

USER'S MANUAL

Dibalmage.dll

FOR INTEGRATIONS



DIBAL

CONTENTS

1- LIBRARY DESCRIPTION.....	3
2- DIBALIMAGE FUNCTIONS.....	4
2.1 IMAGE TO FILE FUNCTIONS.....	4
2.1.1 <i>ImageFileGenerator Function</i>	4
2.1.2 <i>ImageFileGeneratorBytes Function</i>	4
2.2 SEND IMAGE FUNCTIONS.....	5
2.2.1 <i>SendImagetoSingleScale</i>	5
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- COMMUNICATION STATUS WINDOW	11

1- LIBRARY DESCRIPTION

This file is a library developed in C#, which carries out basic operations in order to transform an image into Dibal records and send to the scales.

This manual is based on ASCII/ANSI version of DibalImage.dll. If your software is going to use Chinese characters you must use Unicode version instead of ASCII/ANSI version. We recommend use ASCII/ANSI version before Unicode version unless your software must support Chinese characters.

This library gives two options:

- Create a file which contains the converted registers of an image. This file can be written in append mode. There is one function which works in text mode and the other works in binary mode.
- There are several functions which convert and send the desired images to the scales. You can send one image to one or several scales and also there is the possibility of set a folder where the images are picked and sent to the scale.

The dependencies of the library are the following:

- Microsoft framework (Version 2.0 or higher)
- dibalscop.dll (Version v1.0.1.10 or higher). It is provided in the ImageSDK packet.
- commL.dll (Version v.1.0.1.4 or higher). It is provided in the ImageSDK packet.
- iconv.dll (Version 1.9.0.0) or higher. Only is necessary in the UNICODE version. It is provided in the ImageSDK packet (UNICODE).

The image formats which are supported are the following:

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

2- DiballImage Functions

2.1 IMAGE TO FILE FUNCTIONS

2.1.1 ImageFileGenerator Function

ImageFileGenerator is a method which belongs to ImagePalletized class.

The function creates/append a file with Image registers as the result of converting a Image.

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

Parameters:

- 1) **masterAddress**, Logic address of the scale (Master Address). The registers will be modified to assign this logic address.
Data type: `int` -> Integer without sign (4 bytes).
- 2) **type**, type of image. 1-> Publicity 2-> Article. Int variable of 4 bytes.
- 3) **index**, it is code or ID number of the image. Int variable of 4 bytes.
- 4) **pathImage**, Location and name of the image to process. String variable.
- 5) **pathTxResul**, Name and location of the result file.
- 6) **fileCat**, Overwrite or Append the file. 0-> Overwrite, 1-> Append.
- 7) **assignPlu**, It is used to determine if the image must be associated or not to the article whose code matches with the number value of index field.
- 8) **displaySize**, It is used to determine the size of the display of the scale which will receive the image.

Returned vale: The function returns a bool variable indicating the result of the process. True-> Success, False-> Failure

2.1.2 ImageFileGeneratorBytes Function

ImageFileGeneratorBytes is a method which belongs to ImagePalletized class.

The function creates/append a file with Image registers as the result of converting a Image.

The difference between this function and the previous one is the dealing with the data. The first one does in text mode and the second one in binary mode. The result must be the same but the first one can fail in some codepages (i.e. Chinese)

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

Parameters:

- 1) **masterAddress**, Logic address of the scale (Master Address). The registers will be modified to assign this logic address.
Data type: `int` -> Integer without sign (4 bytes).
- 2) **type**, type of image. 1-> Publicity 2-> Article. Int variable of 4 bytes.
- 3) **index**, it is code or ID number of the image. Int variable of 4 bytes.
- 4) **pathImage**, Location and name of the image to process. String variable.
- 5) **pathTxResul**, Name and location of the result file.
- 6) **fileCat**, Overwrite or Append the file. 0-> Overwrite, 1-> Append.
- 7) **assignPlu**, It is used to determine if the image must be associated or not to the article whose code matches with the number value of index field.
- 8) **displaySize**, It is used to determine the size of the display of the scale which will receive the image.

Returned vale: The function returns a bool variable indicating the result of the process. True-> Success, False-> Failure

2.2 SEND IMAGE FUNCTIONS

When the process started by those functions ends the image is moved to a subfolder of the folder where the image is located. The name of the subfolder varies depending on the result.

- If the result is OK the image is moved to the a subfolder called "Image_Processed".
- If the image can not be sent to certain scale the image is moved to a subfolder called "Image_ErrorCom_"A.B.C.D" where "A.B.C.D" is the IP of the scale.
- If the image is not valid is moved to subfolder called "Image_FormatError".

