



User Guide

Graphics Composer Software Tool

Document Date: 26th January 2010
Document Revision: 2.1

Abstract

This user guide helps users to develop applications on the Graphics Composer and be able to use external memory available in the form of SD cards on the range of 4D products. This document applies to the Graphics Composer version 2.1.3.0 release or above.

These application notes have been developed after a series of testing of all the features of Graphics Composer. The application has been verified after these tests with a number of μ SD cards with sizes ranging from 64MB to 4GB (partitioned).

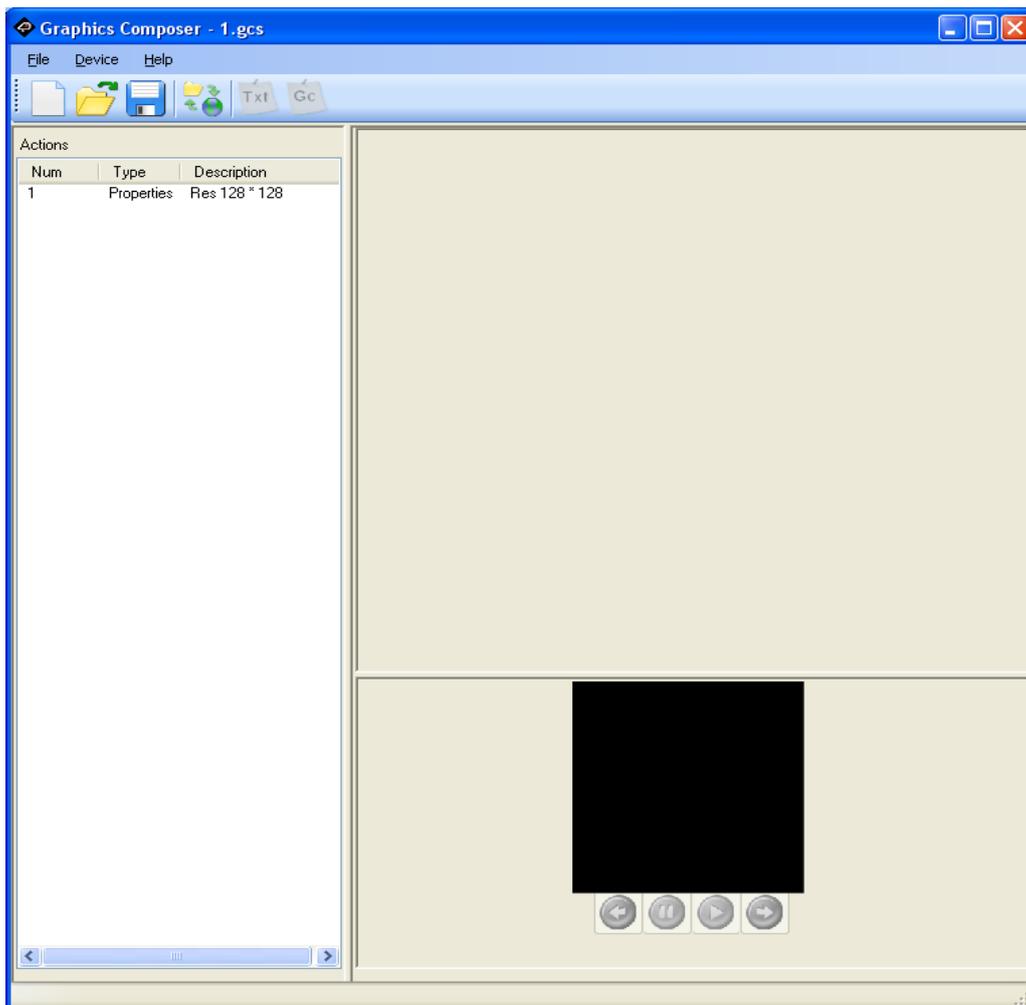
1 INTRODUCTION

Graphics composer is a free software tool for Windows which is an aid to composing slide shows and loading data into the memory card in a format readable by 4D-LAB's processors. Images, animations and movie clips can be downloaded on to the SD/ μ SD/MMC memory cards and used with the appropriate command set. Multimedia objects can be viewed with different display settings and resolutions. After building the application on the graphics composer, .txt or .gc files are generated which provides detailed low level information about the file location, size and platform etc.

1.1 Installation Guide

- The latest version of the Graphics Composer application can be downloaded from the download section on our website: http://www.4dsystems.com.au/downloads/Graphics_Composer/Ver2.XX/GraphicsComposer.zip
- It runs on Microsoft Windows Operating System (XP, Vista, Windows7) that supports the .Net framework version 2.0 or higher. If your existing Windows operating system does not have this installed then a free version can be downloaded and installed from the Microsoft web site.
- You will also need a card Reader/Writer that is capable of writing to μ SD memory cards.

