

VB.NETからNAG Fortranライブラリ (Windows x64 DLL版) を利用する

Microsoft Visual Studio 2005 や 2008 をお持ちのユーザの方々はWindows XP/Vista/7 x64向けNAG Fortran ライブラリで提供されているDLL (FLW6I22DC_nag.d11 及び FLW6I22DC_mk1.d11) をVB.NET から使用できます。

このレポートは、VB.NET からNAGルーチンを使用する際に従わなくてはならないルールを記述しており、例を用いてそれらのルールを説明しています。

このレポートで参照されている例や Declare 文は、内蔵型の DLL、FLW6I22DC_nag.d11 を使用しています。同じ呼び出し規則を使用する別のDLL (例えば、FLW6I22DC_mk1.d11) からNAG Fortran ライブラリルーチンを呼び出す場合には、Declare 文の中の "FLW6I22DC_nag.d11" を利用したい DLL名に変更します。

なお、NAG DLL はパスに含まれている必要がありますのでご注意ください。もしユーザが FLW6I22DC_mk1.d11 などのMKLベースのDLLを使用している場合は、MKL DLLもパスに含まれている必要があります。詳細はユーザノートをご参照ください。

目次

- 1 - VB.NET から FLW6I22DC_nag.d11 のルーチンを呼び出すためのルール
- 2 - スカラーとコールバック関数を引数としてもつルーチンの呼び出し
- 3 - 配列を引数としてもつルーチンの呼び出し
- 4 - CHARACTER を引数としてもつルーチンの呼び出し

1 - VB.NET から FLW6I22DC_nag.d11 のルーチンを呼び出すためのルール

DLLに含まれるルーチンを呼び出すためには、Declare 文の中でそのルーチンを定義する必要があります。このDeclare 文には、ルーチン名、ルーチンを探すためのDLL名、そしてそのルーチンの引数の定義が含まれます。

NAG Fortran ライブラリのドキュメントは

<http://www.nag.co.uk/numeric/FL/FLdocumentation.asp> で参照いただけます。ドキュメントにはルーチン引数に関する必要な情報が全て含まれています。

これらの引数を定義する際には、以下に示すいくつかのルールに従う必要があります：

- 全てのスカラー値は参照渡し (ByRef) で渡されます。
- 全ての配列は値渡し (ByVal) で渡され、Fortran の側から要求される内容によって、以下の形式のうちの一つで表されます：<[In] ()>, <[Out] ()> あるいは <[In] (), Out [] ()>。
さらに、VB.NET と Fortran は同じ方法では配列をメモリに格納しません。(VB.NET が行方向であるのに対し、Fortran は列方向です) VB.NET のすべての配列はFortran サブプログラムを呼び出す前に、置き換える必要があります。
- Fortran が CHARACTER* 型のスカラー引数を要求する場合、実際のVB.NET 引数は値渡し (ByVal) で渡されるStringになります。この String の長さは引数リストの最後で渡される必要があります。
- Fortran が CHARACTER* 型の配列引数を要求する場合、実際のVB.NET 引数はすべての配列要素が連結された一つのStringになります。そのString は値渡し (ByVal) で渡されます。連結された一つの配列要素の長さは、引数リストの最後で渡される必要があります。
- Fortran が引数としてコールバック関数を要求する場合、デリゲートを定義することによってインターフェースがVB.NET側で宣言される必要があります。また、ルーチン引数はこのデリゲート型の引数として値渡し (ByVal) で渡されます。VB.NETからFortranルーチンを呼び出す場合、デリゲートインターフェースを実装している関数が、キーワード AddressOf を用いた引数として渡されます。

VB.NETにより認識されるこれらの宣言文に関して、以下の行がVB.NET ソースコードの先頭で必要になります：

```
Imports System.Runtime.InteropServices
```

この[zip file](#)にはすべてのライブラリルーチンの Declare 文が記述されたVB.NET モジュールが含まれています。

Intel Fortran コンパイラをインストールしていない場合は、この[zip file](#)に含まれるIntel Fortran Runtime DLL が必要となる場合があります。

ここに、VB.NET からの NAG Fortran ライブラリの呼び出し方を説明しているいくつかの例があります。これらの例についてのソースコードは[こちら](#)から参照いただけます。

