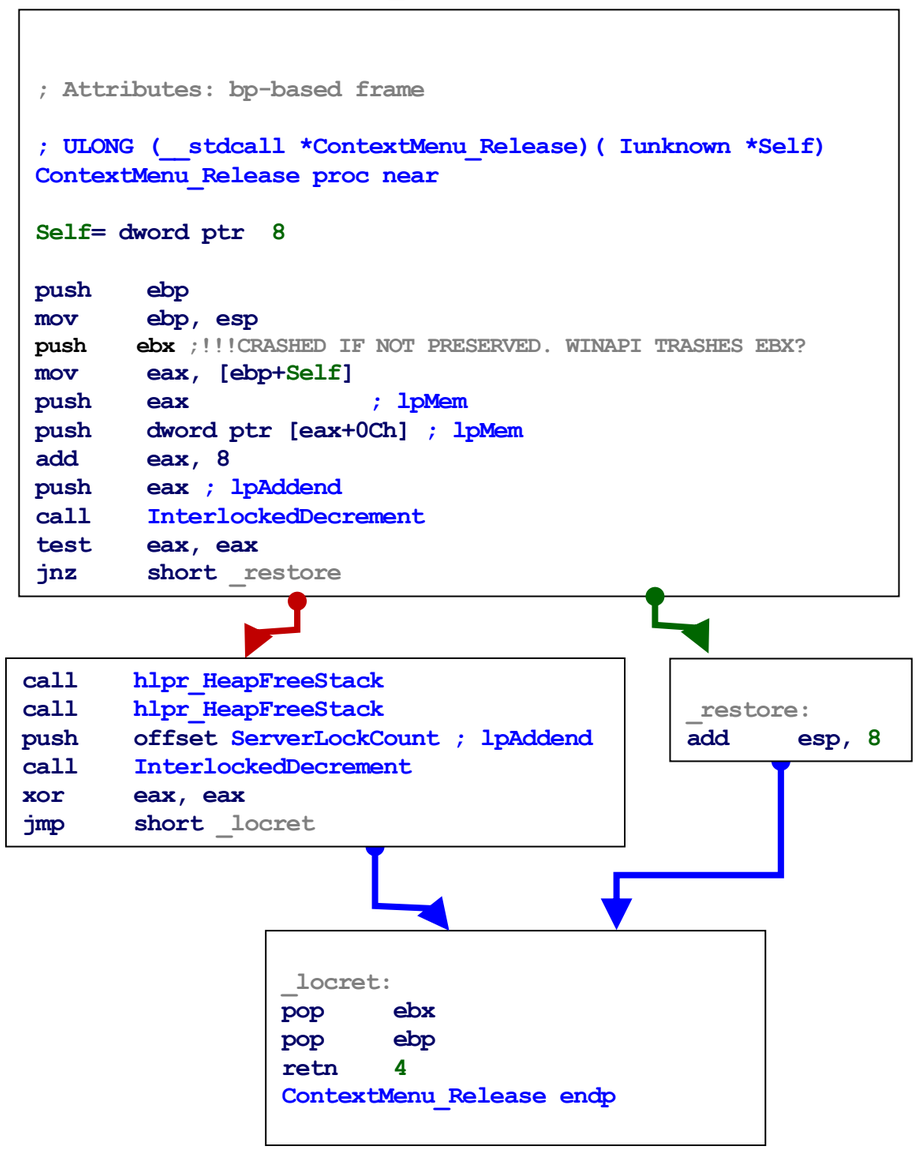
```
_locret:
pop     ebx
retn    0Ch
ContextMenu@QueryInterface endp
```

= Working HLL compiled COM Server =
Page 27

```
; Attributes: bp-based frame

; ULONG (__stdcall *ContextMenu_AddRef)( Iunknown *Self)
ContextMenu_AddRef proc near

Self= dword ptr  8

push    ebp
mov     ebp, esp
mov     eax, [ebp+Self]
add     eax, 8
push    eax ; lpAddend
call    InterlockedIncrement
pop     ebp
retn    4
ContextMenu_AddRef endp
```

```
; ULONG (__stdcall *ContextMenu_AddRef)( Iunknown *Self)
ContextMenu_AddRef proc near

Self= dword ptr  4

mov     eax, [esp+Self]
add     eax, 8
push    eax ; lpAddend
call    ds:InterlockedIncrement
retn    4
ContextMenu_AddRef endp
```

