〈VBA 簡易構文解析機能仕様〉

本書は、エクセル VBA で作成したプログラム(モジュールファイル)の構文解析を実行するために、必要な基本的な機能について仕様を纏めるものである。このような構文解析のことを一般的にパース処理と呼ぶことがありますが、以降、本書では、この構文解析のことをパース処理と表記する。

〈基本仕様〉

- パース処理は、入力データである VBA プログラムコードを解析し、出力結果として、プログラムコードを構成している要素群を、メモリ上に取込みデータ化する機能のことである。ここで言う、要素とは、定数、列挙値、構造体、API 宣言、関数、変数などのことである。また、その出力データを利用したアプリケーションが容易に作成できるよう汎用的なデータ構造となるよう設計する。
- パース処理の入力データは、VBA プログラムが記述されたモジュールと、それらがインポートされているエクセルブック・ファイルである。
- パース処理機能は、**エクセル**により開発された VBA プログラムを対象とする。
- パース処理機能は、エクセルの VBA 開発環境よりエクスポートされたモジュールを対象とする。また、その対象モジュールは、開発環境にて、コンパイルが正常に終了していることが、パース処理が、正常に動作する前提条件とする。**コードの記述されていない、ThisWorkbook オブジェクトモジュールや、Sheet オブジェクトモジュールは不要**です。
- モジュールファイルの文字コード体系は SJIS のみを対象とする(エクセルでエクスポートすると文字コード体系が SJIS のモジュールファイルが出力されるため)。
- 対象となるモジュールファイルは、拡張子が以下のファイルとする。但し、クラスモジュールは、ブックオブジェクトの ThisWorkbook と、名前が Sheet で始まる(例 ： Sheet1、Sheet9　)シートオブジェクトのみを対象とし、**ユーザ定義のクラスモジュールについては、今回の、パース処理の対象からは除外**する仕様とする。
    a) .bas ： 標準モジュール
    b) .frm ： フォームモジュール
    c) .cls ： クラスモジュール
- ソースコードのパース処理を容易にすることを目的とし、最初にコードデータを加工するフィルタ処理を施すものとする。このフィルタ処理の詳細については後述する。
- 定数(列挙値含む)定義の、演算式の展開による定数の演算結果としての値算出は、対応しないものとする。但し、列挙値のオートインクリメント機能による定数値については、対応する。
- パース処理にて、切り出された単語の種別を判定するため以下のマスタデータを使用する。
    a) システムグローバル変数マスタ
        例 ： ThisWorkbook、Me、Application など
    b) VBA の予約語マスタ
        例 ： If、For、GoTo、With など
    c) VBA の標準関数マスタ
        例 ： StrConv、InStr、Lbound、Ubound など
    d) VBA の標準データ型マスタ
        例 ： Long、Integer、Workbook、WorkSheet など
    e) Me によりアクセスできるメソッド/プロパティの IF マスタ
        ThisWorkbook、Sheet1～n、フォームの 3 種
- vbRed や vbCrLf、msoTrue などの VBA の標準定数の判別には、vb や mso などの定数の小文字の接頭辞により判定し、定数マスタは使用しない仕様とする。但し、必要性があると考えられる場合は、定数マスタを利用する仕様を検討する。Nothing や Empty などの特殊な定数はハードコーディングにより対応する。
- コード内の、構造体/クラスオブジェクトのピリオドを含む参照は、先頭の単語のみを変数チェックし、2 つ目のピリオド以降の参照は、メンバ(変数、プロパティ、メソッド)参照扱いとする。また

最初の単語が変数宣言に見つからない場合、グローバルオブジェクトとして、システムグローバル変数マスタに自動で登録する機能を有するものとする。

例 ： メンバ参照のサンプル

clss_object.ArrayObj(idx).Func(arg1, arg2)

変数参照____メンバ参照____メンバ参照

- 変数宣言や関数の戻り値の型などのデータ型のパース処理にて、判別できなかったデータ型が存在した場合、自動で、VBA の標準データ型マスタへ登録する機能を有するものとする。
- 複数の命令を 1 行に記述する場合に、使用するコロン(:)は、改行コード(CR+LF)に変換する。従って、パース処理の結果として、メモリ上に取り込まれるデータには、このコロンの情報は存在しません。但し、ラベル名の末尾付与するコロンは要素情報に含まれる。
- コード内の、括弧の対応を確認できるよう、パース処理にて出力データを作成する。
- コード内の、ピリオドを含む参照は、参照情報として、参照の階層、深度を確認できるよう、パース処理にて出力データを作成する。但し、参照コードに含まれる、コンマや括弧については、この参照情報は保存しないものとする。

