***D2D1 命名空间***

***D2D1::BitmapProperties 函数***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd370903(v=vs.85).aspx>

创建 D2D1\_BITMAP\_PROPERTIES 结构。

D2D1\_BITMAP\_PROPERTIES BitmapProperties(

const D2D1\_PIXEL\_FORMAT &pixelFormat = D2D1::PixelFormat() ,

FLOAT dpiX = 96.0f,

FLOAT dpiY = 96.0f

);

pixelFormat D2D1\_PIXEL\_FORMAT 位图的像素格式和 Alpha 模式。 默认值是 D2D1\_PIXEL\_FORMAT，其格式为 DXGI\_FORMAT\_UNKNOWN 并且 alphaMode 为 D2D1\_ALPHA\_MODE\_UNKNOWN。 有关像素格式的更多信息，请参见支持的像素格式和 Alpha 模式。

dpiX FLOAT 位图的水平 dpi。 默认值为 96.0f。

dpiY FLOAT 位图的垂直 dpi。 默认值为 96.0f。

返回值 D2D1\_BITMAP\_PROPERTIES 用于描述位图的像素格式和 dpi 的结构。

***D2D1::HwndRenderTargetProperties 函数***

<https://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371105(v=vs.85).aspx>

创建 D2D1\_HWND\_RENDER\_TARGET\_PROPERTIES 结构。

WINAPI HwndRenderTargetProperties (

\_\_in HWND hwnd,

\_\_in D2D1\_SIZE\_U pixelSize = D2D1::Size(static\_cast<UINT>(0), static\_cast<UINT>(0)) ,

\_\_in D2D1\_PRESENT\_OPTIONS presentOptions = D2D1\_PRESENT\_OPTIONS\_NONE

);

hwnd [in]

HWND 呈现器目标根据其绘制命令将输出发出到的 HWND。

pixelSize [in] D2D1\_SIZE\_U 呈现器目标的大小（以像素为单位）。默认值为宽度和高度均为 0 的 D2D1\_SIZE\_U。

presentOptions [in] D2D1\_PRESENT\_OPTIONS 一个值，指定呈现器目标在显示后是否保留帧以及呈现器目标在显示前是否等待设备刷新。默认值为 D2D1\_PRESENT\_OPTIONS\_NONE。

返回值 一个结构，包含 ID2D1HwndRenderTarget 的 HWND、像素大小和显示选项。

***D2D1::PixelFormat 函数***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd372327(v=vs.85).aspx>

创建 D2D1\_PIXEL\_FORMAT 结构。

D2D1\_PIXEL\_FORMAT PixelFormat(

\_\_in DXGI\_FORMAT format = DXGI\_FORMAT\_UNKNOWN ,

\_\_in ALPHA\_MODE alphaMode = D2D1\_ALPHA\_MODE\_UNKNOWN

);

format [in] DXGI\_FORMAT 一个用于指定每个像素中的通道大小和排列的值。默认值为 DXGI\_FORMAT\_UNKNOWN。

alphaMode [in] ALPHA\_MODE 一个值，指定 Alpha 通道是使用预乘 Alpha 还是直接 Alpha，或者指定是否应忽略该通道并视为不透明。默认值为 D2D1\_ALPHA\_MODE\_UNKNOWN。

返回值 D2D1\_PIXEL\_FORMAT 一个结构，包含位图或呈现器目标的数据格式和 Alpha 模式。

***D2D1::RenderTargetProperties 函数***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd372350(v=vs.85).aspx>

创建 D2D1\_RENDER\_TARGET\_PROPERTIES 结构。

D2D1\_RENDER\_TARGET\_PROPERTIES RenderTargetProperties(

D2D1\_RENDER\_TARGET\_TYPE type = D2D1\_RENDER\_TARGET\_TYPE\_DEFAULT ,

\_\_in const PIXEL\_FORMAT pixelFormat = D2D1::PixelFormat() ,

FLOAT dpiX = 0.0,

FLOAT dpiY = 0.0,

D2D1\_RENDER\_TARGET\_USAGE usage = D2D1\_RENDER\_TARGET\_USAGE\_NONE ,

D2D1\_FEATURE\_LEVEL minLevel = D2D1\_FEATURE\_LEVEL\_DEFAULT

);