2 GETTING STARTED



- Make sure the .Net environment is running on your system.
- You need to use a card reader for downloading data to the memory card.
- Run Graphics Composer.exe application.
- For 4D Serial Display Platform use following settings:
 - "Device" -> "Load Options" -> "Drive" drive name, e.g E:\
- For 4DGL Platform use one of the following settings:
 - "Device" -> "Load Options" -> "GCI File"
 - "Device" -> "Load Options" -> "µSD Raw"
 - "Drive" drive name, e.g E:\
 - "Sector offset" First sector address where you wish to start writing to the card.
 - "Device" -> "Load Options" -> "Fat16 Raw Partition"
 - "Drive" drive name, e.g E:\
- Click on the "File -> New" menu item or the "New" icon, select a location and name for your Graphics Composer project and click "Save" you should see a new line (Type – Properties) in the "Actions" window indicating the display resolution. If the resolution does not match that of your device, select the line and modify the device X and Y resolutions in the upper-right window. Click on the "Actions" window and the line will be updated.
- Right-click in the "Actions" window and select; Add, Insert or Delete action, then the Delay, Erase Screen, Image or Movie object. In the right-side windows you will see dialog boxes for selecting an image file, movie

file, clear screen command or delay and selecting the attributes that you wish to use and in the bottom-right window a preview of how the image will appear on the attached display.

Note: When using “Insert”, the new item will be inserted above the selected item in the action list.

Note: For 4DGL platform there is no need to add/insert delays or to clear screen.

- Click in the “Action” window to update the current “line”
- Continue adding images, movies, clear screen and delays as above until you have the desired slideshow built.
- Click the “File, Save” or “File, Save As” (if you wish to change the location or name) menu item or the “Save” icon to save your project.
- To load the images onto the device, click the “Device, Load” menu item or the “load Device” icon. A dialog box will show the progress of the download.
- The Graphics Composer generated files and their meanings are explained below,
 - *.gcs file: The project file generated/modified when you save a project on the Graphics Composer.
 - *.txt file: This file is generated/modified when a slide show is burnt to the SD card using “Serial Platform” load option. The file has same name as your project file (*.gcs). Text file contain names and the locations of all the multimedia objects loaded by that project.
 - *.gc file: This file is generated/modified when a slide show is burnt to the SD card using “4DGL Platform”, sub option “uSD Raw” or “FAT16 Raw Partition” selected.
 - *.GCI file: This file is created when “GCI file” Load option is selected. It contains all the multimedia objects the are added in the Graphics Composer project at certain location according to their oder in the project.
 - *.DAT file: This file is created with the GCI file, with same name as that of GCI file. It contains the address locations of the multimedia objects within the GCI file. The arrangement of address in this file is LSW, MSW.

3 CREATING PROJECTS

3.1 Creating a new Project

A new project can be created by clicking 'New' on the file menu. You will be asked to save the project name. Save the project to the appropriate location. You are now ready to build your first slide show. On the Actions List, the 'Properties' parameter will appear which need to be set to the appropriate resolution. It should match your module's display resolution (horizontal * vertical).

As soon as the file is saved a folder is created with .gcs.d extension. Its usage will be elaborated later in the document. Once the file is edited by adding an image or a delay etc. you would be able to save the file. This will create the .gcs file which is the actual project file or a Graphics Composer file.

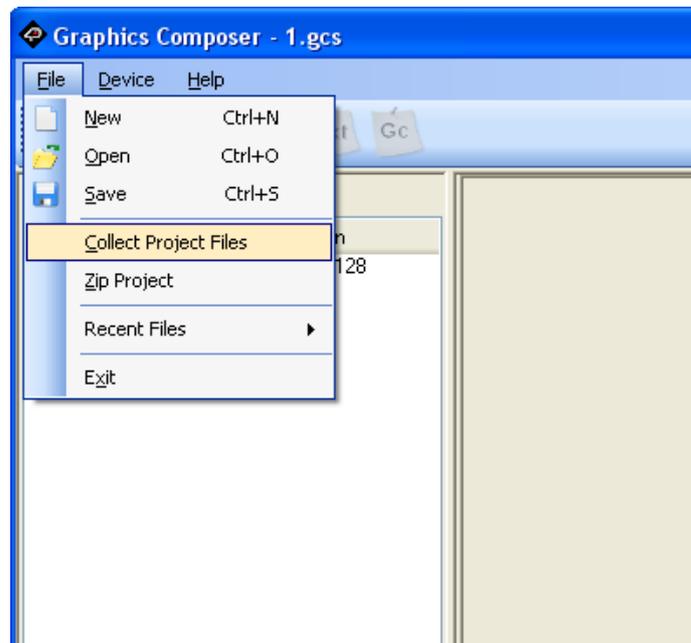
3.2 Opening an existing Project

If you wish to retrieve a previously saved project, click 'Open' in the File menu. Browse for the desired project file location and your project will be loaded. The project files are saved in .gcs format. Hence you must look for the file with the .gcs extension.

3.3 Compiling a Project

The 'Collect Project files' option in the "File" menu allows the user to collect all the multimedia files into one folder, which is created with the project, with an extension .gcs.d.

Subsequently all the user needs to do, after adding the required files into the project and building the slide show, just click 'Collect Project files' and all the multi media objects used will be copied to the projects folder. This is to increase the portability of the projects. The 'Zip Project' is yet another option which allows the user to collect all the project files and zip them together. It includes .gcs file, the .gcs.d directory, the low level details in a text file and other relevant files.



4 COMPOSING IMAGES AND MOVIES

Multimedia objects can be added by simply right clicking the mouse in the 'Actions' List section. If you have not created a project file yet, right click will open up a pop up window 'New'. Save your project to the desired location. The "Properties" item will appear as the first entry index 1 of the list box. Set the resolution to that of your display module.

