



ViSi-Genie: Flash Banks

DOCUMENT DATE: **9th MAY 2020**
DOCUMENT REVISION: **1.0**



Description

This application note shows how to utilize the different flash banks of the Diablo16 processor.

Before getting started, the following are required:

Hardware

- Any [4D Systems display module](#) powered by any of the following processors:
 - o Diablo16
- [Programming Adaptor for target display module](#)
- [uSD Card](#)
- USB Card Reader

Software

- [Workshop4](#)

This application note comes with two (2) ViSi-Genie projects.

Note: Using a non-4D programming interface could damage the processor and void the warranty.

Note: This application note is only applicable for Diablo16 display modules.

Content

Description	2
Content	2
Application Overview	3
Setup Procedure	3
Create a New Project	3
Design the Project	3
<i>Load a Program Directly to the Target Flash Bank</i>	<i>3</i>
<i>Load a Program to the uSD Card</i>	<i>5</i>
<i>Generate the Program File for Flash Bank 1</i>	<i>5</i>
<i>The Upload Bank(s) and Run Program</i>	<i>6</i>
<i>Generate the Program File for Flash Banks 2 to 5</i>	<i>7</i>
<i>Modify the Upload Bank(s) and Run the Program</i>	<i>7</i>
<i>Erasing Flash Banks</i>	<i>9</i>
Run the Program	9
Proprietary Information	10
Disclaimer of Warranties & Limitation of Liability	10

Application Overview

The Diablo16 processor has six flash banks (Bank 0 to Bank 5), each of which has a capacity of 32 kB. As of WS4 version 4.5.0.8, it is now possible for the user to specify the destination flash bank of a ViSi-Genie program. This was not possible in previous versions of Workshop4. Prior to version 4.5.0.8, bank 0 was the only possible flash memory destination of a ViSi-Genie program.

The purpose of this application note is to show how this feature can be utilized. This application note uses the ViSi-Genie environment. This application note is applicable to Diablo16 display modules only.

Setup Procedure

For instructions on how to launch Workshop4, how to open a **ViSi Genie** project, and how to change the target display, kindly refer to the section “**Setup Procedure**” of any of the following application notes:

- [ViSi-Genie Getting Started - First Project for Diablo16 Display Modules](#)

Create a New Project

For instructions on how to create a new **ViSi Genie** project, please refer to the section “**Create a New Project**” of any of the following application notes:

- [ViSi-Genie Getting Started - First Project for Diablo16 Display Modules](#)

Design the Project

For this application note, a gen4-uLCD-35DCT-CLB will be used for the project. The same procedure is applicable for any Diablo16 displays. Also, this application note comes with zip files which contain demo projects needed for the discussions.

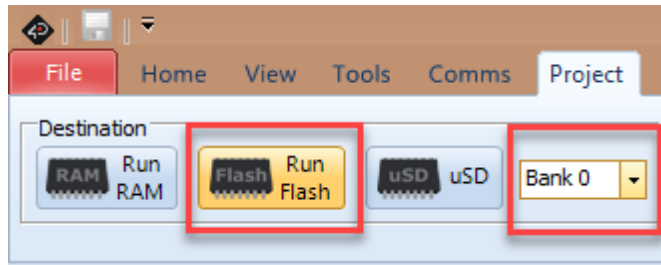
There are two options when utilizing the multiple flash bank feature. The first one is to directly load the program to the target flash bank. The second option is to copy the program file to the uSD card which is directly mounted to the PC. The uSD card can then be unmounted from the PC and mounted to the display module. Another program is then needed to facilitate the transfer of the program on the uSD card to the destination flash bank. These options are discussed in more detail in the following sections.

Load a Program Directly to the Target Flash Bank

Open the ViSi-Genie project inside “FB0.zip” and change the target display if needed.



Under the project menu, the destination should be set to “Run Flash” and the drop-down menu should be set to “Bank 0”, as shown below.

