

Contents

[Video Indexer Documentation](#)

[Overview](#)

[What is Video Indexer?](#)

[Quickstarts](#)

[Get started](#)

[Tutorials](#)

[Use Video Indexer API](#)

[Concepts](#)

[Overview](#)

[Compare Video Indexer and Media Services presets](#)

[Manage multiple tenants](#)

[Customizing content models](#)

[Overview](#)

[Brands](#)

[Language](#)

[Person](#)

[How to guides](#)

[Connect to Azure](#)

[Manage account connected to Azure](#)

[Use API to upload and index videos](#)

[Examine Video Indexer output](#)

[Find exact moments within videos](#)

[Scenes, shots, and keyframes](#)

[View and edit Video Indexer insights](#)

[Use editor to create projects](#)

[Embed widgets into your application](#)

[Customize content models](#)

[Person](#)

[using Video Indexer website](#)

[using Video Indexer API](#)

[Brands](#)

[using Video Indexer website](#)

[using Video Indexer API](#)

[Language](#)

[using Video Indexer website](#)

[using Video Indexer API](#)

[Reference](#)

[Video Indexer API](#)

[Resources](#)

[Azure Roadmap](#)

[Pricing](#)

[Regional availability](#)

[Regions](#)

[FAQs](#)

[Compliance](#)

[Release notes](#)

What is Video Indexer?

5/15/2019 • 5 minutes to read • [Edit Online](#)

Azure Video Indexer is a cloud application built on Azure Media Analytics, Azure Search, Cognitive Services (such as the Face API, Microsoft Translator, the Computer Vision API, and Custom Speech Service). It enables you to extract the insights from your videos using Video Indexer video and audio models described below:

Video insights

- **Face detection:** Detects and groups faces appearing in the video.
- **Celebrity identification:** Video Indexer automatically identifies over 1 million celebrities – such as world leaders, actors, actresses, athletes, researchers, business, and tech leaders across the globe. The data about these celebrities can also be found on various famous websites, for example, IMDB and Wikipedia.
- **Account-based face identification:** Video Indexer trains a model for a specific account. It then recognizes faces in the video based on the trained model. For more information, see [Customize a Person model from the Video Indexer website](#) and [Customize a Person model with the Video Indexer API](#).
- **Thumbnail extraction for faces ("best face"):** Automatically identifies the best captured face in each group of faces (based on quality, size, and frontal position) and extract it as an image asset.
- **Visual text recognition (OCR):** Extracts text that is visually displayed in the video.
- **Visual content moderation:** Detects adult and/or racy visuals.
- **Labels identification:** Identifies visual objects and actions displayed.
- **Scene segmentation:** determines when a scene changes in video based on visual cues. A scene depicts a single event and it is composed by a series of consecutive shots, which are semantically related.
- **Shot detection:** determines when a shot changes in video based on visual cues. A shot is a series of frames taken from the same motion-picture camera. For more information, see [Scenes, shots, and keyframes](#).
- **Black frame detection:** Identifies black frames presented in the video.
- **Keyframe extraction:** Detects stable keyframes in a video.
- **Rolling credits:** identify the beginning and end of the rolling credits in the end of TV shows and movies.

Audio insights

- **Automatic language detection:** Automatically identifies the dominant spoken language. Supported languages include English, Spanish, French, German, Italian, Chinese (Simplified), Japanese, Russian, and Brazilian Portuguese Will fallback to English when the language can't be detected.
- **Audio transcription:** Converts speech to text in 12 languages and allows extensions. Supported languages include English, Spanish, French, German, Italian, Chinese (Simplified), Japanese, Arabic, Russian, Brazilian Portuguese, Hindi, and Korean.
- **Closed captioning:** Creates closed captioning in three formats: VTT, TTML, SRT.
- **Two channel processing:** Auto detects, separate transcript and merges to single timeline.
- **Noise reduction:** Clears up telephony audio or noisy recordings (based on Skype filters).
- **Transcript customization (CRIS):** Trains custom speech to text models to create industry-specific transcripts. For more information, see [Customize a Language model from the Video Indexer website](#) and [Customize a Language model with the Video Indexer APIs](#).
- **Speaker enumeration:** Maps and understands which speaker spoke which words and when.
- **Speaker statistics:** Provides statistics for speakers speech ratios.
- **Textual content moderation:** Detects explicit text in the audio transcript.

- **Audio effects:** Identifies audio effects such as hand claps, speech, and silence.
- **Emotion detection:** Identifies emotions based on speech (what is being said) and voice tonality (how it is being said). The emotion could be: joy, sadness, anger, or fear.
- **Translation:** Creates translations of the audio transcript to 54 different languages.

Audio and video insights (multi channels)

When indexing by one channel partial result for those models will be available

- **Keywords extraction:** Extracts keywords from speech and visual text.
- **Brands extraction:** Extracts brands from speech and visual text.
- **Topic inference:** Makes inference of main topics from transcripts. The 1st-level IPTC taxonomy is included.
- **Artifacts:** Extracts rich set of "next level of details" artifacts for each of the models.
- **Sentiment analysis:** Identifies positive, negative, and neutral sentiments from speech and visual text.

Once Video Indexer is done processing and analyzing, you can review, curate, search, and publish the video insights.

Whether your role is a content manager or a developer, the Video Indexer service is able to address your needs. Content managers can use the Video Indexer web portal to consume the service without writing a single line of code, see [Get started with the Video Indexer website](#). Developers can take advantage of APIs to process content at scale, see [Use Video Indexer REST API](#). The service also enables customers to use widgets to publish video streams and extracted insights in their own applications, see [Embed visual widgets in your application](#).

You can sign up for the service using existing AAD, LinkedIn, Facebook, Google, or MSA account. For more information, see [getting started](#).

Scenarios

Below are a few scenarios where Video Indexer can be useful

- Search – Insights extracted from the video can be used to enhance the search experience across a video library. For example, indexing spoken words and faces can enable the search experience of finding moments in a video where a particular person spoke certain words or when two people were seen together. Search based on such insights from videos is applicable to news agencies, educational institutes, broadcasters, entertainment content owners, enterprise LOB apps and in general to any industry that has a video library that users need to search against.
- Content creation – insights extracted from videos and help effectively create content such as trailers, social media content, news clips etc. from existing content in the organization archive
- Monetization – Video Indexer can help improve the value of videos. As an example, industries that rely on ad revenue (for example, news media, social media, etc.), can deliver more relevant ads by using the extracted insights as additional signals to the ad server (presenting a sports shoe ad is more relevant in the middle of a football match vs. a swimming competition).
- User engagement – Video insights can be used to improve user engagement by positioning the relevant video moments to users. As an example, consider an educational video that explains spheres for the first 30 minutes and pyramids in the next 30 minutes. A student reading about pyramids would benefit more if the video is positioned starting from the 30-minute marker.

Next steps

You're ready to get started with Video Indexer. For more information, see the following articles:

- [Get started with the Video Indexer website](#)
- [Process content with Video Indexer REST API](#)

- Embed visual widgets in your application

Quickstart: How to sign up and upload your first video

5/15/2019 • 2 minutes to read • [Edit Online](#)

This getting started tutorial shows how to sign in to the Video Indexer website and how to upload your first video.

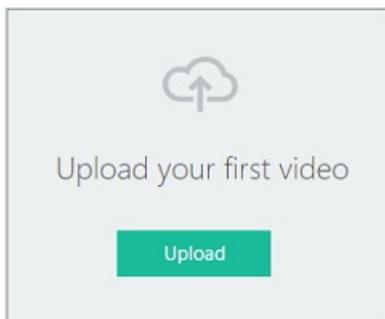
When creating a Video Indexer account, you can choose a free trial account (where you get a certain number of free indexing minutes) or a paid option (where you are not limited by the quota). With free trial, Video Indexer provides up to 600 minutes of free indexing to website users and up to 2400 minutes of free indexing to API users. With paid option, you create a Video Indexer account that is [connected to your Azure subscription and an Azure Media Services account](#). You pay for minutes indexed as well as the Azure Media Services account related charges.

Sign up for Video Indexer

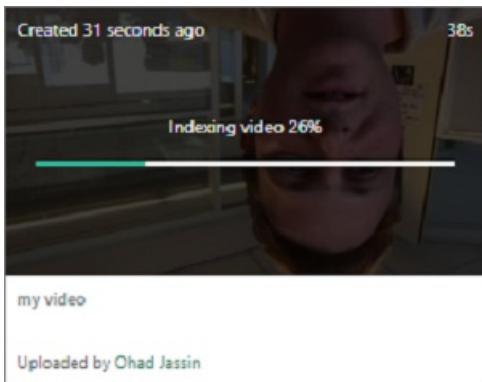
To start developing with Video Indexer, browse to the [Video Indexer](#) website and sign up.

Upload a video using the Video Indexer website

1. Sign in on the [Video Indexer](#) website.
2. To upload a video, press the **Upload** button or link.



Once your video has been uploaded, Video Indexer starts indexing and analyzing the video.



Once Video Indexer is done analyzing, you will get a notification with a link to your video and a short description of what was found in your video. For example: people, topics, OCRs.

Next steps

You can now use the [Video Indexer](#) website or [Video Indexer Developer Portal](#) to see the insights of the video.

See also

[Video Indexer overview](#)

[Start using APIs.](#)

Tutorial: Use the Video Indexer API

5/15/2019 • 6 minutes to read • [Edit Online](#)

Video Indexer consolidates various audio and video artificial intelligence (AI) technologies offered by Microsoft in one integrated service, making development simpler. The APIs are designed to enable developers to focus on consuming Media AI technologies without worrying about scale, global reach, availability, and reliability of cloud platform. You can use the API to upload your files, get detailed video insights, get URLs of insight and player widgets in order to embed them into your application, and other tasks.

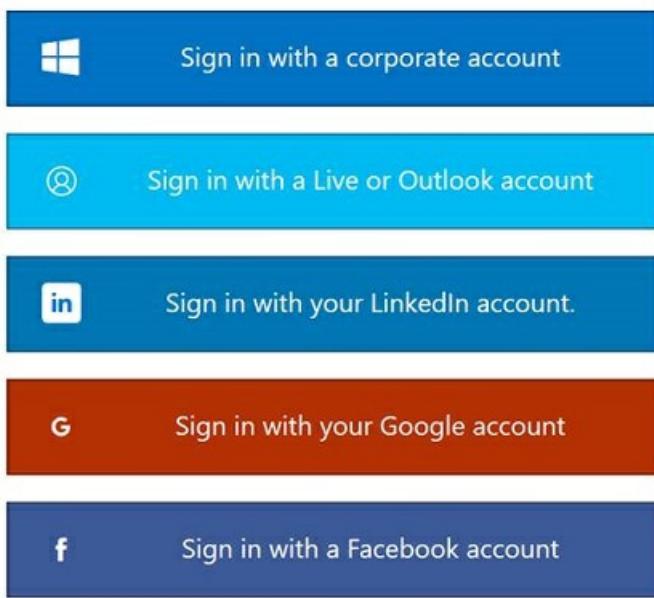
When creating a Video Indexer account, you can choose a free trial account (where you get a certain number of free indexing minutes) or a paid option (where you are not limited by the quota). With free trial, Video Indexer provides up to 600 minutes of free indexing to website users and up to 2400 minutes of free indexing to API users. With paid option, you create a Video Indexer account that is [connected to your Azure subscription and an Azure Media Services account](#). You pay for minutes indexed as well as the Azure Media Services account related charges.

This article shows how the developers can take advantage of the [Video Indexer API](#).

Subscribe to the API

1. Sign in to [Video Indexer Developer Portal](#).

Sign-In



By signing in I agree to [Microsoft Service Agreement](#) and the [Privacy Statement](#).

IMPORTANT

- You must use the same provider you used when you signed up for Video Indexer.
- Personal Google and Microsoft (outlook/live) accounts can only be used for trial accounts. Accounts connected to Azure require Azure AD.
- There can be only one active account per E-Mail. If a user tries to sign-in with user@gmail.com for LinkedIn and after that with user@gmail.com for Google the later will display an error page, saying the user already exist.

2. Subscribe.

Select the [Products](#) tab. Then, select Authorization and subscribe.

HOME APIS PRODUCTS ISSUES

Authorization

VideoIndexer Authorization API

This product contains 1 API:

- [Authorization](#)

[Subscribe](#)

NOTE

New users are automatically subscribed to Authorization.

Once you subscribe, you will be able to see your subscription and your primary and secondary keys. The keys should be protected. The keys should only be used by your server code. They should not be available on the client side (js, .html, etc.).

Your subscriptions

Subscription details		Product	State
Subscription name	Production	Rename	Production
Started on	11/09/2016		Active
Primary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show Regenerate	
Secondary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show Regenerate	

TIP

Video Indexer user can use a single subscription key to connect to multiple Video Indexer accounts. You can then link these Video Indexer accounts to different Media Services accounts.

Obtain access token using the Authorization API

Once you subscribed to the Authorization API, you will be able to obtain access tokens. These access tokens are used to authenticate against the Operations API.

Each call to the Operations API should be associated with an access token, matching the authorization scope of the call.

- User level - user level access tokens let you perform operations on the **user** level. For example, get associated accounts.
- Account level – account level access tokens let you perform operations on the **account** level or the **video** level. For example, Upload video, list all videos, get video insights, etc.
- Video level – video level access tokens let you perform operations on a specific **video**. For example, get video insights, download captions, get widgets, etc.

You can control whether these tokens are readonly or they allow editing by specifying **allowEdit=true/false**.

For most server-to-server scenarios, you will probably use the same **account** token since it covers both **account** operations and **video** operations. However, if you are planning to make client side calls to Video Indexer (for example, from javascript), you would want to use a **video** access token, to prevent clients from getting access to the entire account. That is also the reason that when embedding VideoIndexer client code in your client (for example, using **Get Insights Widget** or **Get Player Widget**) you must provide a **video** access token.

To make things easier, you can use the **Authorization API > GetAccounts** to get your accounts without obtaining a user token first. You can also ask to get the accounts with valid tokens, enabling you to skip an additional call to get an account token.

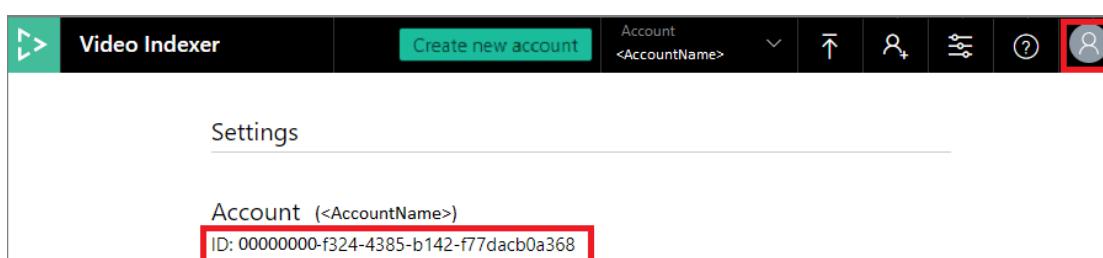
Access tokens expire after 1 hour. Make sure your access token is valid before using the Operations API. If expires, call the Authorization API again to get a new access token.

You are ready to start integrating with the API. Find [the detailed description of each Video Indexer REST API](#).

Account ID

The Account ID parameter is required in all operational API calls. Account ID is a GUID that can be obtained in one of the following ways:

- Use the **Video Indexer website** to get the Account ID:
 1. Browse to the [Video Indexer](#) website and sign in.
 2. Browse to the **Settings** page.
 3. Copy the account ID.



- Use **Video Indexer Developer Portal** to programmatically get the Account ID.

Use the [Get accounts API](#).

TIP

You can generate access tokens for the accounts by defining `generateAccessTokens=true`.

- Get the account ID from the URL of a player page in your account.

When you watch a video, the ID appears after the `accounts` section and before the `videos` section.

```
https://www.videoindexer.ai/accounts/00000000-f324-4385-b142-f77dadb0a368/videos/d45bf160b5/
```

Recommendations

This section lists some recommendations when using Video Indexer API.

- If you are planning to upload a video, it is recommended to place the file in some public network location (for example, OneDrive). Get the link to the video and provide the URL as the upload file param.

The URL provided to Video Indexer must point to a media (audio or video) file. Some of the links generated by OneDrive are for an HTML page that contains the file. An easy verification for the URL would be to paste it into a browser – if the file starts downloading, it's likely a good URL. If the browser is rendering some visualization, it's likely not a link to a file but an HTML page.

- When you call the API that gets video insights for the specified video, you get a detailed JSON output as the response content. [See details about the returned JSON in this topic.](#)

Code sample

The following C# code snippet demonstrates the usage of all the Video Indexer APIs together.

```
var apiUrl = "https://api.videoindexer.ai";
var accountId = "...";
var location = "westus2";
var apiKey = "...";

System.Net.ServicePointManager.SecurityProtocol = System.Net.ServicePointManager.SecurityProtocol |
System.Net.SecurityProtocolType.Tls12;

// create the http client
var handler = new HttpClientHandler();
handler.AllowAutoRedirect = false;
var client = new HttpClient(handler);
client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", apiKey);

// obtain account access token
var accountAccessTokenRequestResult = client.GetAsync($"'{apiUrl}/auth/{location}/Accounts/{accountId}/AccessToken?allowEdit=true").Result;
var accountAccessToken = accountAccessTokenRequestResult.Content.ReadAsStringAsync().Result.Replace("\\\"", "");

client.DefaultRequestHeaders.Remove("Ocp-Apim-Subscription-Key");

// upload a video
var content = new MultipartFormDataContent();
Debug.WriteLine("Uploading...");
// get the video from URL
var videoUrl = "VIDEO_URL"; // replace with the video URL

// as an alternative to specifying video URL, you can upload a file.
// remove the videoUrl parameter from the query string below and add the following lines:
//FileStream video =File.OpenRead(Globals.VIDEOFILE_PATH);
//byte[] buffer =newbyte[video.Length];
//video.Read(buffer, 0, buffer.Length);
//content.Add(newByteArrayContent(buffer));

var uploadRequestResult = client.PostAsync($"'{apiUrl}/{location}/Accounts/{accountId}/Videos?accessToken={accountAccessToken}&name=some_name&description=some_description&privacy=private&partition=some_partition&videoUrl={videoUrl}", content).Result;
var uploadResult = uploadRequestResult.Content.ReadAsStringAsync().Result;

// get the video id from the upload result
```

```

var videoId = JsonConvert.DeserializeObject<dynamic>(uploadResult)["id"];
Debug.WriteLine("Uploaded");
Debug.WriteLine("Video ID: " + videoId);

// obtain video access token
client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", apiKey);
var videoTokenRequestResult = client.GetAsync(${apiUrl}/auth/{location}/Accounts/{accountId}/Videos/{videoId}/AccessToken?allowEdit=true").Result;
var videoAccessToken = videoTokenRequestResult.Content.ReadAsStringAsync().Result.Replace("\\", "");

client.DefaultRequestHeaders.Remove("Ocp-Apim-Subscription-Key");

// wait for the video index to finish
while (true)
{
    Thread.Sleep(10000);

    var videoGetIndexRequestResult = client.GetAsync(${apiUrl}/{location}/Accounts/{accountId}/Videos/{videoId}/Index?accessToken=${videoAccessToken}&language=English").Result;
    var videoGetIndexResult = videoGetIndexRequestResult.Content.ReadAsStringAsync().Result;

    var processingState = JsonConvert.DeserializeObject<dynamic>(videoGetIndexResult)["state"];

    Debug.WriteLine("");
    Debug.WriteLine("State:");
    Debug.WriteLine(processingState);

    // job is finished
    if (processingState != "Uploaded" && processingState != "Processing")
    {
        Debug.WriteLine("");
        Debug.WriteLine("Full JSON:");
        Debug.WriteLine(videoGetIndexResult);
        break;
    }
}

// search for the video
var searchRequestResult = client.GetAsync(${apiUrl}/{location}/Accounts/{accountId}/Videos/Search?accessToken={accountAccessToken}&id={videoId}).Result;
var searchResult = searchRequestResult.Content.ReadAsStringAsync().Result;
Debug.WriteLine("");
Debug.WriteLine("Search:");
Debug.WriteLine(searchResult);

// get insights widget url
var insightsWidgetRequestResult = client.GetAsync(${apiUrl}/{location}/Accounts/{accountId}/Videos/{videoId}/InsightsWidget?accessToken=${videoAccessToken}&widgetType=Keywords&allowEdit=true").Result;
var insightsWidgetLink = insightsWidgetRequestResult.Headers.Location;
Debug.WriteLine("Insights Widget url:");
Debug.WriteLine(insightsWidgetLink);

// get player widget url
var playerWidgetRequestResult = client.GetAsync(${apiUrl}/{location}/Accounts/{accountId}/Videos/{videoId}/PlayerWidget?accessToken=${videoAccessToken}).Result;
var playerWidgetLink = playerWidgetRequestResult.Headers.Location;
Debug.WriteLine("");
Debug.WriteLine("Player Widget url:");
Debug.WriteLine(playerWidgetLink);

```

See also

- [Video Indexer overview](#)

- Regions

Next steps

[Examine details of the output JSON.](#)

Video Indexer concepts

5/15/2019 • 2 minutes to read • [Edit Online](#)

This article describes some concepts of the Video Indexer service.

Summarized insights

Summarized insights contain an aggregated view of the data: faces, topics, emotions. For example, instead of going over each of the thousands of time ranges and checking which faces are in it, the summarized insights contains all the faces and for each one, the time ranges it appears in and the % of the time it is shown.

time range vs. adjusted time range

TimeRange is the time range in the original video. AdjustedTimeRange is the time range relative to the current playlist. Since you can create a playlist from different lines of different videos, you can take a 1-hour video and use just 1 line from it, for example, 10:00-10:15. In that case, you will have a playlist with 1 line, where the time range is 10:00-10:15 but the adjustedTimeRange is 00:00-00:15.

Blocks

Blocks are meant to make it easier to go through the data. For example, block might be broken down based on when speakers change or there is a long pause.

Next steps

For information about how to get started, see [How to sign up and upload your first video](#).

See also

[Video Indexer overview](#)

Compare Azure Media Services v3 presets and Video Indexer

5/15/2019 • 2 minutes to read • [Edit Online](#)

This article compares the capabilities of [Video Indexer APIs](#) and [Media Services v3 APIs](#).

Currently, there is an overlap between features offered by the [Video Indexer APIs](#) and the [Media Services v3 APIs](#). The following table offers the current guideline for understanding the differences and similarities.

Compare

FEATURE	VIDEO INDEXER APIs	VIDEO ANALYZER AND AUDIO ANALYZER PRESETS IN MEDIA SERVICES V3 APIs
Media Insights	Enhanced	Fundamentals
Experiences	See the full list of supported features: Overview	Returns video insights only
Billing	Media Services pricing	Media Services pricing
Compliance	ISO 27001, ISO 27018, SOC 1,2,3, HIPAA, FedRAMP, PCI, and HITRUST certified. For the most current updates, visit current certifications status of Video Indexer .	Media Services is compliant with many certifications. Check out Azure Compliance Offerings.pdf and search for "Media Services" to see if it complies with a certificate of interest.
Free Trial	East US	Not available
Region availability	East US 2, South Central US, West US 2, North Europe, West Europe, Southeast Asia, East Asia, and Australia East. For the most current updates, visit the products by region page.	See Azure status .

Next steps

[Video Indexer overview](#)

[Media Services v3 overview](#)

Manage multiple tenants

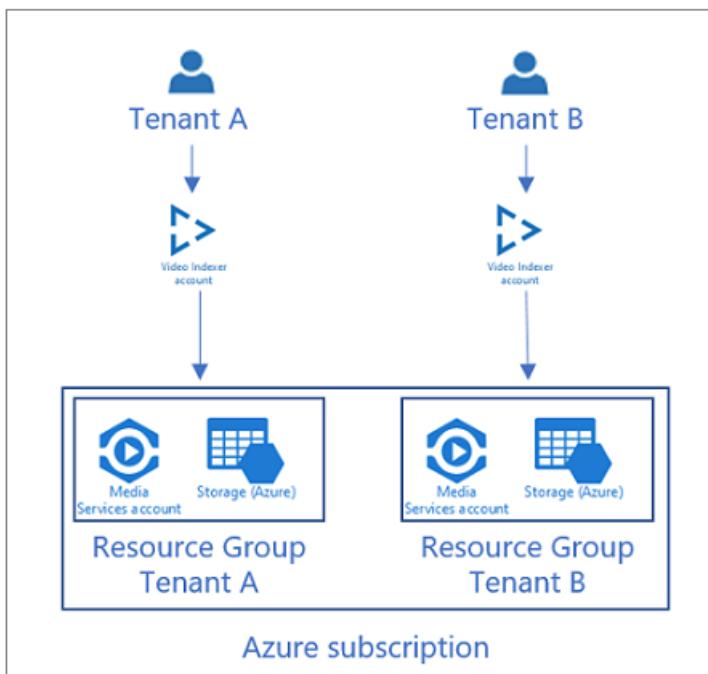
5/20/2019 • 2 minutes to read • [Edit Online](#)

This article discusses different options for managing multiple tenants with Video Indexer. Choose a method that is most suitable for your scenario:

- Video Indexer account per tenant
- Single Video Indexer account for all tenants
- Azure subscription per tenant

Video Indexer account per tenant

When using this architecture, a Video Indexer account is created for each tenant. The tenants have full isolation in the persistent and compute layer.



