

Device Driver Update For SPOT Software

Overview

This utility is used for updating the device drivers that are used by the Spotcam.dll camera driver. This is a standard utility that is included with the SpotCam SDK and will be updated as needed with each release of the SDK. This utility is the recommended method for updating the device drivers for the target computer. Included is a sample application (CopyFiles.exe) with source code and a Visual C++ project file for evaluation. There are two link libraries included with the SpotDrv SDK that can be used for linking with Microsoft Visual C++ and Borland compilers they are SpotDrv.lib and SpotDrvBor.lib respectively.

API Functions

SpotDrvrSetSetting

The **SpotDrvrSetSetting** is used to set options that need to be set before calling [SpotDrvrInstall](#).

Syntax

```
SpotDrvrError SpotDrvrSetSetting(  
    IN SpotDrvrSetting TheSetting,  
    IN void* pData);
```

Parameters

TheSetting

[in] The setting that is to be set. Must be one of the values within the enumeration [SpotDrvrSetting](#).

pData

[in] Points to the value(s) to be set for the specified setting. The data type pointed to depends on the value of *TheSetting*. See section on [SpotDrvrSetting](#) for information on data types.

Return Value

If the operation is successful the return value is [SD_ERROR_SUCCESS](#).

If the operation fails the value of [SD_ERROR_SETTING_INVALID](#) is returned.

Calling [SpotDrvrGetLastError](#) after the function completes will return the same value.

Remarks

Settings that require a boolean data type may also set *pData* to NULL to enable the setting.

See Also

[SpotDrvrInstall](#); [SpotDrvrSetting](#); [SpotDrvrGetLastError](#)

SpotDrvrInstall

The function **SpotDrvrInstall** will start the driver update installation with the current settings.

Syntax

```
SpotDrvrError SpotDrvrInstall(  
    void *pReserved);
```

Parameters

pReserved

Reserved for future use. Must be set to NULL.

Return Value

If the operation is successful the return value is [SD_ERROR_SUCCESS](#).

If the operation fails the value is non zero and is any one of the values in [SpotDrvrError](#).

Remarks

This function will not return until the installation is completed. If the installation needs to be aborted call [SpotDrvrAbort](#) to cancel the installation. If aborted the return value will be [SD_ERROR_ABORTED](#).

See Also

[SpotDrvrAbort](#); [SpotDrvrSetting](#)

SpotDrvrAbort

Use **SpotDrvrAbort** to abort the current installation process.

Syntax

```
SpotDrvrError SpotDrvrAbort (  
    IN bool bAbort);
```

Parameters

bAbort

[in] Boolean value to abort the installation. Set to non zero to abort.

Return Value

If the function is successful the return value is [SD_ERROR_SUCCESS](#).

Remarks

The abort flag must be reset to false (0) if the installation is to be restarted.

See Also

[SpotDrvrInstall](#)

SpotDrvrGetLastError

Use **SpotDrvrGetLastError** to retrieve the last error code.

Syntax

```
SpotDrvrError SpotDrvrGetLastError ();
```

Parameters

NONE

Return Value

The value may be any one of the values in [SpotDrvrError](#).

Remarks

NONE

See Also

[SpotDrvrError](#)

SpotDrvrSetCallBack

The **SpotDrvrSetCallBack** function is used to set a callback function which will be called periodically during the installation.

Syntax

```
SpotDrvError SpotDrvSetCallBack(  
    IN SPOTDRVRCALLBACK pfnCallback,  
    IN DWORD dwUserData);
```

Parameters

pfnCallback
Pointer to a [SPOTDRVRCALLBACK](#) function. This function is called multiple times during the execution of [SpotDrvInstall](#) with status values. To disable the callback after it was already enabled set this parameter to NULL.

dwUserData
[in] Data that is always returned to the callback function.

Return Value

The return value is always [SD_ERROR_SUCCESS](#).

Remarks

This function should be set prior to any call to SpotDrvInstall if the callback is needed.

See Also

[SpotDrvStatus](#), [SPOTDRVRCALLBACK](#), [SpotDrvInstall](#)

SpotDrvLogFile

The **SpotDrvLogFile** function is used to set a log file that will output any system changes to it. The log file will be ANSI text file which writes each operation on a new line in the form <Operation>[-][Description].

