

Nintendo GameCube™ Optical Disc Drive Guidelines

Version 1.41

Table of Contents

1. Overview	4
2. About the Game Discs	4
2.1 Game Disc structure	4
2.2 Game Disc ID.....	4
3. Guidelines for Error-handling.....	5
3.1 Error-Handling Policy with Optical Disc Drive Device Driver.....	5
3.2 Error Types that Developers Need to Resolve	5
3.2.1 Disc Cover Open Error.....	5
3.2.2 No disk Error	5
3.2.3 Wrong Game Disc Error	6
3.2.4 Retry Error.....	6
3.2.5 Fatal Error.....	6
3.3 Changing Error Messages.....	6
3.4 How to Get the Error Type	7
3.5 State Diagram.....	8
3.6 Error-handling During Audio Streaming.....	8
3.7 How to Emulate Errors	9
4. Games with Multiple Game Discs	10
4.1 Game Disc exchange process patterns.....	10
4.2 Specifying Game Discs at time of exchange	11
4.2.1 Wild Card Usage Restrictions (Non-Game Versions).....	11
4.2.2 Wild Card Usage Restrictions (Game Versions).....	12
4.3 The Game Disc exchange procedure	12
4.4 Messages and Error Processing when Switching Game Discs.....	13
4.5 Multi-disc Emulation.....	13
4.6 APIs for multi-disc games.....	13
4.7 Cautions regarding multi-disc games.....	13
4.7.1 Every Game Disc should boot on its own	13
4.7.2 Provide support for the exchange with any Game Disc.....	13
4.7.3 Display for Game Disc exchanges	13
4.7.4 Use labels and banners to distinguish Game Discs	13
4.7.5 Secure memory for FST	14

5. Other Guidelines Regarding Accessing the Optical Disc Drive..... 14

- 5.1 Game Startup Screen 14
- 5.2 Various Causes of Changes in Game Disc Access Time 14
- 5.3 dol File Restrictions 14
- 5.4 Banner File for Game Disc Information and Displaying it in IPL Main Menu 15
- 5.5 Avoid Infinite Loop of Hardware/Audio Streaming Under the Following Conditions 16
 - 5.5.1 Example of Problem and How to Solve It..... 16

List of Figures

- Figure 1: State Changes 8
- Figure 2: An example of Game Disc information display 15

Example Code

- Code 1: Pseudo-code for Error-handling..... 7

Revision History

Version	Revision Date	Description of Revisions
1.41	3/29/02	- Changed recommended message for Disc Cover Open error.
1.4	2/20/02	- Added sections related to the multi-disc function
1.3	12/3/01	- Added paragraph 3.6.
1.2.2	8/31/01	- Added notes to paragraphs 2.2 and 2.4, - Revised paragraph 2.6
1.2.1	8/3/01	- Revised messages in paragraphs 2.2.4 and 2.2.5 - Added paragraph 2.3 - Changed Game Disc to song at end of paragraph 2.6 - Revised paragraph 3.1 - Added paragraph 3.5
1.01	7/13/01	- Released by NOA

1. Overview

This document offers guidelines when using the Nintendo GameCube™ optical disc drive to design games that are user-friendly with regard to Game Disc access. This document differs from the "Disc Drive (DVD.us.pdf)" document, in that it is meant not only for the programmers, but also for the designers who design the screen layout for error messages and the like, and for the bug testers. For information that is more specific to programming, see this section in the *Nintendo GameCube™ Programmer's Guide*.

2. About the Game Discs

2.1 Game Disc structure

The structure of each Game Disc can be broadly divided into two components:

- The program part (the dol file)
- The user file part

The dol file is the game program itself, converted from the elf file when *loadrun/odrun* is executed. Because the format of the elf file built by the program will differ slightly depending on the linker, the elf file is converted into a custom format called dol for storage on the Game Disc. The dol file is loaded and executed by the IPL.

The user file part is the part that is treated as files by the program part of the Game Disc. It does not matter how these files are used (for data, for re-locatable modules, etc.).

2.2 Game Disc ID

Each Game Disc has a region for storing its unique ID. This unique ID is comprised of the following four elements:

- Game code
Each game has its own specific code. The code is set by the NOA Lot Check Department.
- Company code
Each publisher has their own specific code. This code is also set by the NOA Lot Check Department.
- Game Disc number
Every Game Disc for each game is allocated a Game Disc number. The first Game Disc of a game is numbered "0" and subsequent Game Discs are numbered 1, 2, 3...
- Game version
Every version of the game is allocated a game version. It does not matter whether the game has been released or not.

