

ENSAMBLADO .NET

VPARMT

Manual de referencia

VERSIÓN 3.1

Índice de contenido

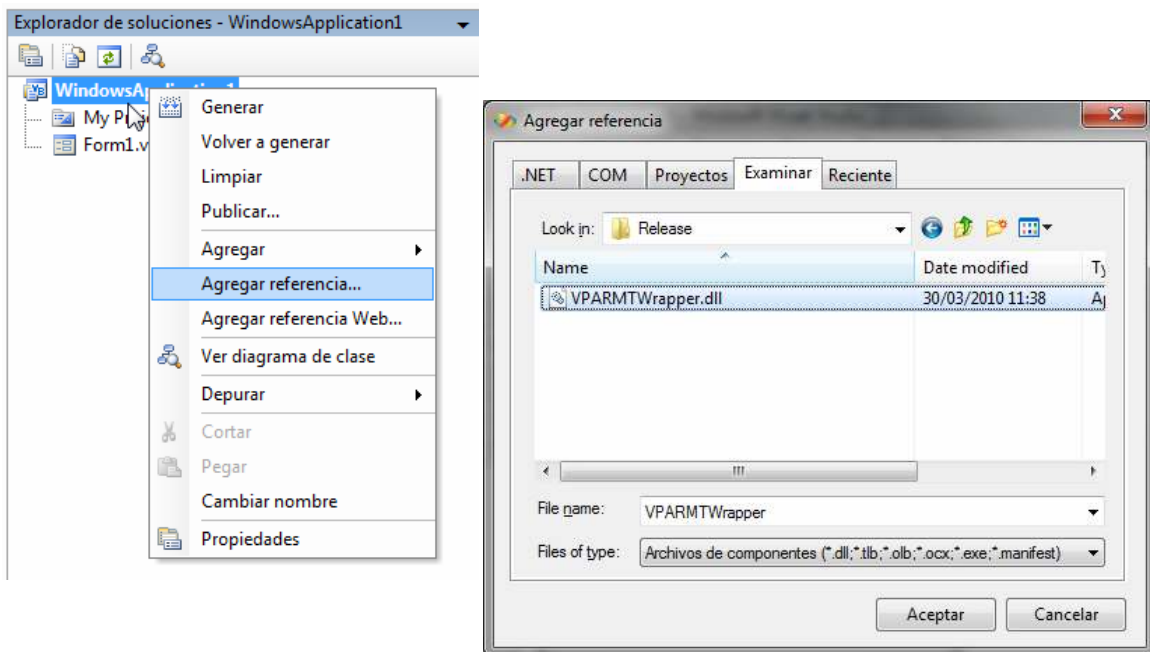
ÍNDICE DE CONTENIDO	2
INTRODUCCIÓN	3
DECLARACIÓN / CONFIGURACIÓN	4
CONFIGURACIÓN.....	4
INICIALIZACIÓN / FINALIZACIÓN	6
INIT.....	6
QUERYEND.....	8
SOLICITUD DE LECTURA DE MATRÍCULAS	9
ADD	9
CONSULTAS DISPONIBLES	12
GETQUEUESIZE	12
ISRUNNING	13
ISPROCESSING	14
GETLICENSEDCORES	15
GETFREECORES.....	16
OBTENCIÓN DE RESULTADOS	17
GETNUMBEROFPLATES.....	17
GETFIRSTPLATE	18
GETPLATE	20
RESULTADOS: MÉTODOS AVANZADOS	21
GETFIRSTELEMENT	21
ANEXO 1 – CÓDIGOS DE PAÍS	23
APPENDIX 2 – ESQUEMA MULTI-PROCESADOR	24
VERSIÓN SIMPLE	24
VERSIÓN DETALLADA	25

Introducción

El presente documento es el manual de referencia del ensamblado “VPMRTWrapper” que encapsula de una forma sencilla para su uso desde .NET las funcionalidades de la librería VPMRT (Vehicle Plate Automatic Reader Multi Thread) capaz de ejecutar hasta 8 procesos simultáneos de reconocimiento de matrículas VPAR. Para evitar problemas de concurrencia la mayoría de métodos, salvo el Init() son estáticos, garantizando el acceso simultaneo a la clase, como por ejemplo añadir imágenes o leer los resultados en paralelo.

