
Revolution SDK

Wii Optical Disc Drive Guidelines

Version 2.1

The contents in this document are highly
confidential and should be handled accordingly.

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo of America Inc. and/or Nintendo Company Ltd. and are protected by Federal copyright law. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

Contents

1	Overview.....	7
2	About the Game Discs.....	8
2.1	Game Disc Structure	8
2.2	Game Disc ID	8
3	Guidelines for Error-handling.....	9
3.1	Error-Handling Policy for the Optical Disc Drive Device Driver	9
3.2	Error Types that Developers Need to Resolve	9
3.2.1	No Disc Error	9
3.2.2	Wrong Disc Error.....	9
3.2.3	Retry Error	10
3.2.4	Fatal Error	10
3.3	Changing Error Messages	10
3.4	Getting Error Types	10
3.5	State Diagram.....	12
3.6	How to Emulate Errors.....	13
4	Games with Multiple Game Discs (Reference).....	14
4.1	Game Disc Exchange Process Patterns.....	14
4.2	Specifying Game Discs at Time of Exchange.....	15
4.2.1	Wild Card Usage Restrictions (Non-Game Versions)	15
4.2.2	Wild Card Usage Restrictions (Game Versions)	16
4.3	The Game Disc Exchange Procedure	17
4.4	Messages and Error Processing when Switching Game Discs	17
4.5	Multi-disc Emulation.....	18
4.6	APIs for Multi-disc Games	18
4.7	Cautions Regarding Multi-disc Games	19
4.7.1	Every Game Disc Should Boot on its Own.....	19
4.7.2	Provide Support for the Exchange with any Game Disc	19
4.7.3	Display for Game Disc Exchanges.....	19
4.7.4	Use Labels and Banners to Distinguish Game Discs.....	19
4.7.5	Secure Memory for the FST.....	19
5	Other Guidelines Regarding Accessing the Optical Disc Drive	20
5.1	Game Startup Screen	20
5.2	Various Causes of Changes in Game Disc Access Time	20
5.3	dol File Restrictions	20
5.4	Prohibition of Non-sequential Access for Extended Periods of Time	21
5.4.1	Examples of Problems and How to Solve Them.....	21
6	Other Cautions	22
6.1	Describing Wii Optical Media.....	22

Code Examples

Code 7-1	Pseudo-code for Error-handling	11
----------	--------------------------------------	----

Figures

Figure 7-1	State Changes	12
------------	---------------------	----

Tables

Table 7-1	Error Emulation Method for Each Development Device	13
-----------	--	----

Revision History

Version	Revision Date	Description of Revisions
2.1	2007/10/19	<ul style="list-style-type: none"> Changed the name of documents referenced in Chapter 1 Overview. Updated section 2.1 Game Disc Structure to use the Wii Menu, rather than the IPL, to load and execute dol files. Changed the error processing example in section 3.1 Error-Handling Policy for the Optical Disc Drive Device Driver to use the insertion of a Game Disc instead of the opening of the Disc Cover. Deleted the “Disc Cover Open Error” item and updated the numbers of the following items. Added the use of the controller and the power button to the rest button for the items to be disabled when a data error occurs, as indicated in section 3.2.4 Fatal Error. Deleted “Error-handling During Audio Streaming”. Updated the numbers of the following sections. In section 5.1 Game Startup Screen, replaced the first screen to display with the wrist strap usage screen. Added information regarding the display of the dol file alone given the size restrictions. It was recommended in section 5.3 dol File Restrictions that the size of dol files should be kept below four megabytes. However, since this proved difficult the section content was changed to indicate that the dol file size should be kept as small as possible. Deleted “Banner File for Game Disc Information and Displaying it in IPL Main Menu”. Updated the numbers of the following sections.
2.0	2007/06/28	<ul style="list-style-type: none"> Deleted all remaining areas with Nintendo GameCube content. Deleted the message examples in section 3.2.2 and changed them to reference DISC_01 and DISC_05 from the Wii Message List. Deleted the message examples in section 3.2.3 and changed them to reference DISC_02 and DISC_07 from the Wii Message List. Deleted the message example in section 3.2.4 and changed it to reference DISC_03 from the Wii Message List. Deleted the message example in section 3.2.5 and changed it to reference DISC_04 from the Wii Message List. Added a reference to the Wii Programming Guideline: “Reset and Shutdown Processing for Fatal Errors [Required]”. In section 3.3, applied the prohibition on changes to messages that reference the Wii Operations Manual. Added supplementary information for Japanese titles stating that kanji notation can be replaced by hiragana, taking into account the user’s age. Deleted the message example in section 4.1 and changed it to reference DISC_08 from the Wii Message List.

