



User's Guide

SanDisk SDK

SD Audio SDDK-07

Security Manager SDDK-06

SD Device Driver and File System SDDK-05

Revision 1.0
October 2003

SanDisk® Corporation general policy does not recommend the use of its products in life support applications where in a failure or malfunction of the product may directly threaten life or injury. Per SanDisk Terms and Conditions of Sale, the user of SanDisk products in life support applications assumes all risk of such use and indemnifies SanDisk against all damages. See "Disclaimer of Liability."

This document is for information use only and is subject to change without prior notice. SanDisk Corporation assumes no responsibility for any errors that may appear in this document, nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. No part of this document may be reproduced, transmitted, transcribed, stored in a retrievable manner or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written consent of an officer of SanDisk Corporation.

All parts of the SanDisk documentation are protected by copyright law and all rights are reserved.

SanDisk and the SanDisk logo are registered trademarks of SanDisk Corporation. CompactFlash is a U.S. registered trademark of SanDisk Corporation.

Product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

© 2003 SanDisk Corporation. All rights reserved.

SanDisk products are covered or licensed under one or more of the following U.S. Patent Nos. 5,070,032; 5,095,344; 5,168,465; 5,172,338; 5,198,380; 5,200,959; 5,268,318; 5,268,870; 5,272,669; 5,418,752; 5,602,987. Other U.S. and foreign patents awarded and pending.

Lit. No. Rev. 1.0 10/2003 Printed in U.S.A.

Table of Contents

1	Introduction	1
2	SD-SDK Architecture Overview	2
3	SanDisk SDK Components	3
4	SD-SDK Directory Structure.....	5
5	Sample Application Installation	6
6	Known Problems and Limitations.....	8
7	Part Numbers.....	9

1 Introduction

This release note contains information about the SanDisk SDK architecture, installation of SD_SDK and sample application in Windows OS environment, system configuration supported by SD_SDK, and about known problems and limitations. If you encounter any problem while installing or using SD_SDK, please check here to see if it is a known problem.

This guide will give an overview of the SDK architecture, supported futures, SD-SDK directory structure, installation of the sample application and SDK on Windows OS. Finally this guide will reveal any known problems and/or limitations and Part Number information for ordering the individual software component.

2 SD-SDK Architecture Overview

The Secure Digital (SD) Memory Card offers protection of copyrighted materials such as digital music. Numerous underlying security elements work together to ensure that protected content is shielded from unauthorized access.

To accelerate the design cycle for products utilizing the SD Memory Card, SanDisk offers the SD Software Development Kit (SD-SDK). The SD-SDK contains all the software layers that developer need to integrate a host platform such as a music player with a SD Memory Card. Tailoring a few of the SDK configuration options and calling the appropriate SDK API functions are all that is necessary to enable a music player to retrieve song titles and initiate playing music files.

The components include:

- **FAT File System Engine:** FAT file system support for accessing files in both the user and protected areas of the SD Memory Card.
- **SD Memory Card Device Driver:** Provides support for low-level access to the SD Memory Card for both normal MultiMediaCard-compatible commands and the SD security commands
- **SD Audio Software Layer:** this layer contains intimate knowledge of the SD-Audio standard. It formulates requisite security key values, parses related files in the user and protected areas, and handles all aspects of decrypting/ playing protected content. It carries out security-related processing by calling into the SanDisk Security Manger (SSM) layer
- **SanDisk Security Manager (SSM):** Controls accesses that result in security commands being generated to the SD Memory Card. Protects critical key values and authentication file system elements.
- **Application Program Interface (API):** All functionality required by a typical music player is easily achieved through this API library. These functions include the ability to: mount/unmount volumes; retrieve play lists and track titles; play/stop/pause tracks; and skip forwards/backwards within a given musical track
- **Sample Music Player Application:** This sample application, written in Visual C++ 6.0 Enterprise Edition, demonstrates how the SD-SDK (SD-Audio) services can be easily incorporated into a music player application. It provides a sample GUI via which audio content can be downloaded, viewed, and played.

3 SanDisk SDK Components

SD-Audio Layer

Using the API shields host application software from the complexity of SD-Audio. Music player software that controls the display and playing of play lists can accomplish these tasks by calling the API functions. There's never any need to deal directly with the internals of the functions, files, and data structures.