2.2.1 SendImagetoSingleScale

SendImagetoSingleScale is a method of ImagePalletized.

This function transforms an image obtained by parameter into image registers and sends the information to the desired scale.

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

Parameters:

- 1) **scaleInfo**, It is a structure which contains the information the scale. The type of the structure is DibalScale.
- 2) **imageType**, The type of the image. 1-> Publicity 2-> Item Image.
- 3) **indexImage**, it is code or ID number of the image. Int variable of 4 bytes.
- 4) **pathImage**, Location and name of the image to process. String variable.
- 5) **assingPlu**, It is used to determine if the image must be associated or not to the article whose code matches with the number value of index field.
- 6) **displaySize**, It is used to determine the size of the display of the scale which will receive the image. 0-> 7 Inches, 1-> 12 Inches, 2-> 15 Inches.
- 7) **showWindow**, If this value is greater than 0 the window showing the status of the communication is shown.
- 8) **closeTime**, It determines the time which the communcation window will wait to close after the process ends.

Returned value: The function returns a string. The possible values are the following:

- "OK" -> The process has finished successfully
- "A.B.C.D" -> The string contains the IP of the scales that have failed. If there is more than one the IPs are spitted by semicolon (";")
- "No commL.dll" -> commL.dll does not exist or can not be loaded.
- "No iconv.dll"-> iconv.dll does not exist or can not be loaded. Only applies in Unicode version.

2.2.2 SendImagetToMultiScales

SendImagetToMultiScales is a method of ImagePalletized.

This function transforms an image obtained by parameter into image registers and sends the information to the desired scales.

```
public static string SendImagetToSingleScale(DibalScale[]
scaleInfo, int scalesNumber, int imageType, int indexImage, string
pathImage, int assignPlu, int displaySize, int showWindow, int
closeTime)
```

- 1) **scaleInfo**, It is an array of structures which contains the information the scales. The type of the structure is DibalScale.
- 2) **scalesNumber**, The number of the scales to send the image.
- 3) **imageType**, The type of the image. 1-> Publicity 2-> Item Image.
- 4) **indexImage**, it is code or ID number of the image. Int variable of 4 bytes.

- 5) **pathImage**, Location and name of the image to process. String variable.
- 6) **assingPlu**, It is used to determine if the image must be associated or not to the article whose code matches with the number value of index field.
- 7) **displaySize**, It is used to determine the size of the display of the scale which will receive the image. 0-> 7 Inches, 1-> 12 Inches, 2-> 15 Inches.
- 8) **showWindow**, If this value is greater than 0 the window showing the status of the communication is shown.
- 9) **closeTime**, It determines the time which the communication window will wait to close after the process ends.

Returned value: The function returns a string. The possible values are the following:

- "OK" -> The process has finished successfully
- "A.B.C.D" -> The string contains the IP of the scales that have failed. If there is more than one the IPs are spitted by semicolon (";")
- "No commL.dll" -> commL.dll does not exist or can not be loaded.
- "No iconv.dll"-> iconv.dll does not exist or can not be loaded. Only applies in Unicode version.

2.2.3 SendMultimagestoSingleScale

SendMultimagestoSingleScale is a method of ImagePalletized.

This function scans a folder passed by parameter, transforms the images located in this folder and sends the information to the desired scale.

The image must have a valid format in order to be processed by the library. The format is the following:

"d.xxx" where d is an decimal value which indicates the index of the image and xxx is the extension of the image file.

If a file contained in this folder is not an image or its name is not formatted correctly the file is moved to the Image_FormatError subfolder.

It exists the possibility of applying a filter in the scanning like this example:

"C:\ImageFolder*.jpg" or "C:\ImageFolder*.*"

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

Parameters:

- 1) **scaleInfo**, It is a structure which contains the information the scale. The type of the structure is DibalScale.
- 2) **imageType**, The type of the image. 1-> Publicity 2-> Item Image.

- 3) **imageFolder**, The path of the image folder. String variable.
- 4) **assingPlu**, It is used to determine if the image must be associated or not to the article whose code matches with the number value of index field.
- 5) **displaySize**, It is used to determine the size of the display of the scale which will receive the image. 0-> 7 Inches, 1-> 12 Inches, 2-> 15 Inches.
- 6) **showWindow**, If this value is greater than 0 the window showing the status of the communication is shown.
- 7) **closeTime**, It determines the time which the communication window will wait to close after the process ends.

