Apple Game Sprockets Reference

(Not Recommended)

Carbon > Games

2006-07-13

Apple Inc.
© 2006 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Carbon, Mac, Mac OS, Macintosh, Quartz, and QuickTime are trademarks of Apple Inc., registered in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, Inc.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Apple Game Sprockets Reference (Not Recommended)

Framework: DrawSprocket/DrawSprocket.h

Declared in DrawSprocket.h

Important: The information in this document is obsolete and should not be used for new development.

Overview

Game Sprockets is a set of libraries that supported the creation of games in Mac OS 8 and 9. The libraries provided services for working with sound, imaging, user input, and network gaming. Most of Game Sprockets—including the InputSprocket, NetSprocket, and SoundSprocket libraries—is not available in Mac OS X. For documentation on these legacy APIs, see *Apple Game Sprockets Legacy Reference*. For detailed information on using Game Sprockets in Mac OS 8 and 9, see Apple Game Sprockets.

A subset of the DrawSprocket library, which provides functions to configure and control displays, has remained available in Carbon to facilitate the porting of legacy applications to Mac OS X. The DrawSprocket API has been deprecated for deployment targets Mac OS X version 10.4 and later. The replacement is Quartz Display Services, a modern Mac OS X API that provides similar functionality. For more information, see *Quartz Display Services Reference*.

Functions by Task

Activating and Deactivating DrawSprocket

`DSpGetVersion` (page 48) Deprecated in Mac OS X v10.4

Determines the version of DrawSprocket installed on the host computer. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpShutdown` (page 51) Deprecated in Mac OS X v10.4

Shuts down DrawSprocket. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpStartup` (page 51) Deprecated in Mac OS X v10.4

Initializes DrawSprocket. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Choosing a Context

`DSpContext_GetAttributes` (page 28) Deprecated in Mac OS X v10.4

Obtains the attributes of a given context. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_GetDisplayID` (page 31) Deprecated in Mac OS X v10.4

Obtains the ID of the display a context is associated with. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpFindBestContext` (page 42) Deprecated in Mac OS X v10.4

Finds the context that best matches the requirements you specify. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpFindBestContextOnDisplayID` (page 43) Deprecated in Mac OS X v10.4

Determines the best context to use for a given display. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpGetCurrentContext` (page 45) Deprecated in Mac OS X v10.4

Obtains a reference to the current display context for a given display. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpGetFirstContext` (page 45) Deprecated in Mac OS X v10.4

Obtains the first context in the list of contexts available for a specified display. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpGetNextContext` (page 47) Deprecated in Mac OS X v10.4

Obtains the next context in a list of available contexts for a display. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Drawing and Double Buffering

`DSpContext_FadeGamma` (page 25) Deprecated in Mac OS X v10.4

Sets brightness of the display to the specified intensity. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_FadeGammaIn` (page 27) Deprecated in Mac OS X v10.4

Completely fades in a display to a color of your choice. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_FadeGammaOut` (page 27) Deprecated in Mac OS X v10.4

Completely fades out a display to a color of your choice. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_GetBackBuffer` (page 29) Deprecated in Mac OS X v10.4

Obtains the back buffer for the context. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_GetFrontBuffer` (page 32) Deprecated in Mac OS X v10.4

Obtains the front buffer for the context. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_GetMonitorFrequency` (page 33) Deprecated in Mac OS X v10.4

Obtains the frequency for the display associated with a context. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

`DSpContext_IsBusy` (page 34) Deprecated in Mac OS X v10.4

Finds out whether a back buffer is available. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_SwapBuffers` (page 41) Deprecated in Mac OS X v10.4

Draws a context's back buffer to the screen. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Handling a Mouse

`DSpContext_GlobalToLocal` (page 34) Deprecated in Mac OS X v10.4

Translates a point in global coordinates into local coordinates for a context. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_LocalToGlobal` (page 35) Deprecated in Mac OS X v10.4

Translates a point from a context's local coordinates into global coordinates. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpFindContextFromPoint` (page 44) Deprecated in Mac OS X v10.4

Finds out which context contains a point given in global coordinates. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpGetMouse` (page 46) Deprecated in Mac OS X v10.4

Obtains the global coordinates of the mouse position. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Manipulating a Context

`DSpContext_GetState` (page 33) Deprecated in Mac OS X v10.4

Determines the current play state of a context. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_Queue` (page 36) Deprecated in Mac OS X v10.4

Queues a context you want to switch to. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_Release` (page 37) Deprecated in Mac OS X v10.4

Releases a context you are finished using. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_Reserve` (page 37) Deprecated in Mac OS X v10.4

Reserves a context so that you can begin using it in your game. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_SetState` (page 39) Deprecated in Mac OS X v10.4

Sets the play state of a context. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_Switch` (page 42) Deprecated in Mac OS X v10.4

Switches display contexts. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpSetBlankingColor` (page 49) Deprecated in Mac OS X v10.4

Assigns a background color to the blanking window for all displays. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Manipulating Color Lookup Tables

`DSpContext_GetCLUTEntries` (page 30) Deprecated in Mac OS X v10.4

Retrieves one or more color entries from a color lookup table. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

`DSpContext_SetCLUTEntries` (page 39) Deprecated in Mac OS X v10.4

Assigns one or more color entries to a color lookup table. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Processing System Events

`DSpProcessEvent` (page 48) Deprecated in Mac OS X v10.4

Passes system events through to DrawSprocket so that it can correctly handle events it must know about. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Utility Functions

`DSpSetDebugMode` (page 50) Deprecated in Mac OS X v10.4

Keeps the screen and system resources visible during debugging. (Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Miscellaneous

`DSpContext_Dispose` (page 25) Deprecated in Mac OS X v10.4

(Deprecated. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Callbacks

DSpBlitDoneProc

Defines a pointer to a blitting completion function. Your callback function handles any tasks required after DrawSprocket finishes blitting between buffers.

```
typedef void (*DSpBlitDoneProc) (
    DSpBlitInfo * info
);
```

If you name your function `My`, you would declare it like this:

```
void DSpBlitDoneProc (
    DSpBlitInfo * info
);
```


Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Parameters

info

A pointer to a data structure containing information about the completed blitting operation. See `DSpBlitInfo` (page 11) for more information.

Return Value

Discussion

If you are performing multiple asynchronous blitting operations, your application-defined completion function can check the blitter information structure passed to it to determine which operation was completed.

Version Notes

Introduced with DrawSprocket 1.1

Availability

Available in Mac OS X v10.0 and later.

Declared In

`DrawSprocket.h`

DSpCallbackProcPtr

Defines a pointer to a callback function. Your callback function performs any necessary tasks in preparation for swapping display buffers or piggybacking VBL tasks to a context.

```
typedef Boolean (*DSpCallbackProcPtr)
(
    DSpContextReference inContext,
    void * inRefCon
);
```

If you name your function `MyDSpCallbackProc`, you would declare it like this:

```
Boolean DSpCallbackProcPtr (
    DSpContextReference inContext,
    void * inRefCon
);
```

Parameters

inContext

A reference to a context.

inRefCon

A reference constant to be handed back to the game by the DrawSprocket function that calls `MyCallbackFunction`.