2 - スカラーとコールバック関数を引数としてもつルーチンの呼び出し

ここで呼び出される NAG Fortran ライブラリルーチンは D01BDFです。NAG Fortran ライブラリのドキュメントによると、このルーチンは以下のFortran プロトタイプを持っています：

```
SUBROUTINE D01BDF(F, A, B, EPSABS, EPSREL, RESULT, ABSERR)
double precision      F, A, B, EPSABS, EPSREL, RESULT, ABSERR
EXTERNAL              F
```

倍精度値を返すEXTERNAL 関数である F を除いて、全ての引数はスカラーです。

詳しく説明しますと、NAG Fortran ライブラリドキュメントでは F の Fortranインターフェースは以下であると示しています：

```
double precision FUNCTION F(X)
double precision      X
```

まず最初に、このコールバック関数の VB.NET プロトタイプを宣言する必要があります。この場合、引数として倍精度値を返し、倍精度値を受け取るデリゲートになります。その宣言文は以下になります：

```
Delegate Function D01BDF_F_DELEGATE( _
    ByRef X As Double _
) as Double
```

ここでデリゲートについて学びましたので、D01BDFに対する Declare 文は簡単に書くことができます：

```
Declare Sub D01BDF Lib "FLW6I22DC_nag.dll" ( _
    ByVal F As D01BDF_F_DELEGATE, _
    ByRef A As Double, _
    ByRef B As Double, _
    ByRef EPSABS As Double, _
    ByRef EPSREL As Double, _
    ByRef RESULT As Double, _
    ByRef ABSERR As Double _
)
```

全てのスカラー引数は参照渡しで渡され、コールバック関数 F は デリゲート型の引数として値渡しで渡されます。

以下は、D01BDF を呼び出している短いVB.NETプログラムです：

```
Option Explicit On
Imports System.Math
```

```

Imports System.Text
Imports System.Runtime.InteropServices
Imports System.Reflection
Imports System.IO
Imports Microsoft.VisualBasic
Module Module1
Delegate Function D01BDF_F_DELEGATE( _
    ByRef X As Double _
    ) as Double
Declare Sub D01BDF Lib "FLW6I22DC_nag.dll" ( _
    ByVal F As D01BDF_F_DELEGATE, _
    ByRef A As Double, _
    ByRef B As Double, _
    ByRef EPSABS As Double, _
    ByRef EPSREL As Double, _
    ByRef RESULT As Double, _
    ByRef ABSERR As Double _
    )
Sub Main()
    Const A As Double = 0.0#, B As Double = 1.0#
    Dim epsabs As Double, epsrel As Double
    Dim myres As Double, abserr As Double
    Dim strResult As New StringBuilder
    epsabs = 0.0
    epsrel = 0.0001
    'THE KEYWORD AddressOf IS USED TO PASS FUN AS ARGUMENT
    Call D01BDF(AddressOf FUN, A, B, epsabs, epsrel, myres, abserr)
    strResult.Append("D01BDF Example Results" & vbCrLf)
    strResult.Append(vbCrLf & "Result of Integration" & vbCrLf)
    strResult.Append(myres)
    MessageBox.Show(strResult.ToString)
    strResult = Nothing
End Sub
Function FUN(ByRef x As Double) As Double
    FUN = x * x * Sin(10 * PI * x)
    Exit Function
End Function
End Module

```

3 - 配列を引数としてもつルーチンの呼び出し

ここで呼び出されるFortran ルーチンはG02BAFです。NAG ライブラリのドキュメントによると、このルーチンは以下の方法で宣言されます：

```
SUBROUTINE G02BAF(N, M, X, LDX, XBAR, STD, SSP, LDSSP, R, LDR, IFAIL)
INTEGER N, M, LDX, LDSSP, LDR, IFAIL
double precision X(LDX,M), XBAR(M), STD(M), SSP(LDSSP,M), R(LDR,M)
```

また、NAG ライブラリのドキュメントによると、配列に対するINTENT 属性は以下のとおりです：

- X INTENT (IN)
- XBAR INTENT (OUT)
- STD INTENT (OUT)
- SSP INTENT (OUT)
- R INTENT (OUT)

