APPLICATION NOTES

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

Designer Displaying Images from the uSD Card – GC RAW

DOCUMENT DATE: **13th April 2019**
DOCUMENT REVISION: **1.1**

Description

This application note is a step by step procedure on how to play a video on 4D display modules. The video is saved on a RAW-formatted uSD card. Since videos are essentially a collection of images or frames, this application note is relevant to both image and video objects. A section at the end of this application note describes how images are displayed on the screen. Before getting started, the following are required:

- Any Picaso, Diablo16, or Goldelox display module. Visit www.4dsystems.com.au to see the latest products using any of these graphics processors.
- 4D Programming Cable / µUSB-PA5/µUSB-PA5-II for non-gen4 displays (uLCD-xxx)
- 4D Programming Cable & gen4-IB / gen4-Pa / 4D-UPA, for gen-4 displays (gen4-uLCD-xxx)
- micro-SD (µSD) memory card
- Workshop 4 IDE (installed according to the installation document)
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Application Overview

Playing a video on a 4D display module is one of the most useful techniques to learn, as it can be used in many different applications. This application note will guide the user through a sequence of steps needed to convert the video into 4D format and play the converted file using a 4DGL program. A series of modifications can then be applied to the video using 4DGL graphics functions.

The video needs to be saved onto the uSD card in a format readable by 4D-LAB's processors. The video therefore is not saved onto the uSD card in its original format (flv/mov/mpeg/etc) but is converted by Workshop into a graphics file, which is called the "GCI" file. The same concept applies to animations, images, widgets, and other objects. Common animation and image file formats are jpeg, gif, png, bmp, etc. GCI stands for Graphics Composer Image and it has its own format. The Graphics Composer or GC is a program used by Workshop to convert multimedia graphics files into a GCI file.

When designing in the Designer environment, the user will need to manually use the Graphics Composer. The general procedure is:

1. Add the video to the GC window.
2. Insert a uSD card into the PC.
3. Build or generate the GCI file and other supporting files and copy them to the uSD card. The user has the option of using either a FAT-formatted uSD card or a RAW-formatted uSD card.

4. Write the 4DGL code for accessing the uSD card and playing the video on the screen. The functions to be used depend on the format of the uSD card – media commands for RAW, file commands for FAT.
5. Compile the code and upload the program to the display.
6. Unmount the uSD card from the PC and insert it to the display module.
7. The program should now run on the display module and the video should now be played.

When designing in both the ViSi and ViSi-Genie environments (ViSi-Genie is not available for Goldelox display modules), the Graphics Composer is invisible to the user. Instead, the user sees only the WYSIWYG (What-You-See-Is-What-You-Get) screen during design time. The general procedure is:

1. Place the video on the WYSIWYG screen.
2. For the codeless ViSi-Genie environment, configure the properties of the video object using the Object Inspector.
3. For the ViSi environment, the code area is beside the WYSIWYG screen. Write the code for the program.
4. Click on the Build/Compile and Upload button.
5. Workshop will prompt for the uSD card drive to which the GCI file and other supporting files will be copied. Insert the uSD card to the PC and select the correct drive. Workshop transfers the files to the uSD card. The uSD card shall be FAT-formatted only.
6. Workshop uploads the program to the display module.
7. Remove the uSD card from the PC and insert it to the display module.
8. The program should now run on the display module and the video should now be played.

For more information on the use of the ViSi and ViSi-Genie environments, refer to the relevant application notes. The user will realize that using the WYSIWYG screen is much easier than using the Graphics Composer. For this reason, ViSi is the recommended environment for users intending to design GUIs (Graphical User Interface) thru coding and drag-and-drop methods at the same time.

