
Multiple Drive Support Users Guide

Version 1.0
June 4, 2008

S60 platform

Legal Notice

Copyright © 2008 Nokia Corporation. All rights reserved.

Nokia and Nokia Connecting People are registered trademarks of Nokia Corporation. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this document at any time, without notice.

License

A license is hereby granted to download and print a copy of this document for personal use only. No other license to any other intellectual property rights is granted herein.

Contents

1.	Introduction	5
1.1	Audience	5
2.	Overview of multiple drive support for the S60 platform	6
2.1	Current drive configuration.....	6
2.2	Future drive configurations.....	6
2.3	Compatibility.....	6
2.4	Multiple drive support in Symbian OS and File Server API	6
3.	Overview of the Multiple Drive Support Plug-in for the S60 3rd Edition SDK for Symbian OS, Feature Pack 2	7
3.1	Multiple Drive Support Plug-in for the S60 3rd Edition SDK for Symbian OS, Feature Pack 2.....	7
3.2	MDS-related API changes.....	7
3.3	Viewing multiple drives in the S60 emulator	7
4.	Installation	9
4.1	Platform requirements for the plug-in.....	9
4.2	Environment setup	9
4.3	Installation steps	9
4.3.1	Install the S60 SDK	9
4.3.2	Install the MDS Plug-in	9
4.3.3	Test the environment.....	10
5.	Using multiple drive support in the emulator	11
5.1	How to simulate various drive configurations with the emulator	11
6.	Known limitations	12
7.	Examples.....	13
7.1	Driveinfo usage: How to use the path to check the drive type.....	13
8.	Terms and abbreviations.....	14
9.	References	15
10.	Evaluate this resource	16

Change History

June 4, 2008	Version 1.0	Initial document release
--------------	-------------	--------------------------

1. Introduction

This Multiple Drive Support Users Guide provides:

- An overview of multiple drive support (MDS) and dynamic drive configurations for the S60 platform in Chapter 2, “Overview of multiple drive support for the S60 platform.”
- An overview of the Multiple Drive Support (MDS) Plug-in for the S60 3rd Edition SDK for Symbian OS, Feature Pack 2 in Chapter 3, “Overview of the Multiple Drive Support Plug-in for the S60 3rd Edition SDK for Symbian OS, Feature Pack 2.”
- Instructions on how to install the MDS Plug-in in Chapter 4, “Installation.”
- Instructions on how to simulate and change between drive configurations using the emulator in Chapter 5, “Using multiple drive support in the emulator.”

1.1 Audience

This user guide is designed for developers who create applications that include file and drive operations and who want to provide multiple drive support in their applications. Before starting, developers should be familiar with the following:

- Symbian OS v9.3 application development.
- The S60 SDK development environment.
- The APIs that provide services for file and drive handling as well as those that provide the UI for accessing files and drives.

2. Overview of multiple drive support for the S60 platform

2.1 Current drive configuration

Currently, the S60 platform supports two drives that are visible to the user (the device memory together with either a memory card or internal mass storage). These drives have a fixed order and fixed drive letters (namely **C:** for the device memory and **E:** for the memory card or mass storage). This has enabled developers to make assumptions about the type of storage in a particular location.

2.2 Future drive configurations

In the future, there will be S60 devices that offer more than two drives. In addition, drive configurations may vary between device models, and it will not be safe to assume that, for example, the C drive is the device's memory. New drives could result from devices' having additional memory cards (both internal and external), hard drive(s), or partitioned memory.

To enable an application to utilize these new drive configurations, developers will need to employ MDS and dynamically detect the drive configuration using the services provided by the MDS Plug-in. Multiple drive services provide the means to determine the order, letters, types, and number of drives, among other drive features.

2.3 Compatibility

New applications using MDS will be compatible with future S60 devices and SDKs for S60 3rd Edition, Feature Pack 2 or later. Applications that utilize MDS will not be compatible with S60 devices or SDKs for S60 3rd Edition, Feature Pack 1 or earlier.

Applications that use the current S60 APIs for file and drive handling will continue to work with future S60 devices and SDKs. However, these applications may have some functional limitations, such as not having access to all user-visible drives.

2.4 Multiple drive support in Symbian OS and File Server API

Symbian OS and its File Server API, in particular, already support multiple drives. MDS services provided for the S60 platform complement the services already available in the File Server API. Therefore, the MDS and Symbian OS APIs should be used in parallel. For more information about the File Server API, please refer to the *Symbian OS v9.3 Library for Developers* (included in the S60 3rd Edition SDK for Symbian OS, Feature Pack 2 SDK documentation).