The Optical Disc Driver uses the Game Disc ID to determine whether the Game Disc is a correct Game Disc. Note:

You may use any numbering scheme to distinguish Game Discs. (For example; Game Disc 1 and Game Disc 2, Game Disc A and Game Disc B, Chapter Blue and Chapter Red, and so on.) The Game Disc to be used when starting the game should be described in the Instruction Booklet, when it is not obvious (such as with Chapter Blue and Chapter Red). Only in the Game Disc Number portion of the Game Disc ID, is the first Game Disc of a set to be numbered "0".

3. Guidelines for Error-handling

3.1 Error-Handling Policy with Optical Disc Drive Device Driver

The error-handling policy of our optical disc drive device driver is “polling”. No complicated error-handling routine is needed. All the game developer needs to do is to poll the “error type” and show the proper message on the screen.

For example, when the Nintendo GameCube™ Disc Cover is open, the optical disc drive device driver will wait for the Disc Cover to be closed, make sure the Nintendo GameCube™ Game Disc is an authorized Game Disc, and then perform the requested command. These processes are carried out automatically. The programmer does not need to create a Disc Cover close wait routine or Game Disc identification routine.

3.2 Error Types that Developers Need to Resolve

There are five “error types” that developers need to resolve.

- Disc Cover Open Error
- No disk Error
- Wrong disk Error
- Retry Error
- Fatal Error

A description of each error type follows.

Note: These errors are generated when the issued command cannot be processed normally for some reason. They will not occur unless a command is issued. For example, even if the Disc Cover is opened when no command has been issued, a “Disc Cover Open error” will not occur. Additionally, the developer does not need to inform the game player about these status changes when they do not generate errors.

3.2.1 Disc Cover Open Error

When the game player opens the Disc Cover of the optical disc drive, this error will occur. Display the following message instructing the user to close the Disc Cover:

“The Disc Cover is open. If you want to continue the game, please close the Disc Cover.”

3.2.2 No disk Error

This error occurs when the Disc Cover is closed, but the optical disc drive cannot find a Nintendo GameCube™ Game Disc. Display the following message and instruct the user to insert the appropriate Game Disc:

“Please insert the <Game Title> Game Disc.”

Display a message like the above, and instruct the user to set a (proper) Game Disc. (Replace “Luigi’s Mansion” with a proper game name according to actual game name. Same applies to the following.)

This error may also occur in the following situations:

- The Game Disc is inserted upside down.
- A game disc that is not a Nintendo GameCube™ Game Disc, like an 8cm CD, is inserted.
- An extremely dirty Game Disc is inserted.

If the game has multiple Game Discs, please display some message like:

“Please insert <Game Title> Game Disc 1”

to clearly indicate which Game Disc must be inserted.

3.2.3 Wrong Game Disc Error

This error occurs when the wrong Game Disc has been placed in the drive. Display the following message and instruct the user to insert the correct Game Disc:

“This is not the <Game Title> Game Disc. Please insert the <Game Title> Game Disc.”

Please be explicit when specifying which Game Disc to insert, when there are several discs in a set. For example:

“Please insert <Game Title> Game Disc 1.”

3.2.4 Retry Error

This error occurs when there is dust, fingerprints, etc. on the Game Disc and it cannot be accessed. Display the following message instructing the user to read the Instruction Booklet. (The Instruction Booklet tells the user to wipe the surface of the Game Disc with a soft cloth and then restart the game.)

“The Game Disc could not be read. Please read the Nintendo GameCube™ Instruction Booklet for more information.”

If the Disc Cover is closed after it has been opened once, the device driver automatically performs a retry.

3.2.5 Fatal Error

This error occurs when a problem is detected that may make it impossible for the optical disc drive to recover. Display the following message instructing the user to read the Instruction Booklet (The Instruction Booklet instructs the user to contact Nintendo Customer Service).

“An error has occurred. Turn the power off and refer to the Nintendo GameCube™ Instruction Booklet for further instructions.”

There may be a problem with the optical disc drive or the Game Disc, so you need to stop the game after displaying an error message.

3.3 Changing Error Messages

The error messages in Section 3.2 “[Error Types that Developers Need to Resolve](#)” are just one example of messages. You may revise these messages, as long as the meaning is accurately conveyed to the user.

However, messages like Retry error and Fatal error reference the Instruction Booklet for procedures to handle their associated errors. In order to avoid user confusion, do not make any major alterations to these messages.

3.4 How to Get the Error Type

This section explains how to get the error type in applications.

