Namespace VMwareSvgaII3D

Classes

SVGAII3DCanvas

Defines a VMWare SVGAII canvas implementation. Please note that this implementation of Cosmos.System.Graphics.Canvas can only be used with virtualizers that do implement SVGAII, meaning that this class will not work on regular hardware.

VMWareSVGAII3D

Structs

SVGA3dArray

SVGA3dArrayRangeHint

SVGA3dCmdClear

SVGA3dCmdDefineContext

SVGA3dCmdDefineShader

SVGA3dCmdDefineSurface

SVGA3dCmdDrawPrimitives

SVGA3dCmdHeader

SVGA3dCmdPresent

SVGA3dCmdSetRenderState

SVGA3dCmdSetRenderTarget

SVGA3dCmdSetShader

SVGA3dCmdSetShaderConst

SVGA3dCmdSetTextureState

SVGA3dCmdSetTransform

SVGA3dCmdSetViewport

SVGA3dCmdSetZRange

SVGA3dCmdSurfaceDMA

SVGA3dCopyBox

SVGA3dCopyRect

SVGA3dGuestImage

SVGA3dPrimitiveRange

SVGA3dRect

SVGA3dRenderState

SVGA3dSize

SVGA3dSurfaceImageId

SVGA3dTextureState

SVGA3dVertexArrayIdentity

SVGA3dVertexDecl

SVGA3dZRange

SVGAGuestPtr

Enums

Capability

ClearFlags

FIFOCommand

Register

Register3D

SVGA3dDeclMethod

SVGA3dDeclType

SVGA3dDeclUsage

SVGA3dPrimitiveType

Enum Capability

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
[Flags]
public enum Capability
```

Fields

`AlphaBlend = 8192`

Alpha blend.

`AlphaCursor = 512`

Alpha cursor.

`Cap3D = 16384`

`Cursor = 32`

Cruser.

`CursorByPass = 64`

Cursor bypass.

`CursorByPass2 = 128`

Cursor bypass2.

`DisplayTopology = 524288`

Display topology.

`EigthBitEmulation = 256`

Eigth bit emulation.

`ExtendedFifo = 32768`

Extended FIFO.

`Glyph = 1024`

Glyph.

`GlyphClipping = 2048`

Glyph clipping.

`Gmr = 1048576`

GMR.

`Gmr2 = 4194304`

GMR2.

`IrqMask = 262144`

IRQ mask.

`LecacyOffscreen = 8`

Lecacy off screen.

`MultiMon = 65536`

Multi monitors.

`None = 0`

None.

`Offscreen1 = 4096`

Offscreen.

`PitchLock = 131072`

Pitch lock.

`RasterOp = 16`

Raster operation.

`RectCopy = 2`

Rectangle copy.

`RectFill = 1`

Rectangle fill.

`RectPatFill = 4`

Rectangle pattern fill.

`ScreenObject2 = 8388608`

Screen objects.

`SurfaceCopy = 65536`

`ThreeD = 16384`

Three D.

`Traces = 2097152`

Traces.

Enum ClearFlags

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
[Flags]
public enum ClearFlags : uint
```

Fields

Color = 1

Depth = 2

Stencil = 4

Enum FIFOCommand

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum FIFOCommand
```

Fields

CLEAR = 1057

DEFINE_ALPHA_CURSOR = 22

Define alpha cursor.

DEFINE_BITMAP = 4

Define bitmap.

DEFINE_BITMAP_SCANLINE = 5

Define bitmap scanline.

DEFINE_CONTEXT = 1045

DEFINE_CURSOR = 19

Define cursor.

DEFINE_PIXMAP = 6

Define pixmap.

DEFINE_PIXMAP_SCANLINE = 7

Define pixmap scanline.

DEFINE_SURFACE = 1040

DEFINE_SURFACE_V2 = 1070

DESTROY_CONTEXT = 1046

DESTROY_SURFACE = 1041

`DISPLAY_CURSOR = 20`

Display cursor.

`DRAW_PRIMITIVES = 1063`

`FREE_OBJECT = 12`

Free object.

`MOVE_CURSOR = 21`

Move cursor.

`PRESENT = 1058`

`RECT_BITMAP_COPY = 10`

Rectange bitmap copy.

`RECT_BITMAP_FILL = 8`

Rectange bitmap fill.

`RECT_COPY = 3`

Rectange copy.

`RECT_FILL = 2`

Rectange fill.

`RECT_PIXMAP_COPY = 11`

Rectange pixmap fill.

`RECT_PIXMAP_FILL = 9`

Rectange pixmap fill.

`RECT_ROP_BITMAP_COPY = 17`

Rectangle raster operation bitmap copy.

`RECT_ROP_BITMAP_FILL = 15`

Rectangle raster operation bitmap fill.

RECT_ROP_COPY = 14

Rectangle raster operation copy.

RECT_ROP_FILL = 13

Rectangle raster operation fill.

RECT_ROP_PIXMAP_COPY = 18

Rectangle raster operation pixmap copy.

RECT_ROP_PIXMAP_FILL = 16

Rectangle raster operation pixmap fill.

SETRENDERSTATE = 1049

SETTEXTURESTATE = 1051

SETTRANSFORM = 1047

SETVIEWPORT = 1055

SETZRANGE = 1048

SET_RENDER_TARGET = 1050

SET_SHADER = 1061

SET_SHADER_CONST = 1062

SET_VIEWPORT = 1055

SET_ZRANGE = 1048

SHADER_DEFINE = 1059

SURFACE_COPY = 1042

SURFACE_DMA = 1044

Update = 1

Update.

Enum Register

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum Register : ushort
```

Fields

BitsPerPixel = 7

BlueMask = 11

Busy = 22

BytesPerLine = 12

Capabilities = 17

Capabilities3D = 34

ConfigDone = 20

CursorCount = 12

CursorID = 24

CursorOn = 27

CursorX = 25

CursorY = 26

Depth = 6

Enable = 1

Enable3D = 32

FifoNumRegisters = 293

FrameBufferOffset = 14

```
FrameBufferSize = 16

FrameBufferStart = 13

GreenMask = 10

Guest3DScratchSize = 33

GuestID = 23

Height = 3

HostBitsPerPixel = 28

ID = 0

MaxHeight = 5

MaxWidth = 4

MemRegs = 30

MemSize = 19

MemStart = 18

NumDisplays = 31

PitchLock = 32

PseudoColor = 8

RedMask = 9

ScratchSize = 29

Sync = 21

VRamSize = 15

Width = 2
```

Enum Register3D

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum Register3D
```

Fields

SVGA_FIFO_3D_CAPS = 32

SVGA_FIFO_3D_CAPS_LAST = 287

SVGA_FIFO_3D_HWVERSION = 7

SVGA_FIFO_3D_HWVERSION_REVISED = 17

SVGA_FIFO_BUSY = 290

SVGA_FIFO_CAPABILITIES = 4

SVGA_FIFO_CURSOR_COUNT = 12

SVGA_FIFO_CURSOR_LAST_UPDATED = 13

SVGA_FIFO_CURSOR_ON = 9

SVGA_FIFO_CURSOR_SCREEN_ID = 15

SVGA_FIFO_CURSOR_X = 10

SVGA_FIFO_CURSOR_Y = 11

SVGA_FIFO_DEAD = 16

SVGA_FIFO_FENCE = 6

SVGA_FIFO_FENCE_GOAL = 289

SVGA_FIFO_FLAGS = 5

SVGA_FIFO_GUEST_3D_HWVERSION = 288