Syntax

```
SpotDrvError SpotDrvLogFile(  
    IN char* pszFilename);
```

Parameters

pszFilename
Pointer to a null terminated string that contains the full file path. File path must not be longer than the defined value of MAX_FILE_PATH.

Return Value

If the function is successful the return value is [SD_ERROR_SUCCESS](#).
If the function fails the returned value is [SD_ERROR_INVALID_PARAMETER](#) path is longer than MAX_PATH or [SD_ERROR_CREATING_FILE](#) if the file cannot be created.

Remarks

If the file specified by pszFilename exists it will be truncated to zero. Each call to this function will reset the log file. If the function fails the previous log file will be invalidated as well leaving no log file specified.

See Also

[SpotDrvInstall](#)

SpotDrvIsAdministrator

The **SpotDrvIsAdministrator** function can be used to determine if the calling process has administrative privileges for the local machine. This is useful to determine if the application should continue or not.

Syntax

```
SpotDrvError SpotDrvIsAdministrator();
```

Parameters

NONE

Return Value

If the calling process has administrative privileges the return value is [SD_ERROR_SUCCESS](#).

If the calling process does not have administrative privileges the returned value is [SD_ERROR_NOT_ADMINISTRATOR](#)

Remarks

The [SpotDrvInstall](#) function checks for administrative privileges before starting the installation as well. Therefore, this function is redundant and not necessary.

See Also

NONE

Enumerations

SpotDrvSetting

SD_SETTING_ALLOW_GUI

Description: Allow the DLL to display GUI dialogs that prompt the user for some action. Otherwise the default action must be specified by the calling application. Note that some versions of Windows® will display a dialog box prompting a user for information even if this setting is set to false this is normal and unavoidable.

Data Type : bool

Inverse Setting: SD_SETTING_NO_GUI

Default Value : true

SD_SETTING_NO_GUI

Description: Disallow the DLL to display GUI dialogs that prompt the user for some action. Note that some versions of Windows® will display a dialog box prompting a user for information even if this setting is set to false this is normal and unavoidable.

Data Type : bool

Inverse Setting: SD_SETTING_ALLOW_GUI

Default Value : false

SD_SETTING_ALLOW_RESTART

Description: Allow the DLL to initiate a system shutdown process if one is necessary. If this is set to false then SpotDrvInstall will return with [SD_ERROR_NEED_TO_RESTART](#) if a restart is needed.

Data Type : bool

Inverse Setting: SD_SETTING_NO_RESTART

Default Value : true

SD_SETTING_NO_RESTART

Description: Disallow the DLL to initiate a system shutdown process if one is necessary. If this is set to true then SpotDrvInstall will return with [SD_ERROR_NEED_TO_RESTART](#) if a restart is needed.

Data Type : bool

Inverse Setting: SD_SETTING_ALLOW_RESTART

Default Value : false

SpotDrvError

These are the return values that the API functions return.

SD_ERROR_SUCCESS

Description: The operation completed successfully.

SD_ERROR_CREATING_FILE

Description: There was an error creating a file.

SD_ERROR_FILE_IS_NEWER

Description: Overwriting a newer file is not allowed.

SD_ERROR_FILE_NOT_FOUND

Description: A file that is needed can not be found.

SD_ERROR_FILE_CORRUPT

Description: The file being read has been damaged and is unreadable.

SD_ERROR_UNKNOWN

Description: An unknown error has occurred.

SD_ERROR_NEED_TO_RESTART

Description: The computer needs to be restarted for the operation to complete.

SD_ERROR_CREATING_DIRECTORY

Description: There was an error trying to create a directory.

SD_ERROR_OS_NOT_SUPPORTED

Description: The operating system is not supported by this function.

SD_ERROR_ABORTED

Description: The function has been forced to abort.

SD_ERROR_SETTING_INVALID

Description: The supplied setting is not a valid setting.

SD_ERROR_INVALID_PARAMETER

Description: One of the parameters supplied is not valid.