Para poder utilizar el ensamblado, los pasos a seguir son los siguientes:

1. Crear o abrir proyecto existente en Visual Studio
2. Agregar referencia



3. Declarar su uso en el código, mediante la instrucción adecuada para cada lenguaje:

```
"Imports VPMRTWrapper" (vb.net)
```

```
"using VPMRTWrapper" (C#)
```

Etc.

4. A partir de este momento podemos utilizar sus funciones, que detallamos a continuación.

Declaración / Configuración

Configuración

El ensamblado .NET está preparado para funcionar como componente estático, facilitando así la concurrencia al mismo, incluso desde diferentes hilos de programación. Por ello no es necesario declarar ninguna variable. Este es un cambio importante en la versión 2.0 del ensamblado, respecto la versión 1.0 del mismo, que era parcialmente estático.

Las funciones de lectura preparan el motor de reconocimiento con la configuración pasada por parámetro y generan las peticiones de lectura, encolándose para ser procesadas por el gestor multithread.

Así pues, el primer parámetro de todas ellas es una estructura inicializada con la configuración definida para esa petición. Esta inicialización puede realizarse a través del constructor o mediante una llamada posterior. En cualquier caso es obligatorio hacerla.

SetConfiguration ([ByVal](#) [Miliseconds](#) [As](#) [Int32](#),
[ByVal](#) [aplicarCorreccion](#) [As](#) [Int32](#),
[ByVal](#) [distance](#) [As](#) [Single](#),
[ByVal](#) [vertCoef](#) [As](#) [Single](#),
[ByVal](#) [horizCoef](#) [As](#) [Single](#),
[ByVal](#) [angle](#) [As](#) [Single](#),
[ByVal](#) [left](#) [As](#) [Int32](#),
[ByVal](#) [top](#) [As](#) [Int32](#),
[ByVal](#) [width](#) [As](#) [Int32](#),
[ByVal](#) [height](#) [As](#) [Int32](#),
[ByVal](#) [numSteps](#) [As](#) [Int32](#), [ByVal](#) [steps\(\)](#) [As](#) [Integer](#))

Parámetros

<i>Miliseconds</i>	Indica al motor de reconocimiento un tiempo de proceso máximo tras el cual retornará con el resultado que haya obtenido hasta el momento. Este valor está expresado en milisegundos. Se recomienda 0
<i>AplicarCorreccion</i>	Corrección Coefficients: En caso de poner este booleano a true, los 4 parámetros siguientes se aplicarán. Para más información sobre estos parámetros, dirigirse al manual de usuario de VPAR.
<i>distance</i>	Parámetro para la distancia.
<i>vertCoef</i>	Parámetro para el coeficiente vertical.
<i>horizCoef</i>	Parámetro para el coeficiente horizontal.

<i>angle</i>	Parámetro para el ángulo.
<i>left</i>	Si alguno de los 4 parámetros (left, top, width, height) es diferente de cero, se utilizará esta ventana en vez de toda la imagen para buscar la matrícula. Left indica la coordenada izquierda de la ventana.
<i>top</i>	Coordenada superior de la ventana.
<i>width</i>	Anchura de la ventana.
<i>height</i>	Altura de la ventana.
<i>numSteps</i>	En caso de poner este valor a 2 se usaran las alturas del vector siguiente.
<i>steps()</i>	Array con el inicio del intervalo en la primera posición y el final del intervalo en la segunda. Por ejemplo 25, 45.