type D2D1\_RENDER\_TARGET\_TYPE 一个值，指定呈现器目标必须使用硬件呈现还是软件呈现。默认值 D2D1\_RENDER\_TARGET\_TYPE\_DEFAULT 指定使用硬件呈现；如果硬件呈现不可用，则呈现器目标使用软件呈现。请注意，WIC 位图呈现器目标不支持硬件呈现。

pixelFormat [in] PIXEL\_FORMAT 呈现器目标的像素格式和 Alpha 模式。默认像素格式为 D2D1::PixelFormat，该值指示 Direct2D 选择呈现器目标支持的像素格式。有关每个呈现器目标所支持的像素格式和 Alpha 模式的列表，请参见支持的像素格式和 Alpha 模式。

dpiX FLOAT 呈现器目标的水平 DPI。默认值为 0.0。如果 dpiX 和 dpiY 均设置为 0.0，则呈现器目标将使用其默认 DPI。有关更多信息，请参见 D2D1\_RENDER\_TARGET\_PROPERTIES。

dpiY FLOAT 呈现器目标的垂直 DPI。默认值为 0.0。如果 dpiX 和 dpiY 均设置为 0.0，则呈现器目标将使用其默认 DPI。有关更多信息，请参见 D2D1\_RENDER\_TARGET\_PROPERTIES。

usage D2D1\_RENDER\_TARGET\_USAGE 指定如何远程处理呈现器目标以及远程目标是否应与 GDI 兼容。默认值 D2D1\_RENDER\_TARGET\_USAGE\_NONE 会创建与 GDI 不兼容并使用 Direct3D命令流远程处理（如果可用）的呈现器目标。

minLevel D2D1\_FEATURE\_LEVEL 硬件呈现所需的最低 Direct3D 功能级别。默认值 D2D1\_FEATURE\_LEVEL\_DEFAULT 表明 Direct2D 应该确定设备的 Direct3D 功能级别是否足够。此字段只在创建ID2D1HwndRenderTarget 和 ID2D1DCRenderTarget 对象时使用。有关更多信息，请参见 D2D1\_RENDER\_TARGET\_PROPERTIES。

返回值 D2D1\_RENDER\_TARGET\_PROPERTIES 一个包含指定设置的 D2D1\_RENDER\_TARGET\_PROPERTIES。

***D2D1::SizeU函数***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd372401(v=vs.85).aspx>

创建包含指定宽度和高度的 D2D1\_SIZE\_U 结构。

D2D1\_SIZE\_U SizeU(

UINT32 width = 0,

UINT32 height = 0

);

width UINT32 大小的宽度。默认值为 0。

height UINT32 大小的高度。默认值为 0。

返回值 D2D1\_SIZE\_U 新的大小结构。

***ID2D1Bitmap 接口***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371109(v=vs.85).aspx>

表示已绑定到 ID2D1RenderTarget 的位图。

***ID2D1Bitmap::CopyFromMemory 方法***

将指定区域从内存复制到当前位图中。

virtual HRESULT CopyFromMemory(

[in, optional] const D2D1\_RECT\_U \*dstRect,

[in] const void \*srcData,

UINT32 pitch

) = 0;

dstRect [in, optional] D2D1\_RECT\_U 在当前位图中，将 srcRect 所指定的区域复制到的区域的左上角。

srcData [in] void 要复制的数据。

pitch UINT32 存储在 srcData 中的源位图的跨距（或间距）。跨距是扫描线（内存中的一行像素）的字节数。跨距可通过下面的公式进行计算：像素宽度 \* 每像素字节数 + 内存填充。

返回值 HRESULT 如果该方法成功，则返回 S\_OK。 否则，将返回错误代码。HRESULT.