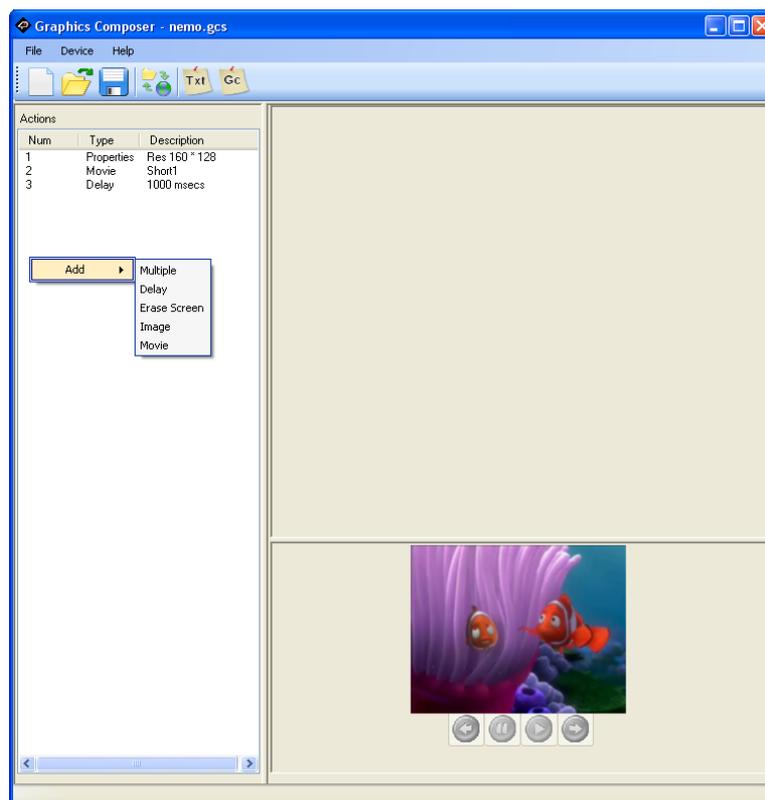
4.1 Slideshows

The slideshow script is only created when selecting the 'Serial Platform'. The script can be ignored and the images used by Serial or 4DGL programs.

4.2 Adding files and features

Right click on the list box, and a pop up window with "Add" will appear. Expanding it, will take you to the list box with the following options:

- Multiple
- Delay
- Erase Screen
- Image
- Movie



4.2.1 Adding an Image

Select the 'Image' option and browse for the desired image file.

Note: When adding an image, do not select a movie/video or gif animation file, the graphics composer will malfunction and you would need to exit and reload it. An animation (gif) or a movie/video can be added only by selecting 'Movie' option.

4.2.2 Adding Movies

Similarly, movie files can be added to the slide show. By selecting the item from the list box, the Multimedia object can be viewed on the Graphics composer as it is. The object can also be viewed, as appearing on the bottom of GC window, how it will appear on the display module.

4.2.3 Multiple additions

Multiple is similar to the 'Image' or a 'video' option, except it allows you to choose multiple files at the same time by using 'Shift' or 'Control' key.

4.2.4 Adding Delays

Delay option allows the user to add a delay between objects in the slide show. A delay value in the range of 1 millisecond to a maximum of 65535 milliseconds can be added.

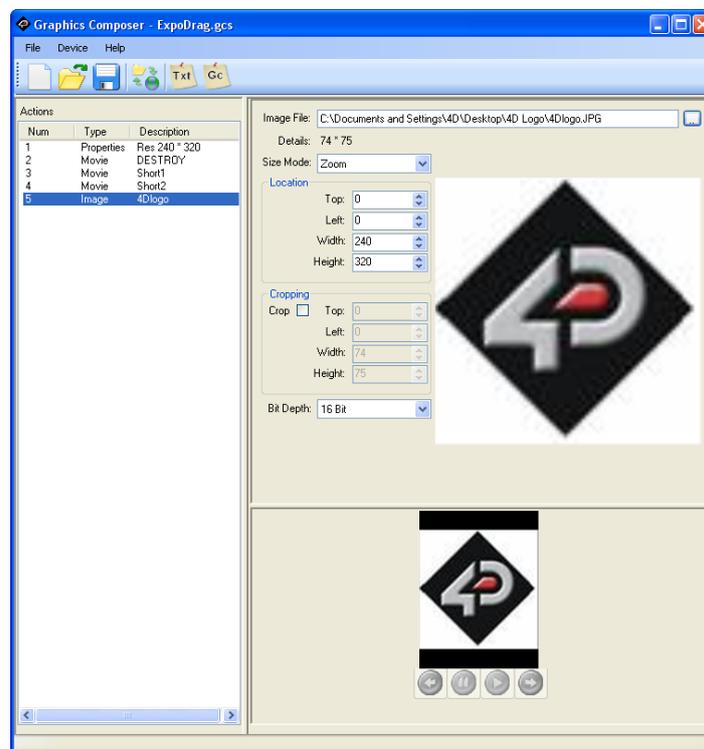
4.2.5 Erase Screen

'Erase screen' clears the screen. It is applicable to slide shows only.

4.3 Settings

4.3.1 Image Settings

The original details of the image can be viewed besides the 'Details' label.



- 'Size mode' can be set by selecting following options:
- Centre, focuses the centre of the image
- Topleft, focuses the top left area of the image
- Stretch, stretches the image to the width and height set under the location tab.
- Zoom, sets the focus to the complete image with a size specified by Width and height under the 'Location Tab'.