Considerations

- Customers do not share storage accounts (unless manually configured by the customer).
- Customers do not share compute (reserved units) and don't impact processing jobs times of one another.
- You can easily remove a tenant from the system by deleting the Video Indexer account.
- There is no ability to share custom models between tenants.

Make sure there is no business requirement to share custom models.

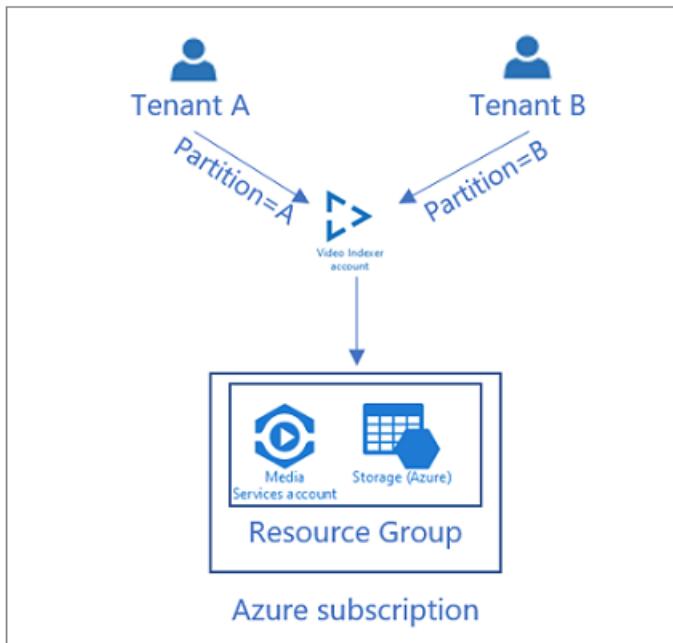
- Harder to manage due to multiple Video Indexer (and associated Media Services) accounts per tenant.

TIP

Create an admin user for your system in [Video Indexer Developer Portal](#) and use the Authorization API to provide your tenants the relevant [account access token](#).

Single Video Indexer account for all users

When using this architecture, the customer is responsible for tenants isolation. All tenants have to use a single Video Indexer account with a single Azure Media Service account. When uploading, searching, or deleting content, the customer will need to filter the proper results for that tenant.



With this option, customization models (Person, Language, and Brands) can be shared or isolated between tenants by filtering the models by tenant.

When [uploading videos](#), you can specify a different partition attribute per tenant. This will allow isolation in the [search API](#). By specifying the partition attribute in the search API you will only get results of the specified partition.

Considerations

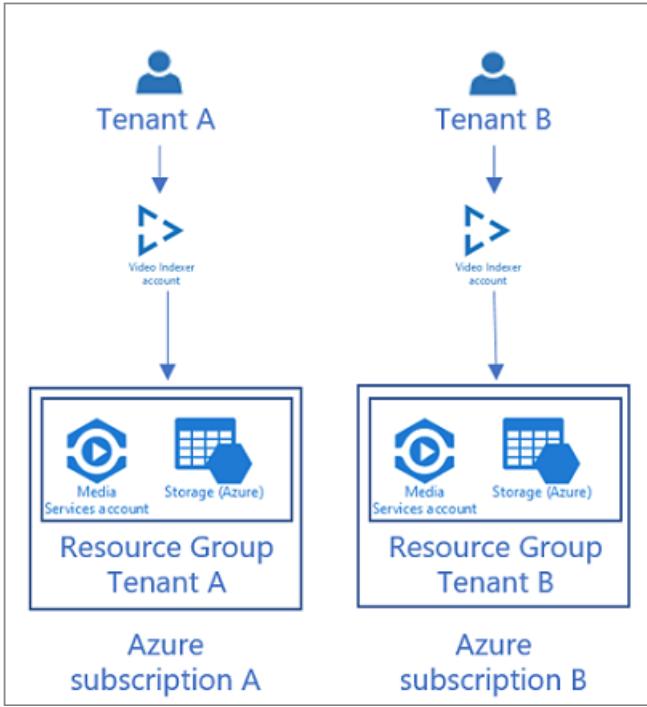
- Ability to share content and customization models between tenants.
- One tenant impacts the performance of other tenants.
- Customer needs to build a complex management layer on top of Video Indexer.

TIP

You can use the [priority](#) attribute to prioritize tenants jobs.

Azure subscription per tenant

When using this architecture, each tenant will have their own Azure subscription. For each user, you will create a new Video Indexer account in the tenant subscription.



Considerations

- This is the only option that enables billing separation.
- This integration has more management overhead than Video Indexer account per tenant. If billing is not a requirement, it is recommended to use one of the other options described in this article.

Next steps

[Overview](#)

Customizing content models in Video Indexer

6/27/2019 • 2 minutes to read • [Edit Online](#)

Video Indexer allows you to customize some of its models to be adapted to your specific use case. These models include [brands](#), [language](#), and [person](#). You can easily customize these models using the Video Indexer website or API.

This article gives links to the conceptual articles that explain the benefits of each type of customization. This article also links to how-to guides that show how you can implement the customization of each model.

Brands model

- [Customizing the brands model overview](#)
- [Customizing the brands model using the Video Indexer website](#)
- [Customizing the brands model using the Video Indexer API](#)

Language model

- [Customizing language models overview](#)
- [Customizing language models using the Video Indexer website](#)
- [Customizing language models using the Video Indexer API](#)

Person model

- [Customizing language models overview](#)
- [Customizing language models using the Video Indexer website](#)
- [Customizing language models using the Video Indexer API](#)

Next steps

[Video Indexer overview](#)

Customize a Brands model in Video Indexer

5/15/2019 • 2 minutes to read • [Edit Online](#)

Video Indexer supports brand detection from speech and visual text during indexing and reindexing of video and audio content. The brand detection feature identifies mentions of products, services, and companies suggested by Bing's brands database. For example, if Microsoft is mentioned in a video or audio content or if it shows up in visual text in a video, Video Indexer detects it as a brand in the content. Brands are disambiguated from other terms using context.

Brand detection is useful in a wide variety of business scenarios such as contents archive and discovery, contextual advertising, social media analysis, retail compete analysis, and many more. Video Indexer brand detection enables you to index brand mentions in speech and visual text, using Bing's brands database as well as with customization by building a custom Brands model for each Video Indexer account. The custom Brands model feature allows you to select whether or not Video Indexer will detect brands from the Bing brands database, exclude certain brands from being detected (essentially creating a black list of brands), and include brands that should be part of your model that might not be in Bing's brands database (essentially creating a white list of brands). The custom Brands model that you create will only be available in the account in which you created the model.

Out of the box detection example

In the [Microsoft Build 2017 Day 2](#) presentation, the brand "Microsoft Windows" appears multiple times. Sometimes in the transcript, sometimes as visual text and never as verbatim. Video Indexer detects with high precision that a term is indeed brand based on the context, covering over 90k brands out of the box, and constantly updating. At 02:25, Video Indexer detects the brand from speech and then again at 02:40 from visual text, which is part of the windows logo.

The screenshot shows the Video Indexer platform. On the left is a video player displaying a man speaking on stage at a Microsoft Build event. The video is titled "Build 2017 Day 2 Keynote". Below the video player are social sharing icons. On the right is a detailed analysis panel. At the top of the panel are tabs for "Insights" and "Transcript", along with a search bar and language selection. Below these are several semantic search boxes showing phrases like "mixed reality", "pulled from the cloud", "design system", etc. Further down are sections for "Annotations" (showing tags like "person", "indoor", "standing", "man", "floor", "sky", "outdoor", "screenshot", "monitor", "grass") and "Brands" (listing Microsoft Windows, Microsoft, Windows 10, Autodesk, iOS, Android, OneDrive, Microsoft Visual Studio, Microsoft Cortana, Windows Store). A timeline at the bottom indicates the video's duration.

This screenshot shows a specific view within the Video Indexer's brand detection feature. It lists various brands, with "Microsoft Windows" highlighted in a black button. Other brands listed include Microsoft, Windows 10, Autodesk, iOS, Android, OneDrive, Microsoft Visual Studio, Microsoft Cortana, and Windows Store. Below this is a detailed view for "Microsoft Windows", showing its "Show description" link and a timeline bar with navigation arrows.

Talking about windows in the context of construction will not detect the word "Windows" as a brand, and same for Box, Apple, Fox, etc., based on advanced Machine Learning algorithms that know how to disambiguate from context. Brand Detection works for all our supported languages. Click here for [full Microsoft Build 2017 Day 2 keynote video and index](#).

To bring your own brands, check out Next steps.

Next steps

[Customize Brands model using APIs](#)

[Customize Brands model using the website](#)

Customize a Language model with Video Indexer

5/15/2019 • 2 minutes to read • [Edit Online](#)

Video Indexer supports automatic speech recognition through integration with the Microsoft [Custom Speech Service](#). You can customize the Language model by uploading adaptation text, namely text from the domain whose vocabulary you'd like the engine to adapt to. Once you train your model, new words appearing in the adaptation text will be recognized, assuming default pronunciation, and the Language model will learn new probable sequences of words. Custom Language models are supported for English, Spanish, French, German, Italian, Chinese (Simplified), Japanese, Russian, Brazilian Portuguese, Hindi, and Korean.

Let's take a word that is highly specific, like "Kubernetes" (in the context of Azure Kubernetes service), as an example. Since the word is new to Video Indexer, it is recognized as "communities". You need to train the model to recognize it as "Kubernetes". In other cases, the words exist, but the Language model is not expecting them to appear in a certain context. For example, "container service" is not a 2-word sequence that a non-specialized Language model would recognize as a specific set of words.

You have the option to upload words without context in a list in a text file. This is considered partial adaptation. Alternatively, you can upload text file(s) of documentation or sentences related to your content for better adaptation.

You can use the Video Indexer APIs or the website to create and edit custom Language models, as described in topics in the [Next steps](#) section of this topic.

Best practices for custom Language models

Video Indexer learns based on probabilities of word combinations, so to learn best:

- Give enough real examples of sentences as they would be spoken.
- Put only one sentence per line, not more. Otherwise the system will learn probabilities across sentences.
- It is okay to put one word as a sentence to boost the word against others, but the system learns best from full sentences.
- When introducing new words or acronyms, if possible, give as many examples of usage in a full sentence to give as much context as possible to the system.
- Try to put several adaptation options, and see how they work for you.
- Avoid repetition of the exact same sentence multiple times. It may create bias against the rest of the input.
- Avoid including uncommon symbols (~, # @ % &) as they will get discarded. The sentences in which they appear will also get discarded.
- Avoid putting too large inputs, such as hundreds of thousands of sentences, because doing so will dilute the effect of boosting.

Next steps

[Customize Language model using APIs](#)

[Customize Language model using the website](#)

Customize a Person model in Video Indexer

5/15/2019 • 2 minutes to read • [Edit Online](#)

Video Indexer supports celebrity recognition in your videos. The celebrity recognition feature covers approximately one million faces based on commonly requested data source such as IMDB, Wikipedia, and top LinkedIn influencers. Faces that are not recognized by Video Indexer are still detected but are left unnamed. Customers can build custom Person models and enable Video Indexer to recognize faces that are not recognized by default. Customers can build these Person models by pairing a person's name with image files of the person's face.

If your account caters to different use-cases, you can benefit from being able to create multiple Person models per account. For example, if the content in your account is meant to be sorted into different channels, you might want to create a separate Person model for each channel.

NOTE

Each Person model supports up to 1 million people and each account has a limit of 50 Person models.

Once a model is created, you can use it by providing the model ID of a specific Person model when uploading/indexing or reindexing a video. Training a new face for a video, updates the specific custom model that the video was associated with.

If you do not need the multiple Person model support, do not assign a Person model ID to your video when uploading/indexing or reindexing. In this case, Video Indexer will use the default Person model in your account.

You can use the Video Indexer website to edit faces that were detected in a video and to manage multiple custom Person models in your account, as described in the [Customize a Person model using a website](#) topic. You can also use the API, as described in [Customize a Person model using APIs](#).

Create a Video Indexer account connected to Azure

5/15/2019 • 7 minutes to read • [Edit Online](#)

When creating a Video Indexer account, you can choose a free trial account (where you get a certain number of free indexing minutes) or a paid option (where you are not limited by the quota). With free trial, Video Indexer provides up to 600 minutes of free indexing to website users and up to 2400 minutes of free indexing to API users. With the paid option, you create a Video Indexer account that is connected to your Azure subscription and an Azure Media Services account. You pay for minutes indexed as well as the Media Account related charges.

This article shows how to create a Video Indexer account that's linked to an Azure subscription and an Azure Media Services account. The topic provides steps for connecting to Azure using the automatic (default) flow. It also shows how to connect to Azure manually (advanced).

Prerequisites

- An Azure subscription.

If you don't have an Azure subscription yet, sign up for [Azure Free Trial](#).

- An Azure Active Directory (AD) domain.

If you don't have an Azure AD domain, create this domain with your Azure subscription. For more information, see [Managing custom domain names in your Azure Active Directory](#)

- A user and member in your Azure AD domain. You'll use this member when connecting your Video Indexer account to Azure.

This user should be an Azure AD user with a work or school account, not a personal account, such as outlook.com, live.com, or hotmail.com.

NAME	USER NAME
JD John Doe	john.doe @microsoft.com

Additional prerequisites for automatic flow

A user and member in your Azure AD domain. You'll use this member when connecting your Video Indexer account to Azure.

This user should be a member in your Azure subscription with either an **Owner** role, or both **Contributor** and **User Access Administrator** roles. A user can be added twice, with 2 roles. Once with Contributor and once with User Access Administrator.

The screenshot shows the 'Access control (IAM)' blade in the Azure portal. On the left, there's a sidebar with links like Overview, Access control (IAM), Diagnose and solve problems, COST MANAGEMENT + BILLING, Partner information, and SETTINGS. The main area has a search bar and buttons for Add, Remove, Roles, Refresh, and Help. A table lists users with their names, types (All), roles (4 selected), and owner status. One user, 'John Doe' (john.doe@microsoft.com), is highlighted with a red box around the 'Owner' role.

Additional prerequisites for manual flow

Register the EventGrid resource provider using the Azure portal.

In the [Azure portal](#), go to **Subscriptions**->[subscription]->**ResourceProviders**.

Search for **Microsoft.Media** and **Microsoft.EventGrid**. If not in the "Registered" state, click **Register**. It takes a couple of minutes to register.

The screenshot shows the 'Resource providers' blade in the Azure portal. On the left, there's a sidebar with links like Create a resource, All services, Favorites, Dashboard, All resources, Resource groups, Recent, App Services, Virtual machines (classic), Virtual machines, SQL databases, Cloud services (classic), Subscriptions (highlighted with a red box), Azure Active Directory, Monitor, Security Center, Cost Management + Billing, Help + support, Advisor, and Key vaults. The main area shows a list of providers. One provider, 'Microsoft.EventGrid', is highlighted with a red box around its name and status. Its status is 'Not Registered' and the 'Register' button is also highlighted with a red box.

Connect to Azure

1. Browse to the [Video Indexer](#) website and sign in.

2. Click on the **Create new account** button:



3. When the subscriptions list appears, select the subscription you want to use.

Connect Video Indexer to an Azure subscription

Azure subscription *

Video Indexer account region * ⓘ

Select a region for your Azure Media Services resource and Azure Storage account.

Azure Media Services account * ⓘ

- Use existing resource Create new resource group

[Switch to manual configuration](#)

Connect

Cancel

4. Select an Azure region from the supported locations: West US 2, North Europe, or East Asia.

5. Under **Azure Media Services account**, choose one of these options:

- To create a new Media Services account, select **Create new resource group**. Provide a name for your resource group.

Azure will create your new account in your subscription, including a new Azure Storage account. Your new Media Services account has a default initial configuration with a Streaming Endpoint and 10 S3 Reserved Units.

- To use an existing Media Services account, select **Use existing resource**. From the accounts list, select your account.

Your Media Services account must have the same region as your Video Indexer account.

NOTE

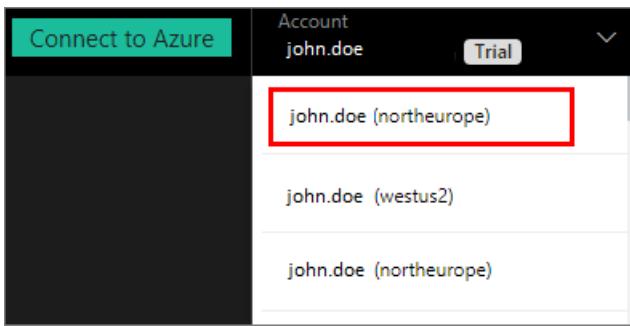
To minimize indexing duration and low throughput, it is highly recommended to adjust the type and number of **Reserved Units** to **10 S3 Reserved Units** in your Media Services account. See [Use portal to change Reserved Units](#).

- To manually configure your connection, click the **Switch to manual configuration** link.

For detailed information, see the [Connect to Azure manually](#) (advanced option) section that follows.

6. When you're done, choose **Connect**. This operation might take up to a few minutes.

After you're connected to Azure, your new Video Indexer account appears in the account list:



7. Browse to your new account

Connect to Azure manually (advanced option)

If the connection to Azure failed, you can attempt to troubleshoot the problem by connecting manually.

NOTE

It is highly recommended to have the following three accounts in the same region: the Video Indexer account that you are connecting with the Media Services account, as well as the Azure storage account connected to the same Media Services account.

Create and configure a Media Services account

1. Use the [Azure](#) portal to create an Azure Media Services account, as described in [Create an account](#).

When creating a storage account for your Media Services account, select **StorageV2** for account kind and **Geo-redundant (GRS)** for replication fields.

Media service	Choose storage account	Create storage account
CREATE MEDIA SERVICE ACCOUNT Account Name (lowercase only): amssaccount Subscription: VideoIndexer-ExternalDomain Resource Group: Create new (selected) Location: East Asia Storage Account: Please select a storage account	These are the storage accounts in the selected subscription and location 'East Asia'. Create new stamaccount (Stam-EA) - East Asia, Standard videoindexersa8a633bbe (vi-resourcegroup-APIE2E-EastAsia) - East Asia, Standard	Name: vitereststorageaccount ,core.windows.net Account kind: StorageV2 (general purpose v2) Performance: Standard Premium Replication: Geo-redundant storage (GRS)

NOTE

Make sure to write down the Media Services resource and account names. You will need it for the steps in the next section.

2. Adjust the type and number of [Reserved Units](#) to **10 S3 Reserved Units** in the Media Services account you created. See [Use portal to change Reserved Units](#).
3. Before you can play your videos in the Video Indexer web application, you must start the default **Streaming Endpoint** of the new Media Services account.

In the new Media Services account, click **Streaming endpoints**. Select the Streaming Endpoint and press start.

4. For Video Indexer to authenticate with Media Services API, an AD application needs to be created. The following steps guide you through the Azure AD authentication process described in [Get started with Azure AD authentication by using the Azure portal](#):

- In the new Media Services account, select [API access](#).
- Select [Service principal authentication method](#).
- Get the client ID and client secret, as described in the [Get the client ID and client secret](#) section.

After you select **Settings->Keys**, add **Description**, press **Save**, the key value gets populated.

If the key expires the account owner will have to contact Video Indexer support to renew the key.

NOTE

Make sure to write down the key value and the Application ID. You will need it for the steps in the next section.

Connect manually

In the **Connect Video Indexer to an Azure subscription** dialog of your [Video Indexer](#) page, select the **Switch to manual configuration** link.

In the dialog, provide the following information:

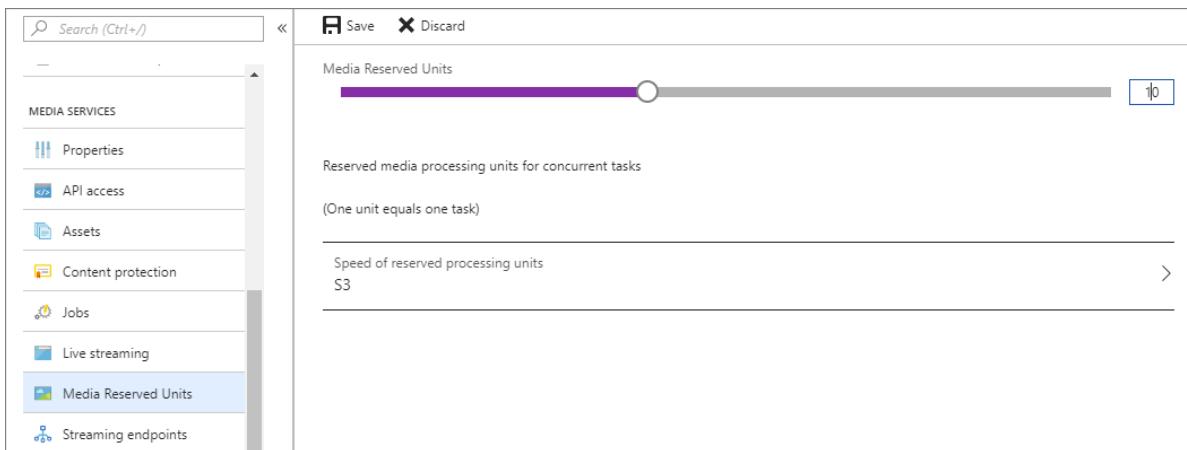
SETTING	DESCRIPTION
Video Indexer account region	The name of the Video Indexer account region. For better performance and lower costs, it is highly recommended to specify the name of the region where the Azure Media Services resource and Azure Storage account are located.
Azure Active Directory (AAD) tenant	The name of the Azure AD tenant, for example "contoso.onmicrosoft.com". The tenant information can be retrieved from the Azure portal. Place your cursor over the name of the signed-in user in the top-right corner. Find the name to the right of Domain .

SETTING	DESCRIPTION
Subscription ID	The Azure subscription under which this connection should be created. The subscription ID can be retrieved from the Azure portal. Click on All services in the left panel, and search for "subscriptions". Select Subscriptions and choose the desired ID from the list of your subscriptions.
Azure Media Services resource group name	The name for the resource group in which you created the Media Services account.
Media service resource name	The name of the Azure Media Services account that you created in the previous section.
Application ID	The Azure AD application ID (with permissions for the specified Media Services account) that you created in the previous section.
Application Key	The Azure AD application key that you created in the previous section.

Considerations

The following Azure Media Services related considerations apply:

- If you connect automatically, you see a new Resource Group, Media Services account, and a Storage account in your Azure subscription.
- If you connect automatically, Video Indexer sets the media **Reserved Units** to 10 S3 units:



- If you connect to an existing Media Services account, Video Indexer does not change the existing media **Reserved Units** configuration.

You might need to adjust the type and number of MediaReserved Units, according to your planned load. Keep in mind that if your load is high and you don't have enough units or speed, videos processing can result in timeout failures.

- If you connect to a new Media Services account, Video Indexer automatically starts the default **Streaming Endpoint** in it:

Name	Status	CDN	Type	Streaming Units
default	✓ Running	Enabled	Standard	N/A

Streaming Endpoints have a considerable startup time. Therefore, it may take several minutes from the time you connected your account to Azure, until your videos can be streamed and watched in the Video Indexer web application.

- If you connect to an existing Media Services account, Video Indexer does not change the default Streaming Endpoint configuration. If there is no running **Streaming Endpoint**, you will not be able to watch videos from this Media Services account or in Video Indexer.

Next steps

You can programmatically interact with your trial account and/or with your Video Indexer accounts that are connected to azure by following the instructions in: [Use APIs](#).

You should use the same Azure AD user you used when connecting to Azure.

Manage a Video Indexer account connected to Azure

5/15/2019 • 2 minutes to read • [Edit Online](#)

This article demonstrates how to manage a Video Indexer account that is connected to your Azure subscription and an Azure Media Services account.