Return Value

The function should return `false` if your tasks or checks are complete. If it returns `true`, the function is still performing necessary tasks.

Discussion

Calls to `MyCallbackFunction` result from calls to `DSpContext_SwapBuffers` (page 41).

Version Notes

Introduced with DrawSprocket 1.0.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`DrawSprocket.h`

Data Types

DSpAltBufferAttributes

Describes the characteristics of an alternate buffer.

```
struct DSpAltBufferAttributes {
    UInt32 width;
    UInt32 height;
    DSpAltBufferOption options;
    UInt32 reserved[4];
};
typedef struct DSpAltBufferAttributes DSpAltBufferAttributes;
```

Fields

`width`
The width of the alternate buffer, in pixels.

`height`
The height of the alternate buffer, in pixels.

`options`
Any desired options for the alternate buffer. See "Alternate Buffer Options Constant" (page 15).

`reserved`
Reserved. Set to 0.

Discussion
When handling allocating an alternate drawing buffer, you can specify additional attributes by passing a structure of type `DSpAltBufferAttributes`.

Version Notes
Introduced with DrawSprocket 1.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`DrawSprocket.h`

DSpAltBufferReference

Represents an alternate drawing buffer,

```
typedef struct OpaqueDSpAltBufferReference * DSpAltBufferReference;
```

Version Notes
Introduced with DrawSprocket 1.0.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`DrawSprocket.h`

DSpBlitInfo

Specifies the type of blitting operation when blitting between buffers.

```
struct DSpBlitInfo {
    Boolean completionFlag;
    char filler[3];
    DSpBlitDoneProc completionProc;
    DSpContextReference srcContext;
    CGrafPtr srcBuffer;
    Rect srcRect;
    UInt32 srcKey;
    DSpContextReference dstContext;
    CGrafPtr dstBuffer;
    Rect dstRect;
    UInt32 dstKey;
    DSpBlitMode mode;
    UInt32 reserved[4];
};
typedef struct DSpBlitInfo DSpBlitInfo;
typedef DSpBlitInfo * DSpBlitInfoPtr;
```

Fields

`completionFlag`
Set to `true` on output when the blitting operation has completed.

`filler`
Reserved. These bytes are included to preserve proper alignment.

`completionProc`
A pointer to the function that DrawSprocket should call when the blitting operation has completed. See `DSpBlitDoneProc` (page 8) for more information about implementing this function. Pass `NULL` if you don't want to specify a completion function.

`srcContext`
A reference to the source context. Pass `NULL` if the source buffer does not belong to a context.

`srcBuffer`
A pointer of type `CGrafPtr` that specifes the buffer containing the image data you want to blit to the destination buffer.

`srcRect`
The rectangle specifying the location of the image data in the source buffer. If the source and destination rectangles are different sizes, DrawSprocket scales the image to fit.

`srcKey`
An integer specifying the source color key. See "Blit Mode Constants" (page 16) for more information on using this key.

`dstContext`
A reference to the destination context. Pass `NULL` if the destination buffer does not belong to a context.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

`dstBuffer`
: A pointer of type `CGrafPtr` that specifes the buffer you want to blit the image to.

`dstRect`
: The rectangle specifying where to write the image data in the destination buffer. If the source and destination rectangles are different sizes, DrawSprocket scales the image to fit.

`dstKey`
: An integer specifying the destination color key. See "Blit Mode Constants" (page 16) for more information on using this key.

`mode`
: The blit mode to use. See "Blit Mode Constants" (page 16) for a list of possible values.

`reserved`
: Reserved.

Version Notes
Introduced with DrawSprocket 1.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`DrawSprocket.h`

DSpContextAttributes

Indicates the characteristics of a drawing context.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
struct DSpContextAttributes {
    Fixed frequency;
    UInt32 displayWidth;
    UInt32 displayHeight;
    UInt32 reserved1;
    UInt32 reserved2;
    UInt32 colorNeeds;
    CTabHandle colorTable;
    OptionBits contextOptions;
    OptionBits backBufferDepthMask;
    OptionBits displayDepthMask;
    UInt32 backBufferBestDepth;
    UInt32 displayBestDepth;
    UInt32 pageCount;
    char filler[3];
    Boolean gameMustConfirmSwitch;
    UInt32 reserved3[4];
};
typedef struct DSpContextAttributes DSpContextAttributes;
typedef DSpContextAttributes * DSpContextAttributesPtr;
```

Fields

`frequency`

Input: Ignored.

Output: The frame-refresh frequency (in Hz) specified by the current resolution mode. (This value is 0 if the actual frequency is not available.

`displayWidth`

Input: The requested display width (in pixels).

Output: The display width for the specified context.

`displayHeight`

Input: The requested display height (in pixels).

Output: The display height for the specified context.

`reserved1`

Reserved. Always set this field to 0.

`reserved2`

Reserved.

`colorNeeds`

Input: A value that specifies whether the display needs to be in color. Valid constants for this field are described in "Color Need Constants" (page 17).

Output: The color support provided by the current resolution mode.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

`colorTable`

Input: A handle to the color table to use with the context to which this attributes structure applies. (This field applies only to indexed devices; direct devices do not use a color table.)

Output: Ignored.

`contextOptions`

Input: A set of bit flags that define requested special display features for which either hardware or software implementation is acceptable. Valid constants for this field are described in "Special Display Feature Constants" (page 20).

Output: The special display features supported in software by the current resolution mode.

`backBufferDepthMask`

Input: A bit array that defines the acceptable pixel depths for the back buffer. Valid constants for this field are described in "Depth Mask Constants" (page 18). This value should match the depth of the front buffer. You must specify a back buffer depth mask and pixel depth when reserving a back buffer context.

Output: The bit depth DrawSprocket recommends for the context.

`displayDepthMask`

Input: A bit array that defines the acceptable pixel depths for the front buffer. Valid constants for this field are described in "Depth Mask Constants" (page 18).

Output: A bit array that specifies the pixel depth of the specified context.

`backBufferBestDepth`

Input: The preferred pixel depth, or video mode, for the back buffer. his value should match the depth of the front buffer. You must specify a back buffer depth mask and pixel depth when reserving a back buffer context.

Output: The bit depth DrawSprocket recommends for the context.

`displayBestDepth`

Input: The preferred pixel depth for the display.

Output: The pixel depth of the specified context.

`pageCount`

Input: Indicates the desired number of video pages. For example, if you desire double-buffering, you should pass 2. For triple buffering, pass 3. If you pass 1, then DrawSprocket only provides a front buffer and does not allocate any memory for back buffers. You cannot pass 0 for the page count.

Output: Gives the number of hardware video pages available. A value of 1 indicates that hardware page flipping is not supported.

`filler`

Reserved. These bytes are included to preserve alignment.

`gameMustConfirmSwitch`

Input: Ignored.

Output: A value of `true` indicates that the context may have problems being displayed on the user's system, and the game should confirm that the context is visible after being set to the active state by asking the user if the display can be seen (via a dialog box or some other mechanism). Additionally, a warning code will be returned from `DSpContext_SetState` indicating that the game should confirm that the context is visible.

`reserved3`

Reserved. Always set this field to 0.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Discussion
You use the context attributes structure to request specific characteristics when creating a context or to retrieve the actual characteristics of a given context. The field descriptions cover their use as both input or output values, but the structure never contains both input and output information at the same time.

You can use the debug version of the DrawSprocket library to catch most context errors.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`DrawSprocket.h`

DSpContextReference

Represents a drawing context.