Manual use of the Graphics Composer is not recommended since the ViSi and Visi-Genie environments are meant to "automate" the process for the user

Setup Procedure

For instructions on how to launch Workshop 4, how to open a **Designer** project, and how to change the target display, kindly refer to the section "**Setup Procedure**" of the application note
Designer Getting Started - First Project

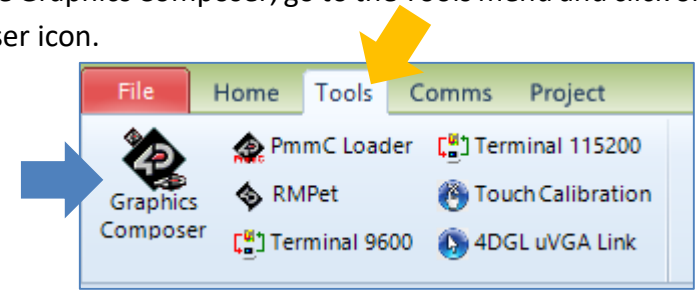
Create a New Project

For instructions on how to create a new **Designer** project, please refer to the section "**Create a New Project**" of the application note
Designer Getting Started - First Project

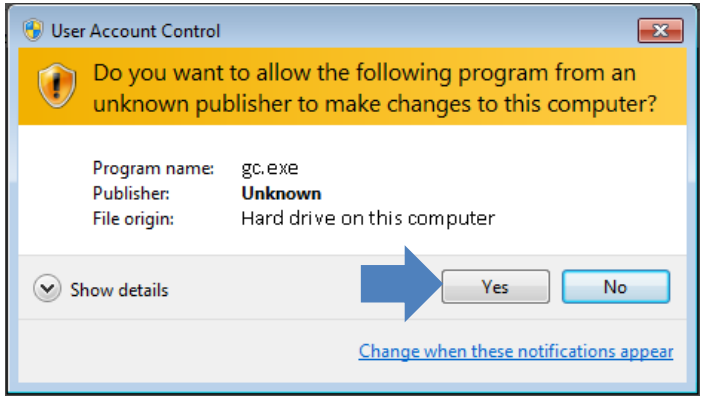
Design the Project

Open the Graphics Composer

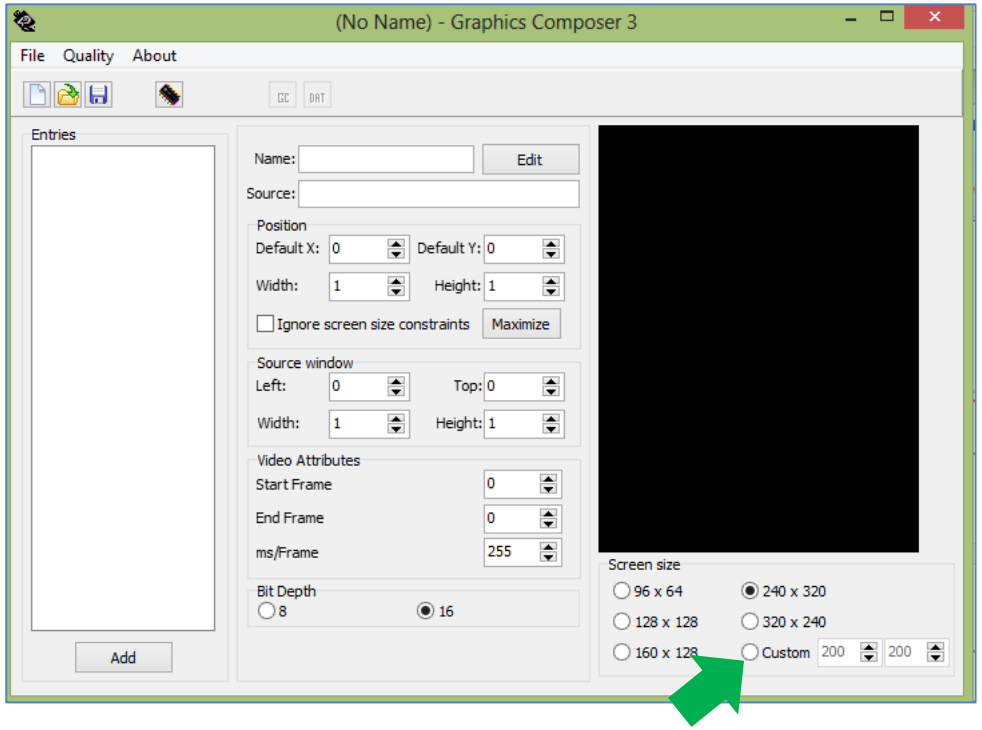
The Graphics Composer has a dedicated user guide which can be downloaded here. In this section only the basics of using it are covered. To open the Graphics Composer, go to the Tools menu and click on the Graphics Composer icon.