NOTE

You have to be the Video Indexer account owner to do account configuration adjustments discussed in this topic.

Prerequisites

Connect your Video Indexer account to Azure, as described in [Connected to Azure](#).

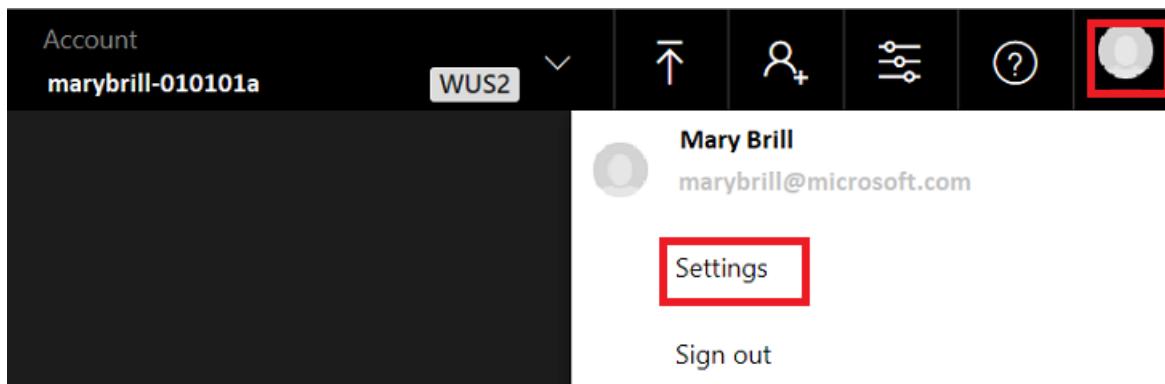
Make sure to follow [Prerequisites](#) and review [Considerations](#) in the article.

Examine account settings

This section examines settings of your Video Indexer account.

To view settings:

1. Click on the user icon in the top right corner and select **Settings**.



2. On the **Settings** page, select the **Account** tab.

If your Videos Indexer account is connected to Azure, you see the following:

- The name of the underlying Azure Media Services account.
- The number of indexing jobs running and queued.
- The number and type of allocated Reserved Units.

If your account needs some adjustments, you will see relevant errors and warnings about your account configuration on the **Settings** page. The messages contain links to exact places in Azure portal where you need to make changes. For more information, see the [errors and warnings](#) section that follows.

Auto-scale reserved units

The **Settings** page enables you to set the autoscaling of Media Reserved Units (RU). If the option is **On**, you can allocate the maximum number of RUs and be sure that Video Indexer stops/starts RUs automatically. With this option, you don't pay extra money for idle time but also do not wait for indexing jobs to complete a long time when

the indexing load is high.

Auto-scale does not scale below 1 RU or above the default limit of the Media Services account. In order to increase the limit, create a service request. For information about quotas and limitations and how to open a support ticket, see [Quotas and limitations](#).

Media service used
videoindexeramsd48c036e

Jobs
0 running, 0 waiting

① Your media service resource must have at least one [enabled streaming endpoint](#) to view uploaded videos. [Learn more](#)

Media reserved units
10 of 10 Premium units used

Autoscale
 On
This will scale up the reserved unit number on Azure but will not exceed the maximum units allocated to this account

[Manage in Azure portal](#)

Errors and warnings

If your account needs some adjustments, you see relevant errors and warnings about your account configuration on the **Settings** page. The messages contain links to exact places in Azure portal where you need to make changes. This section gives more details about the error and warning messages.

- Event Grid

You have to register the EventGrid resource provider using the Azure portal. In the [Azure portal](#), go to **Subscriptions** > [subscription] > **ResourceProviders** > **Microsoft.EventGrid**. If not in the **Registered** state, click **Register**. It takes a couple of minutes to register.

- Streaming Endpoint

Make sure the underlying Media Services account has the default **Streaming Endpoint** in a started state. Otherwise, you will not be able to watch videos from this Media Services account or in Video Indexer.

- Media Reserved Units

You must allocate Media Reserved Units on your Media Service resource in order to index videos. For optimal indexing performance, it's recommended to allocate at least 10 S3 Reserved Units. For pricing information, see the FAQ section of the [Media Services pricing](#) page.

Next steps

You can programmatically interact with your trial account and/or with your Video Indexer accounts that are connected to azure by following the instructions in: [Use APIs](#).

You should use the same Azure AD user you used when connecting to Azure.

Upload and index your videos

5/15/2019 • 7 minutes to read • [Edit Online](#)

When uploading videos with Video Indexer API, you have the following upload options:

- upload your video from a URL (preferred),
- send the video file as a byte array in the request body,
- Use existing Azure Media Services asset by providing the [asset ID](#) (supported in paid accounts only).

The article shows how to use the [Upload video](#) API to upload and index your videos based on a URL. The code sample in the article includes the commented out code that shows how to upload the byte array.

The article also discusses some of the parameters that you can set on the API to change the process and output of the API.

Once your video has been uploaded, Video Indexer, optionally encodes the video (discussed in the article). When creating a Video Indexer account, you can choose a free trial account (where you get a certain number of free indexing minutes) or a paid option (where you are not limited by the quota). With free trial, Video Indexer provides up to 600 minutes of free indexing to website users and up to 2400 minutes of free indexing to API users. With paid option, you create a Video Indexer account that is [connected to your Azure subscription and an Azure Media Services account](#). You pay for minutes indexed as well as the Media Account related charges.

Uploading considerations

- When uploading your video based on the URL (preferred) the endpoint must be secured with TLS 1.2 (or higher)
- The upload size with the URL option is limited to 30GB
- The request URL length is limited to 2048 characters
- The upload size with the byte array option is limited to 2GB
- The byte array option times out after 30 min
- The URL provided in the `videoURL` param needs to be encoded
- Indexing Media Services assets has the same limitation as indexing from URL
- Video Indexer has a max duration limit of 4 hours for a single file

TIP

It is recommended to use .NET framework version 4.6.2. or higher because older .NET frameworks do not default to TLS 1.2.

If you must use older .NET frameworks, add one line into your code before making the REST API call:

```
System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls | SecurityProtocolType.Tls11 |  
SecurityProtocolType.Tls12;
```

Configurations and params

This section describes some of the optional parameters and when you would want to set them.

externalID

This parameter enables you to specify an ID that will be associated with the video. The ID can be applied to external "Video Content Management" (VCM) system integration. The videos that are located in the Video Indexer portal can be searched using the specified external ID.

callbackUrl

A URL that is used to notify the customer (using a POST request) about the following events:

- Indexing state change:

- Properties:

NAME	DESCRIPTION
id	The video ID
state	The video state

- Example: <https://test.com/notifyme?projectName=MyProject&id=1234abcd&state=Processed>

- Person identified in video:

- Properties

NAME	DESCRIPTION
id	The video ID
faceId	The face ID that appears in the video index
knownPersonId	The person ID that is unique within a face model
personName	The name of the person

- Example: https://test.com/notifyme?projectName=MyProject&id=1234abcd&faceid=12&knownPersonId=CCA84350-89B7-4262-861C-3CAC796542A5&personName=Inigo_Montoya

Notes

- Video Indexer returns any existing parameters provided in the original URL.
- The provided URL must be encoded.

indexingPreset

Use this parameter if raw or external recordings contain background noise. This parameter is used to configure the indexing process. You can specify the following values:

- **Default** – Index and extract insights using both audio and video
- **AudioOnly** – Index and extract insights using audio only (ignoring video)
- **DefaultWithNoiseReduction** – Index and extract insights from both audio and video, while applying noise reduction algorithms on audio stream

Price depends on the selected indexing option.

priority

Videos are indexed by Video Indexer according to their priority. Use the **priority** parameter to specify the index priority. The following values are valid: **Low**, **Normal** (default), and **High**.

Priority parameter is only supported for paid accounts.

streamingPreset

Once your video has been uploaded, Video Indexer, optionally encodes the video. Then, proceeds to indexing, and analyzing the video. When Video Indexer is done analyzing, you will get a notification with the video ID.

When using the [Upload video](#) or [Re-Index Video](#) API, one of the optional parameters is `streamingPreset`. If you set `streamingPreset` to `Default`, `SingleBitrate`, or `AdaptiveBitrate`, the encoding process is triggered. Once the indexing and encoding jobs are done, the video is published so you can also stream your video. The Streaming Endpoint from which you want to stream the video must be in the **Running** state.

In order to run the indexing and encoding jobs, the [Azure Media Services account connected to your Video Indexer account](#) requires Reserved Units. For more information, see [Scaling Media Processing](#). Since these are compute intensive jobs, S3 unit type is highly recommended. The number of RUs defines the max number of jobs that can run in parallel. The baseline recommendation is 10 S3 RUs.

If you only want to index your video but not encode it, set `streamingPreset` to `NoStreaming`.

videoUrl

A URL of the video/audio file to be indexed. The URL must point at a media file (HTML pages are not supported). The file can be protected by an access token provided as part of the URI and the endpoint serving the file must be secured with TLS 1.2 or higher. The URL needs to be encoded.

If the `videoUrl` is not specified, the Video Indexer expects you to pass the file as a multipart/form body content.

Code sample

The following C# code snippet demonstrates the usage of all the Video Indexer APIs together.

```
public async Task Sample()
{
    var apiUrl = "https://api.videoindexer.ai";
    var location = "westus2";
    var apiKey = "...";

    System.Net.ServicePointManager.SecurityProtocol =
        System.Net.ServicePointManager.SecurityProtocol | System.Net.SecurityProtocolType.Tls12;

    // create the http client
    var handler = new HttpClientHandler();
    handler.AllowAutoRedirect = false;
    var client = new HttpClient(handler);
    client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", apiKey);

    // obtain account information and access token
    string queryParams = CreateQueryString(
        new Dictionary<string, string>()
    {
        {"generateAccessTokens", "true"},
        {"allowEdit", "true"},
    });
    HttpResponseMessage result = await client.GetAsync($"{apiUrl}/auth/trial/Accounts?{queryParams}");
    var json = await result.Content.ReadAsStringAsync();
    var accounts = JsonConvert.DeserializeObject<AccountContractsSlim[]>(json);
    // take the relevant account, here we simply take the first
    var accountInfo = accounts.First();

    // we will use the access token from here on, no need for the apim key
    client.DefaultRequestHeaders.Remove("Ocp-Apim-Subscription-Key");

    // upload a video
    var content = new MultipartFormDataContent();
    Debug.WriteLine("Uploading...");
    // get the video from URL
    var videoUrl = "VIDEO_URL"; // replace with the video URL

    // as an alternative to specifying video URL, you can upload a file.
    // remove the videoUrl parameter from the query params below and add the following lines:
    //FileStream video =File.OpenRead(Globals.VIDEOFILE_PATH);
```

```

//byte[] buffer =newbyte[video.Length];
//video.Read(buffer, 0, buffer.Length);
//content.Add(newByteArrayContent(buffer));

queryParams = CreateQueryString(
    new Dictionary<string, string>()
{
    {"accessToken", accountInfo.AccessToken},
    {"name", "video_name"},
    {"description", "video_description"},
    {"privacy", "private"},
    {"partition", "partition"},
    {"videoUrl", videoUrl},
});
var uploadRequestResult = await client.PostAsync($""
{apiUrl}/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos?{queryParams}", content);
var uploadResult = await uploadRequestResult.Content.ReadAsStringAsync();

// get the video ID from the upload result
string videoId = JsonConvert.DeserializeObject<dynamic>(uploadResult)["id"];
Debug.WriteLine("Uploaded");
Debug.WriteLine("Video ID:");
Debug.WriteLine(videoId);

// wait for the video index to finish
while (true)
{
    await Task.Delay(10000);

    queryParams = CreateQueryString(
        new Dictionary<string, string>()
    {
        {"accessToken", accountInfo.AccessToken},
        {"language", "English"},
    });
}

var videoGetIndexRequestResult = await client.GetAsync($""
{apiUrl}/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos/{videoId}/Index?{queryParams}");
var videoGetIndexResult = await videoGetIndexRequestResult.Content.ReadAsStringAsync();

string processingState = JsonConvert.DeserializeObject<dynamic>(videoGetIndexResult)["state"];

Debug.WriteLine("");
Debug.WriteLine("State:");
Debug.WriteLine(processingState);

// job is finished
if (processingState != "Uploaded" && processingState != "Processing")
{
    Debug.WriteLine("");
    Debug.WriteLine("Full JSON:");
    Debug.WriteLine(videoGetIndexResult);
    break;
}
}

// search for the video
queryParams = CreateQueryString(
    new Dictionary<string, string>()
{
    {"accessToken", accountInfo.AccessToken},
    {"id", videoId},
});
var searchRequestResult = await client.GetAsync($""
{apiUrl}/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos/Search?{queryParams}");
var searchResult = await searchRequestResult.Content.ReadAsStringAsync();
Debug.WriteLine("");
Debug.WriteLine("Search:");

```

```

Debug.WriteLine(searchResult);

// Generate video access token (used for get widget calls)
client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", apiKey);
var videoTokenRequestResult = await client.GetAsync(${apiUrl}/auth/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos/{videoId}/AccessToken?allowEdit=true");
var videoAccessToken = (await videoTokenRequestResult.Content.ReadAsStringAsync()).Replace("\"", "");
client.DefaultRequestHeaders.Remove("Ocp-Apim-Subscription-Key");

// get insights widget url
queryParams = CreateQueryString(
    new Dictionary<string, string>()
{
    {"accessToken", videoAccessToken},
    {"widgetType", "Keywords"},
    {"allowEdit", "true"},
});
var insightsWidgetRequestResult = await client.GetAsync(${apiUrl}/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos/{videoId}/InsightsWidget?{queryParams}");
var insightsWidgetLink = insightsWidgetRequestResult.Headers.Location;
Debug.WriteLine("Insights Widget url:");
Debug.WriteLine(insightsWidgetLink);

// get player widget url
queryParams = CreateQueryString(
    new Dictionary<string, string>()
{
    {"accessToken", videoAccessToken},
});
var playerWidgetRequestResult = await client.GetAsync(${apiUrl}/{accountInfo.Location}/Accounts/{accountInfo.Id}/Videos/{videoId}/PlayerWidget?{queryParams}");
var playerWidgetLink = playerWidgetRequestResult.Headers.Location;
Debug.WriteLine("");
Debug.WriteLine("Player Widget url:");
Debug.WriteLine(playerWidgetLink);
}

private string CreateQueryString(IDictionary<string, string> parameters)
{
    var queryParameters = HttpUtility.ParseQueryString(string.Empty);
    foreach (var parameter in parameters)
    {
        queryParameters[parameter.Key] = parameter.Value;
    }

    return queryParameters.ToString();
}

public class AccountContractSlim
{
    public Guid Id { get; set; }
    public string Name { get; set; }
    public string Location { get; set; }
    public string AccountType { get; set; }
    public string Url { get; set; }
    public string AccessToken { get; set; }
}

```

Common errors

The status codes listed in the following table may be returned by the Upload operation.

STATUS CODE	ERRORTYPE (IN RESPONSE BODY)	DESCRIPTION
-------------	------------------------------	-------------

STATUS CODE	ERROR TYPE (IN RESPONSE BODY)	DESCRIPTION
400	VIDEO_ALREADY_IN_PROGRESS	Same video is already in progress of being processed in the given account.
400	VIDEO_ALREADY_FAILED	Same video failed to process in the given account less than 2 hours ago. API clients should wait at least 2 hours before re-uploading a video.

Next steps

[Examine the Azure Video Indexer output produced by API](#)

Examine the Video Indexer output produced by API

5/15/2019 • 13 minutes to read • [Edit Online](#)

When you call the **Get Video Index** API and the response status is OK, you get a detailed JSON output as the response content. The JSON content contains details of the specified video insights. The insights include dimensions like: transcripts, OCRs, faces, topics, blocks, etc. The dimensions have instances of time ranges that show when each dimension appeared in the video.

You can also visually examine the video's summarized insights by pressing the **Play** button on the video on the [Video Indexer](#) website. For more information, see [View and edit video insights](#).

The screenshot shows the Video Indexer interface with a video player at the top displaying a man (Satya Nadella) speaking. Below the video are sections for Insights, Timeline, People, Keywords, Emotions, and Keyframes. The People section highlights Satya Nadella as the Microsoft CEO appearing in 17% of the video. The Keywords section lists terms like graph, azure, microsoft, app services, cloud, location services. The Emotions section shows a chart with 33% Fear, 28% Anger, 7% Sadness, and 12% Joy. The Keyframes section displays a grid of frames from the video. At the bottom, there are links to similar videos like Windows Holographic Technical Session and Build 2018 Keynote.

This article examines the JSON content returned by the **Get Video Index** API.

NOTE

Expiration of all the access tokens in Video Indexer is one hour.

Root elements

NAME	DESCRIPTION
accountId	The playlist's VI account ID.
id	The playlist's ID.

NAME	DESCRIPTION
name	The playlist's name.
description	The playlist's description.
userName	The name of the user who created the playlist.
created	The playlist's creation time.
privacyMode	The playlist's privacy mode (Private/Public).
state	The playlist's (uploaded, processing, processed, failed, quarantined).
isOwned	Indicates whether the playlist was created by the current user.
isEditable	Indicates whether the current user is authorized to edit the playlist.
isBase	Indicates whether the playlist is a base playlist (a video) or a playlist made of other videos (derived).
durationInSeconds	The total duration of the playlist.
summarizedInsights	Contains one summarizedInsights .
videos	A list of videos constructing the playlist. If this playlist is constructed of time ranges of other videos (derived), the videos in this list will contain only data from the included time ranges.

```
{
  "accountId": "bca61527-1221-bca6-1527-a90000002000",
  "id": "abc3454321",
  "name": "My first video",
  "description": "I am trying VI",
  "userName": "Some name",
  "created": "2018/2/2 18:00:00.000",
  "privacyMode": "Private",
  "state": "Processed",
  "isOwned": true,
  "isEditable": false,
  "isBase": false,
  "durationInSeconds": 120,
  "summarizedInsights" : { . . . }
  "videos": [{ . . . }]
}
```

summarizedInsights

This section shows the summary of the insights.

ATTRIBUTE	DESCRIPTION
name	The name of the video. For example, Azure Monitor.
id	The ID of the video. For example, 63c6d532ff.
privacyMode	Your breakdown can have one of the following modes: Private , Public , Public - the video is visible to everyone in your account and anyone that has a link to the video. Private - the video is visible to everyone in your account.
duration	Contains one duration that describes the time an insight occurred. Duration is in seconds.
thumbnailVideoId	The ID of the video from which the thumbnail was taken.
thumbnailld	The video's thumbnail ID. To get the actual thumbnail, call Get-Thumbnail and pass it thumbnailVideoId and thumbnailld.
faces	May contain zero or more faces. For more detailed information, see faces .
keywords	May contain zero or more keywords. For more detailed information, see keywords .
sentiments	May contain zero or more sentiments. For more detailed information, see sentiments .
audioEffects	May contain zero or more audioEffects. For more detailed information, see audioEffects .
labels	May contain zero or more labels. For detailed more information, see labels .
brands	May contain zero or more brands. For more detailed information, see brands .
statistics	For more detailed information, see statistics .
emotions	May contain zero or more emotions. For More detailed information, see emotions .
topics	May contain zero or more topics. The topics dimension.

videos

NAME	DESCRIPTION
accountId	The video's VI account ID.
id	The video's ID.
name	The video's name.

NAME	DESCRIPTION
state	The video's state (uploaded, processing, processed, failed, quarantined).
processingProgress	The processing progress during processing (for example, 20%).
failureCode	The failure code if failed to process (for example, 'UnsupportedFileType').
failureMessage	The failure message if failed to process.
externalId	The video's external ID (if specified by the user).
externalUrl	The video's external url (if specified by the user).
metadata	The video's external metadata (if specified by the user).
isAdult	Indicates whether the video was manually reviewed and identified as an adult video.
insights	The insights object. For more information, see insights .
thumbnailld	The video's thumbnail ID. To get the actual thumbnail call Get-Thumbnail and pass it the video ID and thumbnailld.
publishedUrl	A url to stream the video.
publishedUrlProxy	A url to stream the video from (for Apple devices).
viewToken	A short lived view token for streaming the video.
sourceLanguage	The video's source language.
language	The video's actual language (translation).
indexingPreset	The preset used to index the video.
streamingPreset	The preset used to publish the video.
linguisticModelId	The CRIS model used to transcribe the video.
statistics	For more information, see statistics .

```
{
  "videos": [
    {
      "accountId": "2cbbed36-1972-4506-9bc7-55367912df2d",
      "id": "142a356aa6",
      "state": "Processed",
      "privacyMode": "Private",
      "processingProgress": "100%",
      "failureCode": "General",
      "failureMessage": "",
      "externalId": null,
      "externalUrl": null,
      "metadata": null,
      "insights": {. . . },
      "thumbnailId": "89d7192c-1dab-4377-9872-473eac723845",
      "publishedUrl": "https://videomediastreaming.mediaservices.windows.net:443/d88a652d-334b-4a66-a294-3826402100cd/Xamarine.ism/manifest",
      "publishedProxyUrl": null,
      "viewToken": "Bearer=<token>",
      "sourceLanguage": "En-US",
      "language": "En-US",
      "indexingPreset": "Default",
      "linguisticModelId": "00000000-0000-0000-0000-000000000000"
    }
  ],
}
}
```

insights

The insights are a set of dimensions (for example, transcript lines, faces, brands, etc.), where each dimension is a list of unique elements (for example, face1, face2, face3), and each element has its own metadata and a list of its instances (which are time ranges with additional optional metadata).

A face might have an ID, a name, a thumbnail, other metadata, and a list of its temporal instances (for example: 00:00:05 – 00:00:10, 00:01:00 – 00:02:30 and 00:41:21 – 00:41:49.) Each temporal instance can have additional metadata. For example, the face's rectangle coordinates (20,230,60,60).

VERSION	THE CODE VERSION
sourceLanguage	The video's source language (assuming one master language). In the form of a BCP-47 string.
language	The insights language (translated from the source language). In the form of a BCP-47 string.
transcript	The transcript dimension.
ocr	The OCR dimension.
keywords	The keywords dimension.
blocks	May contain one or more blocks
faces	The faces dimension.
labels	The labels dimension.
shots	The shots dimension.
brands	The brands dimension.

VERSION	THE CODE VERSION
audioEffects	The <code>audioEffects</code> dimension.
sentiments	The <code>sentiments</code> dimension.
visualContentModeration	The <code>visualContentModeration</code> dimension.
textualContentModeration	The <code>textualContentModeration</code> dimension.
emotions	The <code>emotions</code> dimension.
topics	The <code>topics</code> dimension.

Example:

```
{
  "version": "0.9.0.0",
  "sourceLanguage": "en-US",
  "language": "es-ES",
  "transcript": ...,
  "ocr": ...,
  "keywords": ...,
  "faces": ...,
  "labels": ...,
  "shots": ...,
  "brands": ...,
  "audioEffects": ...,
  "sentiments": ...,
  "visualContentModeration": ...,
  "textualContentModeration": ...
}
```

blocks

ATTRIBUTE	DESCRIPTION
id	ID of the block.
instances	A list of time ranges of this block.

transcript

NAME	DESCRIPTION
id	The line ID.
text	The transcript itself.
language	The transcript language. Intended to support transcript where each line can have a different language.
instances	A list of time ranges where this line appeared. If the instance is transcript, it will have only 1 instance.

Example:

```

"transcript": [
{
  "id": 0,
  "text": "Hi I'm Doug from office.",
  "language": "en-US",
  "instances": [
    {
      "start": "00:00:00.510000",
      "end": "00:00:02.720000"
    }
  ]
},
{
  "id": 1,
  "text": "I have a guest. It's Michelle.",
  "language": "en-US",
  "instances": [
    {
      "start": "00:00:02.720000",
      "end": "00:00:03.960000"
    }
  ]
}
]

```

ocr

NAME	DESCRIPTION
id	The OCR line ID.
text	The OCR text.
confidence	The recognition confidence.
language	The OCR language.
instances	A list of time ranges where this OCR appeared (the same OCR can appear multiple times).
height	The height of the OCR rectangle
top	The top location in px
left	The left location in px
width	The width of the OCR rectangle

```

"ocr": [
  {
    "id": 0,
    "text": "LIVE FROM NEW YORK",
    "confidence": 675.971,
    "height": 35,
    "language": "en-US",
    "left": 31,
    "top": 97,
    "width": 400,
    "instances": [
      {
        "start": "00:00:26",
        "end": "00:00:52"
      }
    ]
  }
],

```

keywords

NAME	DESCRIPTION
id	The keyword ID.
text	The keyword text.
confidence	The keyword's recognition confidence.
language	The keyword language (when translated).
instances	A list of time ranges where this keyword appeared (a keyword can appear multiple times).

```

{
  id: 0,
  text: "technology",
  confidence: 1,
  language: "en-US",
  instances: [
    {
      adjustedStart: "0:05:15.782",
      adjustedEnd: "0:05:16.249",
      start: "0:05:15.782",
      end: "0:05:16.249"
    },
    {
      adjustedStart: "0:04:54.761",
      adjustedEnd: "0:04:55.228",
      start: "0:04:54.761",
      end: "0:04:55.228"
    }
  ]
}

```

faces

NAME	DESCRIPTION
id	The face ID.

