

MANUAL DE USUARIO DE Dibalmage.dll PARA INTEGRADORES



CONTENIDO

1- DESCRIPCIÓN DE LA LIBRERIA	3
2- DIBALIMAGE FUNCTIONS.....	4
2.1 FUNCIONES DE IMAGEN A FICHERO	4
2.1.1 <i>Función ImageFileGenerator</i>	4
2.1.2 <i>Función ImageFileGeneratorBytes</i>	4
2.2 FUNCIONES DE ENVÍO DE IMÁGENES	5
2.2.1 <i>SendImagetoSingleScale</i>	6
2.2.2 <i>SendImagetoMultiScales</i>	6
2.2.3 <i>SendMultiImagestoSingleScale</i>	7
2.2.4 <i>SendMultiImagestoMultiScales</i>	8
2.2.5 <i>Dibalscale structure</i>	9
3- VENTANA DE INFORMACIÓN DE LA COMUNICACIÓN.....	11

1- DESCRIPCIÓN DE LA LIBRERÍA

Es una librería desarrollada en C# la cual permite al usuario transformar imágenes y enviárselas a las balanzas.

Este manual está basado en la versión ASCII/ANSI de la DibalImage.dll. Si tu software debe soportar caracteres chinos debes usar la versión Unicode. Se recomienda el uso de la versión ASCII/ANSI a no ser que el software que utiliza la librería use o soporte caracteres chinos.

Esta librería permite dos operaciones:

- Crear un fichero que contenga los registros convertidos de una imagen para que puedan ser enviados a balanza.
- Existen varias funciones para convertir y enviar directamente a las balanzas deseadas. Se puede enviar una sola imagen a una o varias balanzas y también se puede establecer un directorio para que el programa recoja las imágenes y se las envíe a una o varias balanzas.

Las dependencias de la librería son las siguientes:

- Microsoft Framework (Versión 2.0 o superior)
- dibalscop.dll (Versión v1.0.1.10 o superior). Incluida en el paquete ImageSDK.
- commL.dll (Versión v.1.0.1.4 o superior). Incluida en el paquete ImageSDK.
- iconv.dll (Versión 1.9.0.0 o superior). Solo es necesaria en la versión UNICODE. Incluida en el paquete ImageSDK. (UNICODE).

Los formatos de imágenes soportados son los siguientes:

- BMP
- EMF
- EXIF
- GIF
- ICO
- JPG/JPEG
- PNG
- TIFF
- WMF

2- DiballImage Functions

2.1 FUNCIONES DE IMAGEN A FICHERO

2.1.1 Función ImageFileGenerator

ImageFileGenerator es un método que pertenece a clase ImagePalletized..

Está función crea un fichero nuevo o añade al final escribiendo los registros resultantes de la conversión de una imagen.

```
public static bool ImageFileGenerator(int masterAddress, int type, int index, string pathImage, string pathTxResul, int fileCat, int assignPlu, int displaySize)
```

Parámetros:

- 1) **masterAddress**, Dirección de maestra de la balanza. Debe ser un valor comprendido entre 0-99.
Tipo dato: `int` -> (4 bytes).
- 2) **type**, tipo de imagen. 1-> Publicidad 2-> Artículo. Variable Int (4 bytes).
- 3) **index**, Es el índice o número de orden de la imagen. Variable Int (4 bytes).
Valor Min: 1 Valor Max: 999999 (Articulo), 200(Publicidad Serie 500), 100(Publicidad D900)
- 4) **pathImage**, Ruta completa de la imagen a tratar. Variable String.
- 5) **pathTxResul**, Ruta completa del fichero donde se escribirán los registros.
Variable String
- 6) **fileCat**, Determina si se debe sobrescribir o añadir al final del fichero. 0-> Sobrescribir, 1-> Añadir.. Variable int(4 bytes)
- 7) **assignPlu**, Determina si la imagen se asocia al artículo cuyo código es igual al valor del parámetro orden. 0 -> No 1-> Si. Variable int(4 bytes)
- 8) **displaySize**, determina el tipo de display. 0-> 7", 1-> 12", 2-> 15".
Variable int(4 bytes)

Valor retornado: Devuelve un valor booleano que indica el resultado de la operación. True-> Satisfactorio, False-> Fallido

2.1.2 Función ImageFileGeneratorBytes

ImageFileGeneratorBytes es un método que pertenece a clase ImagePalletized.