Depending on the user's PC User Account Control settings, Windows might ask for a confirmation to run the program **gc.exe**. This is the Graphics Composer. Click Yes.

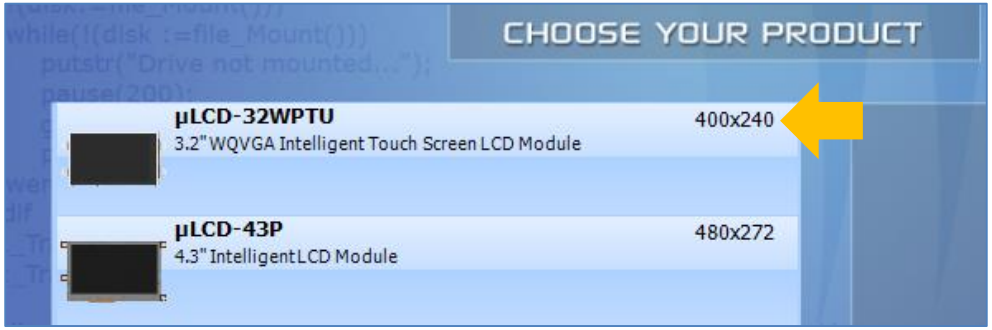


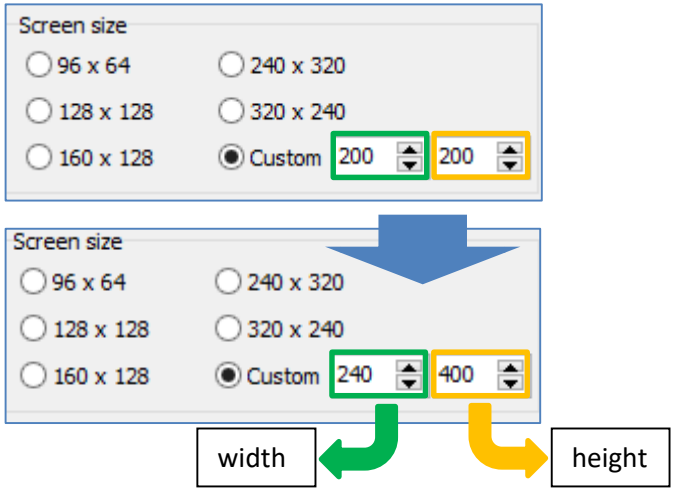
The Graphics Composer now opens.



Set the Screen Size

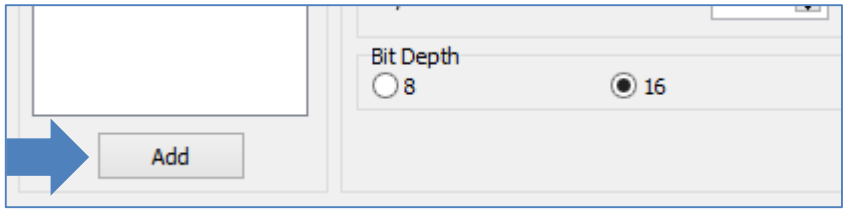
To make the screen size consistent with the selected project display (uLCD-32WPTU Portrait orientation for example), click on the custom button and change the width from 200 to 240 and the height from 200 to 400. When creating a new project, the Choose-Your-Product window also displays the pixel resolutions of the display module.