NAME	DESCRIPTION
name	The name of the face. It can be 'Unknown #0, an identified celebrity or a customer trained person.
confidence	The face identification confidence.
description	A description of the celebrity.
thumbnailld	The ID of the thumbnail of that face.
knownPersonId	If it is a known person, its internal ID.
referenceld	If it is a Bing celebrity, its Bing ID.
referenceType	Currently, just Bing.
title	If it is a celebrity, its title (for example "Microsoft's CEO").
imageUrl	If it is a celebrity, its image url.
instances	These are instances of where the face appeared in the given time range. Each instance also has a thumbnailsId.

```

"faces": [
  {
    "id": 2002,
    "name": "Xam 007",
    "confidence": 0.93844,
    "description": null,
    "thumbnailId": "00000000-aee4-4be2-a4d5-d01817c07955",
    "knownPersonId": "8340004b-5cf5-4611-9cc4-3b13cca10634",
    "referenceId": null,
    "title": null,
    "imageUrl": null,
    "instances": [
      {
        "thumbnailsIds": ["00000000-9f68-4bb2-ab27-3b4d9f2d998e",
        "cef03f24-b0c7-4145-94d4-a84f81bb588c"],
        "adjustedStart": "00:00:07.2400000",
        "adjustedEnd": "00:00:45.6780000",
        "start": "00:00:07.2400000",
        "end": "00:00:45.6780000"
      },
      {
        "thumbnailsIds": ["00000000-51e5-4260-91a5-890fa05c68b0"],
        "adjustedStart": "00:10:23.9570000",
        "adjustedEnd": "00:10:39.2390000",
        "start": "00:10:23.9570000",
        "end": "00:10:39.2390000"
      }
    ]
  }
]

```

labels

NAME	DESCRIPTION
id	The label ID.
name	The label name (for example, 'Computer', 'TV').

NAME	DESCRIPTION
language	The label name language (when translated). BCP-47
instances	A list of time ranges where this label appeared (a label can appear multiple times). Each instance has a confidence field.

```

"labels": [
  {
    "id": 0,
    "name": "person",
    "language": "en-US",
    "instances": [
      {
        "confidence": 1.0,
        "start": "00: 00: 00.000000",
        "end": "00: 00: 25.600000"
      },
      {
        "confidence": 1.0,
        "start": "00: 01: 33.867000",
        "end": "00: 01: 39.200000"
      }
    ]
  },
  {
    "name": "indoor",
    "language": "en-US",
    "id": 1,
    "instances": [
      {
        "confidence": 1.0,
        "start": "00: 00: 06.400000",
        "end": "00: 00: 07.467000"
      },
      {
        "confidence": 1.0,
        "start": "00: 00: 09.600000",
        "end": "00: 00: 10.667000"
      },
      {
        "confidence": 1.0,
        "start": "00: 00: 11.733000",
        "end": "00: 00: 20.267000"
      },
      {
        "confidence": 1.0,
        "start": "00: 00: 21.333000",
        "end": "00: 00: 25.600000"
      }
    ]
  }
]

```

scenes

NAME	DESCRIPTION
id	The scene ID.
instances	A list of time ranges of this scene (a scene can only have 1 instance).

```

"scenes": [
    {
        "id":0,
        "instances":[
            {
                "start":"0:00:00",
                "end":"0:00:06.34",
                "duration":"0:00:06.34"
            }
        ]
    },
    {
        "id":1,
        "instances":[
            {
                "start":"0:00:06.34",
                "end":"0:00:47.047",
                "duration":"0:00:40.707"
            }
        ]
    },
],
]

```

shots

NAME	DESCRIPTION
id	The shot ID.
keyFrames	A list of keyFrames within the shot (each has an ID and a list of instances time ranges). Each keyFrame instance has a thumbnailId field, which holds the keyFrame's thumbnail ID.
instances	A list of time ranges of this shot (a shot can only have 1 instance).

```

"shots": [
    {
        "id": 0,
        "keyFrames": [
            {
                "id": 0,
                "instances": [
                    {
                        "thumbnailId": "00000000-0000-0000-0000-000000000000",
                        "start": "0:00:00.209",
                        "end": "0:00:00.251",
                        "duration": "0:00:00.042"
                    }
                ]
            },
            {
                "id": 1,
                "instances": [
                    {
                        "thumbnailId": "00000000-0000-0000-0000-000000000000",
                        "start": "0:00:04.755",
                        "end": "0:00:04.797",
                        "duration": "0:00:00.042"
                    }
                ]
            }
        ],
        "instances": [
            {
                "start": "0:00:00",
                "end": "0:00:06.34",
                "duration": "0:00:06.34"
            }
        ]
    },
    {
        "id": 1,
        "keyFrames": [
            {
                "id": 0,
                "instances": [
                    {
                        "start": "0:00:00.209",
                        "end": "0:00:00.251",
                        "duration": "0:00:00.042"
                    }
                ]
            }
        ],
        "instances": [
            {
                "start": "0:00:04.755",
                "end": "0:00:04.797",
                "duration": "0:00:00.042"
            }
        ]
    }
]

```

brands

Business and product brand names detected in the speech to text transcript and/or Video OCR. This does not include visual recognition of brands or logo detection.

NAME	DESCRIPTION
id	The brand ID.
name	The brands name.
referenceId	The suffix of the brand wikipedia url. For example, "Target_Corporation" is the suffix of https://en.wikipedia.org/wiki/Target_Corporation .
referenceUrl	The brand's Wikipedia url, if exists. For example, https://en.wikipedia.org/wiki/Target_Corporation .
description	The brands description.
tags	A list of predefined tags that were associated with this brand.
confidence	The confidence value of the Video Indexer brand detector (0-1).

NAME	DESCRIPTION
instances	A list of time ranges of this brand. Each instance has a brandType, which indicates whether this brand appeared in the transcript or in OCR.
<pre> "brands": [{ "id": 0, "name": "MicrosoftExcel", "referenceId": "Microsoft_Excel", "referenceUrl": "http://en.wikipedia.org/wiki/Microsoft_Excel", "referenceType": "Wiki", "description": "Microsoft Excel is a sprea..", "tags": [], "confidence": 0.975, "instances": [{ "brandType": "Transcript", "start": "00: 00: 31.300000", "end": "00: 00: 39.060000" }] }, { "id": 1, "name": "Microsoft", "referenceId": "Microsoft", "referenceUrl": "http://en.wikipedia.org/wiki/Microsoft", "description": "Microsoft Corporation is...", "tags": ["competitors", "technology"], "confidence": 1.0, "instances": [{ "brandType": "Transcript", "start": "00: 01: 44", "end": "00: 01: 45.367000" }, { "brandType": "Ocr", "start": "00: 01: 54", "end": "00: 02: 45.367000" }] }] </pre>	

statistics

NAME	DESCRIPTION
CorrespondenceCount	Number of correspondences in the video.
SpeakerWordCount	The number of words per speaker.
SpeakerNumberOfFragments	The amount of fragments the speaker has in a video.

NAME	DESCRIPTION
SpeakerLongestMonolog	The speaker's longest monolog. If the speaker has silences inside the monolog it is included. Silence at the beginning and the end of the monolog is removed.
SpeakerTalkToListenRatio	The calculation is based on the time spent on the speaker's monolog (without the silence in between) divided by the total time of the video. The time is rounded to the third decimal point.

audioEffects

NAME	DESCRIPTION
id	The audio effect ID.
type	The audio effect type (for example, Clapping, Speech, Silence).
instances	A list of time ranges where this audio effect appeared.

```
"audioEffects": [
{
    "id": 0,
    "type": "Clapping",
    "instances": [
        {
            "start": "00:00:00",
            "end": "00:00:03"
        },
        {
            "start": "00:01:13",
            "end": "00:01:21"
        }
    ]
}]
```

sentiments

Sentiments are aggregated by their sentimentType field (Positive/Neutral/Negative). For example, 0-0.1, 0.1-0.2.

NAME	DESCRIPTION
id	The sentiment ID.
averageScore	The average of all scores of all instances of that sentiment type - Positive/Neutral/Negative
instances	A list of time ranges where this sentiment appeared.
sentimentType	The type can be 'Positive', 'Neutral', or 'Negative'.

```

"sentiments": [
{
  "id": 0,
  "averageScore": 0.87,
  "sentimentType": "Positive",
  "instances": [
    {
      "start": "00:00:23",
      "end": "00:00:41"
    }
  ]
}, {
  "id": 1,
  "averageScore": 0.11,
  "sentimentType": "Positive",
  "instances": [
    {
      "start": "00:00:13",
      "end": "00:00:21"
    }
  ]
}
]

```

visualContentModeration

The visualContentModeration block contains time ranges which Video Indexer found to potentially have adult content. If visualContentModeration is empty, there is no adult content that was identified.

Videos that are found to contain adult or racy content might be available for private view only. Users have the option to submit a request for a human review of the content, in which case the IsAdult attribute will contain the result of the human review.

NAME	DESCRIPTION
id	The visual content moderation ID.
adultScore	The adult score (from content moderator).
racyScore	The racy score (from content moderation).
instances	A list of time ranges where this visual content moderation appeared.

```

"VisualContentModeration": [
  {
    "id": 0,
    "adultScore": 0.00069,
    "racyScore": 0.91129,
    "instances": [
      {
        "start": "00:00:25.484000",
        "end": "00:00:25.526000"
      }
    ]
  },
  {
    "id": 1,
    "adultScore": 0.99231,
    "racyScore": 0.99912,
    "instances": [
      {
        "start": "00:00:35.536000",
        "end": "00:00:35.578000"
      }
    ]
  }
]

```

textualContentModeration

NAME	DESCRIPTION
id	The textual content moderation ID.
bannedWordsCount	The number of banned words.
bannedWordsRatio	The ratio from total number of words.

emotions

Video Indexer identifies emotions based on speech and audio cues. The identified emotion could be: joy, sadness, anger, or fear.

NAME	DESCRIPTION
id	The emotion ID.
type	The emotion moment that was identified based on speech and audio cues. The emotion could be: joy, sadness, anger, or fear.
instances	A list of time ranges where this emotion appeared.

```

"emotions": [
  {
    "id": 0,
    "type": "Fear",
    "instances": [
      {
        "adjustedStart": "0:00:39.47",
        "adjustedEnd": "0:00:45.56",
        "start": "0:00:39.47",
        "end": "0:00:45.56"
      },
      {
        "adjustedStart": "0:07:19.57",
        "adjustedEnd": "0:07:23.25"
      }
    ]
  }
]

```

```
    "adjustedEnd": "0:07:23.25",
    "start": "0:07:19.57",
    "end": "0:07:23.25"
  }],
},
{
  "id": 1,
  "type": "Anger",
  "instances": [
    {
      "adjustedStart": "0:03:55.99",
      "adjustedEnd": "0:04:05.06",
      "start": "0:03:55.99",
      "end": "0:04:05.06"
    },
    {
      "adjustedStart": "0:04:56.5",
      "adjustedEnd": "0:05:04.35",
      "start": "0:04:56.5",
      "end": "0:05:04.35"
    }
  ],
{
  "id": 2,
  "type": "Joy",
  "instances": [
    {
      "adjustedStart": "0:12:23.68",
      "adjustedEnd": "0:12:34.76",
      "start": "0:12:23.68",
      "end": "0:12:34.76"
    },
    {
      "adjustedStart": "0:12:46.73",
      "adjustedEnd": "0:12:52.8",
      "start": "0:12:46.73",
      "end": "0:12:52.8"
    },
    {
      "adjustedStart": "0:30:11.29",
      "adjustedEnd": "0:30:16.43",
      "start": "0:30:11.29",
      "end": "0:30:16.43"
    },
    {
      "adjustedStart": "0:41:37.23",
      "adjustedEnd": "0:41:39.85",
      "start": "0:41:37.23",
      "end": "0:41:39.85"
    }
  ],
{
  "id": 3,
  "type": "Sad",
  "instances": [
    {
      "adjustedStart": "0:13:38.67",
      "adjustedEnd": "0:13:41.3",
      "start": "0:13:38.67",
      "end": "0:13:41.3"
    },
    {
      "adjustedStart": "0:28:08.88",
      "adjustedEnd": "0:28:18.16",
      "start": "0:28:08.88",
      "end": "0:28:18.16"
    }
  ]
},
],
],
```

topics

Video Indexer makes inference of main topics from transcripts. When possible, the 1st-level [IPTC taxonomy](#) is included.

NAME	DESCRIPTION
id	The topic ID.
name	The topic name, for example: "Pharmaceuticals".
referenceId	Breadcrumbs reflecting the topics hierarchy. For example: "Health and wellbeing / Medicine and healthcare / Pharmaceuticals".
confidence	The confidence score in the range [0,1]. Higher is more confident.
language	The language used in the topic.
iptcName	The IPTC media code name, if detected.
instances	Currently, Video Indexer does not index a topic to time intervals, so the whole video is used as the interval.

```
"topics": [{"  
    "id": 0,  
    "name": "INTERNATIONAL RELATIONS",  
    "referenceId": "POLITICS AND GOVERNMENT/FOREIGN POLICY/INTERNATIONAL RELATIONS",  
    "referenceType": "VideoIndexer",  
    "confidence": 1,  
    "language": "en-US",  
    "instances": [{"  
        "adjustedStart": "0:00:00",  
        "adjustedEnd": "0:03:36.25",  
        "start": "0:00:00",  
        "end": "0:03:36.25"  
    }]  
}, {  
    "id": 1,  
    "name": "Politics and Government",  
    "referenceType": "VideoIndexer",  
    "iptcName": "Politics",  
    "confidence": 0.9041,  
    "language": "en-US",  
    "instances": [{"  
        "adjustedStart": "0:00:00",  
        "adjustedEnd": "0:03:36.25",  
        "start": "0:00:00",  
        "end": "0:03:36.25"  
    }]  
}, ...]
```

Next steps

[Video Indexer Developer Portal](#)

For information about how to embed widgets in your application, see [Embed Video Indexer widgets into your applications](#).

Find exact moments within videos

5/15/2019 • 2 minutes to read • [Edit Online](#)

This topic shows you the search options that enable you to find exact moments within videos.

1. Browse to the [Video Indexer](#) website and sign in.
2. Search among all videos in your account.

In the following example, we search for all videos that talk about security and in which Satya appears,

The screenshot shows the Video Indexer search interface. At the top, there are search bars for 'Find topics' containing 'secure' and 'Find people' containing 'satya'. Below these are filters for 'Find in' (Entire video), 'Find by owner' (Video Indexer), and 'Find by language' (All). A 'Search' button is at the bottom right of the search bar area. The main content area shows a grid of video thumbnails. Two thumbnails are visible: 'Build 2017 Keynote' and 'Ignite 2016 - Keynote'. Each thumbnail has a summary of its content below it. The 'Build 2017 Keynote' summary includes moments like 'more engagement in a secure way...' and 'Sa mga secure nasusukat 2'. The 'Ignite 2016 - Keynote' summary includes moments like '...are delivered in a secure way...', '...that all of this is secure...', and '...that all of this is secure...'. The interface also includes tabs for 'Account videos' and 'Sample videos', and buttons for 'Upload' and 'Refresh'.

3. Search the summarized insights of the video.

You can then search within a video by clicking **Play** on the video. Then, you can search within the video by selecting the **Search** tab.

In the following example, we search for "secure" inside the selected video.

The screenshot shows the Microsoft Video Indexer interface. At the top, there's a navigation bar with 'Create new account', 'Account viaccountname', 'Trial', and various icons. Below the navigation bar is a large video player window showing a dark video thumbnail. To the right of the video player is the 'Insights' and 'Timeline' tab, with 'Timeline' selected. A search bar contains the word 'secure'. Below the search bar is a button labeled 'Autoscroll off'. The main area displays a timeline of video clips, each with a timestamp and a description. The descriptions contain several instances of the word 'secure' highlighted in green. The timestamps range from 00:09:18 to 01:17:17.

Timestamp	Description
00:09:18	Users more engagement in a secure way.
00:21:53	In the secure scalable 2 way communication.
00:30:42	Drones deployed to secure area.
00:31:33	Saving everyone time and keeping a workflow moving the digital and physical worlds are coming together to help make everyone more safe secure .
01:02:31	Access to a secure , dedicated cache for your Azure applications.
01:02:39	Access to a secure , dedicated cache for your Azure applications.
01:07:50	Secure
01:10:13	If I have 1000 virtual machines in the cloud that can be a little overwhelming. I want to make sure that those things are secure .
01:16:03	Last month, we announced our new SQL Server 2017 release know SQL Server 2017 is the fastest most secure in most intelligent database on the planet.
01:17:17	Skill secure and fault tolerant without you having to configure anything.

If you click one of the results, the player brings you to that moment in the video. You can achieve the player/insights view and synchronization in your application. For more information, see [Embed Video Indexer widgets into your application](#).

4. Search the detailed breakdown of the video.

If you want to create your own clip based on the video that you found, press the **Edit** button. This page shows you video along with its insights as filters. For more information, see [View and edit Video Indexer insights](#).

You can search within the video to only show the lines you are interested in and use the side insights to filter the parts you want to see. When you finish, you can preview your clip and press **Publish** to create the new clip that appears in your gallery.

In the following example, we searched for the "mixed reality" text. We also applied additional filters, as shown in the screen below.

The screenshot shows the Video Indexer interface. On the left, there's a sidebar titled 'Filters' with sections for 'Keywords' (with a 'Select' button), 'Sentiments' (with a 'Select' button), 'People' (listing 'Catherine Porter' with a remove button), 'Speakers' (with a 'Select' button), and a checked 'Show transcript' option. The main area has a search bar with 'Vision Keynote: Intelligent Cloud and I...' and a 'Save' button. Below it, there's a 'Look for topics or people' input with 'mixed reality' typed in, and dropdowns for 'Language' (English) and 'Privacy' (Private). A video thumbnail for 'Vision Keynote: Intelligent Cloud and Intelligent Edge' (1h42m, 175 segments, forked from Ori Ziv) is shown. The video player controls include a play button, a timestamp (00:00 / 1:42:17), and icons for volume, battery, and network. Below the video player are buttons for 'Preview' and 'Publish'. A large central panel displays a timeline of the video with various segments. The first segment is highlighted with a pink box and contains the text: 'This is technology that connects them to other people through their work. So my teens been hard at work, and today we are incredibly excited to announce to new mixed reality business applications.' Below this, several other segments are listed, each preceded by an 'OCR' icon: 'Announcing two new mixed reality', 'Microsoft Remote Assist', 'Collaborate in Mixed Reality', 'to solve problems faster', 'Microsoft Layout', and 'with Mixed Reality'. At the bottom right of the main panel, there's a 'How to create a new highlight reel' section with four numbered steps: 1. Find moments by searching and filtering; 2. Add or remove parts by using the +/- buttons at the top right corner of each component; 3. Click the Preview button above to watch your new highlight reel; 4. Publish your highlight reel to share it. You can also rename your reel. A note says 'Note: You can always use the "Reload source" button on the sidebar to reset your selections'. There's also a checkbox for 'Got it. Don't show this again.'

Next steps

Once you find the video you want to work with, you can continue processing the video, as described in one of these topics:

- [Use your videos' deep insights](#)
- [Process content with Video Indexer REST API](#)
- [Embed visual widgets in your application](#)

See also

[Video Indexer overview](#)

Scenes, shots, and keyframes

5/15/2019 • 2 minutes to read • [Edit Online](#)

Video Indexer supports segmenting videos into temporal units based on structural and semantic properties. This capability enables customers to easily browse, manage, and edit their video content based on varying granularities. For example, based on scenes, shots, and keyframes, described in this topic. The **scene detection** feature is currently in preview.



Scene detection (preview)

Video Indexer determines when a scene changes in video based on visual cues. A scene depicts a single event and it is composed of a series of consecutive shots, which are semantically related. A scene thumbnail is the first keyframe of its underlying shot. Video indexer segments a video into scenes based on color coherence across consecutive shots and retrieves the beginning and end time of each scene. Scene detection is considered a challenging task as it involves quantifying semantic aspects of videos.

NOTE

Applicable to videos that contain at least 3 scenes.

Shot detection

Video Indexer determines when a shot changes in the video based on visual cues, by tracking both abrupt and gradual transitions in the color scheme of adjacent frames. The shot's metadata includes a start and end time, as well as the list of keyframes included in that shot. The shots are consecutive frames taken from the same camera at the same time.

Keyframe detection

Selects the frame(s) that best represent the shot. Keyframes are the representative frames selected from the entire video based on aesthetic properties (for example, contrast and sturdiness). Video Indexer retrieves a list of keyframe IDs as part of the shot's metadata, based on which customers can extract the keyframe thumbnail.

Keyframes are associated with shots in the output JSON.

Next steps

[Examine the Video Indexer output produced by the API](#)

View and edit Video Indexer insights

5/15/2019 • 2 minutes to read • [Edit Online](#)

This topic shows you how to view and edit the Video Indexer insights of a video.

1. Browse to the [Video Indexer](#) website and sign in.
2. Find a video from which you want to create your Video Indexer insights. For more information, see [Find exact moments within videos](#).
3. Press **Play**.

The page shows the video's summarized insights.

The screenshot displays the Video Indexer Insights interface. At the top, there's a video player showing a man speaking on stage. Below the player, the video title is "Ignite 2016 - Keynote". To the right of the video player, there are several sections: "18 People" (listing Satya Nadella as the Microsoft CEO), "17 Keywords" (including "graph", "azure", "microsoft", "app services", "cloud", "location services"), "Emotions" (showing percentages of fear, anger, sadness, and joy), and "102 Keyframes" (a grid of thumbnail images from the video). The bottom left shows a section for "More videos with similar people and keywords", listing other Microsoft events like "Build 2018 Keynote" and "Windows Holographic Technical Session".

4. View the summarized insights of the video.

Summarized insights show an aggregated view of the data: faces, keywords, sentiments. For example, you can see the faces of people and the time ranges each face appears in and the % of the time it is shown.

The player and the insights are synchronized. For example, if you click a keyword or the transcript line, the player brings you to that moment in the video. You can achieve the player/insights view and synchronization in your application. For more information, see [Embed Azure Indexer widgets into your application](#).

Next steps

[Use your videos' deep insights](#)

See also

Video Indexer overview

Use the Video Indexer editor to create projects

5/10/2019 • 5 minutes to read • [Edit Online](#)

Video Indexer website, enables you to use your videos' deep insights to: find the right media content, locate the parts that you're interested in, and use the results to create an entirely new project. Once created, the project can be rendered and downloaded from Video Indexer and be used in your own editing applications or downstream workflows.

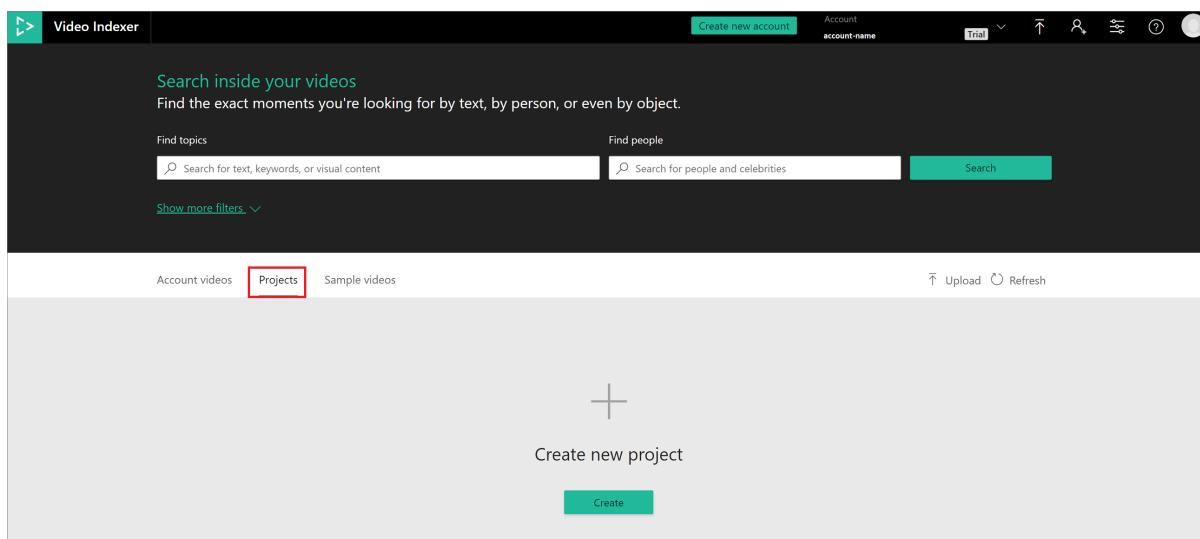
Some scenarios where you may find this feature useful are:

- Creating movie highlights for trailers.
- Using old clips of videos in news casts.
- Creating shorter content for social media.