The error type shows the device driver's status. The device driver status can be obtained by calling the function `DVDGetDriveStatus`.

Note: As mentioned in Section 3.2 "[Error Types that Developers Need to Resolve](#)", errors only occur when commands cannot be processed properly. So, no errors will occur if no commands have been executed. For example, even if the Disc Cover is opened, the function `DVDGetDriveStatus` will not return `DVD_STATE_COVER_OPEN` (described below) if there is no command being executed. The functions `DVDGetFileInfoStatus` and `DVDGetCommandBlockStatus` behave in the same manner.

The following list indicates states the function will return when an error has occurred.

- `DVD_STATE_COVER_OPEN`
- `DVD_STATE_NO_DISK`
- `DVD_STATE_WRONG_DISK`
- `DVD_STATE_RETRY`
- `DVD_STATE_FATAL_ERROR`

There are other states that can be returned from the function `DVDGetDriveStatus()`. They are not mentioned here because they do not indicate errors, and this document focuses on error handling. Refer to the Nintendo GameCube™ Function Reference Manual for details on other states.

Note: For a game that has multiple Game Discs, `DVD_STATE_MOTOR_STOPPED` is returned when Game Discs are exchanged. See Section 4 "[Games with Multiple Game Discs](#)" for details.

Below is pseudo-code for handling errors. See "errorhandling.c" in Patch 2, dated 5/22/01, or later of the Nintendo GameCube™ Software Development Kit for an operational demo.

```
DVDReadAsync();
do{
    status = DVDGetDriveStatus();
    switch (status){
        case DVD_STATE_COVER_OPEN:
            show_message_for_cover_open_on_screen();
            break;
        case DVD_STATE_NO_DISK:
            show_message_for_no_disk_on_screen();
            break;
        case DVD_STATE_WRONG_DISK:
            show_message_for_wrong_disk_on_screen();
            break;
        case DVD_STATE_RETRY:
            show_message_for_retry_on_screen();
            break;
        case DVD_STATE_FATAL_ERROR:
            show_message_for_fatal_error_on_screen();
            break;
    }
} while (status != DVD_STATE_END) && (status != DVD_STATE_FATAL_ERROR);
```

Code 1: Pseudo-code for Error-handling

In order to display an appropriate error message when a problem occurs, this routine calls the function `DVDGetDriveStatus` every frame and polls the state.

The APIs that access the optical disc drive only return when processing is completed or if a fatal error occurs. "Return" here means that **callback** is called (**callback** is specified with an asynchronous function), or API ends processing and returns a value (with a synchronous function).

You can also use the function `DVDGetFileInfoStatus` or `DVDGetCommandBlockStatus` to get the error state. These return the states of the designated file info and command block. The function `DVDGetDriveStatus` returns the state of the currently executing request.

3.5 State Diagram

The following diagram shows how the state (error type) changes.

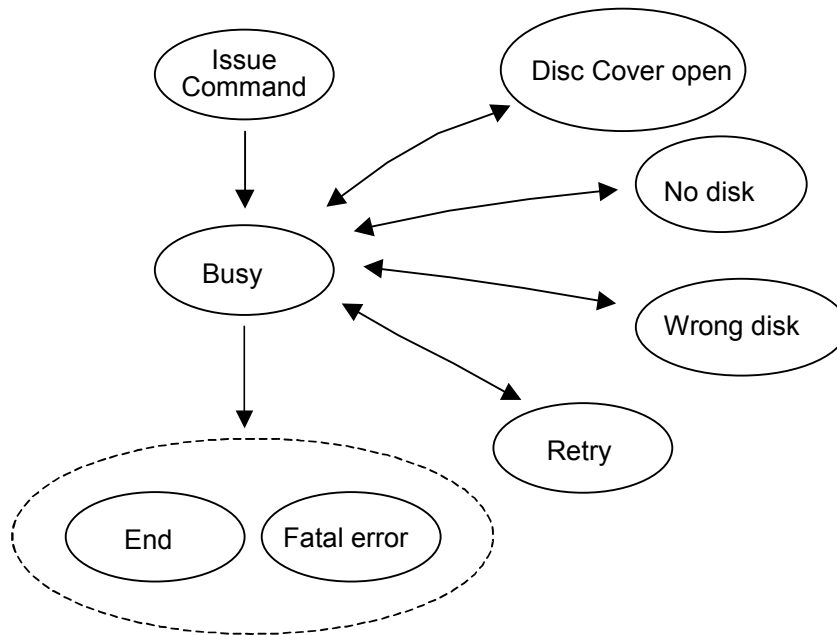


Figure 1: State Changes

An explanation of Figure 1 follows:

* From {Disc Cover Open, No disk, Wrong disk, Retry} to Busy

This state change occurs when the Disc Cover is closed. Note that the state does not change when the Disc Cover is open. For example, in the No disk state, when the Disc Cover is open the state does not change to Disc Cover open (there is no direct path from No disk to Disc Cover open as you can see in the diagram).