此方法不更新当前位图的大小。如果源位图的内容不适合当前位图，则此方法会失败。另请注意，此方法不执行格式转换；两个位图格式应匹配。

向此方法传递无效输入（如无效的目标矩形）可能会产生不可预测的结果，如变形的图像或设备故障。

如果位图在当前批处理中处于活动状态，则调用此方法可能会导致刷新批处理。如果刷新的批处理未成功完成，则此方法会失败。但是，此方法不会清除在其上刷新批处理的呈现器目标的错误状态。在下一次调用 EndDraw 或 Flush 时，会返回失败的 HRESULT 和标记状态。

***ID2D1Factory接口***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371246(v=vs.85).aspx>

创建 Direct2D 资源。

ID2D1Factory 接口是使用 Direct2D 的起点；可使用该接口创建可用于绘制或描述形状的其他 Direct2D 资源。

继承自IUnknown 接口.

***ID2D1Factory::CreateHwndRenderTarget 方法***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371279(v=vs.85).aspx>

创建 ID2D1HwndRenderTarget，一个呈现到窗口的呈现器目标。

HRESULT CreateHwndRenderTarget(

[ref] const D2D1\_RENDER\_TARGET\_PROPERTIES &renderTargetProperties, [ref]constD2D1\_HWND\_RENDER\_TARGET\_PROPERTIES &hwndRenderTargetProperties,

[out] ID2D1HwndRenderTarget \*\*hwndRenderTarget

);

renderTargetProperties D2D1\_RENDER\_TARGET\_PROPERTIES 进行硬件呈现所需的呈现模式、像素格式、远程处理选项、DPI 信息和最低 DirectX 支持。有关支持的像素格式的信息，请参见支持的像素格式和 Alpha 模式。

hwndRenderTargetProperties D2D1\_HWND\_RENDER\_TARGET\_PROPERTIES 窗口句柄、初始大小（以像素为单位）和呈现选项。

hwndRenderTarget [out] ID2D1HwndRenderTarget 此方法返回时将包含一个指针的地址，该指针指向此方法所创建的ID2D1HwndRenderTarget 对象。

返回值 HRESULT 如果该方法成功，则返回 S\_OK。 否则，将返回错误代码。HRESULT.

创建呈现器目标且硬件加速可用时，将分配计算机的 GPU 上的资源。通过创建一次呈现器目标并保留尽量长的时间，可以提高性能。应用程序应创建呈现器目标一次，然后在应用程序的生命周期中将它们保留，或一直保留至接收到 D2DERR\_RECREATE\_TARGET 错误。如果接收到此错误，则需要重新创建呈现器目标（及其创建的所有资源）。

***ID2D1HwndRenderTarget 接口***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371461(v=vs.85).aspx>

将绘制指令呈现到窗口。

***ID2D1HwndRenderTarget::BeginDraw 方法***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371768(v=vs.85).aspx>

开始在此呈现器目标上绘制。

virtual void BeginDraw() = 0;

This 没有参数。

返回值

This 未返回值。

只能在 BeginDraw 和 EndDraw 调用之间发出绘制操作。

BeginDraw 和 EndDraw 用于指示呈现器目标正在由 Direct2D 系统使用。调用 BeginDraw 时，ID2D1RenderTarget 的不同实现的行为可能不同。例如，ID2D1BitmapRenderTarget 可能在 BeginDraw/EndDraw 调用之间锁定；DXGI 图面呈现器目标可能在 BeginDraw 上获得，并在 EndDraw 上释放；而 ID2D1HwndRenderTarget 可能在 BeginDraw 上开始批处理，并可能存在于 EndDraw 上。

虽然状态设置和状态检索操作甚至可以在 BeginDraw/EndDraw 之外执行，但必须在调用 BeginDraw 方法后才能调用呈现操作。

调用 BeginDraw 后，呈现器目标通常会构建一批呈现命令，但会延迟处理这些命令，直到内部缓冲区已满、调用 Flush 方法或调用 EndDraw 为止。EndDraw 方法将使所有成批绘制操作完成，然后返回指示这些操作成功的 HRESULT 以及（可选）发生错误时呈现器目标的标记状态。EndDraw 方法始终会成功：即使前面的 EndDraw 导致失败的 HRESULT，也不应调用该方法两次。

如果在没有匹配的 BeginDraw 调用时调用了 EndDraw，则后者将返回错误，指示在 EndDraw 之前必须调用 BeginDraw。在呈现器目标上调用 BeginDraw 两次会将该目标置于不进一步绘制任何内容的错误状态，并在调用 EndDraw 时返回相应的 HRESULT 和错误信息。