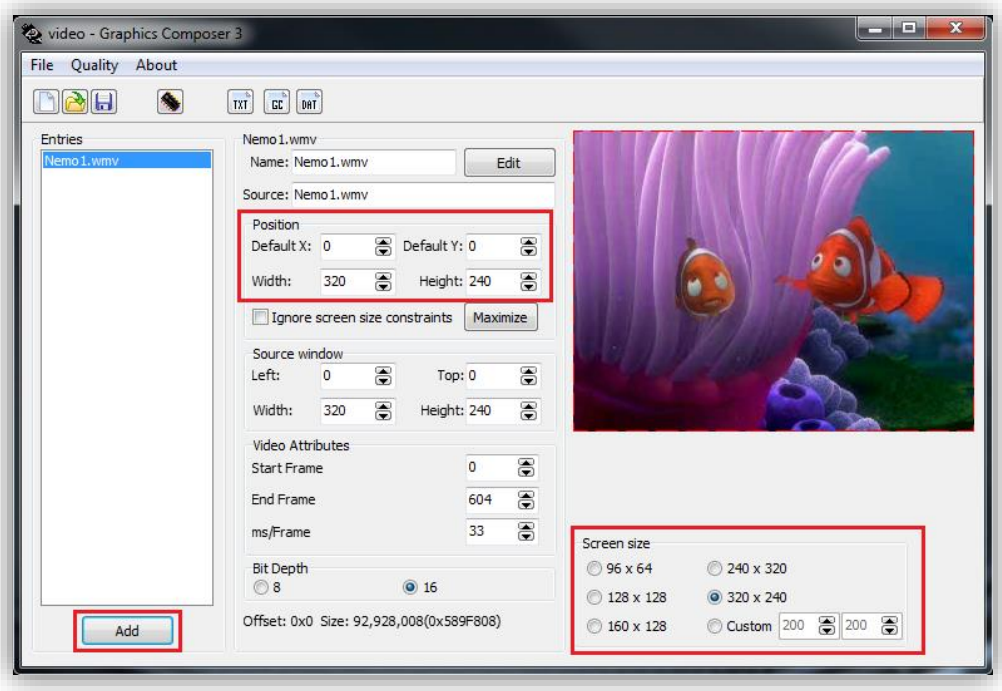


How to Add a Video

To add a video, click on the Add button on the left lower part of the window.



The standard Open window opens. Select the desired video file. The screen will be updated.



The maximum size video possible will be dependent on the size of the µSD card. Adjust various attributes of the video by either manipulating the image in the display area directly, or manually editing the X, Y, Width and Height figures as shown in the figure. Note the End frame, it is useful information required for coding. Also note, the Bit Depth would always be set to 16 bit.