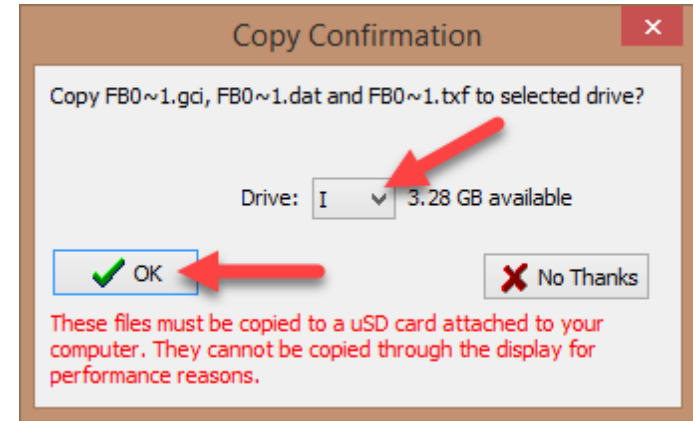


The above settings will cause WS4 to load the program to flash bank 0 of the Diablo16 processor. For this, the display module needs to be connected to the PC thru a programming module. Also, a uSD card needs to be mounted directly to the PC so that Worskhop4 can transfer the graphics files. For more information on how to actually connect a display module to the PC, refer to the section [Run the Program](#).

Under the home menu, click the Build Copy/Load button, as shown below.



The display module now resets. Note that it now runs the program from flash bank 0.



WS4 now copies the graphics files to the uSD card and uploads the program to the flash bank 0 of the display processor.

Now open the ViSi-Genie project inside the attached zip file “FB1.zip”.



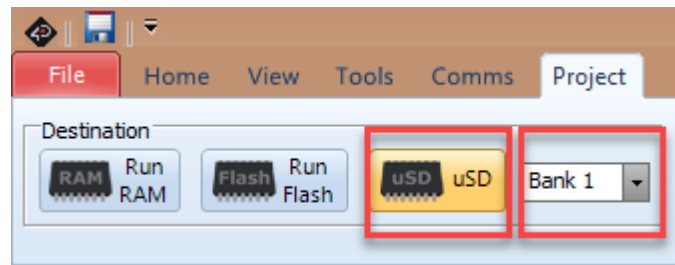
This is because the processor runs the program from flash bank 0 always after a power cycle.

Load a Program to the uSD Card

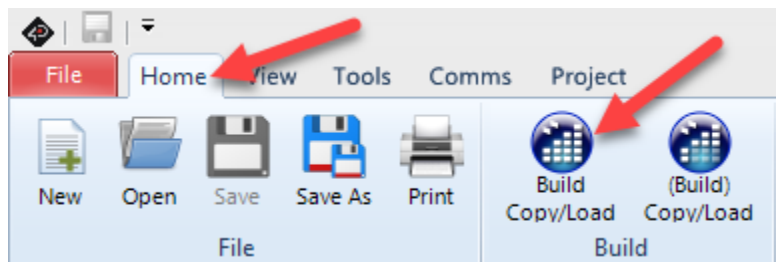
This option does not require the display module to be connected to the PC thru a programming module. However, a uSD card mounted to the PC is still needed.

Generate the Program File for Flash Bank 1

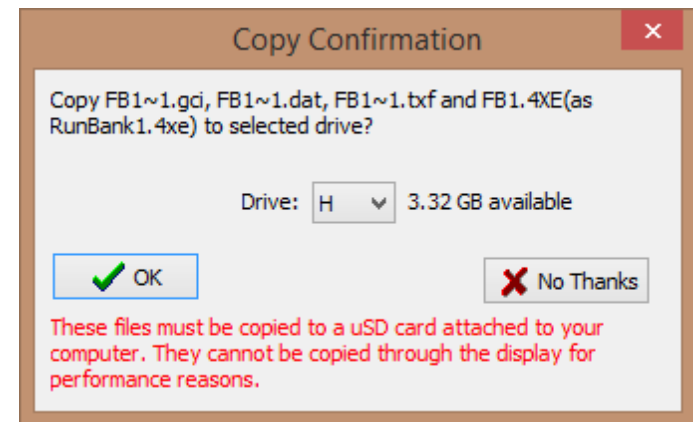
With the ViSi-Genie project “FB1” open in Workshop4, go to the Project menu and configure the settings as shown below.



