MediaCapture Class

Namespace: Windows.Media.Capture

Provides functionality for capturing photos, audio, and videos from a capture device, such as a webcam.

Edit

In this article

C# Copy

```
[Windows.Foundation.Metadata.ContractVersion(typeof(Windows.Foundation.UniversalApiContract), 65536)]
[Windows.Foundation.Metadata.MarshalingBehavior(Windows.Foundation.Metadata.MarshalingType.Standard)]
[Windows.Foundation.Metadata.Threading(Windows.Foundation.Metadata.ThreadingModel.MTA)]
[Windows.Foundation.Metadata.Activatable(65536,
"Windows.Foundation.UniversalApiContract")]
public sealed class MediaCapture : System.IDisposable
```

Inheritance Object → MediaCapture

Attributes ActivatableAttribute, ContractVersionAttribute, MarshalingBehaviorAttribute, ThreadingAttribute

Implements IDisposable

Windows requirements

Device family	Windows 10 (introduced in 10.0.10240.0)

API contract	Windows.Foundation.UniversalApiContract (introduced in v1.0)
App capabilities	backgroundMediaRecording microphone, webcam

Examples

The following code sample shows how to create and initialize a **MediaCapture** object.

C# Copy

```
// Create and initialze the MediaCapture object.
public async void InitMediaCapture()
{
    _mediaCapture = null;
    _mediaCapture = new Windows.Media.Capture.MediaCapture();

    // Set the MediaCapture to a variable in App.xaml.cs to handle
suspension.
    (App.Current as App).MediaCapture = _mediaCapture;

    await _mediaCapture.InitializeAsync(_captureInitSettings);

    CreateProfile();
}
```

For info about how to handle suspension, see Handle app suspend.

XAML Copy

```
<StackPanel Orientation="Horizontal">
    <CaptureElement x:Name="capturePreview" Width="320" Height="240" />
    <Image x:Name="imagePreview" Stretch="None" Width="320" Height="240"
/>
</StackPanel>

<StackPanel Orientation="Horizontal">
    <Button Click="InitCamera_Click" Content="Initialize Camera" />
    <Button Click="StartCapturePreview_Click" Content="Start Capture
Preview" />
    <Button Click="CapturePhoto_Click" Content="Capture Photo"/>
    <Button Click="StopCapturePreview_Click" Content="Stop Capture
Preview" />
</StackPanel>
```

C# Copy

```
Windows.Media.Capture.MediaCapture captureManager;

async private void InitCamera_Click(object sender, RoutedEventArgs e)
{
    captureManager = new MediaCapture();
    await captureManager.InitializeAsync();
}

async private void StartCapturePreview_Click(object sender,
RoutedEventArgs e)
{
    capturePreview.Source = captureManager;
    await captureManager.StartPreviewAsync();
}

async private void StopCapturePreview_Click(object sender,
RoutedEventArgs e)
{
    await captureManager.StopPreviewAsync();
}

async private void CapturePhoto_Click(object sender, RoutedEventArgs e)
{
    ImageEncodingProperties imgFormat =
ImageEncodingProperties.CreateJpeg();

    // create storage file in local app storage
    StorageFile file = await
ApplicationData.Current.LocalFolder.CreateFileAsync(
        "TestPhoto.jpg",
        CreationCollisionOption.GenerateUniqueName);

    // take photo
    await captureManager.CapturePhotoToStorageFileAsync(imgFormat, file);

    // Get photo as a BitmapImage
    BitmapImage bmpImage = new BitmapImage(new Uri(file.Path));

    // imagePreview is a <Image> object defined in XAML
    imagePreview.Source = bmpImage;
}
```

Remarks

The MediaCapture class is used to capture audio, video, and images from a camera. For how-to guidance for displaying the camera preview, see Display the camera preview. To quickly get started capturing photos, audio, or video, see Basic photo, video, and audio capture with MediaCapture.