SD_ERROR_NOT_ADMINISTRATOR

Description: The calling application is doesn't have administrative privileges for the local machine.

SD_ERROR_BUSY

Description: The current function cannot be called because it is currently busy.

SD_ERROR_WRITING_TO_REGISTRY

Description: There was an error writing to the registry.

SpotDrvStatus

These are the status values that are passed back to the callback function.

SD_STATUS_LOCATING_DEVICE

Description: Locating a device and attempting to update the device.

pExtraData : Pointer to a [CAMERA_INTERFACE_TYPE](#). May be NULL.

SD_STATUS_REMOVING_DEVICE

Description: About to update a device's drivers.

pExtraData : Pointer to a [CAMERA_INTERFACE_TYPE](#). May be NULL.

SD_STATUS_UPDATED_DEVICE

Description: Successfully updated a device's drivers.

pExtraData : Pointer to a [CAMERA_INTERFACE_TYPE](#). May be NULL.

SD_STATUS_UPDATING_REMOVED_DEVICE

Description: Removing an uninstalled device's old drivers.

pExtraData : Pointer to a [CAMERA_INTERFACE_TYPE](#). May be NULL.

SD_STATUS_DELETING_FILE

Description: Removing a file from the machine.

pExtraData : Pointer to a null terminated string of chars that contains the full file path.

SD_STATUS_INSTALLING_FILE

Description: Installing a file in the machine.

pExtraData : Pointer to a null terminated string of chars that contains the full file path.

SD_STATUS_CREATING_DIRECTORY

Description: Creating a directory on the machine.

pExtraData : Pointer to a null terminated string of chars that contains the full path.

SD_STATUS_STARTING_SERVICES

Description: Starting device driver services.

pExtraData : NULL.

SD_STATUS_STOPPING_SERVICES

Description: Stopping device driver services.

pExtraData : NULL.

SD_STATUS_RESTARTING

Description: The system shutdown process is going to be initiated. The callback function should not return from this status message until it is safe to shut down the computer.

pExtraData : NULL.

SD_STATUS_UPDATING_REGISTRY

Description: Updating the system registry.

pExtraData : NULL.

SpotDrvrBusType

These are the camera interface bus types that are used to determine what type of `DEVICE_XXX` structure is located in the [CAMERA_INTERFACE_TYPE](#) structure.

SD_BUS_PCI

Description: The [CAMERA_INTERFACE_TYPE](#) structure contains a [DEVICE_PCI](#) struct.

SD_BUS_1394

Description: The [CAMERA_INTERFACE_TYPE](#) structure contains a [DEVICE_1394](#) struct.

Structures

DEVICE_PCI

The structure **DEVICE_PCI** contains information on PCI card identification.

Syntax

```
typedef struct _DEVICE_PCI
{
    WORD        wVendor;
    WORD        wDevice;
    DWORD       dwSubSystem;
    BYTE        byRevision;
    DWORD       dwReserved;
} DEVICE_PCI, *PDEVICE_PCI;
```

Parameters

wVendor
The Vendor ID of the PCI Device.

wDevice
The Device ID of the PCI Device.

dwSubSystem
The Sub System version of the PCI device.

byRevision
The Revision level of the PCI device.

dwReserved
Reserved for future use.

Remarks

This structure is contained in a [CAMERA_INTERFACE_TYPE](#) structure if the [SpotDrvBusType](#) is [SD_BUS_PCI](#).

See Also

[SPOTDRVRCALLBACK](#), [SpotDrvStatus](#), [CAMERA_INTERFACE_TYPE](#)

DEVICE_1394

The structure **DEVICE_1394** contains information on IEEE 1394 (Firewire[®]) devices.

Syntax

```
typedef struct _DEVICE_1394
{
    DWORD       dwUnitSpecId;
    DWORD       dwUnitSwVersion;
    char        szDescription[488];
} DEVICE_1394, *PDEVICE_1394;
```

Parameters

dwUnitSpecId
The Unit Spec ID for the 1394 device.

dwUnitSwVersion
The unit software ID for the 1394 device.
szDescription
A user friendly name for the device.

