VIDEO BLASTER MCI OVERLAY DRIVER

Overview

This document describes the implementation of the Video Blaster MCI Overlay driver (MCIVBLST.DRV). The reader is assumed to have some knowledge of MCI programming in Windows 3.1 or Windows 3.0 with Multimedia Extensions. In particular, the reader should be aware of MCI command messages and strings and how to use them in their applications to control the video overlay driver.

The Video Blaster overlay driver tries to conform as closely as possible to the Overlay driver specifications laid out in the Windows 3.1 Multimedia Programmer's Reference. The following section presents the commands supported by the overlay driver, and describes their behaviour in this implementation.

Installation

At this time, we have not yet created an OEMSETUP.INF file for setting up the MCIVBLST overlay driver. As such we require you to perform the following steps to install the MCIVBLST.DRV overlay driver into your Windows system.

1) Copy the MCIVBLST.DRV file into the Windows directory.
2) Make sure PCVIDEO.DLL is in the Windows directory, or at least the PATH environment variable is set correctly for access to the DLL.
3) Edit your SYSTEM.INI file, and add the following setting to the MCI section:

```
[MCI]
Overlay=mcivblst.drv
```

There are no settings to be done for the driver, as it makes use of the PCVIDEO.DLL to execute the video routines.

Video Overlay Commands

The overlay command set provides a common method for displaying overlay video. With the overlay driver, applications need not program at the DLL level. Further, it is possible to leave much of the live video handling to the driver. Therefore, applications which require basic usage of live video need not include code to manipulate live video, which will instead be taken care of by the overlay driver.

The following describes the set of commands used with video overlay devices, in command-string form. For more details on the MCI command strings and their corresponding command messages, please refer to the Windows 3.1 Multimedia Programmer's Reference.

Note: Although the **notify** and **wait** keywords are implemented in MCIVBLST, both will result in a wait for the specified command to finish execution before returning to the application.

capability

capability *device_id parameter* [**notify**] [**wait**]

Requests information about the capabilities of the video overlay devices.

Parameters:

can eject	Returns **false**.
can freeze	Returns **true**.
can play	Returns **false**.
can record	Returns **false**.
can save	Returns **true**. MCIVBLST supports saving images to disk.
can stretch	Returns **false**. MCIVBLST does not allow video from the source rect to be stretched to the destination rect (see the **put** command for more details).
compound device	Returns **false**. Multiple windows are not supported; there will only be one live video window at any time.
device type	Returns **overlay**.
has audio	Returns **false**. The Video Blaster does not perform audio playback. (The audio mixing and passthrough provided by the Video Blaster is not considered to be audio playback.)
has video	Returns **true**.
uses files	Returns **false**.
windows	Returns 1.

close

close *device_id* [**notify**] [**wait**]

Closes the overlay device and releases the resources used by the driver.

freeze

freeze *device_id* [**notify**] [**wait**]

Disables video acquisition to the video buffer.

The **at** parameter described by the specifications is not implemented in MCIVBLST, but will not cause an error if used. If specified with the **freeze** command, the entire video buffer will be frozen.

info

info *device_id parameter* [**notify**] [**wait**]

Gets textual information from the device.

Parameters:

file	Returns a blank string (""), since MCIVBLST is not a compound device and so files are not used.
product	Returns the MCIVBLST overlay driver product name.
window text	Returns the caption of the window used by MCIVBLST. Defaults to **Video Blaster**.

load

load *device_id* [*parameters*] [**notify**] [**wait**]

Loads an image file into the video buffer. MCIVBLST will determine what type of file from the contents of the file.

Parameters:

filename	Specifies the filename and path of the image file to load into the buffer.
at *rectangle*	Specifies a rectangle relative to the video buffer origin. *rectangle* is specified as *X Y W H*, where *X Y* specify the top-left corner of the rectangle, and *W H* specify the width and height. Note: The width and height information is ignored by MCIVBLST, as it does not support stretching of the loaded image. Also, *X* should be on a 4-pixel boundary, due to the YUV4:1:1 nature of the video buffer.

open

open *device_id* [*parameters*] [**notify**] [**wait**]

Initialises the video overlay device.