例 ： 参照の階層と深度(1 行のデータ)

コード; clss.Property.Method(st.a(st.b), st.c)

階層:　1_____2____3_____2____

深度:　1____2_____3_____

- モジュール内のコメントは以下の規則に従い管理する。シングルクォーテーションと Rem 文のいずれのコメントにも対応するものとする。

  a)　コメントのみの行を、行インデックスにより検索管理できるものとする。

  b)　コードの後ろに続くコメントは、コードのパース処理時にチェックし、必要に応じて保存する。

- モジュールの依存関係を抽出できるものとする。依存関係の抽出は、第 1 段階までとする。例えば、A モジュール内で、呼び出されている関数が実装されているモジュールが B モジュールだとする。これは、A モジュールが B モジュールに依存しているということである。但し、B モジュールの該当関数内で、参照している要素がどのモジュールに依存しているかまでは、A モジュールの依存関係としては、パース処理の対象外とする。
- Windows API 宣言の依存関係は、API がエクスポートされている DLL ファイルに依存しているものとする。
- #Const、#If、#ElseIf、#Else、#End If による**条件付きコンパイルを利用したプログラムコードのパースには、今回は対応しておりません**。

例 ： 条件付きコンパイルを利用したコードのサンプル

```
#Const Debug = True

'構造体の定義
Private Type StructSample
        str As String
        vl As Variant
        ary() As Long
#If Debug Then
        dbg As String
        is_debug As Boolean
#Else
        is_release As Boolean
#End If

EndType
```

<モジュールのデータ構造>
　モジュールファイル内の先頭に記述されている以下のデータ部をヘッダ部とし、開発環境にて記述されたプログラム本体をボディ部と定義する。

例 ： モジュールファイルの中身(ヘッダ部)

```
VERSION 1.0 CLASS
BEGIN
   MultiUse = -1  'True
END
Attribute VB_Name = "VbaParse"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
```

　モジュールのボディ部の最初のコメントを自身のモジュールに対するコメントして扱うものとする。このコメントの終了は、コメントの開始から、何かしらのプログラムコード、もしくは、ブランク行を検出した際に、コメントの終了行とする。すなわち、コメントの開始行からコメントとして連続した領域のコメント群のことである。

<フィルタ処理>
　フィルタ処理は、モジュールファイルの中身のデータを加工し、後のパース処理を効率的かつ容易にすることを目的とした、ソースデータ変換処理のことである。

- ソースコード内に、存在するアンダースコア(_)を削除する。但し、データ内とコメント内のアンダースコアは対象外とする。これにより、複数行にまたがるソースコードが一行のソースコードデータとなる。
- 文字列データ内の連続したダブルクォーテーションは、2 個で 1 個のダブルクォーテーション(以降、DQ と表記する)にする。文字列データの開始と終了の DQ は削除し、開始の DQ は、$st;に変換し、終了の DQ は削除する。
- 文字列データ内の半角 SP は、$sp;に変換する。
- 文字列データ内の半角コンマは、$cm;に変換する。
- 文字列データ内の括弧の開始・終了は、それぞれ、$bs;、$be;に変換する。
- SP とタブ文字が連続している場合、SP1 個に変換する。但し、文字列データ内とコメント内は対象外とする。
- 行頭からの SP とタブ文字は、全て削除する。
- Rem ステートメントによるコメントは、シングルクォーテーション(SQ)に変換する。
- 括弧の開始と SP が連続している場合は、後の SP を削除する。Ex. Sub UserFunc(_
- 複数の命令を 1 行に記述するための:と SP が連続している場合は、改行(CR+LF)に変換する。
  例 ： a = 100: b = a + 100: c = a * b

※フィルタ処理により、プログラム内部で管理しているコメント行のインデックスは、ソースコード上の行数と一致していません。

<データ構造>
　パース処理で利用している主なデータ構造について簡単に以下に説明する。

・ヘッダ部を構成している以下のようなバージョン情報は、単純に String 型に格納する。

例 ： バージョン情報のサンプル

```
VERSION 1.0 CLASS
```

・ヘッダ部を構成している以下のようなBEIGN情報は、BEGINとENDで囲まれた範囲内に複数のデータが存在する可能性があるが、内部のデータを今回の仕様では、特に使用しないので、複数の定義値を単純に１つのString型に格納する。