Remarks

This structure is contained in a [CAMERA_INTERFACE_TYPE](#) structure if the [SpotDrvrBusType](#) is [SD_BUS_1394](#).

See Also

[CAMERA_INTERFACE_TYPE](#)

CAMERA_INTERFACE_TYPE

The structure **CAMERA_INTERFACE_TYPE** contains information on camera identification.

Syntax

```
typedef struct _CAMERA_INTERFACE_TYPE
{
    SpotDrvrBusType    BusType;
    void                *pReserved;
    union
    {
        DEVICE_PCI      stPCIDevice;
        DEVICE_1394      st1394Device;
    };
} CAMERA_INTERFACE_TYPE, *PCAMERA_INTERFACE_TYPE;
```

Parameters

BusType
Specifies which DEVICE_XXX bus type is contained in the structure.
pReserved
Reserved for future use.
stPCIDevice
Valid only if the BusType is SD_BUS_PCI.
st1394Device
Valid only if the BusType is SD_BUS_1394.

Remarks

This structure should contain the appropriate structure depending on the value located in *BusType*.

See Also

[SPOTDRVRCALLBACK](#), [SpotDrvrStatus](#), [DEVICE_PCI](#), [DEVICE_1394](#)

PRODUCT_INFO

The structure **PRODUCT_INFO** contains information on the products identification.

Syntax

```
typedef struct _PRODUCT_INFO
{
    char szManufacturerName[256];
```

```

    char szProductName[256];
    SpotDrvrBusType    BusType;
    union
    {
        DEVICE_PCI      stPCIDevice;
        DEVICE_1394     st1394Device;
    };
} PRODUCT_INFO;

```

Parameters

szManufacturerName
The manufacturer of the product.

szProductName
The name of the product.

BusType
Specifies which DEVICE_XXX bus type is contained in the structure.

stPCIDevice
Valid only if the BusType is SD_BUS_PCI.

st1394Device
Valid only if the BusType is SD_BUS_1394.

Remarks

This structure should contain the appropriate structure depending on the value located in *BusType*.

See Also

[SpotDrvrBusType](#), [DEVICE_PCI](#), [DEVICE_1394](#)

SPOTDRVRCALLBACK

The **SPOTDRVRCALLBACK** is a prototype function declaration for the function [SpotDrvrSetCallBack](#).

Syntax

```

typedef void (WINAPI *SPOTDRVRCALLBACK)
    (SpotDrvrStatus CurrentStatus,
     void *pExtraData,
     DWORD dwUserData);

```

Parameters

CurrentStatus
The current status of the installation.

pExtraData
Pointer to any extra data that the current status may have. See [SpotDrvrStatus](#) for details on data types.

dwUserData
Data that will be returned to the callback function every time it is called.

Remarks

NONE.

See Also

[SpotDrvrStatus](#), [SpotDrvrSetCallBack](#)

Revision History

Version 2.1.3

Updated spotkp.sys to version 3.0.4.

Version 2.1.2

Updated spotkp.sys to version 3.0.3.

Version 2.1.1.2

Updated atmcdwdm.sys to version 4.26.

Version 2.1.1

Added structure PRODUCT_INFO to the API. Updated Spotkp.sys to version 3.0.2.

Version 2.1

Added kernel driver atmcdwdm.sys. Initial support for SPOT Boost cameras.

Version 2.0.2

Corrected an error that would fail to overwrite older version files with newer revision files if they differed by only the one of the last two revision fields. (eg. 1.1.2.1 would fail to overwrite 1.1.1.1)

Version 2.0.1

Added additional ICC color profiles to the installation to support new camera types. (Same files that were added to 1.6.1)

Version 1.6.1

Added additional ICC color profiles to the installation to support new camera types.

Version 2.0

Updated kernel driver SpotKP.sys to version 3.0.1

Removed all VXD type drivers from installing on Windows 9x systems, installed their SYS counterparts.

Version 1.6

Added support to install ICC color profiles into the system profile directory.

Updated file SpotDev2.dll which adds enhanced reliability for FireWire cameras on computers with SpeedStepping technology.

Version 1.5

Added support to install ICC color profiles into the system profile directory.