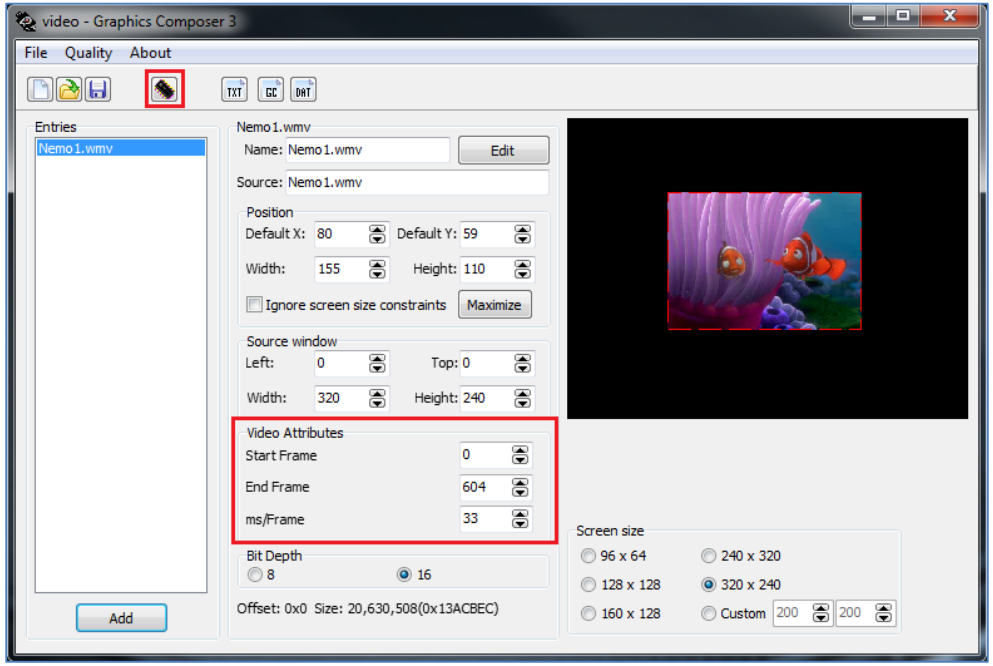
Supported Video File Formats

There are only specific file formats supported for video playback, which include:

- .avi
- .wmv
- .vob
- .mpg
- .gif

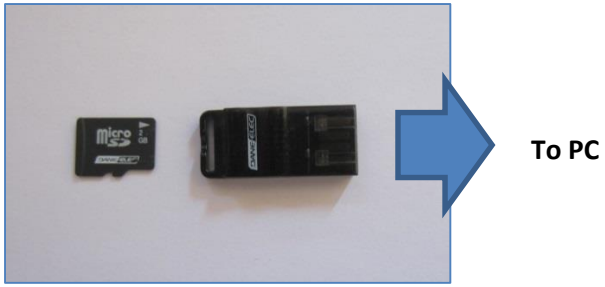
Experimenting with Video Attributes

It should be noted that playback refresh rate is affected by the desired area for the video to play in. The image above shows the video at maximum width and height of the display area. The second snapshot shows a reduced size video playback area, still within the 320x240 display area. Notice that the second video playback will be much smoother than the first. Experiment with the start and end frames as seen in the Video Attributes Section, as well as the ms/frame characteristic as highlighted in the snapshot below. Insert the µSD card into the module and click on the **Build** button in the top left hand area of the screen as denoted by the small chip.

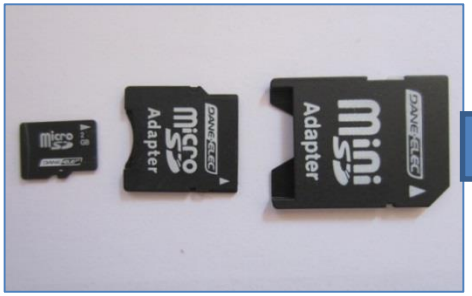


Insert the µSD Card to the PC

Insert the µSD card into the USB adaptor and plug the USB adaptor into a USB port of the PC.

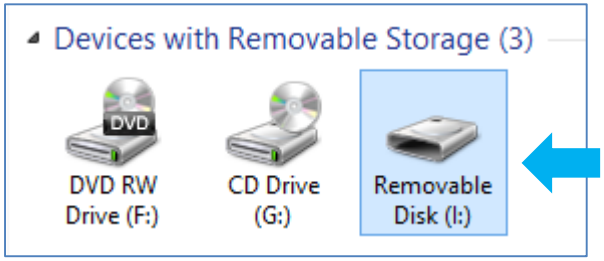


OR insert the µSD card into a µSD to SD card converter and plug the SD card converter into the SD card slot of the PC.