```
SVGA_FIFO_MAX = 1

SVGA_FIFO_MIN = 0

SVGA_FIFO_NEXT_CMD = 2

SVGA_FIFO_NUM_REGS = 291

SVGA_FIFO_PITCHLOCK = 8

SVGA_FIFO_RESERVED = 14

SVGA_FIFO_STOP = 3
```

Struct SVGA3dArray

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dArray
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

offset

```
public uint offset
```

Field Value

uint

stride

```
public uint stride
```

Field Value

uint

surfaceId

```
public uint surfaceId
```

Field Value

uint

Struct SVGA3dArrayRangeHint

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dArrayRangeHint
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

first

```
public uint first
```

Field Value

uint

last

```
public uint last
```

Field Value

uint

Struct SVGA3dCmdClear

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdClear
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

color

```
public uint color
```

Field Value

uint

depth

```
public float depth
```

Field Value

float

flag

```
public ClearFlags flag
```

Field Value

ClearFlags

stencil

```
public uint stencil
```

Field Value

uint

Struct SVGA3dCmdDefineContext

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdDefineContext
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

Struct SVGA3dCmdDefineShader

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdDefineShader
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

shid

```
public uint shid
```

Field Value

uint

type

```
public SVGA3dShaderType type
```

Field Value

SVGA3dShaderType

Struct SVGA3dCmdDefineSurface

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdDefineSurface
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

face

```
public uint* face
```

Field Value

uint*

flags

```
public SVGA3dSurfaceFlags flags
```

Field Value

SVGA3dSurfaceFlags

format

```
public SVGA3dSurfaceFormat format
```

Field Value

SVGA3dSurfaceFormat

sid

```
public uint sid
```

Field Value

uint

Struct SVGA3dCmdDrawPrimitives

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdDrawPrimitives
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

numRanges

```
public uint numRanges
```

Field Value

uint

numVertexDecls

```
public uint numVertexDecls
```

Field Value

uint

Struct SVGA3dCmdHeader

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdHeader
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

id

```
public uint id
```

Field Value

uint

size

```
public uint size
```

Field Value

uint

Struct SVGA3dCmdPresent

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdPresent
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

sid

```
public uint sid
```

Field Value

uint

Struct SVGA3dCmdSetRenderState

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetRenderState
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

Struct SVGA3dCmdSetRenderTarget

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetRenderTarget
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

target

```
public SVGA3dSurfaceImageId target
```

Field Value

SVGA3dSurfaceImageId

type

```
public SVGA3dRenderTargetType type
```

Field Value

SVGA3dRenderTargetType

Struct SVGA3dCmdSetShader

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetShader
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

shid

```
public uint shid
```

Field Value

uint

type

```
public SVGA3dShaderType type
```

Field Value

SVGA3dShaderType

Struct SVGA3dCmdSetShaderConst

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetShaderConst
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

ctype

```
public SVGA3dShaderConstType ctype
```

Field Value

SVGA3dShaderConstType

reg

```
public uint reg
```

Field Value

uint

type

```
public SVGA3dShaderType type
```

Field Value

SVGA3dShaderType

values

```
public uint* values
```

Field Value

uint*

Struct SVGA3dCmdSetTextureState

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetTextureState
```

Inherited Members

ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

Struct SVGA3dCmdSetTransform

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetTransform
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

matrix

```
public float* matrix
```

Field Value

float*

type

```
public SVGA3dTransformType type
```

Field Value

SVGA3dTransformType

Struct SVGA3dCmdSetViewport

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetViewport
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

rect

```
public SVGA3dRect rect
```

Field Value

SVGA3dRect

Struct SVGA3dCmdSetZRange

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSetZRange
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

cid

```
public uint cid
```

Field Value

uint

range

```
public SVGA3dZRange range
```

Field Value

SVGA3dZRange

Struct SVGA3dCmdSurfaceDMA

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCmdSurfaceDMA
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

guest

```
public SVGA3dGuestImage guest
```

Field Value

SVGA3dGuestImage

host

```
public SVGA3dSurfaceImageId host
```

Field Value

SVGA3dSurfaceImageId

transfer

```
public SVGA3dTransferType transfer
```

Field Value

SVGA3dTransferType

Struct SVGA3dCopyBox

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCopyBox
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

d

```
public uint d
```

Field Value

uint

h

```
public uint h
```

Field Value

uint

srcx

```
public uint srcx
```

Field Value

uint

srcy

```
public uint srcy
```

Field Value

uint

srcz

```
public uint srcz
```

Field Value

uint

w

```
public uint w
```

Field Value

uint

x

```
public uint x
```

Field Value

uint

y

```
public uint y
```

Field Value

uint

z

```
public uint z
```

Field Value

uint

Struct SVGA3dCopyRect

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dCopyRect
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

h

```
public uint h
```

Field Value

uint

srcx

```
public uint srcx
```

Field Value

uint

srcy

```
public uint srcy
```

Field Value

uint

w

```
public uint w
```

Field Value

uint

x

```
public uint x
```

Field Value

uint

y

```
public uint y
```

Field Value

uint

Enum SVGA3dDeclMethod

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dDeclMethod
```

Fields

SVGA3D_DECLMETHOD_CROSSUV = 3

SVGA3D_DECLMETHOD_DEFAULT = 0

SVGA3D_DECLMETHOD_LOOKUP = 5

SVGA3D_DECLMETHOD_LOOKUPPRESAMPLED = 6

SVGA3D_DECLMETHOD_PARTIALU = 1

SVGA3D_DECLMETHOD_PARTIALV = 2

SVGA3D_DECLMETHOD_UV = 4

Enum SVGA3dDeclType

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dDeclType
```