= Working HLL compiled COM Server =
Page 29



= Errornous Fasm compiled COM Server =

Page 30

```
; ULONG (__stdcall *ContextMenu_Release)( Iunknown *Self)
ContextMenu_Release proc near

Self= dword ptr  8
push    ebx ;!!! BELOW CODE DON`T TRASH IT, BUT WINAPI DOES?
mov     eax, [esp+Self
add     eax, 8
push    eax ; lpAddend
call    ds:InterlockedDecrement
test    eax, eax
jnz     short _restore
```

```
mov     eax, [esp+Self]
push    eax
push    dword ptr [eax+0Ch]
call    hlpr_HeapFreeStack
call    hlpr_HeapFreeStack
push    offset ServerLockCount ; lpAddend
call    ds:InterlockedDecrement
xor     eax, eax
```

```
_locret:
pop     ebx
retn    4
ContextMenu_Release endp
```

= Working HLL compiled COM Server = Page 31

```
; Attributes: bp-based frame

; HRESULT (__stdcall *ContextMenu_QueryContextMenu)(IContextMenu *Self, HMENU hmenu, UINT indexMenu, UINT idCmdFirst, UINT idCmdLast, UINT uFlags)
ContextMenu@QueryContextMenu proc near

hmenu= dword ptr  0Ch
indexMenu= dword ptr  10h
idCmdFirst= dword ptr  14h
uFlags= dword ptr  1Ch

push    ebp
mov     ebp, esp
mov     eax, [ebp+uFlags]
and     eax, 0Fh
jz      short _handle
```

```
test    eax, CMF_EXPLORE
jz      short _idle
```

```
_handle:
push    ebx
push    esi
push    edi
mov     ebx, [ebp+hmenu]
mov     esi, [ebp+indexMenu]
mov     edi, [ebp+idCmdFirst]
push    0               ; lpNewItem
push    0               ; uIDNewItem
push    MF_SEPARATOR or MF_BYPOSITION ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    InsertMenuA
inc     esi
push    ds:lplpRegMenu  ; offset lpRegMenu
push    edi             ; uIDNewItem
push    MF_BYPOSITION   ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    InsertMenuA
inc     esi
inc     edi
push    ds:lplpUnregMenu; offset lpUnregMenu
push    edi             ; uIDNewItem
push    MF_BYPOSITION   ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    InsertMenuA
inc     esi
push    0               ; lpNewItem
push    0               ; uIDNewItem
push    MF_SEPARATOR or MF_BYPOSITION ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    InsertMenuA
mov     eax, 2
pop     edi
pop     esi
pop     ebx
jmp     short _locret
```

```
_idle:
xor     eax, eax
```

```
_locret:
pop     ebp
retn    18h
ContextMenu@QueryContextMenu endp
```

```
; HRESULT (__stdcall *ContextMenu_QueryContextMenu)(IContextMenu *Self, HMENU hmenu, UINT indexMenu, UINT idCmdFirst, UINT idCmdLast, UINT uFlags)
ContextMenu@QueryContextMenu proc near

hmenu= dword ptr  8
indexMenu= dword ptr  0Ch
idCmdFirst= dword ptr  10h
uFlags= dword ptr  18h

mov     eax, [esp+uFlags]
and     eax, 0Fh
jz      short _handle
```

```
test    eax, CMF_EXPLORE
jz      short _idle
```

```
_idle:
xor     eax, eax
```

```
_handle:
push    ebx
push    esi
push    edi
mov     ebx, [esp+0Ch+hmenu]
mov     esi, [esp+0Ch+indexMenu]
mov     edi, [esp+0Ch+idCmdFirst]
push    0               ; lpNewItem
push    0               ; uIDNewItem
push    MF_SEPARATOR or MF_BYPOSITION ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    ds:InsertMenuA
inc     esi
push    offset lpRegMenu; "register"
push    edi             ; uIDNewItem
push    MF_BYPOSITION   ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    ds:InsertMenuA
inc     esi
inc     edi
push    offset lpUnregMenu; "unregister"
push    edi             ; uIDNewItem
push    MF_BYPOSITION   ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    ds:InsertMenuA
inc     esi
push    0               ; lpNewItem
push    0               ; uIDNewItem
push    MF_SEPARATOR or MF_BYPOSITION ; uFlags
push    esi             ; uPosition
push    ebx             ; hMenu
call    ds:InsertMenuA
mov     eax, 2
pop     edi
pop     esi
pop     ebx
jmp     short _locret
```

```
_locret:
retn    18h
ContextMenu@QueryContextMenu endp
```