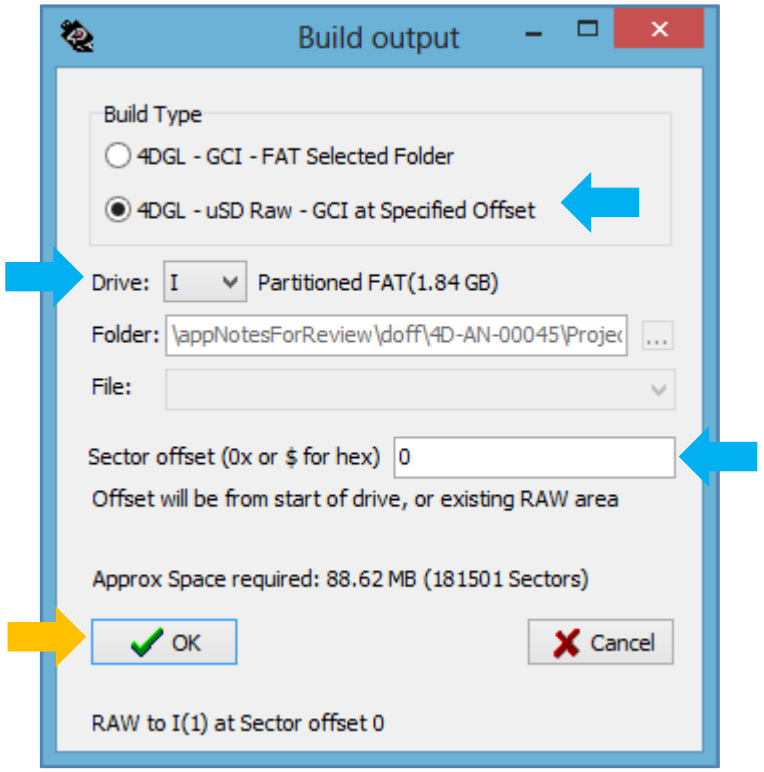
To PC

Check if the µSD card is mounted, here it is mounted as drive I:

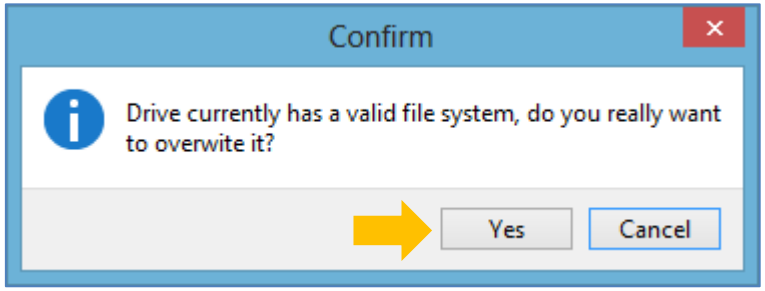


Build Output

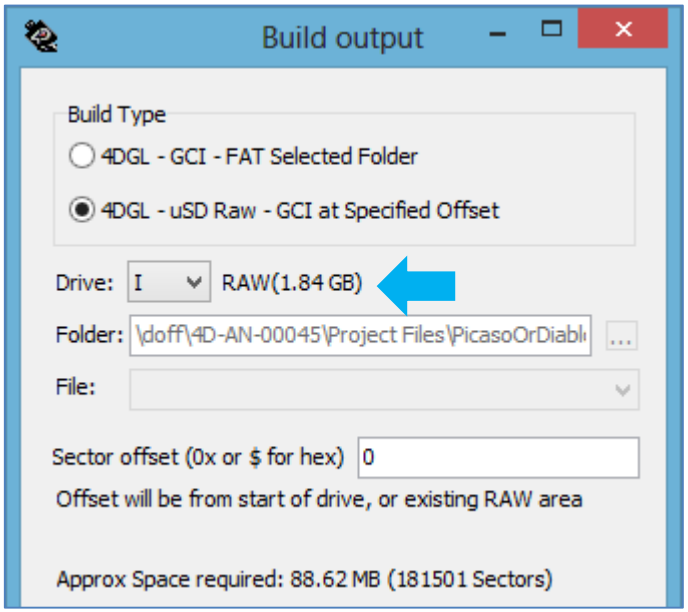
Select the options as seen below, ensuring that the µSD drive is selected and the offset set to zero.