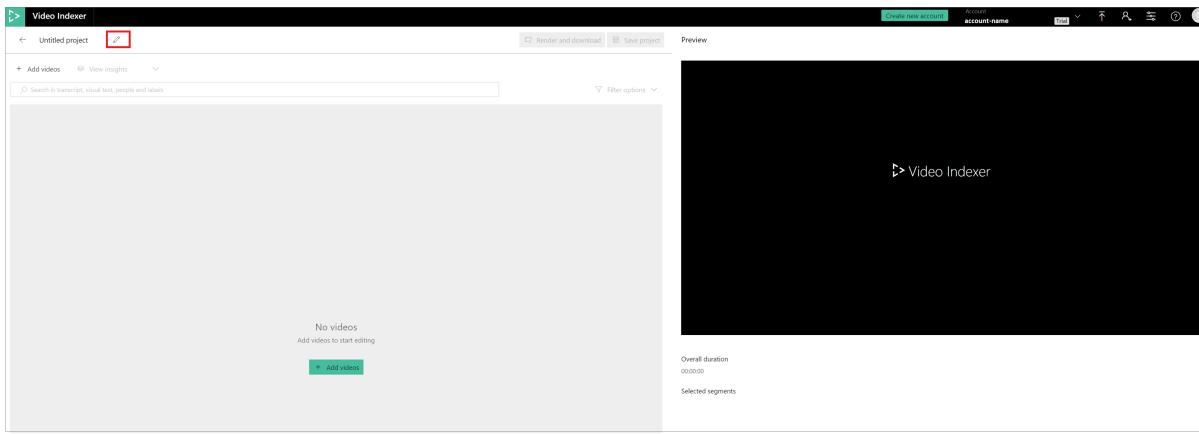
This article shows how to create a project from scratch and also how to create a project from a video in your account.

Create new project and manage videos

1. Browse to the [Video Indexer](#) website and sign in.
2. Select the **Projects** tab. If you have created projects before, you will see all of your other projects here.
3. Click **Create new project**.



4. Give your project a name by clicking on the pencil icon. Replace the text that says "Untitled project" with your project name and click on the check.



Add videos to the project

NOTE

Currently, projects may only contain videos indexed in the same language. Once you select a video in one language, you cannot add the videos in your account that are in a different language.

1. Add videos that you want to work with in this project by selecting **Add videos**.

You will see all the videos in your account and a search box that says "Search for text, keywords, or visual content". To search for videos that have a specified person, label, brand, keyword, or occurrence in the transcript and OCR.

For example, in the image below, we are looking for videos that mention "GitHub".

The screenshot shows a modal dialog titled 'Add videos'. At the top, it says 'A project can only contain videos that share the same indexed language'. Below is a search bar with 'GitHub' typed in. To the right of the search bar is a 'Filter results' dropdown. The main area lists two video results:

Thumbnail	Title	Language
	Azure Friday - Deploy to Azure using GitHub Actions	en-US
	Azure Friday - New App Service Deployment Center UX	en-US

At the bottom right of the dialog are 'Add' and 'Cancel' buttons.

You can further filter your results by selecting **Filter results**. You can filter to show videos that have a certain person in them or to specify that you only want to see video results that are in a certain language or have a specific owner.

You can also specify the scope of your query. For example, if you want to search "GitHub" in the OCR, select **Visual Text**.

Add videos



A project can only contain videos that share the same indexed language

Clear filters ^

Scope ▾

^

(-) (+)

Azure Friday - Deploy to Azure using the Azure portal

Azure Friday - New App Service Deployment Options

- Entire video
- Visual text only
- Transcript only
- Labels only
- Brands only

Add Cancel

You can layer multiple filters to your query. Use the +/- buttons to add/remove filters. Use **Clear filters** to remove all filters.

2. To add videos, select them and then select **Add**.
3. Now, you will see all of the videos you chose. These are the videos from which you are going to select clips for your project.

You can rearrange the order of the videos by dragging and dropping or by selecting the list menu button and selecting **Move down** or **Move up**. From the list menu, you will also be able to remove the video from this project.

Video Indexer Create new

← Azure Friday Highlights Render and download Save project

+ Add videos View insights ▼

Filter options ▾

Azure Friday - Deploy to Azure using the Azure portal No segments selected
Duration: 00:06:01

Azure Friday - New App Service Deployment Options No segments selected
Duration: 00:12:01

[...] ▼

Remove

Clear selection

Move down ▼

You have the option to add more videos to this project at any time by selecting **Add videos**. You can also add multiple occurrences of the same video to your project. You might want to do this if you want to show a clip from one video and then a clip from another and then another clip from the first video.

Select clips to use in your project

If you click on the downward arrow on the right side of each video, you will open up the insights in the video based on time stamps (clips of the video).

1. Select **View Insights** to customize which insights you want to see and which you don't want to see.

The screenshot shows the Video Indexer interface with the 'View insights' section selected. It displays two video clips: 'Azure Friday - Deploy to Azure usi...' and 'Azure Friday - New App Service De...'. Each clip has a duration and a note indicating 'No segments selected'. To the right, there's a preview window showing two men at a desk with laptops, and a summary section showing 'Overall duration 00:00:00' and 'Selected segments'.

2. To create queries for specific clips, use the search box that says "Search in transcript, visual text, people and labels".
3. Add filters to further specify details on what scenes you are looking for by selecting **Filter options**.

The screenshot shows the Video Indexer interface with the search bar containing 'GitHub'. The 'Filter options' dropdown is also visible.

For example, you may want to see clips where GitHub is mentioned while Donovan Brown is on the screen. For this, you need to add an "include" filter that has "People" as the type of insight. You then need to type in "Donovan Brown" in the search box for the filter.

The screenshot shows the Video Indexer interface with the 'Filter options' section open. The 'Include' dropdown is set to 'People', and 'Donovan Brown' is typed into the search bar. The results list shows two clips: 'Deploy to Azure usi...' and 'New App Service De...', both with 'No segments selected'. The preview window on the right shows a man at a desk.

If you want clips where GitHub is mentioned while Donovan Brown is *not* on the screen, you would simply change the "include" filter into an "exclude" filter using the dropdown.

4. Add a clip to your project by selecting the segment you want to add. You can unselect this clip by clicking on the segment again.

Add all segments of a video by clicking on the list menu option next to the video and selecting **Select all segments**.

Azure Friday - Deploy to Azure us... No segments selected
Duration: 00:06:01

- 00:00:00 - 00:00:07 Did you know there's a simple way to automate the deployment of your code from GitHub
- 00:00:09 - 00:00:16 How to use GitHub actions to deploy your solutions directly to Azure today on Azure Friday
- 00:00:20 - 00:00:25 I'm here with Gopi and we're going to be learning about how to use GitHub actions to deploy to Azure.
- 00:00:25 - 00:00:27 So what actually is a GitHub action.
- 00:05:49 - 00:05:54 Thank you so much so we're learning all about using GitHub actions to get to Azure here on Azure Friday.

You can clear all of your selection by selecting **Clear selection**.

TIP

As you are selecting and ordering your clips, you can preview the video in the player on the right side of the page.

Video Indexer

Create new account Account account-name Trial Preview

Azure Friday Highlights Render and download Save project

+ Add videos View insights

GitHub Clear filters

Include All insights Search all insights

Azure Friday - Deploy to Azure us... 2 segments selected Duration: 00:06:01

Azure Friday - New App Service De... 2 segments selected Duration: 00:12:01

Remember to save your project when you make changes by selecting **Save project**.

Render and download the project

NOTE

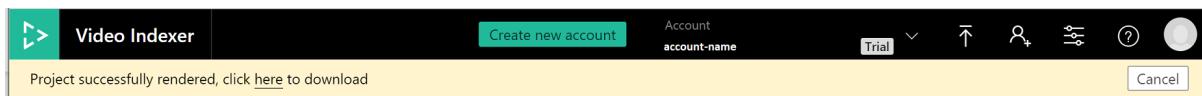
For Video Indexer paid accounts, rendering your project has encoding costs. Video Indexer trial accounts are limited to 5 hours of rendering.

- Once you are done, make sure that your project has been saved. You can now render this project. Select **Render and Download**.

The screenshot shows the Video Indexer interface with two video projects listed. The first project is titled "Azure Friday - Deploy to Azure usi..." with a duration of 00:06:01 and 2 segments selected. The second project is titled "Azure Friday - New App Service De..." with a duration of 00:12:01 and 2 segments selected. Both projects have their own preview thumbnails and dropdown menus.

There will be a popup that tells you that Video indexer will render a file and then the download link will be send to your email. Select Proceed.

You will also see a notification that the project is being rendered on top of the page. Once it is done being rendered, you will see a new notification that the project has been successfully rendered. Click the notification to download the project. It will download the project in mp4 format.



2. You can access saved projects from the **Projects** tab.

If you select this project, you see all the insights and the timeline of this project. If you select **Video editor**, you can continue making edits to this project. Edits include adding or removing videos and clips or renaming the project.

The screenshot shows the Video Indexer Insights and Timeline page for a project titled "Azure Friday - Global Azure Bootcamp". The main area displays a video thumbnail of two men in a studio setting. Below the video, there is a timeline bar with playback controls. To the right, the "Insights" tab is active, showing sections for "2 People" (Scott Hanselman) and "7 Topics" (Tourism / Visiting and travel, Visiting and travel advice, Startup, Microsoft windows os). There are also "Play previous" and "Play next" buttons.

Create a project from your video

You can create a new project directly from a video in your account.

1. Go to the **Library** tab of the Video Indexer website.
2. Open the video that you want to use to create your project. On the insights and timeline page, select the **Video editor** button.

This takes you to the same page that you used to create a new project. Unlike the new project, you see the

timestamped insights segments of the video, that you had started editing previously.

See also

[Video Indexer overview](#)

Embed Video Indexer widgets into your applications

7/1/2019 • 7 minutes to read • [Edit Online](#)

This article shows how you can embed Video Indexer widgets into your applications. Video Indexer supports embedding two types of widgets into your application: **Cognitive Insights** and **Player**.

Starting with version 2, the widget base URL includes the account's region. For example, an account in the West US region generates:

`https://wus2.videoindexer.ai/embed/insights/....`

Widget types

Cognitive Insights widget

A **Cognitive Insights** widget includes all visual insights that were extracted from your video indexing process. The insights widget supports the following optional URL params:

NAME	DEFINITION	DESCRIPTION
widgets	Strings separated by comma	Allows you to control the insights you want to render. Example: <code>https://www.videoindexer.ai/embed/insights/<account>?widgets=people,search</code> will render only people and brands UI insights Available options: people, keywords, annotations, brands, sentiments, transcript, search. not supported via URL at version=2 Note: The widgets URL param is not supported in version 2.
locale	A short language code	Controls the insights language. Default is <code>en</code> . For example: <code>language=de</code> .
tab	The default selected tab	Controls the insights tab that is rendered by default. <code>tab=timeline</code> renders the insights with the timeline tab selected.

Player widget

A **Player** widget enables you to stream the video using adaptive bit rate. The player widget supports the following optional URL params:

NAME	DEFINITION	DESCRIPTION
t	Seconds from the start	Makes the player start playing from the given time point. Example: <code>t=60</code> .
captions	A language code	Fetches the caption in the given language during the widget loading to be available in the captions menu. Example: <code>captions=en-US</code> .
showCaptions	A boolean value	Makes the player load with the captions already enabled. Example: <code>showCaptions=true</code> .
type		Activates an audio player skin (video part is removed). Example: <code>type=audio</code> .
autoplay	A boolean value	Indicates if the player should start playing the video when loaded (default is true). Example: <code>autoplay=false</code> .

NAME	DEFINITION	DESCRIPTION
language	A language code	Controls the player language (default is en-US) Example: <code>language=de-DE</code> .

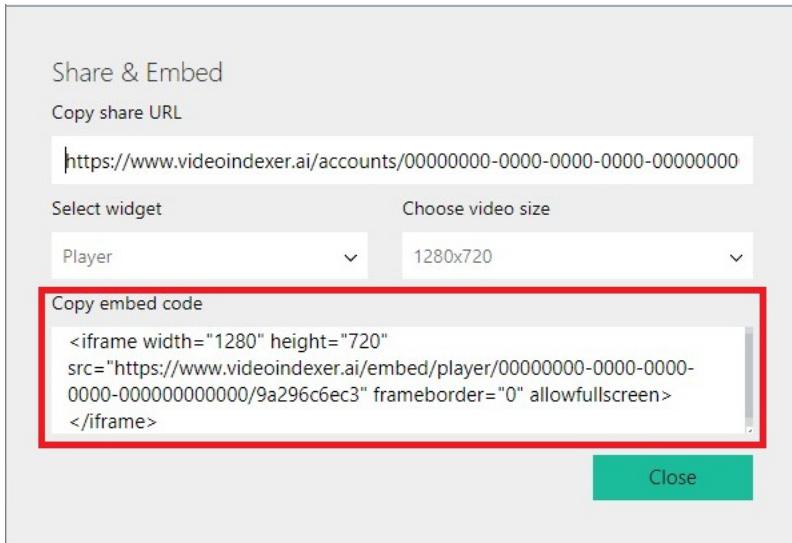
Embedding public content

1. Browse to the [Video Indexer](#) website and sign in.
2. Click on the video you want to work with.
3. Click the "embed" button that appears below the video.



After clicking the button, an embed modal will appear on the screen where you can choose what widget you want to embed in your application. Selecting a widget (**Player** or **Cognitive Insights**), generates the embedded code for you to paste in your application.

4. Choose the type of widget you want (**Cognitive Insights** or **Player**).
5. Copy the embed code, and add to your application.



NOTE

If you have issues with sharing your video URLs, try adding the 'location' parameter to the link. The parameter should be set to the [Azure regions in which Video Indexer exists](#). For example,

```
https://www.videoindexer.ai/accounts/00000000-0000-0000-0000-00000000/videos/b2b2c74b8e/?location=trial
```

Embedding private content

You can get embed codes from embed popups (as shown in the previous section) for **Public** videos only.

If you want to embed a **Private** video, you have to pass an access token in the **iframe's src** attribute:

```
https://www.videoindexer.ai/embed/[insights | player]/<accountId>/<videoId>/?accessToken=<accessToken>
```

Use the [Get Insights Widget](#) API to get the Cognitive Insights widget content, or use [Get Video Access Token](#) and add that as a query param to the URL, as shown above. Specify this URL as the **iframe's src** value.

If you want to provide editing insights capabilities (like we have in our web application) in your embedded widget, you will have to pass an access token with editing permissions. Use [Get Insights Widget](#) or [Get Video Access Token](#) with `&allowEdit=true`.

Widgets interaction

The **Cognitive Insights** widget can interact with a video on your application. This section shows how to achieve this interaction.

The screenshot shows the Video Indexer Cognitive Insights interface. At the top, there's a video player window displaying a man with glasses and a beard. Below the video player are several sections: '7 People' (listing one person named Doug Thomas), '3 keywords' (listing 'hollow lens game', 'magical alien technology', and 'alien invasion'), 'Sentiments' (showing 38.02% Positive and 7.96% Negative), and 'Keyframes' (a strip of thumbnail images from the video). The interface includes navigation controls like 'Play previous' and 'Play next'.

Cross-origin communications

To get Video Indexer widgets to communicate with other components, the Video Indexer service does the following:

- Uses the cross-origin communication HTML5 method **postMessage** and
- Validates the message across VideoIndexer.ai origin.

If you choose to implement your own player code and do the integration with **Cognitive Insights** widgets, it is your responsibility to validate the origin of the message that comes from VideoIndexer.ai.

Embed both types of widgets in your application / blog (recommended)

This section shows how to achieve interaction between two Video Indexer widgets so when a user clicks the insight control on your application, the player jumps to the relevant moment.

```
<script src="https://breakdown.blob.core.windows.net/public/vb.widgets.mediator.js"></script>
```

1. Copy the **Player** widget embed code.
2. Copy the **Cognitive Insights** embed code.
3. Add the **Mediator file** to handle the communication between the two widgets:

```
<script src="https://breakdown.blob.core.windows.net/public/vb.widgets.mediator.js"></script>
```

Now when a user clicks the insight control on your application, the player jumps to the relevant moment.

For more information, see [this demo](#).

Embed the Cognitive Insights widget and use Azure Media Player to play the content

This section shows how to achieve interaction between a **Cognitive Insights** widget and an Azure Media Player instance using the [AMP plugin](#).

1. Add a Video Indexer plugin for the AMP player.

```
<script src="https://breakdown.blob.core.windows.net/public/amp-vb.plugin.js"></script>
```

2. Instantiate Azure Media Player with the Video Indexer plugin.

```

// Init Source
function initSource() {
    var tracks = [
        kind: 'captions',
        // Here is how to load vtt from VI, you can replace it with your vtt url.
        src: this.getSubtitlesUrl("c4c1ad4c9a", "English"),
        srclang: 'en',
        label: 'English'
    ];

    myPlayer.src([
        {
            "src": "//amssamples.streaming.mediaservices.windows.net/91492735-c523-432b-ba01-
faba6c2206a2/AzureMediaServicesPromo.ism/manifest",
            "type": "application/vnd.ms-sstr+xml"
        }
    ], tracks);
}

// Init your AMP instance
var myPlayer = amp('vid1', { /* Options */
    "nativeControlsForTouch": false,
    autoplay: true,
    controls: true,
    width: "640",
    height: "400",
    poster: "",
    plugins: {
        videobreakdown: {}
    }
}, function () {
    // Activate the plugin
    this.videobreakdown({
        videoId: "c4c1ad4c9a",
        syncTranscript: true,
        syncLanguage: true
    });
    // Set the source dynamically
    initSource.call(this);
});

```

3. Copy the **Cognitive Insights** embed code.

You should be able now to communicate with your Azure Media Player.

For more information, see [this demo](#).

Embed Video Indexer Cognitive Insights widget and use your own player (could be any player)

If you use your own player, you have to take care of manipulating your player yourself in order to achieve the communication.

1. Insert your video player.

For example, a standard HTML5 player

```

<video id="vid1" width="640" height="360" controls autoplay preload>
    <source src="//breakdown.blob.core.windows.net/public/Microsoft%20HoloLens-%20RoboRaid.mp4" type="video/mp4" />
    Your browser does not support the video tag.
</video>

```

2. Embed the Cognitive Insights widget.

3. Implement communication for your player by listening to the "message" event. For example:

```

<script>

(function(){
    // Reference your player instance
    var playerInstance = document.getElementById('vid1');

    function jumpTo(evt) {
        var origin = evt.origin || evt.originalEvent.origin;

        // Validate that event comes from the videobreakdown domain.
        if ((origin === "https://www.videobreakdown.com") && evt.data.time !== undefined){

            // Here you need to call your player "jumpTo" implementation
            playerInstance.currentTime = evt.data.time;

            // Confirm arrival to us
            if ('postMessage' in window) {
                evt.source.postMessage({confirm: true, time: evt.data.time}, origin);
            }
        }
    }

    // Listen to message event
    window.addEventListener("message", jumpTo, false);

})()

</script>

```

For more information, see [this demo](#).

Adding subtitles

If you embed Video Indexer insights with your own AMP player, you can use the **GetVttUrl** method to get closed captions (subtitles). You can also call a javascript method from the Video Indexer AMP plugin **getSubtitlesUrl** (as shown earlier).

Customizing embeddable widgets

Cognitive insights widget

You can choose the types of insights you want by specifying them as a value to the following URL parameter added to the embed code you get (from API or from the web application): `&widgets=<list of wanted widgets>`.

The possible values are: people, keywords, sentiments, transcript, search.

For example, if you want to embed a widget containing only people and search insights the iframe embed URL will look like this:

```
https://www.videoindexer.ai/embed/insights/<accountId>/<videoId>/?widgets=people,search
```

The title of the iframe window can also be customized by providing `&title=<YourTitle>` to the iframe URL. (It will customize the html `<title>` value).

For example, if you want to give your iframe window the title "MyInsights", the URL will look like this:

```
https://www.videoindexer.ai/embed/insights/<accountId>/<videoId>/?title=MyInsights
```

Notice that this option is relevant only in cases when you need to open the insights in a new window.

Player widget

If you embed Video Indexer player, you can choose the size of the player by specifying the size of the iframe.

For example:

```
<iframe width="640" height="360" src="https://www.videoindexer.ai/embed/player/<accountId>/<videoId>/" frameborder="0" allowfullscreen>
```

By default Video Indexer player will have auto generated closed captions based on the transcript of the video that was extracted from the video with the source language that was selected when the video was uploaded.

If you want to embed with a different language, you can add `&captions=< Language | "all" | "false" >` to the embed player URL or put "all" as the value if you want to have all available languages captions. If you want the captions to be displayed by default, you can pass `&showCaptions=true`.

The embed URL then will look like this:

`https://www.videoindexer.ai/embed/player/<accountId>/<videoId>/?captions=italian`

If you want to disable captions, you can pass "false" as value for captions parameter.

Auto play – by default the player will start playing the video. you can choose not to by passing &autoplay=false to the embed URL above.

Next steps

For information about how to view and edit Video Indexer insights, see [this article](#).

Also, check out [Video indexer CodePen](#).

Customize a Person model with the Video Indexer website

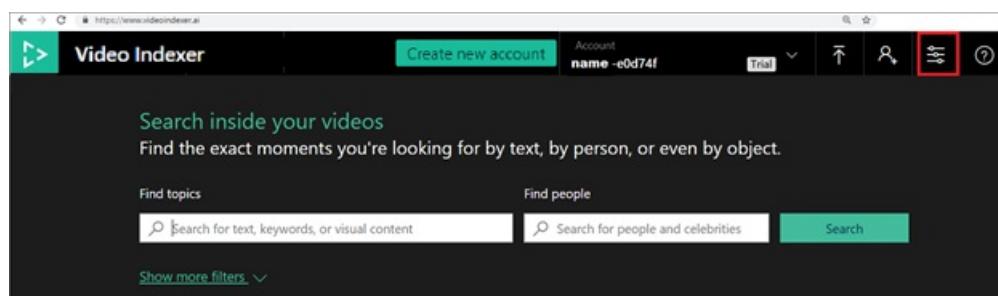
5/15/2019 • 8 minutes to read • [Edit Online](#)

Video Indexer supports celebrity recognition for video content. The celebrity recognition feature covers approximately one million faces based on commonly requested data source such as IMDB, Wikipedia, and top LinkedIn influencers. For a detailed overview, see [Customize a Person model in Video Indexer](#).

You can use the Video Indexer website to edit faces that were detected in a video, as described in this topic. You can also use the API, as described in [Customize a Person model using APIs](#).

Central management of Person models in your account

1. To view, edit, and delete the Person models in your account, browse to the Video Indexer website and sign in.
2. Click on the content model customization button on the top-right corner of the page.



3. Select the People tab.

You will see the Default Person model in your account. The Default Person model holds any faces you may have edited or changed in the insights of your videos for which you did not specify a custom Person model during indexing.

If you created other Person models, they will be listed on this page as well.

A screenshot of the 'Content model customization' page. The top navigation bar has tabs for 'People', 'Language', and 'Brands', with 'People' being the active tab. Below the tabs, there's a note: 'Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.' Underneath this note is a '+ Add model' button. A list of existing models is shown, with 'Default' currently selected. Other models listed include 'Basketball'. Each model entry has a dropdown arrow icon to its right.

Create a new Person model

1. Click the **+ Add model** button.

Content model customization

People Language Brands

Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.

+ Add model

Default

Basketball

- Enter the name of the model and click on the check button next to the name.

Content model customization

People Language Brands

Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.

+ Add model

Soccer ✓

Default

Basketball

You have created a new Person model. You can now add faces to the new Person model.