Fields

SVGA3D_DECLTYPE_D3DCOLOR = 4

SVGA3D_DECLTYPE_DEC3N = 14

SVGA3D_DECLTYPE_FLOAT1 = 0

SVGA3D_DECLTYPE_FLOAT16_2 = 15

SVGA3D_DECLTYPE_FLOAT16_4 = 16

SVGA3D_DECLTYPE_FLOAT2 = 1

SVGA3D_DECLTYPE_FLOAT3 = 2

SVGA3D_DECLTYPE_FLOAT4 = 3

SVGA3D_DECLTYPE_MAX = 17

SVGA3D_DECLTYPE_SHORT2 = 6

SVGA3D_DECLTYPE_SHORT2N = 9

SVGA3D_DECLTYPE_SHORT4 = 7

SVGA3D_DECLTYPE_SHORT4N = 10

SVGA3D_DECLTYPE_UBYTE4 = 5

SVGA3D_DECLTYPE_UBYTE4N = 8

SVGA3D_DECLTYPE_UDEC3 = 13

SVGA3D_DECLTYPE_USHORT2N = 11

```
SVGA3D_DECLTYPE_USHORT4N = 12
```

Enum SVGA3dDeclUsage

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dDeclUsage
```

Fields

SVGA3D_DECLUSAGE_BINORMAL = 7

SVGA3D_DECLUSAGE_BLENDINDICES = 2

SVGA3D_DECLUSAGE_BLENDWEIGHT = 1

SVGA3D_DECLUSAGE_COLOR = 10

SVGA3D_DECLUSAGE_DEPTH = 12

SVGA3D_DECLUSAGE_FOG = 11

SVGA3D_DECLUSAGE_MAX = 14

SVGA3D_DECLUSAGE_NORMAL = 3

SVGA3D_DECLUSAGE_POSITION = 0

SVGA3D_DECLUSAGE_POSITIONT = 9

SVGA3D_DECLUSAGE_PSIZE = 4

SVGA3D_DECLUSAGE_SAMPLE = 13

SVGA3D_DECLUSAGE_TANGENT = 6

SVGA3D_DECLUSAGE_TESSFACTOR = 8

SVGA3D_DECLUSAGE_TEXCOORD = 5

Struct SVGA3dGuestImage

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dGuestImage
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

pitch

```
public float pitch
```

Field Value

float

ptr

```
public SVGAGuestPtr ptr
```

Field Value

SVGAGuestPtr

Struct SVGA3dPrimitiveRange

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dPrimitiveRange
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

indexArray

```
public SVGA3dArray indexArray
```

Field Value

SVGA3dArray

indexBias

```
public int indexBias
```

Field Value

int

indexWidth

```
public uint indexWidth
```

Field Value

uint

primType

```
public SVGA3dPrimitiveType primType
```

Field Value

SVGA3dPrimitiveType

primitiveCount

```
public uint primitiveCount
```

Field Value

uint

Enum SVGA3dPrimitiveType

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dPrimitiveType
```

Fields

SVGA3D_PRIMITIVE_INVALID = 0

SVGA3D_PRIMITIVE_LINELIST = 3

SVGA3D_PRIMITIVE_LINESTRIP = 4

SVGA3D_PRIMITIVE_MAX = 7

SVGA3D_PRIMITIVE_POINTLIST = 2

SVGA3D_PRIMITIVE_TRIANGLEFAN = 6

SVGA3D_PRIMITIVE_TRIANGLELIST = 1

SVGA3D_PRIMITIVE_TRIANGLESTRIP = 5

Struct SVGA3dRect

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dRect
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Constructors

SVGA3dRect(uint, uint, uint, uint)

```
public SVGA3dRect(uint x, uint y, uint w, uint h)
```

Parameters

x uint

y uint

w uint

h uint

Fields

h

```
public uint h
```

Field Value

uint

w

```
public uint w
```

Field Value

uint

x

```
public uint x
```

Field Value

uint

y

```
public uint y
```

Field Value

uint

Struct SVGA3dRenderState

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dRenderState
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Constructors

SVGA3dRenderState(SVGA3dRenderStateName, float)

```
public SVGA3dRenderState(SVGA3dRenderStateName State, float value)
```

Parameters

`State` SVGA3dRenderStateName

`value` float

SVGA3dRenderState(SVGA3dRenderStateName, uint)

```
public SVGA3dRenderState(SVGA3dRenderStateName State, uint value)
```

Parameters

`State` SVGA3dRenderStateName

`value` uint

Fields

floatValue

```
public float floatValue
```

Field Value

float

state

```
public SVGA3dRenderStateName state
```

Field Value

SVGA3dRenderStateName

uintValue

```
public uint uintValue
```

Field Value

uint

Enum SVGA3dRenderStateName

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dRenderStateName : uint
```

Fields

SVGA3D_RS_ALPHAFUNC = 37

SVGA3D_RS_ALPHAREF = 42

SVGA3D_RS_ALPHATESTENABLE = 3

SVGA3D_RS_AMBIENT = 26

SVGA3D_RS_AMBIENTMATERIALSOURCE = 51

SVGA3D_RS_ANTIALIASEDLINEENABLE = 89

SVGA3D_RS_BLENDCOLOR = 56

SVGA3D_RS_BLENDENABLE = 5

SVGA3D_RS_BLENDEQUATION = 34

SVGA3D_RS_BLENDEQUATIONALPHA = 96

SVGA3D_RS_CCWSTENCILFAIL = 59

SVGA3D_RS_CCWSTENCILFUNC = 58

SVGA3D_RS_CCWSTENCILPASS = 61

SVGA3D_RS_CCWSTENCILZFAIL = 60

SVGA3D_RS_CLIPPING = 68

SVGA3D_RS_CLIPPLANEENABLE = 27

SVGA3D_RS_COLORWRITEENABLE = 47

SVGA3D_RS_COLORWRITEENABLE1 = 90

SVGA3D_RS_COLORWRITEENABLE2 = 91

SVGA3D_RS_COLORWRITEENABLE3 = 92

SVGA3D_RS_COORDINATETYPE = 44

SVGA3D_RS_CULLMODE = 35

SVGA3D_RS_DEPTHBIAS = 64

SVGA3D_RS_DIFFUSEMATERIALSOURCE = 49

SVGA3D_RS_DITHERENABLE = 4

SVGA3D_RS_DSTBLEND = 33

SVGA3D_RS_DSTBLENDALPHA = 95

SVGA3D_RS_EMISSIVEMATERIALSOURCE = 52

SVGA3D_RS_FILLMODE = 29

SVGA3D_RS_FOGCOLOR = 25

SVGA3D_RS_FOGDENSITY = 18

SVGA3D_RS_FOGENABLE = 6

SVGA3D_RS_FOGEND = 17

SVGA3D_RS_FOGMODE = 28

SVGA3D_RS_FOGSTART = 16

SVGA3D_RS_FRONTWINDING = 43

SVGA3D_RS_INDEXEDVERTEXBLENDENABLE = 87

SVGA3D_RS_INVALID = 0

SVGA3D_RS_LASTPIXEL = 67

SVGA3D_RS_LIGHTINGENABLE = 9

SVGA3D_RS_LINEAA = 98

SVGA3D_RS_LINEPATTERN = 31

SVGA3D_RS_LINEWIDTH = 99

SVGA3D_RS_LOCALVIEWER = 54

SVGA3D_RS_MAX = 100

SVGA3D_RS_MULTISAMPLEANTIALIAS = 85

SVGA3D_RS_MULTISAMPLEMASK = 86

SVGA3D_RS_NORMALIZENORMALS = 10

SVGA3D_RS_OUTPUTGAMMA = 65

SVGA3D_RS_POINTSCALEENABLE = 12

SVGA3D_RS_POINTSCALE_A = 22

SVGA3D_RS_POINTSCALE_B = 23

SVGA3D_RS_POINTSCALE_C = 24

SVGA3D_RS_POINTSIZE = 19

SVGA3D_RS_POINTSIZEMAX = 21

SVGA3D_RS_POINTSIZEMIN = 20

SVGA3D_RS_POINTSPRITEENABLE = 11

SVGA3D_RS_RANGEFOGENABLE = 46

SVGA3D_RS_SCISSORTESTENABLE = 55

SVGA3D_RS_SEPARATEALPHABLENDENABLE = 93

SVGA3D_RS_SHADEMODE = 30

SVGA3D_RS_SLOPESCALEDEPTHBIAS = 63

SVGA3D_RS_SPECULARENABLE = 7

SVGA3D_RS_SPECULARMATERIALSOURCE = 50

SVGA3D_RS_SRCBLEND = 32

SVGA3D_RS_SRCBLENDALPHA = 94

SVGA3D_RS_STENCILENABLE = 8

SVGA3D_RS_STENCILENABLE2SIDED = 57

SVGA3D_RS_STENCILFAIL = 39

SVGA3D_RS_STENCILFUNC = 38

SVGA3D_RS_STENCILMASK = 14

SVGA3D_RS_STENCILPASS = 41

SVGA3D_RS_STENCILREF = 13

SVGA3D_RS_STENCILWRITEMASK = 15

SVGA3D_RS_STENCILZFAIL = 40

SVGA3D_RS_TEXTUREFACTOR = 53

SVGA3D_RS_TRANSPARENCYANTIALIAS = 97

SVGA3D_RS_TWEENFACTOR = 88

SVGA3D_RS_VERTEXBLEND = 62

SVGA3D_RS_VERTEXMATERIALENABLE = 48

SVGA3D_RS_WRAP0 = 69

SVGA3D_RS_WRAP1 = 70

SVGA3D_RS_WRAP10 = 79

SVGA3D_RS_WRAP11 = 80

SVGA3D_RS_WRAP12 = 81

SVGA3D_RS_WRAP13 = 82

SVGA3D_RS_WRAP14 = 83

SVGA3D_RS_WRAP15 = 84

SVGA3D_RS_WRAP2 = 71