```
typedef struct OpaqueDSpContextReference * DSpContextReference;
```

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`DrawSprocket.h`

DSpContextReferenceConst

```
typedef const struct OpaqueDSpContextReference * DSpContextReferenceConst;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`DrawSprocket.h`

Constants

Alternate Buffer Options Constant

Describes options when allocating an alternate buffer.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
enum DSpAltBufferOption {
    kDSpAltBufferOption_RowBytesEqualsWidth = 1 << 0
};
typedef enum DSpAltBufferOption DSpAltBufferOption;
```

Constants

`kDSpAltBufferOption_RowBytesEqualsWidth`

Forces the row and width of the alternate buffer to have the same number of pixels. The number of row bytes can vary depending on the screen depth. For example, if you specify 16-bit color, then there will be twice as many row bytes as there are pixels in the width, because it takes 2 bytes to represent one pixel.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Version Notes

Introduced with DrawSprocket 1.1.

Blit Mode Constants

Indicate the type of blitter operation to perform.

```
enum DSpBlitMode {
    kDSpBlitMode_Plain = 0,
    kDSpBlitMode_SrcKey = 1 << 0,
    kDSpBlitMode_DstKey = 1 << 1,
    kDSpBlitMode_Interpolation = 1 << 2
};
typedef enum DSpBlitMode DSpBlitMode;
```

Constants

`kDSpBlitMode_Plain`

Copy all pixels from the source to the destination.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

`kDSpBlitMode_SrcKey`

Copies all image data where the source image is not the same color as the source key. For example, say the buffer holds a sprite image on a black background. If you specify the source color key to be black, then DrawSprocket writes only nonblack images (that is, the sprite) to the destination.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

`kDSpBlitMode_DstKey`

Overwrites data in the destination image where the color is the same as the destination key. For example, say the destination buffer holds an image of a a city skyline against a blue sky, and you want to draw a blimp moving behind the buildings. If you set the destination color key to blue, then DrawSprocket will draw the blimp only in areas that are blue. That is, the blimp will not overwrite the nonblue buildings, so it will appear to be behind them.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

`kDSpBlitMode_Interpolation`
Interpolate between color values when scaling pixels.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Discussion
You use these constants in the structure `DSpBlitInfo` (page 11) . Note that you can use these constants in combination with each other.

Version Notes
Introduced with DrawSprocket 1.1.

Buffer Kind Constant

Defines the type of buffer.

```
enum DSpBufferKind {
    kDSpBufferKind_Normal = 0
};
typedef enum DSpBufferKind DSpBufferKind;
```

Constants
`kDSpBufferKind_Normal`
Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Discussion
Currently, DrawSprocket supports only one kind of buffer. You pass this constant to the `DSpContext_GetBackBuffer`, `DSpAltBuffer_InvalRect`, and `DSpAltBuffer_GetCGrafPtr` functions.

Version Notes
Introduced with DrawSprocket 1.0.

Color Need Constants

Specify your program's preferences or requirements for color display in the `colorNeeds` field of the context attributes structure.

```
enum DSpColorNeeds {
    kDSpColorNeeds_DontCare = 0,
    kDSpColorNeeds_Request = 1,
    kDSpColorNeeds_Require = 2
};
typedef enum DSpColorNeeds DSpColorNeeds;
```

Constants
`kDSpColorNeeds_DontCare`
Display can be either color or grayscale.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

`kDSpColorNeeds_Request`
: Color display is preferred, but not required.

  Available in Mac OS X v10.0 and later.

  Declared in `DrawSprocket.h`.

`kDSpColorNeeds_Require`
: Color display is required.

  Available in Mac OS X v10.0 and later.

  Declared in `DrawSprocket.h`.

Version Notes
Introduced with DrawSprocket 1.0.

Depth Mask Constants

Define the allowable bit depth.

```
enum DSpDepthMask {
    kDSpDepthMask_1 = 1 << 0,
    kDSpDepthMask_2 = 1 << 1,
    kDSpDepthMask_4 = 1 << 2,
    kDSpDepthMask_8 = 1 << 3,
    kDSpDepthMask_16 = 1 << 4,
    kDSpDepthMask_32 = 1 << 5,
    kDSpDepthMask_All = -1L
};
typedef enum DSpDepthMask DSpDepthMask;
```

Constants

`kDSpDepthMask_1`
: 1-bit pixel depth is acceptable.

  Available in Mac OS X v10.0 and later.

  Declared in `DrawSprocket.h`.

`kDSpDepthMask_2`
: 2-bit pixel depth is acceptable.

  Available in Mac OS X v10.0 and later.

  Declared in `DrawSprocket.h`.

`kDSpDepthMask_4`
: 4-bit pixel depth is acceptable.

  Available in Mac OS X v10.0 and later.

  Declared in `DrawSprocket.h`.

`kDSpDepthMask_8`
: 8-bit pixel depth is acceptable.

  Available in Mac OS X v10.0 and later.

  Declared in `DrawSprocket.h`.

`kDSpDepthMask_16`
: 16-bit pixel depth is acceptable.

  Available in Mac OS X v10.0 and later.

  Declared in `DrawSprocket.h`.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

`kDSpDepthMask_32`
32-bit pixel depth is acceptable.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

`kDSpDepthMask_All`
Any pixel depth is acceptable.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Discussion
You provide a depth mask in the `backBufferDepthMask` and `displayDepthMask` fields of the context attributes structure to specify the pixel depths that are acceptable to your program.

Version Notes
Introduced with DrawSprocket 1.0.

Play State Constants

Indicate the current state of a drawing context.

```
enum DSpContextState {
    kDSpContextState_Active = 0,
    kDSpContextState_Paused = 1,
    kDSpContextState_Inactive = 2
};
typedef enum DSpContextState DSpContextState;
```

Constants
`kDSpContextState_Active`
The display is completely controlled by your program. The display is configured as specified in its context attributes structure. All system adornments, such as the menu bar, floating windows, and the desktop, are hidden (removed or covered by the blanking window). In this state you cannot make system calls to managers, such as the Window Manager and Dialog Manger, that expect the display to be in a normal state.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

`kDSpContextState_Paused`
The menu bar and other system adornments are restored, although the resolution mode is still that specified in the context attributes structure for the display. The desktop is still not visible; the blanking window covers it. In this state you can make normal system calls. Page flipping and double buffering are inactive in this state; the display page is set to page 0 if page flipping has been enabled.

In this state it is safe for your program to call Mac OS system software functions. The paused state gives the user access to the process menu; if your game is suspended because the user switches to another application, you must call the function `DSpProcessEvent` (page 48).

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

`kDSpContextState_Inactive`

The display is in exactly the state the user has specified from the Monitors control panel. The blanking window is hidden and the resolution mode specified in the context attributes structure for this display is not in effect.

The user's configuration is restored only if there are no other currently active or paused contexts. As long as there is at least one active or paused context, all displays are covered by the blanking window, and the resolution mode for each is that of the context, which may not be what the user has selected in the Monitors control panel.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Discussion
You set the play state of a context by calling the function `DSpContext_SetState` (page 39) and passing it one of these values.

Version Notes
Introduced with DrawSprocket 1.0.

Special Display Feature Constants

Indicate special display features in the `contextOptions` field of the context attributes structure.