Ejemplo 1

```
' VB.net code
Private Sub SetVPMRTConfiguration()
    Dim stepsArray(2) As Int32
    If Me.CheckBoxCharacterHeight.Checked Then
        'Use Automatic height
        stepsArray(0) = 25
        stepsArray(1) = 60
        lCharSize = -1
    Else
        'Use user height
        stepsArray(0) = Convert.ToInt32(Me.userHeigth.Split(",") (0).Trim)
        stepsArray(1) = Convert.ToInt32(Me.userHeigth.Split(",") (1).Trim)
    End If
    VparMTInterface.SetConfiguration(0, 0, 0, 0, 0, 0, 0,
        Convert.ToInt32(Me.NLeft.Value),
        Convert.ToInt32(Me.Ntop.Value),
        Convert.ToInt32(Me.NWidth.Value),
        Convert.ToInt32(Me.NHeight.Value), 2, stepsArray)
End Sub
```

Ejemplo 2

```
// C# code
Int32[] stepsArray = { 35, 65 };

VparMTInterface.SetConfiguration(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, stepsArray);
```

Inicialización / Finalización

Init

Inicializa el **Vehicle Plates Automatic Reader (VPAR)**. Carga las Redes Neuronales Artificiales del OCR e inicializa parámetros. Esta función debe llamarse antes de usar cualquiera de las otras funciones de esta librería excepto `SetConfiguration`, que puede llamarse en cualquier momento.

```
long Init (long lCountryCode,  
  
          long lAverageCharacterHeight,  
  
          long bDuplicateLines = false,  
  
          long lreserved1 = 0,  
  
          long lreserved2 = 0,  
  
          long bTrace = 0,  
  
          bool wTrace = false );
```

Parámetros

lCountryCode Código de país para el que queremos reconocer las matrículas. Referirse al fichero de definición para una lista de todos los países soportados.

lAvCharacterHeight Altura media aproximada de los caracteres de las matrículas a leer (en *pixels*). Si pasamos un `-1` la librería calcula de manera automática la altura. (de 20 a 50 píxeles)

Si se selecciona altura automática, el tiempo de proceso aumenta considerablemente.

bDuplicateLines Para imágenes capturadas con la mitad de las líneas pasaremos el valor 1 (`true`). Para los demás casos 0 (`false`).

lReserved1 Reordena (cuando el país es España) el resultado de las matrículas de 2 líneas. Por defecto el valor 0 no reordena, con valor 1 se reordena el resultado de la lectura.

lReserved2 Activamos filtro especial para tratamiento de color. Los posibles valores son:

- 0 Hace la media de los tres canales (**Recomendado, por defecto**)
- 1 Usa el primer canal (rojo si la imagen/buffer es RGB o azul si es BGR)
- 2 Usa el segundo canal (en principio, sería siempre el verde)
- 3 Usa el tercer canal (azul si la imagen/buffer es RGB o rojo si es BGR)
- < 0 Error
- > 3 Error

bTrace Traza interna. Este parámetro debe valer 0.

wTrace Traza interna. Este parámetro debe valer **false**.

Valor de Retorno

0 → Error.

1 → Ok.

Long Init (**Long** *lcountryCode*);

Realiza la llamada anterior, con los valores por defecto

Init (*lcountryCode*, -1, 0, 0, 0, 0)

Ejemplo

```
// C# code
VparMTInterface.SetConfiguration (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, stepsArray);
duplicateLines = 0;
return VparMTInterface.Init(countryCode, lCharSize, duplicateLines, 0, 0, 0);
```

QueryEnd

Libera la memoria reservada para el **Vehicle Plates Automatic Reader Multi Thread (VPARMT)**.

Para ello espera a que se vacíe la cola de peticiones de reconocimiento acumulada y después libera los recursos. Esta función puede tardar un tiempo significativo en finalizar si se encuentran muchas peticiones pendientes de procesar.

Ejemplo

```
// C# code
// Stop VPAR when finished processing images

if (VparMTInterface.IsRunning())
{
    while (VparMTInterface.IsProcessing)
    {
        Application.DoEvents();
    }
    VparMTInterface.QueryEnd();
}
```


Solicitud de lectura de matrículas

Add

Añade la matrícula a la cola de peticiones de lectura. Cuando le llegue su turno se intentará leer la matrícula contenida en una imagen. Esta función recibe como entrada una imagen o la ruta en la que se encuentra una imagen. **Analiza la imagen** en busca de una matrícula de vehículo y si la encuentra, **lee la matrícula** y colocará el resultado en una cola de matrículas resultados accesibles. Estas funciones retornan ID que servirá de identificador de la petición que se está procesando para la recuperación de resultados. Esta función puede llamarse de dos formas diferentes.