3. Overview of the Multiple Drive Support Plug-in for the S60 3rd Edition SDK for Symbian OS, Feature Pack 2

3.1 Multiple Drive Support Plug-in for the S60 3rd Edition SDK for Symbian OS, Feature Pack 2

The Multiple Drive Support Plug-in for the S60 3rd Edition SDK for Symbian OS, Feature Pack 2 consists of:

- Emulator configuration scripts that allow the drive configurations seen in the emulator to be switched.
- An example application that illustrates the use of the MDS APIs.
- An emulator version of the S60 **File Manager** that implements MDS.
- Documentation describing the MDS APIs and the example application, as well as a user guide.

See Chapter 4, “Installation,” for information on where the items listed above are located after installing the MDS Plug-in.

3.2 MDS-related API changes

The Platform Environment API and Common File Dialogs API provide MDS features for third-party applications. The header files of these APIs are located in the `<S60_SDK_installation_directory>\epoc32\include` folder.

- The Common File Dialogs API contains the following MDS-related header files:
 - `CAknMemorySelectionDialogMultiDrive.h`
 - `CAknMemorySelectionSettingItemMultiDrive.h`
 - `AknCommonDialogsDynMem.h`
- The Platform Environment API contains the `driveinfo.h` MDS-related header file.

3.3 Viewing multiple drives in the S60 emulator

The plug-in replaces the emulator's **File Manager** application with one that has MDS support.

There are several ways in which MDS support can be observed in the replacement **File Manager**. With the **Internal mass storage with removable memory card** configuration enabled, the main view of the **File Manager** displays a list of three available drives, as shown in Figure 1.



Figure 1: The main view of the File Manager application with internal mass memory and removable memory card present

Also, when copying files using the **File Manager**, the drives of the **Internal mass storage with removable memory card** configuration are visible in the **Copy to:** dialog, as shown in Figure 2.



Figure 2: The Copy to: dialog showing internal mass memory and a removable memory card in addition to the phone memory

4. Installation

4.1 Platform requirements for the plug-in

The MDS plug-in can be installed only into the S60 3rd Edition SDK for Symbian OS, Feature Pack 2.

4.2 Environment setup

This chapter describes how to set up the MDS Plug-in. As a result of the environment-setup actions, developers will have the MDS Plug-in installed and can start developing applications that support multiple drives.

The three-step setup procedure is introduced below. Each step is described in more detail in the following chapters. The setup procedure is as follows:

1. Install the S60 3rd Edition SDK for Symbian OS, Feature Pack 2.
2. Install the MDS Plug-in.
3. Test the environment.

4.3 Installation steps

4.3.1 Install the S60 SDK

If it is not already installed, install the S60 3rd Edition SDK for Symbian OS, Feature Pack 2.

4.3.2 Install the MDS Plug-in

The MDS Plug-in is delivered as a ZIP archive. To install the plug-in:

1. Verify that the SDK emulator and SDK documentation browser are not running.
2. Extract the ZIP archive's contents to
`<S60_SDK_installation_directory>` (for example,
`C:\Symbian\9.3\S60_3rd_FP2_Beta`).



Note: Before extracting the ZIP archive's contents, ensure that **Use folder names** has been selected.

Once the plug-in has been installed:

- MDS-related header files can be found under
`<S60_SDK_installation_directory>\epoc32\include\`

See Chapter 3.2, "MDS-related API changes," for more information on the MDS-related header files.

- Drive configuration scripts can be found under

```
<S60_SDK_installation_directory>\epoc32\tools\mds\
```

Once one of the multiple drive configurations has been enabled, the MDS support can be seen in the emulator's **File Manager** application.

See Chapter 5.1, "How to simulate various drive configurations with the emulator," for more information on the drive configurations available.

- An example application demonstrating MDS can be found under

```
<S60_SDK_installation_directory>\S60CppExamples\Exa_CommonDi  
alogs
```

To view the example application documentation, open

```
<S60_SDK_installation_directory>\S60CppExamples\Exa_CommonDi  
alogs\doc\index.html
```

- The *Multiple Drive Support Users Guide* and the release notes for the MDS Plug-in are provided in `<S60_SDK_installation_directory>\docs\MDS`.
- This guide and the *Multiple Drive Support Plug-in API Reference Guide* are viewable in the SDK documentation.

4.3.3 Test the environment

To confirm whether the MDS environment is working, do the following:

1. Enable one of the multiple drive configurations (either internal mass storage with removable memory card or two removable memory cards).

See Chapter 5.1, "How to simulate various drive configurations with the emulator," for instructions on how to simulate drive configurations within the SDK emulator.

2. Launch the SDK emulator.
3. In the emulator, open the **File Manager** application.

The drives of the enabled multiple-drive configurations should be visible in the main view of the **File Manager**. An example view is shown in Figure 1.