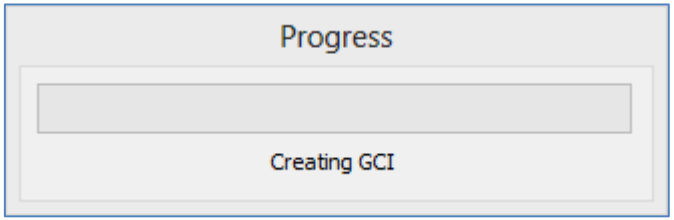
Since the uSD card has an existing file system which will be changed to RAW format, Windows will ask for permission. Click yes only when ready.



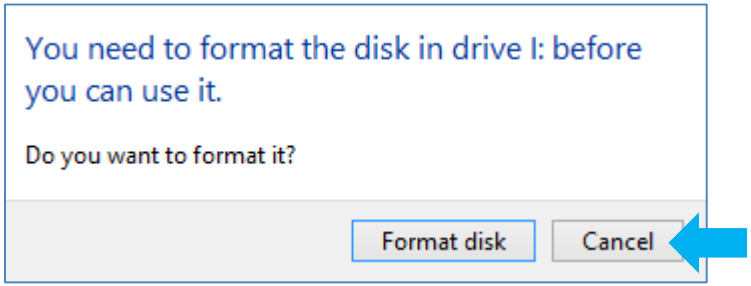
If the uSD card is already RAW-formatted, the Build output window will detect it as such. Windows will not ask for permission to overwrite in this case.



Either way, the GCI and other supporting files will now be generated and copied to the uSD card.



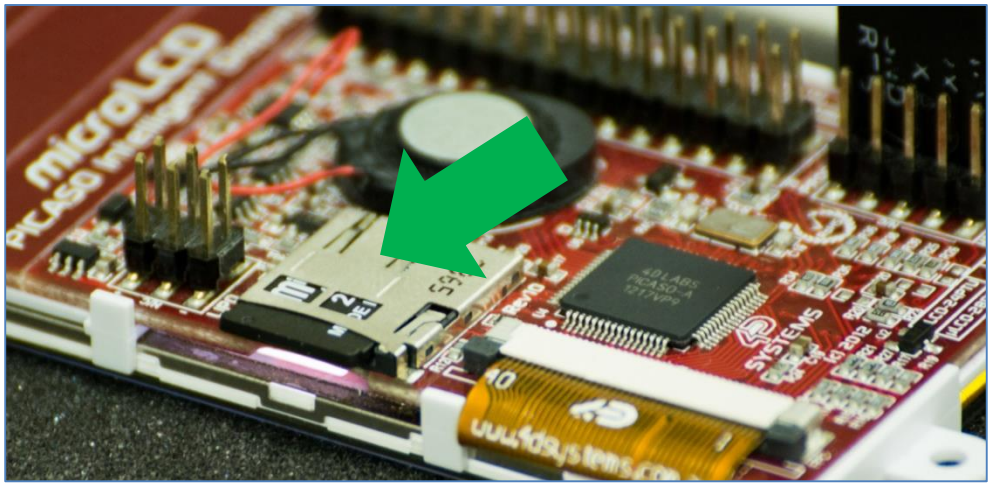
After the files are successfully transferred to the uSD card, the uSD card will now be RAW-formatted. Attempting to open it in the File Explorer will result to:



Click Cancel and proceed to the next step.

Insert the µSD Card to the Display Module

Properly disconnect the µSD card from the PC and plug it to the µSD Card slot of the display module. The next step now is to create a program that will access the GCI file and play the video.



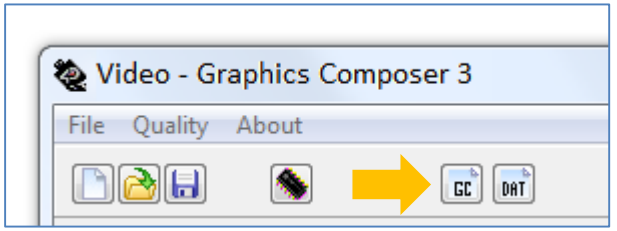
Essential Commands for Video Playback

Playing a video is very easy with 4DGL. The following two commands are the very basics required to select and produce the video to the screen. The sector offsets shown are only an example:

```
media_SetSector(0x001D,0x5001);  //set stream sector
address
media_Video(0,0);
```

Extracting Sector Offsets

The sector offsets are determined by extracting them from the GC or TXT files produced when the video is built. Click on either of the buttons indicated below to view their contents.