Under Home menu, click on the Build Copy/Load button.



WS4 now prompts for the uSD card drive.



As indicated in the image, WS4 will copy four files to the uSD card. FB1~1.4XE is the actual program file. This will be copied to the uSD card as “RunBank1.4XE”. The other files are supporting graphics and font files.

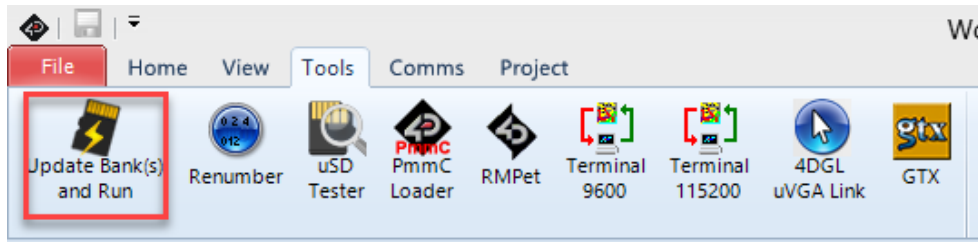
After clicking OK, the uSD card should now contain the following additional files.

Name	Date modified	Type	Size
RunBank1.4xe	8/9/2017 9:36 AM	4XE File	4 KB
FB1~1.txf	8/9/2017 9:36 AM	TXF File	1 KB
FB1~1.gci	8/9/2017 9:36 AM	GCI File	9,793 KB
FB1~1.dat	8/9/2017 9:36 AM	DAT File	1 KB

Now properly unmount the uSD card from the PC and mount it to the display module. Note that the display will still run the program from flash bank 0.

The Upload Bank(s) and Run Program

With the program file **RunBank1.4XE** file already on the uSD card, another program is needed to copy it to the destination bank, which is flash bank 1. Logically, this program should reside in flash bank 0, since this is where control goes to after the display is power cycled. WS4 comes with a utility program for this purpose. This is the Upload Bank(s) and Run program, which can be uploaded to flash bank 0 of the display module by clicking its icon under the Tools menu.



With the display module connected to the PC thru a programming module, click on the Upload Bank(s) and Run icon. This program needs to be loaded to flash bank 0 of the display module only once. After the Upload Bank(s) and Run program is loaded flash bank 0, the display module now shows the message indicated below.

```
Update required: First Run
Bank 1 Update reqd: 2017/08/09 09:36:12 != 0/00/00 00:00:00
Program loaded into Flashbank 1
Flash update successful
```

The first time and date information is that of the program file **RunBank1.4XE** on the uSD card.

```
Update required: First Run
Bank 1 Update reqd: 2017/08/09 09:36:12 != 0/00/00 00:00:00
Program loaded into Flashbank 1
Flash update successful
```

The second time and date information is that of the program file in flash bank 1.

```
Update required: First Run
Bank 1 Update reqd: 2017/08/09 09:36:12 != 0/00/00 00:00:00
Program loaded into Flashbank 1
Flash update successful
```

Note that, during first run, the Upload Bank(s) and Run program assumes that flash bank 1 does not contain any program. Hence, the time and date information contains zero values.

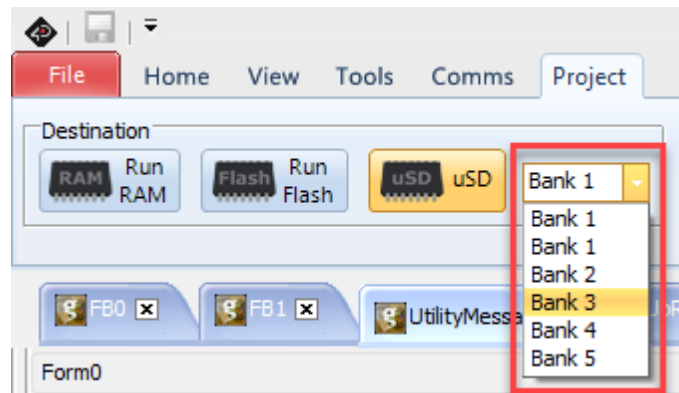
The utility program then compares the time and date information of the program file on the uSD card and that of the program in flash bank 1. If these are not equal, the utility program copies the program file from the uSD card to flash bank 1.