Long Add (*Bitmap image*)

Long Add (*Bitmap image, Object additionalInfo*)

Long Add (*Bitmap image, Bool saveBitmapInfo, Object additionalInfo*)

Long Add (*String image*)

Long Add (*String image, Object additionalInfo*)

Long Add (*Long width, Long height, IntPtr image*)

Long Add (*Long width, Long height, IntPtr image, Object additionalInfo*)

Parámetros

image

Este parámetro contiene la imagen a procesar en formato Windows Bitmap o la ruta en la que se encuentra la imagen o puntero a posición de memoria de la imagen. **Los requisitos de memoria cuando se almacena información de Bitmap son MUY ELEVADOS** por lo que se ha de usar con cuidado y bajo el propio riesgo.

saveBitmapInfo

Parámetro booleano (true / false) que indica si queremos almacenar en los resultados la imagen origen de la que se obtiene la matrícula. Si se activa a true, la imagen se podrá obtener con el método *GetSourceImage()* de la clase [PlatInfo](#).



Valor de Retorno

- -1 → Error o cola de procesamiento llena.
- Número positivo, incluido cero → Ok.
 - El número indica el identificador asociado a la imagen, para su posterior uso.

Ejemplo

```
// C# code
Bitmap bitmap = eventArgs.Frame;

if (VparMTInterface.IsRunning())
{
    VparMTInterface.Add(bitmap, true, "");
}
```

Consultas disponibles

GetQueueSize

Nos indica el número de elementos en cola que tenemos pendientes de procesar por el motor de reconocimiento de matrículas. Imágenes añadidas (Add) que aún no han sido procesadas.

Long GetQueueSize();

Valor de Retorno

- Número de elementos pendientes de procesar por el motor de reconocimiento.

Ejemplo

```
// C# code
// Wait to end processing images
while (VparMTInterface.GetQueueSize > 0)
{
    Application.DoEvents();
}
```

IsRunning

Nos indica si el motor de reconocimiento de matrículas ha sido inicializado correctamente y no se ha detenido mediante el método QueryEnd.

```
bool IsRunning();
```

Valor de Retorno

- Cierto o falso dependiendo si el motor está listo para recibir peticiones de lectura.

Ejemplo

```
// C# code
// Add images while running

while (VparMTInterface.IsRunning)
{
    VparMTInterface.Add(image);
}
```

IsProcessing

Nos indica si el motor de reconocimiento de matrículas está pendiente de procesar resultados. Sea porque tiene elementos en cola o en proceso actualmente.

```
bool IsProcessing();
```

Valor de Retorno

- Cierto o falso dependiendo si el motor está procesando peticiones de lectura.

Ejemplo

```
// C# code
// Stop when ends processing

if (VparMTInterface.IsRunning())
{
    while (VparMTInterface.IsProcessing)
    {
        Application.DoEvents();
    }
    VparMTInterface.QueryEnd();
}
```

GetLicensedCores

Nos indica el número núcleos de procesador disponibles para los que tenemos licencia válida. Será el número máximo de procesos simultáneos que podemos gestionar.

Long GetLicensedCores();

Valor de Retorno

- Número de núcleos de procesador que tenemos licenciados para usar.

Ejemplo

```
// C# code
// Busy cores
labelCU.Text = String.Format("Cores used: {0}",
VparMTInterface.GetLicensedCores - VparMTInterface.GetFreeCores);
```

GetFreeCores

Nos indica el número núcleos de procesador disponibles. Será el número máximo de procesos simultáneos que podemos gestionar.

```
Long GetFreeCores();
```

Valor de Retorno

- Número de núcleos de procesador que tenemos libres en el momento de la consulta.