***ID2D1HwndRenderTarget::CreateBitmap 方法***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371806(v=vs.85).aspx>

创建未初始化的 Direct2D 位图。

HRESULT CreateBitmap(

D2D1\_SIZE\_U size,

[ref] const D2D1\_BITMAP\_PROPERTIES &bitmapProperties,

[out] ID2D1Bitmap \*\*bitmap

);

size D2D1\_SIZE\_U 要创建的位图的尺寸（以像素为单位）。

bitmapProperties D2D1\_BITMAP\_PROPERTIES 要创建的位图的像素格式和每英寸点数 (DPI)。

bitmap [out] ID2D1Bitmap 此方法返回时将包含一个指向新位图的指针的指针。此参数以未初始化的状态传递。

返回值 HRESULT 如果该方法成功，则返回 S\_OK。 否则，将返回错误代码。

***ID2D1HwndRenderTarget::DrawBitmap 方法（有重载）***

将指定位图缩放到指定矩形的大小后绘制该位图。

void DrawBitmap(

[in] ID2D1 Bitmap \*bitmap,

[ref] const D2D1\_RECT\_F &destinationRectangle,

FLOAT opacity,

D2D1\_BITMAP\_INTERPOLATION\_MODE interpolationMode,

[ref] const D2D1\_RECT\_F &sourceRectangle

);

bitmap [in] ID2D1Bitmap 要呈现的位图。

destinationRectangle D2D1\_RECT\_F 位图所绘制到的区域的大小和位置（以呈现器目标的坐标空间中与设备无关的像素表示）。如果矩形未明确排序，则不会进行绘制，但呈现器目标不会进入错误状态。

opacity FLOAT 一个介于 0.0f 和 1.0f 之间的值（包括 0.0f 和 1.0f），它指定要应用到位图的不透明度值；此值将与位图内容的 Alpha 值相乘。

interpolationMode D2D1\_BITMAP\_INTERPOLATION\_MODE 通过绘制操作对位图进行缩放或旋转时使用的内插模式。

sourceRectangle D2D1\_RECT\_F 位图中要进行绘制的区域的大小和位置（以位图的坐标空间中与设备无关的像素表示）。

返回值 This 未返回值。

此方法在失败时不返回错误代码。若要确定一个绘制操作（如 DrawBitmap）是否失败，请检查由 ID2D1RenderTarget::EndDraw 或 ID2D1RenderTarget::Flush 方法返回的结果。

***ID2D1HwndRenderTarget::DrawBitmap 方法（有重载）***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd371876(v=vs.85).aspx>

将指定位图缩放到指定矩形的大小后绘制该位图。

virtual void DrawBitmap(

[in] ID2D1Bitmap \*bitmap,

[in, optional] const D2D1\_RECT\_F \*destinationRectangle = NULL,

FLOAT opacity = 1.0f,

D2D1\_BITMAP\_INTERPOLATION\_MODE interpolationMode =

D2D1\_BITMAP\_INTERPOLATION\_MODE\_LINEAR

,

[in, optional] const D2D1\_RECT\_F \*sourceRectangle = NULL

) = 0;

bitmap [in] ID2D1Bitmap 要呈现的位图。

destinationRectangle [in, optional] D2D1\_RECT\_F 位图所绘制到的区域的大小和位置（以呈现器目标的坐标空间中与设备无关的像素表示）；如果为 NULL，则将在呈现器目标的原点绘制位图的选定部分。如果已指定矩形但未明确排序，则不进行绘制，但呈现器目标不会进入错误状态。

opacity FLOAT 一个介于 0.0f 和 1.0f 之间的值（包括 0.0f 和 1.0f），它指定要应用到位图的不透明度值；此值将与位图内容的 Alpha 值相乘。默认值为 1.0f。

interpolationMode D2D1\_BITMAP\_INTERPOLATION\_MODE 通过绘制操作对位图进行缩放或旋转时使用的内插模式。默认值为 D2D1\_BITMAP\_INTERPOLATION\_MODE\_LINEAR。

sourceRectangle [in, optional] D2D1\_RECT\_F 位图中要进行绘制的区域的位置和大小（以位图的坐标空间中与设备无关的像素表示）；如果为 NULL，将绘制整个位图。

