iDraw OCX Features :

- ActiveX component interface to DirectDraw3
- Simplifies DirectDraw Programming
- Allows access to DirectDraw from any OLE container - VisualBasic, Delphi, Visual C++ and many more

CompleteControl ™

What is iDraw

More About iDraw

iDraw Reference

System Requirements

iDraw Objects

Sample Applications

Examples

Contact Information

How to contact us

Obtaining Technical Support

Registration

Registering your purchase

Licensing and Distribution

License Agreement

Distributing iDraw

Un-Installing iDraw

iDraw is a 32 bit ActiveX Control wrapper around the Microsoft® DirectDraw API. DirectDraw is a powerful technology that allows you to create graphics intensive applications for Windows.

If you are familiar with DirectX you will know it is very difficult to program from environments such as Visual Basic. We have created iDraw to provide easy access to DirectDraw from virtually any programming environment.

iDraw is designed to give you complete access to the DirectDraw API. You should to be familiar with DirectDraw before using iDraw OCX. This on-line help does provide you with a complete iDraw reference however DirectDraw programming concepts are not covered here in depth – please consult the DirectDraw documentation from Microsoft.

iDraw is just one of many software components from our CompleteControl range of products, including other DirectX components - i3D, iSound, iPlay and iInput.

iDraw is supported under the following 32 bit Microsoft® Windows operating systems on Intel® architecture machines.

Windows 95

Windows NT Server and Workstation 4.0

and future versions of these operating systems.

Note:

The current version of iDraw is not supported on beta versions of the above operating systems, Win32s or on non-Intel architecture implementations of Microsoft Windows.

For information about future versions and updates please register with Imagine IT Ltd

To register your copy of iDraw please print out this page and send it to us after completing the details. Alternatively you can also email the required information to us at: registration@imagineit.co.uk

Registered users automatically receive information about updates, new releases and future products from Imagine IT and are eligible for technical support.

Product: iDraw.OCX 2.0

Last Name:

First Name:

Company Name:

Address:

Telephone Number:

Fax Number:

E-Mail Address:

Web URL:

Date of Purchase:

Purchased From:

---

please send to:

Imagine IT Limited, 3rd Floor, Hygeia Building, 66 College Road, Harrow HA1 1BE, United Kingdom

or email to:

registration@imagineit.co.uk

iDraw is designed and developed by Imagine IT Limited.

Imagine IT specialises in object technologies and component software development.

Apart from creating great components we also help customers develop their own line-of-business objects and applications.

Contact us for more information about our products and services :

Write to us at:

Imagine IT Limited
3rd Floor,
Hygeia Building
66 College Road
Harrow, HA1 1FD
United Kingdom

Telephone:

from within the United Kingdom:	0181 324 1240
from other countries:	international access code + 44 + 181 324 1240
from other countries:	international access code + 44 + 181 324 1752

Fax:

from within the United Kingdom:	0181 324 1752
from other countries:	international access code + 44 + 181 324 1752

Email:

for general inquiries:	info@imagineit.co.uk
for technical support inquiries:	suport@imagineit.co.uk
for registration:	registration@imagineit.co.uk

World Wide Web:

find us at:	www.imagineit.co.uk

For the fastest possible response please direct all technical support enquiries via email to support@imagineit.co.uk.

Or fax your enquiry marked for the attention of “Technical Support”.

If you do not have access to email or fax then please call us at the above telephone number and ask for “Technical Support”

NOTE: We can only provide technical support to registered users.

IMPORTANT

THE USE OF THIS SOFTWARE IS SUBJECT TO THE TERMS OF THE LICENSE AGREEMENT PRINTED BELOW. PLEASE READ THE LICENSE AGREEMENT CAREFULLY. IF YOU DO NOT AGREE WITH ALL OF THE TERMS, THEN YOU SHOULD NOT USE THE SOFTWARE IN ANY WAY WHATSOEVER INCLUDING COPYING OR DISTRIBUTING THE SOFTWARE OR ANY PART OF THE SOFTWARE TO ANY THIRD PARTY. IF YOU USE THE SOFTWARE IN ANY WAY WHATSOEVER IT WILL BE DEEMED TO INDICATE YOUR ACCEPTANCE OF ALL THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT.

THE AGREEMENT

THIS DOCUMENT IS A LEGAL AGREEMENT BETWEEN IMAGINE IT LIMITED, UNITED KINGDOM, HEREAFTER REFERRED TO AS "IMAGINE IT" AND THE USER OF THE SOFTWARE HEREIN REFERRED TO AS "LICENSEE"

DEFINITIONS

"SOFTWARE" MEANS THE SOFTWARE KNOWN AS "IDRAW OCX" AND ALL ITS COMPONENTS, PARTS AND DOCUMENTATION, WHICH HAVE BEEN DEVELOPED BY IMAGINE IT

"END USER APPLICATION" MEANS ANY APPLICATION DEVELOPED WITH THE HELP OF OR BY USING THE SOFTWARE

PURPOSE

THE PURPOSE OF THIS AGREEMENT IS TO DEFINE THE RELATIONSHIP BETWEEN IMAGINE IT AND LICENSEE, THE TERMS, CONDITIONS, AND LIMITATIONS OF USE, THE RIGHTS OF BOTH PARTIES AND THE LIMITATION OF LIABILITY ARISING FROM THE USE OF THE SOFTWARE

EFFECTIVE DATE

THIS AGREEMENT IS DEEMED TO HAVE BEEN MADE EFFECTIVE ON THE FIRST DATE AT WHICH LICENSEE OPENS THE PACKAGE CONTAINING THE SOFTWARE AND ITS ASSOCIATED DOCUMENTATION, OR IF THE SOFTWARE IS DISTRIBUTED ELECTRONICALLY, THE FIRST DATE ON WHICH THE LICENSEE OBTAINS THE SOFTWARE

GRANT OF LIMITED LICENSE

I. EVALUATION LICENSE

IF THE LICENSEE HAS NOT PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN IMAGINE IT GRANTS THE LICENSEE A NON EXCLUSIVE, NON TRANSFERABLE, PERSONAL LICENSE TO USE THE SOFTWARE AND ITS DOCUMENTATION FOR EVALUATION PURPOSES ONLY AND FOR NO OTHER PURPOSE. THE LICENSE DOES NOT PERMIT THE DEVELOPMENT AND DISTRIBUTION OF ANY END USER APPLICATIONS USING THE SOFTWARE

II. FULL DEVELOPMENT LICENSE

IF THE LICENSEE HAS PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN IMAGINE IT GRANTS THE LICENSEE A NON EXCLUSIVE, NON TRANSFERABLE, PERSONAL LICENSE TO USE THE SOFTWARE AND ITS DOCUMENTATION FOR THE PURPOSE OF DEVELOPING AND DISTRIBUTING ANY NUMBER OF END USER APPLICATIONS AND TO COPY AND DISTRIBUTE ANY PARTS OF THE SOFTWARE DEFINED BELOW UNDER REDISTRIBUTABLE COMPONENTS TOGETHER WITH AND AS PART OF THE END USER APPLICATION, SUBJECT TO ALL THE TERMS AND CONDITIONS AND RESTRICTIONS IN THIS AGREEMENT.

RESTRICTIONS

A USER OF ANY END USER APPLICATIONS MAY NOT FURTHER USE PARTS OF THE SOFTWARE FOR SOFTWARE DEVELOPMENT, COPYING OR DISTRIBUTION. THE LICENSEE MUST ENFORCE THIS IN A SEPARATE AGREEMENT WITH THE USER OF ANY END USER APPLICATIONS.

THE LICENSEE MAY USE ONLY ONE COPY OF THE SOFTWARE AT ANY TIME ON ONE IBM PC OR COMPATIBLE SYSTEM.

ADDITIONAL PRODUCT KEYS FOR THE SOFTWARE MUST BE PURCHASED IF IT IS REQUIRED TO BE USED IN A MULTI USER NETWORKED ENVIRONMENT IN QUANTITIES OF ONE KEY FOR EACH PERSON HAVING ACCESS TO AND USING THE SOFTWARE.

THE LICENSEE MAY MAKE A SINGLE BACKUP COPY OF THE SOFTWARE.

THE LICENSEE MAY NOT USE, COPY, MODIFY, REVERSE ENGINEER, DISASSEMBLE, SELL, TRANSFER, HIRE, LEND OR OTHERWISE DISTRIBUTE THE SOFTWARE OR ANY OF ITS DOCUMENTATION OR COMPONENTS IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS AGREEMENT.

THE END USER APPLICATION MUST NOT BE ANOTHER DEVELOPMENT TOOL DESIGNED TO BE USED FOR CREATING OTHER END USER APPLICATIONS USING THE SOFTWARE. THE LICENSEE MUST ENFORCE THIS IN A SEPARATE AGREEMENT WITH THE USER OF THE END USER APPLICATION.

THE LICENSEE HAS NO RIGHTS TO THE ORIGINAL SOURCE CODE OF THE SOFTWARE WHATSOEVER.

THE LICENSEE MUST INFORM IMAGINE IT OF THE NATURE OF EACH AND EVERY NEW END USER APPLICATION IT DEVELOPS WITH THE USE OF THE SOFTWARE.

THE LICENSEE SHALL COMPLY WITH ALL LOCAL AND INTERNATIONAL LAWS AND EXPORT / IMPORT REGULATIONS WHEN DISTRIBUTING PARTS OF THIS SOFTWARE AS PROVIDED FOR BELOW ALONG WITH ANY END USER APPLICATIONS

REDISTRIBUTABLE COMPONENTS

THE SOFTWARE, BY ITS NATURE, INCLUDES COMPONENTS WHICH ARE REQUIRED TO BE DISTRIBUTED WITH ANY END USER APPLICATIONS IN ORDER FOR THAT END USER APPLICATION TO FUNCTION.

I. EVALUATION LICENSE

IF THE LICENSEE HAS NOT PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN THERE ARE NO DISTRIBUTABLE COMPONENTS. THE LICENSEE SHALL NOT DISTRIBUTE ANY COMPONENTS OF THE SOFTWARE

II. FULL DEVELOPMENT LICENSE

IF THE LICENSEE HAS PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THENTHE FOLLOWING FILES DEFINE THE COMPONENTS WHICH THE LICENSEE MAY DISTRIBUTE PROVIDED THAT THEY ARE DISTRIBUTED AS PART OF THE END USER APPLICATION AND PROVIDED THAT THEY ARE DISTRIBUTED WITHOUT ANY MODIFICATION AND ARE COPIED FROM THE ORIGINAL DISKS OR CDS. ALL FILES LISTED BELOW MUST BE DISTRIBUTED WITH THE END USER APPLICATION.

1. **IDRAW.OCX**
2. **IDRAW.RTL**

NO OTHER FILES SUPPLIED WITH THE SOFTWARE MAY BE DISTRIBUTED WITH THE END USER APPLICATION.

THE SOFTWARE BY ITS NATURE ALSO REQUIRES SOME FILES WHICH ARE DISTRIBUTED BY MICROSOFT(R) TO BE PRESENT ON THE END USER'S COMPUTER FOR THE END USER APPLICATION TO FUNCTION. THIS LICENSE AGREEMENT DOES NOT IN ANY WAY PROVIDE FOR LICENSEE TO DISTRIBUTE THESE COMPONENTS WITH THE END USER AGREEMENT. THE LICENSEE MUST VERIFY THEIR RIGHTS TO DISTRIBUTE THESE COMPONENTS, SEPARATELY WITH MICROSOFT(R) BEFORE DISTRIBUTING THESE COMPONENTS WITH THE END USER APPLICATION

1. **MFC42.DLL**
1. **OLEPRO32.DLL**
2. **REGSVR32.EXE**
3. **DDRAW.DLL**

TECHNICAL SUPPORT

ONLY THOSE LICENSEES WHO HAVE PURCHASED A FULL PRODUCT KEY AND REGISTERED WITH IMAGINE IT BY FULLY COMPLETING AND RETURNING ALL THE INFORMATION ASKED FOR IN THE REGISTRATION FORM OR RELEVANT SECTION IN THE ON-LINE HELP DOCUMENTATION ARE QUALIFIED FOR RECEIVING TECHNICAL SUPPORT HELP ON THE SOFTWARE.

TECHNICAL SUPPORT IS RESTRICTED TO ANSWERING QUESTIONS ABOUT THE USE OF THE SOFTWARE OR ERRORS IN THE SOFTWARE IF ANY.

TECHNICAL SUPPORT DOES NOT INCLUDE DEBUGGING AND OTHER PROBLEM SOLVING TASKS ASSOCIATED WITH THE DEVELOPMENT OF ANY END USER APPLICATIONS.

USERS OF THE END USER APPLICATION ARE NOT ELIGIBLE FOR OBTAINING TECHNICAL SUPPORT FROM IMAGINE IT UNDER ANY CIRCUMSTANCES.

THIS SERVICE WILL BE PROVIDED ACCORDING TO IMAGINE IT'S TECHNICAL SUPPORT POLICIES FROM TIME TO TIME, AND IS LIMITED TO SUCH TIME AS IMAGINE IT SUPPORTS THE SOFTWARE OR OWNS THE SOFTWARE.

TECHNICAL SUPPORT MAY BE PROVIDED BY MEANS OF E-MAIL, FAX, BULLETIN BOARDS, ON-LINE SERVICES, WRITTEN CORRESPONDENCE OR TELEPHONE AT IMAGINE IT'S SOLE DISCRETION.