After a successful copy, the utility program then transfers control to flash bank 1.

If the display module is power cycled at this point, the utility program in flash bank 0 will run, and it will detect that the time and date information of the program in flash bank 1 is equal to that of the program file on the uSD card. Hence, the utility program will not copy the program file from the uSD card to flash bank 1.

Generate the Program File for Flash Banks 2 to 5

It is important to note that the Upload Bank(s) and Run program checks the time and date information of all program files for the five flash banks (banks 1 to 5). The program files will be copied to their target flash bank accordingly. To generate the program files for the other banks, follow the procedure described in section [Generate the Program File for Flash Bank 1](#). Set the target bank in the drop-down menu accordingly.



Note that bank 0 is not available in this case, since it is reserved for the Upload Bank(s) and Run utility program.

Modify the Upload Bank(s) and Run the Program

By default, the utility program copies a program file to its destination bank when the time and date information are not equal. This can be changed so that the utility program copies a program file from the uSD card to the destination bank only when the time and date information of the program file on the uSD card is newer as compared to that of the program file in the target flash bank. The logic of the utility program can be changed by modifying its include file, which can be found in the folder:

C:\Users\Public\Documents\4D Labs\4DUpdates\Utils

The filename of the include file is "UpRunBank.inc".

Name	Date modified	Type	Size
UpRunBank.inc	8/8/2017 12:02 PM	INC File	8 KB
TOUCHCALR.INC	3/24/2017 7:25 AM	INC File	14 KB
FlashBoot.INC	3/2/2017 11:05 AM	INC File	1 KB
Gen4-LITE-43PT.inc	1/16/2017 7:19 AM	INC File	2 KB
Gen4-LITE-32PT.inc	1/16/2017 7:19 AM	INC File	2 KB
Gen4-LITE-28PT.inc	1/16/2017 7:19 AM	INC File	2 KB
Gen4-LITE-24PT.inc	1/16/2017 7:18 AM	INC File	2 KB

Before editing, make sure to create a backup copy and save it in a separate folder. The include file can be opened in WS4. Lines 142, 143, and 147 of the include file are shown below.

```

137     pause(400);
138     SystemReset();
139     endif
140     decodeDateTime(); //
141     file_Close(hndl);
142     // if (uSDisNewer(i)) // uncomment either this line
143     if (UsdIsDifferent(i)) // or this line
144     douppdate := 1;
145     print("Bank ", i, " Update reqd: ");
146     printDateTime(ndt);
147     print(" != "); // change to > if using uSDisNewer
148     printDateTime(block+i*ElementSZ);

```

Modify the lines like as shown below.

```

129     to(chkstr);
130     print([STR] ProgName, i, ".4xe");
131     chkstrp := str_Ptr(chkstr);
132     if (file_Exists(chkstrp))
133     k++;
134     hndl := file_Open(chkstrp, 'r');
135     if (!hndl)
136     print("Can't open ", [STR] chkstrp, " Internal error!");
137     pause(400);
138     SystemReset();
139     endif
140     decodeDateTime(); // decod
141     file_Close(hndl);
142     if (uSDisNewer(i)) // uncomment either this line
143     // if (UsdIsDifferent(i)) // or this line
144     douppdate := 1;
145     print("Bank ", i, " Update reqd: ");
146     printDateTime(ndt);
147     print(" > "); // change to != if using UsdIsDifferent
148     printDateTime(block+i*ElementSZ);

```

Save the file after making the changes then upload the Modify the Upload Bank(s) and Run program to the display module again. The utility program

should now copy the program file from the uSD card file to the target flash bank only when the program file on the uSD card has a newer time and date information as compared to that of the program in the target flash bank.