```
enum DSpContextOption {
    kDSpContextOption_PageFlip = 1 << 1,
    kDSpContextOption_DontSyncVBL = 1 << 2,
    kDSpContextOption_Stereoscopic = 1 << 3
};
typedef enum DSpContextOption DSpContextOption;
```

Constants
`kDSpContextOption_PageFlip`

Use page flipping (a hardware feature). Note that you should never allow page flipping unless you have tested your code extensively on a computer with page-flipping capability.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

`kDSpContextOption_DontSyncVBL`

Do not synchronize context updates with the vertical retrace of the display.

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

`kDSpContextOption_Stereoscopic`

Available in Mac OS X v10.0 and later.

Declared in `DrawSprocket.h`.

Version Notes
Introduced with DrawSprocket 1.0.

Result Codes

The most common result codes returned by Game Sprockets are listed below.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Result Code	Value	Description
noErr	0	No error Available in Mac OS X v10.0 and later.
kNSpInitializationFailedErr	-30360	NetSprocket could not be initialized Available in Mac OS X v10.0 and later.
kNSpAlreadyInitializedErr	-30361	NetSprocket has already been initialized Available in Mac OS X v10.0 and later.
kNSpTopologyNotSupportedErr	-30362	The requested topology is unimplemented, or invalid value Available in Mac OS X v10.0 and later.
kNSpProtocolNotAvailableErr	-30366	A protocol reference indicated a protocol that is unavailable Available in Mac OS X v10.0 and later.
kNSpInvalidGameRefErr	-30367	An invalid game reference was passed Available in Mac OS X v10.0 and later.
kNSpInvalidParameterErr	-30369	A generic parameter error occurred Available in Mac OS X v10.0 and later.
kNSpOTNotPresentErr	-30370	Open Transport is not installed or not installed correctly Available in Mac OS X v10.0 and later.
kNSpOTVersionTooOldErr	-30371	The version of Open Transport available is too old to use with NetSprocket Available in Mac OS X v10.0 and later.
kNSpMemAllocationErr	-30373	NetSprocket has run out of memory Available in Mac OS X v10.0 and later.
kNSpAlreadyAdvertisingErr	-30374	The game is already being advertised on the specified protocol Available in Mac OS X v10.0 and later.
kNSpNotAdvertisingErr	-30376	The game is not being advertised at this time Available in Mac OS X v10.0 and later.
kNSpInvalidAddressErr	-30377	An invalid address was passed Available in Mac OS X v10.0 and later.
kNSpFreeQExhaustedErr	-30378	NetSprocket has exhausted its allocated queue elements and new messages will be dropped Available in Mac OS X v10.0 and later.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Result Code	Value	Description
kNSpRemovePlayerFailedErr	-30379	Your attempt to remove a player failed Available in Mac OS X v10.0 and later.
kNSpAddressInUseErr	-30380	You are attempting to use an address that is already in use Available in Mac OS X v10.0 and later.
kNSpCreateGroupFailedErr	-30388	The attempt to create a group failed Available in Mac OS X v10.0 and later.
kNSpTimeoutErr	-30393	A time out error has occurred Available in Mac OS X v10.0 and later.
kNSpGameTerminatedErr	-30394	An attempt to terminate the game has failed Available in Mac OS X v10.0 and later.
kNSpConnectFailedErr	-30395	A connection attempt has failed Available in Mac OS X v10.0 and later.
kNSpSendFailedErr	-30396	An attempt to send a message has failed Available in Mac OS X v10.0 and later.
kNSpMessageTooBigErr	-30397	The message you wanted to send was too long Available in Mac OS X v10.0 and later.
kNSpCantBlockErr	-30398	The player you sent a message to is not playing the game Available in Mac OS X v10.0 and later.
kNSpJoinFailedErr	-30399	The attempt to join the game failed Available in Mac OS X v10.0 and later.
kISpInternalErr	-30420	Internal error. Available in Mac OS X v10.0 and later.
kISpSystemListErr	-30421	Operation is not allowed a system-maintained element list. Available in Mac OS X v10.0 and later.
kISpBufferToSmallErr	-30422	The buffer is too small. Available in Mac OS X v10.0 and later.
kISpElementInListErr	-30423	The element is already in the element list. Available in Mac OS X v10.0 and later.
kISpElementNotInListErr	-30424	The element is not in the element list. Available in Mac OS X v10.0 and later.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Result Code	Value	Description
kISpSystemInactiveErr	-30425	InputSprocket is currently inactive. Available in Mac OS X v10.0 and later.
kISpDeviceInactiveErr	-30426	The device is currently inactive. Available in Mac OS X v10.0 and later.
kISpSystemActiveErr	-30427	Input Sprocket is currently active. Available in Mac OS X v10.0 and later.
kISpDeviceActiveErr	-30428	The device is currently active. Available in Mac OS X v10.0 and later.
kISpListBusyErr	-30429	The element list is marked as busy. Available in Mac OS X v10.0 and later.
kDSpNotInitializedErr	-30440	DSpStartup has not yet been called. Available in Mac OS X v10.0 and later.
kDSpSystemSWTooOldErr	-30441	System software too old. Available in Mac OS X v10.0 and later.
kDSpInvalidContextErr	-30442	Invalid context reference. Available in Mac OS X v10.0 and later.
kDSpInvalidAttributesErr	-30443	Some field in an attributes structure has an invalid value. Available in Mac OS X v10.0 and later.
kDSpContextAlreadyReservedErr	-30444	The context is already reserved. Available in Mac OS X v10.0 and later.
kDSpContextNotReservedErr	-30445	The context is not reserved. Available in Mac OS X v10.0 and later.
kDSpContextNotFoundErr	-30446	DrawSprocket couldn't find the context. Available in Mac OS X v10.0 and later.
kDSpFrameRateNotReadyErr	-30447	Not enough time has passed for DrawSprocket to calculate a frame rate. Available in Mac OS X v10.0 and later.
kDSpConfirmSwitchWarning	-30448	The gameMustConfirmSwitch flag is set. Available in Mac OS X v10.0 and later.
kDSpInternalErr	-30449	Corrupted DrawSprocket or other error. Available in Mac OS X v10.0 and later.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Result Code	Value	Description
kDSpStereoContextErr	-30450	DrawSprocket attempted to process a stereo context. (DrawSprocket no longer supports GoggleSprocket.) Available in Mac OS X v10.0 and later.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Deprecated Apple Game Sprockets Reference (Not Recommended) Functions

A function identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.4

DSpContext_Dispose

(Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_Dispose (
   DSpContextReference inContext
);
```