- ‘Cropping’ allows you to crop the image. Click the Cropping Check box; following options will be enabled:
 - Top, crops the top of the image
 - Left, crops the image from the left
 - Width, crops the width of the image
 - Height, crops the height of the image
- ‘Bit Depth’ is used to set the number of bits to represent one pixel. An 8 bit dept will not give a good visual quality but it will fit in to a smaller space. A 16 bit depth utilizes 16 bits to represent 1 pixel and hence takes more space and gives a better visual quality.

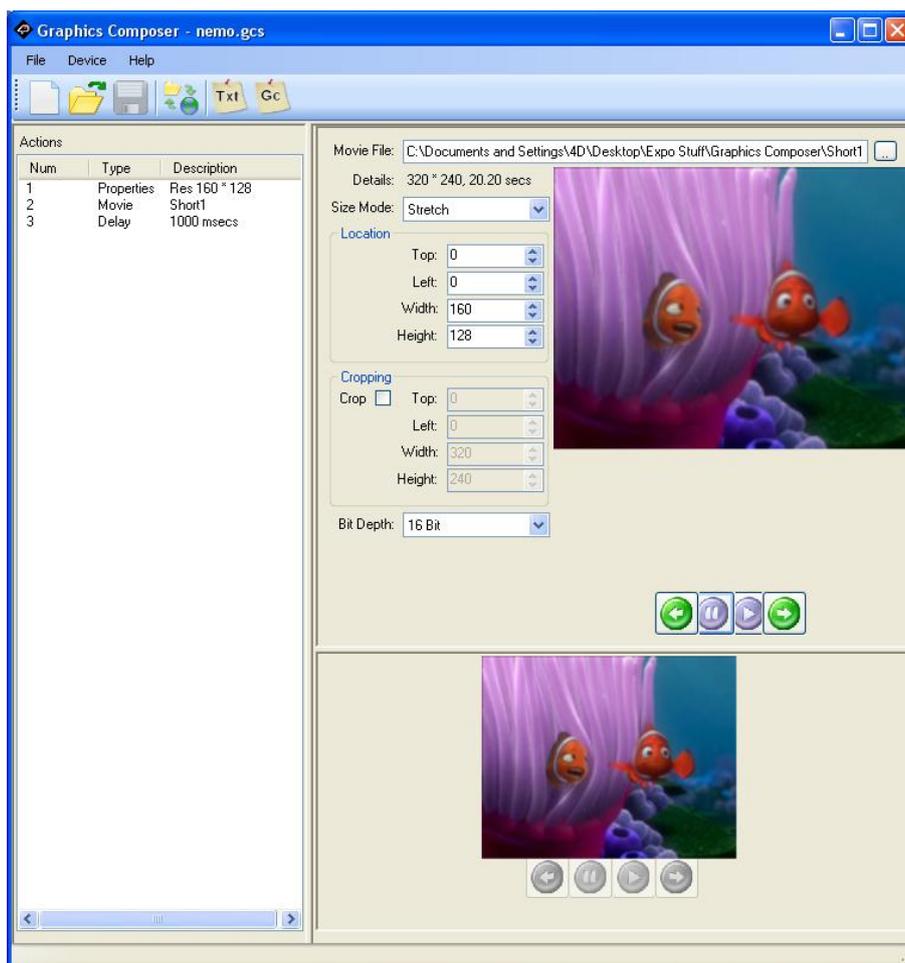
4.3.2 Video Settings

The details of the video can be seen besides the ‘Details’ label.

All the settings for the images apply equally to the videos. The location, cropping and bit depth details can be set as required.

4.4 Previewing

The previewing option allows the user to preview the multimedia objects by selecting them from the list box and using the control buttons.

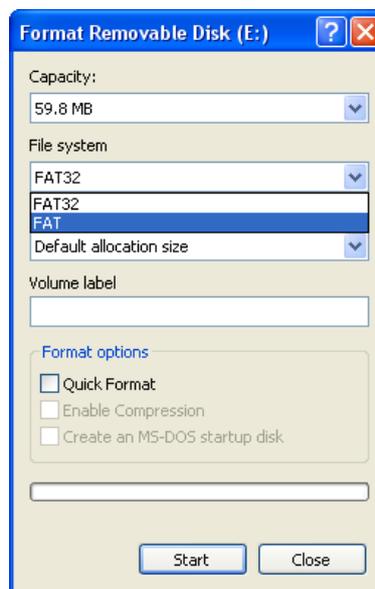


5 LOADING AN APPLICATION TO THE μ SD CARD

There are two options of which the user can select, based on the two platforms:

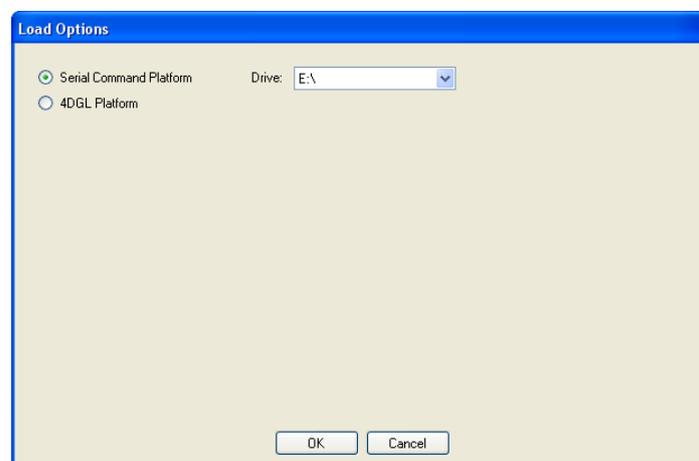
1. Serial Platform
2. 4DGL Platform
 - GCI File
 - FAT16 Raw Partition
 - μ SD Raw

Make sure that the μ SD card is FAT16 formatted before downloading on to the memory card. In case of a FAT16 Raw Partition, the user is asked for the formatting in the middle of loading. This is to create the partition in the memory card to spare the unused space to be used under FAT16 format. On Windows XP and later versions of windows it is named as "FAT" instead of FAT16. Make sure you do a complete format, as a quick format does not format the boot sector.