OWNERSHIP

IMAGINE IT SHALL REMAIN THE OWNERS OF ALL RIGHTS, TITLE AND INTEREST IN THE SOFTWARE

THIS LICENSE DOES NOT CONFER ANY OWNERSHIP RIGHTS TO THE LICENSEE

THE LICENSEE AGREES NOT TO USE IMAGINE IT'S NAME IN ANY WAY OR FORM

THE LICENSEE AGREES NOT TO IMPLY THAT ANY END USER APPLICATIONS IT DEVELOPS ARE EITHER APPROVED OR OTHERWISE SANCTIONED BY IMAGINE IT

THE LICENSEE AGREES TO CARRY A COPYRIGHT NOTICE IN THE ABOUT BOX OR HELP FILE OF EACH AND EVERY COPY OF ANY END USER APPLICATIONS IT DEVELOPS WITH THE HELP OF THIS SOFTWARE, WITH THE FOLLOWING WORDS :-

"PORTIONS OF THIS APPLICATION PROVIDED BY IMAGINE IT LIMITED, UNITED KINGDOM, COPYRIGHT (C) 1996-1997 IMAGINE IT LIMITED. ALL RIGHTS RESERVED"

TERMINATION

THIS AGREEMENT SHALL BE TERMINATED IN WRITING BY IMAGINE IT AT ANY TIME IF THE LICENSEE BREACHES ANY TERMS AND CONDITIONS CONTAINED IN THIS AGREEMENT OR CARRIES OUT ANY ACTIONS EXPRESSLY PROHIBITED IN THIS AGREEMENT, AND CONTINUES TO BE IN BREACH FOR 30 DAYS AFTER WRITTEN NOTICE IS GIVEN TO THE LICENSEE BY IMAGINE IT OR ITS REPRESENTATIVES.

UPON TERMINATION LICENSEE SHALL RETURN ALL COPIES OF THE SOFTWARE, ITS COMPONENTS AND DOCUMENTATION TO IMAGINE IT.

ALL RIGHTS GRANTED TO THE LICENSEE IN THIS AGREEMENT SHALL CEASE UPON TERMINATION OF THIS AGREEMENT EXCEPT FOR THOSE APPLYING TO ANY COPIES OF ANY END USER APPLICATIONS ALREADY PROPERLY DISTRIBUTED AND LICENSED WITHIN THE TERMS OF THIS AGREEMENT, PRIOR TO TERMINATION OF THIS AGREEMENT.

CONFIDENTIALITY

THE LICENSEE ACKNOWLEDGES THAT THE SOFTWARE AND ITS DOCUMENTATION AND DESIGN CONSTITUTE CONFIDENTIAL AND PROPRIETARY INFORMATION BELONGING TO IMAGINE IT.

THE LICENSEE AGREES NOT TO DISCLOSE ANY OF THIS INFORMATION TO OTHER THIRD PARTIES EXCEPT FOR ITS EMPLOYEES WHO ARE ENGAGED IN USING THE SOFTWARE AND ARE BOUND BY THESE SAME TERMS THROUGH A SEPARATE WRITTEN AGREEMENT WITH LICENSEE, OR EXCEPT WHERE REQUIRED TO DISCLOSE THE INFORMATION BY LAW.

WARRANTIES AND DISCLAIMERS

THE SOFTWARE IS PROVIDED "AS IS". IMAGINE IT DOES NOT WARRANT THAT THE SOFTWARE WILL OPERATE WITHOUT ERRORS, OR THAT IT WILL MEET ALL OF THE LICENSEE'S REQUIREMENTS.

ALL WARRANTIES, EXPRESSED OR IMPLIED, ARE EXCLUDED FROM THIS AGREEMENT INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE BUT NOT EXCLUDING ANY STATUTARY RIGHTS PROVIDED FOR BY LAW.

IMAGINE IT SHALL NOT BE LIABLE FOR ANY DAMAGES INCLUDING BUT NOT LIMITED TO INCIDENTAL OR CONSEQUENTIAL LOSS, LOSS OF PROFIT, LOSS OF SALES, INJURY, DEATH, LOSS OF OR DAMAGE TO PROPERTY OR ANY OTHER LOSS ARISING DIRECTLY OR INDIRECTLY FROM THE USE OF THE SOFTWARE.

IMAGINE IT'S SOLE REMEDY TO THE LICENSEE, AT IMAGINE IT'S SOLE DISCRETION SHALL BE LIMITED TO EITHER THE REPAIR OR REPLACEMENT OF FAULTY SOFTWARE OR THE REPAYMENT OF THE LICENSE FEE PAID BY THE LICENSEE UPON RETURN OF ALL COPIES OF THE SOFTWARE. IMAGINE IT'S TOTAL LIABILITY SHALL NOT EXCEED THE AMOUNTS PAID BY LICENSEE FOR THE PURCHASE OF THIS LICENSE.

APPLICABLE LAW

THIS AGREEMENT SHALL BE GOVERNED BY TO THE LAWS OF ENGLAND.

END

In order to use **iDraw** in your applications you need to distribute the OCX with your application. In addition you also need to ship some Microsoft® shared libraries which the OCX uses at run-time.

Before distributing any files which have been shipped with this product please read the license agreement and ensure that you comply with all its terms and conditions. Note that you do not have a license to distribute any Microsoft® files as part of this license and you should ensure that you have the appropriate licenses from Microsoft®. You should not distribute these third party components from the disks supplied with this product.

Shipping iDraw

To ship **iDraw** to your customers you must ship the following files - you may NOT ship any other files belonging to this product under any circumstances.

iDraw.OCX

iDraw.RTL

You should install these files on the target system in a separate sub directory below the system's Windows Directory, for example :-

<WindowsDirectory> \ iDraw \ iDraw.OCX

After installing these files and the other Microsoft® files identified below, you should register iDraw on the target system,. By running the following command :-

REGSVR32.EXE /s iDraw.OCX

Shipping Microsoft® Components

The following files are required on the target system :-

1. MFC40.DLL
2. OLEPRO32.DLL
3. REGSVR32.EXE
4. DDRAW.DLL

These should be installed in the target system's Windows System Directory but only if these files are either not already installed or they are a later version than those already on the target system.

When you install and register a control, you should also register OLEPRO32.DLL. Using the following command :-

REGSCR32.EXE /s OLEPRO32.DLL

Perform this registration step only if you need to install OLEPRO32.DLL. If the DLL is installed already, you should assume that it has been registered.

You should also register MFC40.DLL. Unlike OLEPRO32.DLL, you should always register this DLL, even if it is already installed. To register this DLL run the following command :-

REGSVR32.EXE /s MFC40.DLL

To UnInstall **iDraw** from your development system please follow these instructions :-

1. Un-Register the OCX by running the following command

REGSVR32.EXE /u iDraw.OCX

2. Run the Add / Remove programs applet from the Control Panel and select the **iDraw** component to be removed.

iDraw consists of a number of objects that provide access to the core DirectDraw functions. These objects are modelled largely around the DirectDraw interfaces.

Each iDraw object presents a number of properties and methods which relate to DirectDraw interfaces and functions.

In addition we have also created some new objects that will make your programming task easier.

The iDraw object provides the main interface to DirectDraw. You access all other objects via iDraw methods and properties. iDraw is synonymous with the DirectDraw IDirectDraw2 interface.

iDraw Properties

iDraw Methods

The iSurface object is synonymous with the DirectDraw IDirectDrawSurface2 interface. First create an iSurface object using the iDraw::CreateSurface method and then use the following properties and methods.

iSurface Properties

iSurface Methods

The iClipper object is synonymous with the DirectDraw IDirectDrawClipper interface. First create an iClipper object using the iDraw::CreateClipper method and then use the following properties.

iClipper Properties

The iPalette object is synonymous with the DirectDraw IDirectDrawPalette interface. First create an iPalette object using the iDraw::CreatePalette method and then use the following properties and methods.

iPalette Methods

The iBltFx object is synonymous with the DirectDraw DDBLTFX structure. Create an iBltFx object using the iDraw::CreateBltFx method.

iBltFx Properties

The iColorKey object is synonymous with the DirectDraw DDCOLORKEY structure. Create an iColorKey object using the iDraw::CreateColorKey method.

iColorKey Properties

The iCaps object is synonymous with the DirectDraw DDCAPS structure. Create an iCaps object using the iDraw::CreateCaps method.

iCaps Properties

The iOverlayFx object is synonymous with the DirectDraw DDOVERLAYFX structure. Create an iOverlayFx object using the iDraw::CreateOverlayFx method.

iOverlayFx Properties

The iPixelFormat object is synonymous with the DirectDraw DDPIXELFORMAT structure. Create an iPixelFormat object using the iDraw::CreatePixelFormat method.

iPixelFormat Properties

The iSurfaceDesc object is synonymous with the DirectDraw DDSURFACEDESC structure. Create an iSurfaceDesc object using the iDraw::CreateSurfaceDesc method.

iSurfaceDesc Properties

iSurfaceDesc Methods

The iDevice object is special to iDraw. It provides information about display devices and can be used to enumerate display devices on the target system.

iDevice Properties

The iColorTable object is special to iDraw. It provides the means to specify color palette information. Use this object in place of the PALETTEENTRY structures required by DirectDraw.

iColorTable Properties

DeviceList[nIndex]	List of devices enumerated by the EnumDevices() method
DisplayModeList[nIndex]	List of display modes enumerated by the EnumDisplayModes() method
DisplayMode	The current display mode
FourCCList[nIndex]	List of FourCC codes enumerated by the EnumFourCC() method
FreeVidMem[SCaps]	Amount of display memory currently free for the specified type of surface
GDISurface	The surface memory that is being treated as the primary surface by the GDI.
HALCaps	Hardware capabilities of the device driver
HELCaps	Hardware Emulation Layer capabilities of DirectDraw
LastErrorCode	The last error code
LastErrorString	The last error string
MonitorFrequency	The frequency of the monitor
ScanLine	The line that is currently being drawn on the monitor.
SurfaceList[nIndex]	A surface in the list of enumerated surfaces
TotalVidMem[Scaps]	Total amount of display memory available for the specified type of surface
VerticalBlankStatus	The status of the vertical blank.

See Also

iDraw Methods

Use this property to retrieve the hardware capabilities of the device driver

Syntax

controlname.**HALCaps**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

iCaps Object

Remarks

This property can be used to determine the capabilities of the display device driver.

VB Example

```
DIM oDeviceCap AS Object
Set oDeviceCap = iDRAW1.HALCaps
```

Direct Draw Compatibility

This property is equivalent to the first parameter of IDirectDraw2::GetCaps()

Use this property to retrieve the capabilities of the hardware emulation layer (HEL) of the device driver

Syntax

controlname.**HELCaps**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

iCaps Object

Remarks

This property can be used to determine the capabilities of the HEL which provides software-based emulation of features that are not present in hardware.

VB Example

```
DIM oDeviceCap AS Object
Set oDeviceCap = iDRAW1.HELCaps
```

Direct Draw Compatibility

This property is equivalent to the second parameter of IDirectDraw2::GetCaps()

Use this property to get the display mode

Syntax

controlname.**DisplayMode**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

iSurfaceDesc Object

Remarks

This property can be used to retrieve the current display mode. The information returned must not be used to retore the display mode during clean-up.

VB Example

```
DIM oSurfaceDesc AS Object
Set oSurfaceDesc = iDRAW1.DisplayMode
```

Direct Draw Compatibility

This property is equivalent to IDirectDraw2::GetDisplayMode()

Use this property to retrieve the GDI surface

Syntax

controlname.**GDISurface**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

iSurface Object

Remarks

This property can be used to retrieve the surface object that currently represents the primary surface.

VB Example

```
DIM oGDISurface AS Object
Set oGDISurface = iDRAW1.GDISurface
```

Direct Draw Compatibility

This property is equivalent to IDirectDraw2::GetGDISurface().

Use this property to retrieve the frequency of the monitor.

Syntax

controlname.**MonitorFrequency**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

long

Remarks

The frequency value is returned in Hz multiplied by 100, for example, 10Hz is returned as 1000.

Direct Draw Compatibility

This property is equivalent to IDirectDraw2::GetMonitorFrequency()

Use this property to retrieve the scanline.

Syntax

controlname.**ScanLine**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

long

Remarks

This property can be used to get the scan line that is in the process of being drawn on the monitor.

Direct Draw Compatibility

This property is equivalent to IDirectDraw2::GetScanLine()

Use this property to get the status of the vertical blank.

Syntax

controlname.**VerticalBlankStatus**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

bool

Remarks

Returns TRUE if the display vertical blank is occuring, FALSE otherwise.

Use the WaitForVerticalBlank method to synchronize with the vertical blank.

Direct Draw Compatibility

This property is equivalent to IDirectDraw2::GetVerticalBlankStatus()

Use this property to retrieve the last error code

Syntax

controlname.**LastErrorCode**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

long

Remarks

This error code for the last error that occured. Use the LastErrorString property to get the equivalent error message

Direct Draw Compatibility

There is no Direct Draw equivalent of this property

Use this property to retrieve the last error string

Syntax

controlname.**LastErrorString**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

string

Remarks

A verbose error message for the last error that occurred.

Direct Draw Compatibility

There is no Direct Draw equivalent of this property

Use this property to retrieve a device from the list of enumerated devices

Syntax

controlname.**DeviceList [nIndex]**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

iDevice Object

Remarks

This property can be used to retrieve a device object from the list of devices enumerated by the EnumDevices method. See the iDevice properties for more information.