Está función crea un fichero nuevo o añade al final escribiendo los registros resultantes de la conversión de una imagen.

La diferencia con la anterior función es la forma en la que trata los datos que son escritos en el fichero. El primero lo hace en modo texto y el segundo en modo binario. El resultado debe ser el mismo pero es posible que el primero pueda fallar en algunos codepages (ej. Chino).

```
public static bool ImageFileGeneratorBytes(int masterAddress, int
type, int index, string pathImage, string pathTxResul,int fileCat, int
assignPlu, int displaySize)
```

Parámetros:

- 9) **masterAddress**, Dirección de maestra de la balanza. Debe ser un valor comprendido entre 0-99.
Tipo dato: `int` -> (4 bytes).
- 10) **type**, tipo de imagen. 1-> Publicidad 2-> Artículo. Variable Int (4 bytes).
- 11) **index**, Es el índice o número de orden de la imagen. Variable Int (4 bytes).
Valor Min: 1 Valor Max: 999999 (Articulo), 200(Publicidad Serie 500), 100(Publicidad D900)
- 12) **pathImage**, Ruta completa de la imagen a tratar. Variable String.
- 13) **pathTxResul**, Ruta completa del fichero donde se escribirán los registros.
Variable String
- 14) **fileCat**, Determina si se debe sobrescribir o añadir al final del fichero. 0-> Sobrescribir, 1-> Añadir.. Variable int(4 bytes)
- 15) **assignPlu**, Determina si la imagen se asocia al artículo cuyo código es igual al valor del parámetro orden. 0 -> No 1-> Si. Variable int(4 bytes)
- 16) **displaySize**, determina el tipo de display. 0-> 7", 1-> 12", 2-> 15".
Variable int(4 bytes)

Valor retornado: Devuelve un valor booleano que indica el resultado de la operación. True-> Satisfactorio, False-> Fallido

2.2 FUNCIONES DE ENVÍO DE IMÁGENES

Una vez que el proceso iniciado por alguna de estas funciones finaliza la imagen es movida a un subdirectorio del directorio donde se encuentra la imagen a tratar. El nombre del directorio varía en función del resultado:

- Si el resultado es correcto, la imagen es movida a un subdirectorio llamado "Image_Processed".
- Si la imagen no puede ser enviada a una cierta balanza la imagen es movida a un subdirectorio llamado "Image_ErrorCom_A.B.C.D" donde "A.B.C.D" es la dirección IP de la balanza.
- Si la imagen no es válida es movida a un subdirectorio llamado "Image_FormatError".

2.2.1 SendImagetoSingleScale

SendImagetoSingleScale es un método que pertenece a clase ImagePalletized..

Esta función transforma una imagen, cuyo nombre y localización es pasada por parámetro, en registros de imágenes y los envía a la balanza deseada.

```
public static string SendImagetoSingleScale(DibalScale scaleInfo, int
imageType, int indexImage, string pathImage, int assignPlu, int
displaySize, int showWindow, int closeTime)
```

Parameters:

- 1) **scaleInfo**, Es una estructura de tipo DibalScale. Esta estructura contiene la información de la balanza a la que se desea enviar la imagen.
- 2) **imageType**, tipo de imagen. 1-> Publicidad 2->Imagen de artículo.
Variable Int (4 bytes).
- 3) **indexImage**, Es el índice o número de orden de la imagen.
Variable Int (4 bytes).
Valor Min: 1 Valor Max: 999999 (Articulo), 200(Publicidad Serie 500), 100(Publicidad D900)
- 4) **pathImage**, Ruta completa de la imagen a tratar. Variable String.
- 5) **assingPlu**, Determina si la imagen se asocia al artículo cuyo código es igual al valor del parámetro orden. 0 -> No 1-> Si. Variable int(4 bytes)
- 6) **displaySize**, determina el tipo de display. 0-> 7", 1-> 12", 2-> 15".
Variable int(4 bytes)
- 7) **showWindow**, Determina si se debe mostrar la ventana información de la comunicación o no. 0-> NO 1-> Si Variable int(4 bytes)
- 8) **closeTime**, Determina el tiempo en la que la ventana se queda en espera antes de cerrar una vez que finaliza el proceso.

Valor retornado: El valor retornado es un string. Puede tomar los siguientes valores:

- "OK" -> El proceso ha finalizado correctamente
- "A.B.C.D" -> La cadena contiene la IP de las balanzas que han fallado. Si hubiese más de una las IPs son separadas por punto y coma(,)
- commL.dll" -> commL.dll no existe o se ha producido un error al cargar.
- "No iconv.dll"-> iconv.dll no existe o se ha producido un error al cargar. Solo aplica para la versión Unicode.

2.2.2 SendImagetoMultiScales

SendImagetoMultiScales es un método que pertenece a clase ImagePalletized..

Esta función transforma una imagen, cuyo nombre y localización es pasada por parámetro, en registros de imágenes y los envía a las balanzas deseadas.

```
public static string SendImagetoSingleScale(DibalScale[]
scaleInfo, int scalesNumber, int imageType, int indexImage, string
```

pathImage, int assignPlu, int displaySize, int showWindow, int closeTime)

Parámetros:

- 1) **scaleInfo**, Es un array de estructuras de tipo DibalScale. Cada estructura contiene la información de las balanzas a la que se debe enviar la imagen.
- 2) **scalesNumber**, número de balanzas.
- 3) **imageType**, tipo de imagen. 1-> Publicidad 2->Imagen de artículo.
Variable Int (4 bytes).
- 4) **indexImage**, Es el índice o número de orden de la imagen.
Variable Int (4 bytes).
Valor Min: 1 Valor Max: 999999 (Articulo), 200(Publicidad Serie 500), 100(Publicidad D900)
- 5) **pathImage**, Ruta completa de la imagen a tratar. Variable String.
- 6) **assingPlu**, Determina si la imagen se asocia al artículo cuyo código es igual al valor del parámetro orden. 0 -> No 1-> Si. Variable int(4 bytes)
- 7) **displaySize**, determina el tipo de display. 0-> 7", 1-> 12", 2-> 15".
Variable int(4 bytes)
- 8) **showWindow**, Determina si se debe mostrar la ventana información de la comunicación o no. 0-> NO 1-> Si Variable int(4 bytes)
- 9) **closeTime**, Determina el tiempo en la que la ventana se queda en espera antes de cerrar una vez que finaliza el proceso.

Valor retornado: El valor retornado es un string. Puede tomar los siguientes valores:

- "OK" -> El proceso ha finalizado correctamente
- "A.B.C.D" -> La cadena contiene la IP de las balanzas que han fallado. Si hubiese más de una las IPs son separadas por punto y coma(,)
- "commL.dll" -> commL.dll no existe o se ha producido un error al cargar.
- "No iconv.dll"-> iconv.dll no existe o se ha producido un error al cargar. Solo aplica para la versión Unicode.

2.2.3 SendMultimagestoSingleScale

SendMultimagestoSingleScale es un método que pertenece a clase ImagePalletized..

Esta función escanea un directorio el cual es indicado por parámetro. Transforma las imágenes localizadas en el directorio y envía la información a la balanza deseada.

La imagen debe tener un formato válido para ser procesada por la librería. El formato es siguiente:

"d.xxx" donde d es un valor decimal que indica el orden o índice de la imagen y xxx es la extensión del fichero.