Also, by default, the utility program transfers control to flash bank 1 after the check-and-copy process. The utility program can be made to transfer control to another flash bank besides flash bank 0. For this, line 183 of the include file, shown below, needs to be modified.

```

177     print("Flash update Failed\n");
178     repeat forever
179     endif
180     pause(2000);
181     endif
182     //pause(5000);
183     flash_Run(FLASHBANK_1);
184     endfunc

```

Change this line according to the desired target flash bank. For example:

```

174     if (i)
175     print("Flash update successful\n");
176     else
177     print("Flash update Failed\n");
178     repeat forever
179     endif
180     pause(2000);
181     endif
182     //pause(5000);
183     // flash_Run(FLASHBANK_1);
184     flash_Run(FLASHBANK_2);
185     // flash_Run(FLASHBANK_3);
186     // flash_Run(FLASHBANK_4);
187     // flash_Run(FLASHBANK_5);
188     endfunc

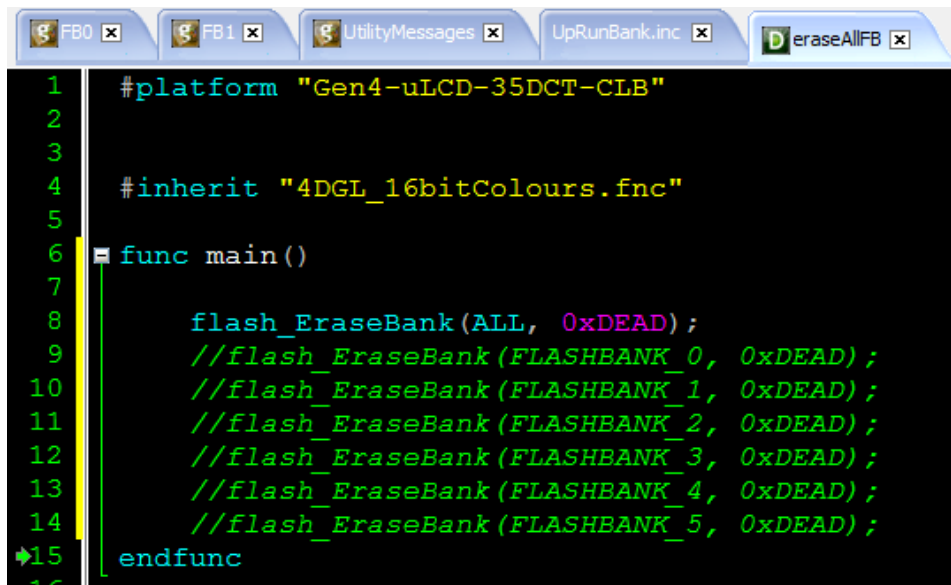
```


Of course, make sure that a program is indeed present in the destination flash bank before passing control to it.

Erasing Flash Banks

To erase the programs inside the flash banks, use the Designer project inside the zip file “**eraseAllFB.zip**”. For more information on how to open a Designer project and upload a Designer program to the display module, refer to the application note Designer Getting Started - First Project.

This Designer project contains the 4DGL commands for erasing the contents of a specific flash bank and those of all flash banks. Modify the source code of the project as desired.



```
1 #platform "Gen4-uLCD-35DCT-CLB"
2
3
4 #inherit "4DGL_16bitColours.fnc"
5
6 func main()
7
8     flash_EraseBank(ALL, 0xDEAD);
9     //flash_EraseBank(FLASHBANK_0, 0xDEAD);
10    //flash_EraseBank(FLASHBANK_1, 0xDEAD);
11    //flash_EraseBank(FLASHBANK_2, 0xDEAD);
12    //flash_EraseBank(FLASHBANK_3, 0xDEAD);
13    //flash_EraseBank(FLASHBANK_4, 0xDEAD);
14    //flash_EraseBank(FLASHBANK_5, 0xDEAD);
15 endfunc
16
```

Use this only when you are sure that you want to clear the flash banks. This is a good way to 'clean up' the entire flash when starting new projects.

Run the Program

For instructions on how to save a **ViSi Genie** 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 any of the following application notes:

- [ViSi-Genie Getting Started - First Project for Diablo16 Display Modules](#)

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 endeavours 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 recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement 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.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application 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.