これらのことから、G02BAFに対して以下のDeclare 文を決定することができます：

```
Declare Sub G02BAF Lib "FLW6I22DC_nag.dll" ( _
    ByRef N As Integer, _
    ByRef M As Integer, _
    <[In]()> ByVal X As Double(,), _
    ByRef LDX As Integer, _
    <[Out]()> ByVal XBAR As Double(), _
    <[Out]()> ByVal STD As Double(), _
    <[Out]()> ByVal SSP As Double(,), _
    ByRef LDSSP As Integer, _
    <[Out]()> ByVal R As Double(,), _
    ByRef LDR As Integer, _
    ByRef IFAIL As Integer _
)
```

ここで、Fortran INTENT 属性がVB.NET ソースコードの中で<[IN]()> あるいは <[OUT]()>で指定されています。Fortran INTENT 属性がINTENT(INOUT)であった場合、対応する VB.NET ソースコードは<[IN](), OUT[]()> になります。

以下は、Fortran サンプルプログラム g02baf.f を変換したもので、VB.NETからG02BAF を使用しています。

```
Option Explicit On
Imports System.Math
Imports System.Text
Imports System.Runtime.InteropServices
Imports System.Reflection
Imports System.IO
```

```
Imports Microsoft.VisualBasic
```

```
Module Module1
```

```
Declare Sub G02BAF Lib "FLW6I22DC_nag.dll" _
    (ByRef N As Integer, ByRef M As Integer, _
     <[In]()> ByVal X As Double(), _
     ByRef LDX As Integer, _
     <[Out]()> ByVal XBAR As Double(), _
     <[Out]()> ByVal STD As Double(), _
     <[Out]()> ByVal SSP As Double(), _
     ByRef LDSSP As Integer, _
     <[Out]()> ByVal R As Double(), _
    ByRef LDR As Integer, ByRef IFAIL As Integer)
Sub Main()
Dim M As Integer = 3
Dim N As Integer = 5
Dim IA As Integer = N
Dim ISSP As Integer = M
Dim ICORR As Integer = M
Dim I As Integer
Dim IFAIL As Integer
Dim J As Integer
Dim A(M - 1, IA - 1) As Double
Dim AMEAN(M - 1) As Double
Dim CORR(M - 1, ICORR - 1) As Double
Dim SSP(M - 1, ISSP - 1) As Double
Dim STD(M - 1) As Double
'START DATA INITIALISATION - THE ARRAY A IS TRANSPOSED
A(0, 0) = 2.0
A(0, 1) = 4.0
A(0, 2) = 9.0
A(0, 3) = 0.0
A(0, 4) = 12.0
A(1, 0) = 3.0
```

```

A(1, 1) = 6.0
A(1, 2) = 9.0
A(1, 3) = 12.0
A(1, 4) = -1.0
A(2, 0) = 3.0
A(2, 1) = 4.0
A(2, 2) = 0.0
A(2, 3) = 2.0
A(2, 4) = 5.0
'END DATA INITIALISATION
Console.WriteLine("G02BAF Example Program Results")
Console.WriteLine("Number of variables (columns) = " & M)
Console.WriteLine("Number of cases (rows) = " & N)
Console.WriteLine()
Console.WriteLine("Data matrix is:")
For J = 1 To M
    Console.Write(" " & J)
Next
Console.WriteLine()
For I = 0 To N - 1
    For J = 0 To M - 1
        Console.Write(Format(A(J, I), "###0.0") & " ")
    Next
    Console.WriteLine()
Next
IFAIL = 1
'CALL TO THE NAG LIBRARY ROUTINE
Call G02BAF(N, M, A, IA, AMEAN, STD, SSP, ISSP, CORR, ICORR, IFAIL)
If (IFAIL <> 0) Then
    Console.WriteLine("Routine fails, IFAIL = " & IFAIL)
Else
    Console.WriteLine("Variable Mean St. dev.")
    For I = 0 To M - 1
        Console.Write(" " & Format((I + 1), "###0"))
        strResult.Append(" ")
        Console.Write(Format(AMEAN(I), "###0.000"))
        strResult.Append(" ")
        Console.WriteLine(Format(STD(I), "###0.000"))
    Next
    Console.WriteLine()
    Console.WriteLine _

```

```

        ("Sums of squares and cross-products of derivations")
    For I = 0 To M - 1
        For J = 0 To M - 1
            Console.WriteLine(Format(SSP(J, I), "###000.00") & " ")
        Next
        Console.WriteLine()
    Next
    Console.WriteLine("Correlation coefficients")
    Console.WriteLine()
    For I = 0 To M - 1
        For J = 0 To M - 1
            Console.WriteLine(Format(CORR(J, I), "###0.000") & " ")
        Next
        Console.WriteLine()
    Next
    End If
    Console.ReadLine()
End Sub
End Module