Parameters:

alias *device_alias*	Specifies an alternate name for the device element. If specified, it must be used as the *device_id* in subsequent commands.
parent *hwnd*	Specifies the window handle of the parent window.
shareable	Not supported by MCIVBLST. Only one application may use the MCIVBLST overlay driver at any one time.
style *style_type*	Indicates a window style.
style child	Opens a window with a child window style. This parameter can only be specified when the **parent** parameter is present. Note: When this style is specified, the application is responsible for all messages sent to the child window. This includes processing for putting live video into the window (which may be done using the **put** command, described below) whenever the child window is moved or resized.
style popup	Opens a window with a popup window style.
type *device_type*	Specifies the device type, which should be **overlay**. Note that this command is only useful

with the command messages, as the command string format interprets this command to be a device with an element, which is not supported by MCIVBLST.

put

put *device_id parameters* [**notify**] [**wait**]

The **put** command defines one or more of the following rectangles:

+ The **video** rectangle defines the region of the incoming video image to capture. In other words, it imposes a crop-window on the video source, *before* acquisition into the video buffer.
+ The **frame** rectangle defines the region of the frame buffer that receives the incoming video image. This rectangle specifies the live video region in the video buffer.
+ The **source** rectangle defines which region of the frame buffer is copied to the destination. MCIVBLST takes this rectangle to specify the portion of the video buffer to display on the screen.
+ The **destination** rectangle defines the region of the display window client area that receives the video image. This rectangle specifies the portion of the display window that will be used to show the live video defined by the **source** rectangle.

This is the command that gives the application control over the placement of live video what to display and where to display it. By means of these rectangles, the application may specify what part of the video to be captured (the **video** rectangle), where to capture it into (the **frame** rectangle), which part of the buffer to show (the **source** rectangle) and which part of the client window to show it (the **destination** rectangle).

This command should be used to place live video when the application takes over control of the video display window (with the **open**...**parent**...**child** command or the **window**...**handle** command).

The following are some implementation details of the MCIVBLST that the developer should take note of:

+ If the **video** rectangle is larger than the **frame** rectangle, scaling will occur from the former to the latter.
+ If the **video** rectangle is smaller than the **frame** rectangle, scaling will not occur, but the capture size will be clipped to that of the **video** rectangle.
+ *Stretching from **video** to **frame** is not supported.*
+ The **source** rectangle directly references the video buffer, and thus may be made to show static images, or live video, or part of both. Naturally, if specified totally wrongly, live video may not appear at all, which could be alarming to the user.
+ The **destination** rectangle is specified with respect to the client window area.
+ *Stretching and scaling from **source** to **destination** is not supported.*
+ Instead, the actual size of the video shown will be clipped to the smaller of the **source** and **destination** rectangles and the window client area size.

Parameters:

video Selects the default incoming video image to capture in the frame buffer. The default incoming video image is as defined during the setting up of the Video Blaster software.

video at *rectangle* Selects a portion of the incoming video image to capture in the frame buffer. The *rectangle* coordinates are relative to the video origin and are specified as *X Y W H*. The coordinates *X Y* specify the top-left corner of the rectangle, and the coordinates *W H* specify the width and height.

frame Selects the entire frame buffer to receive the incoming video images.

frame at *rectangle* Selects a portion of the frame buffer to receive the incoming video images. The *rectangle* coordinates are relative to the video origin and are specified as *X Y W H*. The coordinates *X Y* specify the top-left corner of the rectangle, and the coordinates *W H* specify the width and height.

source Selects the video buffer to display in the **destination** window.

source at *rectangle* Selects a portion of the video buffer to display in the **destination** window. The *rectangle* coordinates are relative to the video origin and are specified as *X Y W H*. The coordinates *X Y* specify the top-left corner of the rectangle, and the coordinates *W H* specify the width and height.

destination Selects the entire client area of the **destination** window to display the video data from the frame buffer.

destination at *rectangle* Selects a portion of the client area of the **destination** window to display the video data from the frame buffer. The *rectangle* coordinates are relative to the video origin and are specified as *X Y W H*. The coordinates *X Y* specify the top-left corner of the rectangle, and the coordinates *W H* specify the width and height.

Examples:

The following suit of commands fit video to a display window of size 200,200:

```
put vboard frame at 0 0 200 200
put vboard source at 0 0 200 200
put vboard destination at 0 0 200 200
```

The following suit of commands fit video to an area (of size 200,200) in the middle of a window with size 400,400:

```
put vboard frame at 0 0 200 200
put vboard source at 0 0 200 200
put vboard destination at 100 100 200 200
```

The following suit of commands display video buffer data with size 300,300 in the middle of a window with size 400,400. In this video buffer display, live video is shown in the top-right quadrant; the rest shows static video:

```
put vboard frame at 150 0 150 150
put vboard source at 0 0 300 300
put vboard destination at 50 50 300 300
```

save

save *device_id filename* [*parameters*] [**notify**] [**wait**]

Saves the contents of the video buffer to a disk file.