= Working HLL compiled COM Server =
Page 33

```
; Attributes: bp-based frame

; int __stdcall hlpr_RegisterCOMServer(LPCSTR lpProcName, int hwnd, LPCSTR lpLibFileName)
hlpr_RegisterCOMServer proc near

lpProcName= dword ptr  8
hwnd= dword ptr  0Ch
lpLibFileName= dword ptr  10h

push    ebp
mov     ebp, esp
push    ebx
push    esi
push    edi
xor     ebx, ebx
mov     eax, [ebp+lpLibFileName]
push    eax             ; lpLibFileName
call    LoadLibraryA
mov     edi, eax
test    edi, edi
jz      short _locret
```

```
mov     eax, [ebp+lpProcName]
push    eax             ; lpProcName
push    edi             ; hModule
call    GetProcAddress
mov     esi, eax
test    eax, eax
jz      short _freelib
```

```
call    esi
test    eax, 80000000h
setz    bl
```

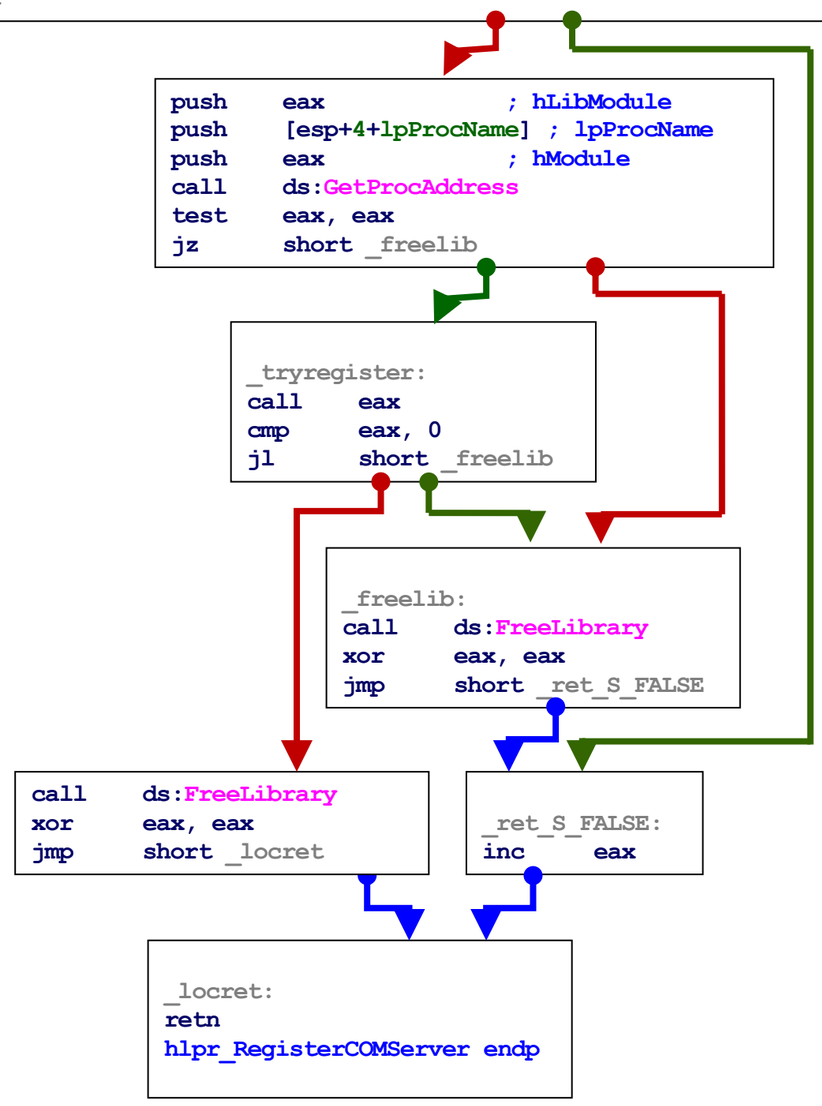
```
_freelib:
push    edi             ; hLibModule
call    FreeLibrary
```

```
_locret:
mov     eax, ebx
pop     edi
pop     esi
pop     ebx
pop     ebp
retn    0Ch
hlpr_RegisterCOMServer endp
```