5. Using multiple drive support in the emulator

5.1 How to simulate various drive configurations with the emulator

The MDS plug-in includes scripts that provide easy-to-use emulator drive configurations.

The drive configurations that can be simulated with the scripts are:

- A basic drive configuration, containing phone memory and memory card.
- An internal mass storage configuration with removable memory card and phone memory.
- A two-removable-memory-card configuration (phone memory, memory card 1, and memory card 2).

Script usage is explained below:

- To enable the **Basic drive** configuration, run the `SetupEmu_PhoneMemMmc.cmd` script.
- To enable the **Internal mass storage with removable memory card** configuration, run the `SetupEmu_PhoneMemMassStorageMmc.cmd` script.
- To enable the **Two removable memory cards** configuration, run the `SetupEmu_PhoneMemMmc1Mmc2.cmd` script.

Script usage:

1. Make sure that the development environment and the emulator are not running.
2. Open the command prompt window and run `epoc32\tools\mds\SetupEmu_<script_identifier>.cmd`.
3. Start the SDK emulator.



Note: Do *not* edit the drive configurations in the `epoc.ini` or `epoc_XXX.ini` files while the configuration scripts are in use.

Setting the memory card to a folder changes drive attributes, and the drive may not be recognized correctly.

6. Known limitations

This section lists known limitations of MDS in general, as well as MDS differences between the emulator and S60 devices.

- It is not possible to distinguish between internal (not physically removable) mass storage and an external (physically removable/hot-swappable) memory card while using the File Server API. The Platform Environment API must be used for this purpose (see Chapter 7.1, “Driveinfo usage: How to use the path to check the drive type”).



Note: Even though internal mass storage cannot be physically removed from the device, it can be dismantled temporarily from the device’s file system.

Dismantling happens, for example, when the device is in mass storage mode and the USB cable is connected. The Platform Environment API provides the “in use” drive status (`EDriveInUse`) in this situation.

- In the emulator environment, the **F5** key removes or sets all memory cards at once; it is not possible to remove or set memory cards one by one. This limitation affects the **Internal mass storage with removable memory card** configuration and the **Two removable memory cards** configuration.

After memory cards are removed using the **F5** key, the internal mass storage is in the “not present” state, even though the **File Manager** indicates that it is still “in use.” The **File Manager** reports the drive status in this way because in an S60 device, the internal mass storage cannot be in the “not present” state at any time.

7. Examples

7.1 Driveinfo usage: How to use the path to check the drive type

This example shows how to distinguish between internal (not physically removable) mass storage and an external (physically removable/hot-swappable) memory card using the Platform Environment API. The method relies on status information provided about each drive type:

- Internal mass storage drives can have drive statuses of `EDriveInternal` | `EDriveExternallyMountable`.
- External memory cards can have drive statuses of `EDriveRemovable` | `EDriveExternallyMountable`.
- Phone memory has the drive status of `EDriveInternal`, and it cannot be dismantled.

The code in Example 1 illustrates how this information is used to determine the drive type.

```
#include <driveinfo.h>
void CMyContainer::CheckDriveTypeL( const TDesC& aFullPath )
{
    // Convert path to drive id, assume that the length
    // of aFullPath is always over zero
    TInt drv = TDriveUnit( aFullPath );
    TInt drvStatus( 0 );
    User::LeaveIfError( DriveInfo::GetDriveStatus(iFs,drv,drvStatus));
    if ( !( drvStatus & DriveInfo::EDriveUserVisible ) )
    {
        // Drive is system drive like RAM or ROM drive
    }
    else if ( drvStatus & DriveInfo::EDriveInternal )
    {
        if ( drvStatus & DriveInfo::EDriveExternallyMountable )
        {
            // Drive is internal mass storage
        }
        else
        {
            // Drive is phone memory
        }
    }
    else if ( drvStatus & DriveInfo::EDriveRemovable )
    {
        // Drive is physically removable drive
        // like memory card or other external mass storage
    }
    else if ( drvStatus & DriveInfo::EDriveRemote )
    {
        // Drive is remote drive
    }
}
```

Example 1: Using the path to check the drive type

8. Terms and abbreviations

Term or abbreviation	Meaning
API	application programming interface
Internal mass storage	Internal memory (for example, memory card or hard drive) that a user cannot physically remove from the device
MDS	multiple drive support
UI	user interface
USB	Universal Serial Bus

9. References

File Server API documentation	Symbian OS v9.3 Library for Developers
-------------------------------	--

10. Evaluate this resource

Please spare a moment to help us improve documentation quality and recognize the resources you find most valuable, by [rating this resource](#).