```

この結果は、NAG Fortranライブラリで提供される結果あるいはNAG Fortranライブラリのドキュメントに記載されている結果と比較することができます。

4 - CHARACTER を引数としてもつルーチンの呼び出し

ここで学ぶNAG Fortran ライブラリルーチンは M01CCF です。NAG Fortranライブラリのドキュメントによると、このルーチンの Fortran プロトタイプは以下になります：

```

SUBROUTINE M01CCF(CH, M1, M2, L1, L2, ORDER, IFAIL)
INTEGER M1, M2, L1, L2, IFAIL
CHARACTER*(*) CH(M2)
CHARACTER*1 ORDER

```

次に、対応する Declare 文を定義することができます：

```

Declare Sub M01CCF Lib "FLW6I22DC_nag.dll" ( _
    ByVal CH As string, _
    ByRef M1 As Integer, _
    ByRef M2 As Integer, _
    ByRef L1 As Integer, _

```



```

ByRef L2 As Integer, _
ByVal ORDER As string, _
ByRef IFAIL As Integer, _
ByVal CHLength As Integer, _
ByVal ORDERLength As Integer _
)

```

このDeclare文では、String 型の二つの引数があります。一つめは、CHARACTER*(*) のFortran 配列で、二つめはCHARACTER*1 の単一のFortran 変数です。受け渡しされる引数の長さは、引数リストの最後に引数が現れている順番で定義されています。

引数リストで最初に現れている長さは、CH の中の連結された一つの配列要素の長さで、2番目に現れている長さはORDERの実際の長さです。

以下は、このDeclare文を使用した例です：

```

Option Explicit On
Imports System.Math
Imports System.Text
Imports System.Runtime.InteropServices
Imports System.Reflection
Imports System.IO
Imports Microsoft.VisualBasic
Module module1
Declare Sub M01CCF Lib "FLW6I22DC_nag.dll" ( _
    ByVal CH As string, _
    ByRef M1 As Integer, _
    ByRef M2 As Integer, _
    ByRef L1 As Integer, _
    ByRef L2 As Integer, _
    ByVal ORDER As string, _
    ByRef IFAIL As Integer, _
    ByVal CHLength As Integer, _
    ByVal ORDERLength As Integer _
)
Sub Main()
    Dim ch(10) As String
    Dim ch_len As Integer
    Dim i As Integer, ifail As Integer, l1 As Integer, l2 As Integer, m As
Integer

```

```

Dim strResult As New StringBuilder
Dim ch_all As String
ch(0) = "A02AAF 289"
ch(1) = "A02ABF 523"
ch(2) = "A02ACF 531"
ch(3) = "C02ADF 169"
ch(4) = "C02AEF 599"
ch(5) = "C05ADF 1351"
ch(6) = "C05AGF 240"
ch(7) = "C05AJF 136"
ch(8) = "C05AVF 211"
ch(9) = "C05AXF 183"
ch(10) = "C05AZF 2181"

m = 11
l1 = 7
l2 = 12
ifail = 1
ch_all = ""
'THE ARRAY TO SORT IS CONCATENATED IN ONE STRING
For i = 0 To m - 1
    ch_all = ch_all & ch(i)
Next
'ch_len IS THE LENGTH OF ONE ELEMENT OF CH
ch_len = 12
M01CCF(ch_all, 1, m, l1, l2, "Reverse ASCII", ifail, ch_len, 13)
For i = 0 To m - 1
    ch(i) = ch_all.Substring(i * ch_len, ch_len)
Next
strResult.Append("M01CCF Example Results" & vbCrLf & vbCrLf)
strResult.Append("Records sorted on columns " & l1.ToString)
strResult.Append(" to " & l2.ToString & vbCrLf & vbCrLf)
For i = 0 To m - 1
    strResult.Append(ch(i).ToString & vbCrLf)
Next
strResult.Append(vbCrLf & "ifail on exit = " & _
ifail.ToString & vbCrLf)
MessageBox.Show(strResult.ToString)
strResult = Nothing
End Sub
End Module

```