* High-speed Change of Status due to Interrupt

For example, if the Disc Cover is closed, the interrupt will take place and be processed internally. The device driver changes states so polling is not required, making the processing minimal.

* Calibration Time when Disc Cover is Closed

Please note that when the Disc Cover is closed, the calibration processing of the optical disc drive takes time. For example, if in Wrong disk state, it takes time to change Game Discs and then a driver checks the Game Disc ID. However, it does not take as long if no Game Disc is inserted.

3.6 Error-handling During Audio Streaming

You do not need to do anything in particular. If the Disc Cover is open when you call the function `DVDPrepareStream`, the function `DVDPrepareStream` waits for the game player to close the Disc Cover like the function `DVDRead` does.

However, as stated in Section 3.4 [“How to Get the Error Type”](#), you must check the state of the device driver using the `DVDGetDriveStatus` function and send a correct message to the game player.

If the function `DVDPrepareStream` is called, and if the game player opens the Disc Cover after audio streaming has begun, audio streaming will be halted. However, even if the game player closes the Disc Cover at this point, be aware that audio streaming will not restart. Additionally, there is no way to know if the Disc Cover is open or not at this point (the function `DVDGetDriveStatus` will not return a Disc Cover open error). In order to detect if the Disc Cover has been opened during audio streaming, the best way is to periodically call a DVD function (i.e.- the function `DVDGetStreamErrorStatus`). The function `DVDGetStreamErrorStatus` waits internally until a normal end or a fatal error occurs, as does the function `DVDRead`. Therefore it is possible to acquire a Disc Cover open or No disk status with the function `DVDGetDriveStatus`.

The DTK library is set up to run `DVDGetStreamErrorStatus` once a second by default.

When you want to resume the track, first check if the function `DVDGetStreamErrorStatus` is in Disc Cover close wait state. If it is, you have to call the function `DVDPrepareStream` again, after the end of the function `DVDGetStreamErrorStatus`. However, you cannot tell at what point the Game Disc was being played when the Disc Cover was opened (although to a certain extent it is possible to tell at what point the Game Disc was being played last by periodically polling the function `DVDGetStreamPlayAddr`). The most a developer can do at this point is to start playing from the beginning of the song.

3.7 How to Emulate Errors

During development, it is very important to make sure the game can handle errors properly (i.e. can show proper message on the TV). You will be able to use the Game Disc emulator (DDH/GDEV) to emulate all kinds of errors, including wrong Game Disc error, retry error and fatal error.

Currently, DDH and NPDP-GDEV have the following emulation capabilities.

Emulation possible on DDH:

We are planning to be able to emulate all errors since AMCDDK Version. 3.0.2 patch #2. Please see the AMCDDK manual for more details.

NPDP-GDEV:

With Version 1.01 or later, all errors can be emulated. Please see the NPDP-GDEV manual for more details.

4. Games with Multiple Game Discs

Please contact the NOA Licensing Department in advance, when considering games with several Game Discs.

At the present time, such discussions are only for two-disc games, for which both discs are sold at the same time. Please hold separate discussions with the Nintendo Licensing Department for games with three or more Game Discs, or for the separate marketing of Game Discs.

If more than one Game Disc is needed for a single game, then the Game Discs must be exchanged at some appropriate place in the game. The exchange is done by specifying the Game Disc ID for the next Game Disc (see Section 2.2 "[Game Disc ID](#)").

The rest of this chapter explains the different types of exchange patterns for multi-disc games and how to specify Game Discs. It provides a broad description of the procedure for exchanging Game Discs, and gives an example of a message displayed to instruct the player to exchange Game Discs.

4.1 Game Disc exchange process patterns

With multi-disc games, the Game Disc exchange process can be broadly classified into two patterns, depending on whether or not the dol program part (Section 2.1 "[Game Disc structure](#)") of the newly inserted Game Disc is loaded and executed after Game Discs are exchanged. As an example, consider the case where Game Disc 1 is removed and replaced with Game Disc 2.