返回值 This 未返回值。

此方法在失败时不返回错误代码。若要确定一个绘制操作（如 DrawBitmap）是否失败，请检查由 ID2D1RenderTarget::EndDraw 或 ID2D1RenderTarget::Flush 方法返回的结果。

***ID2D1HwndRenderTarget::EndDraw 方法***

结束对呈现器目标进行的绘制操作，并指示当前错误状态和关联的标记。

virtual HRESULT EndDraw(

[out, optional] D2D1\_TAG \*tag1 = NULL,

[out, optional] D2D1\_TAG \*tag2 = NULL

) = 0;

tag1 [out, optional] D2D1\_TAG 此方法返回时将包含导致发生错误的绘制操作的标记，如果没有发生错误，则返回 0。此参数以未初始化的状态传递。

tag2 [out, optional] D2D1\_TAG 此方法返回时将包含导致发生错误的绘制操作的标记，如果没有发生错误，则返回 0。此参数以未初始化的状态传递。

返回值 HRESULT 如果该方法成功，则返回 S\_OK。否则将返回 HRESULT 错误代码D2DERR\_RECREATE\_TARGET，并将tag1 和tag2 设置为发生错误时处于活动状态的标记。

***IUnknown 接口***

***IUnknown::AddRef 方法***

<https://msdn.microsoft.com/zh-cn/library/windows/desktop/ms691379(v=vs.85).aspx>

为一个项目里的一个接口添加引用计数。这个方法应该在每一个接口复制点被调用。

ULONG AddRef ( );

返回值 这个方法返回新的引用计数。这个值倾向用于测试目的。

***D2D1CreateFactory 函数（有重载）***

<https://msdn.microsoft.com/en-us/library/windows/desktop/dd368034(v=vs.85).aspx>

创建一个factory对象，用来创建D2D资源。

template <class Factory>

HRESULT D2D1CreateFactory (

\_\_in D2D1\_FACTORY\_TYPE factoryType,

\_\_out Factory \*\*factory

);

Factory 创建的ID2D1Factory类型

factoryType [in] 类型: D2D1\_FACTORY\_TYPE factory的线程模型与它创建的资源。

factory 当这个方法返回时，包含一个新factory指针的地址。

***D2D1\_BITMAP\_PROPERTIES 结构***

描述位图的像素格式和 dpi。

struct D2D1\_BITMAP\_PROPERTIES {

D2D1\_PIXEL\_FORMAT pixelFormat;

FLOAT dpiX;

FLOAT dpiY;

};

pixelFormat D2D1\_PIXEL\_FORMAT 位图的像素格式和 Alpha 模式。

dpiX FLOAT 位图的水平 dpi。

dpiY FLOAT 位图的垂直 dpi。

***D2D1\_FACTORY\_TYPE 枚举***

<https://msdn.microsoft.com/zh-cn/library/windows/desktop/dd368104(v=vs.85).aspx>

指定 Direct2D 是否为 ID2D1Factory 和它所创建的资源提供同步，以便可从多个线程进行安全访问。

typedef enum {

D2D1\_FACTORY\_TYPE\_SINGLE\_THREADED = 0,

D2D1\_FACTORY\_TYPE\_MULTI\_THREADED = 1

} D2D1\_FACTORY\_TYPE ;

D2D1\_FACTORY\_TYPE\_SINGLE\_THREADED 没有为访问或写入工厂或其创建的对象提供同步。如果从多个线程调用该工厂或对象，则由应用程序决定提供访问锁定。

D2D1\_FACTORY\_TYPE\_MULTI\_THREADED Direct2D 为访问和写入工厂及其创建的对象提供同步，因此能够从多个线程安全地进行访问。

***D2D1\_HWND\_RENDER\_TARGET\_PROPERTIE结构***

包含 ID2D1HwndRenderTarget 的 HWND、像素大小和演示选项。

struct D2D1\_HWND\_RENDER\_TARGET\_PROPERTIES {

HWND hwnd;