Direct Draw Compatibility

There is no Direct Draw equivalent of this property

Use this property to retrieve a display mode from the list of display modes enumerated

Syntax

controlname.**DisplayModeList [nIndex]**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

iSurfaceDesc Object

Remarks

This property can be used to retrieve a surface descriptor object from the list of display modes enumerated by the EnumDisplayModes method

VB Example

```
DIM oSurfaceDesc AS Object
n = iDARW1.EnumDisplayModes  ‘enumerate the display modes & get the number of modes n
Set oSurfaceDesc = iDRAW1.DisplayModeList [ n-1 ]  ‘ get the last display mode
```

Direct Draw Compatibility

There is no Direct Draw equivalent of this property

Use this property to retrieve a surface from the list of surfaces enumerated

Syntax

controlname.**SurfaceList [nIndex]**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

iSurface Object

Remarks

Gets a surface from the list of surfaces enumerated using the EnumSurfaces method

VB Example

```
DIM oSurface AS Object
n = iDARW1.EnumSurfaces   ‘enumerate the surfaces & get the number of surfaces n
Set oSurface = iDRAW1.SurfaceList [ n-1 ]   ‘ get the last surface
```

Direct Draw Compatibility

There is no Direct Draw equivalent of this property

Use this property to retrieve a FourCC code from the list of FourCC codes enumerated

Syntax

controlname.**FourCCList [nIndex]**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

long

Remarks

Gets a FourCC code from the list of FourCC codes enumerated by the EnumFourCC method

VB Example

```
n = iDARW1.EnumFourCC   ‘enumerate the FourCC Codes
i = iDRAW1.FourCCList [ n-1 ]   ‘ get the last FourCC Code
```

Direct Draw Compatibility

There is no Direct Draw equivalent of this property

The amount of display memory currently free for a given surface

Syntax

controlname.**FreeVidMem[SCaps]**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

long

Remarks

Gets the free video memory.

SCaps is an iSurfaceDesc object that specifies the hardware capabilities of the surface for which the free video memory is being determined. The iSurfaceDesc object encapsulates the DirectDraw DDSURFACEDESC structure, which contains information about the surface like its type, colorkeys, pixel format and its capabilities. This value is can only be approximated since the amount of free display memory keeps changing with the creation and release of surfaces.

Direct Draw Compatibility

This property is equivalent to the third parameter of IDirectDraw2::GetAvailableVidMem()

The total amount of available display memory for a given surface

Syntax

controlname.**TotalVidMem[SCaps]**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Type

long

Remarks

Gets the total video memory.

SCaps is an iSurfaceDesc object that specifies the hardware capabilities of the surface for which the total amount of available video memory is being determined. The iSurfaceDesc object encapsulates the DDSURFACEDESC structure in Direct Draw, which contains information about the surface like its type, colorkeys, pixel format and its capabilities.

Direct Draw Compatibility

This property is equivalent to the second parameter of IDirectDraw2::GetAvailableVidMem()

CloseDevice	Closes all open display devices
Compact	At present this method is only a stub; it has not yet been implemented.
CreateBltFx	Creates an iBltFx object
CreateCaps	Creates an iCaps object
CreateClipper	Creates an iClipper object
CreateColorKey	Creates an iColorKey object
CreateColorTable	Creates an iColorTable object
CreateDevice	Creates an iDevice object
CreateOverlayFx	Creates an iOverlayFx object
CreatePalette	Creates an iPalette object
CreatePixelFormat	Creates an iPixelFormat object
CreateSurfaceDesc	Creates an iSurfaceDesc object
CreateSurface	Creates an iSurface object
DuplicateSurface	Duplicates an iSurface object.
EnumDevices	Enumerates all device drivers installed on the system.
EnumDisplayModes	Enumerates all the display modes
EnumFourCC	Enumerates all of the existing FourCC codes
EnumSurfaces	Enumerates all of the existing or possible surfaces that meet the search criterion specified.
FlipToGDISurface	The surface that GDI writes to is made the primary surface.
OpenDevice	Opens a specified display device
RestoreDisplayMode	Restores the display mode to what it was before it was set using SetDisplayMode.
SetCooperativeLevel	Specifies the top-level behaviour of the application.
SetDisplayMode	Sets the display mode of the display-device.
WaitForVerticalBlank	Synchronizes the application with the vertical-blank interval.

See Also

iDraw Properties

For the current version of iDraw this method remains unimplemented.

Syntax

controlname.**Compact()**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

None

Remarks

This method gathers all the free surface memory fragments into one single block. Ensure that the cooperative level has been set to exclusive and that no other operation is in progress before calling this method.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::Compact()

Create an iClipper object

Syntax

controlname.**CreateClipper()**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iClipper Object

Parameters

None

Remarks

Use this method to create an iClipper object. Set the clipper properties as appropriate and then assign the clipper object to an iSurface.

VB Example

```
DIM oClipper as Object
DIM oSurface1 as Object

‘assume oSurface1 (an iSurface) is created

Set oClipper = iDRAW1.CreateClipper ‘ create a clipper object
oClipper.HWnd = me.hWnd ‘set the form window to provide the clipping information
oSurface1.Clipper = oClipper     ‘attach the clipper to a surface
```

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::CreateClipper()

Duplicates an iSurface object.

Syntax

controlname.**DuplicateSurface(**Object SourceSurface**)**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iSurface Object

Parameters

SourceSurface	The iSurface object to be duplicated

Remarks

This method returns an iSurface object that shares the same memory as that of the iSurface object passed as parameter.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::DuplicateSurface()

Use GDI as the primary surface.

Syntax

controlname.**FlipToGDISurface()**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

None

Remarks

A page-flipping application can call this method to ensure that the display memory being used by the GDI becomes the primary surface.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::FlipToGDISurface ()

See Also

GDISurface property

Restores the display mode to what it was before calling the SetDisplayMode method.

Syntax

controlname.**RestoreDisplayMode()**

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

None

Remarks

Ensure that the cooperative level is set to exclusive before using this method.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::RestoreDisplayMode ()

See Also

EnumDisplayModes method

SetCooperativeLevel method

SetDisplayMode method

Set the top-level behavior of the application.

Syntax

controlname.**SetCooperativeLevel** (OLE_HANDLE hWnd, long lFlags)

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

hWnd	Window handle used for the application.
lFlags	The preferred flags

Remarks

When exclusive mode is set, it prevents other applications from changing the display mode or the palette.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::SetCooperativeLevel ()

See Also

EnumDisplayModes method

SetDisplayMode method

Change the display mode.

Syntax

controlname.**SetDisplayMode** (long lWidth, long lHeight, long lBPP, long lRefreshRate, long lFlags)

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

lWidth	Width of the new mode
lHeight	Height of the new mode
lBPP	Bits per pixel (bpp) of the new mode
lRefreshRate	Refresh rate of the new mode
lFlags	This parameter is currently not used and must be set to 0.

Remarks

Ensure that you have set the cooperative level to exclusive before using this method.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::SetDisplayMode ()

See Also

EnumDisplayModes method

SetCooperativeLevel method

RestoreDisplayMode method

DisplayMode property

Synchronize with the vertical-blank interval.

Syntax

controlname.**WaitForVerticalBlank** (long lFlags)

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

lFlags	Specifies whether to wait for the vertical blank to start or to end

Remarks

This method returns when the vertical blank interval begins or the vertical blank interval ends depending on the value of lFlags.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::WaitForVerticalBlank ()

See Also

ScanLine property

VerticalBlankStatus property

Opens a display device

Syntax

controlname.**OpenDevice** (Object Device)

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

Device	The device to open. NULL to open the default device.

Remarks

You must call this method to initialise and open a display device. To open the default device (for example if there is only one display device) set the Device parameter to NULL. Otherwise use EnumDevices() to enumerate the devices and then pass one of the devices in DeviceList as the parameter.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

VB Example

```
DIM oDevice as Object

' to open the default display device :
Set oDevice = NULL
iDRAW1.OpenDevice ( oDevice )  ' open the default display

' or to open an enumerated display device:
n = iDRAW1.EnumDevices()  'enumerate all the devices, n = number of devices found
if (n > 0) then
       iDRAW1.OpenDevice ( iDRAW1.DeviceList [ 0 ] ) 'open the first device
endif
```

Direct Draw Compatibility

This method is equivalent to DirectDrawCreate ()

Closes all open display devices

Syntax

controlname.**CloseDevice** ()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

long

Parameters

None

Remarks

This method closes all devices that have been opened by OpenDevice()

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

There is no equivalent method in Direct Draw

Create an iSurface object

Syntax

controlname.**CreateSurface**(object SurfaceDesc)

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iSurface Object

Parameters

SurfaceDesc	The type of surface to be created

Remarks

Use this method to create a DirectDraw Surface. You must first obtain an iSurfacedesc object and set its properties as required. Then pass this iSurfaceDesc object as a parameter to CreateSurface().

VB Example

```
DIM oSD as Object 'a surface descriptor
DIM oSurface as Object 'the surface

' get a surface descriptor
Set oSD = iDRAW1.CreateSurfaceDesc()

' set surface descriptor properties - for a primary surface in this case
oSD.SCaps = DDSCAPS_PRIMARYSURFACE

' create the surface using the surface descriptor
Set oSurface = iDRAW1.CreateSurface ( oSD )
```

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::CreateSurface ()

Enumerates all display device drivers installed on the system

Syntax

controlname.**EnumDevices**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

short

Parameters

None

Remarks

This method initialises the DeviceList property with a list of all available display devices.

Returns the number of devices found.

This method must be called before using the DeviceList property.

Direct Draw Compatibility

This method is equivalent to DirectDrawEnumerate ()

Enumerates all possible display modes supported by the hardware.

Syntax

controlname.**EnumDisplayModes**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

short

Parameters

None

Remarks

This method initialises the DisplayModeList property with a list of all supported display modes for an open device (see the OpenDevice method).

Returns the number of display modes supported.

This method must be called before using the DisplayModeList property.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::EnumDisplayModes()

Enumerates surfaces that meet a search criterion.

Syntax

controlname.**EnumSurfaces**(long lFlags, object oSurfaceDesc)

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

short

Parameters

lFlags	Type of search
oSurfaceDesc	Describes the search criterion

Remarks

This method will enumerate all surfaces that meet the criterion described by oSurfaceDesc. iFlags indicates the type of results and can be one of:

DDENUMSURFACES_ALL	Enumerates all of the surfaces that meet the search criterion.
DDENUMSURFACES_CANBECREATED	Enumerates the first surface that can be created and meets the search criterion.
DDENUMSURFACES_DOESEXIST	Enumerates the already existing surfaces that meet the search criterion.
DDENUMSURFACES_MATCH	Searches for any surface that matches the surface description.
DDENUMSURFACES_NOMATCH	Searches for any surface that does not match the surface description.

Returns the number of surfaces found.

This method must be called before using the SurfaceList property.

This method tries to temporarily create a surface matching the search criterion if the DDENUMSURFACES_CANBECREATED flag is set. Remember to call the Release method after enumerating a surface using with this flag.

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::EnumSurfaces ()

Enumerates all supported FourCC codes.

Syntax

controlname.**EnumFourCC**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

short

Parameters

None

Remarks

Non-RGB surface formats are described by FOURCC codes.

This method initialises the FourCCList property with a list of all supported FOURCC codes.

Returns the number of FOURCC codes supported.

This method must be called before using the FourCCList property.

Direct Draw Compatibility

This method is equivalent to the IDirectDraw2::GetFourCCCodes method

Creates an iPalette object.

Syntax

controlname.**CreatePalette**(long lFlags, object oColorTable)

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iPalette Object

Parameters

lFlags	Specifies the type of palette
oColorTable	A ColorTable object that describes the palette entires

Remarks

Use this method to create a DirectDraw Palette. In order to create a palette you must first create and fill in a ColorTable object that defines all the colors of the palette. A ColorTable object is synonymous with the Win32 PALETTEENTRY structure.

lFlags may be set to :

DDPCAPS_1BIT	1 bit Palette. There are 2 entries in the color table.
DDPCAPS_2BIT	2 bit palette. There are 4 entries in the color table.
DDPCAPS_4BIT	4 bit palette. There are 16 entries in the color table.
DDPCAPS_8BITENTRIES	The color table entries are indexes to a target surface's 8-bit palette. This flag is valid only when used with the DDPCAPS_1BIT, DDPCAPS_2BIT, or DDPCAPS_4BIT flag, and when the target surface is 8 bits per pixel.
DDPCAPS_8BIT	8 bit palette. There are 256 entries in the color table.
DDPCAPS_ALLOW256	Palette can have all 256 entries defined.

VB Example

```
DIM oCT as Object 'a color table
DIM oPalette as Object 'the palette

' get a color table
Set oCT = iDRAW1.CreateColorTable()

' initialise the color table - we will create an 8-bit palette (256 colors)
I = 0
While (I < 256)
       oCT.Red = I
       oCT.Green = I
       oCT.Blue = I
       I = I + 1
Wend

' specify the palette type - an 8-bit palette
iFlag = DDPCAPS_8BIT

' create the palette
Set oPalette = iDRAW1.CreatePalette ( iFlag, oCT )
```

Direct Draw Compatibility

This method is equivalent to IDirectDraw2::CreatePalette ()

Creates an iBltFx object.

Syntax

controlname.**CreateBltFx**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iBltFx Object

Parameters

None

Remarks

The iBltFx object encapsulates the DDBLTFX structure used to pass information to the Blt and BltBatch methods.

Use this method to first create an iBltFx object. You then set the objects properties and pass the object as a parameter to other methods.

Direct Draw Compatibility

The iBltFx object is equivalent to the DDBLTFX structure

Creates an iCaps object.