Supported Functionality with 1.0 Release

- SD-Audio navigation support
 - This function's behavior model is to return the number of play lists, play list's name, number of tracks in each play list, and track information for each play list
 - This function also contains the list of available/free file number (AOBxxx.SA1) for download purpose
 - This function also contains links and information of a song that is stored in multiple files (2 or more files)
 - Handles the play list selection, and track selection
- Audio Playback support
 - Support continues play of the entire tracks in the play lists
 - Support single track selection playback
 - Random track selection playback
 - Also supports playback of all tracks that exists' in the SD Card
- Delete Operation
 - Multiple file deletion of the long song
 - Single track deletion
- Download Operation:
 - MP3 Audio format
 - AAC Audio format
- Playback of long song that is required more than one SDA file format to store
 - These files can be stored in sequence SDA files such as (AOB001.SA1, AOB002.SA1, etc).
 - These files can be stored out of order SDA files such as (AOB001.SA1, AOB004.SA1, etc).
- Download of long song, that requires more than one SDA file format to store
 - These files can be stored in sequence SDA files such as (AOB001.SA1, AOB002.SA1, etc).
 - These files can be stored out of order SDA files such as (AOB001.SA1, AOB004.SA1, etc).
- Next Track, Previous track function is supported in this release
- Fast Forward, Fast Backward support

- At any given operational state, the maximum buffer required is 1K, except during the download or playback states. These two depends on the host system

Security Manager Layer

This layer encapsulates the SD Security, and SD Secure Files System. Using the API, SD-Audio layer can access this layer, and all the complexity of MKB process, AKE process, Bus Encryption/Decryption, and Audio Content Encryption/Decryption is hidden from the application layer.

Supported Functionality with 1.0 Release

- AKE process is handled in this layer
- Bus Encryption/Decryption is performed immediately after each AKE process successfully finished
- MKB process is handled in this layer, its been invoked during the SD-Audio initialization
- To support the download operations, generating title key method implemented in this layer
- Title Key encryption/decryption
- Audio content encryption/decryption
- Secure file manipulation such as (open, read, delete, write, etc.) handled in this layer
- Required buffer in this layer is 1.5 Kbytes for internal use, this is need for the MKB process after the MKB process is complete this layer only require 512 bytes for other operations

File System Layer

This layer provides full FAT file system interface to SD card user area with API functions such as create, delete, insert, merge files, sub-directories, file date/time, file attributes and volume labels.

Supported Functionality with 1.0 Release

- Support for FAT12, FAT16 and FAT32
- Support for short and long file names

Device Driver Layer

This layer handles all low-level I/O access to MMC card and SD card. It provides eight functions. They are: xxx_init, xxx_drive_open, xxx_drive_close, xxx_read, xxx_write, xxx_erase, xxx_read_serial, where xxx is the peripheral bus name, such as MMC or SPI.

Supported Functionality with 1.0 Release

- Hardware access to memory card is through SanDisk verilog host controller.
- SD protocol, MMC protocol are implemented and tested on the top of the verilog host controller.
- SPI protocol is implemented but not tested on the top of the verilog host controller, due to the limitation of the host controller in SPI mode.

4 SD-SDK Directory Structure

There are four directories under the SD_SDK directory.

1. sdaudlib
2. sdbuild
3. uiapp
4. set32env

The **uiapp** directory is unnecessary if this package is used in any environment other than the Windows OS.

SD-SDK High-level Directory Structure

```
SD_SDK:
  README
  UIApp
    . base
    . config
    . debug
    . io
    . Imc
    . res
    . SdApp
      . Debug
      .SdApp.exe
      . res

  SdAudLib
    . Debug
      . sdaudlib.lib
    . Release
      .sdaudlib.lib

  SdBuild
    . Fatfiles
    . Header
    . Intrface
    . Oem
      . oem.lib
    . SD_Oem
      . sdoem.lib
    . Platform
    . TstSampl
      . README
    . Security
      . Tstsampl
      . README
      . security.lib

  Set32Env
    . set32.bat
```


5 Sample Application Installation

The required components necessary to install the sample application include a PCI-SD board, Windows OS, and a SD Card w/development key.