Ejemplo

```
// C# code  
// Busy cores  
labelCU.Text = String.Format("Cores used: {0}",  
VparMTInterface.GetLicensedCores - VparMTInterface.GetFreeCores);
```


Obtención de resultados

GetNumberOfPlates

Nos indica el número de matrículas que tenemos en la cola de resultados. Si el valor es mayor a cero, nos indica que tenemos alguna matrícula, cuyo resultado aún no hemos leído.

Long GetNumberOfPlates();

Valor de Retorno

Número de matriculas en la cola de resultados.

Ejemplo 1

```
' vb.NET code
Dim ptinfo As VPARMTWrapper.PlateInfo
While VparMTInterface.GetNumberOfPlates > 0

    ptinfo = VparMTInterface.GetFirstPlate()

    Dim plateNumber As [String] = ptinfo.GetPlateNumber()

End While
```

Ejemplo 2

```
// C# code

VPARMTWrapper.PlateInfo ptinfo = default(VPARMTWrapper.PlateInfo);

while (VparMTInterface.GetNumberOfPlates() > 0)
{
    ptinfo = VparMTInterface.GetFirstPlate();
}
```

GetFirstPlate

Retorna la información de la primera matrícula que se encuentra en la cola de resultados y elimina esta información de la cola de resultados pendientes, por lo que si no la almacenamos, esta información se perderá.

PlateInfo `GetFirstPlate();`

Valor de Retorno

Información de la primera matrícula en la cola de resultados. En caso de no haber ningún resultado pendiente, se retorna null (o Nothing).

Estructura PlateInfo

Los métodos disponibles en la clase PlateInfo son los siguientes

<i>GetLeftPlatePosition</i> ()	Esta propiedad retorna la coordenada izquierda de la matrícula encontrada.
<i>GetRightPlatePosition</i> ()	Esta propiedad retorna la coordenada derecha de la matrícula encontrada.
<i>GetTopPlatePosition</i> ()	Esta propiedad retorna la coordenada superior de la matrícula encontrada.
<i>GetBottomPlatePosition</i> ()	Esta propiedad retorna la coordenada inferior de la matrícula encontrada.
<i>GetPlateNumber</i> ()	Esta propiedad retorna el texto con la matrícula encontrada.
<i>GetSourcePath</i> ()	Esta propiedad retorna la ruta (si disponible) de la imagen usada para buscar la matrícula encontrada.
<i>GetId</i> ()	Esta propiedad retorna el identificador asignado a esta solicitud de lectura de matrícula, es el valor retornado por el método Add.
<i>GetGlobalConfidence</i> ()	Esta propiedad retorna un valor que expresa la fiabilidad de la lectura de la matrícula analizada por el 'Vehicle Plates Automatic Reader'. Este valor está expresado en tanto por ciento (0% – 100%).
<i>GetAverageCharacterHeigth</i> ()	Esta propiedad retorna el valor medio de la altura de los caracteres de la matrícula encontrada.
<i>GetNumCharacters</i> ()	Esta propiedad retorna el número de caracteres de la matrícula encontrada.

`GetCharConfidence()` Esta propiedad retorna un array conteniendo las fiabilidades de cada uno de los caracteres de la matrícula encontrada. El array va de cero hasta `GetNumCharacters() - 1`.

`GetPlateFormat()` Esta propiedad retorna el código del país que se ajusta al formato de matrícula leído. Por ejemplo si la matrícula es española, retornará 101. En caso de no identificar el país se retorna un cero. En caso de identificar continente se retorna por ejemplo 100 para Europa.

Si tenemos una lista de elementos `PlateInfo`, podemos ordenarla en función del valor de su coordenada izquierda utilizando el método sobrecargado `Sort`.