```
SVGA3D_RS_WRAP3 = 72

SVGA3D_RS_WRAP4 = 73

SVGA3D_RS_WRAP5 = 74

SVGA3D_RS_WRAP6 = 75

SVGA3D_RS_WRAP7 = 76

SVGA3D_RS_WRAP8 = 77

SVGA3D_RS_WRAP9 = 78

SVGA3D_RS_ZBIAS = 45

SVGA3D_RS_ZENABLE = 1

SVGA3D_RS_ZFUNC = 36

SVGA3D_RS_ZVISIBLE = 66

SVGA3D_RS_ZWRITEENABLE = 2
```

Enum SVGA3dRenderTargetType

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dRenderTargetType : uint
```

Fields

`Color = 2`

`Depth = 0`

Enum SVGA3dShaderConstType

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dShaderConstType
```

Fields

SVGA3D_CONST_TYPE_BOOL = 2

SVGA3D_CONST_TYPE_FLOAT = 0

SVGA3D_CONST_TYPE_INT = 1

Enum SVGA3dShaderType

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dShaderType
```

Fields

SVGA3D_SHADERTYPE_MAX = 3

SVGA3D_SHADERTYPE_PS = 2

SVGA3D_SHADERTYPE_VS = 1

Struct SVGA3dSize

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dSize
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

depth

```
public uint depth
```

Field Value

uint

height

```
public uint height
```

Field Value

uint

width

```
public uint width
```

Field Value

uint

Enum SVGA3dSurfaceFlags

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dSurfaceFlags : uint
```

Fields

SVGA3D_SURFACE_AUTOGENMIPMAPS = 1024

SVGA3D_SURFACE_CUBEMAP = 1

SVGA3D_SURFACE_HINT_DEPTHSTENCIL = 128

SVGA3D_SURFACE_HINT_DYNAMIC = 4

SVGA3D_SURFACE_HINT_INDEXBUFFER = 8

SVGA3D_SURFACE_HINT_RENDERTARGET = 64

SVGA3D_SURFACE_HINT_STATIC = 2

SVGA3D_SURFACE_HINT_TEXTURE = 32

SVGA3D_SURFACE_HINT_VERTEXBUFFER = 16

SVGA3D_SURFACE_HINT_WRITEONLY = 256

SVGA3D_SURFACE_MASKABLE_ANTIALIAS = 512

Enum SVGA3dSurfaceFormat

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dSurfaceFormat : uint
```

Fields

SVGA3D_A16B16G16R16 = 41

SVGA3D_A2R10G10B10 = 26

SVGA3D_A2W10V10U10 = 31

SVGA3D_A8R8G8B8 = 2

SVGA3D_ALPHA8 = 32

SVGA3D_ARGB_S10E5 = 24

SVGA3D_ARGB_S23E8 = 25

SVGA3D_AYUV = 45

SVGA3D_BC4_UNORM = 108

SVGA3D_BC5_UNORM = 111

SVGA3D_BUFFER = 37

SVGA3D_BUMPL6V5U5 = 21

SVGA3D_BUMPL8V8U8 = 23

SVGA3D_BUMPU8V8 = 20

SVGA3D_BUMPX8L8V8U8 = 22

SVGA3D_CxV8U8 = 29

SVGA3D_DXT1 = 15

```
SVGA3D_DXT2 = 16

SVGA3D_DXT3 = 17

SVGA3D_DXT4 = 18

SVGA3D_DXT5 = 19

SVGA3D_FORMAT_INVALID = 0

SVGA3D_FORMAT_MAX = 121

SVGA3D_G16R16 = 40

SVGA3D_LUMINANCE16 = 13

SVGA3D_LUMINANCE4_ALPHA4 = 12

SVGA3D_LUMINANCE8 = 11

SVGA3D_LUMINANCE8_ALPHA8 = 14

SVGA3D_NV12 = 44

SVGA3D_Q8W8V8U8 = 28

SVGA3D_RG_S10E5 = 35

SVGA3D_RG_S23E8 = 36

SVGA3D_R_S10E5 = 33

SVGA3D_R_S23E8 = 34

SVGA3D_UYVY = 42

SVGA3D_V16U16 = 39

SVGA3D_V8U8 = 27

SVGA3D_X8L8V8U8 = 30

SVGA3D_X8R8G8B8 = 1

SVGA3D_YUY2 = 43

SVGA3D_Z_D16 = 8
```

```
SVGA3D_Z_D24S8_INT = 120

SVGA3D_Z_D24X8 = 38

SVGA3D_Z_D32 = 7

SVGA3D_Z_DF16 = 118

SVGA3D_Z_DF24 = 119
```

Struct SVGA3dSurfaceImageId

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dSurfaceImageId
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

face

```
public uint face
```

Field Value

uint

mipmap

```
public uint mipmap
```

Field Value

uint

sid

```
public uint sid
```

Field Value

uint

Struct SVGA3dTextureState

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dTextureState
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Constructors