Syntax

controlname.**CreateCaps**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iCaps Object

Parameters

None

Remarks

The iCaps object encapsulates the DDCAPS structure that is used to specify the hardware capabilities.

Use this method to first create an iCaps object. You then set the objects properties and pass the object as a parameter to other methods.

Direct Draw Compatibility

The iCaps object is equivalent to the DDCAPS structure

Creates an iColorKey object

Syntax

controlname.**CreateColorKey**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iColorKey Object

Parameters

None

Remarks

The iColorKey object encapsulates the DDCOLORKEY structure that is used to specify the color key for the source or destination or color space.

Use this method to first create an iColorkey object. You then set the objects properties and pass the object as a parameter to other methods.

Direct Draw Compatibility

The iColorKey object is equivalent to the DDCOLORKEY structure

Creates an object of iColorTable

Syntax

controlname.**CreateColorTable**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iColorTable Object

Parameters

None

Remarks

The iColorTable object can be used to specify the palette entries required to create an iPalette object. See the CreatePalette method.

Use this method to first create an iColorTable object. You then set the objects properties and pass the object as a parameter to other methods.

Direct Draw Compatibility

The iColorTable object is similar to the Win32 PALETTEENTRY structure

Creates an iDevice object.

Syntax

controlname.**CreateDevice**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iDevice Object

Parameters

None

Remarks

iDevice object maintains information like the driver name, driver description and GUID of a device. You would not normally use this method. Instead use the EnumDevices method and DeviceList property to get information about a device.

Direct Draw Compatibility

This method has no equivalent in Direct Draw

Creates an iOverlayFx object

Syntax

controlname.**CreateOverlayFx**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iOvelayFx Object

Parameters

None

Remarks

The iOverlayFx object encapsulates the DDOVERLAYFX structure used to pass information to the UpdateOverlay method

Use this method to first create an iOverlayFx object. You then set the objects properties and pass the object as a parameter to other methods.

Direct Draw Compatibility

The iOverlayFx object is equivalent to the DDOVERLAYFX structure

Creates an iPixelFormat object.

Syntax

controlname.**CreatePixelFormat**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iPixelFormat Object

Parameters

None

Remarks

The iPixelFormat object encapsulates the DDPIXELFORMAT structure which is used to specify the pixel format of a surface.

Use this method to first create an iPixelFormat object. You then set the objects properties and pass the object as a parameter to other methods.

Direct Draw Compatibility

The iPixelFormat object is equivalent to the DDPIXELFORMAT structure

Creates an iSurfaceDesc object.

Syntax

controlname.**CreateSurfaceDesc**()

controlname is the name of the **iDraw** Control object, for example, iDraw1.

Return Value

iSurfaceDesc Object

Parameters

None

Remarks

The iSurfaceDesc object encapsulates the DDSURFACEDESC structure which is used to describe a surface

Use this method to first create an iSurfaceDesc object. You then set the objects properties and pass the object as a parameter to other methods.

Direct Draw Compatibility

The iSurfaceDesc object is equivalent to the DDSURFACEDESC structure

AttachedSurface[nIndex]	The surface with the specified capabilities that is attached to this surface.
Caps	The iCaps object associated with this surface.
Clipper	The iClipper object associated with this surface
ColorKey[lFlags]	The iColorKey object associated with this surface.
Palette	The iPalette object associated with this surface
PixelFormat	The iPixelFormat object associated with this surface.
SurfaceDesc	The iSurfaceDesc object that contains the description of this surface

See Also

iSurface Methods

Set or Get the iClipper object associated with this surface.

Syntax

objectname.**Clipper =** [object]

objectname is the name of the iSurface object

Type

iClipper Object

Remarks

This property can be set for any surface but is mainly used when the surface is being overlaid on or blitted to the primary surface.

VB Example

```
Dim oClipper as Object
Dim oSurface as Object

Set oClipper = iDRAW1.CreateClipper

‘assume oSurface is properly setup first

oSurface.Clipper = oClipper
```

Direct Draw Compatibility

This property encapsulates IDirectDrawSurface2::SetClipper() and IDirectDrawSurface2::GetClipper ().

Set or Get the iPalette object associated with this surface.

Syntax

objectname.**Palette** = [object]

objectname is the name of the iSurface object

Type

iPalette Object

Remarks

Use this method to attach a palette to a surface. Once you attach a palette to a surface, it is used for all future operations.

VB Example

```
Dim oPalette as Object
Dim oSurface as Object

Set oPalette = iDRAW1.CreatePalette

‘assume oSurface is properly setup first

oSurface.Palette = oPalette
```

Direct Draw Compatibility

This property encapsulates IDirectDrawSurface2::SetPalette() and IDirectDrawSurface2::GetPalette().

Get the iSurfaceDesc object that describes the surface.

Syntax

objectname.**SurfaceDesc**

objectname is the name of the iSurface object

Type

iSurfaceDesc Object

Remarks

This property describes the surface characteristics.

Direct Draw Compatibility

This property encapsulates IDirectDrawSurface2::GetSurfaceDesc().

Get the iPixelFormat object associated with this surface

Syntax

objectname.**PixelFormat**

objectname is the name of the iSurface object

Type

iPixelFormat Object

Remarks

This property can be used to get the color and pixel format of the surface.

Direct Draw Compatibility

This property encapsulates IDirectDrawSurface2::GetPixelFormat().

Get the iCaps object associated with this surface

Syntax

objectname.**GetCaps**

objectname is the name of the iSurface object

Type

iCaps Object

Remarks

This property can be used to retrieve the capabilities of the surface.

Direct Draw Compatibility

This property encapsulates IDirectDrawSurface2::GetCaps ().

Get or Set the iColorKey object associated with this surface

Syntax

objectname.**ColorKey[lFlags]** = [object]

objectname is the name of the iSurface object

Type

iColorKey Object

Remarks

This property can be used to get or set the color key value for the surface

Direct Draw Compatibility

This property encapsulates IDirectDrawSurface2::GetColorKey() and IDirectDrawSurface2::SetColorKey().

Obtains an attached surface from the list of enumerated attached surfaces

Syntax

objectname.**AttachedSurface[nIndex]**

objectname is the name of the iSurface object

Type

iSurface Object

Remarks

Lists the surfaces currently attached to this surface. An attached surface can be a z-buffer, alpha channel or a back buffer. You must call the EnumAttachedSurfaces method before using this property.

Direct Draw Compatibility

This property is equivalent to IDirectDrawSurface2::GetAttachedSurface()

AddAttachedSurface	Attaches a surface to another surface.
AddOverlayDirtyRect	Creates a list of the rectangles that have to be updated the next time the UpdateOverlayDisplay method is called.
BltFast	Performs a source copy blit or transparent blit
Blt	Performs a bit block transfer.
DeleteAttachedSurface	Detaches two attached surfaces.
EnumAttachedSurfaces	Enumerates all the surfaces attached to a given surface.
EnumOverlayZorders	Enumerates the overlay surfaces on the specified destination.
Flip	Switches the surface memory associated with the DDSCAPS_BACKBUFFER surface and the front-buffer surface.
GetBltStatus	Obtains the blitter status.
GetDC	Returns a GDI-compatible handle of a device context for the surface.
GetFlipStatus	Indicates whether the surface has finished its flipping process.
GetOverlayXPos	Returns the x coordinate of a visible active overlay surface.
GetOverlayYPos	Returns the y coordinate of a visible active overlay surface.
IsLost	Determines if the surface memory associated with an iSurface object has been freed.
LoadBitmap	Loads a bitmap onto the surface
LoadPalettemethod_loadpalette	Loads a palette from the bitmap blitted onto this surface
Lock	Locks the surface memory.
PageLock	Prevents a surface created in system-memory from being paged out
PageUnlock	Unlocks a system-memory surface, allowing it to be paged out.
ReleaseDC	Releases the handle of a device context previously obtained by using the GetDC method.
Restore	Restores a surface that has been lost.
SetOverlayPosition	Alters the display coordinates of an overlay surface.
UpdateOverlayDisplay	Repaints the rectangles in the dirty rectangle list of all the active overlays.
UpdateOverlay	Repositions or modifies the visual attributes of

	an overlay surface.
UpdateOverlayZorder	Sets the z-order of an overlay.

See Also

iSurface Properties

Performs a bit block transfer.

Syntax

```
objectname.Blt (  long lDstLeft,
                  long lDstTop,
                  long lDstRight,
                  long lDstBottom,
                  object DstSurface,
                  long lSrcLeft,
                  long lSrcTop,
                  long lSrcRight,
                  long lSrcBottom,
                  long lFlags,
                  object SrcBltFx)
```

objectname is the name of the iSurface object.

Return Value

long

Parameters

lDstLeft	The left coordinate of the destination rectangle
lDstTop	The top coordinate of the destination rectangle
lDstRight	The right coordinate of the destination rectangle
lDstBottom	The bottom coordinate of the destination rectangle
DstSurface	The destination surface
lSrcLeft	The left coordinate of the source rectangle
lSrcTop	The top coordinate of the source rectangle
lSrcRight	The right coordinate of the source rectangle
lSrcBottom	The bottom coordinate of the source rectangle
lFlags	The type of blit
SrcBltFx	The raster operations, effects, and override information

Remarks

A surface can be created in display memory or in system memory. This method can be used for blitting from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be used for blitting from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

An important point to note is that if the source rectangle is smaller than the destination rectangle, then it is expanded to fit the destination rectangle and if the source rectangle is bigger than the destination rectangle, then it is compressed to fit the destination rectangle.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectSurface2::Blt()

Performs a fast bit block transfer

Syntax

objectname.**BltFast** (long DstX,
long DstY,
object lpdispSrcSurface,
long lSrcLeft,
long lSrcTop,
long lSrcRight,
long lSrcBottom,
long lTrans)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lDstX	The x coordinate to blit to on the destination surface.
lDstY	The y-coordinate to blit to on the destination surface.
lpdispSurface	The surface that is the source for the blit operation.
lSrcLeft	The left coordinate of the source rectangle
lSrcTop	The top coordinate of the source rectangle
lSrcRight	The right coordinate of the source rectangle
lSrcBottom	The bottom coordinate of the source rectangle
lTrans	Type of transfer.

Remarks

This method uses the source color key or destination color key to perform a source copy blit or transparent blit. An asynchronous blit is attempted if it is supported by the hardware.

This method has two drawbacks. First of all it works only on display memory surfaces. Secondly, it cannot clip when blitting. BltFast is slightly faster than the Blt method if display hardware is not being used for the blt.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectSurface2::BltFast method

Attaches a surface to this surface.

Syntax

objectname.**AddAttachedSurface**(object oAttachedSurface)

objectname is the name of the iSurface object.

Return Value

long

Parameters

oAttachedSurface	The surface that is to be attached - an iSurface object

Remarks

oAttachedSurface can be a z-buffer, alpha channel, or a back buffer. A z-buffer stores the depth value for each pixel in a scene, an alpha channel specifies the opacity of an image and a back buffer is a surface onto which images can be blitted before displaying them on the primary surface.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectSurface2::AddAttachedSurface()

See Also

EnumAttachedSurfaces method

DeleteAttachedSurface method

Flip method

Enumerates all the surfaces attached to this surface.

Syntax

objectname.**EnumAttachedSurfaces**()

objectname is the name of the iSurface object.

Return Value

long

Parameters

None

Remarks

This method creates a list of all attached surfaces. An attached surface can then be obtained using the AttachedSurface property.

Returns the number of attached surfaces found.

Direct Draw Compatibility

This method is equivalent to IDirectSurface2::EnumAttachedSurfaces()

Detaches two attached surfaces.

Syntax

objectname.**DeleteAttachedSurfaces**(long lFlags, object oAttachedSurface)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	This parameter is currently not used and must be set to 0.
oAttachedSurface	The surface to be detached. If this parameter is NULL, all attached surfaces are detached.

Remarks

Note that this method cannot be used to detach surfaces that have been attached by any means other than by using the AddAttachedSurface method.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::DeleteAttachedSurface()

See Also

Flip method

Performs a flip operation from a back buffer to a front buffer

Syntax

objectname.**Flip**(object oTargetSurface, long lFlags)

objectname is the name of the iSurface object.

Return Value

long

Parameters

oTargetSurface	The surface that will be flipped to
lFlags	This parameter determines whether to continue trying to flip even if DDERR_WASSTILLDRAWING error is received

Remarks

Ensure that this surface has the DDSCAPS_FLIP and DDSCAPS_FRONTBUFFER values set.

The oTargetSurface parameter is used in rare cases when the back buffer is not the buffer that should become the front buffer. Typically this parameter is NULL.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::Flip()

See Also

GetFlipStatus method

Determines whether the surface memory associated with a surface has been released or not.

Syntax

objectname.**IsLost**()

objectname is the name of the iSurface object.

Return Value

long

Parameters

None

Remarks

This method together with the Restore method can be used to reallocate surface memory.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::IsLost()

See Also

Restore method

Restores a surface whose surface memory has been freed

Syntax

objectname.**Restore**()

objectname is the name of the iSurface object.

Return Value

long

Parameters

None

Remarks

When a surface loses its surface memory, no functions can be performed on it. This method reallocates surface memory and reattaches it to the surface

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::Restore()

See Also

AddAttachedSurface method

IsLost method

Builds a list of rectangles that have to be updated the next time the UpdateOverlayDisplay method is called.