A When loading/running a Game Disc 2 dol file

After the switch from Game Disc 1 to Game Disc 2 is complete, please run the restart process. Restarting will start the dol program on Game Disc 2.

B When not loading/running a Game Disc 2 dol file

After the switch from Game Disc 1 to Game Disc 2 is complete, it will be possible to access the Game Disc 2 user files (Section 2.1 "[Game Disc structure](#)"). There is nothing particular that has to be done after the switch.

In this case, the system can be classified in two ways: to allow Game Disc 2 to start up by itself or not.

B1 When allowing individual start up by Game Disc 2

It is conceivable that the programs on Game Disc 1 and Game Disc 2 could be the same, and only the user file sections would be different. There may be a prompt to return to Game Disc 1, depending upon the state of progress of the game in the Memory Card.

B2 When not allowing individual start up by Game Disc 2

It is conceivable that Game Disc 2 could be a data only file. In this case, please display a message like "This is a data only Game Disc. Please insert the <Game Title> Game Disc 1," and give the user appropriate instructions when in individual start-up mode.

We don't really recommend B2 above. That's because it is possible that there could be Game Disc switching all the time, when a game is being run. Unless there is a particular reason for needing a data-only Game Disc, we feel it is nicer to the user to have the same dol file on the second Game Disc as on the first Game Disc, and make it possible to start-up on the second Game Disc by itself.

Comparing A and B1, B1 requires no loading of the dol program section on Game Disc 2, and the shift to the second Game Disc will be somewhat faster. If the dol program portion of the first and second Game Discs can be made the same, it would probably be better to choose B1.

Note that in either case, we recommend making it possible to save data before switching Game Discs, for the following reasons.

- Data loss can be avoided if there is a Game Disc read error after the switch.
- It will make debugging easier by saving the data to the Memory Card before exchanging the Game Discs. When you have a bug either during or after the Game Disc exchange, you can easily reproduce the bug using the saved data. If you don't save the data before the Game Disc exchange, you will have to begin game play from the Game Disc 1 again, to reproduce the bug.

Caution

File structures that were open prior to the exchange will contain old data, so please do not use such data after the exchange. Even if Game Disc 1 and Game Disc 2 have files with the same name, please be sure to re-open them. To be safe, we recommend closing all open files before switching.

4.2 Specifying Game Discs at time of exchange

The Game Disc ID (see Section 2.2 "[Game Disc ID](#)") is used for specifying Game Discs at the time of an exchange. For details, refer to the DVDChangeDisk* function in the Nintendo GameCube *Dolphin Reference Manual*.

With the DVDChangeDisk* function, wildcards can be specified for Game Disc ID components. Thus, if 0xff is specified for the Game Disc Number, then no matter which numbered Game Disc is inserted during the exchange it will be treated as a correct Game Disc and the exchange will be approved, when the other three Game Disc ID components (i.e., the Game Code, Company Code and Version Number) are correct.

In the following section, we offer some precautionary statements concerning wild cards.

4.2.1 Wild Card Usage Restrictions (Non-Game Versions)

Wild cards can only be used on game versions (not on Game Codes, Company Codes or Game Disc numbers). The reason is that if wild cards are used, the operation check combinations become enormous and cause a lot of problems.

If you are nevertheless considering the use of wild cards for the following elements, be certain to discuss the matter with Nintendo in advance.

4.2.2 Wild Card Usage Restrictions (Game Versions)

When using wild cards for game versions, there are advantages and disadvantages to using and not using them (when given explicit instructions). The following are some examples. Please review them and select one or the other for use.

A When the Game Version is Explicitly Specified

Advantages

- The corresponding Game Disc versions have a one-to-one correspondence, so it is sufficient to debug in that particular combination.

Disadvantages

- When there are future version changes, it will be necessary to upgrade the version on both Game Discs. In other words, if Game Disc 1 specifies Game Disc 2 version 0, and Game Disc 2 was modified to version 1 after it was sold, then once Game Disc 1 had been modified, the versions on both Game Discs would have to be upgraded and resubmitted.
- When a group of friends have their own copies of the same game and the versions are different and two friends switch Game Discs, they will not work properly when Game Disc 1 and Game Disc 2 are exchanged.
- An identifying mark will be required on the label so that the version can be identified.

B When a Wild Card is Used in a Game Version

Advantages

- Independent version upgrades can be released at future dates on their own Game Discs.
- Game Discs with different versions will keep running when they are used together, so friends can switch Game Discs without any problems.

Disadvantages

- Care is required when releasing future version upgrades. In other words, when upgrading for example, Game Disc 2, it will be necessary to assure that it can be switched with all of the Game Disc 1 versions that are out circulating in the market.