5.1 Serial Platform

Serial platform, in the Graphics Composer, is mainly designed to work with the SGC based modules. An additional feature of the serial platform is the Auto Run or Slide show feature. The G1(SGC), GMD1, PMD2 and PMD3 modules have a 'run' jumper. The delay and Erase Screen only exist to aid in this process. the slide show repeats forever, until either the RUN jumper is taken off or the module is powered off.



Choose the Serial Platform option and the appropriate drive from the combo box. Click 'OK'. Now, click 'Load' from the Device menu. The slide show will be downloaded on to the μ SD card. Plug your card in to the supported module. Install the Run jumper and power up your module. Make sure the memory card and the Run jumper is installed before the module is powered up. Also, make sure relevant SGC PmmC file is loaded in to the.

In order to control the module serially and send commands through the serial platform, the user needs to remove the Run jumper. Now, you are ready to send serial commands via the serial interface. The command set to control these images is listed in the txt file generated after loading the memory card under the serial platform. The txt files are discussed in details underneath.

Following modules comply with the Serial Command Platform of the Graphics Composer:

- xxxx-G1(SGC)
- xxxx-P1X(SGC)
- xxxx-GMD1
- xxxx-PMD2
- xxxx-PMD3

Note: Please make sure you load the SGC PmmC file to use the SD card under "Serial Command platform" option.

Note: For PICASO-SGC modules you need to ensure the image format is set to "old format" (this is the default). You can use following command to set the image format.

Display Control Functions - 59Hex

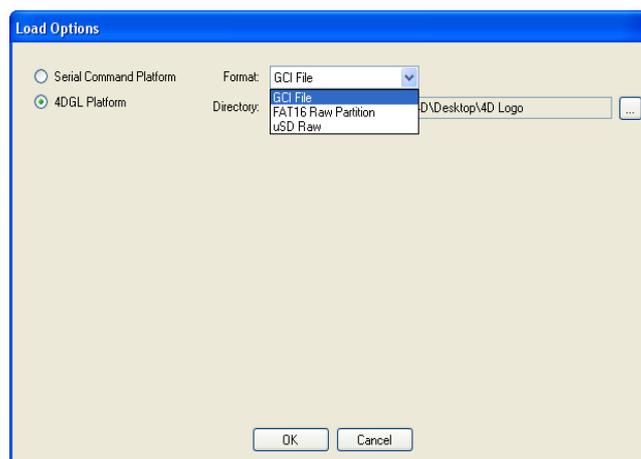
command: cmd, mode, value

Refer to Sec 2.1.6 on the PICASO-SGC Command Set manual for details.

5.2 4DGL Platform

There are three possible options under the 4DGL platform that provides a complete control of the multimedia object to the user.

- GCI File
- μ SD Raw Partition
- FAT16 Raw Partition



PICASO Processor, that supports FAT formatting, does not support subdirectories. There should not be any subdirectories.

5.2.1 GCI File

A GCI file option stores the complete slide show in to one GCI file that holds all the multimedia objects. While another file with an extension of DAT is created. It holds the information regarding the organization of the multimedia objects inside the GCI file. Hence both the files are used in conjunction. Select '4DGL Platform' in the Load option; select the 'GCI File' from the Format Combo box. Set the location where you need to save the files and press 'OK'. Click 'Load', the files will be created and saved to the set location. The files need to be copied to the μ SD card via a memory card reader on the computer. The multimedia object can now be accessed via 4dgl commands through the module.

TestImageControl.4dg in the Project Picaso – Image Control is a good example of using the 4DGL instructions to access and control the multimedia objects in the μ SD card in the form of GCI files. Make sure Test1.GCI and Test1.dat are copied to the μ SD card. These files are already created and can be found in the X:\...\4DGL Workshop Beta X.X\Resource\Copy to μ SD folder.

Example

```
file_LoadImageControl("controlfile", "imagefile", mode);
hImageList := file_LoadImageControl("fname1.dat", "fname2.gci", 0);
```

Reads a control file from the disk to create an image list.

Returns NULL if function fails.

Returns a handle (pointer to the memory allocation) to the image control list that has been created.

This function Calculates the size of a chunk of memory required for an image list and populates it from the image control file (*.dat) therefore, when imagelist is no longer required, you must de-allocate the image list memory by using mem_Free(hImageList) to restore the heap. Both files "fname1.dat" and "fname2.gci" are created by the Graphics Composer software tool by selecting GCI File load option.

Note: GCI File format is not supported by **GOLDELOX** based modules as the processor does not support FAT system.