Syntax

objectname.**AddOverlayDirtyRect**(long lRectLeft, long lRectTop, lRectRight, lRectBottom)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lRectLeft	The left coordinate of the rectangle that needs to be updated.
lRectTop	The top coordinate of the rectangle that needs to be updated.
lRectRight	The right coordinate of the rectangle that needs to be updated.
lRectBottom	The bottom coordinate of the rectangle that needs to be updated.

Remarks

This method is not used if the hardware supports overlays.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2:: AddOverlayDirtyRect()

See Also

UpdateOverlayDisplay method

Modify the display coordinates of an overlay surface.

Syntax

objectname.**SetOverlayPosition**(long lXPos, long lYPos)

objectname is the name of the IiSurface object.

Return Value

long

Parameters

lXPos	New x -display coordinate.
lYPos	New y-display coordinate.

Remarks

An overlay is the top-most screen component. It also stores information about the primary surface on which it is used.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::SetOverlayPosition()

See Also

SetOverlayPosition method

UpdateOverlay method

Relocate or change the visual attributes of an overlay surface.

Syntax

objectname.**UpdateOverlay**(long lSrcLeft,
long lSrcTop,
long lSrcRight,
long lSrcBottom,
object oDestSurface,
long lDstLeft,
long lDstTop,
long lDstRight,
long lDstBottom,
long lFlags,
object oOverlayFx)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lSrcLeft	X- coordinate of the region on the source surface being used as the overlay.
lSrcTop	Y- coordinate of the region on the source surface being used as the overlay.
lSrcRight	Width of the region on the source surface being used as the overlay.
lSrcBottom	Height of the region on the source surface being used as the overlay
oDestSurface	The surface that is being overlaid.
lDstLeft	X- coordinate of the region on the destination surface that the overlay should be moved to.
lDstTop	Y- coordinate of the region on the destination surface that the overlay should be moved to.
lDstRight	Width of the region on the destination surface that the overlay should be moved to.
lDstBottom	Height of the region on the destination surface that the overlay should be moved to.
lFlags	The type of update
oOverlayFx	Passes override information

Remarks

An overlay surface is the top-most screen component. It also stores information about the primary surface on which it is used.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2:: UpdateOverlay()

Redraw the rectangles in the dirty rectangle list of all visible overlays.

Syntax

objectname.**UpdateOverlayDisplay**(long lFlags)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	The type of update to perform.

Remarks

This method does nothing if the hardware supports overlays.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::UpdateOverlayDisplay()

See Also

AddOverlayDirtyRect method

Specifies the z-order of an overlay.

Syntax

objectname.**UpdateOverlayZOrder**(long lFlags, object oReference)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	Specifies if the overlay has to be moved in front of or behind another overlay, or if its position in the overlay chain has to be modified.
oReference	The overlay surface in the overlay chain in case the overlay is being moved in front of or behind a particular overlay.

Remarks

The z-order of overlays determine the order in which they clip each other,

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::UpdateOverlayZOrder()

See Also

EnumOverlayZorders method

Enumerates all overlay surface z-orders

Syntax

objectname.**EnumOverlayZOrders**(long lFlags)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	Specifies whether to enumerate the overlays back to front. or front to back

Remarks

The overlays can be enumerated in any order: either front-to-back or back-to-front.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2:: EnumOverlayZOrders()

Locks the surface memory associated with the surface

Syntax

objectname.**Lock**(long lDstLeft,
long lDstTop,
long DstRight,
long DstBottom,
object lpdispSurfaceDesc,
long lFlags,
OLE_HANDLE hEvent)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lDstLeft	The left coordinate of the rectangle that identifies the region of surface that is being locked.
lDstTop	The top coordinate of the rectangle that identifies the region of surface that is being locked.
lDstRight	The right coordinate of the rectangle that identifies the region of surface that is being locked.
lDstBottom	The bottom coordinate of the rectangle that identifies the region of surface that is being locked.
lpdispSurfaceDesc	Details about the surface.
lFlags	Specifies the flags to be used
hEvent	Handle of a system event that is triggered when the surface is ready to be locked.

Remarks

When a surface is locked, you get access to its memory until you unlock the surface using the UnLock method. No blitting is possible from or to any part of a locked surface.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::Lock()

See Also

GetDC method

ReleaseDC method

Prevents a surface created in system-memory from being paged out

Syntax

objectname.**PageLock**(long lFlags)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	This parameter is currently not used and must be set to 0.

Remarks

When PageLock is called, the lock count for the surface is incremented and when PageUnlock is called, it is decremented. The memory is unlocked when the count becomes zero.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::PageLock()

See Also

PageUnlock method

Unlocks a system-memory surface

Syntax

objectname.**PageUnLock**(long lFlags)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	This parameter is currently not used and must be set to 0.

Remarks

When PageLock is called, the lock count for the surface is incremented and when PageUnlock is called, it is decremented. The memory is unlocked when the count becomes zero.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::PageUnLock()

See Also

PageLock method

Obtains the status of the blitter.

Syntax

objectname.**GetBltStatus**(long lFlags)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	This flag can be set to determine whether a blit involving this surface can be performed immediately or to determine if a blit involving this surface is complete.

Remarks

This method can be called before performing a blit operation.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::GetBltStatus()

Determines the status of a flip operation

Syntax

objectname.**GetFlipStatus**(long lFlags)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lFlags	This flag can be set to determine whether a flip involving this surface can be performed immediately or to determine if a flip involving this surface is complete

Remarks

This method can be called before performing a flip operation.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::GetFlipStatus()

Load a bitmap onto the surface.

Syntax

objectname.**LoadBitmap**(string lpctstrBitmap)

objectname is the name of the iSurface object.

Return Value

long

Parameters

lpctstrBitmap	The path of the bitmap file

Remarks

This is a utility method provided to easily load a bitmap onto a surface.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method has no equivalent in Direct Draw.

Load a palette from the bitmap loaded onto the surface

Syntax

objectname.**LoadPalette**()

objectname is the name of the iSurface object.

Return Value

long

Parameters

None

Remarks

This is a utility method provided to easily load a palette onto a surface.

This method creates a default 3-3-2 RGB palette if it does not succeed in loading the palette associated with the bitmap.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method has no equivalent in Direct Draw.

Gets a GDI-compatible handle of a device context for the surface.

Syntax

objectname.**GetDC**()

objectname is the name of the iSurface object.

Return Value

OLE_HANDLE

Parameters

None

Remarks

This method in turn calls the Lock method to lock the surface. The lock is maintained until the ReleaseDC method is called.

Returns the handle to a device contect (HDC).

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::GetDC()

See Also

Lockmethod

Releases the device context handle obtained using the GetDC method.

Syntax

objectname.**ReleaseDC**(OLE_HANDLE hDC)

objectname is the name of the IiSurface object.

Return Value

long

Parameters

hDC	Handle of a device context previously obtained by GetDC.

Remarks

This method in turn calls the UnLock method to unlock the surface

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::GetDC method

See Also

Lockmethod

Gets the x-coordinate of a visible, active overlay surface

Syntax

objectname.**GetOverlayXPos**()

objectname is the name of the IiSurface object.

Return Value

long

Parameters

None

Remarks

An overlay surface is a surface which has the DDSCAPS_OVERLAY flag set.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::GetOverlayPosition(), except that it returns only the x coordinate.

See Also

SetOverlayPositionmethod

UpdateOverlaymethod

Gets the y-coordinate of a visible, active overlay surface

Syntax

objectname.**GetOverlayYPos**()

objectname is the name of the iSurface object.

Return Value

long

Parameters

None

Remarks

An overlay surface is a surface which has the DDSCAPS_OVERLAY flag set.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compatibility

This method is equivalent to IDirectDrawSurface2::GetOverlayPosition(), except that it returns only the y coordinate.

See Also

SetOverlayPositionmethod

UpdateOverlaymethod

HWnd	The window handle that will receive the clipping information.

Set or Gets the window handle that will receive the clipping information.

Syntax

objectname.**HWnd =** [HWnd]

objectname is the name of the **iClipper** object.

Type

OLE_HANDLE

Remarks

The clipping information includes clip lists. A clip list is a list of rectangles specifying areas of the surface that are visible.

Direct Draw Compatibilty

This property is the equivalent to the window handle set using IDirectDrawClipper::SetHWnd() and retreived using IDirectDrawClipper::GetHWnd().

GetCaps	Retrieves the palette capabilities
GetEntries	Queries the palette entries.
SetEntries	Changes the palette entries

Gets the capabilities of the palette

Syntax

objectname.**GetCaps** ()

objectname is the name of the **iPalette** object.

Return Value

long

Parameters

None

Remarks

The capabilities of the palette include the number of entries in the color table.

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compability

This method is equivalent to IDirectDrawPalette::GetCaps().

Sets the palette entries.

Syntax

objectname.**SetEntries** (long lFlags, long lStart, long lCount, object oColorTable)

objectname is the name of the **iPalette** object.

Return Value

long

Parameters

lFlags	This parameter is currently not used and must be set to 0.
lStart	First entry to be set.
lCount	Number of palette entries to be changed.
oColorTable	New palette entries - an iColorTable object

Remarks

Ensure that the palette has been associated with a surface using the SetPalette method before calling this method

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compability

This method is equivalent to the IDirectDrawPalette::SetEntries().

See Also

SetEntries method

Palette property

Gets the palette entries.

Syntax

objectname.**GetEntries** (long lFlags, long lStart, long lCount, object oColorTable)

objectname is the name of the **iPalette** object.

Return Value

long

Parameters

lFlags	This parameter is currently not used and must be set to 0.
lStart	Start of the entries that should be retrieved sequentially.
lCount	Number of palette entries to be returned
oColorTable	Palette entries.

Remarks

Ensure that the palette has been associated with a surface using the SetPalette method before calling this method

Returns DD_OK if successful, otherwise an error code. Use the LastErrorString property to get the error message.

Direct Draw Compability

This method is equivalent to the IDirectDrawPalette::GetEntries().

See Also

SetEntries method

AlphaDestConstBitDepth	Destination alpha constant.bit depth
AlphaDestConst	Alpha channel destination.constant.
AlphaDestSurface	Alpha channel destination.surface
AlphaEdgeBlendBitDepth	Alpha edge blend.bit depth constant
AlphaEdgeBlend	Alpha edge blend constant.
AlphaSrcConstBitDepth	Source alpha constant.bit depth
AlphaSrcConst	Alpha channel source.constant.
AlphaSrcSurface	Alpha channel source.surface
DDFX	The type of FX operation.
DDROP	DirectDraw raster operations.
DestColorkey	Destination color key override.
FillColor	Color used to fill a surface when DDBLT_COLORFILL is specified
FillDepth	Z-buffer.depth value
PatternSurface	Pattern.surface.
ROP	Win32 raster operations.
RotationAngle	Angle of rotation for the blit.
SrcColorKey	Source color key override.
ZBufferBaseDest	Z-buffer destination base value
ZBufferDestSurface	Z-buffer destination surface.
ZBufferHigh	Z-buffer high limit.
ZBufferLow	Z-buffer low limit.
ZBufferOpCode	Z-buffer compares.
ZBufferSrcSurface	Z-buffer source.surface
ZDestConstBitDepth	Destination z-constant bit depth
ZDestConst	Z-buffer destination constant
ZSrcConstBitDepth	Source z-constant bit depth
ZSrcConst	Z-buffer source.constant

The type of FX operations.

Syntax

objectname.**DDFX = [**ddfx**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

This property specifies the type of stretching and also the angle by which the surface is to be rotated during the blit operation.

Direct Draw Compatibility

This property is the equivalent to the dwDDFX member of the DDBLTFX structure.

The Win32 raster operations.

Syntax

objectname.**ROP = [**rop**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

Raster-operation codes define how the graphics device interface (GDI) combines the bits from the selected pen with the bits in the destination bitmap.

Direct Draw Compatibility

This property is the equivalent to the dwROP member of the DDBLTFX structure.

DirectDraw raster operations.

Syntax

objectname.**DDROP = [**ddrop**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

Raster-operation codes define how the graphics device interface (GDI) combines the bits from the selected pen with the bits in the destination bitmap.

Direct Draw Compatibility

This property is the equivalent to the dwDDROP member of the DDBLTFX structure.

Angle of rotation for the blit.

Syntax

objectname. **RotationAngle = [**RotationAngle**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

During the blit operation, the image loaded on the surface is rotated by the specified angle and then blit onto the destination.

Direct Draw Compatibility

This property is the equivalent to the dwRotationAngle member of the DDBLTFX structure.

The Z-buffer compare code.

Syntax

objectname.**ZBufferOpCode = [**ZBufferOpCode**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferOpCode member of the DDBLTFX structure.

The z-buffer low limit.

Syntax

objectname. **ZBufferLow = [**ZBufferLow**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferLow member of the DDBLTFX structure.

The z-buffer high limit.

Syntax

objectname. **ZBufferHigh = [**ZBufferHigh**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferHigh member of the DDBLTFX structure.

The z-buffer destination base value

Syntax

objectname. **ZBufferBaseDest = [**ZBufferBaseDest**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferBaseDest member of the DDBLTFX structure.

The destination z-buffer constant bit depth

Syntax

objectname. **ZDestConstBitDepth = [**ZDestConstBitDepth**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZDestConstBitDepth member of the DDBLTFX structure.

The z-buffer destination constant.

Syntax

objectname. **ZDestConst = [**ZDestConst**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZDestConst member of the DDBLTFX structure.

The z-buffer destination surface.

Syntax

objectname. **ZBufferDestSurface = [**ZBufferDestSurface**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferDestSurface member of the DDBLTFX structure.