4.3 The Game Disc exchange procedure

Below is an outline of the procedure that takes place from the time the player exchanges Game Discs until the Nintendo GameCube can access the new Game Disc.

1. The Optical Disc Drive motor is stopped.
2. The optical disc drive motor is confirmed to be stopped, and a message is displayed telling the game player to exchange Game Discs (see Section 4.4 "[Messages and Error Processing when Switching Game Discs](#)").
3. The system waits until the Disc Cover is opened by the player and then closed.
4. After the exchange, the Game Disc ID is checked to determine whether it is the proper Game Disc.
5. The Game Disc's File Symbol Table (FST) is loaded and access is enabled.

All of these procedures, except for the message display performed in (2), are processed by the `DVDChangeDisk` function (See Section 4.6 "[APIs for multi-disc games](#)").

You can determine the duration of the message displayed in Step 2, by using the `DVDGetDriveStatus` function to check the status of the Device Driver. You check the status of the Device Driver using the same method you use for error handling (see Section 3.4 "[How to get the error type](#)"). When the optical disc drive motor is stopped, the optical disc driver's status is `DVD_STATE_MOTOR_STOPPED`. Only display the message in Step 2 when the optical disc driver is in this state.

If the exchanged Game Disc is not the proper Game Disc, or if there is no Game Disc in the optical disc drive, then display an appropriate message like those shown in Section 4.4 "[Messages and Error Processing when Switching Game Discs](#)". The `DVDChangeDisk*` function can cancel the Game Disc exchange process by calling the `DVDCancel` function.

4.4 Messages and Error Processing when Switching Game Discs

This section provides an example of a message to be displayed to the game player when Game Discs are being exchanged. These sample displays are like the ones in Section 3.3 "[Changing Error Messages](#)," and they may be modified to communicate their intent to the game player.

- Instructing the player to exchange Game Discs

While the Device Driver is in the DVD_STATE_MOTOR_STOPPED state, display this message, instructing the game player to insert the next appropriate Game Disc:

“Press the OPEN Button to open the Disc Cover and replace Game Disc 1 with Game Disc 2. After you have exchanged the Game Discs, close the Disc Cover”

If the game is a multi-disc game, be sure to provide clear instructions about which Game Disc should be inserted next.

When you issue the `DVDChangeDisk` function, perform the same error handling that you would with the `DVDRead` function. For details, see Section 3.2 "[Error Types that Developers Need to Resolve](#)" (If the specified Game Disc is not inserted, then a Wrong Disk error should be generated.)

4.5 Multi-disc Emulation

At the present time, multi-disc emulation can only be executed on the GDEV. Support for DDH is expected to come soon, with the release of the AMCDDK Version 3.0.2, along with patch 2.

4.6 APIs for multi-disc games

The following APIs apply to multi-disc games. For details about each API refer to the Nintendo GameCube *Dolphin Reference Manual*.

- `DVDChangeDisk*` (`DVDChangeDisk` as well as `DVDChangeDiskAsync`)
- `DVDCompareDiskID`
- `DVDGenerateDiskID`
- `DVDGetCurrentDiskID`

4.7 Cautions regarding multi-disc games

4.7.1 Every Game Disc should boot on its own

Please design every Game Disc so it can boot up on its own when inserted in the Nintendo GameCube. Have the Game Disc boot on its own -- even if the game's execution program cannot start up without the execution program that is on some other Game Disc -- and then display a message instructing the game player to exchange Game Discs (see Section 4.7.3 "[Display for Game Disc exchanges](#)")

4.7.2 Provide support for the exchange with any Game Disc

Make sure the system does not lock up, no matter what Game Disc is exchanged. For example, if Game Disc 3 is only supposed to operate when it is booted after Game Disc 2, do not let the system lock up if it is exchanged after Game Disc 1 instead.

4.7.3 Display for Game Disc exchanges

If an exchange of Game Discs is necessary, please display instructions to this effect on the screen. See Section 4.4 "Messages and Error Processing when Switching Game Discs" to read about the contents of this screen message.

4.7.4 Use labels and banners to distinguish Game Discs

Print a label for each Game Disc, so game players can differentiate Game Discs and exchange them properly when instructed to do so. Also, please use different banner image data, titles or comments on each Game Disc to make it possible to distinguish the Game Discs on the IPL game play menu, so that the game players will be able to tell which number Game Disc they are using. (See Section 5.4 "[Banner File for Game Disc Information and Displaying it in IPL Main Menu](#)".)