- Click the list menu button and choose **+ Add person**.

Content model customization

People Language Brands

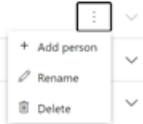
Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.

+ Add model

Soccer

Default

Basketball



Add a new person to a Person model

NOTE

Video Indexer allows you to add multiple people with the same name in a Person model. However, it is recommended that you give unique names to each person in your model for usability and clarity.

- To add a new face to a Person model, click on the list menu button next to the Person model that you want to add the face to.
- Click **+ Add person** from the menu.

Content model customization

People Language Brands

Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.

+ Add model

Soccer

Default

Basketball

+ Add person

Rename

Delete

A pop-up will prompt you to fill out the Person's details. Type in the name of the person and click on the check button.

Person's details

Enter name

Rename Add images Delete

Drag images here or choose files

Close

You can then choose from your file explorer or drag and drop the face images of the face. Video Indexer will take all standard image file types (ex: JPG, PNG, and more).

Video Indexer should be able to detect occurrences of this person in the future videos that you index and the current videos that you had already indexed, using the Person model to which you added this new face to. Recognition of the person in your current videos might take some time to take effect, as this is a batch process.

Rename a Person model

You can rename any Person model in your account including the Default Person model. Even if you rename your default Person model, it will still serve as the Default person model in your account.

1. Click on the list menu button next to the Person model that you want to rename.
2. Click **Rename** from the menu.

Content model customization

People Language Brands

Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.

+ Add model

Soccer

Default

Basketball

+ Add person

Rename

Delete

3. Click on the current name of the model and type in your new name.

Content model customization

People Language Brands

Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.

+ Add model

Soccer Team	✓	⋮ ⏺
Default		⋮ ⏺
Basketball		⋮ ⏺

4. Click on the check button for your model to be renamed.

Delete a Person model

You can delete any Person model that you created in your account. However, you cannot delete your Default person model.

1. Click **Delete** from the menu.

Content model customization

People Language Brands

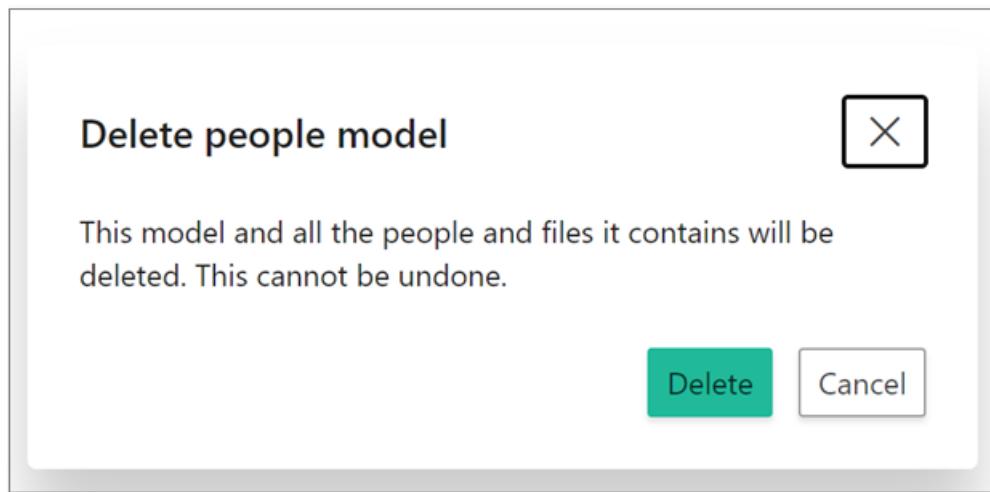
Enhance Video Indexer's model of a person by adding photos of that person, or other identifying metadata. Accepted file formats are JPG and PNG. If there are duplicate entities for the same person, you can merge them into one.

+ Add model

Soccer	⋮
Default	⋮
Basketball	⋮

+ Add person
Rename
Delete

A pop-up will show up and notify you that this action will delete the Person model and all of the people and files that it contains. This action cannot be undone.



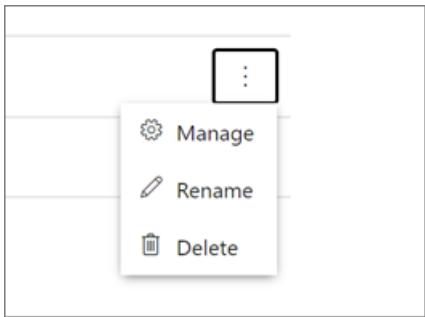
2. If you are sure, click delete again.

NOTE

The existing videos that were indexed using this (now deleted) Person model will not support the ability for you to update the names of the faces that appear in the video. You will be able to edit the names of faces in these videos only after you reindex them using another Person model. If you reindex without specifying a Person model, the default model will be used.

Manage existing people in a Person model

To look at the contents of any of your Person models, click on the arrow next to the name of the Person model. The drop-down shows you all of the people in that particular Person model. If you click on the list menu button next to each of the people, you see manage, rename, and delete options.



Rename a person

1. To rename a person in your Person model, click the list menu button and choose **Rename** from the list menu.
2. Click the current name of the person and type in your new name.
3. Click on the check button, and the person will be renamed.

Delete a person

1. To delete a person from your Person model, click the list menu button and choose **Delete** from the list menu.
2. A pop-up tells you that this action will delete the person and that this action cannot be undone.
3. Click Delete again and this will remove the person from the Person model.

Manage a person

If you click on **Manage**, you see all the faces that this Person model is being trained from. These faces come from occurrences of that person in videos that use this Person model or from images that you have manually uploaded.

You can add more faces to the person by clicking Add images.

You can use the manage pane to rename the person and to delete the person from the Person model.

Use a Person model to index a video

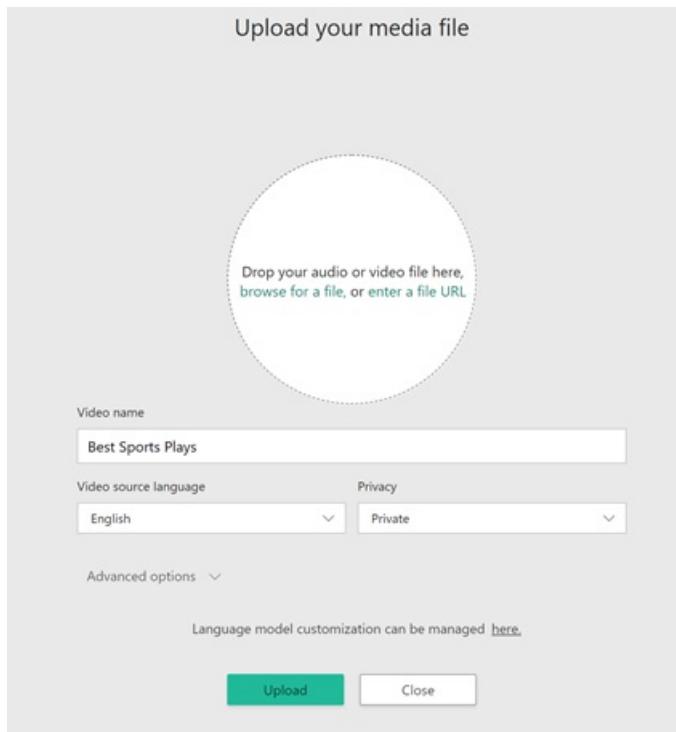
You can use a Person model to index your new video by assigning the Person model during the upload of the video.

To use your Person model on a new video, do the following:

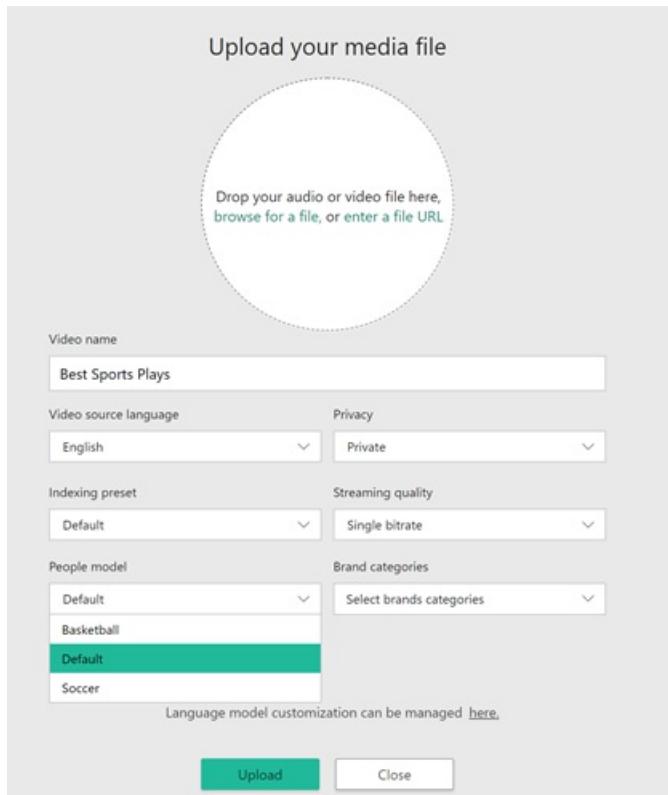
1. Click the **Upload** button on the top of the page.



2. Drop your video file in the circle or browse for your file.
3. Click the Advanced options arrow.



4. Click the drop-down and select the Person model that you created.



5. Click the Upload option in the bottom of the page, and your new video will be indexed using your Person model.

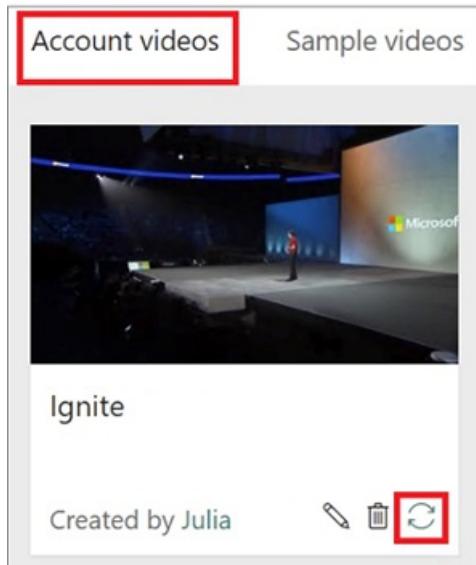
If you do not specify a Person model during the upload, Video Indexer will index the video using the Default Person model in your account.

Use a Person model to reindex a video

To use a Person model to reindex a video in your collection, go to your Account videos on the Video Indexer home page and hover over the name of the video that you want to reindex.

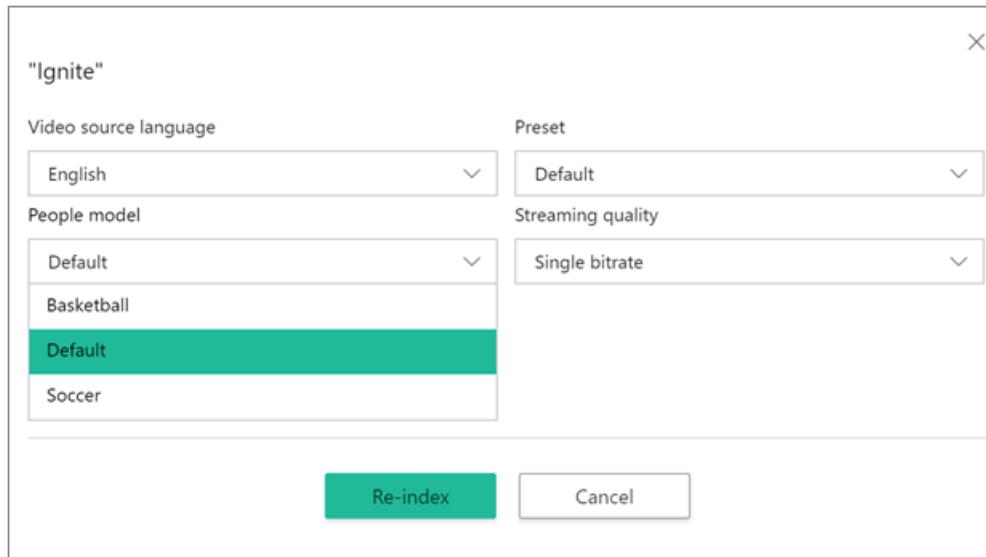
You see options to edit, delete, and reindex your video.

1. Click the option to reindex your video.



You can now select the Person model to reindex your video with.

2. Click the drop-down and select the Person model that you want to use.



3. Click the **Reindex** button, and your video will be reindexed using your Person model.

Any new edits that you make to the faces detected and recognized in the video that you just reindexed will be saved in the Person model that you used to reindex the video.

Managing people in your videos

You can manage the faces that are detected and people that were recognized in the videos that you index by editing and deleting faces.

Deleting a face, removes a specific face from the insights of the video.

Editing a face, renames a face that is detected and possibly recognized in your video. When you edit a face in your video, that name is saved as a person entry in the Person model that was assigned to the video during upload and indexing.

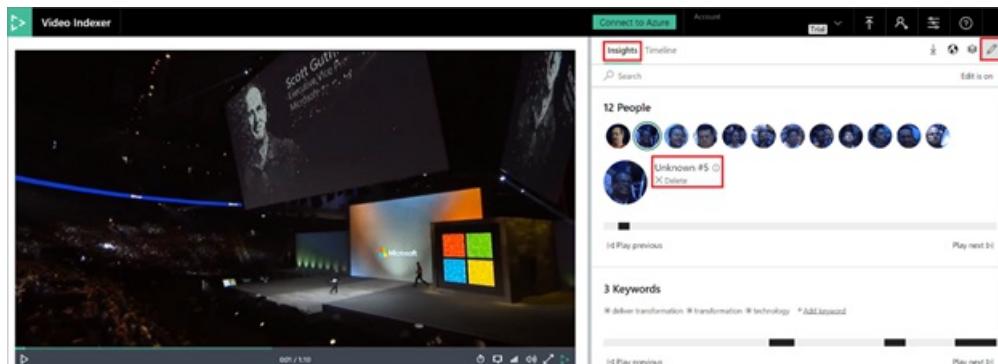
If you do not assign a Person model to the video during upload, your edit is saved in your account's Default person model.

Edit a face

NOTE

If a Person model has two or more different people with the same name, you will not be able to tag that name within the videos that use that Person model. You will only be able to make changes to people that share that name in the People tab of the content model customization page in Video Indexer. For this reason, it is recommended that you give unique names to each person in your Person model.

1. Browse to the Video Indexer website and sign in.
2. Search for a video you want to view and edit in your account.
3. To edit a face in your video, go to the Insights tab and click on the pencil icon on the top-right corner of the window.



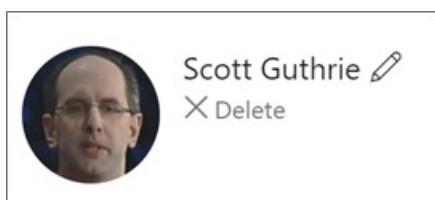
4. Click on any of the detected faces and change their names from "Unknown #X" (or the name that was previously assigned to the face).
5. After typing in the new name, click on the check icon next to the new name. This saves the new name and recognizes and names all occurrences of this face in your other current videos and in the future videos that you upload. Recognition of the face in your other current videos might take some time to take effect, as this is a batch process.

If you name a face with the name of an existing person in the Person model that the video is using, this will merge the detected face images from this video of that person with what already exists in the model. If you name a face with a completely new name, this will create a new Person entry in the Person model that the video is using.

! [Edit a face in your video] (./media/customize-face-model/edit-face2.png)

Delete a face

To delete a detected face in your video, go to the Insights pane and click on the pencil icon on the top-right corner of the pane. Click the Delete option underneath the name of the face. This will remove that detected face from the video. The person's face will still be detected in the other videos in which it appears, but you can delete the face from those videos as well after they have been indexed. The person, if it had been named, will also continue to exist in the Person model that was used to index the video from which you deleted the face unless you specifically delete the person from the Person model.



Next steps

Customize Person model using APIs

Customize a Person model with the Video Indexer API

5/15/2019 • 6 minutes to read • [Edit Online](#)

Video Indexer supports face detection and celebrity recognition for video content. The celebrity recognition feature covers approximately one million faces based on commonly requested data source such as IMDB, Wikipedia, and top LinkedIn influencers. Faces that are not recognized by the celebrity recognition feature are detected; however, they are left unnamed. After you upload your video to Video Indexer and get results back, you can go back and name the faces that were not recognized. Once you label a face with a name, the face and name get added to your account's Person model. Video Indexer will then recognize this face in your future videos and past videos.

You can use the Video Indexer API to edit faces that were detected in a video, as described in this topic. You can also use the Video Indexer website, as described in [Customize Person model using the Video Indexer website](#).

Managing multiple Person models

Video Indexer supports multiple Person models per account. This feature is currently available only through the Video Indexer APIs.

If your account caters to different use-case scenarios, you might want to create multiple Person models per account. For example, if your content is related to sports, you can then create a separate Person model for each sport (football, basketball, soccer, etc.).

Once a model is created, you can use it by providing the model ID of a specific Person model when uploading/indexing or reindexing a video. Training a new face for a video updates the specific custom model that the video was associated with.

Each account has a limit of 50 Person models. If you do not need the multiple Person model support, do not assign a Person model ID to your video when uploading/indexing or reindexing. In this case, Video Indexer uses the default custom Person model in your account.

Create a new Person model

Create a new Person model in the specified account.

Request URL

This is a POST request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels?name={name}&accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X POST "https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels?name={name}&accessToken={accessToken}"
```

[See required parameters and test out using the Video Indexer Developer Portal](#).

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
name	string	Yes	The name for the Person model
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides the name and generated model ID of the Person model that you just created following the format of the example below.

```
{
  "id": "227654b4-912c-4b92-ba4f-641d488e3720",
  "name": "Example Person Model"
}
```

You should then use the **id** value for the **personModelId** parameter when [uploading a video to index](#) or [reindexing a video](#).

Delete a Person model

Delete a custom Person model from the specified account.

Once the Person model is deleted successfully, the index of your current videos that were using the deleted model will remain unchanged until you reindex them. Upon reindexing, the faces that were named in the deleted model will not be recognized by Video Indexer in your current videos that were indexed using that model; however, those faces will still be detected. Your current videos that were indexed using the deleted model will now use your account's default Person model. If faces from the deleted model are also named in your account's default model, those faces will continue to be recognized in the videos.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels/{id}?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X DELETE
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels/{id}?accessToken={accessToken}"
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
id	string	Yes	The Person model id (generated when the Person model is created)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

There is no returned content when the Person model is deleted successfully.

Get all Person models

Get all Person models in the specified account.

Request Call

This is a GET request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X GET "https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels?accessToken={accessToken}"
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides a list of all of the Person models in your account (including the default Person model in the specified account) and each of their names and ids following the format of the example below.

```
[
  {
    "id": "59f9c326-b141-4515-abe7-7d822518571f",
    "name": "Default"
  },
  {
    "id": "9ef2632d-310a-4510-92e1-cc70ae0230d4",
    "name": "Test"
  }
]
```

You can choose which model you want to use for a video by using the **id** value of the Person model for the **personModelId** parameter when [uploading a video to index](#) or [reindexing a video](#).

Update a face

This command allows you to update a face in your video with a name using the ID of the video and ID of the face. This then updates the Person model that the video was associated with upon uploading/indexing or reindexing. If no Person model was assigned, it updates the account's default Person model.

Once this happens, it recognizes the occurrences of the same face in your other current videos that share the same Person model. Recognition of the face in your other current videos might take some time to take effect as this is a batch process.

You can update a face that Video Indexer recognized as a celebrity with a new name. The new name that you give will take precedence over the built-in celebrity recognition.

Request Call

This is a POST request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Videos/{videoId}/Index/Faces/{faceId}?
accessToken={accessToken}&newName={newName}
```

Below is the request in Curl.

```
curl -v -X PUT  
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Videos/{videoId}/Index/Faces/{faceId}?  
accessToken={accessToken}&newName={newName}"
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
videoId	string	Yes	Id for the video in which the face that you want to update appears. This is created when the video is uploaded and indexed.
faceId	integer	Yes	Id for the face that will be updated. You can get the faceId from the video index
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.
name	string	Yes	New name to update the face with.

Names are unique for Person models, so if you give two different faces in the same Person model the same **name** parameter value, Video Indexer views the faces as the same person and converges them once you reindex your video.

Request body

There is no further request body required for this call.

Response

There is no returned content when the face has been updated successfully.

Next steps

[Customize Person model using the Video Indexer website](#)

Customize a Brands model with the Video Indexer website

5/15/2019 • 3 minutes to read • [Edit Online](#)

Video Indexer supports brand detection from speech and visual text during indexing and reindexing of video and audio content. The brand detection feature identifies mentions of products, services, and companies suggested by Bing's brands database. For example, if Microsoft is mentioned in a video or audio content or if it shows up in visual text in a video, Video Indexer detects it as a brand in the content. A custom Brands model allows you to select whether or not you Video Indexer will detect brands from the Bing brands database, exclude certain brands from being detected (essentially creating a black list of brands), and include brands that should be part of your model that might not be in Bing's brands database (essentially creating a white list of brands).

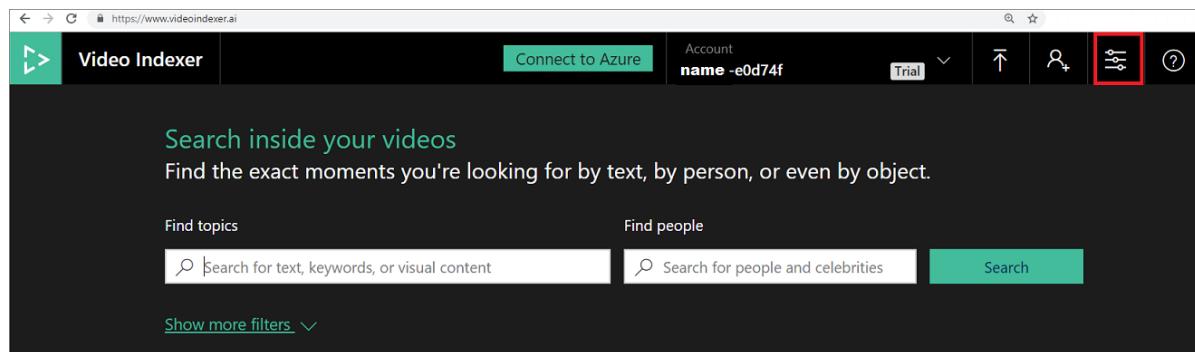
For a detailed overview, see [Overview](#).

You can use the Video Indexer website to create, use, and edit custom Brands models detected in a video, as described in this topic. You can also use the API, as described in [Customize Brands model using APIs](#).

Edit the settings of the Brands model

You have the option to set whether or not you want brands from the Bing brands database to be detected. For this, you need to edit the settings of your Brands model.

1. Browse to the [Video Indexer](#) website and sign in.
2. To customize a model in your account, click on the **Content model customization** button on the top-right corner of the page.



3. To edit brands, select the **Brands** tab.

Content model customization

Language **Brands**

Customize which brands Video Indexer recognizes by including or excluding specific brands.

Video Indexer includes brands suggested by Bing, but you can instruct the content model to ignore them and they will not be included.

Show brands suggested by Bing.

ⓘ Video Indexer includes brands suggested by Bing, but you can instruct the content model to ignore them and they will not be included.

Enter your search

Include brands + Add brand

No included brands entered

Exclude brands + Add brand

No excluded brands entered

4. Check the **Show brands suggested by Bing** option, if you want for Video Indexer to include brands suggested by Bing. Leave the option unchecked if you do not want Video Indexer to detect brands suggested by Bing in your content.

Include brands in the model

The **Include brands** section represents custom brands that you want for Video Indexer to detect even if they are not suggested by Bing.

Add a brand

1. Click "+ Add brand".

Include brands				<small>+ Add brand</small>
BRAND NAME	CATEGORY	UPDATED BY	LAST UPDATED	
Azure	Cloud			<small>trash bin icon</small> <small>up arrow icon</small>
Brand name description		Wikipedia URL		
Enter brand description here... (optional)		https://en.wikipedia.org/wiki/Microsoft_Azure		
<small>Update</small>		<small>Cancel</small>		

Provide a name (required), category (optional), description (optional), and reference URL (optional). The category field is meant to help you tag your brands. This field shows up as the brand's *tags* when using the Video Indexer APIs. For example, the brand "Azure" can be tagged or categorized as "Cloud".

The reference URL field can be any reference website for the brand such as a link to its Wikipedia page.

2. Click "Add brand" and you will see that the brand has been added to the **Include brands** list.

Edit a brand

1. Click on the pencil icon next to the brand that you want to edit.

You can update the category, description, or reference URL of a brand. You can't change the name of a

brand because names of brands are unique. If you need to change the brand name, delete the entire brand (see next section) and create a new brand with the new name.

2. Click the **Update** button to update the brand with the new information.

Delete a brand

1. Click on the trash icon next to the brand that you want to delete.
2. Click "Delete" and the brand will no longer appear in your *Include brands* list.

Exclude brands from the model

The **Exclude brands** section represents the brands that you want for Video Indexer not to detect.

Add a brand

1. Click "+ Add brand".

Provide a name (required), category (optional).

2. Click "Add brand" and you will see that the brand has been added to the *Exclude brands* list.

Edit a Brand

1. Click on the pencil icon next to the brand that you want to edit.

You can only update the category of a brand. You can't change the name of a brand because names of brands are unique. If you need to change the brand name, delete the entire brand (see next section) and create a new brand with the new name.

2. Click the **Update** button to update the brand with the new information.

Delete a Brand

1. Click on the trash icon next to the brand that you want to delete.
2. Click "Delete" and the brand will no longer appear in your *Exclude brands* list.

Next steps

[Customize Brands model using APIs](#)

Customize a Brands model with the Video Indexer API

5/15/2019 • 8 minutes to read • [Edit Online](#)

Video Indexer supports brand detection from speech and visual text during indexing and reindexing of video and audio content. The brand detection feature identifies mentions of products, services, and companies suggested by Bing's brands database. For example, if Microsoft is mentioned in a video or audio content or if it shows up in visual text in a video, Video Indexer detects it as a brand in the content. A custom Brands model allows you to exclude certain brands from being detected and include brands that should be part of your model that might not be in Bing's brands database.

For a detailed overview, see [Overview](#).

You can use the Video Indexer APIs to create, use, and edit custom Brands models detected in a video, as described in this topic. You can also use the Video Indexer website, as described in [Customize Brands model using the Video Indexer website](#).

Create a Brand

This creates a new custom brand and adds it to the custom Brands model for the specified account.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Brands?accessToken={accessToken}
```

See required parameters and test out using the [Video Indexer Developer Portal](#).