= Errornous Fasm compiled COM Server =

Page 34

```
; int __stdcall hlpr_RegisterCOMServer(LPCSTR lpProcName, int hwnd, LPCSTR lpLibFileName)
hlpr_RegisterCOMServer proc near

lpProcName= dword ptr  4
hwnd= dword ptr  8
lpLibFileName= dword ptr  0Ch

push    [esp+lpLibFileName] ; lpLibFileName
call    ds:LoadLibraryA
test    eax, eax
jz      short _locret
```



= Working HLL compiled COM Server =
Page 35

```
; Attributes: bp-based frame

; int __stdcall hlpr_MessagingInvokeCommand(LPCSTR fmt, int fError, LPCSTR lpProcName, HWND hWnd, LPCSTR lpLibFileName)
hlpr_MessagingInvokeCommand proc near

fmt= dword ptr  8
fError= dword ptr  0Ch
lpProcName= dword ptr  10h
hWnd= dword ptr  14h
lpLibFileName= dword ptr  18h

push    ebp
mov     ebp, esp
push    [ebp+lpLibFileName]
push    [ebp+lpProcName]
push    [ebp+fmt]       ; LPCSTR
push    offset Buffer   ; LPSTR
call    wsprintfA
add     esp, 10h
mov     eax, [ebp+fError]
push    ds:MBCases[eax*4] ; uType
push    ds:QRCases[eax*4] ; lpCaption
push    offset Buffer   ; lpText
push    [ebp+hWnd]      ; hWnd
call    MessageBoxA
mov     eax, [ebp+fError]
pop     ebp
retn    14h
hlpr_MessagingInvokeCommand endp
```

```
MBCases:
dd MB_ICONINFORMATION or MB_OK
dd MB_ICONERROR or MB_OK
```

```
QRCases:
dd offset cptSuccessed ; "Quick Register"
dd offset cptFailed    ; "Quick Register - Error"
```

= Errornous Fasm compiled COM Server =
Page 36

```
; int __stdcall hlpr_MessagingInvokeCommand(LPCSTR fmt, int fError, LPCSTR lpProcName, HWND hWnd, LPCSTR lpLibFileName)
hlpr_MessagingInvokeCommand proc near

fmt= dword ptr  4
fError= dword ptr  8
lpProcName= dword ptr  0Ch
hWnd= dword ptr  10h
lpLibFileName= dword ptr  14h

push    [esp+lpLibFileName]
push    [esp+4+lpProcName]
push    [esp+8+fmt]     ; LPCSTR
push    offset Buffer   ; LPSTR
call    ds:wsprintfA
add     esp, 10h
mov     eax, [esp+fError]
push    MBCases[eax*4]    ; uType
push    QRCases[eax*4] ; lpCaption
push    offset Buffer   ; lpText
push    [esp+0Ch+hWnd]  ; hWnd
call    ds:MessageBoxA
mov     eax, [esp+fError]
retn    14h
hlpr_MessagingInvokeCommand endp
```

```
MBCases:
dd MB_ICONINFORMATION or MB_OK
dd MB_ICONERROR or MB_OK
```

```
QRCases:
dd offset cptSuccessed ; "Quick Register"
dd offset cptFailed    ; "Quick Register - Error"
```