Parameters

inContext

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
DrawSprocket.h

DSpContext_FadeGamma

Sets brightness of the display to the specified intensity. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_FadeGamma (
   DSpContextReference inContext,
   SInt32 inPercentOfOriginalIntensity,
   RGBColor *inZeroIntensityColor
);
```

Parameters

inContext

A reference to the context whose display is to be faded. If you pass NULL for this parameter, the fade operation applies simultaneously to all displays.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

inPercentOfOriginalIntensity

The percentage (0–100) of the display's full intensity that you want to achieve with this call. Values above 100 percent begin to converge on white. If you have specified an intensity color, values less than zero begin to converge on black.

inZeroIntensityColor

A pointer to the color that is to correspond to zero intensity (represented by a value of 0 in the `inPercentOfOriginalIntensity` parameter). If you pass `NULL` for this parameter, the zero-intensity color is black.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
Fading the display is an aesthetically pleasing way to transition into and out of your game and between different sections of it. When performing a resolution-mode switch (as when activating and deactivating your context's play state), it is important to fade the display to hide the flash that occurs.

DSpContext_FadeGamma performs a gamma fade, which gives better results than a simple indexed fade.

Fading using `DSpContext_FadeGamma` is an incremental process. That is, over a period of time, you make repeated, timed calls to `DSpContext_FadeGamma`, each time passing it an incrementally different value for the `inPercentOfOriginalIntensity` parameter, until the final desired intensity is achieved. The intensity value you pass is usually an integer between 0 and 100. It can be greater than 100, if you want to use fading to create a high-intensity burst of light, or less than 100 if you have specified a zero-intensity color and want to fade the color toward black.

The zero-intensity value that you fade out to is by default black, but it can be any color that you specify in the `inZeroIntensityColor` parameter. You can achieve special effects by fading partially toward one zero-intensity color and then completing the fade to a different one. At the point when you actually switch resolution modes, the zero-intensity color must be black and your display must be completely faded if there is to be no visible flash.

To automatically accomplish a smooth fade all the way from full intensity to zero intensity, or vice versa, in a single operation, use the function `DSpContext_FadeGammaIn` (page 27) and the `DSpContext_FadeGammaOut` (page 27) function.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
OpenGLMovieQT

Declared In
`DrawSprocket.h`

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

DSpContext_FadeGammaIn

Completely fades in a display to a color of your choice. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_FadeGammaIn (
   DSpContextReference inContext,
   RGBColor *inZeroIntensityColor
);
```

Parameters

inContext

A reference to the context whose display is to be faded. The function fades the display from 0 percent to 100 percent intensity over a period of one second. If you pass `NULL` for this parameter, the fade operation applies simultaneously to all displays.

inZeroIntensityColor

The color that is to correspond to zero intensity. If you pass `NULL` for this parameter, the zero-intensity color is black.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Discussion

A key press or a mouse-button click will jump the fade to its end point immediately.

You can perform a manual fade with the function `DSpContext_FadeGamma` (page 25).

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code

DrawSprocketTestOld

GlyphaIVOld

Simple DrawSprocket

Declared In

`DrawSprocket.h`

DSpContext_FadeGammaOut

Completely fades out a display to a color of your choice. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpContext_FadeGammaOut (
   DSpContextReference inContext,
   RGBColor *inZeroIntensityColor
);
```

Parameters

inContext

A reference to the context whose display is to be faded. The function fades the display from 100 percent to 0 percent intensity over a period of one second. If you pass `NULL` for this parameter, the fade operation applies simultaneously to all displays.

inZeroIntensityColor

A pointer to the color that is to correspond to zero intensity. If you pass `NULL` for this parameter, the zero-intensity color is black.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
The initial gamma is that set by DrawSprocket when `DSpStartup` (page 51) was called, or the last gamma value set by calling the `DSpContext_FadeGamma` (page 25) function. If you had changed the system gamma to a different value, you may see a flash at the beginning of the fade due to the change in the initial gamma.

A key press or a mouse button click will jump the fade to its end point immediately.

You can perform a manual fade with the `DSpContext_FadeGamma` function.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld
GlyphaIVOld
Simple DrawSprocket

Declared In
`DrawSprocket.h`

DSpContext_GetAttributes

Obtains the attributes of a given context. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpContext_GetAttributes (
   DSpContextReferenceConst inContext,
   DSpContextAttributesPtr outAttributes
);
```

Parameters

inContext

The context whose attributes you want to obtain.

outAttributes

On return, `outAttributes` points to a context attributes structure. See `DSpContextAttributes` (page 12) for more information.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld

Declared In
`DrawSprocket.h`

DSpContext_GetBackBuffer

Obtains the back buffer for the context. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_GetBackBuffer (
   DSpContextReference inContext,
   DSpBufferKind inBufferKind,
   CGrafPtr *outBackBuffer
);
```

Parameters

inContext

A reference to the context whose back buffer is to be returned.

inBufferKind

The kind of buffer. Currently the only supported buffer kind is `kDSpBufferKind_Normal`.

outBackBuffer

On return, a pointer to the back buffer (that is, to a `CGrafPort`).

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
The back buffer, which is where the game should draw to, is the next buffer that will be displayed on a call to the function `DSpContext_SwapBuffers` (page 41).

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

The pointer to the back buffer may change after a call to `DSpContext_SwapBuffers`, so you must call this function before rendering every frame.

If you have specified an underlay for the context, the back buffer will have the underlay image restored before this call returns.

If there are no available back buffers (they are all queued up for display), this function will block until one is available. To avoid blocking, call the `DSpContext_IsBusy` (page 34) function until it returns `false`.

Note that 3D hardware accelerators typically must draw using a graphics device (`GDevice`) rather than a graphics port.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld
GlyphaIVOld

Declared In
`DrawSprocket.h`

DSpContext_GetCLUTEntries

Retrieves one or more color entries from a color lookup table. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_GetCLUTEntries (
   DSpContextReferenceConst inContext,
   ColorSpec *outEntries,
   UInt16 inStartingEntry,
   UInt16 inLastEntry
);
```

Parameters

inContext
The context whose color lookup table is to be accessed.

outEntries
On return, an array of color specification records that contain the retrieved table entries.

inStartingEntry
The (zero-based) index position in the color lookup table of the first entry to retrieve.

inLastEntry
The number of entries to retrieve.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Discussion
After you get the entries you can modify them and reassign them to the color table, for purposes such as color-table animation, with the function `DSpContext_SetCLUTEntries` (page 39).