SVGA3dTextureState(SVGA3dTextureStateName, float, uint)

```
public SVGA3dTextureState(SVGA3dTextureStateName State, float value, uint stage = 0)
```

Parameters

`State` SVGA3dTextureStateName

`value` float

`stage` uint

SVGA3dTextureState(SVGA3dTextureStateName, uint, uint)

```
public SVGA3dTextureState(SVGA3dTextureStateName State, uint value, uint stage = 0)
```

Parameters

`State` SVGA3dTextureStateName

`value` uint

`stage` uint

Fields

floatValue

```
public float floatValue
```

Field Value

float

stage

```
public uint stage
```

Field Value

uint

state

```
public SVGA3dTextureStateName state
```

Field Value

SVGA3dTextureStateName

value

```
public uint value
```

Field Value

uint

Enum SVGA3dTextureStateName

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dTextureStateName : uint
```

Fields

SVGA3D_TS_ADDRESSU = 8

SVGA3D_TS_ADDRESSV = 9

SVGA3D_TS_ADDRESSW = 24

SVGA3D_TS_ALPHAARG0 = 29

SVGA3D_TS_ALPHAARG1 = 6

SVGA3D_TS_ALPHAARG2 = 7

SVGA3D_TS_ALPHAOP = 5

SVGA3D_TS_BIND_TEXTURE = 1

SVGA3D_TS_BORDERCOLOR = 13

SVGA3D_TS_BUMPENVLOFFSET = 27

SVGA3D_TS_BUMPENVLSCALE = 26

SVGA3D_TS_BUMPENVMAT00 = 17

SVGA3D_TS_BUMPENVMAT01 = 18

SVGA3D_TS_BUMPENVMAT10 = 19

SVGA3D_TS_BUMPENVMAT11 = 20

SVGA3D_TS_COLORARG0 = 28

SVGA3D_TS_COLORARG1 = 3

SVGA3D_TS_COLORARG2 = 4

SVGA3D_TS_COLOROP = 2

SVGA3D_TS_GAMMA = 25

SVGA3D_TS_INVALID = 0

SVGA3D_TS_MAGFILTER = 11

SVGA3D_TS_MAX = 30

SVGA3D_TS_MINFILTER = 12

SVGA3D_TS_MIPFILTER = 10

SVGA3D_TS_TEXCOORDGEN = 16

SVGA3D_TS_TEXCOORDINDEX = 14

SVGA3D_TS_TEXTURETRANSFORMFLAGS = 15

SVGA3D_TS_TEXTURE_ANISOTROPIC_LEVEL = 23

SVGA3D_TS_TEXTURE_LOD_BIAS = 22

SVGA3D_TS_TEXTURE_MIPMAP_LEVEL = 21

Enum SVGA3dTransferType

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dTransferType
```

Fields

SVGA3D_READ_HOST_VRAM = 2

SVGA3D_WRITE_HOST_VRAM = 1

Enum SVGA3dTransformType

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public enum SVGA3dTransformType
```

Fields

SVGA3D_TRANSFORM_INVALID = 0

SVGA3D_TRANSFORM_MAX = 15

SVGA3D_TRANSFORM_PROJECTION = 3

SVGA3D_TRANSFORM_TEXTURE0 = 4

SVGA3D_TRANSFORM_TEXTURE1 = 5

SVGA3D_TRANSFORM_TEXTURE2 = 6

SVGA3D_TRANSFORM_TEXTURE3 = 7

SVGA3D_TRANSFORM_TEXTURE4 = 8

SVGA3D_TRANSFORM_TEXTURE5 = 9

SVGA3D_TRANSFORM_TEXTURE6 = 10

SVGA3D_TRANSFORM_TEXTURE7 = 11

SVGA3D_TRANSFORM_VIEW = 2

SVGA3D_TRANSFORM_WORLD = 1

SVGA3D_TRANSFORM_WORLD1 = 12

SVGA3D_TRANSFORM_WORLD2 = 13

SVGA3D_TRANSFORM_WORLD3 = 14

Struct SVGA3dVertexArrayIdentity

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dVertexArrayIdentity
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

method

```
public SVGA3dDeclMethod method
```

Field Value

SVGA3dDeclMethod

type

```
public SVGA3dDeclType type
```

Field Value

SVGA3dDeclType

usage

```
public SVGA3dDeclUsage usage
```

Field Value

SVGA3dDeclUsage

usageIndex

```
public uint usageIndex
```

Field Value

uint

Struct SVGA3dVertexDecl

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dVertexDecl
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

array

```
public SVGA3dArray array
```

Field Value

SVGA3dArray

identity

```
public SVGA3dVertexArrayIdentity identity
```

Field Value

SVGA3dVertexArrayIdentity

rangeHint

```
public SVGA3dArrayRangeHint rangeHint
```

Field Value

SVGA3dArrayRangeHint

Struct SVGA3dZRange

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGA3dZRange
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

max

```
public float max
```

Field Value

float

min

```
public float min
```

Field Value

float

Struct SVGAGuestPtr

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public struct SVGAGuestPtr
```

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.GetType() , object.ReferenceEquals(object, object)

Fields

gmrId

```
public uint gmrId
```

Field Value

uint

offset

```
public uint offset
```

Field Value

uint

Class SVGAII3DCanvas

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

Defines a VMWare SVGAII canvas implementation. Please note that this implementation of Cosmos.System.Graphics.Canvas can only be used with virtualizers that do implement SVGAII, meaning that this class will not work on regular hardware.

```
public class SVGAII3DCanvas : Canvas
```

Inheritance

object ← Canvas ← SVGAII3DCanvas

Inherited Members

Canvas.Clear() , Canvas.DrawLine(Color, int, int, int, int) , Canvas.DrawCircle(Color, int, int, int) , Canvas.DrawFilledCircle(Color, int, int, int) , Canvas.DrawEllipse(Color, int, int, int, int) , Canvas.DrawFilledEllipse(Color, int, int, int, int) , Canvas.DrawArc(int, int, int, int, Color, int, int) , Canvas.DrawPolygon(Color, params Point[]) , Canvas.DrawSquare(Color, int, int, int) , Canvas.DrawTriangle(Color, int, int, int, int, int, int) , Canvas.DrawImage(Image, int, int, int, int, bool) , Canvas.DrawImageAlpha(Image, int, int, bool) , Canvas.CheckIfModeIsValid(Mode) , Canvas.ThrowIfModeIsNotValid(Mode) , Canvas.ThrowIfCoordNotValid(int, int) , Canvas.TrimLine(ref int, ref int, ref int, ref int) , Canvas.AlphaBlend(Color, Color, byte) , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

SVGAII3DCanvas()

Initializes a new instance of the SVGAII3DCanvas class.

```
public SVGAII3DCanvas()
```

SVGAII3DCanvas(Mode)

Initializes a new instance of the SVGAII3DCanvas class.

```
public SVGAII3DCanvas(Mode aMode)
```

Parameters

`aMode` Mode

The graphics mode.