The source z-buffer constant bit depth.

Syntax

objectname. **ZSrcConstBitDepth = [**ZSrcConstBitDepth**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZSrcConstBitDepth member of the DDBLTFX structure.

The z-buffer source constant.

Syntax

objectname. **ZSrcConst = [**ZSrcConst**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZSrcConst member of the DDBLTFX structure.

The z-buffer source surface

Syntax

objectname. **ZBufferSrcSurface = [**ZBufferSrcSurface**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferSrcSurface member of the DDBLTFX structure.

The alpha edge blend bit depth.

Syntax

objectname. **AlphaEdgeBlendBitDepth = [**AlphaEdgeBlendBitDepth**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

Alpha edge blending is a display hardware feature that is not yet available. In alpha edge blending, the mixing of colors is done at the hardware pixel level.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaEdgeBlendBitDepth member of the DDBLTFX structure.

The alpha edge blend.

Syntax

objectname. **AlphaEdgeBlend = [**AlphaEdgeBlend**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

Alpha edge blending is a display hardware feature that is not yet available. In alpha edge blending, the mixing of colors is done at the hardware pixel level.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaEdgeBlend member of the DDBLTFX structure.

The destination alpha constant bit depth.

Syntax

objectname. **AlphaDestConstBitDepth = [**AlphaDestConstBitDepth**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

The alpha constant is a value that determines the clarity of a surface.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaDestConstBitDepth member of the DDBLTFX structure.

The alpha channel destination constant.

Syntax

objectname. **AlphaDestConst = [**AlphaDestConst**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

The alpha channel specifies the clarity of a surface.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaDestConst member of the DDBLTFX structure.

The alpha channel destination surface.

Syntax

objectname. **AlphaDestSurface = [**AlphaDestSurface**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

The alpha channel specifies the clarity of a surface.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaDestSurface member of the DDBLTFX structure.

The source alpha constant bit depth.

Syntax

objectname. **AlphaSrcConstBitDepth = [**AlphaSrcConstBitDepth**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

The alpha constant is a value that determines the clarity of a surface.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaSrcConstBitDepth member of the DDBLTFX structure.

The alpha channel source constant

Syntax

objectname. **AlphaSrcConst = [**AlphaSrcConst**]**

objectname is the name of the **iBltFx** object.

Type

long

Remarks

The alpha channel specifies the clarity of a surface.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaSrcConst member of the DDBLTFX structure.

The alpha channel source surface.

Syntax

objectname. **AlphaSrcSurface** = [AlphaSrcSurface]

objectname is the name of the **iBltFx** object.

Type

long

Remarks

The alpha channel specifies the clarity of a surface.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaSrcSurface member of the DDBLTFX structure.

The color used to fill a surface when DDBLT_COLORFILL is specified.

Syntax

objectname. **FillColor** = [FillColor]

objectname is the name of the **iBltFx** object.

Type

long

Remarks

Depending on the type or surface, an RGB triple or a palette index can be specified.

Direct Draw Compatibility

This property is the equivalent to the dwFillColor member of the DDBLTFX structure.

The z-buffer depth value.

Syntax

objectname. **FillDepth =** [FillDepth]

objectname is the name of the **iBltFx** object.

Type

long

Remarks

A z-buffer stores the depth of each pixel in an image.

Direct Draw Compatibility

This property is the equivalent to the dwFillDepth member of the DDBLTFX structure.

The pattern surface.

Syntax

objectname. **PatternSurface** = [PatternSurface]

objectname is the name of the **iBltFx** object.

Type

long

Remarks

This pattern is used when a source and destination is combined during a blit operation.

Direct Draw Compatibility

This property is the equivalent to the dwPatternSurface member of the DDBLTFX structure.

The destination color key override.

Syntax

objectname. **DestColorKey** = [DestColorKey]

objectname is the name of the **iBltFx** object.

Type

long

Remarks

Destination color keying specifies a color or color range that is overwritten on the destination.

Direct Draw Compatibility

This property is the equivalent to the dwDestColorKey member of the DDBLTFX structure.

The source color key override.

Syntax

objectname. **SrcColorKey** = [SrcColorKey]

objectname is the name of the **iBltFx** object.

Type

long

Remarks

Source color keying specifies a color or color range that is not replaced from the source onto the destination.

Direct Draw Compatibility

This property is the equivalent to the dwSrcColorKey member of the DDBLTFX structure.

ColorSpaceHighValue	The color range high value that is to be used as the color key.
ColorSpaceLowValue	The color range low value that is to be used as the color key.

The color range low value that is to be used as the color key.

Syntax

objectname.**ColorSpaceLowValue =** [ColorSpaceLowValue]

objectname is the name of the **iColorKey** object.

Type

long

Remarks

A color key is valid only if the low value and the high value are identical.

Direct Draw Compatibility

This property is the equivalent to the dwColorSpaceLowValue member of the DDCOLORKEY structure.

The color range high value that is to be used as the color key.

Syntax

objectname.**ColorSpaceHighValue =** [ColorSpaceHighValue]

objectname is the name of the **iColorKey** object.

Type

long

Remarks

A color key is valid only if the low value and the high value are identical.

Direct Draw Compatibility

This property is the equivalent to the dwColorSpaceHighValue member of the DDCOLORKEY structure.

AlignBoundaryDest	The alignment of the destination rectangle
AlignBoundarySrc	The alignment of the source rectangle
AlignSizeDest	The byte size of the destination rectangle
AlignSizeSrc	The byte size of the source rectangle
AlignstrideAlign	The stride alignment
AlphaBltConstBitDepths	Indicates the bit depth of the alpha blt constant
AlphaBltPixelBitDepths	Indicates the bit depth of the alpha blt pixel
AlphaBltSurfaceBitDepths	Indicates the bit depth of the alpha blt surface
AlphaOverlayConstBitDepths	Indicates the bit depth of the alpha overlay constant
AlphaOverlayPixelBitDepths	Indicates the bit depth of the alpha overlay pixel
AlphaOverlaySurfaceBitDepths	Indicates the bit depth of the alpha overlay surface
Caps2	More capabilities of the device driver
Caps	The capabilities of the device driver
CKeyCaps	Capabilities of the color key
CurrVisibleOverlays	The current number of visible overlays
FXAlphaCaps	The alpha capabilities of the device driver
FXCaps	The stretching and effects capabilities of the device driver
MaxHwCodecStretch	Maximum hardware codec stretch factor multiplied by 1000
MaxLiveVideoStretch	Maximum live video stretch multiplied by 1000
MaxOverlayStretch	Maximum overlay stretch factor multiplied by 1000
MaxVisibleOverlays	The maximum number of visible overlays
MinHwCodecStretch	Minimum hardware codec stretch factor multiplied by 1000
MinLiveVideoStretch	Minimum live video stretch multiplied by 1000
MinOverlayStretch	Minimum overlay stretch factor multplied by 1000
NumFourCCCodes	The number of FourCC codes
PalCaps	The palette capabilities of the device driver
Rops[RopSpace]	Raster operations supported.
Scaps	The general capabilities
SSBCaps	The capabilities of the device driver for system memory to system memory blits
SSBCFXCaps	The FX capabilities of the device driver for system memory to system memory blits.

SSBCKeyCaps	The color-key capabilities of the device driver for system memory to system memory blits.
SSBRops[RopSpace]	Raster operations supported for system-memory-to-system-memory blits.
SVBCaps	The capabilities of the device driver for system-memory to display memory blits
SVBCKeyCaps	The color-key capabilities of the device driver for system memory to display memory blits.
SVBFXCaps	The FX capabilities of the device driver for system memory to display memory blits.
SVBRops[RopSpace]	Raster operations supported for system-memory-to-display-memory blits.
SVCaps	The stereo vision capabilities of the device driver
VideMemTotal	The total amount of display memory
VidMemFree	The amount of free display memory.
VSBCaps	The capabilities of the device driver for display memory to system memory blits
VSBCKeyCaps	The color-key capabilities of the device driver for display memory to system memory blits.
VSBFXCaps	The FX capabilities of the device driver for display memory to system memory blits.
VSBRops[RopSpace]p	Raster operations supported for display-memory-to-system-memory blits.
ZBufferBitDepths	Indicates the bit depth of the z-buffer

The capabilities of the device driver

Syntax

objectname. **Caps =** [Caps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The capabilities of the device driver include its ability to provide 3D acceleration and its support for alpha channels, blitting, clipping, color keying, palettes and overlays.

Direct Draw Compatibility

This property is the equivalent to the dwCaps member of the DDCAPS structure.

More capabilities of the device driver.

Syntax

objectname. **Caps2=** [Caps2]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The advanced capabilities of the device driver include its ability to blit to or lock surfaces being used by Direct3D.

Direct Draw Compatibility

This property is the equivalent to the dwCaps2 member of the DDCAPS structure.

The color key capabilities.

Syntax

objectname. **CKeyCaps =** [CKeyCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The color key capabilities include the type of support for transparent blitting and overlaying.

Direct Draw Compatibility

This property is the equivalent to the dwCKeyCaps member of the DDCAPS structure.

The stretching and effects capabilities of the device driver.

Syntax

objectname. **FXCaps =** [FXCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The FX capabilties include the usage of arithmetic operations for stretching and shrinking, the type of mirroring and the angle of rotation during a blit operation.

Direct Draw Compatibility

This property is the equivalent to the dwFXCaps member of the DDCAPS structure.

The alpha capabilities of the device driver.

Syntax

objectname. **FXAlphaCaps =** [FXAlphaCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The alpha capabilities of the device driver include its support for alpha buffers, alpha channels and alpha blending.

Direct Draw Compatibility

This property is the equivalent to the dwFXAlphaCaps member of the DDCAPS structure.

The palette capabilities of the device driver.

Syntax

objectname. **PalCaps =** [PalCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The palette capabilities specify the number of entries in the color table.

Direct Draw Compatibility

This property is the equivalent to the dwPalCaps member of the DDCAPS structure.

The stereo vision capabilities of the device driver.

Syntax

objectname. **SVCaps =** [SVCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The stereo vision capabilities specify how the stereo view is accomplished.

Direct Draw Compatibility

This property is the equivalent to the dwSVCaps member of the DDCAPS structure.

The bit depth of the alpha blt constant

Syntax

objectname. **AlphaBltConstBitDepths =** [AlphaBltConstBitDepths]

objectname is the name of the **iCaps** object.

Type

long

Remarks

This property can specify 2-, 4- or 8-bits per pixel

Direct Draw Compatibility

This property is the equivalent to the dwAlphaBltConstBitDepths member of the DDCAPS structure.

The bit depth of the alpha blt pixel.

Syntax

objectname. **AlphaBltPixelBitDepths =** [AlphaBltPixelBitDepths]

objectname is the name of the **iCaps** object.

Type

long

Remarks

This property can specify 1-,2-, 4- or 8-bits per pixel

Direct Draw Compatibility

This property is the equivalent to the dwAlphaBltPixelBitDepths member of the DDCAPS structure.

The bit depth of the alpha blt surface.

Syntax

objectname. **AlphaBltSurfaceBitDepths =** [AlphaBltSurfaceBitDepths]

objectname is the name of the **iCaps** object.

Type

long

Remarks

This property can specify 1-,2-, 4- or 8-bits per pixel

Direct Draw Compatibility

This property is the equivalent to the dwAlphaBltSurfaceBitDepths member of the DDCAPS structure.

The bit depth of the alpha overlay constant.

Syntax

objectname. **AlphaOverlayConstBitDepths =** [AlphaOverlayConstBitDepths]

objectname is the name of the **iCaps** object.

Type

long

Remarks

This property can specify 2-, 4- or 8-bits per pixel

Direct Draw Compatibility

This property is the equivalent to the dwAlphaOverlayConstBitDepths member of the DDCAPS structure.

The bit depth of the alpha overlay pixel.

Syntax

objectname. **AlphaOverlayPixelBitDepths =** [AlphaOverlayPixelBitDepths]

objectname is the name of the **iCaps** object.

Type

long

Remarks

This property can specify 1-, 2-, 4- or 8-bits per pixel

Direct Draw Compatibility

This property is the equivalent to the dwAlphaOverlayPixelBitDepths member of the DDCAPS structure.

The bit depth of the alpha overlay surface.

Syntax

objectname. **AlphaOverlaySurfaceBitDepths =** [AlphaOverlaySurfaceBitDepths]

objectname is the name of the **iCaps** object.

Type

long

Remarks

This property can specify 1-, 2-, 4- or 8-bits per pixel

Direct Draw Compatibility

This property is the equivalent to the dwAlphaOverlaySurfaceBitDepths member of the DDCAPS structure.

The bit depth of the z-buffer.

Syntax

objectname. **ZBufferBitDepths =** [ZBufferBitDepths]

objectname is the name of the **iCaps** object.

Type

long

Remarks

This property can specify 8-, 16-, 24-, or 32-bits per pixel.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferBitDepths member of the DDCAPS structure.

The amount of display memory.

Syntax

objectname. **VidMemTotal =** [VidMemTotal]

objectname is the name of the **iCaps** object.

Type

long

Remarks

When a surface cannot be created in display memory, it has to be created in system memory. Therefore, it is useful to know the total amount of display memory available.

Direct Draw Compatibility

This property is the equivalent to the dwVidMemTotal member of the DDCAPS structure.

The amount of free display memory.

Syntax

objectname. **VidMemFree =** [VidMemFree]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The amount of free display memory is only an approximation as it keeps changing with the creation and destruction of surfaces.