D2D1\_SIZE\_U pixelSize;

D2D1\_PRESENT\_OPTIONS presentOptions;

};

hwnd HWND 呈现器目标根据其绘制命令将输出发出到的 HWND。

pixelSize D2D1\_SIZE\_U 呈现器目标的大小（以像素为单位）。

presentOptions D2D1\_PRESENT\_OPTIONS 一个值，指定呈现器目标在显示后是否保留帧以及呈现器目标在显示前是否等待设备刷新。

评论

当您调用 CreateHwndRenderTarget 方法以创建新的 ID2D1HwndRenderTarget 时使用此结构。

为方便起见，Direct2D 提供 D2D1::HwndRenderTargetProperties 函数，用于创建新的 D2D1\_HWND\_RENDER\_TARGET\_PROPERTIES 结构。

***D2D1\_PIXEL\_FORMAT 结构***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd368138(v=vs.85).aspx>

struct D2D1\_PIXEL\_FORMAT {

DXGI\_FORMAT format;

D2D1\_ALPHA\_MODE alphaMode;

};

format DXGI\_FORMAT 一个用于指定每个像素中的通道大小和排列的值。

alphaMode D2D1\_ALPHA\_MODE 一个用于指定 Alpha 通道是使用预乘 Alpha 还是直接 Alpha，或者指定是否应忽略该通道并视为未知的值。

有关每个呈现目标所支持的像素格式和 Alpha 模式的更多信息，请参见支持的像素格式和 Alpha 模式。

***D2D1\_RENDER\_TARGET\_PROPERTIES 结构***

<http://msdn.microsoft.com/zh-cn/windows/apps/dd368155>

包含呈现器目标的呈现选项（硬件或软件）、像素格式、DPI 信息、远程处理选项和 Direct3D 支持要求。

struct D2D1\_RENDER\_TARGET\_PROPERTIES {

D2D1\_RENDER\_TARGET\_TYPE type;

D2D1\_PIXEL\_FORMAT pixelFormat;

FLOAT dpiX;

FLOAT dpiY;

D2D1\_RENDER\_TARGET\_USAGE usage;

D2D1\_FEATURE\_LEVEL minLevel;

};

type D2D1\_RENDER\_TARGET\_TYPE 一个用于指定呈现器目标是应该使用硬件呈现还是使用软件呈现的值。 值为 D2D1\_RENDER\_TARGET\_TYPE\_DEFAULT，指定呈现器目标应该使用硬件呈现(如果可用)，否则使用软件呈现。 请注意，WIC 位图呈现器目标不支持硬件呈现。

pixelFormat D2D1\_PIXEL\_FORMAT 呈现器目标的像素格式和 Alpha 模式。 可以使用 D2D1::PixelFormat 函数来创建像素格式，该函数指定 Direct2D 应该为您选择像素格式和 Alpha 模式。 有关每个呈现器目标所支持的像素格式和 Alpha 模式的列表，请参见支持的像素格式和 Alpha 模式。

dpiX FLOAT 呈现器目标的水平 DPI。 要使用默认 DPI，请将 dpiX 和 dpiY 设置为零。有关更多信息，请参见备注部分。

dpiY FLOAT 呈现器目标的垂直 DPI。 要使用默认 DPI，请将 dpiX 和 dpiY 设置为零。有关更多信息，请参见备注部分。

usage D2D1\_RENDER\_TARGET\_USAGE 一个用于指定如何远程处理呈现器目标以及远程目标是否应与 GDI 兼容的值。 设置为 D2D1\_RENDER\_TARGET\_USAGE\_NONE 可以创建与 GDI 不兼容的呈现器目标，并使用 Direct3D 命令流远程处理（如果可用）。

minLevel D2D1\_FEATURE\_LEVEL 一个用于指定硬件呈现所需的最低 Direct3D 功能级别的值。 如果指定的最低级别不可用，当 type 成员设置为 D2D1\_RENDER\_TARGET\_TYPE\_DEFAULT 时，呈现器目标将使用软件呈现；当 type 设置为 D2D1\_RENDER\_TARGET\_TYPE\_HARDWARE 时，创建呈现器目标失败。 值 D2D1\_FEATURE\_LEVEL\_DEFAULT 表明 Direct2D 应该确定设备的 Direct3D 功能级别是否足够。 此字段只在创建 ID2D1HwndRenderTarget 和 ID2D1DCRenderTarget 对象时使用。