Version	Revision Date	Description of Revisions
2.0	2007/06/28	<ul style="list-style-type: none"> Deleted the message example in section 4.4 and changed it to reference DISC_05 from the Wii Message List. Deleted content in section 4.7.1 and indicated that Wii Programming Guidelines, [Multiple Disc Games Only] Independent Startup of All Discs [Required], should be referenced. Changed section 4.7.4 to reference the Icons and Banners chapter of the Wii Programming Guidelines. Deleted content in section 5.2 and indicated that Wii Programming Guidelines, Prohibition of Dependency on Drive Specifications [Required], should be referenced. Changed the title of section 5.5 from "Avoid Infinite Loop of Hardware/ Audio Streaming Under the Following Conditions" to "Prohibition of Non-sequential Access for Extended Periods of Time". Deleted content and indicated that Wii Programming Guidelines, Prohibition of Non-sequential Access for Extended Periods of Time [Required], should be referenced.
1.46 + Revisions for the Wii	2006/09/29	<ul style="list-style-type: none"> Added revisions for the Wii to provide tentative information. Plan to move to the Optical Disc Drive Library Programming Manual.
1.46	2004/05/28	<ul style="list-style-type: none"> Added Chapter 6 Other Cautions and section 6.1 Regarding the Description of Nintendo GameCube Optical Media.
	2004/04/21	<ul style="list-style-type: none"> Added a table to section 3.7 of error emulation methods for each development instrument.
1.45	2003/11/28	<ul style="list-style-type: none"> Made the restriction in section 5.3 a recommendation, that the total section size should be less than 4 megabytes.
1.44	2003/02/05	<ul style="list-style-type: none"> Changed the instructions in section 4.7.4 for distinguishing discs.
	2003/01/09	<ul style="list-style-type: none"> Added a warning to section 5.1 about problems with the boot program.
1.43	2002/11/08	<ul style="list-style-type: none"> Deleted parts of section 3.2.5 to reflect updates to the console user's manual. Added a recommendation to disable RESET when a Fatal error has occurred in the disc drive.
	2002/10/09	<ul style="list-style-type: none"> Divided the explanation in section 3.6 into three subsections (3.6.1 - 3.6.2).
	2002/10/01	<ul style="list-style-type: none"> Added error emulation methods using NPDP to section 3.7. Revised section 4.7.5 to explain memory allocation methods for FST in more detail. Added a note to section 5.4 when creating banners.

Version	Revision Date	Description of Revisions
1.42	2002/08/09	<ul style="list-style-type: none">Replaced disc number with game version as an example of a wild card in section 4.2.
	2002/08/06	<ul style="list-style-type: none">Revised section 3.7 to reflect added functionality in the optical disc emulator.Changed the technical support notation in Chapter 4.Revised section 3.7 to reflect updates to the AMCDDK.Revised section 4.5 to reflect updates to the AMCDDK.
	2002/07/30	<ul style="list-style-type: none">Revised the recommended message for WRONG_DISK_ERROR in section 3.2.3Revised the recommended message for switching Game Discs in section 4.4
	2002/07/25	<ul style="list-style-type: none">Added an explanation to the note in section 3.2.Added a note to section 3.6.

1 Overview

This document offers guidelines when using the Wii optical disc drive to design games that are user-friendly with regard to Game Disc access. This document differs from the *Revolution Optical Disc Drive Library Programming Manual* (DVD.pdf) 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 the *Revolution Optical Disc Drive Library Programming Manual*.

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 `ndrun` 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 `.dol` file for storage on the Game Disc. The `.dol` file is loaded and executed by the Wii Menu.

The user file part is the component 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 relocatable modules, and so on).

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 obtained through the NOA Lot Check Department for the US market and NOE Lot Check Department for the European market.
- 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.

3 Guidelines for Error-handling

3.1 Error-Handling Policy for the Optical Disc Drive Device Driver

The Nintendo optical disc drive device driver uses polling for error-handling so no complicated error-handling routine is needed. All a game developer needs to do is to poll the error type and display the proper message on the screen.

For example, the optical disc drive device driver confirms that a disc is valid when it is inserted and executes the requested command. Since this processing is automatic, game developers do not need to create a disc identification routine.