Direct Draw Compatibility

This property is the equivalent to the dwVidMemFree member of the DDCAPS structure.

The maximum number of visible overlays.

Syntax

objectname. **MaxVisibleOverlays =** [MaxVisibleOverlays]

objectname is the name of the **iCaps** object.

Type

long

Remarks

An overlay surface stores information about the primary surface on which it is displayed and is assumed to be the top-most component on the screen.

Direct Draw Compatibility

This property is the equivalent to the dwMaxVisibleOverlays member of the DDCAPS structure.

The current number of visible overlays.

Syntax

objectname. **CurrVisibleOverlays =** [CurrVisibleOverlays]

objectname is the name of the **iCaps** object.

Type

long

Remarks

An overlay surface stores information about the primary surface on which it is displayed and is assumed to be the top-most component on the screen.

Direct Draw Compatibility

This property is the equivalent to the dwCurrVisibleOverlays member of the DDCAPS structure.

The number of FourCC codes.

Syntax

objectname. **NumFourCCCodes =** [NumFourCCCodes]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Four-character codes (FOURCC codes) are used to describe non-RGB surface formats.

Direct Draw Compatibility

This property is the equivalent to the dwNumFourCCCodes member of the DDCAPS structure.

The alignment of the source rectangle

Syntax

objectname. **AlignBoundarySrc =** [AlignBoundarySrc]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Direct Draw Compatibility

This property is the equivalent to the dwAlignBoundrySrc member of the DDCAPS structure.

The byte size of the source rectangle

Syntax

objectname. **AlignSizeSrc =** [AlignSizeSrc]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Direct Draw Compatibility

This property is the equivalent to the dwAlignSizeSrc member of the DDCAPS structure.

The alignment of the destination rectangle

Syntax

objectname. **AlignBoundaryDest =** [AlignBoundaryDest]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Direct Draw Compatibility

This property is the equivalent to the dwAlignBoundryDest member of the DDCAPS structure.

The byte size of the destination rectangle

Syntax

objectname. **AlignSizeDest =** [AlignSizeDest]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Direct Draw Compatibility

This property is the equivalent to the dwAlignSizeDest member of the DDCAPS structure.

The stride alignment.

Syntax

objectname. **AlignStrideAlign =** [AlignStrideAlign]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Direct Draw Compatibility

This property is the equivalent to the dwAlignStrideAlign member of the DDCAPS structure.

The general capabilities.

Syntax

objectname. **SCaps =** [SCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The general capabilities include the type of surface.

Direct Draw Compatibility

This property is the equivalent to the ddsCaps member of the DDCAPS structure.

The minimum overlay stretch factor.

Syntax

objectname. **MinOverlayStretch =** [MinOverlayStretch]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The value returned is the minimum overlay stretch factor multiplied by 1000.

Direct Draw Compatibility

This property is the equivalent to the dwMinOverlayStretch member of the DDCAPS structure.

The maximum overlay stretch factor.

Syntax

objectname. **MaxOverlayStretch =** [MaxOverlayStretch]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The value returned is the maximum overlay stretch factor multiplied by 1000.

Direct Draw Compatibility

This property is the equivalent to the dwMaxOverlayStretch member of the DDCAPS structure.

The minimum live video stretch factor

Syntax

objectname. **MinLiveVideoStretch =** [MinLiveVideoStretch]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The value returned is the minimum live video stretch factor multiplied by 1000.

Direct Draw Compatibility

This property is the equivalent to the dwMinLiveVideoStretch member of the DDCAPS structure.

The maximum live video stretch factor

Syntax

objectname. **MaxLiveVideoStretch =** [MaxLiveVideoStretch]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The value returned is the maximum live video stretch factor multiplied by 1000.

Direct Draw Compatibility

This property is the equivalent to the dwMaxLiveVideoStretch member of the DDCAPS structure.

The minimum hardware codec stretch factors

Syntax

objectname. **MinHwCodecStretch =** [MinHwCodecStretch]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The value returned is the minimum hardware codec factor multiplied by 1000.

Direct Draw Compatibility

This property is the equivalent to the dwMinHwCodecStretch member of the DDCAPS structure.

The maximum hardware codec stretch factors

Syntax

objectname. **MaxHwCodecStretch =** [MaxHwCodecStretch]

objectname is the name of the **iCaps** object.

Type

long

Remarks

The value returned is the maximum hardware codec factor multiplied by 1000.

Direct Draw Compatibility

This property is the equivalent to the dwMaxHwCodecStretch member of the DDCAPS structure.

The capabilities of the device driver for system-memory-to-display-memory blits.

Syntax

objectname. **SVBCaps =** [SVBCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwSVBCaps member of the DDCAPS structure.

The color-key capabilities of the device driver for system-memory-to-display-memory blits.

Syntax

objectname. **SVBCKeyCaps =** [SVBCKeyCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwSVBCKeyCaps member of the DDCAPS structure.

The FX capabilities of the device driver for system-memory-to-display-memory blits.

Syntax

objectname. **SVBFXCaps =** [SVBFXCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwSVBFXCaps member of the DDCAPS structure.

The capabilities of the device driver for display-memory-to-system-memory blits.

Syntax

objectname. **VSBCaps =** [VSBCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwVSBCaps member of the DDCAPS structure.

The color-key capabilities of the device driver for display-memory-to-system-memory blits.

Syntax

objectname. **VSBCKeyCaps =** [VSBCKeyCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwVSBCKeyCaps member of the DDCAPS structure.

The FX capabilities of the device driver for display-memory-to-system-memory blits.

Syntax

objectname. **VSBFXCaps =** [VSBFXCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwVSBFXCaps member of the DDCAPS structure.

The capabilities of the device driver for system-memory-to-system-memory blits.

Syntax

objectname. **SSBCaps =** [SSBCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwSSBCaps member of the DDCAPS structure.

The color-key capabilities of the device driver for system-memory-to-system-memory blits.

Syntax

objectname. **SSBKeyCaps =** [SSBKeyCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwSSBKeyCaps member of the DDCAPS structure.

FX capabilities of the device driver for system-memory-to-system-memory blits.

Syntax

objectname. **SSBCFXCaps =** [SSBCFXCaps]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Blitting can be done from a surface created in display memory to a surface created in system memory or from a surface created in system memory to a surface created in display memory. It can also be done from a surface created in display memory to another surface created in display memory or from a surface created in system memory to another surface created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwSSBCFXCaps member of the DDCAPS structure.

The raster operations supported.

Syntax

objectname. **Rops**[**RopSpace**]**=** Rops[RopSpace]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Raster-operation codes define how the graphics device interface (GDI) combines the bits from the selected pen with the bits in the destination bitmap.

Direct Draw Compatibility

This property is the equivalent to the dwRops[DD_ROP_SPACE] member of the DDCAPS structure.

The raster operations supported for system-memory-to-display-memory blits.

Syntax

objectname. **SVBRops**[**RopSpace**]**=** SVBRops[RopSpace]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Raster-operation codes define how the graphics device interface (GDI) combines the bits from the selected pen with the bits in the destination bitmap.

Direct Draw Compatibility

This property is the equivalent to the dwSVBRops[DD_ROP_SPACE] member of the DDCAPS structure.

The raster operations supported for display-memory-to-system-memory blits.

Syntax

objectname. **VSBRops**[**RopSpace**]**=** VSBRops[RopSpace]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Raster-operation codes define how the graphics device interface (GDI) combines the bits from the selected pen with the bits in the destination bitmap.

Direct Draw Compatibility

This property is the equivalent to the dwVSBRops[DD_ROP_SPACE] member of the DDCAPS structure.

The raster operations supported for system-memory-to-system-memory blits.

Syntax

objectname. **SSBRops**[**RopSpace**]**=** SSBRops[RopSpace]

objectname is the name of the **iCaps** object.

Type

long

Remarks

Raster-operation codes define how the graphics device interface (GDI) combines the bits from the selected pen with the bits in the destination bitmap.

Direct Draw Compatibility

This property is the equivalent to the dwSSBRops[DD_ROP_SPACE] member of the DDCAPS structure.

AlphaDestConstBitDepth	Alpha constant for a destination.
AlphaDestConst	Alpha channel value for a destination.
AlphaDestSurface	Alpha channel surface for a destination.
AlphaEdgeBlendBitDepth	The bit depth for an alpha edge blend.
AlphaEdgeBlend	Alpha value for an edge blend.
AlphaSrcConstBitDepth	Alpha constant for a source.
AlphaSrcConst	Alpha channel value for a source.
AlphaSrcSurface	Alpha channel surface for a source.
DDFX	Overlay FX flags.
DestColorKey	Destination color key override.
Flags	This member is currently not used and must be set to 0.
SrcColorKey	Source color key override.

The bit depth for an alpha edge blend.

Syntax

objectname. **AlphaEdgeBlendBitDepth=** AlphaEdgeBlendBitDepth

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

Alpha edge blending is a display hardware feature that is not yet available. In alpha edge blending, the mixing of colors is done at the hardware pixel level.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaEdgeBlendBitDepth member of the DDOVERLAYFX structure.

The alpha value for an edge blend.

Syntax

objectname. **AlphaEdgeBlend =** AlphaEdgeBlend

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

Alpha edge blending is a display hardware feature that is not yet available. In alpha edge blending, the mixing of colors is done at the hardware pixel level.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaEdgeBlend member of the DDOVERLAYFX structure.

The alpha constant for a destination

Syntax

objectname. **AlphaDestConstBitDepth =** AlphaDestConstBitDepth

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

Alpha constant is a level of opacity applied to an entire surface

Direct Draw Compatibility

This property is the equivalent to the dwAlphaDestConstBitDepth member of the DDOVERLAYFX structure.

The alpha channel value for a destination.

Syntax

objectname. **AlphaDestConst =** AlphaDestConst

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

Alpha channel is the opacity of an image

Direct Draw Compatibility

This property is the equivalent to the dwAlphaDestConst member of the DDOVERLAYFX structure.

The alpha channel surface for a destination.

Syntax

objectname. **AlphaDestSurface =** AlphaDestSurface

objectname is the name of the **iOverlayFx** object.

Type

object

Remarks

Alpha channel is the opacity of an image

Direct Draw Compatibility

This property is the equivalent to the lpDDSAlphaDest member of the DDOVERLAYFX structure.

The alpha constant for a source

Syntax

objectname. **AlphaSrcConstBitDepth =** AlphaSrcConstBitDepth

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

Alpha constant is a level of opacity applied to an entire surface

Direct Draw Compatibility

This property is the equivalent to the dwAlphaSrcConstBitDepth member of the DDOVERLAYFX structure.

The alpha channel value for a source.

Syntax

objectname. **AlphaSrcConst =** AlphaSrcConst

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

Alpha channel is the opacity of an image

Direct Draw Compatibility

This property is the equivalent to the dwAlphaSrcConst member of the DDOVERLAYFX structure.

The alpha channel surface for a source

Syntax

objectname. **AlphaSrcSurface =** AlphaSrcSurface

objectname is the name of the **iOverlayFx** object.

Type

object

Remarks

Alpha channel is the opacity of an image

Direct Draw Compatibility

This property is the equivalent to the lpDDSAlphaSrc member of the DDOVERLAYFX structure.

The destination color key override.

Syntax

objectname. **DestColorKey =** DestColorKey

objectname is the name of the **iOverlayFx** object.

Type

object

Remarks

The destination color key specifies a color or color range that is covered up on the destination.

Direct Draw Compatibility

This property is the equivalent to the dckDestColorkey member of the DDOVERLAYFX structure.

The source color key override.

Syntax

objectname. **SrcColorKey =** SrcColorKey

objectname is the name of the **iOverlayFx** object.

Type

object

Remarks

The source color key specifies a color or color range that is not visible on the destination.

Direct Draw Compatibility

This property is the equivalent to the dckSrcColorkey member of the DDOVERLAYFX structure.

The overlay FX flags.

Syntax

objectname. **DDFX =** DDFX

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

The overlay FX flags determine what type of stretching is to be done and how the overlay is to be mirrored.

Direct Draw Compatibility

This property is the equivalent to the dwDDFX member of the DDOVERLAYFX structure.

This property is currently not used and must be set to 0.

Syntax

objectname. **Flags =** Flags

objectname is the name of the **iOverlayFx** object.

Type

long

Remarks

Not used

Direct Draw Compatibility

This property is the equivalent to the dwFlags member of the DDOVERLAYFX structure.

AlphaBitDepth	Bit depth of the alpha channel.
BBitMask	Mask for blue bits.
Flags	Control flags.
FourCC	FourCC code.
GBitMask	Mask for green bits.
RBitMask	Mask for red bits.
RGBAlphaBitMask	Mask for alpha channel.
RGBBitCount	RGB bits per pixel
UBitMask	Mask for U bits.
VBitMask	Mask for V bits.
YBitMask	Mask for Y bits.
YUVAlphaBitMask	Mask for alpha channel.
YUVBitCount	YUV bits per pixel
ZBufferBitDepth	Bit depth of the z-buffer

The FourCC code.

Syntax

objectname. **FourCC =** [FourCC]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

Four-character codes (FOURCC codes).are used to describe non-RGB surface formats.

Direct Draw Compatibility

This property is the equivalent to the dwFourCC member of the DDPIXELFORMAT structure.

The RGB bits per pixel

Syntax

objectname. **RGBBitCount =** [RGBBitCount]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

RGB is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwRGBBitCount member of the DDPIXELFORMAT structure.

The YUV bits per pixel

Syntax

objectname. **YUVBitCount =** [YUVBitCount]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

YUV is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwYUVBitCount member of the DDPIXELFORMAT structure.