Running the Sample Application

Option 1

1. Unzip the SD_SDK package
2. Change directory to `sd_sdk\uiapp\sdapp\debug`
3. Run `sdapp.exe`

Note: Ensure that the SD Card is properly inserted and has at least one track in it.

Option 2

1. Follow SD_SDK installation guide to rebuild the `sdapp.exe` from source code
2. Run `sdapp.exe`

SD_SDK Installation Guide

The sequence enumerated below should be followed.

1. Install Visual C++ 6.0 Enterprise edition at the default folder.
2. Copy SD_SDK source file or unzip all the source file to `C:\SD_SDK` directory
3. Change `config.sys` file to include the following line: `Shell=C:\command.com c:\ /p /e:8192`
(see `C:\SD_SDK\set32env\config.sys` for reference)
4. Reboot machine
5. Setup environment variable for building library under DOS prompt
6. Type: `cd C:\SD_SDK\set32env`
`set32`
7. Build file system `OEM.lib` by typing:
`cd C:\SD_SDK\sdbuilt\OEM`
`nmake clean`
`nmake`
8. Build secure key system `SD_OEM.lib` :
type "`cd C:\SD_SDK\sdbuilt\SD_OEM`"
modify `sd_oem.c` to include OEM device key info to the SDK system
type "`nmake clean`"
type "`nmake`"

9. saf;lj Build security system Security.lib by typing:

```
cd C:\SD_SDK\sdbuild\SECURITY
nmake clean
nmake
```
10. Build SD audioAPI SdAudlib.lib by opening Visual C++ 6.0 sdaudlib.dsw workspace which is under C:\SD_SDK\sdaudlib\sdaudlib.dsw and choose menu “rebuild all” under “Build” menu item. You may need to set include path by choose menu “options” under “Tools” menu item. Add “C:\SD_SDK\UIApp\LMC\XINGMP3\INCLUDE” to “Directories”.
11. Build SD download utility application SdApp.exe by opening Visual C++ 6.0 sdApp.dsw workspace which is under C:\SD_SDK\UIApp\sdApp\sdApp.dsw and choose menu “rebuild all” under “Build” menu item. You may need to set include path by choose menu “options” under “Tools” menu item. Add “C:\SD_SDK\UIApp\LMC\XINGMP3\INCLUDE” to “Directories”.
12. Launch SdApp.exe by open Visual C++ 6.0 sdApp.dsw workspace which is under C:\SD_SDK\UIApp\sdApp\sdApp.dsw and choose “Execute SdApp.exe “ under Menu Build item. Ensure that the path to sdaudlib.lib, and security.lib is included in the make file settings.

6 Known Problems and Limitations

For the UIAPP, the following items are relevant.

- SdApp.exe is tested on Window 98 only
- Some of the songs, when being played, will be stuck due to the freeware source code used in the SdApp. Just click forward(>>) will let you continue to play
- Play function only supports MP3 songs, not AAC songs
- When click forward(>>), the timer will display 4 seconds, not 2 second forwarded, but in reality, the contents is only forwarded by 2 seconds

For the SD-Audio, the following items are not supported.

- Modify, Add, and Deletion of Playlists is not supported
- WMA audio format is not supported
- Creating a new playlist manager, track manager, key manager is not supported in this release

For the SD Secure Manager, the following items are relevant

- The file date of xxx.key file (in secure area) is not being updated even though the content of the file has been changed

7 Part Numbers

- **SDDK-05: SD Device Driver and File System**

This software component includes all the files under directory “SDBuild” except “SD_Oem” directory and “Security” directory. With those files, you would be able to build “Oem.lib”.

- **SDDK-06: Security Manager SDK**

This software component includes all the files under directory “SD_Oem” and directory “Security”. With those files and “Oem.lib”, you would be able to build “Security.lib”.

- **SDDK-07: SD Audio SDK**

This software component includes all the files under directory “SdAudLib”. With those files, you would be able to build “sdaudlib.lib”.

SanDisk Corporation
Corporate Headquarters
140 Caspian Court
Sunnyvale, CA 94089
408-542-0500
FAX: 408-542-0503
URL: <http://www.sandisk.com>