Potential Issues

Using these two commands alone can result in unusual behaviour if used without additional code. As such, the following issue should be taken into account when playing a video file:

- No µSD card inserted

Functions to Combat Issues

To deal with the above mentioned issue, the following command can be included:

- file_Mount()

Example Code

The following code snippet takes into account these issues:

```
while(media_Init()==0); //wait if no uSD card detected
media_SetSector(0x00,0x00);  //set stream sector
address
media_Video(0,0);
```

Complete Application Example

Finally, a fully scripted example is shown that includes error messages for any potential issues that may be encountered:

```
#platform "uLCD-24PTU"

/************************************************
*
* Filename: Video.4dg
* Created: 3rd November 2011
* Author: 4D team
* Description: playing a video file
************************************************
*/

#inherit "4DGL_16bitColours.fnc"

func main()

    gfx_Set(SCREEN_MODE,1);
    while (media_Init()==0);  // wait if no SD
card detected
    print("SD Card Detected");
    media_SetSector(0x0000, 0x0000);
    print("Sector Set");
    print("Video Playing...");
    media_Video(66,60);
    print("Video Done Playing");
    repeat
    forever

endfunc
```

Video vs Image

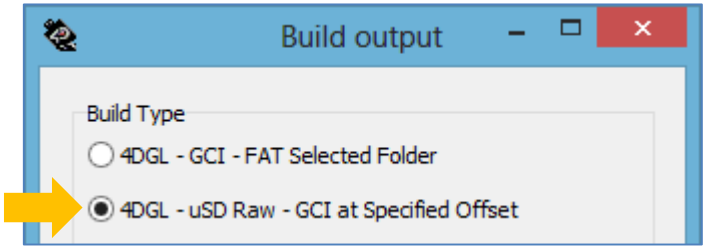
Since videos are essentially a collection of images or frames, this application note is relevant to both image and video objects. For displaying images, the commands are illustrated by the example below.

```
media_SetSector(0x00,0x00);  //set stream sector
address
media_Image(0,0);
```

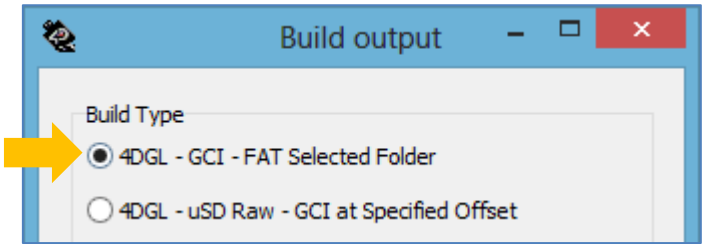
For more information on how images are added to the Graphics Composer window, refer to the application note Designer Displaying Images from the uSD Card - GC FAT16.

RAW vs FAT/FAT16

Note that if the desired uSD card format is RAW, choose the build type "4DGL-uSD Raw…" option when building the output GCI file. When coding in 4DGL, the **media** functions are needed to access the GCI file of a RAW-formatted uSD card.



If the desired uSD card format is FAT or FAT16, choose the build type option "4DGL-GCI-FAT…". When coding in 4DGL, the **file** functions are needed to access the GCI file of a FAT-formatted uSD card. Refer to the internal functions reference manual of your display module's processor for more information on the **media** and **file** commands.



The RAW and FAT options are available to Picaso and Diablo16 display modules. For Goldelox display modules, only the RAW option is possible.

Build and Upload the Project

For instructions on how to save a **Designer** project, how to connect the target display to the PC, how to select the program destination, and how to compile and download a program, please refer to the section "**Run the Program**" of the application note

Designer Getting Started - First Project

Proprietary Information

Disclaimer of Warranties & Limitation of Liability