Si un archivo contenido en el directorio indicado no es una imagen o es una imagen de formato no soportado el archivo es movido a subdirectorio Image_FormatError.

Existe la posibilidad de aplicar un filtro a la hora de indicar el directorio a escanear, Por ejemplo:

“C:\ImageFolder*.jpg” or “C:\ImageFolder*.*”

```
public static string SendMultiImagestoSingleScale(DibalScale
scaleInfo, int imageType, string imageFolder, int assignPlu, int
displaySize, int showWindow, int closeTime)
```

Parámetros:

- 1) **scaleInfo**, Es una estructura de tipo DibalScale. Esta estructura contiene la información de la balanza a la que se desea enviar las imágenes.
- 2) **imageType**, tipo de imagen. 1-> Publicidad 2->Imagen de artículo.
Variable Int (4 bytes).
- 3) **folderImage**, Directorio a sondear o escanear. Variable String.
- 4) **assingPlu**, Determina si la imagen se asocia al artículo cuyo código es igual al valor del parámetro orden. 0 -> No 1-> Si. Variable int(4 bytes)
- 5) **displaySize**, determina el tipo de display. 0-> 7”, 1-> 12”, 2-> 15”.
Variable int(4 bytes)
- 6) **showWindow**, Determina si se debe mostrar la ventana información de la comunicación o no. 0-> NO 1-> Si Variable int(4 bytes)
- 7) **closeTime**, Determina el tiempo en la que la ventana se queda en espera antes de cerrar una vez que finaliza el proceso.

Valor retornado: El valor retornado es un string. Puede tomar los siguientes valores:

- “OK” -> El proceso ha finalizado correctamente
- “A.B.C.D” -> La cadena contiene la IP de las balanzas que han fallado. Si hubiese más de una las IPs son separadas por punto y coma(;))
- commL.dll” -> commL.dll no existe o se ha producido un error al cargar.
- “No iconv.dll”-> iconv.dll no existe o se ha producido un error al cargar. Solo aplica para la versión Unicode.

2.2.4 SendMultImagestoMultiScales

SendMultImagestoMutiScales es un método que pertenece a clase ImagePalletized..

Esta función escanea un directorio el cual es indicado por parámetro. Transforma las imágenes localizadas en el directorio y envía la información a las balanzas deseadas.

La imagen debe tener un formato válido para ser procesada por la librería. El formato es siguiente:

“d.xxx” donde d es un valor decimal que indica el orden o índice de la imagen y xxx es la extensión del fichero.

Si un archivo contenido en el directorio indicado no es una imagen o es una imagen de formato no soportado el archivo es movido a subdirectorio Image_FormatError.

Existe la posibilidad de aplicar un filtro a la hora de indicar el directorio a escanear, Por ejemplo:

“C:\ImageFolder*.jpg” or “C:\ImageFolder*.*”

```
public static string SendMultiImagestoMultiScales(DibalScale[]
scaleInfo, int scalesNumber, int imageType, string imageFolder,
int assignPlu, int displaySize, int showWindow, int closeTime)
```

Parámetros:

- 1) **scaleInfo**, Es un array de estructuras de tipo DibalScale. Cada estructura contiene la información de las balanzas a la que se debe enviar las imágenes.
- 2) **imageType**, tipo de imagen. 1-> Publicidad 2->Imagen de artículo.
Variable Int (4 bytes).
- 3) **folderImage**, Directorio a sondear o escanear. Variable String.
- 4) **assingPlu**, Determina si la imagen se asocia al artículo cuyo código es igual al valor del parámetro orden. 0 -> No 1-> Si. Variable int(4 bytes)
- 5) **displaySize**, determina el tipo de display. 0-> 7”, 1-> 12”, 2-> 15”.
Variable int(4 bytes)
- 6) **showWindow**, Determina si se debe mostrar la ventana información de la comunicación o no. 0-> NO 1-> Si Variable int(4 bytes)
- 7) **closeTime**, Determina el tiempo en la que la ventana se queda en espera antes de cerrar una vez que finaliza el proceso.