4.7.5 Secure memory for FST

When a normal Game Disc is exchanged, the File Symbol Table (FST) is overwritten by what is on the new Game Disc. If the FST of the new Game Disc is smaller than the FST of the Game Disc that was swapped out, then there is no problem. But if the new FST is larger, then it cannot be loaded. An ASSERT message will be output, and the system will lock up.

To prevent this from happening, either specify a large memory size when securing memory for the FST, or use the `MULTIDISK` function, in the latest version of GDEV, to store the settings to a mdf file. (When multiple Game Discs are registered with mdf, the memory for the FST will be set automatically to the appropriate size.)

5. Other Guidelines Regarding Accessing the Optical Disc Drive

5.1 Game Startup Screen

After the dol file is loaded by the boot program, data and program files are read from the Game Disc. At this time, in order to shorten the time the screen is blank as much as possible, do not read any data files. Use only the dol file to display the first screen.

Also, the dol file should use the error-handling routines described in Section 3.4 "[How to Get the Error Type](#)". If there are no error-handling routines, there is no way to notify the user if some kind of problem occurs with the read before the error-handling routine.

5.2 Various Causes of Changes in Game Disc Access Time

If a problem occurs with Game Disc access, the hardware will carry out a retry internally. This means that even if a read has succeeded, it may have been the result of going through several retries.

Also, the calibration process requires a large amount of time, as mentioned in Section 3.5 "[State Diagram](#)". Therefore, when a game player opens the Disc Cover while reading, much longer time is required than just one read.

Furthermore, due to individual differences, the access may be completed slightly faster than with the particular optical disc drive the developer is using.

Because Game Disc access time can vary for all of these various reasons, do not design processes that depend on access time.

5.3 dol File Restrictions

Be aware that following two restrictions apply to the dol file (see Section 2.1 "[Game Disc structure](#)").

- The total section size must be less than 4 Mbytes.
In order to shorten the read time during startup, the total size excluding the BSS section (data area for uninitialized variables) is limited to less than 4 Mbytes.
- Addresses that can be used by dol files are below 0x8070_0000
Addresses from 0x8070_0000 and on are reserved by the system so dol files can only use addresses below this. Note that this restriction also applies to BSS section.

5.4 Banner File for Game Disc Information and Displaying it in IPL Main Menu

On the Nintendo GameCube™, when the game is not started immediately and is started after going into the IPL Main Menu once, disc information is displayed on the screen. Banner files are used in this situation.

In order to display disc information, the following information must be specified in the banner file:

```
u8 image[2 * DVD_BANNER_WIDTH * DVD_BANNER_HEIGHT]
(DVD_BANNER_WIDTH=96; DVD_BANNER_HEIGHT=32)
    Banner-format image data.
    The size is 96x32 (the same as the memory card banner), but the only format that can be used is RGB5A3.
    Identical data is used on the menu screen and on the gameplay screen.
    There is no problem if the data does not match the memory card banner (although you can use the same data).
```

```
u8 shortTitle[32]
    The title of the game displayed on the Main Menu screen.
```

```
u8 shortMaker[32]
    The name of the maker displayed on the Main Menu screen.
    You can include just the maker name, or the maker name and the production date (the year).
```

```
u8 longTitle[64]
    The title of the game displayed during gameplay.
```

```
u8 longMaker[64]
    The name of the maker displayed on the gameplay screen.
    You can include just the maker name, or the maker name and the production date (the year).
```

```
u8 comment[128]
    Comments displayed on the game play screen.
```

Example: Luigi's Mansion

```
shortTitle    Luigi's Mansion
shortMaker    2001 Nintendo
longTitle     Luigi's Mansion: Luigi's Big Adventure
longMaker     2001 Nintendo
comment       The house Luigi won in a sweepstakes is a haunted mansion!
              To find his brother Mario, he must explore the big mansion!
```