Ejemplo

```
// C# code
VPARMTWrapper.PlateInfo ptinfo;
while (VparMTInterface.GetNumberOfPlates() > 0)
{
    ptinfo =VparMTInterface.GetFirstPlate();
    if (Convert.ToDecimal(ptinfo.GetGlobalConfidence) > 60)
    {
        String plateNumber = ptinfo.GetPlateNumber;
        String lastNumber = "";
        if (this.listBoxResults.SelectedItem != null)
            lastNumber = this.listBoxResults.SelectedItem.ToString();
        if (!String.Equals(plateNumber, lastNumber))
        {
            this.listBoxResults.Items.Add(ptinfo.GetPlateNumber);
            this.listBoxResults.SelectedIndex = this.listBoxResults.Items.Count - 1;
        }
    }
}
ptinfo = null;
```

Ejemplo 2

```
' vb.NET code

Dim ptinfo As VPARMTWrapper.PlateInfo
WhileVparMTInterface.GetNumberOfPlates > 0
    ptinfo =VparMTInterface.GetFirstPlate()
    Me.avgPtime = Me.avgPtime + ptinfo.GetProcessingTime
    If (ptinfo.GetGlobalConfidence > Me.NumericUpDownMinConf.Value) Then
        If Me.CheckBoxBeep.Checked Then
            System.Media.SystemSounds.Exclamation.Play()
            Me.fiabilidades(Me.activePanel).Text = String.Format("{0:00}%",
                ptinfo.GetGlobalConfidence)
            Me.fiabilidades(Me.activePanel).Visible = True
            Me.activePanel = (Me.activePanel + 1) Mod 4
            Me.LabelACH.Text = "Avg. Char Heigth: " +
                Convert.ToInt32(ptinfo.GetAverageCharacterHeigth).ToString
        End If
    End While
```

GetPlate

Retorna la información de la matrícula que se encuentra en la cola de resultados con el identificador que se le pasa por parámetro y elimina esta información de la cola de resultados pendientes, por lo que si no la almacenamos, esta información se perderá.

```
PlateInfo GetPlate(Long id);
```

Parámetros

id Este parámetro contiene el id de la imagen de la que queremos obtener el resultado. Puede haber más de un resultado para el mismo id (Varias matrículas en una misma imagen) se retorna el primer resultado encontrado.

Valor de Retorno

Información de la matrícula en la cola de resultados con el id (Retornado por la función 'Add') que le pasamos por parámetro. En caso de no haber ningún resultado con este id, se retorna null (o Nothing).

Ejemplo

```
' vb.NET code
If Not (vpar Is Nothing) Then
id = VparMTInterface.Add(Me.BytesToImage(image))
    If (id >= 0) Then 'Success enqueued
        Do
            System.Threading.Thread.Sleep(100)
            plate = VparMTInterface.GetPlate(id)
        Loop While plate Is Nothing 'Loop until get result (First result)
        listOfPlates.Add(plate.GetPlateNumber)

        Do
            plate =VparMTInterface.GetPlate(id)
            If Not (plate Is Nothing) Then
                listOfPlates.Add(plate.GetPlateNumber)
            Loop While Not (plate Is Nothing) 'Enqueue if more results available
        Else
            'listOfPlates.Add(String.Format("Process failed."))
        End If
    Else
        'Error Vpar
    End If
```

Resultados: Métodos avanzados

Los siguientes métodos y funciones están indicados cuando se requiere mayor funcionalidad y acceso a la información una vez realizado el proceso de reconocimiento y también cuando se esperan leer varias matrículas en una sola imagen. Estos métodos permiten recuperar la información completa de los parámetros usados y de cualquier otra información añadida en el momento de realizar la llamada a la función Add. Permite conocer la información de fecha y hora exactas en que se añadieron las imágenes a la cola de procesamiento y otra información, normalmente no necesaria pero importante cuando se realizan operaciones de detección por radar y objetivos similares avanzados.

Si se usan varias cámaras y se quiere recuperar la información de los parámetros usados o la información adicional, estos métodos proporcionan esta funcionalidad adicional.

GetFirstElement

Retorna información completa de la primera imagen con todos los resultados disponible. Esto incluye los parámetros usados para el reconocimiento de la imagen, cualquier información adicional pasada mediante la función 'Add' e información de TODAS las matrículas leídas en la imagen. Este método elimina esta información de la cola de resultados pendientes, por lo que si no se guarda se perderá.