Fields

driver

```
public readonly VMWareSVGAII3D driver
```

Field Value

VMWareSVGAII3D

Properties

AvailableModes

The available graphics modes.

```
public override List<Mode> AvailableModes { get; }
```

Property Value

List <Mode>

DefaultGraphicsMode

The default graphics mode.

```
public override Mode DefaultGraphicsMode { get; }
```

Property Value

Mode

Mode

Get and set graphics mode.

```
public override Mode Mode { get; set; }
```

Property Value

Mode

Exceptions

ArgumentOutOfRangeException

(set) Thrown if mode is not suppoted.

Methods

Clear(Color)

Clears the entire canvas with the specified color.

```
public override void Clear(Color color)
```

Parameters

`color` Color

The color to clear the screen with.

Clear(int)

Clears the entire canvas with the specified color.

```
public override void Clear(int color)
```

Parameters

`color` int

The ARGB color to clear the screen with.

CopyPixels(int, int, int, int, int, int)

Performs a bit blit operation, copying pixels from one region to another.

```
public void CopyPixels(int srcX, int srcY, int dstX, int dstY, int width = 1, int height
= 1)
```

Parameters

`srcX` int

The source X coordinate.

`srcY` int

The source Y coordinate.

`dstX` int

The destination X coordinate.

`dstY` int

The destination Y coordinate.

`width` int

The width of the region.

`height` int

The height of the region.

Exceptions

NotImplementedException

Thrown if VMWare SVGA 2 has no rectangle copy capability

CreateCursor()

Creates the hardware cursor.

```
public void CreateCursor()
```

CroppedDrawImage(Image, int, int, int, int, bool)

Draws the given image at the specified coordinates, cropping the image to fit within the maximum width and height.

```
public override void CroppedDrawImage(Image image, int x, int y, int width, int height, bool
preventOffBoundPixels = true)
```

Parameters

`image` Image

The image to draw.

`x` int

The X coordinate where the image will be drawn.

`y` int

The Y coordinate where the image will be drawn.

`width` int

`height` int

`preventOffBoundPixels` bool

Prevents drawing outside the bounds of the canvas.

Disable()

Disables the canvas.

```
public override void Disable()
```

Display()

Updates the screen to display the underlying frame-buffer. Call this method in order to synchronize the screen with the canvas.

```
public override void Display()
```

DrawArray(Color[], int, int, int, int)

Draws an array of pixels to the canvas, starting at the given coordinates, using the given width.

```
public override void DrawArray(Color[] colors, int x, int y, int width, int height)
```

Parameters

`colors` Color[]

The pixels to draw.

`x` int

The X coordinate.

`y` int

The Y coordinate.

`width` int

The width of the drawn bitmap.

`height` int

This parameter is unused.

DrawArray(int[], int, int, int, int)

Draws an array of pixels to the canvas, starting at the given coordinates, using the given width.

```
public override void DrawArray(int[] colors, int x, int y, int width, int height)
```

Parameters

`colors` int[]

The pixels to draw.

`x` int

The X coordinate.

`y` int

The Y coordinate.

`width` int

The width of the drawn bitmap.

`height` int

The height of the drawn bitmap.

DrawArray(int[], int, int, int, int, int)

Draws an array of pixels to the canvas, starting at the given coordinates, using the given width.

```
public override void DrawArray(int[] colors, int x, int y, int width, int height,
int startIndex)
```

Parameters

`colors` int[]

The pixels to draw.

`x` int

The X coordinate.

`y` int

The Y coordinate.

`width` int

The width of the drawn bitmap.

`height` int

The height of the drawn bitmap.

`startIndex` int

int[] colors tarting position

DrawChar(char, Font, Color, int, int)

Draws a single character using the given bitmap font.

```
public override void DrawChar(char c, Font font, Color color, int x, int y)
```

Parameters

`c` char

The character to draw.

`font` Font

The bitmap font to use.

`color` Color

The color to write the string with.

`x` int

The origin X coordinate.

`y` int

The origin Y coordinate.

DrawFilledRectangle(Color, int, int, int, int, bool)

Draws a filled rectangle.

```
public override void DrawFilledRectangle(Color color, int xStart, int yStart, int width, int
height, bool preventOffBoundPixels = true)
```

Parameters

`color` Color

The color to draw the rectangle with.

`xStart` int

The starting point X coordinate.

`yStart` int

The starting point Y coordinate.

`width` int

The width of the rectangle.

`height` int

The height of the rectangle.

`preventOffBoundPixels` bool

DrawImage(Image, int, int, bool)

Draws the given image at the specified coordinates.

```
public override void DrawImage(Image image, int x, int y, bool preventOffBoundPixels = true)
```

Parameters

`image` Image

The image to draw.

`x` int

The origin X coordinate.

`y` int

The origin Y coordinate.

`preventOffBoundPixels` bool

Prevents drawing outside the bounds of the canvas.

DrawPoint(Color, int, int)

Sets the pixel at the given coordinates to the specified `color`.

```
public override void DrawPoint(Color color, int x, int y)
```

Parameters

`color` Color

The color to draw with.

`x` int

The X coordinate.

`y` int

The Y coordinate.

DrawPoint(int, int, int)

Sets the pixel at the given coordinates to the specified `color`. without ToArgb()

```
public override void DrawPoint(int color, int x, int y)
```

Parameters

`color` int

The color to draw with (raw argb).

`x` int

The X coordinate.

`y` int

The Y coordinate.

DrawPoint(uint, int, int)

Sets the pixel at the given coordinates to the specified `color`, without unnecessary color operations.

```
public override void DrawPoint(uint color, int x, int y)
```

Parameters

`color` uint

The color to draw with (raw argb).

`x` int

The X coordinate.

`y` int

The Y coordinate.

DrawRectangle(Color, int, int, int, int)

Draws a rectangle.

```
public override void DrawRectangle(Color color, int x, int y, int width, int height)
```

Parameters

`color` Color

The color to draw with.

`x` int

The X coordinate.

`y` int

The Y coordinate.

`width` int

The width of the rectangle.

`height` int

The height of the rectangle.

DrawString(string, Font, Color, int, int)

Draws a string using the given bitmap font.

```
public override void DrawString(string str, Font font, Color color, int x, int y)
```

Parameters

`str` string

The string to draw.

`font` Font

The bitmap font to use.

`color` Color

The color to write the string with.

`x` int

The origin X coordinate.

`y` int

The origin Y coordinate.

GetImage(int, int, int, int)

Creates a bitmap by copying a portion of your canvas from the specified coordinates and dimensions.

```
public override Bitmap GetImage(int x, int y, int width, int height)
```

Parameters

`x` int

The starting X coordinate of the region to copy.

`y` int

The starting Y coordinate of the region to copy.

`width` int

The width of the region to copy.

`height` int

The height of the region to copy.

Returns

Bitmap

A new Cosmos.System.Graphics.Bitmap containing the copied region.

GetPixel(int, int)

```
public Color GetPixel(int x, int y)
```

Parameters

x int

y int

Returns

Color

GetPointColor(int, int)

Gets the color of the pixel at the given coordinates.

```
public override Color GetPointColor(int x, int y)
```

Parameters

x int

The X coordinate.

y int

The Y coordinate.