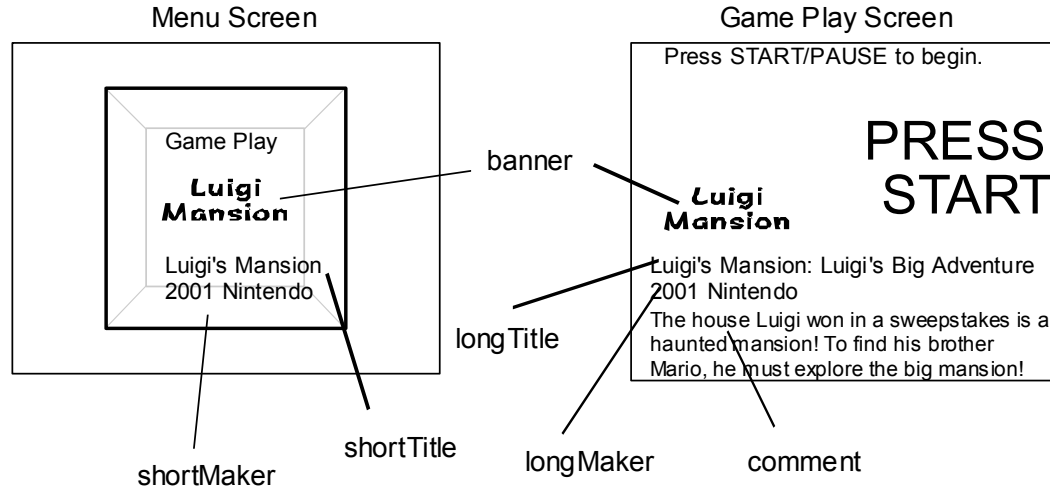


Figure 2: An example of Game Disc information display

The characters that can be used are the same as those that can be used by the FONT API. Game Discs intended for the Japanese market can accommodate Level 1 Shift-JIS characters as well as ASCII and single-byte ("hankaku") katakana characters. Discs intended for the non-Japanese market can accommodate ANSI 8-bit (WinLatin1) characters. Line feed codes can also be used in comments. However, if you use many line breaks and characters with wide character widths, there is a possibility that the comment will not be displayed completely within the frame. In such cases, you would need to revise your text to make it fit completely within the frame.

On development tools, the Nintendo GameCube™ startup program displays the information in the `opening.bnr` file stored in the Game Disc's root directory. You can create `opening.bnr` by using the `makebanner.exe` tool included with the Nintendo GameCube™ SDK in the `/X86/bin` directory. To use this tool, execute `makebanner.exe`. The structure of banner files is defined in `include/dolphin/dvd/DVDBanner.h`.

You can use the `listdemo` program (located in `build/demos/carddemo`) to check the Game Disc banner and game information located in the `opening.bnr` file.

5.5 Avoid Infinite Loop of Hardware/Audio Streaming Under the Following Conditions

Avoid an infinite loop in the following situations:

1. Hardware/Audio Streaming while Data Streaming
2. Synchronous Playback of more than one data stream

Note: The necessary data is read from the Game Disc all at once and then used as required. This method is defined as Burst Transfer. The data is read when required, one by one in order, and is not read all at once. This method is defined as Data Streaming. A typical example of data streaming transfer would be the playback of a movie.

The reason that we need to avoid the infinite loop is because the player may leave the Nintendo GameCube with such condition. Nintendo GameCube's optical disc drive moves the head with high frequency to access the Game Disc. There is a possibility that this may cause bad effects on the optical disc drive. Also, it is meaningless to have optical disc drive access without the player present, so please avoid this situation.

Other than "1" and "2" above, if there is a need for prolonged access that is not sequential with high frequency, either make the access sequential or prevent it from entering an infinite loop.

You can reduce the seek noise that is generated during synchronous playback considerably, by placing those files that are played back synchronously close to each other. Place these files close to each other, as much as possible.

5.5.1 Example of Problem and How to Solve It

[1] There is a demo that plays different audio files as BGM while playing back a movie file. It is possible that this demo will enter an infinite loop.

Solution 1: Compile the movie file and the audio file into one file by interleaving them, and playback the audio portion using software. After this is done, access to the optical disc drive will be sequential.

Solution 2: Stop BGM after the fade-out of the third loop. By doing this, only the movie is played back, and access to the optical disc drive will be sequential.

Solution 3: Switch the movie to a static image after three loops. By doing this, it will be only hardware/audio streaming, and the access to the optical disc drive will be sequential.

[2] If two movies are played back synchronously there is a possibility that the program will enter an infinite loop.

Solution 1: Compile the two movie files into one file by interleaving them. By doing this, access to the optical disc drive will be sequential.

Solution 2: Switch both movies to a static image after three loops. This will prevent an infinite loop.

[3] There is an artificial intelligence type dialog game. There is a portion of this game in which voice data is read consecutively, while the audio file is played as BGM. This might continue infinitely, unless there is an input to the Controller.

Solution: If there is no input to the Controller for 5 minutes, make characters in the game go to sleep, so that they will no longer talk. By doing this, access to the optical disc drive will be only for playing back BGM.