3.2 Error Types that Developers Need to Resolve

There are four error types that developers need to resolve.

- No Disc Error
- Wrong Disc Error
- Retry Error
- Fatal Error

A description of each error type follows.

Note: These errors are generated when the issued command is not processed normally for some reason. They will not occur if no command has been issued. For example, when no command has been issued, a No Disc error will not occur, even if a disc is ejected. Additionally, the developer does not need to inform the game player about status changes when they do not generate errors. In other words, even if the game player ejects the disc during gameplay, you do not need to display this message until an attempt to access the Game Disc is made.

3.2.1 No Disc Error

This error occurs when the optical disc drive cannot find a Wii Disc. Display a message similar to `DISC_01` in the Wii Message List and instruct the user to insert the appropriate disc.

This error may also occur in the following situations:

- The Wii Disc is inserted upside down.
- A disc that is not a Wii Disc (for example, a 12cm CD) is inserted.
- An extremely dirty Wii Disc is inserted.

If the game has multiple Game Discs, please display a message similar to `DISC_06` in the Wii Message List to clearly indicate which Game Disc must be inserted.

3.2.2 Wrong Disc Error

This error occurs when the wrong Game Disc has been placed in the drive. Display a message similar to `DISC_02` from the Wii Message List and instruct the user to insert the correct Game Disc:

Please be explicit when specifying which Game Disc to insert, similar to `DISC_07` from the Wii Message List, when there are several discs in a set.

3.2.3 Retry Error

This error occurs when there is dust, fingerprints, etc., on the Game Disc, and it cannot be accessed. Display the `DISC_03` message from the Wii Message List, 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.) If a disc is inserted after a disc has been ejected, the device driver automatically performs a retry.

3.2.4 Fatal Error

This error occurs when a problem is detected that may make it impossible for the optical disc drive to recover. Display the `DISC_04` message from the Wii Message List. 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.

Note: When there is a fatal error, disable controller processing, and the RESET and POWER buttons during the presentation of the error message. (Give priority to the display of fatal error message). Refer to “Reset and Shutdown Processing for Fatal Errors [Required]” in the *Wii Programming Guidelines*.

3.3 Changing Error Messages

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

However, messages like those for the retry and fatal errors reference the Wii operations manual for procedures to handle their associated errors. To avoid user confusion, alterations to these messages are prohibited.

For Japanese versions, you may replace error message expressions in kanji with hiragana, taking into account the age of the user.

3.4 Getting Error Types

This section explains how an application can get the optical disc error type.

The error type shows the device driver’s status. Get the device driver status by calling the `DVDGetDriveStatus` function.

Note: As mentioned in section [3.2 Error Types that Developers Need to Resolve](#), errors only occur when commands cannot be processed properly. No errors will therefore occur if no commands have been run. For example, even when the disc is ejected, the function `DVDGetDriveStatus` will not return `DVD_STATE_NO_DISK` (described below) if no command is being run. The functions `DVDGetFileInfoStatus` and `DVDGetCommandBlockStatus` behave in the same manner.

The following list indicates states returned when an error occurs.

- `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 `DVDGetDriveStatus()` function. They are not mentioned here because they do not indicate errors, and this document focuses on error handling. Refer to the *Revolution SDK 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 Chapter [4 Games with Multiple Game Discs \(Reference\)](#), for details.

Below is pseudo-code for handling errors. For an operational demo, see the DVD sample, `errorhandling`, in the Revolution SDK.

Code 7–1 Pseudo-code for Error-handling

```
DVDReadAsync();
do{
    status = DVDGetDriveStatus();
    switch (status){
        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);
```

To display an appropriate error message when a problem occurs, this routine calls the `DVDGetDriveStatus` function at each frame and polls the state.