Note: GCI File format can also be used for **PICASO(SGC)** based modules. Following command can be sent from the host controller to display multimedia objects on the screen, while GCI file is loaded to the SD card. DAT file provides details of the image location within the file.

Display Image-Icon from Card (FAT) - @6Dhex

```
Command: ext_cmd, cmd, "file_name", terminator, x(msb:lsb), y(msb:lsb),
imagePos (msb:lsb)
```

File_Name: Must be GCI filename with ".GCI" extension

x(msb:lsb), y(msb:lsb): Can be extracted from the .DAT file.

Following range of display modules support SD cards programmed under GCI File option.

- xxxx-PMD2
- xxxx-PMD3
- xxxx-P1X
- xxxx-P1X(SGC)

5.2.2 μ SD Raw

μ SD raw copies the slide show to the SD card in the raw format. Choose the μ SD Raw in the Format combo box, the appropriate drive SD card is plugged in, and the sector offset. Click OK. Press Load and you are ready to plug in the SD card in the module and access it from the module using 4DGL commands. Once the program is downloaded from the graphics composer all the space inside the memory card becomes inaccessible by the computer. In other words the file formatting is lost. The raw form of data in the SD card is accessible from the module. A range of 4DGL commands are available under 4DGL to access the data. Some are given below.

Example:

```
uSD_SetSector (SectHiWord, SectLoWord);
```

The instruction sets μ SD card internal Sector address pointer for sector block access. Each sector = 512 bytes. SectHiWord specifies the high word (upper 2 bytes) of a 4 byte μ SD card sector address location. SectLoWord specifies the low word (lower 2 bytes) of a 4 byte μ SD card sector address location.

```
uSD_Image(x, y);
media_Image(x,y); //For GOLDELOX-GFX2 based modules
```

Displays an image from the μ SD card at screen location specified by x,y (top left corner). The location of the Image in the μ SD is as specified by uSD_setSector() or media_setSector function. Image parameters such as width, height and pixel width are embedded into the image data block.

```
uSD_Video(x, y);
media_Video(x,y); //For GOLDELOX-GFX2 based modules
```

Plays a Video/Animation clip from the μ SD card at screen location specified by x,y (top left corner). The location of the Video clip in the μ SD card is as specified by uSD_SetSector() or media_setSector function. Video parameters such as width, height, frames, and inter-frame delay are embedded into the video data block.

Writing multimedia objects to the SD card in Raw format saves a lot of code and time on the processor. The multimedia objects are displayed faster as compared with the GCI file option.

PICASO-SGC MODULES

With the Picaso-SGC modules, a partitioned 4GB card can be used. You can make two partitions on the 4GB card. You can do this by using our software tool called REMPET. Say, you allocate 50% for the first partion. You would end up getting half of the 4GB card with FAT16 (You will have to format the first partition with FAT16 once you have created parttions) and rest usable as a RAW. Use the first NonFS sector as the offset on your "uSD Raw" option on the Graphics Composer to write RAW data to the RAW partion leaving the first partion still usable under FAT16 formatting. This setting is useful when you need higher frame rates which can be achived only when multimedia objects are burnt to the SD card under RAW or Serial RAW options. Once the images/videos are loaded to the SD card, you can use following commands to display them.

Use this command to set the image format. You need to set the image format to "new format"

Display Control Functions - 59Hex

```
command: cmd, mode, value
```

used to set "Image Format". Refer to Sec 2.1.6 on the PICASO-SGC Command Set manual.

Display Image-Icon from Card (RAW) - @49hex

```
command: ext_cmd, cmd, x(msb:lsb), y(msb:lsb), width(msb:lsb), height(msb:lsb),
colourMode, SectorAdd(hi:mid:lo)
```

Used to display image-icon at a specific location on the display.

Display Video-Animation Clip from Card (RAW) - @56hex

```
Command: ext_cmd, cmd, x(msb:lsb), y(msb:lsb), delay, SectorAdd(hi:mid:lo)
```

Used to display video-animation clip at a specific location.

Following range of display modules support this option,

- xxxx-PMD2
- xxxx-PMD3
- xxxx-G1(GFX)
- xxxx-P1X
- xxxx-P1X(SGC)

5.2.3 FAT16 Raw Partition

A FAT16 Raw partition allows the user to write directly to the card through the computer yet save some space for the data storage unlike μ SD Raw. Choose FAT16 Raw Partition from the Format combo box and set correct drive where the memory card is plugged in. Click OK and then Load. The txt file is created where the project is saved. This file contains low level details of the multimedia objects loaded in to the card such as sector address, width or height of the object, frames and delay etc.

The greatest advantages of the FAT16 Raw partition is their ability to occupy only the required space and spare the rest for user unlike the μ SD Raw or Serial platform options. Command set similar to the one shown above can be used to access the data.

Writing multimedia objects to the SD card in FAT16 Raw format is even better than uSD Raw. You can write videos, which usually need faster processing to achieve better frame rates, to the SD card in raw format with the left over space still accessible from the processor under FAT formatting. The multimedia objects are displayed faster as compared with the GCI file option.