例 ： BIGIN情報のサンプル

```
BEGIN
   MultiUse = -1  'True
END
```

・ヘッダ部のAttribute情報は、以下の構造体に格納する。

例 ： Attribute情報のサンプル

```
Attribute VB_Name = "VbaParse"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
```

```
'Attribute情報構造体
Public Type StructAttributeInfo
       name As String                  '属性名
       vl As String                    '属性値
       is_str As Boolean               '文字列値/数値
End Type
```

・ボディ部のモジュール内に記述されているコメントのみの行データは、以下のコメント情報構造体にデータを格納する。

例 ： コメントのサンプル

```
 Rem mdl_code_モジュール

 ‘このモジュールは、VBAパース処理を行なうために、
 ‘効率的かつ、パース処理を容易にするため、先にフィルタ処理を
 ‘行なうための関数群を実装したモジュールである。
```

```
'コメント情報構造体
Public Type StructCommentInfo
       str As String                   'コメント文字列
       is_valid As Boolean             'True:コメント、False:コメントでない。
End Type
```

・ボディ部のモジュール定義部に記述されているWindwosAPI宣言情報は以下の構造体に格納する。

例 ： API宣言情報のサンプル

```
Public Declare Sub DllRtlMoveMemory Lib "kernel32.dll" Alias "RtlMoveMemory" _
                                (ByVal dst As LongPtr, _
                                ByVal src As LongPtr, _
                                ByVal lgth As Long)         'メモリコピー

'MP3の再生に対応しているAPI
Declare Function DllMciSendString Lib "winmm.dll" Alias "mciSendStringW" _
                                (ByVal lpszCommand As LongPtr, _
                                ByVal lpszReturnString As LongPtr, _
                                ByVal cchReturn As Long, _
```

```
                                        ByVal hwndCallback As Long) As Long
```

```
'WinAPI情報構造体
Public Type StructWinApiInfo
        name As String                          'VBAでのWinAPI呼び出し名
        dll As String                           'ライブラリ名(dllファイル)
        alias As String                         'WinAPI名
        args() As StructArgumentInfo            '引数情報
        args_num As Long                        '引数情報数
        typ As String                           '戻り値のデータ型
        modify As String                        'Sub/Function
        is_sub As Boolean                       'Subかどうか
        access As String                        'アクセス性 Public,Private
        is_public As Boolean                    'Publicかどうか
        cmnt_r As String                        'コメント(右側)
        cmnt_t As String                        'コメント(上側行頭)
End Type
```

・ボディ部のモジュール定義部に記述されている定数と列挙定数値情報は以下の構造体に格納する。

例 : 定数情報のサンプル

```
        Private Const VBA_END_WITH = VBA_END & S_SP &  VBA_WITH
        Public Const CHR_SP = &H20
```

```
'定数情報構造体(列挙定数と兼用)
Public Type StructConstInfo
        name As String                          '定数名
        vl_str As String                        '定数値(演算子含む文字列ソース)
        vl_numeric As Variant                   '定数値(v_strを展開した演算結果の数値)
        typ As String                           'データ型
        access As String                        'アクセス性 Public,Private
        is_public As Boolean                    'Publicかどうか
        is_calc As Boolean                      '演算式を含んだ定数かどうか
        cmnt_r As String                        'コメント(右側)
        cmnt_t As String                        'コメント(上側行頭)
End Type
```

・ボディ部のモジュール定義部に記述されている列挙値情報は以下の構造体に格納する。

例 : 列挙値情報のサンプル

```
        'モジュール読込み時のエラー列挙値
        Public Enum EnumModuleReadError
                EMRE_FILE_IO = -100             'ファイルI/Oエラー
                EMRE_EXT                        '対象外の拡張子
                EMRE_NO_FILE                    'ファイルが存在しない
                EMRE_ZERO_SIZE                  'ファイルのサイズが0
                EMRE_NO_ERROR = 0               '正常な場合、読み込んだサイズを返却
        End Enum
```

```
'列挙値情報構造体
Public Type StructEnumInfo
        name As String                          '列挙値名
        cnsts() As StructConstInfo              '列挙値(定数)情報
        cnsts_num As Long                       '列挙値数
        access As String                        'アクセス性 Public,Private
        is_public As Boolean                    'Publicかどうか
        cmnt_r As String                        'コメント(右側)
        cmnt_t As String                        'コメント(上側行頭)
End Type
```