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld

Declared In
`DrawSprocket.h`

DSpContext_GetDisplayID

Obtains the ID of the display a context is associated with. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_GetDisplayID (
   DSpContextReferenceConst inContext,
   DisplayIDType *outDisplayID
);
```

Parameters

inContext
A reference to the context whose monitor display ID you want to determine.

outDisplayID
On return, the display ID for the monitor associated with the context.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
Note that 3D hardware accelerators (such as RAVE) typically must draw using a graphics device (`GDevice`) rather than a graphics port. To do so, you can call `DSpContext_GetDisplayID` to get the display ID of the device associated with the context and then call the Display Manager function `DMGetDeviceByDisplayID` to obtain the `GDevice`.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Related Sample Code
DrawSprocketTestOld
Simple DrawSprocket

Declared In
`DrawSprocket.h`

DSpContext_GetFrontBuffer

Obtains the front buffer for the context. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_GetFrontBuffer (
   DSpContextReferenceConst inContext,
   CGrafPtr *outFrontBuffer
);
```

Parameters

`inContext`
A reference to the context whose front buffer is to be returned.

`outFrontBuffer`
On return, a pointer to the front buffer (that is, to a `CGrafPort`).

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
The front buffer is the screen display. Typically you use this function when you are not using backbuffers and you want to pass a `CGrafPtr` so another interface can draw to the screen (for example, by using OpenGL or QuickTime, or you simply want to change resolutions). However, if you are drawing to the screen yourself, you must call `DSpContext_GetFrontBuffer` each time through your game's drawing cycle to compensate for possible page flipping.

Note that 3D hardware accelerators (such as RAVE) typically must draw using a graphics device (`GDevice`) rather than a graphics port. To do so, you should call `DSpContext_GetDisplayID` (page 31) to get the display ID of the device the context is on and then call the Display Manager function `DMGetDeviceByDisplayID` to obtain the `GDevice`.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.1.2.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
Simple DrawSprocket

Declared In
`DrawSprocket.h`

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

DSpContext_GetMonitorFrequency

Obtains the frequency for the display associated with a context. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_GetMonitorFrequency (
   DSpContextReferenceConst inContext,
   Fixed *outFrequency
);
```

Parameters

inContext

A reference to a context for which you want to get the display frequency.

outFrequency

On return, the display frequency. The context must have been active for a reasonable amount of time (at least two seconds) in order to receive a correct value, because the value given by this parameter on return is calculated by timing the frame rate of the active context.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Declared In

`DrawSprocket.h`

DSpContext_GetState

Determines the current play state of a context. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_GetState (
   DSpContextReferenceConst inContext,
   DSpContextState *outState
);
```

Parameters

inContext

A reference to the context whose play state you want to get.

outState

On return, the play state of the context. Valid return values are `kDSpContextState_Active`, `kDSpContextState_Paused`, and `kDSpContextState_Inactive`. See "Play State Constants" (page 19) for more information.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpContext_GlobalToLocal

Translates a point in global coordinates into local coordinates for a context. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_GlobalToLocal (
   DSpContextReferenceConst inContext,
   Point *ioPoint
);
```

Parameters

inContext

A reference to the context whose local coordinates you want to translate into.

ioPoint

Takes a point in global coordinates. On return, contains the point in local coordinates.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpContext_IsBusy

Finds out whether a back buffer is available. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpContext_IsBusy (
   DSpContextReferenceConst inContext,
   Boolean *outBusyFlag
);
```

Parameters

inContext

A reference to the context associated with the desired back buffer.

outBusyFlag

On return, contains `true` if no back buffer is available, `false` if a back buffer is available.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
You can use this function to determine whether a call to `DSpContext_GetBackBuffer` (page 29) will block.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpContext_LocalToGlobal

Translates a point from a context's local coordinates into global coordinates. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_LocalToGlobal (
   DSpContextReferenceConst inContext,
   Point *ioPoint
);
```

Parameters

inContext

The context whose local coordinate system describes the point's coordinates.

ioPoint

Takes a point's local coordinates. On return, contains the point's global coordinates.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpContext_Queue

Queues a context you want to switch to. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_Queue (
   DSpContextReference inParentContext,
   DSpContextReference inChildContext,
   DSpContextAttributesPtr inDesiredAttributes
);
```

Parameters

inParentContext
The current active context.

inChildContext
The context you want to switch to.

inDesiredAttributes
A pointer to a context attributes structure that describes the context you want to switch to.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
Typically, you use this function to queue up contexts in addition to the one specified by the function `DSpContext_Reserve` (page 37). After you queue a context, you make it active by calling the function `DSpContext_Switch` (page 42). To release a queued context, you must call the function `DSpContext_Release` (page 37).

Calling `DSpContext_Queue` also determines whether the desired context switch is actually possible. For example, among other things, DrawSprocket will check to see that both contexts are on the same display. If the contexts are incompatible, this call returns an error.

Note that you can also use this function to modify attributes of the context to be switched to.

Version Notes
Introduced with DrawSprocket 1.7.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

DSpContext_Release

Releases a context you are finished using. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_Release (
   DSpContextReference inContext
);
```

Parameters

inContext

A reference to the context to be released. Releasing the context does not necessarily remove the blanking window from the corresponding display. All displays remain covered by the blanking window until all contexts have been released or put in an inactive play state.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Discussion

You must release the context whether it was reserved or queued.

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code

DrawSprocketTestOld

GlyphaIVOld

OpenGLMovieQT

Simple DrawSprocket

Declared In

`DrawSprocket.h`

DSpContext_Reserve

Reserves a context so that you can begin using it in your game. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpContext_Reserve (
   DSpContextReference inContext,
   DSpContextAttributesPtr inDesiredAttributes
);
```

Parameters

inContext

A reference to the context to reserve. When the context is reserved, it is in the inactive state. There will be no visible indication that the context has been reserved at this point. To enable your context, call `DSpContext_SetState` (page 39). The context will show up on the display once the context has been placed in the active state.

inDesiredAttributes

A pointer to an attributes structure that specifies the configuration you would like for the display when it is in the active or paused state. If you would like to override the attributes of the context, you may do so in the attributes structure. For example, if you ask for a 320x240x16 display but the closest match is a context that is 640x480x32, passing in your requested attributes when you reserve the context will cause the `DSpContext_GetBackBuffer` function to return a graphics pointer that refers to a 320x240x16 drawing environment.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Discussion

You should turn off features that you are not interested in when you reserve the context. For example, if the context supports page flipping (and you know this because you requested the actual capabilities of the context using `DSpContext_GetAttributes`), you can turn off the page-flipping bit in your desired attributes so that you will be assured of using software buffering.

You should only specify a back buffer bit depth different from the display bit depth when you absolutely must, as it is the worst case scenario for DrawSprocket and will result in a synchronous call to `CopyBits` to bring your back buffer to the display.

To release a reserved context, you must call the function `DSpContext_Release` (page 37).