The Camera page is the main hub for how-to guidance for using **MediaCapture** in your app. In addition to the basic camera tasks, this page links to how-to articles for advanced scenarios including:

- Using the hardware camera button on devices that have one
- Handling device and screen orientation
- Using camera profiles to determine device capabilities

- Setting the format, resolution, and frame rate of captured video
- Using AdvancedPhotoCapture to capture HDR or low-light photos
- Using the VideoDeviceController to access manual camera controls like exposure, white balance, auto-focus, and flash
- Using effects while capturing video
- Capturing photo sequences
- Using MediaFrameReader to get a stream of frames from one or more cameras, including rgb, infrared, and depth cameras
- Getting a frame from the preview stream

The Camera article also links to all of the UWP SDK samples for camera, such as the Camera starter kit sample.

The InitializeAsync method, which initializes the **MediaCapture** object, must be called before you can start previewing or capturing from the device. In C# or C++ apps, the first use of the **MediaCapture** object to call **InitializeAsync** should be on the STA thread. Calls from an MTA thread may result in undefined behavior. InitializeAsync will launch a consent prompt to get the user's permission for the app to access the microphone or camera. **InitializeAsync** should be called from the main UI thread of your app. Apps must handle app suspension or termination by properly cleaning up media capture resources. For information on shutting down the **MediaCapture** object properly, see Basic photo, video, and audio capture with MediaCapture.

> On Windows, music and media capture apps should monitor the SystemMediaTransportControls.SoundLevel to determine whether the audio streams on the app have been Muted. For apps using the MediaCapture object, capture will be automatically stopped when the capture streams of the app are muted. Capture is not re-started automatically when the audio streams are unmuted, so the SoundLevel changed notification can be used to restart capture. Use the SystemMediaTransportControls.PropertyChanged event to determine when the SoundLevel property changes.

> For Windows Phone 8.x apps, music and media apps should clean up the MediaCapture object and associated resources in the Suspending event handler and recreate them in the Resuming event handler.

In Windows 8.1 audio only apps, if the MediaCategory setting is Other, then high latency mode is used. For low latency, set the MediaCategory setting to Communications.

Adding an in-place editing Media Foundation Transform effect into the capture preview will have no effect on the stream.

Windows 8 UWP apps that have declared both the webcam and microphone capabilities will not function in Windows 8.1 if the user has not enabled both the webcam and microphone privacy settings.

MediaCapture only supports one pass CBR encoding.

Notes on JPEG: JPEG types are passthrough only. To capture an image, the image encoding profile can be set to Auto or you need to specify an encoding profile that matches the native type. To add an effect, you need to switch to an uncompressed video native media type, such as NV12 or RGB32.

Notes on H.264: If the native type is H.264, you can record using a video media type with type identical to the native type. You cannot add an effect to an H.264 native type stream. To capture video, the image encoding profile can be set to Auto or you need to specify an encoding profile that matches the native type.

> ⓘ Note
>
> This class is not agile, which means that you need to consider its threading model and marshaling behavior. For more info, see **Threading and Marshaling (C++/CX)** and **Using Windows Runtime objects in a multithreaded environment (.NET)**.

Version history

Windows version	SDK version	Value added
1607	14393	CreateFrameReaderAsync(MediaFrameSource)
1607	14393	CreateFrameReaderAsync(MediaFrameSource,String)
1607	14393	CreateFrameReaderAsync(MediaFrameSource,String,BitmapSize)
1607	14393	FrameSources
1607	14393	PauseRecordWithResultAsync
1607	14393	RemoveEffectAsync
1607	14393	StopRecordWithResultAsync
1703	15063	CaptureDeviceExclusiveControlStatusChanged

Windows version	SDK version	Value added
1703	15063	CreateMultiSourceFrameReaderAsync
2004	19041	CreateRelativePanelWatcher

Constructors

MediaCapture()	Creates a new instance of the MediaCapture object.

Properties

AudioDeviceController	Gets an object that controls settings for the microphone.
CameraStreamState	Gets the current stream state of the camera stream.
FrameSources	Gets a read-only dictionary of MediaFrameSource objects that can be used simultaneously to acquire media frames.
MediaCaptureSettings	Gets the configuration settings for the MediaCapture object.
ThermalStatus	Gets a value that indicates the current thermal status of the capture device.
VideoDeviceController	Gets an object that controls settings for the video camera.