・ボディ部のモジュール定義部や関数コード内にに宣言されている変数情報は、以下の構造体に格納する。

例 : 変数情報のサンプル

```
        Private BufNum As Long
        Dim str As String
        Dim cls As New CParse
        Static bufs(1024 - 1) As Byte

'変数情報構造体
Public Type StructVariableInfo
        name As String                          '変数名
        'vl_str As String                       '定数値文字列(コンパイル条件付き)
        'vl_numeric As Variant                  '定数値(コンパイル条件付き)
        typ As String                           'データ型
        sz_dim1 As String                       '配列内の括弧内の添え字 サイズ1次元目
        sz_dim2 As String                       '配列内の括弧内の添え字 サイズ2次元目
        access As String                        'アクセス性 Public,Private
        sub_inf As Variant                      '補足情報:配列の括弧を含む文字列など
        kind_ary As EnumArrayVariableKind       '配列の種類
        is_public As Boolean                    'Publicかどうか
        cmnt_r As String                        'コメント(右側)
        cmnt_t As String                        'コメント(上側行頭)
End Type
```

・ボディ部の関数のIF内にに宣言されている引数情報は、以下の構造体に格納する。

例 : 引数情報のサンプル

```
        Public Function VpParseVBA(ByRef prs_inf As StructParseInfo) As Long


'引数情報構造体
Public Type StructArgumentInfo
        name As String                          '引数名
        typ As String                           'データ型
        modify As String                        'ByVal,ByRef,Optional,省略
        sub_inf As Variant                      '付加的な情報:括弧付きの配列名など
        is_array As Boolean                     '配列かどうか
        vl_dflt As String                       '省略した場合の値
        is_optional As Boolean                  '省略可能かどうか
End Type
```

・ボディ部のモジュール内に記述されている関数情報は、以下の構造体に格納する。関数ソースコード内の1行ずつに分割した個々のデータがStructCodeLineである。また、その1行ごとのデータをさらに詳細に要素ごとに分割したものをStructCodeElementInfoに格納する仕様。

例 : 関数コードのサンプル

```
        'ファイルのフルパスからファイル名を取得。
        Public Function FILE_NAME(ByVal fname As String) As String
                Dim name As String
                Dim lgth As String
                Dim s As String
                Dim i As Long

                lgth = Len(fname)
                For i = lgth To 1 Step -1
                        s = Mid(fname, i, 1)
                        If s = "¥" Then
                                FILE_NAME = Right(fname, lgth - i)
                                Exit Function

                        End If

                Next i

                FILE_NAME = ""

        End Function


'プログラムコード1行単位情報
Public Type StructCodeLine
        line As String                          'コードライン1行
```

```
        is_skip As Boolean                  'スキップするかどうか
End Type


'関数コードを構成している要素情報
Public Type StructCodeElementInfo
        elmt As String                      '定数/関数/変数名などのコード構成要素名
        kind As EnumCodeElementKind         'コード構成要素種別
        typ As String                       'データ型
        idx_brckt As Long                   '括弧の情報
        idx_vba_word As Long                'VBAの予約語インデックス
        mem_inf(1) As Byte                  '構造体/クラスメンバの階層と深度情報

        'Ex.     clss.Property.Method(st.A, st.B)
        'layer:  1_____________________2___1_2___1

        'Ex.     clss.Property.Method
        'depth:  1____2________3_____

        is_public As Boolean                'Public/Private
        sub_inf As Variant                  '付加的な情報
        note As String                      '備考
End Type

'関数情報構造体
Public Type StructProcedureInfo
        name As String                      '関数名
        args() As StructArgumentInfo        '引数情報
        args_num As Long                    '引数情報数
        typ As String                       '戻り値のデータ型
        modify As String                    'Sub/Function
        is_sub As Boolean                   'Subかどうか
        access As String                    'アクセス性 Public,Private
        is_public As Boolean                'Publicかどうか
        vars() As StructVariableInfo        'ローカル変数情報
        vars_num As Long                    'ローカル変数情報数
        src_rng As StructRangeIndex         '関数内のソースコード範囲
        elmts() As StructCodeElementInfo    'コードを構成している要素のリスト
        elmts_num As Long                   'コードを構成している要素数
        lines() As StructCodeLine           'ソースコード行単位情報
        lines_num As Long                   'ソースコード行単位情報の数
        cmnt_r As String                    'コメント(右側)
        cmnt_t As String                    'コメント(上側行頭)
End Type
```