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code

DrawSprocketTestOld

GlyphaIVOld

OpenGLMovieQT

Simple DrawSprocket

Declared In

`DrawSprocket.h`

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

DSpContext_SetCLUTEntries

Assigns one or more color entries to a color lookup table. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_SetCLUTEntries (
   DSpContextReference inContext,
   const ColorSpec *inEntries,
   UInt16 inStartingEntry,
   UInt16 inLastEntry
);
```

Parameters

inContext
The context whose color lookup table is to be modified.

inEntries
A pointer to an array of color specification records.

inStartingEntry
The (zero-based) index position in the color lookup table of the first entry to replace.

inLastEntry
The number of entries to replace.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Discussion

The `DSpContext_SetCLUTEntries` function allows you to change a range of entries in a color lookup table, for purposes such as color-table animation.

Because of video hardware limitations, the changes you make to a color table with this function may not take effect until the next vertical retrace. Nevertheless, this function attempts to execute asynchronously and return immediately, so your program can continue execution without having to wait for the changes to be made.

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code

DrawSprocketTestOld

Declared In

`DrawSprocket.h`

DSpContext_SetState

Sets the play state of a context. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpContext_SetState (
   DSpContextReference inContext,
   DSpContextState inState
);
```

Parameters

inContext

A reference to the context whose play state you want to set.

inState

A constant specifying the desired play state. Valid input values for this parameter are `kDSpContextState_Active`, `kDSpContextState_Paused`, and `kDSpContextState_Inactive`. See "Play State Constants" (page 19) for more information.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Discussion

In summary, you can make these choices:

- A context's initial play state is inactive. When all contexts for a display are set to `kDSpContextState_Inactive`, the display looks exactly as it does when the user is using their Macintosh normally: the monitor resolutions are set to the default, the menu bar is available, and so on.
- Set the play state to `kDSpContextState_Active` to use the display. In this state, the attributes of the context are used to change the display resolution, remove the menu bar, and so on. When at least one context is active, all the display devices in the system are covered by a blanking window. When a context is in the active state, the display is completely owned by the game.
- Set the play state to `kDSpContextState_Paused` to temporarily restore system adornments, while maintaining the attributes used by the context. This gives the user the opportunity to use the menus and switch to other applications. While the context is in the paused state, it is very important to call `DSpProcessEvent` to allow DrawSprocket to correctly handle events such as suspend or resume (see `DSpProcessEvent` (page 48)). Page flipping and double buffering are inactive in this state, and the context will be placed back at page 0 if page flipping was being used.

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code

DrawSprocketTestOld

GlyphaIVOld

OpenGLMovieQT

Simple DrawSprocket

Declared In

`DrawSprocket.h`

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

DSpContext_SwapBuffers

Draws a context's back buffer to the screen. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_SwapBuffers (
   DSpContextReference inContext,
   DSpCallbackUPP inBusyProc,
   void *inUserRefCon
);
```

Parameters

inContext

A reference to the context whose buffers are to be swapped. The function causes the invalid parts of the back buffer of the context specified in this parameter (or the entire back buffer, if its invalid-rectangle list is empty) to be drawn to the screen.

inBusyProc

A pointer to a callback function that performs any required pre-swap tasks. See `DSpCallbackProcPtr` (page 9) for more information about implementing this function.

inUserRefCon

A reference constant to be handed back by DrawSprocket when it calls the callback specified by the `inBusyProc` parameter.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Discussion

This function returns immediately, even if the buffer swap has not yet occurred. To determine when the next call to `DSpContext_GetBackBuffer` will not block, you can repeatedly call the function `DSpContext_IsBusy` (page 34) until it returns a value of `false`.

Before performing the buffer swap, DrawSprocket repeatedly calls an application-supplied callback function, pointed to by the `inBusyProc` parameter, to make sure that any constraints you impose are satisfied before the swap occurs. When DrawSprocket calls the callback routine, it passes the reference constant you passed to `DspContext_SwapBuffers` in the `refCon` parameter.

See the callback function under `DSpCallbackProcPtr` (page 9) for more information.

In a worst case scenario where the back buffer and the display have different bit depths, `DSpContext_SwapBuffers` immediately calls `CopyBits` to transfer the data. To avoid this, and to use the optimized DrawSprocket blitters, always insure that your back buffer and display bit depths are identical.

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code

DrawSprocketTestOld

GlyphalVOld

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Declared In
`DrawSprocket.h`

DSpContext_Switch

Switches display contexts. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpContext_Switch (
   DSpContextReference inOldContext,
   DSpContextReference inNewContext
);
```

Parameters

inOldContext
The current display context.

inNewContext
The display context to switch to.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
Calling this function switches the display context immediately without any intermediate switch to the default display mode. Note that switching contexts will kill any piggyback VBL routines attached to the context you are switching out.

If you did not queue the contexts you want to switch (by calling the function `DSpContext_Queue` (page 36)), `DSpContext_Switch` returns an error.

Version Notes
Introduced with DrawSprocket 1.7.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpFindBestContext

Finds the context that best matches the requirements you specify. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpFindBestContext (
   DSpContextAttributesPtr inDesiredAttributes,
   DSpContextReference *outContext
);
```

Parameters

inDesiredAttributes

A pointer to a context attributes structure describing the desired display characteristics of the context, such as display height and width, preferred pixel depth, and color capability. See `DSpContextAttributes` (page 12) for more information about this structure.

outContext

On return, a reference to the context that best meets or exceeds the specified attribute requirements, or `NULL` if no such context exists.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20). If no context meets the requirements you specified, the function returns `kDSpContextNotFoundErr`.

Discussion

Even if the call to `DSpFindBestContext` returns successtully, the game should check the attributes of the chosen context by calling the function `DSpContext_GetAttributes` (page 28). It is possible that the game may want to use attributes of the context that exceed those asked for. For example, the game may request a mode such as 320x200x8 but the best match is a 640x480x8 display; the game can adapt to a full screen mode once it is aware of the situation.

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code

DrawSprocketTestOld

GlyphaIVOld

OpenGLMovieQT

Declared In

`DrawSprocket.h`

DSpFindBestContextOnDisplayID

Determines the best context to use for a given display. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpFindBestContextOnDisplayID (
   DSpContextAttributesPtr inDesiredAttributes,
   DSpContextReference *outContext,
   DisplayIDType inDisplayID
);
```

Parameters

inDesiredAttributes

A pointer to a structure describing the desired attributes for the context. See DSpContextAttributes (page 12) for more information.

outContext

On return, outContext points to the context that best matches the desired attributes, or NULL if no such context exists.

inDisplayID

The ID of the display to check for contexts.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
You can obtain the display ID of a monitor by calling the Display Manager.