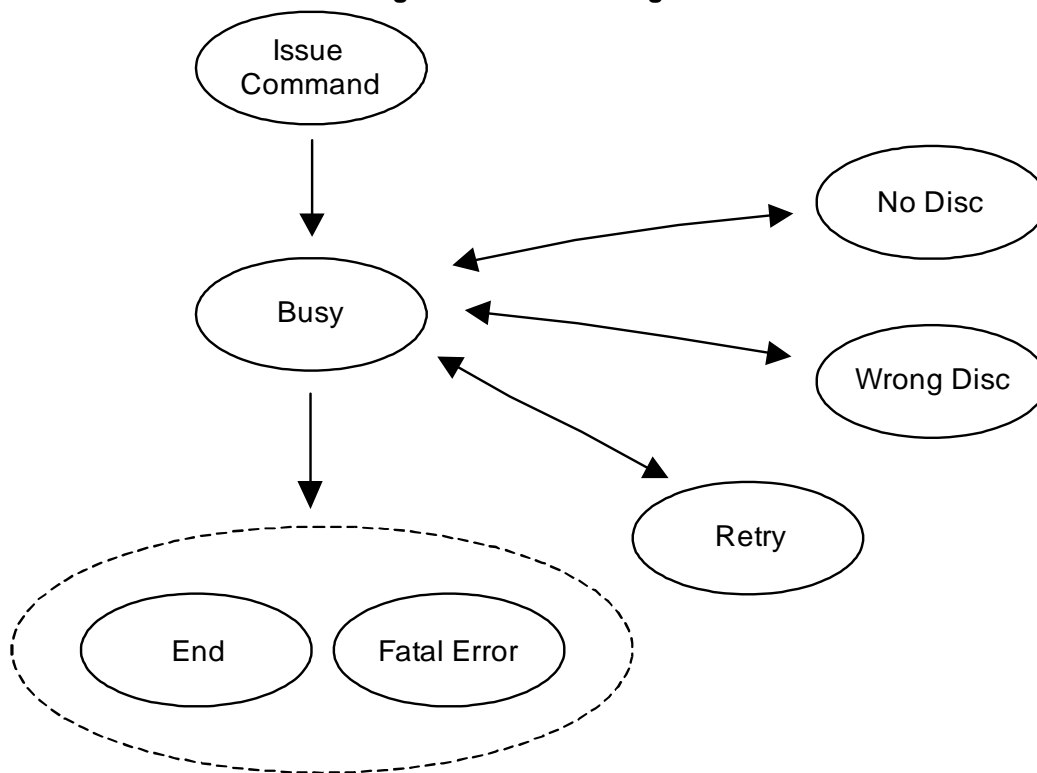
The APIs that access the optical disc drive only return when processing is completed or if a fatal error occurs. “Return” here means that **callback** is called (**callbacks** are specified with asynchronous functions), or that the API ends processing and returns a value (with a synchronous function).

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

3.5 State Diagram

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

Figure 7-1 State Changes



An explanation of [Figure 7-1](#) follows:

- From {No Disc, Wrong Disc, Retry} to Busy

This state change occurs when the disc is inserted. Note that the state does not change when the disc is ejected.

- Calibration Time when disc is inserted

When the disc is inserted, the calibration processing of the optical disc drive takes time. For example, if in the `WRONG_DISK` state, it takes time to change discs and then a driver checks the Disc ID.

3.6 How to Emulate Errors

Developers must check that appropriate error-handling routines (for example, displaying appropriate error messages on the TV) are executed in their program. All of the different kinds of errors (Wrong Disc, Retry, and Fatal) can be emulated using NDEV.

NDEV:

All errors can be emulated. For details see the NDEV manual.

Table 7–1 Error Emulation Method for Each Development Device

	Retry Error	Fatal Error (Note 1)	Wrong Disk Error	No Disk Error
NDEV	ODEM (RETRY button)	ODEM (FATAL button)	ODEM (Select DLF File) ODEM (Eject) ODEM (Load)	ODEM (No Disk) ODEM (Eject) ODEM (Load)
RVT-R Reader	None	None	Exchange disk (Note 2)	Remove disk

Note 1: Error appears when you hold down the button until the next disc access.

Note 2: Use different disc IDs for the disc after exchange and the disc before exchange.

4 Games with Multiple Game Discs (Reference)

Note: Please contact the NOA Licensing Department in advance when considering games with multiple Game Discs.

Note: 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 marketing of separate Game Discs.

If more than one Game Disc is needed for a single game, then the Game Discs must be exchanged at an 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 (see 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 (see section [2.1 Game Disc Structure](#)). There is nothing particular that has to be done after the switch.

In this case, there are two different patterns: allowing 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 Wii console NAND memory.

- **B2 When not allowing individual start up by Game Disc 2**

It is conceivable that Game Disc 2 could contain only data files. In this case, please display a message similar to `DISC_08` from the Wii Message List and give the user appropriate instructions when in individual start-up mode.