・関数、定数、列挙値、構造体などが記述されたモジュールの情報は、以下の構造体に格納する。

```
'モジュール情報構造体
Public Type StructModuleInfo
        head As StructModuleHeaderInfo      'ヘッダ情報
        name As String                      'モジュール名
        file As String                      'モジュールファイル名
        kind As EnumModuleFileKind          'モジュールファイル種別
        cnsts() As StructConstInfo          '定数情報
        cnsts_num As Long                   '定数情報数
        enums() As StructEnumInfo           '列挙値情報
        enums_num As Long                   '列挙値情報数
        vars() As StructVariableInfo        '変数情報(グローバル)
        vars_num As Long                    '変数情報数(グローバル)
        typs() As StructTypeInfo            '構造体情報
        typs_num As Long                    '構造体情報数
        prcs() As StructProcedureInfo       '関数情報
        prcs_num As Long                    '関数情報数
        apis() As StructWinApiInfo          'WinAPI宣言情報
        apis_num As Long                    'WinAPI宣言数
        cmnts_t() As StructCommentInfo      'コメント情報(行頭:コードの右側以外のもの)
        cmnts_t_num As Long                 'コメント情報数
        cmnt_t As String                    'コメント情報(モジュールの先頭)
        dpds() As String                    '依存しているモジュール名のリスト
        dpds_num As Long                    '依存しているモジュール名のリストの要素数
        is_user_class As Boolean            'ユーザ定義のクラスモジュール
```

```
        is_option_explicit As Boolean           'Option Explict ステートメント
        is_parsed As Boolean                    'パースの成功/不成功

        objs() As StructOleObjectInfo           'シートの OLE オブジェクト情報
        objs_num As Long                        'シートの OLE オブジェクト数

        '*** ソースコード管理情報 ***
        bufs() As Byte                          'ソースコードバッファ(フィルタ処理済)
        bufs_num As Long                        'ソースコードバッファサイズ
        bin_bufs() As Byte                      'フォームのバイナリデータバッファ
        bin_bufs_num As Long                    'フォームのバイナリデータバッファサイズ
        ln_idxs()   As Long                     '行インデックス情報
        idx_cur As Long                         '現在位置
        idx_prev As Long                        '一つ前の現在位置
End Type
```

・パース処理の入力や、パース処理結果となる出力データを格納するデータが以下の構造体である。

```
'VBA ソースコード解析情報構造体
Public Type StructParseInfo
        mdls() As StructModuleInfo              'モジュールリスト
        mdls_num As Long                        'モジュールリストの要素数
        pubs() As StructPublicInfo              'Public 関数/変数/定数のリスト
        pubs_num As Long                        'Public 関数/変数/定数の要素数
        frms() As String                        'フォームオブジェクトの名前情報
        frms_num As Long                        'フォームオブジェクトの数

        '*** シート名情報 ***
        shts() As StructSheetName               'シート名情報
        shts_num As Long                        'シート名情報の数

        '*** ブック情報 ***
        bk_name As String                       'ブック名

        '** パース処理と結果出力設定 **
        is_out_comment As Boolean               'コメントの出力設定
        is_parse_elmt As Boolean                '要素情報のパース設定
        is_add_vba_proc As Boolean              '不明単語を VBA 標準関数として登録
        is_add_vba_glob As Boolean              '不明単語を VBA グローバル変数として登録
        is_add_dat_type As Boolean              '不明単語をデータ型として登録
End Type
```

<StructCodeElementInfo の sub_inf について>

StructCodeElementInfo 構造体の sub_inf は Variant 型であるが、要素に応じて、様々なデータを格納している。それらについて、ここで、簡単に記述する。

- 補足情報としての文字列データ
- 要素が配列の場合は、括弧付きの配列名 Ex. Ary() → elmt=”Ary”、sub_inf=”Ary()”
- 要素が括弧の場合は、括弧の終了に対する括弧の開始位置(要素インデックス)
- 要素が VBA の予約語場合は、4 バイトの Long 型の値として使用。予約語に該当するマスタデータにおけるインデックスを下位 2 バイトに、上位 2 バイトに、さらなる付加的な情報として値を格納。例えば、End における If であること指し示すビット情報など。
- 要素が VBA のシステムグローバルな変数の場合は、該当データのマスタデータにおけるインデックス。
- 要素が VBA の標準関数の場合は、該当データのマスタデータにおけるインデックス。

以上