Methods

AddAudioEffectAsync(IAudioEffectDefinition)	Adds an audio effect to the capture pipeline.
AddEffectAsync(MediaStreamType, String, IPropertySet)	Adds an audio or video effect.
AddVideoEffectAsync(IVideoEffectDefinition, MediaStreamType)	Adds a video effect to the capture pipeline.
CapturePhotoToStorageFileAsync(ImageEncoding	Captures a photo to a storage file.

Properties, IStorageFile)	
CapturePhotoToStreamAsync(ImageEncodingProperties, IRandomAccessStream)	Captures a photo to a random-access stream.
ClearEffectsAsync(MediaStreamType)	Removes all audio and video effects from a stream.
Close()	Closes the media capture object.
CreateFrameReaderAsync(MediaFrameSource)	Creates a MediaFrameReader that is used to acquire frames from a MediaFrameSource.
CreateFrameReaderAsync(MediaFrameSource, String)	Creates a MediaFrameReader that is used to acquire frames with the specified media encoding subtype from a MediaFrameSource.
CreateFrameReaderAsync(MediaFrameSource, String, BitmapSize)	Creates a MediaFrameReader that is used to acquire frames with the specified media encoding subtype and size from a MediaFrameSource.
CreateMultiSourceFrameReaderAsync(IEnumerable<MediaFrameSource>)	Creates a MultiSourceMediaFrameReader that is used to acquire time-correlated frames from one or more MediaFrameSource objects.
CreateRelativePanelWatcher(StreamingCaptureMode, DisplayRegion)	Creates a new instance of the MediaCaptureRelativePanelWatcher class, which monitors the panel associated with the provided DisplayRegion, so that the app receives notifications when the relative location of the panel changes.
Dispose()	Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
FindAllVideoProfiles(String)	Retrieves the list of all video profiles supported by the specified video capture device.
FindConcurrentProfiles(String)	Retrieves the list of video profiles supported by the specified video capture device that can be used while another profile is used on a different capture device.
FindKnownVideoProfiles(String, KnownVideoProfile)	Retrieves the list of all video profiles supported by the specified video capture device that match the specified KnownVideoProfile value.

GetEncoderProperty(MediaStreamType, Guid)	Gets the value of an encoding property.
GetPreviewFrameAsync()	Gets a preview frame from the capture device.
GetPreviewFrameAsync(VideoFrame)	Gets a preview frame from the capture device, copied into the provided destination VideoFrame and converted into the destination frame's format.
GetPreviewMirroring()	Queries whether the video stream is mirrored horizontally.
GetPreviewRotation()	Gets the rotation of the video preview stream.
GetRecordRotation()	Gets the rotation of the recorded video.
InitializeAsync()	Initializes the MediaCapture object, using default settings.
InitializeAsync(MediaCaptureInitializationSettings)	Initializes the MediaCapture object.
IsVideoProfileSupported(String)	Gets a boolean value indicating whether video profiles are supported by the specified video capture device.
PauseRecordAsync(MediaCapturePauseBehavior)	Pauses an ongoing record operation.
PauseRecordWithResultAsync(MediaCapturePauseBehavior)	Pauses an ongoing media record operation and provides a MediaCapturePauseResult that can be used to help the user align the camera with the last captured frame when resuming recording.
PrepareAdvancedPhotoCaptureAsync(ImageEncodingProperties)	Initializes the advanced photo capture and provides the AdvancedPhotoCapture object used to manage the recording.
PrepareLowLagPhotoCaptureAsync(ImageEncodingProperties)	Initializes the low shutter lag photo capture and provides the LowLagPhotoCapture object used to manage the recording.
PrepareLowLagPhotoSequenceCaptureAsync(ImageEncodingProperties)	Initializes the low shutter lag photo sequence capture and provides the LowLagPhotoSequenceCapture object used to manage the recording.
PrepareLowLagRecordToCustomSinkAsync(Media	Initializes the low lag recording using the specified custom sink to store the recording. This method provides the LowLagMediaRecording object used to managed the capture.