Returns

Color

GetRawPointColor(int, int)

Gets the color of the pixel at the given coordinates in ARGB.

```
public override int GetRawPointColor(int x, int y)
```

Parameters

x int

The X coordinate.

y int

The Y coordinate.

Returns

int

MovePixel(int, int, int, int)

Moves a single pixel.

```
public void MovePixel(int x, int y, int newX, int newY)
```

Parameters

x int

The X coordinate.

y int

The Y coordinate.

newX int

The new X coordinate.

newY int

The new Y coordinate.

Name()

The name of the Canvas implementation.

```
public override string Name()
```

Returns

string

SetCursor(bool, int, int)

Sets the state of the cursor.

```
public void SetCursor(bool visible, int x, int y)
```

Parameters

`visible` bool

Whether the cursor should be visible.

`x` int

The X coordinate of the cursor.

`y` int

The Y coordinate of the cursor.

Class VMWareSVGAII3D

Namespace: VMwareSvgaII3D

Assembly: VMwareSvgaII3D.dll

```
public class VMWareSVGAII3D
```

Inheritance

object ← VMWareSVGAII3D

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

VMWareSVGAII3D()

```
public VMWareSVGAII3D()
```

Fields

FrameOffset

```
public uint FrameOffset
```

Field Value

uint

FrameSize

```
public uint FrameSize
```

Field Value

uint

HW3DVer

```
public uint HW3DVer
```

Field Value

uint

Is3DEnabled

```
public bool Is3DEnabled
```

Field Value

bool

capabilities

Capabilities.

```
public readonly uint capabilities
```

Field Value

uint

videoMemory

Video memory block.

```
public readonly MemoryBlock videoMemory
```

Field Value

MemoryBlock

Methods

Clear(uint)

Clear screen to specified color.

```
public void Clear(uint color)
```

Parameters

`color` uint

Color.

Exceptions

Exception

Thrown on memory access violation.

NotImplementedException

Thrown if VMWare SVGA 2 has no rectange copy capability

Clear3D(uint, ClearFlags, SVGA3dRect, uint, float, uint)

Clears the currenly binded render targets

```
public void Clear3D(uint cid, ClearFlags flags, SVGA3dRect ClearRect, uint color = 0, float
dpeth = 1, uint stencil = 0)
```

Parameters

`cid` uint

id of the context

`flags` ClearFlags

clear flags

`ClearRect` SVGA3dRect

area to clear

`color` uint

color value

`dpeth` float

depth value

`stencil` uint

stenil buffer value

Examples

canv.driver.Clear3D(context,ClearFlags.Color | ClearFlags.Depth,rect, 0x113366,1f,0);

Copy(uint, uint, uint, uint, uint, uint)

Copy rectangle.

```
public void Copy(uint x, uint y, uint newX, uint newY, uint width, uint height)
```

Parameters

`x` uint

Source X coordinate.

`y` uint

Source Y coordinate.

`newX` uint

Destination X coordinate.

`newY` uint

Destination Y coordinate.

`width` uint

Width.

`height` uint

Height.

Exceptions

NotImplementedException

Thrown if VMWare SVGA 2 has no rectange copy capability

CreateStaticArrayBuffer<T>(T[])

Creates a static array buffer

```
public uint CreateStaticArrayBuffer<T>(T[] data) where T : unmanaged
```

Parameters

`data` T[]

data of the buffer

Returns

uint

id of the buffer

Type Parameters

`T`

this parameter can be anything that has a Sequential layout

Examples

var VertBuffer = canv.driver.CreateStaticArrayBuffer(VertData);

DefineAlphaCursor(uint, uint, int[])

Define alpha cursor.

```
public void DefineAlphaCursor(uint width, uint height, int[] data)
```

Parameters

`width` uint

`height` uint

`data` int[]

DefineContext()

Defines a 3D context

```
public uint DefineContext()
```

Returns

uint

DefineCursor()

Define cursor.

```
public void DefineCursor()
```

DefineShader(uint, SVGA3dShaderType, byte[])

Defines a shader

```
public uint DefineShader(uint cid, SVGA3dShaderType type, byte[] bytecode)
```

Parameters

`cid` uint

id of the context

`type` SVGA3dShaderType

type of the shader

`bytecode` byte[]

shader code in HLSL bytecode

Returns

uint

id of the shader

DefineSurface(uint, uint, SVGA3dSurfaceFormat)

Defines a surface

```
public SVGA3dSurfaceImageId DefineSurface(uint width, uint height,
SVGA3dSurfaceFormat format)
```

Parameters

`width` uint

width of the surface

`height` uint

height of the surface

`format` SVGA3dSurfaceFormat

format of the surface

Returns

SVGA3dSurfaceImageId

Surface object

Examples

var ColorSurface = canv.driver.DefineSurface(1280,720,SVGA3dSurfaceFormat.SVGA3D_X8R8G8B8);

DefineSurfaceFromImage(Image)

Defines a surface from an image

```
public SVGA3dSurfaceImageId DefineSurfaceFromImage(Image image)
```

Parameters

`image` Image

image to define the surface with

Returns

SVGA3dSurfaceImageId

Surface object

Examples

var tex = canv.driver.DefineSurfaceFromImage(new Bitmap(Data));

Disable()

Disable the SVGA Driver, returns to text mode.

```
public void Disable()
```

DoubleBufferUpdate()

Update video memory.

```
public void DoubleBufferUpdate()
```

DrawPrimitives(uint, SVGA3dVertexDecl[], SVGA3dPrimitiveRange[])

Draws primitives

```
public void DrawPrimitives(uint cid, SVGA3dVertexDecl[] decls, SVGA3dPrimitiveRange[]
ranges)
```

Parameters

`cid` uint

id of the context

`decls` SVGA3dVertexDecl[]

vertex format declarations

`ranges` SVGA3dPrimitiveRange[]

index range declarations

Enable()

Enable the SVGA Driver, only needed after Disable() has been called.

```
public void Enable()
```

Fill(uint, uint, uint, uint, uint)

Fill rectangle.

```
public void Fill(uint x, uint y, uint width, uint height, uint color)
```

Parameters

`x` uint

X coordinate.

`y` uint

Y coordinate.

`width` uint

Width.

`height` uint

Height.

`color` uint

Color.

Exceptions

Exception

Thrown on memory access violation.

NotImplementedException

Thrown if VMWare SVGA 2 has no rectange copy capability

GetFIFO(FIFO)

Get FIFO.

```
public uint GetFIFO(FIFO cmd)
```

Parameters

`cmd` FIFO

FIFO command.

Returns

uint

uint value.

GetPixel(uint, uint)

Get pixel.

```
public uint GetPixel(uint x, uint y)
```

Parameters

`x` uint

X coordinate.

`y` uint

Y coordinate.

Returns

uint

uint value.

Exceptions

Exception

Thrown on memory access violation.