***D2D\_SIZE\_U 结构***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd368191(v=vs.85).aspx>

存储有序整数对，通常为矩形的宽度和高度。

struct D2D\_SIZE\_U {

UINT32 width;

UINT32 height;

};

width UINT32 此大小的水平分量。

height UINT32 此大小的垂直分量。

***D2D1\_ALPHA\_MODE 枚举***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd368058(v=vs.85).aspx>

***D2D1\_FEATURE\_LEVEL 枚举***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd756628(v=vs.85).aspx>

描述呈现器目标进行硬件呈现所需的最低 DirectX 支持。

typedef enum {

D2D1\_FEATURE\_LEVEL\_DEFAULT = 0,

D2D1\_FEATURE\_LEVEL\_9 = D3D10\_FEATURE\_LEVEL\_9\_1,

D2D1\_FEATURE\_LEVEL\_10 = D3D10\_FEATURE\_LEVEL\_10\_0

} D2D1\_FEATURE\_LEVEL;

常量

D2D1\_FEATURE\_LEVEL\_DEFAULT Direct2D 确定视频卡是否提供了足够的硬件呈现支持。

D2D1\_FEATURE\_LEVEL\_9 视频卡必须支持 DirectX 9。

D2D1\_FEATURE\_LEVEL\_10 视频卡必须支持 DirectX 10。

***D2D1\_RENDER\_TARGET\_TYPE 枚举***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd756630(v=vs.85).aspx>

描述呈现器目标是使用硬件还是软件进行呈现，或者 Direct2D 是否应选择该呈现模式。

语法

typedef enum {

D2D1\_RENDER\_TARGET\_TYPE\_DEFAULT = 0,

D2D1\_RENDER\_TARGET\_TYPE\_SOFTWARE = 1,

D2D1\_RENDER\_TARGET\_TYPE\_HARDWARE = 2

} D2D1\_RENDER\_TARGET\_TYPE;

D2D1\_RENDER\_TARGET\_TYPE\_DEFAULT 呈现目标使用硬件呈现（如果可用）；否则使用软件呈现。

D2D1\_RENDER\_TARGET\_TYPE\_SOFTWARE 呈现目标只使用软件呈现。

D2D1\_RENDER\_TARGET\_TYPE\_HARDWARE 呈现目标只使用硬件呈现。

并非所有呈现目标都支持硬件呈现。

***D2D1\_RENDER\_TARGET\_USAGE 枚举***

<http://msdn.microsoft.com/zh-cn/library/windows/desktop/dd368157(v=vs.85).aspx>

描述如何远程处理呈现器目标并且是否应与 GDI 兼容。此枚举允许对其成员值进行按位组合。

typedef enum {

D2D1\_RENDER\_TARGET\_USAGE\_NONE = 0x00000000,

D2D1\_RENDER\_TARGET\_USAGE\_FORCE\_BITMAP\_REMOTING = 0x00000001,

D2D1\_RENDER\_TARGET\_USAGE\_GDI\_COMPATIBLE = 0x00000002

} D2D1\_RENDER\_TARGET\_USAGE;

D2D1\_RENDER\_TARGET\_USAGE\_NONE 如果流远程处理失败，呈现目标将尝试使用 Direct3D 命令流远程处理，并使用按位远程处理。呈现目标与 GDI 不兼容。

D2D1\_RENDER\_TARGET\_USAGE\_FORCE\_BITMAP\_REMOTING 呈现目标在本地呈现内容，并将其作为位图发送到终端服务客户端。

D2D1\_RENDER\_TARGET\_USAGE\_GDI\_COMPATIBLE 呈现目标可与 GDI 有效使用。

***DXGI\_FORMAT 枚举***

<http://msdn.microsoft.com/en-us/library/bb173059(VS.85).aspx>