Valor retornado: El valor retornado es un string. Puede tomar los siguientes valores:

- “OK” -> El proceso ha finalizado correctamente
- “A.B.C.D” -> La cadena contiene la IP de las balanzas que han fallado. Si hubiese más de una las IPs son separadas por punto y coma(;))
- “commL.dll” -> commL.dll no existe o se ha producido un error al cargar.
- “No iconv.dll”-> iconv.dll no existe o se ha producido un error al cargar. Solo aplica para la versión Unicode.

2.2.5 Dibalscale structure

Dibalscale es una estructura pública de DibalImage.dll. Es usada para contener la información de la balanza. En algunas funciones es pasada como un array de estructuras y entra como un solo dato.

La estructura de Dibalscale es la siguiente:

```

public struct DibalScale
{
    public int masterAddress;
    public string IPAddress;
    public int txPort;
    public int rxPort;
    public string model;
    public string display;
    public string section;
    public int group;
    public string logsPath;
}

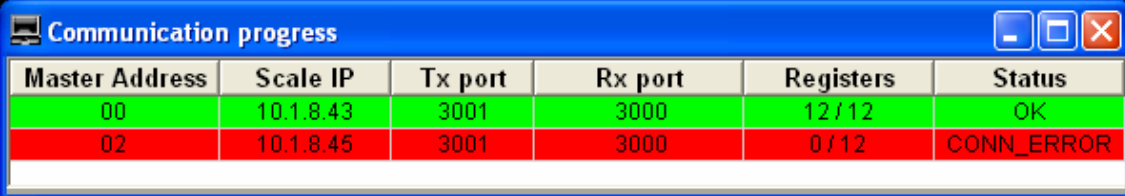
```

Miembros:

- 1) **masterAddress**, Dirección de maestra de la balanza. Valor comprendido entre 0 y 99. Variable int (4 bytes).
- 2) **IPAddress**, Dirección IP de la balanza. Valor de tipo String.
- 3) **txPort**, Puerto de transmisión de la balanza (TX). Variable int (4 bytes)
- 4) **rxPort**, Puerto de recepción (RX) de la balanza. Variable int (4 bytes)
- 5) **model**, Es una variable de tipo string que define el modelo de la balanza. Dos posibles valores:
"500RANGE"
"LSERIES"
- 6) **display**, Es una variable de tipo string que define el tipo de display. No está en uso.
- 7) **section**, Es una variable de tipo string que determina la sección a la que pertenece la balanza.
- 8) **group**, El grupo de la balanza. Valor comprendido entre 0 y 99. Variable int (4 bytes).
- 9) **logsPath**, Ruta para crear el fichero de logs de comunicación. Si está vacío no se crea. Tipo string..

3- Ventana de información de la comunicación

Dibalscop.dll (Una librería usada por DIBallImage.dll) lanza una ventana mostrando el estado de las comunicaciones (El número de registros enviados y el resultado).



Master Address	Scale IP	Tx port	Rx port	Registers	Status
00	10.1.8.43	3001	3000	12 / 12	OK
02	10.1.8.45	3001	3000	0 / 12	CONN_ERROR

La columna status puede mostrar los siguientes valores o resultados:

- **No CommL:** La comunicación no se ha podido iniciar porque commL no se encuentra o se ha producido un error al cargar dicha librería.
- **No iconv:** Solo aplica en la versión UNICODE. Se muestra cuando se produce un error al cargar la librería iconv.dll.
- **OK:** Comunicación finalizada de forma satisfactoria.
- **CONN_ERROR:** No se ha podido establecer comunicación con la balanza.
- **SEND_ERROR:** Se ha producido un error durante el envío. La comunicación se había establecido correctamente pero se ha interrumpido. En la columna register puede verse el número de registros enviados.