We don't really recommend B2 above. That pattern could result in 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 kinder 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 the A and B1 patterns, 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.

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 Game Disc transitions easier (case A only: if the reading and passing of data from Game Disc 1 to Game Disc 2 is restricted to only pass through Wii console NAND memory, if a bug occurs during the Game Disc switch, Game Disc 2 can be started independently and debugging performed).

Note: 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 *Revolution SDK Function Reference Manual*.

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

In the following section, we offer some cautions 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 when wild cards are used, the operation check combinations become enormous and cause a lot of problems.

If you are still considering the use of wild cards for the elements other than game version, you must 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 (with explicit instructions). The following are some examples. Please review them and select one.

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 then 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. 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 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 Wii 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 first Game Disc is ejected and then the second Game Disc is inserted.
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 step 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 Getting Error Types](#)). 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 a message similar to `DISC_05` from the Wii Message List, instructing the game player to insert the next appropriate Game Disc. 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

Currently, multi-disc emulation is not supported.

4.6 APIs for Multi-disc Games

The following APIs apply to multi-disc games. For details about each API refer to the *Revolution SDK Function 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

(Refer to the *Wii Programming Guidelines*, [Multiple Disc Games Only] Independent Startup of All Discs [Required])

4.7.2 Provide Support for the Exchange with any Game Disc

Make sure the system does not hang, no matter which 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 hang if it is exchanged for Game Disc 1 instead.

4.7.3 Display for Game Disc Exchanges

If an exchange of 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 learn more 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. For details on banners and icons displayed by the Wii console, see the *Wii Programming Guidelines*, Icons and Banners chapter.

4.7.5 Secure Memory for the 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. If the new FST is larger, however, it cannot be loaded. An ASSERT message will be output, and the system will hang.

To prevent this from happening with a game that has multiple discs, be sure to secure a memory space of adequate size for the FST.

For further details, read the various development environment manuals.

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, to shorten the time that the screen is blank as much as possible, do not read any data files. Use only the `dol` file to display the wrist strap usage screen. However, if content restrictions make it difficult to display the wrist strap usage screen using the `dol` file alone, it is acceptable to not do so, provided that the screen is displayed as quickly as possible.

Also, the `dol` file should use the error-handling routines described in section [3.4 Getting Error Types](#). If there are no error-handling routines (including necessary data) built into the `dol` file, there is no way to notify the user if some kind of problem occurs with the read before the error-handling routine's data is loaded.

For details on the wrist strap usage screen, refer to the *Wii Programming Guidelines*, Explanatory Message for Wii Remote Strap Use [Required].

5.2 Various Causes of Changes in Game Disc Access Time

Refer to the *Wii Programming Guidelines*, Prohibition on Drive Specification Dependency [Required].

5.3 dol File Restrictions

Be aware that the following restriction applies to the `dol` file (see section [2.1 Game Disc Structure](#)).

- 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.

In addition, we recommend that the total file size, excluding the `BSS` and `SBSS` sections (the data region for uninitialized variables), be made as small as possible. This is recommended to reduce the read time when starting up.

5.4 Prohibition of Non-sequential Access for Extended Periods of Time

Refer to the *Wii Programming Guidelines*, Prohibition of Sustained Continuous Non-Sequential Access [Required].

5.4.1 Examples of Problems and How to Solve Them

- There is a demo that plays different audio files as background music while playing back a movie file. This demo may enter an infinite loop.
Solution 1: Compile the movie file and the audio file into one file by interleaving them, and play back the audio portion in software. After this is done, access to the optical disc drive will be sequential.
Solution 2: Stop the background music after the fade-out of the third loop. By doing so, only the movie is played back and access to the optical disc drive will be sequential.
- If two movies are played back synchronously, the program may 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.
- 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 background music. This might continue infinitely unless there is an input to the controller.
Solution: If there is no input to the controller for five 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 background music.

6 Other Cautions

6.1 Describing Wii Optical Media

When describing Wii optical media to a game player, such as in a game message or in the Instruction Booklet, use “Game Disc”, “Disc”, or “disc” instead of “Disk”.

Dolby, Pro Logic and the Double-D symbol are trademarks of Dolby Laboratories.

IBM is a trademark of International Business Machines Corporation.

Roland GS Sound Set is a trademark of Roland Corporation U.S.

All other trademarks and copyrights are property of their respective owners.

© 2007 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed or loaned in whole or in part without the prior approval of Nintendo.