NLInfo `GetFirstElement()`;

Valor de retorno

Información de la primera imagen disponible en la cola de resultados. Si no hay ningún resultado pendiente, se retornará 'null' (o 'Nothing').

Estructura NLInfo

Los métodos disponibles en la clase NLInfo son los siguientes:

<code>GetId ()</code>	Esta propiedad retorna el id de la imagen procesada.
<code>Configuration ()</code>	Esta propiedad retorna la estructura de información MTConfiguration usada cuando se realizó la llamada al motor de reconocimiento. Ver la documentación 'Vparmt – Vpar multithread' para disponer de más información sobre esta estructura de datos.
<code>GetFirstPlate ()</code>	Esta propiedad retorna el primer resultado PlateInfo de la imagen.
<code>PlateCount ()</code>	Esta propiedad retorna el número de matrículas disponible en la imagen procesada.

ReadPlates () Esta propiedad retorna el número de matrículas leídas de la imagen procesada.

ImageInformation () Esta propiedad retorna la estructura de información [PicInformation](#) de la imagen procesada.

Estructura PicInformation

Los métodos disponibles en la clase PicInformation son los siguientes:

GetId () Esta propiedad retorna el id de la imagen procesada

GetPath () Esta propiedad retorna la ruta o 'Nothing' (Si no hay ruta disponible) de la imagen procesada.

GetImage () Esta propiedad retorna el Bitmap o 'Nothing' (si el bitmap no se encuentra disponible) de la imagen procesada. **Los requisitos de memoria cuando se almacena información de Bitmap son MUY ELEVADOS** por lo que se ha de usar con cuidado y bajo el propio riesgo.

GetObject () Esta propiedad retorna el Objeto pasado como parámetro cuando se ha usado la función 'Add'.

GetTimeStamp () Esta propiedad retorna la fecha y hora del momento en el que la imagen se añadió a la cola de procesamiento.

GetUTCTimeStamp () Como en la propiedad anterior, pero se usan Fecha y hora universales.

Ejemplo

```
Dim ptElement As NLInfo = Nothing
'Get first element where all results are ready
ptElement = MTInterface.GetFirstElement

Dim ptinfo As PlateInfo = Nothing
ptinfo = ptElement.GetFirstPlate()

While Not ptinfo Is Nothing
    DisplayInfo(Me.ListBoxResults, ptElement, ptinfo)
    'Retrieve GPS information stored when Adding image
    Dim gpsCoordinates As GPS = ptElement.ImageInformation.GetObject
    ptinfo = ptElement.GetFirstPlate()
    Application.DoEvents()
End While
```

Anexo 1 – Códigos de país

En CD del software contiene siempre la versión más actualizada de la lista

LOC_CODE_ESP ; 101 ; Spain

LOC_CODE_POR ; 102 ; Portugal

LOC_CODE_FRA ; 103 ; France

LOC_CODE_ITA ; 104 ; Italy

LOC_CODE_UK ; 105 ; United Kingdom

LOC_CODE_GRE ; 106 ; Greece

LOC_CODE_IRL ; 107 ; Ireland (ROI)

LOC_CODE_GER ; 108 ; Germany

LOC_CODE_PYR ; 109 ; Pyrinees (Spain, Andorra & France)