Special Considerations

Version Notes
Introduced with DrawSprocket 1.7.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
DrawSprocket.h

DSpFindContextFromPoint

Finds out which context contains a point given in global coordinates. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpFindContextFromPoint (
   Point inGlobalPoint,
   DSpContextReference *outContext
);
```

Parameters

inGlobalPoint

A point in global coordinates.

outContext

On return, a reference to the context that contains that point.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Discussion
If the user moves the mouse, the game needs to know which context contains it so that the global coordinates can be properly translated into local coordinates for the context.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpGetCurrentContext

Obtains a reference to the current display context for a given display. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpGetCurrentContext (
   DisplayIDType inDisplayID,
   DSpContextReference *outContext
);
```

Parameters

inDisplayID
The ID of the display whose context you want to obtain.

outContext
On return, `outContext` points to the current context in the given display.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Version Notes
Introduced with DrawSprocket 1.7.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpGetFirstContext

Obtains the first context in the list of contexts available for a specified display. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpGetFirstContext (
   DisplayIDType inDisplayID,
   DSpContextReference *outContext
);
```

Parameters

inDisplayID

The ID of the display whose context you desire. You can obtain the display ID by calling the Display Manager.

outContext

On return, a reference to the first context in the list of available contexts for the specified display. You cannot use this context with any function other than `DSpContext_GetAttributes`, `DSpContext_GetFlattendSize`, `DSpContext_Flatten`, and `DSpContext_GetDisplayID` unless you reserve it with `DSpContext_Reserve`.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
Using the function DSpGetFirstContext in combination with `DSpGetNextContext` (page 47) allows you to iterate over the list of contexts and choose one that best suits your needs. You may also have DrawSprocket find one for you with `DSpFindBestContext` or let the user select one by calling `DSpUserSelectContext`.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld

Declared In
`DrawSprocket.h`

DSpGetMouse

Obtains the global coordinates of the mouse position. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpGetMouse (
   Point *outGlobalPoint
);
```

Parameters

outGlobalPoint

On return, the global coordinates of the mouse position.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Special Considerations

Do not call at interrupt time.

Version Notes

Introduced with DrawSprocket 1.0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Declared In

`DrawSprocket.h`

DSpGetNextContext

Obtains the next context in a list of available contexts for a display. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpGetNextContext (
   DSpContextReference inCurrentContext,
   DSpContextReference *outContext
);
```

Parameters

inCurrentContext

A reference to a context in the list of contexts available for a display. This should be a reference that was just returned by `DSpGetFirstContext` or `DSpGetNextContext`. If this parameter contains the last context in the list, `DSpGetNextContext` returns an error.

outContext

On return, a reference to the next context in the list of available contexts.

Return Value

A result code. See "Game Sprockets Result Codes" (page 20).

Discussion

Using the function `DSpGetNextContext` in combination with `DSpGetFirstContext` (page 45) allows you to iterate over the list of contexts and choose one that best suits your needs. For example, you could have code such as the following:

```
DSpContextReference theContext;
theError = DSpGetFirstContext(theDisplayID, &theContext)
/* process the error */
while (theContext)
{
    /* process the context */
    /* get the next context */
    theError = DSpGetNextContext(theContext, &theContext)
    /* process the error */
}
```

You may also have DrawSprocket find a display context for you by calling `DSpFindBestContext` (page 42) or `DSpFindBestContextOnDisplayID` (page 43).

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld
OpenGLMovieQT

Declared In
`DrawSprocket.h`

DSpGetVersion

Determines the version of DrawSprocket installed on the host computer. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
NumVersion DSpGetVersion (
   void
);
```

Parameters

Return Value
The version number of DrawSprocket installed.

Discussion

Special Considerations

Version Notes
Introduced with DrawSprocket 1.7.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
OpenGLMovieQT
Simple DrawSprocket

Declared In
`DrawSprocket.h`

DSpProcessEvent

Passes system events through to DrawSprocket so that it can correctly handle events it must know about. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpProcessEvent (
   EventRecord *inEvent,
   Boolean *outEventWasProcessed
);
```

Parameters

inEvent

A pointer to the event to be passed to DrawSprocket.

outEventWasProcessed

On return, `true` if DrawSprocket processed the event; `false` if the event was not processed.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
Whenever your game receives a suspend or resume event, it must call the `DSpProcessEvent` function so that DrawSprocket can correctly set the system state for the process switch.

When DrawSprocket is suspended, it returns the display to the resolution mode it was in before your context's play state first became active. When DrawSprocket resumes, it restores the display to the resolution mode used by your context. However, it is your responsibility to update the contents of the display at this time.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
OpenGLMovieQT
Simple DrawSprocket

Declared In
`DrawSprocket.h`

DSpSetBlankingColor

Assigns a background color to the blanking window for all displays. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpSetBlankingColor (
   const RGBColor *inRGBColor
);
```

Parameters

inRGBColor

A pointer to the background color to use for the blanking window.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Discussion
The blanking color replaces the desktop and system adornments, such as the menu bar, for all display devices as long as any context is active.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld
Simple DrawSprocket

Declared In
`DrawSprocket.h`

DSpSetDebugMode

Keeps the screen and system resources visible during debugging. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpSetDebugMode (
   Boolean inDebugMode
);
```

Parameters

inDebugMode

Set this value to `true` if the desktop display is to remain visible, even after fading; `false` otherwise.

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
During development, if you drop into the debugger when the display has been faded out, you cannot fade the display back in so that you can see the debugger screen. Calling the `DSpSetDebugMode` function with the `inDebugMode` flag set to a value of `true` causes your program to enter a mode in which the blanking window is not drawn and every fade operation (either in or out) causes only a partial dimming and immediate restoration of the screen intensity. Calling this function with the `inDebugMode` flag set to a value of `false` ends the mode and resumes normal operation.

To make use of this function, you must call it before activating your context. Once the blanking window is in place, this function effects only gamma fades.

This function is ignored in nondebugging builds of DrawSprocket.

Note the when using debugging builds, you can also enter debug mode by placing a folder named `DSpDebugMode` in your application folder.

Version Notes
Introduced with DrawSprocket 1.0.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Declared In
`DrawSprocket.h`

DSpShutdown

Shuts down DrawSprocket. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

```
OSStatus DSpShutdown (
   void
);
```

Parameters

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
You must call this function before quitting the application.

Special Considerations
Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.
Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld
GlyphaIVOld
OpenGLMovieQT
Simple DrawSprocket

Declared In
`DrawSprocket.h`

DSpStartup

Initializes DrawSprocket. (Deprecated in Mac OS X v10.4. Use Quartz Display Services instead; see *Quartz Display Services Reference*.)

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

```
OSStatus DSpStartup (
   void
);
```

Parameters

Return Value
A result code. See "Game Sprockets Result Codes" (page 20).

Discussion
You must call this function before attempting to call any DrawSprocket functions (except for `DSpGetVersion` (page 48)).

Note that the debug version of DrawSprocket will notify you if you did not call the `DSpStartup` function.

Special Considerations

Do not call at interrupt time.

Version Notes
Introduced with DrawSprocket 1.0.

Availability
Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.4.

Related Sample Code
DrawSprocketTestOld

GlyphaIVOld

Simple DrawSprocket

Declared In
`DrawSprocket.h`

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Document Revision History

This table describes the changes to *Apple Game Sprockets Reference*.

Date	Notes
2006-07-13	Made minor formatting changes.
2006-07-24	Added information about deprecated functions.
2003-01-01	Moved legacy API to Game Sprockets Legacy Reference.

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Document Revision History

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

Index

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.

K

N

P

S

Legacy Document | 2006-07-13 | © 2006 Apple Computer, Inc. All Rights Reserved.