InitializeFIFO()

Initialize FIFO.

```
public void InitializeFIFO()
```

Present(SVGA3dSurfaceImageId, SVGA3dRect)

Presents a surface to the screen

```
public void Present(SVGA3dSurfaceImageId image, SVGA3dRect PresentRect)
```

Parameters

`image` SVGA3dSurfaceImageId

surface to present

`PresentRect` SVGA3dRect

area to present to

Examples

canv.driver.Present(ColorSurface,new(0, 0, 1280, 720));

PresentToImage(SVGA3dSurfaceImageId, SVGA3dRect)

Presents a surface to an image

```
public Image PresentToImage(SVGA3dSurfaceImageId image, SVGA3dRect rect)
```

Parameters

`image` SVGA3dSurfaceImageId

surface to present

`rect` SVGA3dRect

area to present to

Returns

Image

Examples

var img = canv.driver.PresentToImage(ColorSurface,new(0, 0, 1280, 720));

Remarks

This functions is really slow, and allocates a lot of memory

ReadRegister(Register)

Read register.

```
public uint ReadRegister(Register register)
```

Parameters

`register` Register

A register.

Returns

uint

uint value.

ReserveFIFO(uint)

```
public void* ReserveFIFO(uint bytes)
```

Parameters

`bytes` uint

Returns

void*

ReserveFIFO3D(uint, uint)

Reserves space in the FIFO for a 3D command

```
public void* ReserveFIFO3D(uint cmd, uint cmdSize)
```

Parameters

`cmd` uint

command id

`cmdSize` uint

size to resereve

Returns

void*

command pointer

SVGA3DUtil_AllocDMABuffer(uint, out SVGAGuestPtr)

Allocates a framebuffer-backed guest pointer

```
public void* SVGA3DUtil_AllocDMABuffer(uint size, out SVGAGuestPtr ptr)
```

Parameters

`size` uint

size to allocate

`ptr` SVGAGuestPtr

returned guest pointer

Returns

void*

void pointer

Exceptions

InvalidOperationException

gmr exception

OutOfMemoryException

vram exception

SetCursor(bool, uint, uint)

Sets the cursor position and draws it.

```
public void SetCursor(bool visible, uint x, uint y)
```

Parameters

`visible` bool

Visible.

`x` uint

X coordinate.

`y` uint

Y coordinate.

SetDepthRange(uint, float, float)

Sets the depth range for a context

```
public void SetDepthRange(uint cid, float min, float max)
```

Parameters

`cid` uint

id of the context

`min` float

minimum depth

`max` float

maximum depth

Examples

canv.driver.SetDepthRange(context, 0.0f, 1.0f);

SetFIFO(FIFO, uint)

Set FIFO.

```
public uint SetFIFO(FIFO cmd, uint value)
```

Parameters

`cmd` FIFO

Command.

`value` uint

Value.

Returns

uint

SetMode(uint, uint, uint)

Set video mode.

```
public void SetMode(uint width, uint height, uint depth = 32)
```

Parameters

`width` uint

Width.

`height` uint

Height.

`depth` uint

Depth.

SetPixel(uint, uint, uint)

Set pixel.

```
public void SetPixel(uint x, uint y, uint color)
```

Parameters

`x` uint

X coordinate.

`y` uint

Y coordinate.

`color` uint

Color.

Exceptions

Exception

Thrown on memory access violation.

SetRenderState(uint, SVGA3dRenderState[])

Sets render states for a context

```
public void SetRenderState(uint cid, SVGA3dRenderState[] states)
```

Parameters

`cid` uint

id of the context

`states` SVGA3dRenderState[]

states to set

SetRenderTarget(uint, SVGA3dRenderTargetType, SVGA3dSurfaceImageId)

Sets a surface as a render target for a context

```
public void SetRenderTarget(uint cid, SVGA3dRenderTargetType type,
SVGA3dSurfaceImageId target)
```

Parameters

`cid` uint

id of the context

`type` SVGA3dRenderTargetType

type of render target

`target` SVGA3dSurfaceImageId

target surface

Examples

canv.driver.SetRenderTarget(context,SVGA3dRenderTargetType.Color,ColorSurface);

SetShader(uint, SVGA3dShaderType, uint)

Sets a shader as current for a context

```
public void SetShader(uint cid, SVGA3dShaderType type, uint shid)
```

Parameters

`cid` uint

id of the context

`type` SVGA3dShaderType

type of the shader

`shid` uint

id of the shader

SetShaderUniform<T>(uint, uint, SVGA3dShaderType, SVGA3dShaderConstType, T)

Sets a shader uniform value for a context

```
public void SetShaderUniform<T>(uint cid, uint reg, SVGA3dShaderType type,
SVGA3dShaderConstType ctype, T value) where T : unmanaged
```

Parameters

`cid` uint

id of the context

`reg` uint

register id of the value

`type` SVGA3dShaderType

shader type

`ctype` SVGA3dShaderConstType

type of value

`value` T

value to set

Type Parameters

`T`

this parameter can be anything that is between 1 and 4 bytes in size and that has a Sequential layout

SetTextureState(uint, SVGA3dTextureState[])

Sets texture states for a context

```
public void SetTextureState(uint cid, SVGA3dTextureState[] states)
```

Parameters

`cid` uint

id of the context

`states` SVGA3dTextureState[]

states to set

SetTransform<T>(uint, SVGA3dTransformType, T)

Sets a transformation matrix for a context

```
public void SetTransform<T>(uint cid, SVGA3dTransformType type, T matrix4x4)
```

Parameters

`cid` uint

id of the context

`type` SVGA3dTransformType

type of the transform

`matrix4x4` T

matrix data

Type Parameters

`T`

this parameter has to be a column major matrix4x4 that has a Sequential layout

Examples

canv.driver.SetTransform(context, SVGA3dTransformType.SVGA3D_TRANSFORM_PROJECTION, projection);

Exceptions

ArgumentException

invalid matrix size

SetViewport(uint, SVGA3dRect)

Sets the viewport for a context

```
public void SetViewport(uint cid, SVGA3dRect rect)
```

Parameters

`cid` uint

id of the context

`rect` SVGA3dRect

viewport rectangle

Examples

canv.driver.SetViewport(context,new(0, 0, 1280, 720));

TestDebugBuffer()

Tests a debug buffer by creating a surface, filling it with a color and presenting it to the screen

```
public uint TestDebugBuffer()
```

Returns

uint

sid

Update(uint, uint, uint, uint)

Update FIFO.

```
public void Update(uint x, uint y, uint width, uint height)
```

Parameters

`x` uint

X coordinate.

`y` uint

Y coordinate.

`width` uint

Width.

`height` uint

Height.

WaitForFifo()

Wait for FIFO.

```
public void WaitForFifo()
```

WriteRegister(Register, uint)

Write register.

```
public void WriteRegister(Register register, uint value)
```

Parameters

`register` Register

A register.

`value` uint

A value.

WriteToFifo(uint)

Write to FIFO.

```
public void WriteToFifo(uint value)
```

Parameters

`value` uint

Value to write.