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

In addition to these parameters, you must provide a request body JSON object that provides information on the new brand following the format of the example below.

```
{
  "name": "Example",
  "enabled": true,
  "tags": ["Tag1", "Tag2"],
  "description": "This is an example",
  "referenceUrl": "https://en.wikipedia.org/wiki/Example"
}
```

Setting **enabled** to true puts the brand in the *Include* list for Video Indexer to detect. Setting **enabled** to false puts the brand in the *Exclude* list, so Video Indexer will not detect it.

The **referenceUrl** value can be any reference websites for the brand such as a link to its Wikipedia page.

The **tags** value is a list of tags for the brand. This shows up in the brand's *Category* field in the Video Indexer website. For example, the brand "Azure" can be tagged or categorized as "Cloud".

Response

The response provides information on the brand that you just created following the format of the example below.

```
{
  "referenceUrl": "https://en.wikipedia.org/wiki/Example",
  "id": 97974,
  "name": "Example",
  "accountId": "SampleAccountId",
  "lastModifierUserName": "SampleUserName",
  "created": "2018-04-25T14:59:52.7433333",
  "lastModified": "2018-04-25T14:59:52.7433333",
  "enabled": true,
  "description": "This is an example",
  "tags": [
    "Tag1",
    "Tag2"
  ]
}
```

Delete a Brand

Removes a brand from the custom Brands model for the specified account. The account is specified in the **accountId** parameter. Once called successfully, the brand will no longer be in the *Include* or *Exclude* brands lists.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Brands/{id}?accessToken={accessToken}
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .

NAME	TYPE	REQUIRED	DESCRIPTION
accountId	string	Yes	Globally unique identifier for the account
id	integer	Yes	The brand id (generated when the brand was created)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

There is no returned content when the brand is deleted successfully.

Get a specific Brand

This lets you search for the details of a brand in the custom Brands model for the specified account using the brand id.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Brands?accessToken={accessToken}
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
id	integer	Yes	The brand ID (generated when the brand was created)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides information on the brand that you searched (using brand ID) following the format of the example below.

```
{  
    "referenceUrl": "https://en.wikipedia.org/wiki/Example",  
    "id": 128846,  
    "name": "Example",  
    "accountId": "SampleAccountId",  
    "lastModifierUserName": "SampleUserName",  
    "created": "2018-01-06T13:51:38.3666667",  
    "lastModified": "2018-01-11T13:51:38.3666667",  
    "enabled": true,  
    "description": "This is an example",  
    "tags": [  
        "Tag1",  
        "Tag2"  
    ]  
}
```

NOTE

enabled being set to **true** signifies that the brand is in the *Include* list for Video Indexer to detect, and **enabled** being false signifies that the brand is in the *Exclude* list, so Video Indexer will not detect it.

Update a specific brand

This lets you search for the details of a brand in the custom Brands model for the specified account using the brand ID.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Brands/{id}?accessToken={accessToken}
```

See required parameters and test out using the [Video Indexer Developer Portal](#).

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
id	integer	Yes	The brand ID (generated when the brand was created)

NAME	TYPE	REQUIRED	DESCRIPTION
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

In addition to these parameters, you must provide a request body JSON object that provides updated information on the brand that you want to update following the format of the example below.

```
{
  "name": "Example",
  "enabled": false,
  "tags": ["Tag1", "NewTag2"],
  "description": "This is an update example",
  "referenceUrl": "https://en.wikipedia.org/wiki/Example",
  "lastModifierUserName": "SampleUserName",
  "created": "2018-04-25T14:59:52.743333",
  "lastModified": "2018-04-28T15:52:22.3413983",
}
```

NOTE

In this example the brand that was created in the example request body in the **Create a Brand** section is being updated here with a new tag and new description. The **enabled** value has also been changed to false to put it in the *Exclude* list.

Response

The response provides the updated information on the brand that you updated following the format of the example below.

```
{
  "referenceUrl": null,
  "id": 97974,
  "name": "Example",
  "accountId": "SampleAccountId",
  "lastModifierUserName": "SampleUserName",
  "Created": "2018-04-25T14:59:52.743333",
  "lastModified": "2018-04-25T15:37:50.67",
  "enabled": false,
  "description": "This is an update example",
  "tags": [
    "Tag1",
    "NewTag2"
  ]
}
```

Get all of the Brands

This returns all of the brands in the custom Brands model for the specified account regardless of whether the brand is meant to be in the *Include* or *Exclude* brands list.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Brands?accessToken={accessToken}
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides a list of all of the brands in your account and each of their details following the format of the example below.

```
[  
  {  
    "ReferenceUrl": null,  
    "id": 97974,  
    "name": "Example",  
    "accountId": "AccountId",  
    "lastModifierUserName": "UserName",  
    "Created": "2018-04-25T14:59:52.7433333",  
    "LastModified": "2018-04-25T14:59:52.7433333",  
    "enabled": true,  
    "description": "This is an example",  
    "tags": ["Tag1", "Tag2"]  
},  
  {  
    "ReferenceUrl": null,  
    "id": 97975,  
    "name": "Example2",  

```

NOTE

The brand named *Example* is in the *Include* list for Video Indexer to detect, and the brand named *Example2* is in the *Exclude* list, so Video Indexer will not detect it.

Get Brands model settings

This returns the Brands model settings in the specified account. The Brands model settings represent whether detection from the Bing brands database is enabled or not. If Bing brands are not enabled, Video Indexer will only detect brands from the custom Brands model of the specified account.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Brands?accessToken={accessToken}
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response shows whether Bing brands are enabled following the format of the example below.

```
{  
  "state": true,  
  "useBuiltIn": true  
}
```

NOTE

useBuiltIn being set to true represents that Bing brands are enabled. If *useBuiltIn* is false, Bing brands are disabled. The **state** value can be ignored as it has been deprecated.

Update Brands model settings

This updates the Brands model settings in the specified account. The Brands model settings represent whether detection from the Bing brands database is enabled or not. If Bing brands are not enabled, Video Indexer will only detect brands from the custom Brands model of the specified account.

Request URL:

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/BrandsModelSettings?accessToken={accessToken}
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

In addition to these parameters, you must provide a request body JSON object that provides information on the new brand following the format of the example below.

```
{  
    "useBuiltIn":true  
}
```

NOTE

useBuiltIn being set to true represents that Bing brands are enabled. If *useBuiltin* is false, Bing brands are disabled.

Response

There is no returned content when the Brands model setting is updated successfully.

Next steps

[Customize Brands model using website](#)

Customize a Language model with the Video Indexer website

5/15/2019 • 5 minutes to read • [Edit Online](#)

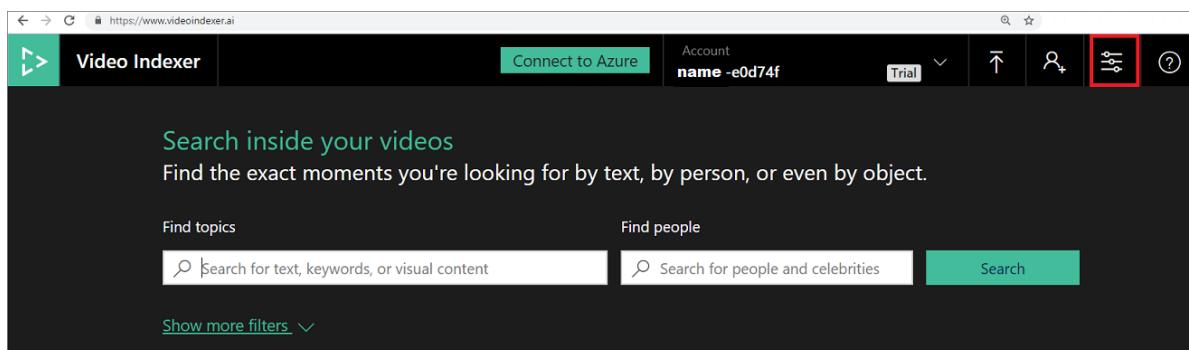
Video Indexer lets you create custom Language models to customize speech recognition by uploading adaptation text, namely text from the domain whose vocabulary you'd like the engine to adapt to. Once you train your model, new words appearing in the adaptation text will be recognized.

For a detailed overview and best practices for custom language models, see [Customize a Language model with Video Indexer](#).

You can use the Video Indexer website to create and edit custom Language models in your account, as described in this topic. You can also use the API, as described in [Customize Language model using APIs](#).

Create a Language model

1. Browse to the [Video Indexer](#) website and sign in.
2. To customize a model in your account, click on the **Content model customization** button on the top-right corner of the page.



3. Select the **Language** tab.

A screenshot of the "Content model customization" page. The title "Content model customization" is at the top. Below it, there are two tabs: "Language" (which is highlighted with a red box) and "Brands". Under the "Language" tab, there's a paragraph about customizing the language model by adding .txt files. It includes a link to read more about the capability. Below this, there's a "Browse for model files..." button with a magnifying glass icon. A dropdown menu shows "English" with a downward arrow, and a "Add model" button with a plus sign. A red box highlights the "Language" tab in the navigation bar.

4. Under the language that you want, click **Add model**.

5. Type in the name for the Language model and hit enter.

This creates the model and gives the option to upload text files to the model.

6. To add a text file, click **Add file**. This opens up your file explorer.

7. Navigate to and select the text file. You can add multiple text files to a Language model.

You can also add a text file by clicking on the ... button on the right side of the Language model and selecting **Add file**.

8. Once you are done uploading the text files, click on the green **Train** option.

The screenshot shows a list of text files under the 'Sports' model. The 'Train' button is highlighted with a red box. Other options include 'Tennis.txt', 'Baseball.txt', and a '+ Add file' button.

The training process can take a few minutes. Once the training is done, you see **Trained** next to the model. You can preview, download, and delete the file from the model.

The screenshot shows the 'Sports' model has been trained. A context menu is open over the model, listing options: 'Rename' (with a pencil icon), 'Delete' (with a trash bin icon), '+ Add file', and 'Details'. The 'Details' section shows the Linguistic model ID: ffb13e73-e91c-4973-8dfc-75bfe8.

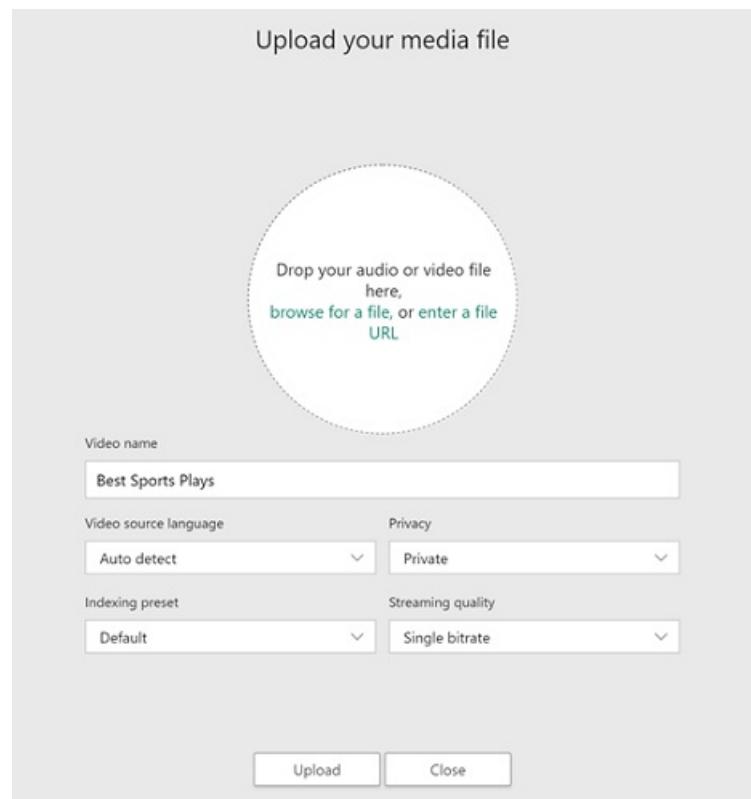
Using a Language model on a new video

To use your Language model on a new video, do one of the following:

- Click on the **Upload** button on the top of the page



- Drop your audio or video file in the circle or browse for your file



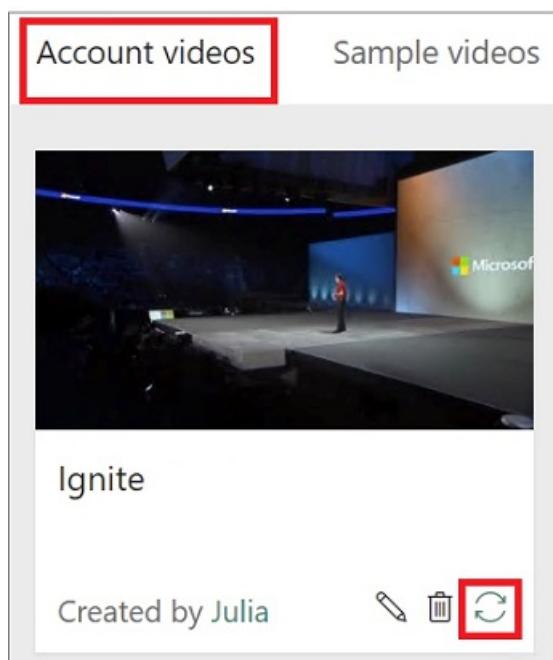
This will give you the option to select the **Video source language**. Click on the drop-down and select a Language model that you created from the list. It should say the language of your Language model and the name that you gave it in parentheses.

Click the **Upload** option in the bottom of the page, and your new video will be indexed using your Language model.

Using a Language model to reindex

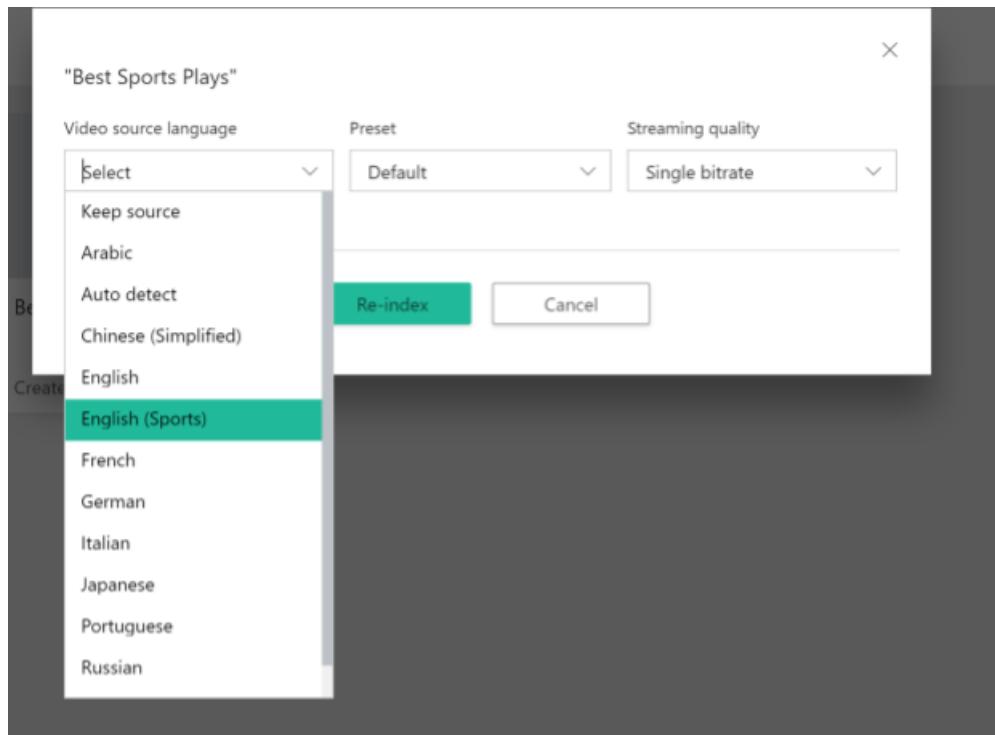
To use your Language model to reindex a video in your collection, go to your **Account videos** on the [Video Indexer](#) home page and hover over the name of the video that you want to reindex.

You see options to edit your video, delete you video, and reindex your video. Click on the option to reindex your video.



This gives you the option to select the **Video source language** to reindex your video with. Click on the drop-down and select a Language model that you created from the list. It should say the language of your language

model and the name that you gave it in parentheses.



Click the **Re-index** button, and your video will be reindexed using your Language model.

Edit a Language model

You can edit a Language model by changing its name, adding files to it, and deleting files from it.

If you add or delete files from the Language model, you will have to train the model again by clicking in the green **Train** option.

Rename the Language model

You can change the name of the Language model by clicking ... on the right side of the Language model and selecting **Rename**.

Type in the new name and hit enter.

Add files

To add a text file, click **Add file**. This opens up your file explorer.

Navigate to and select the text file. You can add multiple text files to a Language model.

You can also add a text file by clicking on the ... button on the right side of the Language model and selecting **Add file**.

Delete files

To delete a file from the Language model, click the ... button on the right side of the text file and select **Delete**. This brings up a new window telling you that the deletion cannot be undone. Click the **Delete** option in the new window.

This action removes the file completely from the Language model.

Delete a Language model

To delete a Language model from your account, click the ... button on the right side of the Language model and select **Delete**.

This brings up a new window telling you that the deletion cannot be undone. Click the **Delete** option in the new window.

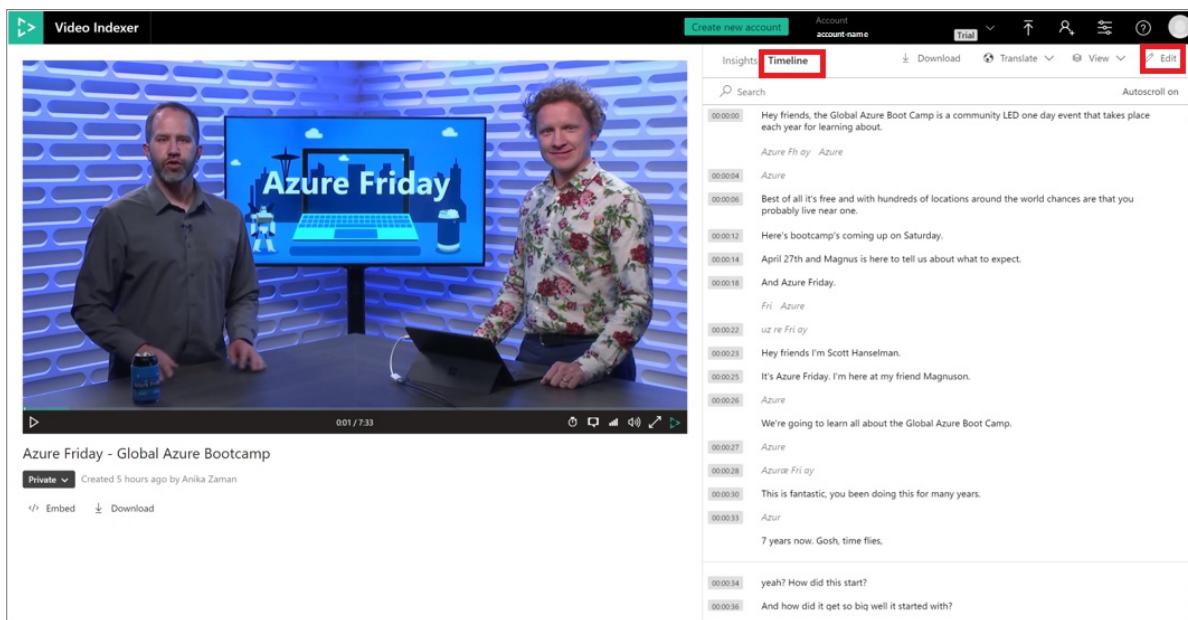
This action removes the Language model completely from your account. Any video that was using the deleted Language model will keep the same index until you re-index the video. If you re-index the video, you can assign a new Language model to the video. Otherwise, Video Indexer will use its default model to re-index the video.

Customize Language models by correcting transcripts

Video Indexer supports automatic customization of Language models based on the actual corrections users make to the transcriptions of their videos.

1. To make corrections to a transcript, open up the video that you want to edit from your Account Videos.

Select the **Timeline** tab.



The screenshot shows the Video Indexer web interface. At the top, there's a navigation bar with 'Create new account', 'Account' (account-name), 'Trial', and other icons. Below the navigation is a video player showing two men at a desk. The video title is 'Azure Friday - Global Azure Bootcamp'. Underneath the video player, it says 'Created 5 hours ago by Anika Zaman'. Below that are buttons for 'Private', 'Embed', and 'Download'. On the right side of the screen, there's a 'Timeline' tab which is highlighted with a red box. Below the timeline tab, there's a search bar and an 'Edit' button. The main area shows a list of subtitles with timestamps from 00:00:00 to 00:00:36. Each subtitle includes the speaker's name and the spoken text. For example, at 00:00:00, it says 'Hey friends, the Global Azure Boot Camp is a community LED one day event that takes place each year for learning about.' At 00:00:12, it says 'Here's bootcamp's coming up on Saturday.' At 00:00:25, it says 'It's Azure Friday. I'm here at my friend Magnuson.' At 00:00:34, it says 'yeah? How did this start?' and at 00:00:36, it says 'And how did it get so big well it started with?'

2. Click on the pencil icon to edit the transcript of your transcription.

The screenshot shows the Video Indexer interface with the 'Timeline' tab selected. A transcript of a video is displayed, with various lines of text corresponding to specific time points. A callout box highlights the line 'on Azure Friday.' at 00:00:18. The callout box contains the text: 'Changes you make here will be added to your language model as a 'From transcript edits' file.' There is also a checkbox labeled 'Don't show this again'.

Video Indexer captures all lines that are corrected by you in the transcription of your video and adds them automatically to a text file called "From transcript edits". These edits are used to re-train the specific Language model that was used to index this video.

If you did not specify a Language model when indexing this video, then all edits for this video will be stored in a default Language model called Account adaptations within the detected language of the video.

In case multiple edits have been made to the same line, only the last version of the corrected line will be used for updating the Language model.

NOTE

Only textual corrections are used for the customization. This means that corrections that do not involve actual words (for example, punctuation marks or spaces) are not included.

3. You will see transcript corrections show up in the Language tab of the Content model customization page.

The screenshot shows the 'Content model customization' page with the 'Language' tab selected. It displays a section for 'Account adaptations' under the 'English' language model. The 'From transcript edits' section is visible, showing a list of files and options to 'Train' or '...' the model. Other language models like 'German' are listed below.

To look at the "From transcript edits" file for each of your Language models, click on it to open it.

You can look at the [From transcript edits](#) file for each of your Language models by clicking on it to open it.



From transcript edits



Edit made on: '3de26cde9a'

Hey friends, the Global Azure Boot Camp is a community ~~LED~~ one day event that takes place each year for learning about.

Hey friends, the Global Azure Boot Camp is a community led one day event that takes place each year for learning about.

Here's ~~bootcamp~~'s coming up on Saturday.

This year's bootcamp is coming up on Saturday.

April 27th and Magnus is here to tell us about what to ~~expect~~.

April 27th and Magnus is here to tell us about what to expect today.

And Azure Friday.

on Azure Friday.

It's Azure Friday. I'm here at my friend ~~Magnus~~:

It's Azure Friday. I'm here at my friend Magnus and

Next steps

[Customize language model using APIs](#)

Customize a Language model with the Video Indexer APIs

5/15/2019 • 13 minutes to read • [Edit Online](#)

Video Indexer lets you create custom Language models to customize speech recognition by uploading adaptation text, namely text from the domain whose vocabulary you'd like the engine to adapt to. Once you train your model, new words appearing in the adaptation text will be recognized.

For a detailed overview and best practices for custom Language models, see [Customize a Language model with Video Indexer](#).

You can use the Video Indexer APIs to create and edit custom Language models in your account, as described in this topic. You can also use the website, as described in [Customize Language model using the Video Indexer website](#).

Create a Language model

The following command creates a new custom Language model in the specified account. You can upload files for the Language model in this call. Alternatively, you can create the Language model here and upload files for the model later by updating the Language model.

NOTE

You must still train the model with its enabled files for the model to learn the contents of its files. Directions on training a language are in the next section.

Request URL

This is a POST request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels?name={name}&accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X POST "https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language?accessToken={accessToken}&modelName={modelName}&language={language}"  
--data-ascii "{body}"
```

[See required parameters and test out using the Video Indexer Developer Portal](#).