LOC_CODE_POL ; 110 ; Poland

LOC_CODE_BUL ; 111 ; Bulgaria

LOC_CODE_CHI ; 201 ; Chile

LOC_CODE_COL ; 202 ; Colombia

LOC_CODE_BRA ; 203 ; Brazil

LOC_CODE_ARG ; 204 ; Argentina

LOC_CODE_MEX ; 205 ; Mexico

LOC_CODE_ECU ; 206 ; Ecuador

LOC_CODE_VEN ; 207 ; Venezuela

LOC_CODE_RUS ; 301 ; Russia

LOC_CODE_TUR ; 302 ; Turkey

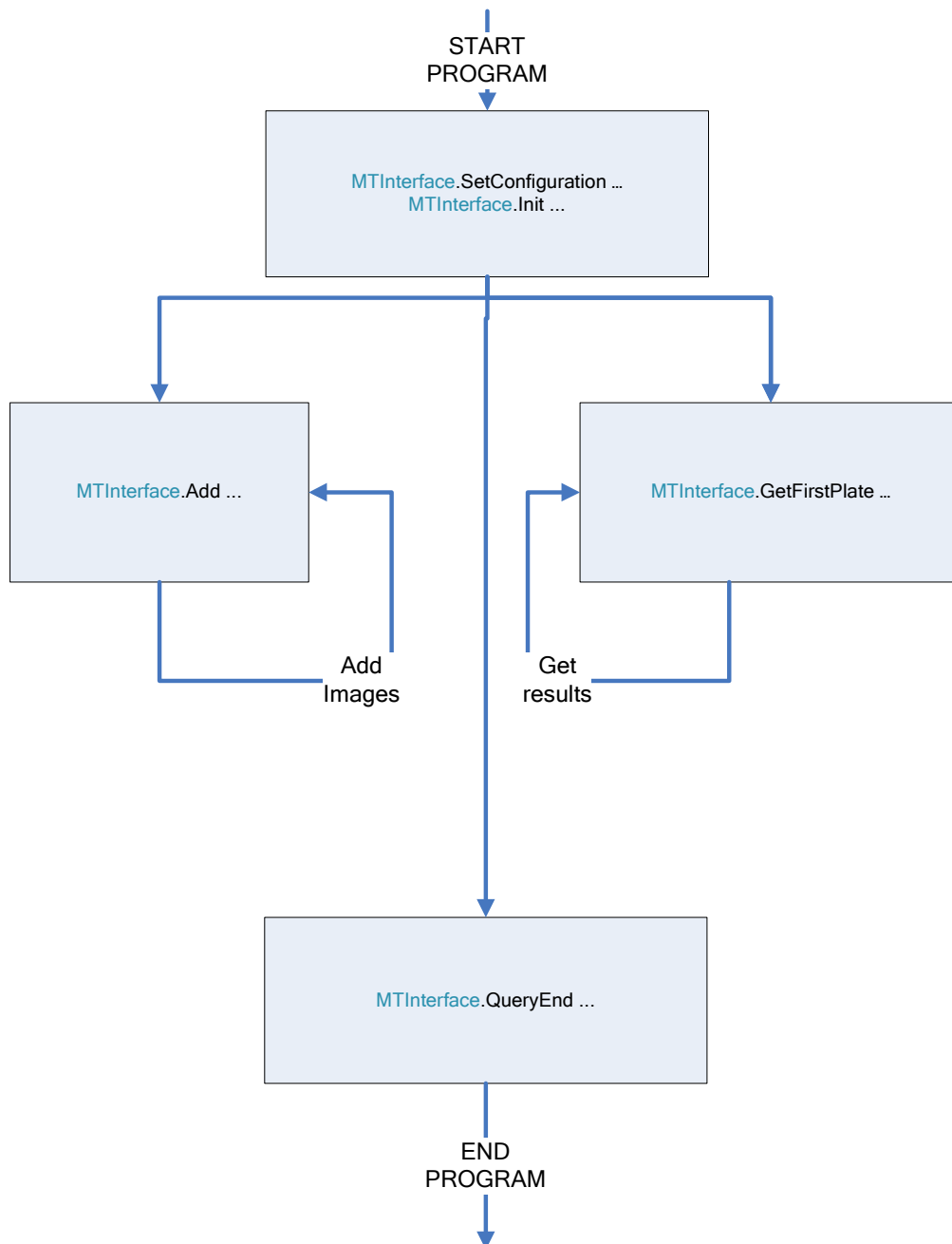
LOC_CODE_VIE ; 303 ; Vietnam

LOC_CODE_SAF ; 401 ; South Africa

Appendix 2 – Esquema Multi-procesador

Versión simple

VPAR Multi thread workflow (.NET Assembly). Simple version



VPAR Multi thread workflow (.NET Assembly). Complete version