The bit depth of the Z-buffer.

Syntax

objectname. **ZBufferBitDepth =** [ZBufferBitDepth]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

Z-buffer is a buffer that stores a depth value for each pixel in a scene.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferBitDepth member of the DDPIXELFORMAT structure.

The bit depth of the alpha channel

Syntax

objectname. **AlphaBitDepth =** [AlphaBitDepth]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

Alpha channel is the opacity of an image.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaBitDepth member of the DDPIXELFORMAT structure.

The mask for red bits.

Syntax

objectname. **RBitMask =** [RBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

RGB is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwRBitMask member of the DDPIXELFORMAT structure.

The mask for Y bits.

Syntax

objectname. **YBitMask =** [YBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

YUV is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwYBitMask member of the DDPIXELFORMAT structure.

The mask for G bits.

Syntax

objectname. **GBitMask =** [GBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

RGB is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwGBitMask member of the DDPIXELFORMAT structure.

The mask for U bits.

Syntax

objectname. **UBitMask =** [UBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

YUV is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwUBitMask member of the DDPIXELFORMAT structure.

The mask for blue bits.

Syntax

objectname. **BBitMask =** [BBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

RGB is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwBBitMask member of the DDPIXELFORMAT structure.

The mask for V bits.

Syntax

objectname. **VBitMask =** [VBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

YUV is a type of color space used to encode and visualize color.

Direct Draw Compatibility

This property is the equivalent to the dwVBitMask member of the DDPIXELFORMAT structure.

The mask for the alpha channel.

Syntax

objectname. **RGBAlphaBitMask =** [RGBAlphaBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

The alpha channel specifies the clarity of an image.

Direct Draw Compatibility

This property is the equivalent to the dwRGBAlphaBitMask member of the DDPIXELFORMAT structure.

The mask for the alpha channel.

Syntax

objectname. **YUVAlphaBitMask =** [YUVAlphaBitMask]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

The alpha channel specifies the clarity of an image.

Direct Draw Compatibility

This property is the equivalent to the dwYUVAlphaBitMask member of the DDPIXELFORMAT structure.

The control flags.

Syntax

objectname. **Flags =** [Flags]

objectname is the name of the **iPixelFormat** object.

Type

long

Remarks

These flags specify the pixel format of the surface and decide whether the RGB and the YUV data are to be considered or not.

Direct Draw Compatibility

This property is the equivalent to the dwFlags member of the DDPIXELFORMAT structure.

AlphaBitDepth	Bit depth of alpha buffer.
BackBufferCount	Number of back buffers.
DestBltColorKey	Destination blit color key
DestOverlayColorkey	Destination overlay color key.
Flags	Control flags.
Height	Height of surface.
MipMapCount	Number of mipmap levels.
Pitch	The distance to start of next line
Pixelformat	Pixel format of the surface.
RefreshRate	Rate of refresh
SCaps	Capabilities.of the surface.
SrcBltColorkey	Source blit color key
SrcOverlayColorkey	Source overlay color key.
Width	Width of surface.
ZBufferBitDepth	Bit depth of z-buffer.

See Also

Methods

The height of a surface.

Syntax

objectname. **Height =** [Height]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

The height of a surface should not be specified if the surface being created is the primary surface. If the surface being created is an offscreen surface and if its height is greater than the primary surface, then it has to be created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwHeight member of the DDSURFACEDESC structure.

The width of a surface.

Syntax

objectname. **Width =** [Width]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

The width of a surface should not be specified if the surface being created is the primary surface. If the surface being created is an offscreen surface and if its width is greater than the primary surface, then it has to be created in system memory.

Direct Draw Compatibility

This property is the equivalent to the dwWidth member of the DDSURFACEDESC structure.

To the start of the next line

Syntax

objectname. **Pitch**

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

In the case of rectangular memory, the pitch of the display memory includes the width of the bitmap plus part of a cache.

Direct Draw Compatibility

This property is the equivalent to the dwPitch member of the DDSURFACEDESC structure.

The number of back buffers.

Syntax

objectname. **BackBufferCount =** [BackBufferCount]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

A back buffer is a nonvisible surface used for loading bitmaps and other images

Direct Draw Compatibility

This property is the equivalent to the dwBackBufferCount member of the DDSURFACEDESC structure.

The number of mipmap levels.

Syntax

objectname. **MipMapCount =** [MipMapCount]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

A mipmap is a sequence of rectangular arrays of pixels. Each mipmap level represents the same image but the resolution decreases as the mipmap level increases.

Direct Draw Compatibility

This property is the equivalent to the dwMipMapCount member of the DDSURFACEDESC structure.

The bit depth of the z-buffer.

Syntax

objectname. **ZBufferBitDepth =** [ZBufferBitDepth]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

A z-buffer is a buffer that stores the depth of each pixel in a scene.

Direct Draw Compatibility

This property is the equivalent to the dwZBufferBitDepth member of the DDSURFACEDESC structure.

The refresh rate.

Syntax

objectname. **RefreshRate =** [RefreshRate]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

This property is used when the display mode is described

Direct Draw Compatibility

This property is the equivalent to the dwRefreshRate member of the DDSURFACEDESC structure.

The bit depth of alpha buffer.

Syntax

objectname. **AlphaBitDepth =** [AlphaBitDepth]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

The alpha buffer specifies the opacity or clarity of a surface.

Direct Draw Compatibility

This property is the equivalent to the dwAlphaBitDepth member of the DDSURFACEDESC structure.

The destination overlay color key.

Syntax

objectname. **DestOverlayColorkey =** [DestOverlayColorkey]

objectname is the name of the **iSurfaceDesc** object.

Type

object

Remarks

In the case of overlays, destination color keying specifies a color or color range that is not shown on the destination.

Direct Draw Compatibility

This property is the equivalent to the ddckCKDestOverlay member of the DDSURFACEDESC structure.

The destination blit color key.

Syntax

objectname. **DestBltColorKey =** [DestBltColorKey]

objectname is the name of the **iSurfaceDesc** object.

Type

object

Remarks

In the case of blitting, destination color keying specifies a color or color range that is overwritten on the destination.

Direct Draw Compatibility

This property is the equivalent to the ddckCKDestBlt member of the DDSURFACEDESC structure.

The source overlay color key.

Syntax

objectname. **SrcOverlayColorkey =** [SrcOverlayColorkey]

objectname is the name of the **iSurfaceDesc** object.

Type

object

Remarks

In the case of overlays, source color keying specifies a color or color range that is not seen on the destination.

Direct Draw Compatibility

This property is the equivalent to the ddckCKSrcOverlay member of the DDSURFACEDESC structure.

The source blit color key.

Syntax

objectname. **SrcBltColorKey =** [SrcBltColorKey]

objectname is the name of the **iSurfaceDesc** object.

Type

object

Remarks

In the case of blitting, source color keying specifies a color or color range that is not replaced from the source onto the destination.

Direct Draw Compatibility

This property is the equivalent to the ddckCKSrcBlt member of the DDSURFACEDESC structure.

The pixel format description of the surface.

Syntax

objectname. **Pixelformat =** [Pixelformat]

objectname is the name of the **iSurfaceDesc** object.

Type

object

Remarks

This property is an iPixelFormat object which is used to set the color and pixel formats of surfaces.

Direct Draw Compatibility

This property is the equivalent to the ddpfPixelFormat member of the DDSURFACEDESC structure.

The capabilities of the surface.

Syntax

objectname.**SCaps =** [SCaps]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

A surface can be an alpha surface, a back buffer, a front buffer, a complex surface, a mipmap level, an offscreen surface, an overlay or a primary surface.

Direct Draw Compatibility

This property is the equivalent to the ddsCaps member of the DDSURFACEDESC structure.

The control flags.

Syntax

objectname. **Flags =** [Flags]

objectname is the name of the **iSurfaceDesc** object.

Type

long

Remarks

The control flags are optional.

Direct Draw Compatibility

This property is the equivalent to the Flags member of the DDSURFACEDESC structure.

Refresh	Resets all member variables to 0 or NULL as applicable so that the object can be reused.

See Also

Properties

This method can be used to reset all member variables to 0 or NULL so that the object can be reused.

Syntax

objectname.**Refresh** ()

objectname is the name of the **iSurfaceDesc** object.

Return Value

NONE

Parameters

None

Remarks

This method allows the iSurfaceDesc object to be reused in your code. For example if you are creating two surfaces with different characteristics, you can reuse the same iSurfaceDesc object when creating both surfaces.

Direct Draw Compability

This method has no equivalent in Direct Draw.

DriverDescription	The description of the device driver
DriverName	The name of the device driver

The name of the device driver

Syntax

objectname. **DriverName**

objectname is the name of the **iDevice** object.

Type

long

Remarks

The iDevice object contains information about a device driver which includes its name and description

Direct Draw Compatibility

This property has no equivalent in Direct Draw.

The description of the device driver

Syntax

objectname. **DriverDescription**

objectname is the name of the **iDevice** object.

Type

long

Remarks

The iDevice object contains information about a device driver which includes its name and description

Direct Draw Compatibility

This property has no equivalent in Direct Draw.

Blue[nIndex]	Specifies a blue intensity
Flag[nIndex]	Specifies how the color table is used
Green[nIndex]	Specifies a green intensity
Red[nIndex]	Specifies a red intensity
SurfacePalette[nIndex]	The palette entry for the specifed index
Type	Specifies the type of the color table.

Specifes the number of entries in the color table

Syntax

objectname.**Type**

objectname is the name of the **iColorTable** object.

Type

long

Remarks

Specifies the number of entries in the color table. May be set to one of :

DDPCAPS_1BIT	1 bit Palette. There are 2 entries in the color table.
DDPCAPS_2BIT	2 bit palette. There are 4 entries in the color table.
DDPCAPS_4BIT	4 bit palette. There are 16 entries in the color table.
DDPCAPS_8BITENTRIES	The color table entries are indexes to a target surface's 8-bit palette. This flag is valid only when used with the DDPCAPS_1BIT, DDPCAPS_2BIT, or DDPCAPS_4BIT flag, and when the target surface is 8 bits per pixel.
DDPCAPS_8BIT	8 bit palette. There are 256 entries in the color table.
DDPCAPS_ALLOW256	Palette can have all 256 entries defined.

Direct Draw Compatibility

This property has no equivalent in Direct Draw. However it has the same meaning as the dwFlags parameter to IDirectDraw2::CreatePalette()

The red intensity for the specified index.

Syntax

objectname. **Red[nIndex] =** [Red[nIndex]]

objectname is the name of the **iColorTable** object.

Type

short

Remarks

Get or set the red intensity of a color table entry. Only applies if Type is set to DDPCAPS_1BIT, DDPCAPS_2BIT, DDPCAPS_4BIT, DDPCAPS_8BIT or DDPCAPS_ALLOW256

Direct Draw Compatibility

This property is equivalent to the peRed member of the PALETTEENTRY structure.

The green intensity for the specified index.

Syntax

objectname. **Green[nIndex] =** [Green[nIndex]]

objectname is the name of the **iColorTable** object.

Type

short

Remarks

Get or set the green intensity of a color table entry. Only applies if Type is set to DDPCAPS_1BIT, DDPCAPS_2BIT, DDPCAPS_4BIT, DDPCAPS_8BIT or DDPCAPS_ALLOW256

Direct Draw Compatibility

This property is equivalent to the peGreen member of the PALETTEENTRY structure.

The blue intensity for the specified index.

Syntax

objectname. **Blue[nIndex] =** [Blue[nIndex]]

objectname is the name of the **iColorTable** object.

Type

short

Remarks

Get or set the blue intensity of a color table entry. Only applies if Type is set to DDPCAPS_1BIT, DDPCAPS_2BIT, DDPCAPS_4BIT, DDPCAPS_8BIT or DDPCAPS_ALLOW256

Direct Draw Compatibility

This property is equivalent to the peBlue member of the PALETTEENTRY structure.

How the palette entry is to be used for the specified index.

Syntax

objectname. **Flag[nIndex] =** [Flag[nIndex]]

objectname is the name of the **iColorTable** object.

Type

short

Remarks

This property is not used and is provided only for compatibility. Either set it to NULL explicitly or use the default value.

Direct Draw Compatibility

This property is equivalent to the peFlag member of the PALETTEENTRY structure.

The index into the surface palette.

Syntax

objectname. **SurfacePalette[nIndex] =** [SurfacePalette[nIndex]]

objectname is the name of the **iColorTable** object.

Type

short

Remarks

Associates a color table entry with the surface palette entry. Only applies if the Type is set to DDPCAPS_8BITENTRIES.

For example, by setting SurfacePalette [0] = 128, you are specifying that the color table entry 0 is the same color as the surface palette entry 128.

Do not use this property together with the Red, Green and Blue properties.

Direct Draw Compatibility

This property has no equivalent in Direct Draw

In order to help you get started quickly with iDraw, we have provided sample programs that show how to use iDraw OCX.

There are samples for Visual Basic as well as Visual C++

In the installed samples directory you will find the following :

Samples Directory

VB		- contains Visual Basic samples
	Flip	- demonstrates flipping between surfaces
	Blit	- demonstrates blitting to the primary surface
	Caps	- demonstrates how to get devices capabilities
	Pal	- demonstrates use of palettes
MFC		- contains Visual C++ / MFC samples
	Flip	- demonstrates flipping between surfaces
	Blit	- demonstrates blitting to the primary surface
	Caps	- demonstrates how to get devices capabilities
	Pal	- demonstrates use of palettes