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .

NAME	TYPE	REQUIRED	DESCRIPTION
accountId	string	Yes	Globally unique identifier for the account
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.
modelName	string	Yes	The name for the Language model
language	string	Yes	The language of the Language model. The language parameter must be given the language in BCP-47 format of 'language tag-region' (e.g: 'en-US'). Supported languages are English (en-US), German (de-DE), Spanish (es-SP), Arabic (ar-EG), French (fr-FR), Hindi (hi-HI), Italian (it-IT), Japanese (ja-JP), Portuguese (pt-BR), Russian (ru-RU), and Chinese (zh-CN).

Request body

To upload files to be added to the Language model, you must upload files in the body using form-data in addition to providing values for the required parameters above. There are two ways to do this:

1. Key will be the file name and value will be the txt file
2. Key will be the file name and value will be a URL to txt file

Response

The response provides metadata on the newly created Language model along with metadata on each of the model's files following the format of the example JSON output.

```
{
  "id": "dfaef5745-6f1d-4edd-b224-42e1ab57a891",
  "name": "TestModel",
  "language": "En-US",
  "state": "None",
  "languageModelId": "00000000-0000-0000-0000-000000000000",
  "files": [
    {
      "id": "25be7c0e-b6a6-4f48-b981-497e920a0bc9",
      "name": "hellofile",
      "enable": true,
      "creator": "John Doe",
      "creationTime": "2018-04-28T11:55:34.6733333"
    },
    {
      "id": "33025f5b-2354-485e-a50c-4e6b76345ca7",
      "name": "worldfile",
      "enable": true,
      "creator": "John Doe",
      "creationTime": "2018-04-28T11:55:34.86"
    }
  ]
}
```

Train a Language model

The following command trains a custom Language model in the specified account with the contents in the files that were uploaded to and enabled in the language model.

NOTE

You must first create the Language model and upload its files. You can upload files either when creating the language model or by updating the Language model.

Request URL

This is a PUT request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Train?
accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X PUT
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Train?
accessToken={accessToken}"
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
------	------	----------	-------------

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
modelId	string	Yes	The language model id (generated when the Language model is created)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides metadata on the newly trained Language model along with metadata on each of the model's files following the format of the example JSON output.

```
{
  "id": "41464adf-e432-42b1-8e09-f52905d7e29d",
  "name": "TestModel",
  "language": "En-US",
  "state": "Waiting",
  "languageModelId": "531e5745-681d-4e1d-b124-12e5ab57a891",
  "files": [
    {
      "id": "84fcf1ac-1952-48f3-b372-18f768eedf83",
      "name": "RenamedFile",
      "enable": false,
      "creator": "John Doe",
      "creationTime": "2018-04-27T20:10:10.5233333"
    },
    {
      "id": "9ac35b4b-1381-49c4-9fe4-8234bfdd0f50",
      "name": "hellofile",
      "enable": true,
      "creator": "John Doe",
      "creationTime": "2018-04-27T20:10:10.68"
    }
  ]
}
```

You should then use the **id** value of the language model for the **linguisticModelId** parameter when [uploading a video to index](#) and for the **languageModelId** parameter when [reindexing a video](#).

Delete a Language model

The following command deletes a custom Language model from the specified account. Any video that was using the deleted Language model will keep the same index until you re-index the video. If you re-index the video, you

can assign a new Language model to the video. Otherwise, Video Indexer will use its default model to re-index the video.

Request URL

This is a DELETE request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X DELETE  
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}?accessToken={accessToken}"
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
modelId	string	Yes	The Language model id (generated when the Language model is created)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

There is no returned content when the Language model is deleted successfully.

Update a Language model

The following command updates a custom Language person model in the specified account.

NOTE

You must have already created the Language model. You can use this call to enable or disable all files under the model, update the name of the Language model, and upload files to be added to the language model.

Request URL

This is a PUT request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}?accessToken={accessToken} [&modelName][&enable]
```

Below is the request in Curl.

```
curl -v -X PUT "https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}?accessToken={accessToken}&modelName={string}&enable={string}"  
--data-ascii "{body}"
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
modelId	string	Yes	The Language model id (generated when the Language model is created)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.
modelName	string	No	New name that you can give to the model
enable	boolean	No	Choose whether all files under this model are enabled (true) or disabled (false)

Request body

To upload files to be added to the Language model, you must upload files in the body using form-data in addition to providing values for the required parameters above. There are two ways to do this:

1. Key will be the file name and value will be the txt file
2. Key will be the file name and value will be a URL to txt file

Response

The response provides metadata on the newly trained Language model along with metadata on each of the model's files following the format of the example JSON output.

```
{
  "id": "41464adf-e432-42b1-8e09-f52905d7e29d",
  "name": "TestModel",
  "language": "En-US",
  "state": "Waiting",
  "languageModelId": "531e5745-681d-4e1d-b124-12e5ab57a891",
  "files": [
    {
      "id": "84fcf1ac-1952-48f3-b372-18f768eedf83",
      "name": "RenamedFile",
      "enable": true,
      "creator": "John Doe",
      "creationTime": "2018-04-27T20:10:10.5233333"
    },
    {
      "id": "9ac35b4b-1381-49c4-9fe4-8234bfdd0f50",
      "name": "hellofile",
      "enable": true,
      "creator": "John Doe",
      "creationTime": "2018-04-27T20:10:10.68"
    }
  ]
}
```

You can use the **id** of the files returned here to download the contents of the file.

Update a file from a Language model

The following command allows you to update the name and **enable** state of a file in a custom Language model in the specified account.

Request URL

This is a PUT request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Files/{ fileId}?accessToken={accessToken} [&fileName][&enable]
```

Below is the request in Curl.

```
curl -v -X PUT
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Files/{ fileId}?accessToken={accessToken}?fileName={string}&enable={string}"
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account

NAME	TYPE	REQUIRED	DESCRIPTION
modelId	string	Yes	Id of the Language model that holds the file (generated when the Language model is created)
fileId	string	Yes	Id of the file that is being updated (generated when the file is uploaded at the creation or updating of the Language model)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.
fileName	string	No	Name to update the file name to
enable	boolean	No	Update whether this file is enabled (true) or disabled (false) in the language model

Request body

There is no further request body required for this call.

Response

The response provides metadata on the file that you updated following the format of the example JSON output below.

```
{
  "id": "84fcf1ac-1952-48f3-b372-18f768eedf83",
  "name": "RenamedFile",
  "enable": false,
  "creator": "John Doe",
  "creationTime": "2018-04-27T20:10:10.523333"
}
```

You can use the **id** of the file returned here to download the contents of the file.

Get a specific Language model

The following command returns information on the specified Language model in the specified account such as language and the files that are in the Language model.

Request URL

This is a GET request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X GET "https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}?accessToken={accessToken}"
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters and request body

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
modelId	string	Yes	The Language model id (generated when the Language model is created)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides metadata on the specified Language model along with metadata on each of the model's files following the format of the example JSON output below.

```
{
  "id": "dfae5745-6f1d-4edd-b224-42e1ab57a891",
  "name": "TestModel",
  "language": "En-US",
  "state": "None",
  "languageModelId": "00000000-0000-0000-0000-000000000000",
  "files": [
    {
      "id": "25be7c0e-b6a6-4f48-b981-497e920a0bc9",
      "name": "hellofile",
      "enable": true,
      "creator": "John Doe",
      "creationTime": "2018-04-28T11:55:34.6733333"
    },
    {
      "id": "33025f5b-2354-485e-a50c-4e6b76345ca7",
      "name": "worldfile",
      "enable": true,
      "creator": "John Doe",
      "creationTime": "2018-04-28T11:55:34.86"
    }
  ]
}
```

You can use the **id** of the file returned here to download the contents of the file.

Get all the Language models

The following command returns all of the custom Language models in the specified account in a list.

Request URL

This is a GET Request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X GET "https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language?  
accessToken={accessToken}"
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides a list of all of the Language models in your account and each of their metadata and files following the format of the example JSON output below.

```
[
  {
    "id": "dfaef5745-6f1d-4edd-b224-42e1ab57a891",
    "name": "TestModel",
    "language": "En-US",
    "state": "None",
    "languageModelId": "00000000-0000-0000-0000-000000000000",
    "files": [
      {
        "id": "25be7c0e-b6a6-4f48-b981-497e920a0bc9",
        "name": "hellofile",
        "enable": true,
        "creator": "John Doe",
        "creationTime": "2018-04-28T11:55:34.6733333"
      },
      {
        "id": "33025f5b-2354-485e-a50c-4e6b76345ca7",
        "name": "worldfile",
        "enable": true,
        "creator": "John Doe",
        "creationTime": "2018-04-28T11:55:34.86"
      }
    ]
  },
  {
    "id": "dfaef5745-6f1d-4edd-b224-42e1ab57a892",
    "name": "AnotherTestModel",
    "language": "En-US",
    "state": "None",
    "languageModelId": "00000000-0000-0000-0000-000000000001",
    "files": []
  }
]
```

Delete a file from a Language model

The following command deletes the specified file from the specified Language model in the specified account.

Request URL

This is a DELETE request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Files/{ fileId}?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X DELETE
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Files/{ fileId}?accessToken={accessToken}"
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
------	------	----------	-------------

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
modelId	string	Yes	Id of the Language model that holds the file (generated when the Language model is created)
fileId	string	Yes	Id of the file that is being updated (generated when the file is uploaded at the creation or updating of the Language model)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

There is no returned content when the file is deleted from the Language model successfully.

Get metadata on a file from a Language model

This returns the contents of and metadata on the specified file from the chosen Language model in the your account.

Request URL

This is a GET request.

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/PersonModels?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X GET
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Files/{fileId}?accessToken={accessToken}"
```

[See required parameters and test out using the Video Indexer Developer Portal.](#)

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
modelId	string	Yes	Id of the language model that holds the file (generated when the language model is created)
fileId	string	Yes	Id of the file that is being updated (generated when the file is uploaded at the creation or updating of the language model)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response provides the contents and metadata of the file in JSON format, similar to this:

```
{
  "content": "hello\r\nworld",
  "id": "84fcf1ac-1952-48f3-b372-18f768eedf83",
  "name": "Hello",
  "enable": true,
  "creator": "John Doe",
  "creationTime": "2018-04-27T20:10:10.5233333"
}
```

NOTE

The contents of this example file are the words "hello" and "world" in two separate lines.

Download a File from a Language model

The following command downloads a text file containing the contents of the specified file from the specified Language model in the specified account. This text file should match the contents of the text file that was originally uploaded.

Request URL

```
https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Files/{fileId}/download?accessToken={accessToken}
```

Below is the request in Curl.

```
curl -v -X GET  
"https://api.videoindexer.ai/{location}/Accounts/{accountId}/Customization/Language/{modelId}/Files/{fileId}/download?accessToken={accessToken}"
```

See required parameters and test out using the Video Indexer Developer Portal.

Request parameters

NAME	TYPE	REQUIRED	DESCRIPTION
location	string	Yes	The Azure region to which the call should be routed. For more information, see Azure regions and Video Indexer .
accountId	string	Yes	Globally unique identifier for the account
modelId	string	Yes	Id of the Language model that holds the file (generated when the Language model is created)
fileId	string	Yes	Id of the file that is being updated (generated when the file is uploaded at the creation or updating of the Language model)
accessToken	string	Yes	Access token (must be of scope Account Access Token) to authenticate against the call. Access tokens expire within 1 hour.

Request body

There is no further request body required for this call.

Response

The response will be the download of a text file with the contents of the file in the JSON format.

Next steps

[Customize Language model using website](#)

Azure regions in which Video Indexer exists

5/15/2019 • 2 minutes to read • [Edit Online](#)

Video Indexer APIs contain a **location** parameter that you should set to the Azure region to which the call should be routed. This must be an [Azure region in which Video Indexer is available](#).

Locations

The **location** parameter must be given the Azure region code name as its value. If you are using Video Indexer in preview mode, you should put "trial" as the value. Otherwise, to get the code name of the Azure region that your account is in and that your call should be routed to, you can run the following line in [Azure CLI](#):

```
az account list-locations
```

Once you run the line shown above, you get a list of all Azure regions. Navigate to the Azure region that has the *displayName* you are looking for, and use its *name* value for the **location** parameter.

For example, for the Azure region West US 2 (displayed below), you will use "westus2" for the **location** parameter.

```
{
  "displayName": "West US 2",
  "id": "/subscriptions/35c2594a-23da-4fce-b59c-f6fb9513eeeb/locations/westus2",
  "latitude": "47.233",
  "longitude": "-119.852",
  "name": "westus2",
  "subscriptionId": null
}
```

Next steps

- [Customize Language model using APIs](#)
- [Customize Brands model using APIs](#)
- [Customize Person model using APIs](#)

Frequently asked questions

5/15/2019 • 9 minutes to read • [Edit Online](#)

This article answers frequently asked questions about Video Indexer.

General questions

What is Video Indexer?

Video Indexer is an artificial intelligence service that is part of Microsoft Azure Media Services. Video Indexer provides an orchestration of multiple machine learning models that enable you to easily extract deep insight from a video. To provide advanced and accurate insights, Video Indexer makes use of multiple channels of the video: audio, speech, and visual. Video Indexer's insights may be used in many ways, like improving content discoverability and accessibility, creating new monetization opportunities, or building new experiences that use the insights. Video Indexer provides a web-based interface for testing, configuration, and customization of models in your account. Developers can use a REST-based API to integrate Video Indexer into production system.

What can I do with Video Indexer?

Some of the operations that Video Indexer can perform on media files include:

- Identifying and extracting speech and identify speakers.
- Identifying and extracting on-screen text in a video.
- Detecting objects in a video file.
- Identify brands (for example: Microsoft) from audio tracks and on-screen text in a video.
- Detecting and recognizing faces from a database of celebrities and a user-defined database of faces.
- Extracting topics discussed but not necessarily mentioned in audio and video content.
- Creating closed captions or subtitles from the audio track.

For more information and more Video Indexer features, see [Overview](#).

How do I get started with Video Indexer?

Video Indexer includes a free trial offering that provides you with 600 minutes in the web-based interface and 2,400 minutes via the API. You can [login to the Video Indexer web-based interface](#) and try it for yourself using any web identity and without having to set up an Azure Subscription.

To index videos and audio files at scale, you can connect Video Indexer to a paid Microsoft Azure subscription. You can find more information on pricing on the [pricing](#) page.

You can find more information on getting started in [Get started](#).

Do I need coding skills to use Video Indexer?

You can use the Video Indexer web-based interface to evaluate, configure, and manage your account with **no coding required**. When you are ready to develop more complex applications, you can use the [Video Indexer API](#) to integrate Video Indexer into your own applications, web sites, or [custom workflows using serverless technologies like Azure Logic Apps](#) or Azure Functions.

Do I need machine learning skills to use Video Indexer?

No, Video Indexer provides the integration of multiple machine learning models into one pipeline. Indexing a video or audio file via Video Indexer retrieves a full set of insights extracted on one shared timeline without any machine learning skills or knowledge on algorithms needed on the customer's part.

What media formats does Video Indexer support?

Video Indexer supports most common media formats. Refer to the [Azure Media Encoder standard formats](#) list for more details.

How to do I upload a media into Video Indexer?

In the Video Indexer web-based portal, you can upload a media file using the file upload dialog or by pointing to a URL that directly hosts the source file (see [example](#)). Any URL that hosts the media content using an iFrame or embed code will not work (see [example](#)). The Video Indexer API requires you to specify the input file via a URL or a byte array. Uploads via a URL using the API are limited to 10 GB, but do not have a time duration limit. For more information, please see this [how-to guide](#).

How long does it take Video Indexer to extract insights from media?

The amount of time it takes to index a video or audio file, both using the Video Indexer API and the Video Indexer web-based interface, depends on multiple parameters such as the file length and quality, the number of insights found in the file, the number of [reserved units](#) available, and whether the [streaming endpoint](#) is enabled or not. We recommend that you run a few test files with your own content and take an average to get a better idea.

Can I create customized workflows to automate processes with Video Indexer?

Yes, you can integrate Video Indexer into serverless technologies like Logic Apps, Flow, and [Azure Functions](#). You can find more details on the [Logic App](#) and [Flow](#) connectors for Video Indexer [here](#).

In which Azure regions is Video indexer available?

You can see which Azure regions Video Indexer is available on the [regions](#) page.

What is the SLA for Video Indexer?

Azure Media Service's SLA covers Video Indexer and can be found on the [SLA](#) page. The SLA only applies to Video Indexer paid accounts and does not apply to the free trial.

Privacy Questions

Are video and audio files indexed by Video Indexer stored?

Yes, unless you delete the file from Video Indexer, either using the Video Indexer website or API, your video and audio files are stored. For the free trial, the video and audio files that you index are stored in the Azure region East US. Otherwise, your video and audio files are stored in the storage account of your Azure subscription.

Can I delete my files that are stored in Video Indexer Portal?

Yes, you can always delete your video and audio files as well as any metadata and insights extracted from them by Video Indexer. Once you delete a file from Video Indexer, the file and its metadata and insights are permanently removed from Video Indexer. However, if you have implemented your own backup solution in Azure storage, the file remains in your Azure storage.

Can I control user access to my Video Indexer account?

Yes, only account admins can invite and uninvite people to their accounts, as well as assign who has editing privileges and who has read-only access.

Who has access to my video and audio files that have been indexed and/or stored by Video Indexer and the metadata and insights that were extracted?

Your video or audio content that have public as its privacy setting can be accessed by anyone who has the link to your video or audio content and its insights. Your video or audio content that have private as its privacy setting can only be accessed by users that were invited to the account of the video or audio content. The privacy setting of your content also applies to the metadata and insights that Video Indexer extracts. You assign the privacy setting when you upload your video or audio file. You can also change the privacy setting after indexing.

What access does Microsoft have to my video or audio files that have been indexed and/or stored by Video Indexer and the metadata and insights that were extracted?

Per the [Azure Online Services Terms](#) (OST), you completely own your content, and Microsoft will only access your content and the metadata and insights that Video Indexer extracts from your content according to the OST and the Microsoft Privacy Statement.

Are the custom models that I build in my Video Indexer account available to other accounts?

No, the custom models that you create in your account are not available to any other account. Video Indexer currently allows you to build custom [brands](#), [language](#), and [person](#) models in your account. These models are only available in the account in which you created the models.

Is the content indexed by Video Indexer kept within the Azure region where I am using Video Indexer?

Yes, the content and its insights are kept within the Azure region unless you have a manual configuration in your Azure subscription that uses multiple Azure regions.

What is the Privacy policy for Video Indexer?

Video Indexer is covered by the [Microsoft Privacy Statement](#). The privacy statement explains the personal data Microsoft processes, how Microsoft processes it, and for what purposes Microsoft processes it. To learn more about privacy, visit the [Microsoft Trust Center](#).

What certifications does Video Indexer have?

Video Indexer currently has the SOC certification. To review Video Indexer's certification, please refer to the [Microsoft Trust Center](#).

API Questions

What APIs does Video Indexer offer?

Video Indexer's APIs allow for indexing, extracting metadata, asset management, translation, embedding, customization of models and more. To find more detailed information on using the Video Indexer API, refer to the [Video Indexer Developer Portal](#).

What client SDKs does Video Indexer offer?

There are currently no client SDKs offered. The Video Indexer team is working on the SDKs and plans to deliver them soon.

How do I get started with Video Indexer's API?

Follow [Tutorial: get started with the Video Indexer API](#).

What is the difference between the Video Indexer API and the Azure Media Service v3 API?

Currently there are some overlaps in features offered by the Video Indexer API and the Azure Media Service v3 API. You can find more information on how to compare both services [here](#).

What is an API access token and why do I need it?

The Video Indexer API contains an Authorization API and an Operations API. The Authorizations API contains calls that give you access token. Each call to the Operations API should be associated with an access token, matching the authorization scope of the call.

Access tokens are needed to use the Video Indexer APIs for security purposes. This ensures that any calls are coming from you or those who have access permissions to your account.

What is the difference between Account access token, User access token, and Video access token?

- Account level – account level access tokens let you perform operations on the account level or the video level. For example, upload a video, list all videos, get video insights.
- User level - user level access tokens let you perform operations on the user level. For example, get associated accounts.
- Video level – video level access tokens let you perform operations on a specific video. For example, get video

insights, download captions, get widgets, etc.

How often do I need to get a new access token? When do access tokens expire?

Access tokens expire every hour, so you need to generate a new access token every hour.

Billing questions

How much does Video Indexer cost?

Video Indexer uses a simple pay-as-you-go pricing model based on the duration of the content input that you index. Additional charges may apply for encoding, streaming, storage, network usage, and media reserved units. For more information, see the [pricing](#) page.

When am I billed for using Video Indexer?

When sending a video to be indexed, the user will define the indexing to be video analysis, audio analysis or both. This will determine which SKUs will be charged. If there is a critical level error during processing, an error code will be returned as a response. In such a case, no billing occurs. A critical error can be caused by a bug in our code or a critical failure in an internal dependency the service has. Errors such as wrong identification or insight extraction are not considered as critical and a response is returned. In any case where a valid (non-error code) response is returned, billing occurs.

Does Video Indexer offer a free trial?

Yes, Video Indexer offers a free trial that gives full service and API functionality. There is a quota of 600 minutes worth of videos for web-based interface users and 2,400 minutes for API users.

Next steps

[Overview](#)

Azure Media Services Video Indexer release notes

6/26/2019 • 2 minutes to read • [Edit Online](#)

To stay up-to-date with the most recent developments, this article provides you with information about:

- The latest releases
- Known issues
- Bug fixes
- Deprecated functionality

June 2019

Video Indexer deployed to Japan East

You can now create a Video Indexer paid account in the Japan East region.

Create and repair account API (Preview)

Added a new API that enables you to [update the Azure Media Service connection endpoint or key](#).

Improve error handling on upload

A descriptive message is returned in case of misconfiguration of the underlying Azure Media Services account.

Player timeline Keyframes preview

You can now see an image preview for each time on the player's timeline.

Editor semi-select

You can now see a preview of all the insights that are selected as a result of choosing a specific insight timeframe in the editor.

May 2019

Update custom language model from closed caption file

[Create custom language model](#) and [Update custom language models](#) APIs now support VTT, SRT, and TTML file formats as input for language models.

When calling the [Update Video transcript API](#), the transcript is added automatically. The training model associated with the video is updated automatically as well. For information on how to customize and train your language models, see [Customize a Language model with Video Indexer](#).

New download transcript formats – TXT and CSV

In addition to the closed captioning format already supported (SRT, VTT, and TTML), Video Indexer now supports downloading the transcript in TXT and CSV formats.

Next steps

[Overview](#)