Returned value: The function returns a string. The possible values are the following:

- "OK" -> The process has finished successfully
- "A.B.C.D" -> The string contains the IP of the scales that have failed. If there is more than one the IPs are spitted by semicolon (";")
- "No commL.dll" -> commL.dll does not exist or can not be loaded.
- "No iconv.dll"-> iconv.dll does not exist or can not be loaded. Only applies in Unicode version.

2.2.4 SendMultimagestoMultiScales

SendMultimagestoMultiScales is a method of ImagePalletized.

This function scans a folder passed by parameter, transforms the images located in this folder and sends the information to the desired scale.

The image must have a valid format in order to be processed by the library. The format is the following:

"d.xxx" where d is an decimal value which indicates the index of the image and xxx is the extension of the image file.

If a file contained in this folder is not an image or its name is not formatted correctly the file is moved to the Image_FormatError subfolder.

It exists the possibility of applying a filter in the scanning like this example:

"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)
```

Parameters:

- 1) **scaleInfo**, It is an array of structures which contains the information the scales. The type of the structure is DibalScale.
- 2) **scalesNumber**, The number of the scales to send the image.
- 3) **imageType**, The type of the image. 1-> Publicity 2-> Item Image.
- 4) **imageFolder**, The path of the image folder. String variable.

- 5) **assingPlu**, It is used to determine if the image must be associated or not to the article whose code matches with the number value of index field.
- 6) **displaySize**, It is used to determine the size of the display of the scale which will receive the image. 0-> 7 Inches, 1-> 12 Inches, 2-> 15 Inches.
- 7) **showWindow**, If this value is greater than 0 the window showing the status of the communication is shown.
- 8) **closeTime**, It determines the time which the communication window will wait to close after the process ends.

Returned value: The function returns a string. The possible values are the following:

- "OK" -> The process has finished successfully
- "A.B.C.D" -> The string contains the IP of the scales that have failed. If there is more than one the IPs are spitted by semicolon (";")
- "No commL.dll" -> commL.dll does not exist or can not be loaded.
- "No iconv.dll"-> iconv.dll does not exist or can not be loaded. Only applies in Unicode version.

2.2.5 Dibalscale structure

Dibalscale is a public structure of DibalImage.dll. It is used to contain the information of the scale. In some function the parameter must be passed as an array structures and others is only a single structure value.

The structure of Dibalscale is the following:

```
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;
}
```

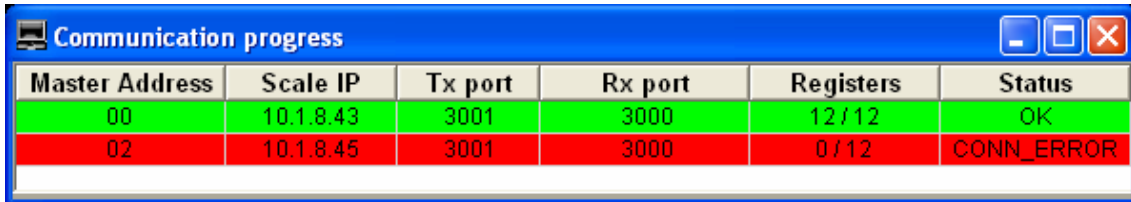
Members:

- 1) **masterAddress**, It is the master address of scale. The minimum value is 0 and the maximum 99. It is an int value(4 bytes).
- 2) **IpAddress**, The scale IP adress. It is a string value.
- 3) **txPort**, The transmission(TX) port set in the scale.
- 4) **rxPort**, The reception(RX) port set in the scale

- 5) **model**, It is an string value which defines the model the scale. Two possible values:
"500RANGE"
"LSERIES"
- 6) **display**, It is an string which defines the type of display. Not in use.
- 7) **section**, it defines the section of the scale. It is an string value.
- 8) **group**, It is the group of the scale. The minimum value is 0 and the maximum 99. It is an int value(4 bytes).
- 9) **logsPath**, Path for the logs file, this file have all the registers of communication. If it is empty the communication logs will not be recorded.

3- COMMUNICATION STATUS WINDOW

Dibalscop.dll shows a window containing data about the scales to communicate, the number of the registers sent, and the final state of the communication process.



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

“Status” column shows the final state of the communication and can contain the following messages:

- **NO CommL**: Communication has not started, because “commL.dll” file, which is necessary to communicate with scales, is not found.
- **No iconv**: Only applies in UNICODE version. It appears when an error occurs loading iconv.dll.
- **OK**: Communication successfully completed. All registers have been sent to the scale.
- **CONN_ERROR**: Communication has not started, because the scale is not found and the connection can’t be established.
- **SEND_ERROR**: An error happened when sending registers. The connection with the scale is successfully established, but all the items could not be sent. “Registers” column shows the number of registers successfully sent.