Following range of display modules support this option,

- xxxx-PMD2
- xxxx-PMD3
- xxxx-P1X
- xxxx-P1X(SGC)

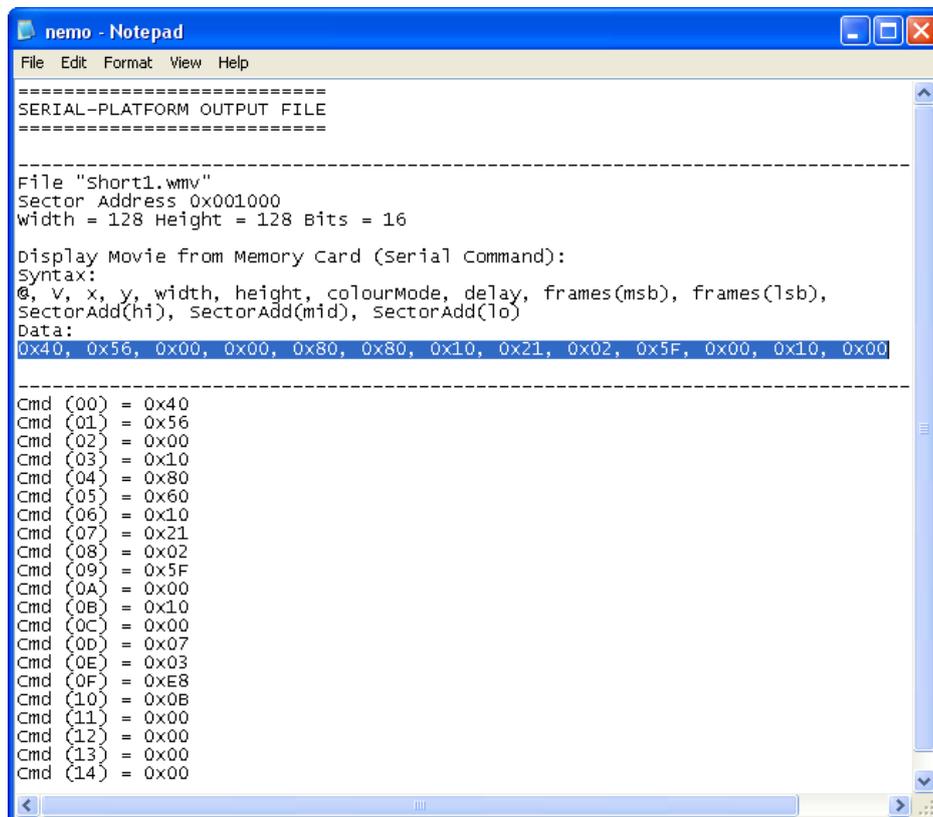
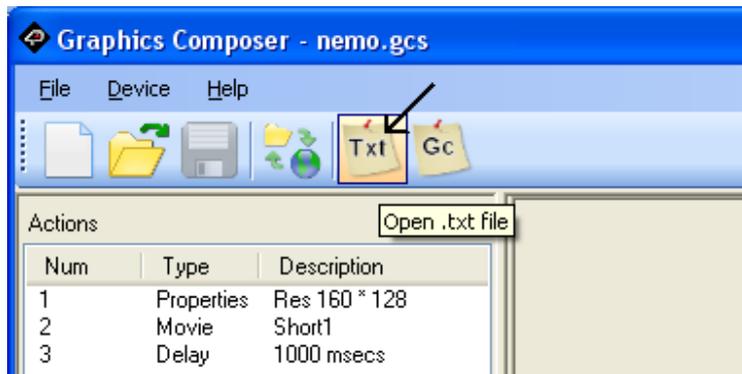
6 LOW LEVEL DETAILS

As soon as the project is downloaded the text or GC file is updated. These files contain the low level/byte-wise details. They can be used to control the multimedia objects. There are two types of files generated in two different cases.

- Txt file for the Serial Command Platform
- GC file for the 4DGL Platform

6.1 Txt File

The text file created with the project can be viewed by clicking the following icon. Every time the memory card is loaded with serial command platform the txt file is updated. Click on the txt icon as shown in the figure below,



The file is titled 'Serial-Platform Output File' with the details as shown in the sample above. In order to control the multimedia objects on the memory card, under serial platform, the user needs to remove the Run jumper if he does

not want to auto run the Slide Show. In such case, the txt file as shown below provides the command set used to display the image/animation/video on the module.

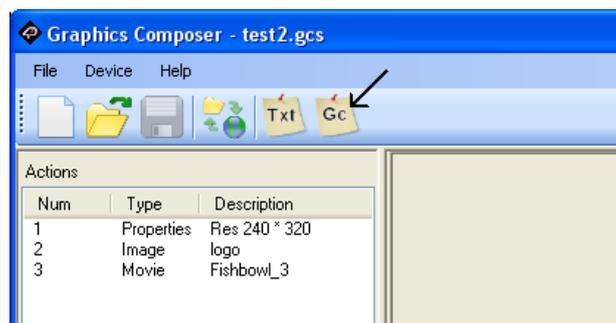
Example

A series of commands in the form of serial data need to be sent to the serial port of the module with the memory card plugged in to the module.