EncodingProfile, IMedia Extension)	
PrepareLowLagRecord ToCustomSinkAsync(Media EncodingProfile, String, IPropertySet)	Initializes the low lag recording using the specified custom sink to store the recording. This method provides the LowLagMediaRecording object used to managed the recording.
PrepareLowLagRecord ToStorageFileAsync(Media EncodingProfile, IStorageFile)	Initializes the low lag recording using the specified file to store the recording. This method provides the LowLagMediaRecording object used to managed the recording.
PrepareLowLagRecord ToStreamAsync(Media EncodingProfile, IRandom AccessStream)	Initializes the low lag recording using the specified random-access stream to store the recording. This method provides the LowLagMediaRecording object used to managed the recording.
PrepareVariablePhoto SequenceCaptureAsync(Image EncodingProperties)	Initializes the variable photo sequence capture and provides the VariablePhotoSequenceCapture object used to manage the recording.
RemoveEffectAsync(IMedia Extension)	Removes the specified effect from the capture pipeline.
ResumeRecordAsync()	Resumes a paused recording operation.
SetEncoderProperty(Media StreamType, Guid, Object)	Sets an encoding property.
SetEncodingProperties Async(MediaStreamType, IMediaEncodingProperties, MediaPropertySet)	Asynchronously sets the media encoding properties.
SetPreviewMirroring(Boolean)	Enables or disables horizontal mirroring of the video preview stream. This is not the preferred method for mirroring. See the Remarks section below for details.
SetPreviewRotation(Video Rotation)	Rotates the video preview stream.
SetRecordRotation(Video Rotation)	Rotates the recorded video.
StartPreviewAsync()	Starts preview.

StartPreviewToCustomSinkAsync(MediaEncodingProfile, IMediaExtension)	Starts sending a preview stream to a custom media sink using the specified encoding profile.
StartPreviewToCustomSinkAsync(MediaEncodingProfile, String, IPropertySet)	Starts sending a preview stream to a custom media sink using the specified encoding profile and sink settings.
StartRecordToCustomSinkAsync(MediaEncodingProfile, IMediaExtension)	Start recording to a custom media sink using the specified encoding profile.
StartRecordToCustomSinkAsync(MediaEncodingProfile, String, IPropertySet)	Start recording to a custom media sink using the specified encoding profile and sink settings.
StartRecordToStorageFileAsync(MediaEncodingProfile, IStorageFile)	Starts recording asynchronously to a storage file.
StartRecordToStreamAsync(MediaEncodingProfile, IRandomAccessStream)	Starts recording to a random-access stream.
StopPreviewAsync()	Stops preview.
StopRecordAsync()	Stops recording.
StopRecordWithResultAsync()	Asynchronously stops the media recording and provides a MediaCaptureStopResult that can be used to help the user align the camera with the last captured frame when restarting recording.

Events

CameraStreamStateChanged	Occurs when the state of the camera stream changes.
CaptureDeviceExclusiveControlStatusChanged	Occurs when the exclusive control status of the capture device changes.
Failed	Raised when an error occurs during media capture.
FocusChanged	Occurs when the capture device changes focus.
PhotoConfirmationCaptured	Occurs when a photo confirmation frame is captured.

RecordLimitationExceeded	Occurs when the record limit is exceeded.
ThermalStatusChanged	Occurs when the thermal status of the capture device changes.

Applies to

Product	Versions
WinRT	Build 10240, Build 10586, Build 14383, Build 15063, Build 16299, Build 17134, Build 17763, Build 18362, Build 19041, Build 20348, Build 22000 (Preview)

See also

- Camera
- Basic photo, video, and audio capture with MediaCapture
- Media capture sample
- Camera capture UI sample
- Camera options UI sample
- Device enumeration sample
- Real-Time communication sample
- Media extension sample
- Supported audio and video formats
- Real-time communication sample (Windows 10)
- Camera preview frame sample
- Video stabilization sample
- Holographic Mixed Reality Capture sample

Is this page helpful?

👍 Yes 👎 No