Parameters:

at *rectangle* Specifies a rectangle relative to the video buffer origin. The *rectangle* is specified as *X Y W H*. The coordinates *X Y* specify the top-left corner of the rectangle, and the coordinates *W H* specify the width and height. If this parameter is omitted, the driver will save from the displayed video area.

Note: Due to the YUV4:1:1 nature of the video buffer, *X* and *W* should be on a 4-pixel boundaries.

type *file_type* Specifies the image type to save as. Valid types are:

Ordinal value		Default ext
DIB 24-bit		0
DIB 8-bit		1
	.BMP	
DIB 8-bit gray	2	
Targa 32-bit	4	
Targa 24-bit	5	
Targa 16-bit	6	
	.TGA	
IBM MMotion	7	
	.MMP	
TIFF 24-bit	8	
TIFF 8-bit		9
	.TIF	
TIFF 8-bit gray		10
PCX 8-bit		11
	.PCX	
PCX 8-bit gray		12
GIF 8-bit		14

.GIF

GIF 8-bit gray

15

If this parameter is specified, the file type will always override the file extension.

If not specified, the file type used will depend on the extension of the filename. The actual format used is as indicated in the table above.

If this parameter is not specified and no file extension is given for the filename, or the extension is not one of the above, the file type defaults to DIB 8-bit.

Note: This parameter is not defined in the Windows 3.1 MCI specifications.

set

set *device_id parameters*

Establishes control settings for the driver.

Parameters:

audio all off Disables audio output. Not supported by MCIVBLST.

audio all on Enables audio output. Not supported by MCIVBLST.

audio left off Disables output to the left audio channel. Not supported by MCIVBLST.

audio left on Enables output to the left audio channel. Not supported by MCIVBLST.

audio right off Disables output to the right audio channel. Not supported by MCIVBLST.

audio right on Enables output to the right audio channel. Not supported by MCIVBLST.

time format milliseconds Not supported by MCIVBLST.

video off Disables video output.

video on Enables video output.

status

status *device_id parameter* [**notify**] [**wait**]

Gets status information for the device.

Parameters:

media present Returns **true**.

mode Not supported by MCIVBLST.

ready Returns **true**.

stretch Returns **false**. MCIVBLST does not support stretching of video from the **source** rectangle to the **destination** rectangle.

window handle Returns the handle of the window used for the video overlay display in the low word of the return value.

unfreeze

unfreeze *device_id* [**notify**] [**wait**]

Enables the frame buffer to acquire video data.

The **at** parameter described by the MCI specifications is not implemented in MCIVBLST, but will not cause an error if used. If specified with the **unfreeze** command, the entire video buffer will be unfrozen.

where

where *device_id parameter* [**notify**] [**wait**]

Gets the rectangle specifying the **video**, **frame**, **source** or **destination** area.

Parameter:

video Returns the offset and extent of the **video** rectangle.

frame Returns the offset and extent of the **frame** rectangle.

source Returns the offset and extent of the **source** rectangle.

destination Returns the offset and extent of the **destination** rectangle.

window

window *device_id parameters* [**notify**] [**wait**]

Controls the destination window. The destination window is the window in which the image is displayed. You can change the display characteristics of the window or provide a destination window for the driver to use in place of the default destination window.

MCIVBLST creates a window when opened, but does not display it. The window may be displayed with a **window**...**show** command after creation. If the application provides a window to the driver, it is responsible for managing the messages sent to the window. In particular, it is also responsible for the placement of live video within the window, which may be done using the **put** command.

The **window** command provides several flags that let you manipulate the window. Since you can use the **status** command to get the handle to the destination window, you can also use the standard window manager functions (like **ShowWindow**) to manipulate the window.

Parameter:

fixed Not supported by MCIVBLST; stretching is not supported.

handle *window_handle* Specifies the handle of a window to use instead of the default destination window.

handle default Specifies that MCIVBLST manage its own destination window. This flag can be used to set the display back to the driver's default window.

state hide Hides the destination window.

state iconic Displays the destination window as an icon.

state maximized Maximizes the destination window.

state minimize Minimizes the destination window and activates the top-level window in the window-manager's list.

state minimized Minimizes the destination window.

state no action Displays the destination window in its current state. The window currently active remains active.

state no activate Displays the destination window in its most recent size and state. The currently active window remains active.

state normal Displays the destination window as it was created.

state show Shows the destination window.

stretch Not supported by MCIVBLST; stretching is not supported.

text *caption* Specifies the *caption* for the destination window.