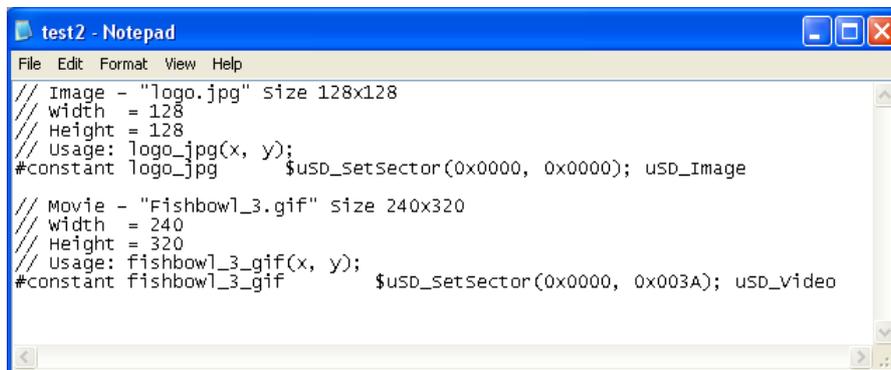
These are the 8 bit (Hex) numbers that need to be sent in the listed sequence to invoke the short1.wmv on the screen of the module.

```
0x40, 0x56, 0x00, 0x00, 0x80, 0x80, 0x10, 0x21, 0x02, 0x5F, 0x00, 0x10, 0x00
```

6.2 GC File



Click on the 'GC' icon, a file in the text format will be opened as shown in the figure below.



Every time the memory card is loaded with 4DGL command platform the GC file is updated. The GC file provides size, width, height, location and syntax of usage to help user for a quick start.

Example

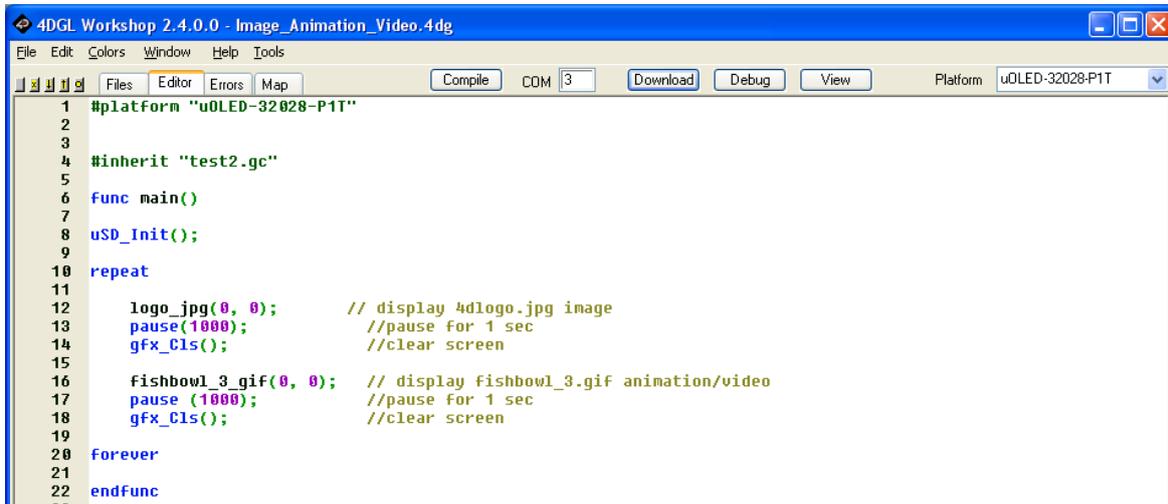
The above file provides the sector address which can be pointed to load the video. Such as, following commands can be used to display the video on the supported modules.

```
uSD_SetSector(0x0000,0x003A);
uSD_Video(0,0);

media_SetSector(0x0000,0x003A); //For GOLDELOX-GFX2 based modules
media_Video(0,0); //For GOLDELOX-GFX2 based modules
```

It will display the video on the 0,0 (x,y) position of the display.

There is a more simple method which can be used to display the images/animation/video on the display. User needs to inherit the .gc file in the 4DGL program and call the functions, more easily, defined as constants. Such as, the figure below shows how to display logo.jpg or fishbowl_3_gif.



```
1 #platform "uOLED-32028-P1T"
2
3
4 #inherit "test2.gc"
5
6 func main()
7
8   uSD_Init();
9
10  repeat
11
12     logo_jpg(0, 0);      // display 4dlogo.jpg image
13     pause(1000);        //pause for 1 sec
14     gfx_Cls();          //clear screen
15
16     fishbowl_3_gif(0, 0); // display fishbowl_3.gif animation/video
17     pause(1000);        //pause for 1 sec
18     gfx_Cls();          //clear screen
19
20  forever
21
22  endfunc
```

Note: Inheriting ".gc" file does not work for GOLDELOX-GFX2 modules as the syntax for SD card function is changed from "uSD" to "media". You can edit the GC file and change "uSD" to "media" to fix it.

7 CONCLUSION

Graphics Composer is a useful tool to enhance the features and functional possibilities with the 4D System's range of products. User can choose from a number of formats to store the data in the memory cards. A number of multimedia formats are supported such as wma, gif, jpeg, bmp, mpeg etc. A range of 4DGL instruction set can be found on the 4D Systems web site or GFX2 Command set manual, which can help the user develop programs.

PROPRIETARY INFORMATION

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavors to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.

All trademarks belong to their respective owners and are recognized and acknowledged.

DISCLAIMER OF WARRANTIES & LIMITATION OF LIABILITY

4D Systems makes no warranty, either express or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, noninfringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

Use of 4D Systems' devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.

CONTACT INFORMATION

For Technical Support: support@4dsystems.com.au

For Sales Support: sales@4dsystems.com.au

Website: www.4dsystems.com.au

Copyright 4D Systems Pty. Ltd. 2000-2009.